Assurance Activity Report for
Ivanti Endpoint Manager Mobile (EPMM) 12

Ivanti Endpoint Manager Mobile (EPMM) System 12 Security Target
Version 3.5

Protection Profile for Mobile Device Management, Version 4.0, 25 April 2019 (PP_MDM_V4.0)
PP-Module for MDM Agents, Version 1.0, 25 April 2019 (MOD_MDM_AGENT_V1.0)
Functional Package for Transport Layer Security (TLS), Version 1.1, 01 March 2019 [PKG_TLS_V1.1]

AAR Version 0.4, 10/13/2025

Evaluated by:

2400 Research Blvd, Suite 395
Rockville, MD 20850

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:

Ivanti, Inc.

The Author of the Security Target:
Intertek Acumen Security

The TOE Evaluation was Sponsored by:
Ivanti, Inc.
10377 South Jordan Gateway, Suite 110 South Jordan, Utah 84095

Evaluation Personnel:
Elliot Keen
Eric Isaac
Shyam Sundar Krishnamurthy

Common Criteria Version
Common Criteria Version 3.1 Revision 5

Common Evaluation Methodology Version
CEM Version 3.1 Revision 5

REVISION HISTORY

VERSION	DATE	CHANGES
0.1	04/18/2025	Initial Release
0.2	08/13/2025	Second Release
0.3	09/10/2025	Revised document : Addressed inconsistencies
0.4	10/13/2025	Addressed ECR comments

CONTENTS

1 TOE OVERVIEW 17

2 ASSURANCE ACTIVITIES IDENTIFICATION 18

3 TEST BED DESCRIPTIONS 19

3.1 TEST BED (EPMM) 19
3.2 TEST BED – S22 (MDM AGENT, TLSC, AUTH) 19
3.3 TEST BED – S23 (MDM AGENT, TLSC, AUTH) 20
3.4 CONFIGURATION INFORMATION (EPMM) 21
3.5 CONFIGURATION INFORMATION – S22 (MDM AGENT, TLSC, AUTH) 23
3.6 CONFIGURATION INFORMATION – S23 (MDM AGENT, TLSC, AUTH) 24
3.7 TEST TIME AND LOCATION 26

4 DETAILED TEST CASES (TSS AND AGD ACTIVITIES) 27

4.1 SECURITY FUNCTIONAL REQUIREMENTS 27
4.1.1 Security Audit (FAU) 27
4.1.1.1 FAU_ALT_EXT.1 Server Alerts *(MDM_PP)* 27
4.1.1.1.1 FAU_ALT_EXT.1.1 TSS 27
4.1.1.1.2 FAU_ALT_EXT.1.1 AGD 28
4.1.1.2 FAU_ALT_EXT.2 Agent Alerts *(AGT_PPM)* 28
4.1.1.2.1 FAU_ALT_EXT.2 Agent Alerts TSS 28
4.1.1.2.2 FAU_ALT_EXT.2 Agent Alerts AGD 31
4.1.1.3 FAU_CRP_EXT.1 Support for Compliance Reporting of Mobile Device Configuration *(MDM_PP)* 31
4.1.1.3.1 FAU_CRP_EXT.1.1 TSS 31
4.1.1.3.2 FAU_CRP_EXT.1.1 AGD 31
4.1.1.4 FAU_GEN.1(1) Audit Data Generation *(MDM_PP)* 31
4.1.1.4.1 FAU_GEN.1.1(1) TSS 31
4.1.1.4.2 FAU_GEN.1.1(1) AGD 33
4.1.1.4.3 FAU_GEN.1.2(1) TSS 34
4.1.1.4.4 FAU_GEN.1.2(1) AGD 35
4.1.1.5 FAU_GEN.1(2)/AGT Audit Data Generation *(AGT_PPM)* 36
4.1.1.5.1 FAU_GEN.1(2)/AGT Audit Data Generation TSS 36
4.1.1.5.2 FAU_GEN.1(2)/AGT Audit Data Generation AGD 37
4.1.1.6 FAU_GEN.1(2)/SRV Audit Generation (MAS Server) *(MDM_PP)* 37
4.1.1.6.1 FAU_GEN.1.1(2)/SRV TSS 37
4.1.1.6.2 FAU_GEN.1.1(2)/SRV AGD 38
4.1.1.6.3 FAU_GEN.1.2(2)/SRV TSS 38
4.1.1.6.4 FAU_GEN.1.2(2)/SRV AGD 39
4.1.1.7 FAU_NET_EXT.1 Network Reachability Review *(MDM_PP)* 40
4.1.1.7.1 FAU_NET_EXT.1.1 TSS 40
4.1.1.7.2 FAU_NET_EXT.1.1 AGD 40
4.1.1.8 FAU_SAR.1 Audit Review *(MDM_PP)* 41
4.1.1.8.1 FAU_SAR.1.1 TSS 41
4.1.1.8.2 FAU_SAR.1.1 AGD 41
4.1.1.8.3 FAU_SAR.1.2 TSS 41
4.1.1.8.4 FAU_SAR.1.2 AGD 42
4.1.1.9 FAU_SEL.1(2) Security Audit Event Selection *(AGT_PPM)* 42
4.1.1.9.1 FAU_SEL.1(2) Security Audit Event Selection TSS 42

4.1.1.9.2 FAU_SEL.1(2) Security Audit Event Selection AGD 42
4.1.1.10 FAU_STG_EXT.1 External Trail Storage *(MDM_PP)* 43
4.1.1.10.1 FAU_STG_EXT.1.1 TSS 43
4.1.1.10.2 FAU_STG_EXT.1.1 AGD 44
4.1.1.11 FAU_STG_EXT.2 Audit Event Storage *(MDM_PP)* 45
4.1.1.11.1 FAU_STG_EXT.2.1 TSS 45
4.1.1.11.2 FAU_STG_EXT.2.1 AGD 45
4.1.2 Communication (FCO) *46*
4.1.2.1 FCO_CPC_EXT.1 Component Registration Channel Definition 46
4.1.2.1.1 FCO_CPC_EXT.1.1 TSS 46
4.1.2.1.2 FCO_CPC_EXT.1.1 AGD 46
4.1.2.1.3 FCO_CPC_EXT.1.2 TSS 46
4.1.2.1.4 FCO_CPC_EXT.1.2 AGD 46
4.1.2.1.5 FCO_CPC_EXT.1.3 TSS 46
4.1.2.1.6 FCO_CPC_EXT.1.3 AGD 47
4.1.3 Cryptographic Support (FCS) *47*
4.1.3.1 FCS_CKM.1 Cryptographic Key Generation *(MDM_PP)* 47
4.1.3.1.1 FCS_CKM.1.1 TSS 47
4.1.3.1.2 FCS_CKM.1.1 AGD 48
4.1.3.2 FCS_CKM.2 Cryptographic Key Establishment *(MDM_PP)* 49
4.1.3.2.1 FCS_CKM.2.1 TSS 49
4.1.3.2.2 FCS_CKM.2.1 AGD 50
4.1.3.3 FCS_CKM_EXT.4 Cryptographic Key Destruction *(MDM_PP)* 51
4.1.3.3.1 FCS_CKM_EXT.4.1 TSS 51
4.1.3.3.2 FCS_CKM_EXT.4.1 AGD 51
4.1.3.3.3 FCS_CKM_EXT.4.2 TSS 51
4.1.3.3.4 FCS_CKM_EXT.4.2 AGD 53
4.1.3.4 FCS_COP.1(1) Cryptographic Operation (Confidentiality Algorithms) *(MDM_PP)* 53
4.1.3.4.1 FCS_COP.1.1(1) TSS 53
4.1.3.4.2 FCS_COP.1.1(1) AGD 53
4.1.3.5 FCS_COP.1(2) Cryptographic Operation (Hashing Algorithms) *(MDM_PP)* 53
4.1.3.5.1 FCS_COP.1.1(2) TSS 53
4.1.3.5.2 FCS_COP.1.1(2) AGD 54
4.1.3.6 FCS_COP.1(3) Cryptographic Operation (Signature Algorithms) *(MDM_PP)* 55
4.1.3.6.1 FCS_COP.1.1(3) TSS 55
4.1.3.6.2 FCS_COP.1.1(3) AGD 55
4.1.3.7 FCS_COP.1(4) Cryptographic Operation (Keyed-Hash Message Authentication) *(MDM_PP)* 55
4.1.3.7.1 FCS_COP.1.1(4) TSS 55
4.1.3.7.2 FCS_COP.1.1(4) AGD 56
4.1.3.8 FCS_IV_EXT.1 INITIALIZATION VECTOR GENERATION (PP_MDM_V4.0) 56
4.1.3.8.1 FCS_IV_EXT.1.1 TSS 56
4.1.3.8.2 FCS_IV_EXT.1.1 AGD 57
4.1.3.9 FCS_HTTPS_EXT.1 HTTPS Protocol *(MDM_PP)* 57
4.1.3.9.1 FCS_HTTPS_EXT.1.2 TSS 57
4.1.3.9.2 FCS_HTTPS_EXT.1.2 AGD 57
4.1.3.10 FCS_RBG_EXT.1/AGT Extended: Random Bit Generation *(MDM_PP)* 57
4.1.3.10.1 FCS_RBG_EXT.1.1/AGT TSS 57
4.1.3.10.2 FCS_RBG_EXT.1.1/AGT AGD 58
4.1.3.10.3 FCS_RBG_EXT.1.2/AGT TSS 58
4.1.3.10.4 FCS_RBG_EXT.1.2/AGT AGD 58
4.1.3.11 FCS_RBG_EXT.1/SRV Extended: Random Bit Generation *(MDM_PP)* 58

4.1.3.11.1 FCS_RBG_EXT.1.1/SRV TSS 58
4.1.3.11.2 FCS_RBG_EXT.1.1/SRV AGD 59
4.1.3.11.3 FCS_RBG_EXT.1.2/SRV TSS 59
4.1.3.11.4 FCS_RBG_EXT.1.2/SRV AGD 59
4.1.3.12 FCS_STG_EXT.1 Cryptographic Key Storage *(MDM_PP)* 59
4.1.3.12.1 FCS_STG_EXT.1.1 TSS 59
4.1.3.12.2 FCS_STG_EXT.1.1 AGD 61
4.1.3.13 FCS_STG_EXT.1(2) Cryptographic Key Storage *(AGT_PPM)* 61
4.1.3.13.1 FCS_STG_EXT.1(2) Cryptographic Key Storage TSS 61
4.1.3.13.2 FCS_STG_EXT.1(2) Cryptographic Key Storage AGD 62
4.1.3.14 FCS_STG_EXT.2 ENCRYPTED CRYPTOGRAPHIC KEY STORAGE (PP_MDM_V4.0) 62
4.1.3.14.1 FCS_STG_EXT.2.1 TSS 62
4.1.3.14.2 FCS_STG_EXT.2.1 AGD 63
4.1.3.15 FCS_TLS_EXT.1 TLS Protocol *(TLS_PKG)* 63
4.1.3.15.1 FCS_TLS_EXT.1.1 TSS 63
4.1.3.15.2 FCS_TLS_EXT.1.1 AGD 63
4.1.3.16 FCS_TLSC_EXT.1 TLS Client Protocol *(TLS_PKG)* 63
4.1.3.16.1 FCS_TLSC_EXT.1.1 TSS 63
4.1.3.16.2 FCS_TLSC_EXT.1.1 AGD 64
4.1.3.16.3 FCS_TLSC_EXT.1.2 TSS 64
4.1.3.16.4 FCS_TLSC_EXT.1.2 AGD 65
4.1.3.16.5 FCS_TLSC_EXT.1.3 TSS 66
4.1.3.16.6 FCS_TLSC_EXT.1.3 AGD 66
4.1.3.17 FCS_TLSC_EXT.2 TLS Client Support For Mutual Authentication *(TLS_PKG)* 66
4.1.3.17.1 FCS_TLSC_EXT.2.1 TSS 66
4.1.3.17.2 FCS_TLSC_EXT.2.1 AGD 67
4.1.3.18 FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension *(TLS_PKG)* 68
4.1.3.18.1 FCS_TLSC_EXT.5.1 TSS 68
4.1.3.18.2 FCS_TLSC_EXT.5.1 AGD 68
4.1.3.19 FCS_TLSS_EXT.1 TLS Server Protocol *(TLS_PKG)* 68
4.1.3.19.1 FCS_TLSS_EXT.1.1 TSS 68
4.1.3.19.2 FCS_TLSS_EXT.1.1 AGD 69
4.1.3.19.3 FCS_TLSS_EXT.1.2 TSS 69
4.1.3.19.4 FCS_TLSS_EXT.1.2 AGD 70
4.1.3.19.5 FCS_TLSS_EXT.1.3 TSS 70
4.1.3.19.6 FCS_TLSS_EXT.1.3 AGD 71
4.1.3.20 FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication *(TLS_PKG)* 71
4.1.3.20.1 FCS_TLSS_EXT.2.1 TSS 71
4.1.3.20.2 FCS_TLSS_EXT.2.1 AGD 71
4.1.3.20.3 FCS_TLSS_EXT.2.2 TSS **[TD0770]** 72
4.1.3.20.4 FCS_TLSS_EXT.2.2 AGD 72
4.1.3.20.5 FCS_TLSS_EXT.2.3 TSS 73
4.1.3.20.6 FCS_TLSS_EXT.2.3 AGD 73
4.1.4 Identification and Authentication (FIA) *74*
4.1.4.1 FIA_CLI_EXT.1 Client Authorization 74
4.1.4.1.1 FIA_CLI_EXT.1.1 TSS **[TD0754]** 74
4.1.4.1.2 FIA_CLI_EXT.1.1 AGD **[TD0754]** 74
4.1.4.2 FIA_ENR_EXT.1 Enrollment of Mobile Device into Management (MDM_PP) 74
4.1.4.2.1 FIA_ENR_EXT.1.1 TSS 74
4.1.4.2.2 FIA_ENR_EXT.1.1 AGD 75
4.1.4.2.3 FIA_ENR_EXT.1.2 TSS 75

4.1.4.2.4 FIA_ENR_EXT.1.2 AGD 76
4.1.4.3 FIA_ENR_EXT.2 Agent Enrollment of Mobile Device into Management *(AGT_PPM)* 76
4.1.4.3.1 FIA_ENR_EXT.2 Agent Enrollment of Mobile Device into Management TSS 76
4.1.4.3.2 FIA_ENR_EXT.2 Agent Enrollment of Mobile Device into Management AGD 77
4.1.4.4 FIA_UAU.1 Timing of Authentication *(MDM_PP)* 77
4.1.4.4.1 FIA_UAU.1.1 TSS 77
4.1.4.4.2 FIA_UAU.1.1 AGD 77
4.1.4.4.3 FIA_UAU.1.2 TSS 78
4.1.4.4.4 FIA_UAU.1.2 AGD 78
4.1.4.5 FIA_X509_EXT.1(1) X.509 Certificate Validation *(MDM_PP)* 78
4.1.4.5.1 FIA_X509_EXT.1.1(1) TSS **[TD0641]** 78
4.1.4.5.2 FIA_X509_EXT.1.1(1) AGD **[TD0641]** 79
4.1.4.5.3 FIA_X509_EXT.1.2(1) TSS 79
4.1.4.5.4 FIA_X509_EXT.1.2(1) AGD 80
4.1.4.6 FIA_X509_EXT.2 X.509 Certificate Authentication *(MDM_PP)* 80
4.1.4.6.1 FIA_X509_EXT.2.1 TSS 80
4.1.4.6.2 FIA_X509_EXT.2.1 AGD 80
4.1.4.6.3 FIA_X509_EXT.2.2 TSS 80
4.1.4.6.4 FIA_X509_EXT.2.2 AGD 81
4.1.5 Security Management (FMT) *82*
4.1.5.1 FMT_MOF.1(1) Management of Functions Behavior *(MDM_PP)* 82
4.1.5.1.1 FMT_MOF.1.1(1) TSS 82
4.1.5.1.2 FMT_MOF.1.1(1) AGD 83
4.1.5.2 FMT_MOF.1(2) Management of Functions Behavior (Enrollment) *(MDM_PP)* 83
4.1.5.2.1 FMT_MOF.1.1(2) TSS 83
4.1.5.2.2 FMT_MOF.1.1(2) AGD 83
4.1.5.3 FMT_MOF.1(3) Management of Functions in (MAS Server Downloads) 83
4.1.5.3.1 FMT_MOF.1.1(3)TSS 83
4.1.5.3.2 FMT_MOF.1.1(3) AGD 84
4.1.5.4 FMT_POL_EXT.1 Trusted Policy Update *(MDM_PP)* 84
4.1.5.4.1 FMT_POL_EXT.1.1 TSS **[TD0754]** 84
4.1.5.4.2 FMT_POL_EXT.1.1 AGD 85
4.1.5.5 FMT_POL_EXT.2 Agent Trusted Policy Update *(AGT_PPM)* 86
4.1.5.5.1 FMT_POL_EXT.2 Agent Trusted Policy Update TSS 86
4.1.5.5.2 FMT_POL_EXT.2 Agent Trusted Policy Update AGD 87
4.1.5.6 FMT_SMF.1(1) Specification of Management Functions (Server Configuration of Agent) *(MDM_PP)* 87
4.1.5.6.1 FMT_SMF.1.1(1) TSS **[TD0479]** 87
4.1.5.6.2 FMT_SMF.1.1(1) AGD 91
4.1.5.7 FMT_SMF.1(2) Specification of Management Functions (Server Configuration of Server) *(MDM_PP)* 91
4.1.5.7.1 FMT_SMF.1.1(2) TSS 91
4.1.5.7.2 FMT_SMF.1.1(2) AGD 92
4.1.5.8 FMT_SMF.1(3) Specification of Management Functions (MAS Server) *(MDM_PP)* 92
4.1.5.8.1 FMT_SMF.1.1(3) TSS 92
4.1.5.8.2 FMT_SMF.1.1(3) AGD 93
4.1.5.9 FMT_SMF_EXT.4 Specification Of Management Functions (AGT_PPM) 94
4.1.5.9.1 FMT_SMF_EXT.4 Specification Of Management Functions TSS 94
4.1.5.9.2 FMT_SMF_EXT.4 Specification Of Management Functions AGD 95
4.1.5.10 FMT_SMR.1(1) Security Management Roles *(MDM_PP)* 96
4.1.5.10.1 FMT_SMR.1.2(1) TSS 96
4.1.5.10.2 FMT_SMR.1.2(1) AGD 99
4.1.5.11 FMT_SMR.1(2) Security Management Roles (MAS Server) 99

4.1.5.11.1 FMT_SMR.1.2(2) TSS 99
4.1.5.11.2 FMT_SMR.1.2(2) AGD 100
4.1.5.12 FMT_UNR_EXT.1 User Unenrollment Prevention (AGT_PPM) 100
4.1.5.12.1 FMT_UNR_EXT.1 User Unenrollment Prevention TSS 100
4.1.5.12.2 FMT_UNR_EXT.1 User Unenrollment Prevention AGD 101
4.1.6 Protection of the TSF (FPT) 102
4.1.6.1 FPT_API_EXT.1 Use of Supported Services and APIs *(MDM_PP)* 102
4.1.6.1.1 FPT_API_EXT.1.1 TSS 102
4.1.6.1.2 FPT_API_EXT.1.1 AGD 102
4.1.6.2 FPT_ITT.1(2) Internal TOE TSF Data Transfer (MDM Agent) *(MDM_PP)* 103
4.1.6.2.1 FPT_ITT.1.1(2) TSS 103
4.1.6.2.2 FPT_ITT.1.1(2) AGD 103
4.1.6.3 FPT_LIB_EXT.1/AGT Use of Third Party Libraries *(MDM_PP)* 104
4.1.6.3.1 FPT_LIB_EXT.1.1/AGT TSS 104
4.1.6.3.2 FPT_LIB_EXT.1.1/AGT AGD 104
4.1.6.4 FPT_LIB_EXT.1/SRV Use of Third Party Libraries *(MDM_PP)* 104
4.1.6.4.1 FPT_LIB_EXT.1.1/SRV TSS 104
4.1.6.4.2 FPT_LIB_EXT.1.1/SRV AGD 105
4.1.6.5 FPT_TST_EXT.1 Functionality Testing *(MDM_PP)* 105
4.1.6.5.1 FPT_TST_EXT.1.2 TSS 105
4.1.6.5.2 FPT_TST_EXT.1.2 AGD 107
4.1.6.6 FPT_TUD_EXT.1 Trusted Update *(MDM_PP)* 107
4.1.6.6.1 FPT_TUD_EXT.1.1 TSS 107
4.1.6.6.2 FPT_TUD_EXT.1.1 AGD 107
4.1.6.6.1 FPT_TUD_EXT.1.2 TSS 108
4.1.6.6.2 FPT_TUD_EXT.1.2 AGD 108
4.1.6.6.3 FPT_TUD_EXT.1.3 TSS 108
4.1.6.6.4 FPT_TUD_EXT.1.3 AGD 109
4.1.7 Trusted Path/Channels (FTP) 110
4.1.7.1 FTP_ITC.1(1) Inter-TSF Trusted Channel (Authorized IT Entities) *(MDM_PP)* 110
4.1.7.1.1 FTP_ITC.1.1(1) TSS 110
4.1.7.1.2 FTP_ITC.1.1(1) AGD 111
4.1.7.1.3 FTP_ITC.1.2(1) 111
4.1.7.1.4 FTP_ITC.1.2(1) AGD 111
4.1.7.1.5 FTP_ITC.1.3(1) TSS 111
4.1.7.1.6 FTP_ITC.1.3(1) AGD 112
4.1.7.2 FTP_ITC_EXT.1 Trusted Channel *(MDM_PP)* 112
4.1.7.2.1 FTP_ITC_EXT.1.1 TSS 112
4.1.7.2.2 FTP_ITC_EXT.1.1 AGD 112
4.1.7.3 FTP_TRP.1(1) Trusted Path (for Remote Administration) *(MDM_PP)* 113
4.1.7.3.1 FTP_TRP.1.1(1) TSS 113
4.1.7.3.2 FTP_TRP.1.1(1) AGD 113
4.1.7.3.3 FTP_TRP.1.2(1) TSS 113
4.1.7.3.4 FTP_TRP.1.2(1) AGD 113
4.1.7.3.5 FTP_TRP.1.3(1) TSS 113
4.1.7.3.6 FTP_TRP.1.3(1) AGD 114
4.1.7.4 FTP_TRP.1(2) Trusted Path (for Enrollment) *(MDM_PP)* 114
4.1.7.4.1 FTP_TRP.1.1(2) TSS 114
4.1.7.4.2 FTP_TRP.1.1(2) AGD 115
4.1.7.4.3 FTP_TRP.1.2(2) TSS 115
4.1.7.4.4 FTP_TRP.1.2(2) AGD 115

4.1.7.4.5 FTP_TRP.1.3(2) TSS 115
4.1.7.4.6 FTP_TRP.1.3(2) AGD 116

5 SECURITY ASSURANCE REQUIREMENTS 117

5.1 ADV: DEVELOPMENT 117
5.1.1 Basic Functional Specification (ADV_FSP.1) 117
5.1.1.1 Evaluation Activity 117
5.2 AGD: GUIDANCE DOCUMENTATION 117
5.2.1 Operational User Guidance (AGD_OPE.1) 117
5.2.1.1 Evaluation Activity 117
5.2.2 Preparative Procedures (AGD_PRE.1) 119
5.2.2.1 Evaluation Activity 120
5.3 ALC: LIFE-CYCLE SUPPORT 120
5.3.1 Labeling of the TOE (ALC_CMC.1) 120
5.3.1.1 Evaluation Activity 120
5.3.2 TOE CM Coverage (ALC_CMS.1) 121
5.3.2.1 Evaluation Activity 121
5.4 ATE: TESTS 121
5.4.1 Independent Testing - Conformance (ATE_IND.1) 121
5.4.1.1 Evaluation Activity 121
5.5 AVA: VULNERABILITY ANALYSIS 123
5.5.1 Vulnerability Survey (AVA_VAN.1) 123
5.5.1.1 Evaluation Activity 123

6 DETAILED TEST CASES (TEST ACTIVITIES) 126

6.1 ENDPOINT MANAGER MOBILE (EPMM) SYSTEM 12 126
6.1.1 Audit Test Cases 126
6.1.1.1 FAU_ALT_EXT.1.1 Test #1 126
6.1.1.2 FAU_ALT_EXT.1.1 Test #2 126
6.1.1.3 FAU_ALT_EXT.1 Test #3 – NA 127
6.1.1.4 FAU_CRP_EXT.1.1 Test #1 127
6.1.1.5 FAU_CRP_EXT.1.1 Test #2 127
6.1.1.6 FAU_GEN.1.1(1) Test #1 127
6.1.1.7 FAU_GEN.1.2(1) Test #1 128
6.1.1.8 FAU_GEN.1.1(2)/SRV Test #1 128
6.1.1.9 FAU_GEN.1.2(2) Test #1 129
6.1.1.10 FAU_NET_EXT.1.1 Test #1 129
6.1.1.11 FAU_SAR.1.1 Test #1 129
6.1.1.12 FAU_SAR.1.2 Test #1 130
6.1.1.13 FAU_STG_EXT.1.1 Test #1 130
6.1.1.14 FAU_STG_EXT.2.1 Test #1 131
6.1.1.15 FAU_STG_EXT.2.1 Test #2 131
6.1.2 Auth Test Cases 131
6.1.2.1 FIA_ENR_EXT.1.1 Test #1 131
6.1.2.2 FIA_ENR_EXT.1.1 Test #2 132
6.1.2.3 FIA_ENR_EXT.1.2 Test #1 132
6.1.2.4 FIA_UAU.1.1 Test #1 132
6.1.2.5 FIA_UAU.1.2 Test #1 133
6.1.2.6 FIA_UAU.1.2 Test #2 133

6.1.2.7 FIA_X509_EXT.1.1(1) Test #1 **[TD0641]** ... 133
6.1.2.8 FIA_X509_EXT.1.1(1) Test #2 **[TD0641]** ... 136
6.1.2.9 FIA_X509_EXT.1.1(1) Test #3 **[TD0641]** ... 137
6.1.2.10 FIA_X509_EXT.1.1(1) Test #4 **[TD0641]** ... 139
6.1.2.11 FIA_X509_EXT.1.1(1) Test #5 **[TD0641]** ... 140
6.1.2.12 FIA_X509_EXT.1.1(1) Test #6 **[TD0641]** ... 140
6.1.2.13 FIA_X509_EXT.1.1(1) Test #7 **[TD0641]** ... 141
6.1.2.14 FIA_X509_EXT.1.1(1) Test #8a **[TD0641]** ... 142
6.1.2.15 FIA_X509_EXT.1.1(1) Test #8b **[TD0641]** ... 142
6.1.2.16 FIA_X509_EXT.1.2(1) Test #1 ... 143
6.1.2.17 FIA_X509_EXT.1.2(1) Test #2 ... 144
6.1.2.18 FIA_X509_EXT.1.2(1) Test #3 ... 144
6.1.2.19 FIA_X509_EXT.2.1 Test #1 ... 145
6.1.2.20 FIA_X509_EXT.2.2 Test #1 ... 145
6.1.3 Crypto Test Cases ... 146
6.1.3.1 FCS_CKM.1 RSA**[TD0951]** ... 146
6.1.3.2 FCS_CKM.1 ECC **[TD0951]** ... 147
6.1.3.3 FCS_CKM.2 SP800-56A ... 147
6.1.3.4 FCS_CKM.2 RSA ... 149
6.1.3.5 FCS_CKM_EXT.4.1 Test #1 ... 149
6.1.3.6 FCS_CKM_EXT.4.2 Test #1 ... 149
6.1.3.7 FCS_CKM_EXT.4.2 TEST #2 – NA ... 150
6.1.3.8 FCS_COP.1.1(1) AES-GCM ... 151
6.1.3.9 FCS_COP.1.1(2) Hashing ... 152
6.1.3.10 FCS_COP.1.1(3) SigGen ECDSA **[TD0951]** ... 153
6.1.3.11 FCS_COP.1.1(3) SigGen RSA ... 153
6.1.3.12 FCS_COP.1.1(4) Keyed-Hash ... 154
6.1.3.13 FCS_IV_EXT.1.1 ... 154
6.1.3.14 FCS_RBG_EXT.1.1/SRV ... 155
6.1.3.15 FCS_RBG_EXT.1.2/SRV ... 156
6.1.3.16 FCS_STG_EXT.1.1 ... 156
6.1.3.17 FCS_STG_EXT.2.1 ... 156
6.1.3.18 FCS_HTTPS_EXT.1.2 Test #1 ... 156
6.1.3.19 FCS_TLSC_EXT.1.1 Test #1 ... 157
6.1.3.20 FCS_TLSC_EXT.1.1 Test #2 ... 158
6.1.3.21 FCS_TLSC_EXT.1.1 Test #3 ... 158
6.1.3.22 FCS_TLSC_EXT.1.1 Test #4 ... 159
6.1.3.23 FCS_TLSC_EXT.1.1 Test #5.1 ... 159
6.1.3.24 FCS_TLSC_EXT.1.1 Test #5.2 ... 160
6.1.3.25 FCS_TLSC_EXT.1.1 Test #5.3 ... 160
6.1.3.26 FCS_TLSC_EXT.1.1 Test #5.4 ... 160
6.1.3.27 FCS_TLSC_EXT.1.1 Test #5.5 ... 161
6.1.3.28 FCS_TLSC_EXT.1.1 Test #5.6 ... 161
6.1.3.29 FCS_TLSC_EXT.1.1 Test #5.7 ... 162
6.1.3.30 FCS_TLSC_EXT.1.2 Test #1 **[TD0499]** ... 162
6.1.3.31 FCS_TLSC_EXT.1.2 Test #2 **[TD0499]** ... 163
6.1.3.32 FCS_TLSC_EXT.1.2 Test #3 **[TD0499]** - NA ... 163
6.1.3.33 FCS_TLSC_EXT.1.2 Test #4 **[TD0499]** ... 164
6.1.3.34 FCS_TLSC_EXT.1.2 Test #5.1 **[TD0499]** ... 164
6.1.3.35 FCS_TLSC_EXT.1.2 Test #5.2 **[TD0499]** ... 165
6.1.3.36 FCS_TLSC_EXT.1.2 Test #5.3 **[TD0499]** ... 166

6.1.3.37 FCS_TLSC_EXT.1.2 Test #5.4 **[TD0499]** - NA 167
6.1.3.38 FCS_TLSC_EXT.1.2 Test #6 **[TD0499]** - NA 167
6.1.3.39 FCS_TLSC_EXT.1.2 Test #7 **[TD0499]** - NA 168
6.1.3.40 FCS_TLSC_EXT.1.3 Test #1a **[TD0513]** 168
6.1.3.41 FCS_TLSC_EXT.1.3 Test #1b **[TD0513]** 169
6.1.3.42 FCS_TLSC_EXT.1.3 Test #1c **[TD0513]** - NA 169
6.1.3.43 FCS_TLSC_EXT.1.3 Test #2 170
6.1.3.44 FCS_TLSC_EXT.1.3 Test #3 170
6.1.3.45 FCS_TLSC_EXT.1.3 Test #4 170
6.1.3.46 FCS_TLSC_EXT.2.1 Test #1 171
6.1.3.47 FCS_TLSC_EXT.2.1 Test #2 171
6.1.3.48 FCS_TLSC_EXT.5.1 Test #1 171
6.1.3.49 FCS_TLSS_EXT.1.1 Test #1 172
6.1.3.50 FCS_TLSS_EXT.1.1 Test #2 173
6.1.3.51 FCS_TLSS_EXT.1.1 Test #3 173
6.1.3.52 FCS_TLSS_EXT.1.1 Test #4.1 **[TD0469]** 174
6.1.3.53 FCS_TLSS_EXT.1.1 Test #4.2 174
6.1.3.54 FCS_TLSS_EXT.1.1 Test #4.3i [TD0779] - NA 174
6.1.3.55 FCS_TLSS_EXT.1.1 Test #4.3ii **[TD0779]** 175
6.1.3.56 FCS_TLSS_EXT.1.1 Test #4.3iii **[TD0779]** 176
6.1.3.57 FCS_TLSS_EXT.1.1 Test #4.4 177
6.1.3.58 FCS_TLSS_EXT.1.2 Test #1 178
6.1.3.59 FCS_TLSS_EXT.1.3 Test #1 **[TD0739]** 178
6.1.3.60 FCS_TLSS_EXT.1.3 Test #2 – NA 179
6.1.3.61 FCS_TLSS_EXT.1.3 Test #3 179
6.1.3.62 FCS_TLSS_EXT.2.2 Test #1 **[TD0770]** 180
6.1.3.63 FCS_TLSS_EXT.2.2 Test #2 **[TD0770]** 180
6.1.3.64 FCS_TLSS_EXT.2.2 Test #3 **[TD0770]** 181
6.1.3.65 FCS_TLSS_EXT.2.2 Test #4 **[TD0770]** 182
6.1.3.66 FCS_TLSS_EXT.2.2 Test #5 **[TD0770]** 183
6.1.3.67 FCS_TLSS_EXT.2.2 Test #6 **[TD0770]** 183
6.1.3.68 FCS_TLSS_EXT.2.2 Test #7 **[TD0770]** 184
6.1.3.69 FCS_TLSS_EXT.2.3 Test #1 185
6.1.4 System Manager TLSS Test Cases 186
6.1.4.1 FCS_TLSS_EXT.1.1 Test #1 186
6.1.4.2 FCS_TLSS_EXT.1.1 Test #2 187
6.1.4.3 FCS_TLSS_EXT.1.1 Test #3 187
6.1.4.4 FCS_TLSS_EXT.1.1 Test #4.1 **[TD0469]** 187
6.1.4.5 FCS_TLSS_EXT.1.1 Test #4.2 188
6.1.4.6 FCS_TLSS_EXT.1.1 Test #4.3i [TD0779] - NA 188
6.1.4.7 FCS_TLSS_EXT.1.1 Test #4.3ii **[TD0779]** 189
6.1.4.8 FCS_TLSS_EXT.1.1 Test #4.3iii **[TD0779]** 190
6.1.4.9 FCS_TLSS_EXT.1.1 Test #4.4 191
6.1.4.10 FCS_TLSS_EXT.1.2 Test #1 192
6.1.4.11 FCS_TLSS_EXT.1.3 Test #1 **[TD0739]** 192
6.1.4.12 FCS_TLSS_EXT.1.3 Test #2 – NA 193
6.1.4.13 FCS_TLSS_EXT.1.3 Test #3 193
6.1.5 Management Functions Test Cases 194
6.1.5.1 FMT_MOF.1.1(1) Test #1 194
6.1.5.2 FMT_MOF.1.1(1) Test #2 194
6.1.5.3 FMT_MOF.1.1(2) Test #1 194

6.1.5.4 FMT_MOF.1.1(3) Test #1 194
6.1.5.5 FMT_POL_EXT.1.1 Test #1 195
6.1.5.6 FMT_SMF.1.1(1) Test #1 195
6.1.5.7 FMT_SMF.1.1(2) Test #1 196
6.1.5.8 FMT_SMF.1.1(2) Test #2 197
6.1.5.9 FMT_SMF.1.1(2) Test #3 197
6.1.5.10 FMT_SMF.1.1(3) Test #1 197
6.1.5.11 FMT_SMR.1.2(1) Test #1 198
6.1.5.12 FMT_SMR.1.2(2) Test #1 198
6.1.6 Data Transfer Test Cases 199
6.1.6.1 FPT_API_EXT.1.1 Test #1 199
6.1.6.2 FPT_LIB_EXT.1.1 Test #1 199
6.1.6.3 FPT_ITT.1.1(2) Test #1 199
6.1.6.4 FPT_ITT.1.1(2) Test #2 199
6.1.7 Update Test Cases 200
6.1.7.1 FPT_TST_EXT.1.2 Test #1 200
6.1.7.2 FPT_TST_EXT.1.2 Test #2 200
6.1.7.3 FPT_TUD_EXT.1.1 Test #1 200
6.1.7.4 FPT_TUD_EXT.1.2 Test #2 201
6.1.7.5 FPT_TUD_EXT.1.3 Test #1 201
6.1.7.6 FPT_TUD_EXT.1.3 Test #2 201
6.1.8 Trusted Channels Test Cases 202
6.1.8.1 FCO_CPC_EXT.1.1 Test #1 202
6.1.8.2 FCO_CPC_EXT.1.2 Test #1 202
6.1.8.3 FCO_CPC_EXT.1.3 Test #1 202
6.1.8.4 FCO_CPC_EXT.1.3 Test #2 **[TD0594]** 202
6.1.8.5 FCO_CPC_EXT.1.3 Test #3 **[TD0594]** 203
6.1.8.6 FTP_ITC_EXT.1.1 Test #1 203
6.1.8.7 FTP_ITC.1.1(1) Test #1 203
6.1.8.8 FTP_ITC.1.2(1) Test #1 203
6.1.8.9 FTP_ITC.1.3(1) Test #1 203
6.1.8.10 FTP_ITC.1.3(1) Test #2 204
6.1.8.11 FTP_ITC.1.3(1) Test #3 204
6.1.8.12 FTP_ITC.1.1(2) Test #1 205
6.1.8.13 FTP_ITC.1.2(2) Test #1 205
6.1.8.14 FTP_ITC.1.3(2) Test #1 205
6.1.8.15 FTP_ITC.1.3(2) Test #2 205
6.1.8.16 FTP_ITC.1.3(2) Test #3 205
6.1.8.17 FTP_TRP.1.1(1) Test #1 206
6.1.8.18 FTP_TRP.1.2(1) Test #1 206
6.1.8.19 FTP_TRP.1.3(1) Test #1 206
6.1.8.20 FTP_TRP.1.3(1) Test #2 207
6.1.8.21 FTP_TRP.1.3(1) Test #3 207
6.1.8.22 FTP_TRP.1.1(2) Test #1 207
6.1.8.23 FTP_TRP.1.2(2) Test #1 207
6.1.8.24 FTP_TRP.1.3(2) Test #1 208
6.1.8.25 FTP_TRP.1.3(2) Test #2 208
6.1.8.26 FTP_TRP.1.3(2) Test #3 208
6.1.8.27 209
6.2 GALAXY S22 ULTRA 5G RUNNING ANDROID 13 209

6.2.1 MDM Agent Test Cases 209
6.2.1.1 FCS_STG_EXT.1(2) Test #1 - NA 209
6.2.1.2 FAU_ALT_EXT.2 Test #1 209
6.2.1.3 FAU_ALT_EXT.2 Test #2 209
6.2.1.4 FAU_ALT_EXT.2 Test #3 210
6.2.1.5 FAU_ALT_EXT.2 Test #4 210
6.2.1.6 FAU_GEN.1(2)/AGT Test #1 211
6.2.1.7 FAU_SEL.1(2) Test #1 211
6.2.1.8 FAU_SEL.1(2) Test #2 - NA 211
6.2.1.9 FIA_ENR_EXT.2 Test #1 211
6.2.1.10 FMT_POL_EXT.2 Test #1 212
6.2.1.11 FMT_POL_EXT.2 Test #2 212
6.2.1.12 FMT_SMF_EXT.4 Test #1 213
6.2.1.13 FMT_SMF_EXT.4 Test #2 **[TD0755]** 213
6.2.1.14 FMT_SMF_EXT.4 Test #3 213
6.2.1.15 FMT_SMF_EXT.4 Test #4 214
6.2.1.16 FMT_SMF_EXT.4 Test #5 214
6.2.1.17 FMT_UNR_EXT.1 Test #1 214
6.2.1.18 FMT_UNR_EXT.1 Test #2 – NA 215
6.2.2 Auth Test Cases 215
6.2.2.1 FIA_X509_EXT.1.1(1) Test #1 **[TD0641]** 215
6.2.2.2 FIA_X509_EXT.1.1(1) Test #2 **[TD0641]** 217
6.2.2.3 FIA_X509_EXT.1.1(1) Test #3 **[TD0641]** 217
6.2.2.4 FIA_X509_EXT.1.1(1) Test #4 **[TD0641]** 219
6.2.2.5 FIA_X509_EXT.1.1(1) Test #5 **[TD0641]** 220
6.2.2.6 FIA_X509_EXT.1.1(1) Test #6 **[TD0641]** 221
6.2.2.7 FIA_X509_EXT.1.1(1) Test #7 **[TD0641]** 221
6.2.2.8 FIA_X509_EXT.1.1(1) Test #8a **[TD0641]** 222
6.2.2.9 FIA_X509_EXT.1.1(1) Test #8b **[TD0641]** 223
6.2.2.10 FIA_X509_EXT.1.2(1) Test #1 224
6.2.2.11 FIA_X509_EXT.1.2(1) Test #2 224
6.2.2.12 FIA_X509_EXT.1.2(1) Test #3 225
6.2.2.13 FIA_X509_EXT.2.1 Test #1 226
6.2.2.14 FIA_X509_EXT.2.2 Test #1 226
6.2.3 TLSC Test Cases 227
6.2.3.1 FCS_TLSC_EXT.1.1 Test #1 227
6.2.3.2 FCS_TLSC_EXT.1.1 Test #2 228
6.2.3.3 FCS_TLSC_EXT.1.1 Test #3 229
6.2.3.4 FCS_TLSC_EXT.1.1 Test #4 229
6.2.3.5 FCS_TLSC_EXT.1.1 Test #5.1 229
6.2.3.6 FCS_TLSC_EXT.1.1 Test #5.2 230
6.2.3.7 FCS_TLSC_EXT.1.1 Test #5.3 230
6.2.3.8 FCS_TLSC_EXT.1.1 Test #5.4 230
6.2.3.9 FCS_TLSC_EXT.1.1 Test #5.5 231
6.2.3.10 FCS_TLSC_EXT.1.1 Test #5.6 231
6.2.3.11 FCS_TLSC_EXT.1.1 Test #5.7 232
6.2.3.12 FCS_TLSC_EXT.1.2 Test #1 **[TD0499]** 232
6.2.3.13 FCS_TLSC_EXT.1.2 Test #2 **[TD0499]** 233
6.2.3.14 FCS_TLSC_EXT.1.2 Test #3 **[TD0499]** - NA 233
6.2.3.15 FCS_TLSC_EXT.1.2 Test #4 **[TD0499]** 234
6.2.3.16 FCS_TLSC_EXT.1.2 Test #5.1 **[TD0499]** - NA 234

6.2.3.17 FCS_TLSC_EXT.1.2 Test #5.2 **[TD0499]** - NA 235
6.2.3.18 FCS_TLSC_EXT.1.2 Test #5.3 **[TD0499]** - NA 236
6.2.3.19 FCS_TLSC_EXT.1.2 Test #6 **[TD0499]** - NA 236
6.2.3.20 FCS_TLSC_EXT.1.2 Test #7 **[TD0499]** - NA 237
6.2.3.21 FCS_TLSC_EXT.1.2 Test #5.4 **[TD0499]** 237
6.2.3.22 FCS_TLSC_EXT.1.3 Test #1a **[TD0513]** 238
6.2.3.23 FCS_TLSC_EXT.1.3 Test #1b **[TD0513]** 238
6.2.3.24 FCS_TLSC_EXT.1.3 Test #1c **[TD0513]** - NA 239
6.2.3.25 FCS_TLSC_EXT.1.3 Test #2 239
6.2.3.26 FCS_TLSC_EXT.1.3 Test #3 239
6.2.3.27 FCS_TLSC_EXT.1.3 Test #4 240
6.2.3.28 FCS_TLSC_EXT.2.1 Test #1 240
6.2.3.29 FCS_TLSC_EXT.2.1 Test #2 240
6.2.3.30 FCS_TLSC_EXT.5.1 Test #1 241
6.2.4 Crypto Test Cases 241
6.2.4.1 FCS_CKM.1 RSA[**TD0951**] 241
6.2.4.2 FCS_CKM.1 ECC **[TD0951]** 242
6.2.4.3 FCS_CKM.2 SP800-56A 243
6.2.4.4 FCS_CKM.2 RSA 245
6.2.4.5 FCS_CKM_EXT.4.1 Test #1 245
6.2.4.6 FCS_CKM.4.2 Test#1 – NA. 245
6.2.4.7 FCS_CKM.4.2 Test #2 – NA. 246
6.2.4.8 FCS_COP.1.1(1) AES-GCM 246
6.2.4.9 FCS_COP.1.1(2) Hashing 247
6.2.4.10 FCS_COP.1.1(3) SigGen ECDSA **[TD0951]** 248
6.2.4.11 FCS_COP.1.1(3) SigGen RSA 249
6.2.4.12 FCS_COP.1.1(4) Keyed-Hash 250
6.2.4.13 FCS_IV_EXT.1.1 250
6.2.4.14 FCS_RBG_EXT.1.1/AGT 250
6.2.4.15 FCS_RBG_EXT.1.2/AGT 251
6.3 GALAXY S23 ULTRA 5G RUNNING ANDROID 14 251
6.3.1 MDM Agent Cases 252
6.3.1.1 FCS_STG_EXT.1(2) Test #1 - NA 252
6.3.1.2 FAU_ALT_EXT.2 Test #1 252
6.3.1.3 FAU_ALT_EXT.2 Test #2 252
6.3.1.4 FAU_ALT_EXT.2 Test #3 252
6.3.1.5 FAU_ALT_EXT.2 Test #4 253
6.3.1.6 FAU_GEN.1(2)/AGT Test #1 253
6.3.1.7 FAU_SEL.1(2) Test #1 253
6.3.1.8 FAU_SEL.1(2) Test #2 - NA 254
6.3.1.9 FIA_ENR_EXT.2 Test #1 254
6.3.1.10 FMT_POL_EXT.2 Test #1 254
6.3.1.11 FMT_POL_EXT.2 Test #2 255
6.3.1.12 FMT_SMF_EXT.4 Test #1 255
6.3.1.13 FMT_SMF_EXT.4 Test #2 **[TD0755]** 256
6.3.1.14 FMT_SMF_EXT.4 Test #3 256
6.3.1.15 FMT_SMF_EXT.4 Test #4 256
6.3.1.16 FMT_SMF_EXT.4 Test #5 257
6.3.1.17 FMT_UNR_EXT.1 Test #1 257
6.3.1.18 FMT_UNR_EXT.1 Test #2 – NA 257

6.3.2 Auth Test Cases 257
6.3.2.1 FIA_X509_EXT.1.1(1) Test #1 **[TD0641]** 257
6.3.2.2 FIA_X509_EXT.1.1(1) Test #2 **[TD0641]** 259
6.3.2.3 FIA_X509_EXT.1.1(1) Test #3 **[TD0641]** 260
6.3.2.4 FIA_X509_EXT.1.1(1) Test #4 **[TD0641]** 261
6.3.2.5 FIA_X509_EXT.1.1(1) Test #5 **[TD0641]** 262
6.3.2.6 FIA_X509_EXT.1.1(1) Test #6 **[TD0641]** 262
6.3.2.7 FIA_X509_EXT.1.1(1) Test #7 **[TD0641]** 263
6.3.2.8 FIA_X509_EXT.1.1(1) Test #8a **[TD0641]** 263
6.3.2.9 FIA_X509_EXT.1.1(1) Test #8b **[TD0641]** 264
6.3.2.10 FIA_X509_EXT.1.2(1) Test #1 264
6.3.2.11 FIA_X509_EXT.1.2(1) Test #2 265
6.3.2.12 FIA_X509_EXT.1.2(1) Test #3 265
6.3.2.13 FIA_X509_EXT.2.1 Test #1 266
6.3.2.14 FIA_X509_EXT.2.2 Test #1 266
6.3.3 TLSC Test Cases 266
6.3.3.1 FCS_TLSC_EXT.1.1 Test #1 266
6.3.3.2 FCS_TLSC_EXT.1.1 Test #2 267
6.3.3.3 FCS_TLSC_EXT.1.1 Test #3 268
6.3.3.4 FCS_TLSC_EXT.1.1 Test #4 268
6.3.3.5 FCS_TLSC_EXT.1.1 Test #5.1 269
6.3.3.6 FCS_TLSC_EXT.1.1 Test #5.2 269
6.3.3.7 FCS_TLSC_EXT.1.1 Test #5.3 270
6.3.3.8 FCS_TLSC_EXT.1.1 Test #5.4 270
6.3.3.9 FCS_TLSC_EXT.1.1 Test #5.5 270
6.3.3.10 FCS_TLSC_EXT.1.1 Test #5.6 271
6.3.3.11 FCS_TLSC_EXT.1.1 Test #5.7 271
6.3.3.12 FCS_TLSC_EXT.1.2 Test #1 **[TD0499]** 272
6.3.3.13 FCS_TLSC_EXT.1.2 Test #2 **[TD0499]** 272
6.3.3.14 FCS_TLSC_EXT.1.2 Test #3 **[TD0499]** - NA 273
6.3.3.15 FCS_TLSC_EXT.1.2 Test #4 **[TD0499]** 273
6.3.3.16 FCS_TLSC_EXT.1.2 Test #5.1 **[TD0499]** - NA 274
6.3.3.17 FCS_TLSC_EXT.1.2 Test #5.2 **[TD0499]** - NA 274
6.3.3.18 FCS_TLSC_EXT.1.2 Test #5.3 **[TD0499]** - NA 275
6.3.3.19 FCS_TLSC_EXT.1.2 Test #5.4 **[TD0499]** 275
6.3.3.20 FCS_TLSC_EXT.1.2 Test #6 **[TD0499]** - NA 276
6.3.3.21 FCS_TLSC_EXT.1.2 Test #7 **[TD0499]** - NA 277
6.3.3.22 FCS_TLSC_EXT.1.3 Test #1a **[TD0513]** 277
6.3.3.23 FCS_TLSC_EXT.1.3 Test #1b **[TD0513]** 277
6.3.3.24 FCS_TLSC_EXT.1.3 Test #1c **[TD0513]**- NA 278
6.3.3.25 FCS_TLSC_EXT.1.3 Test #2 278
6.3.3.26 FCS_TLSC_EXT.1.3 Test #3 279
6.3.3.27 FCS_TLSC_EXT.1.3 Test #4 279
6.3.3.28 FCS_TLSC_EXT.2.1 Test #1 279
6.3.3.29 FCS_TLSC_EXT.2.1 Test #2 280
6.3.3.30 FCS_TLSC_EXT.5.1 Test #1 280
6.3.4 Crypto Test Cases 280
6.3.4.1 FCS_CKM.1 RSA[**TD0951**] 280
6.3.4.2 FCS_CKM.1 ECC **[TD0951]** 282
6.3.4.3 FCS_CKM.2 SP800-56A 282
6.3.4.4 FCS_CKM.2 RSA 284

6.3.4.5 FCS_CKM_EXT.4.1 Test #1 284
6.3.4.6 FCS_CKM.4.2 Test#1 – NA. 284
6.3.4.7 FCS_CKM.4.2 Test #2 – NA. 285
6.3.4.8 FCS_COP.1.1(1) AES-GCM 285
6.3.4.9 FCS_COP.1.1(2) Hashing 286
6.3.4.10 FCS_COP.1.1(3) SigGen ECDSA **[TD0951]** 287
6.3.4.11 FCS_COP.1.1(3) SigGen RSA 288
6.3.4.12 FCS_COP.1.1(4) Keyed-Hash 289
6.3.4.13 FCS_IV_EXT.1.1 289
6.3.4.14 FCS_RBG_EXT.1.1/AGT 289
6.3.4.15 FCS_RBG_EXT.1.2/AGT 291

7 CONCLUSION 291

1 TOE OVERVIEW

This Security Target is for the Common Criteria evaluation of the Ivanti Endpoint Manager Mobile (EPMM) System version 12 against the Protection Profile for Mobile Device Management, (PP_MDM_V4.0), Version 4.0; PP-Module for MDM Agents, (MOD_MDM_AGENT_V1.0) Version 1.0; and Functional Package for Transport Layer Security (TLS), (PKG_TLS_v1.1), Version 1.1.

The Target of Evaluation (TOE) is the Ivanti Endpoint Mobile Manager (EPMM) 12 System (Version 12). The TOE (EPMM) is a distributed TOE and consists of two software components: the EPMM MDM Server and the EPMM MDM Agent (also referred to as MDM Server and MDM Agent).The MDM Agent is installed on Android devices as an application that is supplied by Ivanti. This Ivanti CC evaluation includes the Mobile Device Management (MDM) and the Mobile Application Management (MAM) functionality of the Ivanti EPMM. The Mobile Content Management (MCM) functionality of the EPMM is excluded from the CC evaluation. Additionally, the Target of Evaluation (TOE) includes Mobile Application Store (MAS) functionality that hosts applications, authenticates Agents, and securely transmits applications to enrolled mobile devices. The MDM Server is deployed as a virtual machine and manages and monitors one or more instances of an MDM Agent. The MDM Agent is an application running on an Android device. This evaluation includes MDM Agent running on two different Android devices and operating system version:

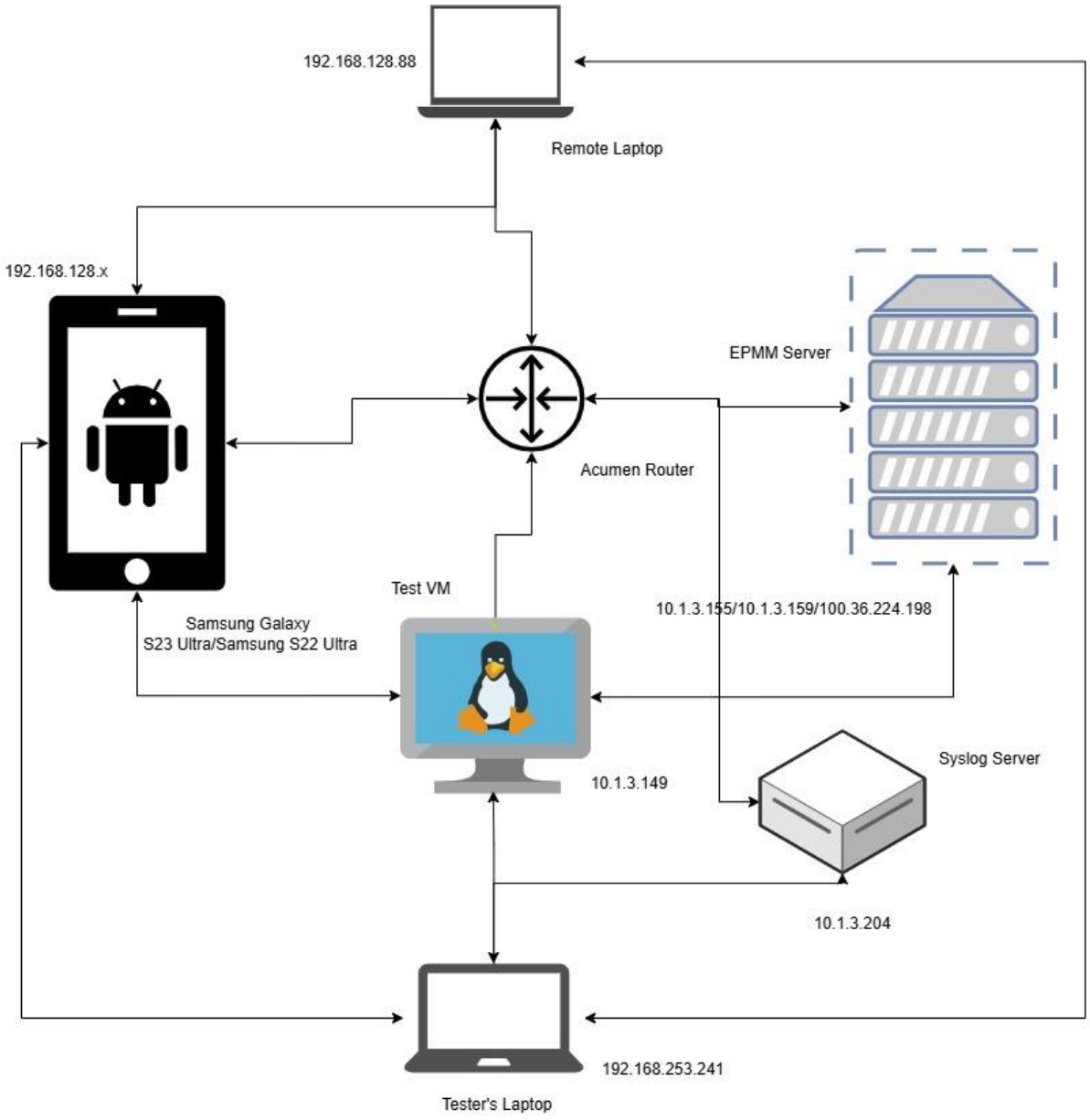
- Galaxy S22 Ultra 5G running Android 13
- Galaxy S23 Ultra 5G running Android 14

2 ASSURANCE ACTIVITIES IDENTIFICATION

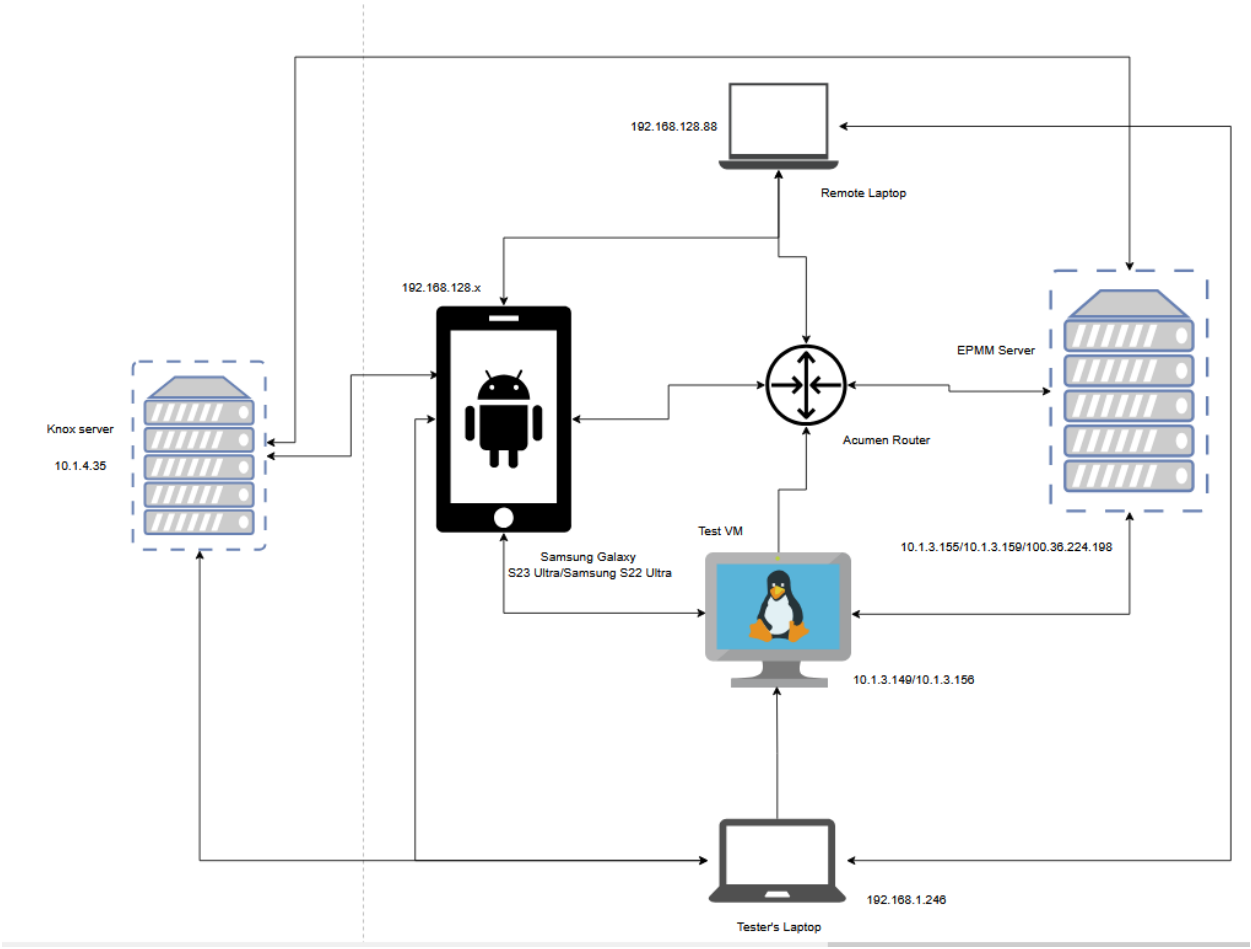
The Assurance Activities contained within this document include all those defined within the PP_MDM_V4.0, MOD_MDM_Agent_V1.0 and PKG_TLS_V1.1 based upon the core SFRs and those implemented based on selections within the PPs/EPs.

3 TEST BED DESCRIPTIONS

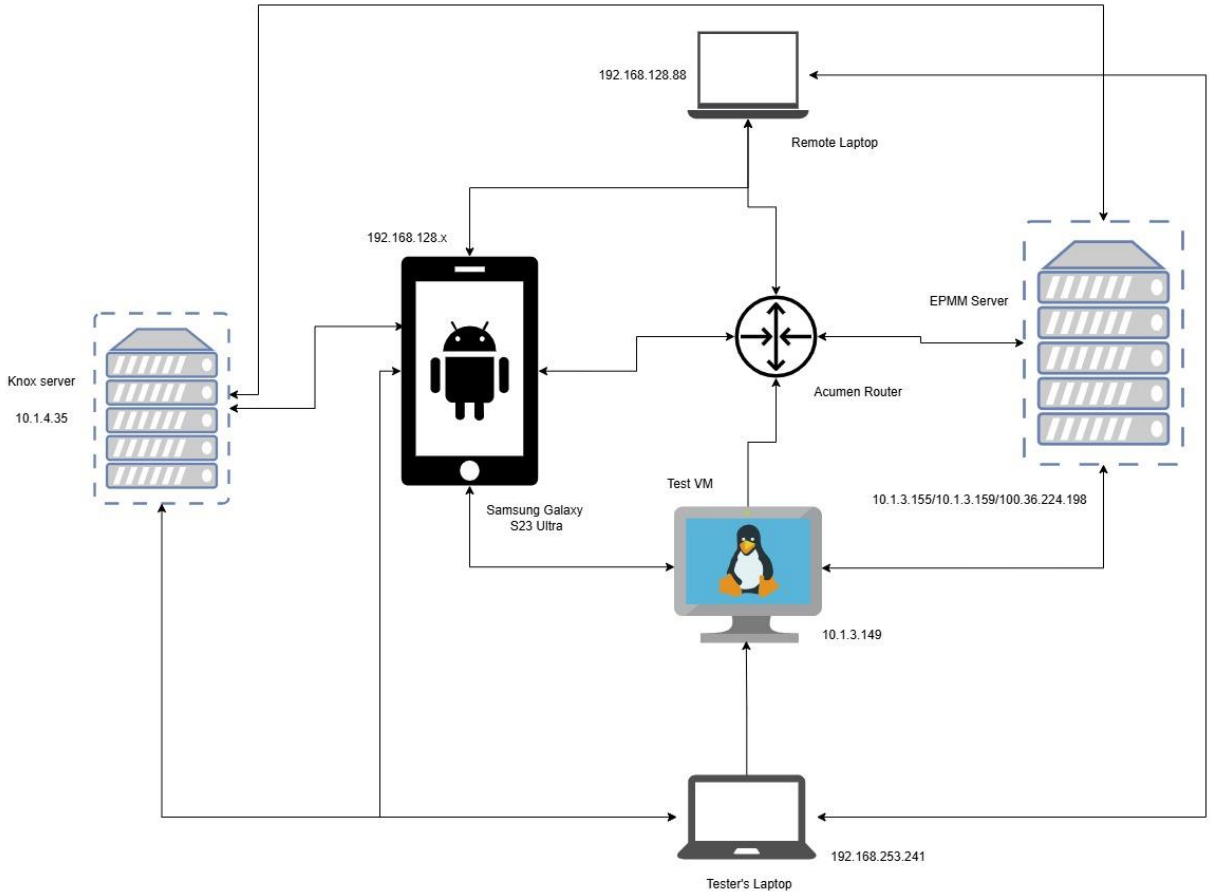
3.1 TEST BED (EPMM)



3.2 TEST BED – S22 (MDM AGENT, TLSC, AUTH)



3.3 TEST BED – S23 (MDM AGENT, TLSC, AUTH)



3.4 CONFIGURATION INFORMATION (EPMM)

The following table provides configuration information about each device in the test environment.

Device Details Table							
Device Details		**Network Details**			**System Details**		
Device Name	**Function**	**IP Address**	**MAC Address**	**Protocols**	**OS, including version**	**Timing Source**	**Software & Tools, including version**
EPMM Server	TOE	10.1.3.155	BA:AA:3F:0E:74:70	TLS v1.2		Manual	Mobileiron-12.3.1.0-59

		100.36.224.198	BA:AA:3F:0E:74:70	TLS v1.2	Mobileiron-12.3.1.0-59		
Samsung Galaxy S23 Ultra	TOE	192.168.128.0/24	A4:75:B9:22:79:A0	TLS v1.2,TCP	Android 14	NTP	MDMpp-MIClient-12.5.0.0.p.apk, TeamViewer host 15.63.667
Samsung Galaxy S22 Ultra	TOE	192.168.128.0/24	FE:26:15:d4:cc:7d	TLS v1.2,TCP	Android 13	NTP	MDMpp-MIClient-12.5.0.0.p.apk, TeamViewer host 15.63.667
Acumen Router	Lab Router	100.36.224.194	E0:CB:BC:40:5C:90	TLS, TCP, UDP	N/A	NTP	MX 18.107.10
Laptop	Tester's Laptop, ADB interface	192.168.253.241	00:05:9A:3C:7A:00	SSH, RDP	Windows 11 Pro	NTP	Wireshark 4.4.7, MobaXterm v23.2 Build 5082, ADB v1.0.41, Wireshark v4.4.7
Remote Laptop	Laptop hosting team viewer, ADB interface	192.168.128.88	74-70-FD-4B-7F-80	RDP, TCP	Windows 11 Pro	NTP	TeamViewer 15.63.4, ADB v1.0.41

Test VM on 10.1.3.149	HTTP server for CRL check & OCSP responder	10.1.3.149	00:50:56:8B:A4:B9	SSH, HTTP,RDP	Ubuntu 22.04.5 LTS	NTP	Wireshark 3.6.2, OpenSSL 3.0.2
Syslog Server on 10.1.3.204	Syslog server	10.1.3.204	00:50:56:8b:bd:75	SSH,RDP,TLS v1.2	Ubuntu 22.04.5 LTS	NTP	Wireshark 3.6.2, OpenSSL 3.0.2

Note: The mobile device IP's will be falling under the range 192.168.128.x as the device was connected over DHCP when registered and tested with different certificates.

3.5 CONFIGURATION INFORMATION – S22 (MDM AGENT, TLSC, AUTH)

Device Details Table							
Device Details		**Network Details**			**System Details**		
Device Name	**Function**	**IP Address**	**MAC Address**	**Protocols**	**OS, including version**	**Timing Source**	**Software & Tools, including version**
EPMM Server	TOE	10.1.3.155	BA:AA:3F:0E:74:70	TLS v1.2	Mobileiron-12.3.1.0-59	Manual	Mobileiron-12.3.1.0-59
		100.36.224.198	BA:AA:3F:0E:74:70	TLS v1.2			
		10.1.3.159	00:50:56:8b:aa:5b	TLS v1.2			
Samsung Galaxy S22 Ultra	TOE	192.168.128.x	A4:75:B9:22:79:A0	TLS v1.2,TCP	Android 13	NTP	MDMpp-MIClient-12.5.0.0 papk, Teamviwer host 15.63.667
Acumen Router	Lab Router	100.36.224.194	E0:CB:BC:40:5C:90	TLS, TCP, UDP	N/A	NTP	MX 18.10710

							N/A
Laptop	Tester's Laptop, ADB interface	192.168.1.246	CC-15-31-1A-C5-24	SSH, RDP	Windows 11 Pro	NTP	Wireshark 4 4.3, MobaXterm v23.2 Build 5082, ADB v1.0.41
Remote Laptop	Laptop hosting team viewer, ADB interface	192.168.128.88	74-70-FD-4B-7F-80	RDP, TCP	Windows 11 Pro	NTP	TeamViewer 15.63.4, ADB v1.0.41
Test VM on 10.1.3149	HTTP server for CRL check and TLS test server	10.1.3.149	00:50:56:8B:A4:B9	SSH, HTTP,RDP, TLS v1.2	Ubuntu 22.04.5 LTS	NTP	N/A
Test VM on 10.1.3.156	HTTP server for CRL check and TLS test server	10.1.3.156	00:50:56:8b:c0:ca	SSH, HTTP,RDP, TLS v1.2	Ubuntu 22.04.5 LTS	NTP	N/A

Note: The mobile device IP's will be falling under the range 192.168.128.x as the device was connected over DHCP when registered and tested with different certificates.

3.6 CONFIGURATION INFORMATION – S23 (MDM AGENT, TLSC, AUTH)

Device Details Table							
Device Details		**Network Details**			**System Details**		
Device Name	**Function**	**IP Address**	**MAC Address**	**Protocols**	**OS, including version**	**Timing Source**	**Software & Tools, including version**

EPMM Server	TOE	10.1.3.155	BA:AA:3F:0E:74:70	TLS v1.2	Mobileiron-12.3.1.0-59	Manual	Mobileiron-12.3.1.0-59
		100.36.224.198	BA:AA:3F:0E:74:70	TLS v1.2			
		10.1.3.159	00:50:56:8b:aa:5b	TLS v1.2			
Samsung Galaxy S23 Ultra	TOE	192.168.128.x	A4:75:B9:22:79:A0	TLS v1.2,TCP	Android 14	NTP	MDMpp-MIClient-12.5.0.0 papk, Teamviwer host 15.63.667
Acumen Router	Lab Router	100.36.224.194	E0:CB:BC:40:5C:90	TLS, TCP, UDP	N/A	NTP	MX 18.10710 N/A
Knox server	Samsung Knox server	10.1.4.35	00:50:56:8B:71:A5	HTTPS	Windows Server 2022 Standard Evaluation	NTP	knox-installer v2.9, Apache Tomcat 9.0, Microsoft SQL Server 2022
Laptop	Tester's Laptop, ADB interface	192.168.253.241	00:05:9A:3C:7A:00	SSH, RDP	Windows 11 Pro	NTP	Wireshark 4 4.7, MobaXterm v23.2 Build 5082, ADB v1.0.41
Remote Laptop	Laptop hosting team viewer, ADB interface	192.168.128.88	74-70-FD-4B-7F-80	RDP, TCP	Windows 11 Pro	NTP	TeamViewer 15.63.4, ADB v1.0.41

Test VM on 10.1.1 149	HTTP server for CRL check, OCSP respond er and TLS test server	10.1.3.149	00:50:56:8B:A4: B9	SSH, HTTP,RD P, TLS v1.2	Ubuntu 22.04.5 LTS	NTP	Wireshark 3.6.2, OpenSSL 3.0.2

Note: The mobile device IP's will be falling under the range 192.168.128.x as the device was connected over DHCP when registered and tested with different certificates

3.7 TEST TIME AND LOCATION

All testing was carried out at Acumen Security LLC located at 2400 Research Blvd, Suite 395, Rockville MD 20850. Testing occurred from December,2024 to October,2025.

The TOE was in a physically protected, access controlled, designated test lab with no unattended entry/exit ways. At the start of each day, the test bed was verified to ensure that it was not compromised. All evaluation documentation was always kept in a secure repository.

4 DETAILED TEST CASES (TSS AND AGD ACTIVITIES)

4.1 SECURITY FUNCTIONAL REQUIREMENTS

4.1.1 SECURITY AUDIT (FAU)

4.1.1.1 FAU_ALT_EXT.1 SERVER ALERTS *(MDM_PP)*

4.1.1.1.1 FAU_ALT_EXT.1.1 TSS

The evaluator shall examine the TSS and verify that it describes how the alert system is implemented.

Evaluator Findings:

The evaluator reviewed the TSS, section **6.1.1** titled "**FAU_ALT_EXT.1 Server Alerts (MDM_PP)"**, and verified that it describes how the alert system is implemented. The implementation details are

The MDM Server component of the TOE supports sending email Alerts. These alerts can also be viewed by an administrator on the MDM server. The MDM Server will alert the admin of the following changes:

- **Changes to enrolled devices.**
  - **Device Registered Alert: This alert is sent when a device is registered. Specifically, the Mobile Device user will be provided with the MDM Server's FQDN and login credentials instructing the user on how to enroll in the EPMM system.**
  - **Unenroll Alert: This alert is sent when a device is unenrolled (retired). Unenrollment is accomplished by the MDM Server sending a Wipe command to the device. Once wipe is initiated the device factory resets, thereby making the device to the preconfigured mode i.e. unenrolled.**
- **Failure to apply policies to a mobile device**
  - **Policy Update Failure Alert: This alert is sent to the administrator when a policy change issued by the MDM server fails to be applied on the MDM Agent. This alert can be viewed by the administrator under the devices tab where it will indicate that the policy is either not applied or unsupported.**

The evaluator shall also verify that a description of each assigned event is provided in the TSS.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.1** titled "**FAU_ALT_EXT.1 Server Alerts (MDM_PP)"**, and verified that a description of each assigned event is provided. The assigned events are **The MDM Server will alert the admin of the following changes:** • **Changes to enrolled devices.** ○ **Device Registered Alert: This alert is sent when a device is registered. Specifically, the Mobile Device user will be provided with the MDM Server's FQDN and login credentials instructing the user on how to enroll in the EPMM system.** ○ **Unenroll Alert: This alert is sent when a device is unenrolled (retired). Unenrollment is accomplished by the MDM Server sending a Wipe command to the device. Once wipe is initiated the device factory resets, thereby making the device to the preconfigured mode i.e. unenrolled.** • **Failure to apply policies to a mobile device** ○ **Policy Update Failure Alert: This alert is sent to the administrator when a policy change issued by the MDM server fails to be applied on the MDM Agent. This alert can be viewed by the administrator under the devices tab where it will indicate that the policy is either not applied or unsupported.**

Verdict:
PASS.

4.1.1.1.2 FAU_ALT_EXT.1.1 AGD

The evaluator shall examine the guidance document and verify that it describes how the alerts can be configured, if configurable.

Evaluator Findings:
The evaluator reviewed the guidance document, section **3.3.9** titled **Configuring Email Alerts** and section **7.3.6** titled **Configuring Server Alerts**, and verified that it describes how the alerts can be configured. The guidance document explains the steps to configure the SMTP server settings to enable email alerts, and further details how server alerts can be enabled and managed through the administrative console.

Verdict:
PASS.

4.1.1.2 FAU_ALT_EXT.2 AGENT ALERTS *(AGT_PPM)*

4.1.1.2.1 FAU_ALT_EXT.2 AGENT ALERTS TSS

The evaluator shall examine the TSS and verify that it describes how the alerts are implemented.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.2** titled **"FAU_ALT_EXT.2 Agent Alerts *(AGT_PMM)"***, and verified that it describes how the alerts are implemented. The implementation details are **The TOE supports the ability to periodically synchronize the MDM Server and MDM Agents. The synchronization includes retrieving information about the policies that are installed, thereby ensuring the MDM Server is informed about which policies have been applied.** **The Agents, in the evaluated configuration, are configured to check in every eight hours but this parameter can be modified.** **When a mobile device checks in, the mobile device performs a compliance check based on configured policies. If any of the settings have not been reported to and acknowledged by the MDM Server, the mobile device reports those changes. Hence, if something happens, such as a network disruption, that prevents the MDM Server from receiving the mobile device compliance information or prevents the mobile device from receiving an acknowledgement from the MDM Server, that information will be sent the next time the device connects until it is finally acknowledged.** **The MDM Agent will store unsent alerts in the application storage available on the mobile device and that space is limited only by the space available on the mobile device.**

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.2** titled **"FAU_ALT_EXT.2 Agent Alerts *(AGT_PMM)"***, and verified that it describes how the candidate policy updates are obtained and the actions that take place for successful and unsuccessful cases. The process details are **Refer to FMT_POL_EXT.1 below for a description of how policy updates are obtained and the actions that take place for success and unsuccessful cases.** Furthermore, the evaluator reviewed the TSS, section **6.5.4** titled **"FMT_POL_EXT.1 Trusted Policy Update *(MDM_PP) and* FMT_POL_EXT.2 Agent Trusted Policy Update *(AGT_PPM)*".** **Policies are distributed by the MDM Server to the MDM Agent on the mutually authenticated trusted TLS channel (FTP_ITC.1(2))** **The MDM Server provides digitally signed policies and policy updates to the MDM Agent.** **When the MDM Agent receives the policy or policy update, it creates a SHA-256 hash of its contents (less the signed hash), verifies the hash matches and then verifies that the hash was signed using the same MDM server portal certificate that was used to receive the policy or policy update. The MDM Agent will only install policies and policy updates if both the hash matches and the hash was signed by the expected certificate. If the verification fails, the policy or policy update is discarded**

and an alert is sent to the MDM Server, the agent will close the connection at the TCP layer, and will no longer send HTTP requests payloads to the server.
This process is repeated during the next check-In since the policy current on the MDM Server has not been applied.
When the MDM Agent checks-in with the MDM Server, the MDM Server will send any policy or policy updates to the MDM Agent via the secure TLS channel if the policy or policy update has not yet been applied by the MDM Agent.
If either step fails, the MDM Agent does not apply the policy and returns the success or failure of applying the policy to the server.

The software components that are performing the processing must also be identified in the TSS and verified by the evaluator.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.2** titled **"FAU_ALT_EXT.2 Agent Alerts *(AGT_PMM)"***, and confirmed that the software components performing the processing are identified. The identified components are **The TOE supports the ability to periodically synchronize the MDM Server and MDM Agents. The synchronization includes retrieving information about the policies that are installed, thereby ensuring the MDM Server is informed about which policies have been applied.**

The 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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.2** titled **"FAU_ALT_EXT.2 Agent Alerts *(AGT_PMM)"***, and ensured that it describes how reachability events are implemented, and if configurable, are selected in FMT_SMF_EXT.4.2. The description indicates that **The TOE supports the ability to periodically synchronize the MDM Server and MDM Agents.** **The Agents, in the evaluated configuration, are configured to check in every eight hours but this parameter can be modified.** **When a mobile device checks in, the mobile device performs a compliance check based on configured policies.**

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.2** titled **"FAU_ALT_EXT.2 Agent Alerts *(AGT_PMM)"***, and verified that it describes the circumstances under which the alert may not be generated, how alerts are queued, and the maximum amount of storage for queued messages. The circumstances and queuing

details are **if something happens, such as a network disruption, that prevents the MDM Server from receiving the mobile device compliance information or prevents the mobile device from receiving an acknowledgement from the MDM Server, that information will be sent the next time the device connects until it is finally acknowledged. The MDM Agent will store unsent alerts in the application storage available on the mobile device and that space is limited only by the space available on the mobile device.**

Verdict:
PASS.

4.1.1.2.2 FAU_ALT_EXT.2 AGENT ALERTS AGD

None.

4.1.1.3 FAU_CRP_EXT.1 SUPPORT FOR COMPLIANCE REPORTING OF MOBILE DEVICE CONFIGURATION *(MDM_PP)*

4.1.1.3.1 FAU_CRP_EXT.1.1 TSS

None.

4.1.1.3.2 FAU_CRP_EXT.1.1 AGD

The evaluator shall check to ensure that the operational guidance contains instructions on how to access the MDM Server's compliance reporting interface.

Evaluator Findings:
The evaluator reviewed the AGD, section **2.2** titled **"ACCESSING ADMINISTRATIVE INTERFACES", 5.2** titled **"DEVICE ENROLLMENT PROCESS"**, section **7.2** titled **"FMT_SMF SFR MANAGEMENT COMMANDS"** and section **7.3** titled **"SYSTEM MANAGEMENT COMMANDS"**, and verified that it contains instructions on how to access the MDM Server's compliance reporting interface. The AGD states that the compliance reporting interface for the Mobile Device Management (MDM) server is accessed through the administrative portal. This portal is available via a secure web connection at the URL https://<server-fqdn>:443/mifs, where <server-fqdn> represents the fully qualified domain name of the server. Including this information ensures that administrators and evaluators can reliably locate and utilize the interface to monitor device compliance and generate necessary reports as part of their management and oversight responsibilities.

Verdict:
PASS.

4.1.1.4 FAU_GEN.1(1) AUDIT DATA GENERATION *(MDM_PP)*

4.1.1.4.1 FAU_GEN.1.1(1) TSS

The evaluator shall check the TSS and ensure that it lists all of the auditable events.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.4** titled **"FAU_GEN.1(1) Audit Data Generation *(MDM_PP)*"**, and verified that it lists all of the auditable events. The listed auditable events are **The MDM Server component of the TOE produces audit events reporting multiple activities. The following types of audit events are supported by the MDM Server:** • **Administrative Actions: the TSF will produce an audit record for every Administrative command issued by an Administrator from the Admin Portal and the System Manager.** • **Required Audit Events: This type includes every required audit record identified by the PP and Module. Specifically,**   ○ **The defined auditable events listed in Table 16 of the ST required by the MDM Server.**   ○ **The defined auditable events listed in FAU_GEN.1(2)/SRV required by the MAS functionality.**

The evaluator shall check to make sure that every audit event type mandated by the PP is described in the TSS.

The evaluator reviewed the TSS, section **6.1.4** titled "**FAU_GEN.1(1) Audit Data Generation (MDM_PP)"**, and verified that every audit event type mandated by the PP is described in the TSS. The mandated audit event types are as per the TSS are as stated below **Required Audit Events: This type includes every required audit record identified by the PP and Module. Specifically,** **The defined auditable events listed in Table 15 required by the MDM Server.** **The defined auditable events listed in FAU_GEN.1(2)/SRV required by the MAS functionality.**

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).

The evaluator reviewed the TSS, section **6.1.4** titled "**FAU_GEN.1(1) Audit Data Generation (MDM_PP)"**, and verified that for every audit event described in the TSS, the description indicates where the audit event is generated (TSF, TOE platform). The generation locations are • **who generated the audit event (TSF or 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).

The "invoke platform-provided functionality" is not a selection; hence, this assurance activity is not applicable.

Verdict:

PASS.

4.1.1.4.2 FAU_GEN.1.1(1) AGD

The evaluator shall check the administrative guide and ensure that it lists all of the auditable events.

Evaluator Findings:
The evaluator reviewed the administrative guide, section **11** titled **"EPMM System Audit Record Examples"**, and verified that it lists all of the auditable events. The guide provides a comprehensive table of audit events along with their corresponding messages, ensuring that administrators can identify the events generated by the TOE.

The evaluator shall check to make sure that every audit event type mandated by the PP is described.

Evaluator Findings:
The evaluator also checked to ensure that every audit event type mandated by the PP is described. The administrative guide describes the following audit event types mandated by the PP: The audit event types mandated by the Protection Profile (PP) are detailed in section 11.1 "AUDIT EVENTS AND THEIR CORRESPONDING MESSAGES" within the guidance documentation. This section outlines the specific audit requirements associated with the TOE, ensuring traceability and compliance with PP_MDM_v4.0.

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.

Evaluator Findings:
The evaluator determined the administrative actions that are relevant in the context of this PP include those listed in the Management section. The administrative actions include operations that directly impact the security posture of the TOE, such as locking a device, performing a full wipe of protected data, unenrolling a device from management, installing policies, importing X.509v3 certificates, installing or removing applications, and updating system software. Each of these actions either enforces a security policy, modifies the state of the device in a way that affects confidentiality or integrity, or interacts with trust anchors and update mechanisms. These actions are listed in the "**EPMM Server Management**" section of the guidance document and are aligned with the Security Functional Requirements (SFRs) defined in the PP.

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.

Evaluator Findings:
The evaluator also determined 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 administrative actions including installing policies that define device behavior, importing certificates into the Trust Anchor Database to establish secure

communications, updating system software to maintain trusted integrity, and managing applications that may introduce or remove functionality. Additionally, commands such as Force Device Check-In and adjustments to the Sync Interval parameter influence the timing and enforcement of these configurations. These commands are essential for enabling or disabling security mechanisms and ensuring that the TOE operates in compliance with the PP_MDM_v4.0.

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.

Evaluator Findings:
The methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to this PP are performed as part of the activities associated with AGD_OPE.

Verdict:

PASS.

4.1.1.4.3 FAU_GEN.1.2(1) TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.4** titled **"FAU_GEN.1(1) Audit Data Generation *(MDM_PP)*"**, and verified that it provides a format for audit records. Each audit record format type is covered, along with a brief description of each field. The audit record formats and field descriptions are **Audit records generated by the MDM Server that are displayed via the Admin Portal Logs GUI page have the following format:** **ACTION | STATE | PERFORMD BY | ACTION DATE | COMPLETED AT | PERFORMED ON** **The information panel displays:** • **ACTION - a short description of the event.** • **STATE – success or failure.** • **PERFORMED BY – login of person who caused the event.** • **ACTION DATE – date the event was started. Specifically, day of week | month | day | year | time | time zone.** • **COMPLETED AT – date the event occurred. Specifically, day of week | month | day | year | time | time zone.** • **PERFORMED ON – the system the event occurred on.** **Audit records generated by the MDM Server that are sent to the syslog server have the following format. The following is an example of a syslog message and the fields are described below.** **<13>Oct 22 12:34:56 myhostname myapp[1234] This is a sample syslog message.**

- **<13> = Priority: A numberical value that combines the facility and severity level of the message. It is enclosed in angle brackets.**
- **Oct 22 12:34:56 = Timestamp: Indicates the time when the message was generated. It is typically in the format of MMM DD HH:MM:SS (month, day, hour, minute, second).**
- **myhostname = Hostname: Identifies the origin of the syslog message, indicating the system or device that generated the log.**
- **myapp[1234] = Tag: Tag or Process ID represents the application or process that generated the message. It is often user-defined.**
- **This is a sample syslog message = Message: The actual log information or event description.**

Verdict:

PASS.

4.1.1.4.4 FAU_GEN.1.2(1) AGD

The evaluator shall check the administrative guide and ensure that it provides a format for audit records.

Evaluator Findings:
The evaluator reviewed the AGD, section **10.2** titled **"VIEWING AUDITABLE EVENTS ON EPMM SERVER"**, and verified that it provides a format for audit records. The audit record format types is presented through the information panel, which displays key fields associated with each audit event. These fields include ACTION, STATE, PERFORMED BY, ACTION DATE, COMPLETED AT, and PERFORMED ON. This structured presentation ensures that audit records are consistently formatted and interpretable by administrators.

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).

Evaluator Findings:
The evaluator also verified that the AGD provides a brief description of each field in the audit records. The fields are mentioned as below, • ACTION provides the type of event, fulfilling the requirement to identify the nature of the security-relevant action. • STATE indicates the outcome of the event (success or failure). • PERFORMED BY captures the identity of the subject (administrator) who initiated the action. • ACTION DATE and COMPLETED AT together provide the full timestamp of the event, including day, date, time, and time zone. • PERFORMED ON identifies the system or device on which the event occurred. The evaluator ensured that the description of the fields contains the information required in FAU_GEN.1.2(1).

Verdict:

PASS.

4.1.1.5 FAU_GEN.1(2)/AGT AUDIT DATA GENERATION *(AGT_PPM)*

4.1.1.5.1 FAU_GEN.1(2)/AGT AUDIT DATA GENERATION TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.5** titled **"FAU_GEN.1(2)/AGT Audit Data Generation *(AGT_PPM)*"**, and verified that it provides a format for audit records. Each audit record format type is covered, along with a brief description of each field. The audit record formats and field descriptions are **Audit records generated by the MDM Agent that are generated have the following format. The following is an example of an audit message and the fields are described below.** **2025-05-21 16:26:47.99 32754 676 E=OpenSSLWrapper: error:0A0000F8:SSL routines:::unknown cipher returned in ssl/statem/statem_clnt c (set_client_ciphrsuite): 1313 data: flages:** • **2025-05-21 *= Date: the date the message was generated (year-month-day).*** • **16:26:47.99 = Timestamp: Indicates the time when the message was generated. It is in the format of HH:MM:SS (hour, minute, second).** • **32754 = Process ID: Process ID represents the application or process that generated the message.** • ***676 =* Message ID*: A numerical value that represents the message ID.*** • ***E=* Severity*: The Severity value of the message. Available values are:***   ○ **Alert**   ○ **Critical**   ○ **Error (the default)**   ○ **Warning**   ○ **Notice**   ○ **D*ebug***   ○ **Information** • **OpenSSLWrapper:…. = Message: The actual log information or event description.** **The information displayed as Process ID and Message ID will vary based on the Severity of the audit record. Agent audit records are identified in Appendix A.**

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).

Evaluator Findings:
The "invoke platform-provided functionality" is not a selection in the ST; hence, this activity is not applicable.

Verdict:

PASS.

4.1.1.5.2 FAU_GEN.1(2)/AGT AUDIT DATA GENERATION AGD

None.

4.1.1.6 FAU_GEN.1(2)/SRV AUDIT GENERATION (MAS SERVER) *(MDM_PP)*

4.1.1.6.1 FAU_GEN.1.1(2)/SRV TSS

The evaluator shall check the TSS and ensure that it provides a format for audit records.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.6 titled "FAU_GEN.1(2)/SRV Audit Generation (MAS Server) (MDM_PP)"**, and verified that it provides a format for audit records. The TSS states that **MAS Server events generated by the Admin Portal are available for view using the Admin Portal > Logs option. Refer to 6.1.4 FAU_GEN.1(1) Audit Data Generation *(MDM_PP)* for a description of the format of the audit logs.** Upon further investigation, the section 6.1.4 FAU_GEN.1(1) Audit Data Generation *(MDM_PP)* states that **Audit records generated by the MDM Server that are displayed via the Admin Portal Logs GUI page have the following format:** **ACTION | STATE | PERFORMD BY | ACTION DATE | COMPLETED AT | PERFORMED ON** **The information panel displays:** • ***ACTION - a short description of the event.*** • ***STATE – success or failure.*** • ***PERFORMED BY – login of person who caused the event.*** • ***ACTION DATE – date the event was started. Specifically, day of week | month | day | year | time | time zone.*** • ***COMPLETED AT – date the event occurred. Specifically, day of week | month | day | year | time | time zone.*** • ***PERFORMED ON – the system the event occurred on.*** • ***DETAILS – displays the outcome of the event*** **Audit records generated by the MDM Server that are sent to the syslog server have the following format. The following is an example of a syslog message and the fields are described below.** **<13>Oct 22 12:34:56 myhostname myapp[1234] This is a sample syslog message.** • ***<13> = Priority: A numerical value that combines the facility and severity level of the message. It is enclosed in angle brackets.*** • **Oct 22 12:34:56 = Timestamp: Indicates the time when the message was generated. It is typically in the format of MMM DD HH:MM:SS (month, day, hour, minute, second).** • ***myhostname = Hostname: Identifies the origin of the syslog message, indicating the system or device that generated the log.*** • **myapp[1234] = Tag: Tag or Process ID represents the application or process that generated the message. It is often user-defined.**

- **This is a sample syslog message = Message: The actual log information or event description.**

Verdict:
PASS.

4.1.1.6.2 FAU_GEN.1.1(2)/SRV AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **10.2** titled **"VIEWING AUDITABLE EVENTS ON EPMM SERVER"**, and verified that it provides a format for audit records. The AGD includes the following audit record format types: ACTION, STATE, PERFORMED BY, ACTION DATE, COMPLETED AT, and PERFORMED ON. The evaluator also verified that the AGD provides a brief description of each field in the audit records. The AGD describes the fields as follows: • ACTION – a short description of the event. • STATE – success or failure. • PERFORMED BY – login of person who caused the event. • ACTION DATE – date the event was started. Specifically, day of week | month | day | year | time zone. • COMPLETED AT – date the event occurred. Specifically, day of week | month | day | year | time | time zone. • PERFORMED ON – the system on which the event occurred on. • DETAILS – displays the outcome of the event

Verdict:
PASS.

4.1.1.6.3 FAU_GEN.1.2(2)/SRV TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.6** titled **"FAU_GEN.1(2)/SRV Audit Generation (MAS Server) (MDM_PP)"**, and verified that it provides a format for audit records. Each audit record format type is covered, along with a brief description of each field. The details are The MAS Server functionality of the TOE will include the following information in each MAS Server event: • date and time of the event • type of event • mobile device identity.

Verdict:
PASS.

4.1.1.6.4 FAU_GEN.1.2(2)/SRV AGD

The evaluator shall check the administrative guide and ensure that it provides a format for audit records.

Evaluator Findings:
The evaluator reviewed the AGD, section **10.2** titled **"VIEWING AUDITABLE EVENTS ON EPMM SERVER"** and verified that it provides a format for audit records. The audit record format types are structured The information panel includes fields such as ACTION, which identifies the type of event; STATE, indicating the success or failure of the action; PERFORMED BY, which captures the identity of the user who initiated the event; ACTION DATE and COMPLETED AT, which together provide a comprehensive timestamp including day, date, time, and time zone; and PERFORMED ON, specifying the system where the event occurred.

Each audit record format type must be covered, along with a brief description of each field.

Evaluator Findings:
The evaluator reviewed the AGD, section **10.2** titled **"VIEWING AUDITABLE EVENTS ON EPMM SERVER"**, and verified that each audit record format type is covered, along with a brief description of each field. Each audit record type is as mentioned below • ACTION provides the type of event, fulfilling the requirement to identify the nature of the security-relevant action. • STATE indicates the outcome of the event (success or failure). • PERFORMED BY captures the identity of the subject (administrator) who initiated the action. • ACTION DATE and COMPLETED AT together provide the full timestamp of the event, including day, date, time, and time zone. • PERFORMED ON identifies the system or device on which the event occurred

The evaluator shall check to make sure that the description of the fields contains the information required in FAU_GEN.1.2(2).

Evaluator Findings:
The evaluator reviewed the AGD, section **10.2** titled **"VIEWING AUDITABLE EVENTS ON EPMM SERVER"**, and verified that the description of the fields contains the information required in FAU_GEN.1.2(2). The AGD states that the audit records display the following information: • ACTION provides the type of event, fulfilling the requirement to identify the nature of the security-relevant action. • STATE indicates the outcome of the event (success or failure). • PERFORMED BY captures the identity of the subject (administrator) who initiated the action. • ACTION DATE and COMPLETED AT together provide the full timestamp of the event, including day, date, time, and time zone. • PERFORMED ON identifies the system or device on which the event occurred

Verdict:
PASS.

4.1.1.7 FAU_NET_EXT.1 NETWORK REACHABILITY REVIEW *(MDM_PP)*

4.1.1.7.1 FAU_NET_EXT.1.1 TSS

The evaluator ensures that the TSS describes how reachability events are implemented for each supported mobile platform.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.7** titled **"FAU_NET_EXT.1 Network Reachability Review *(MDM_PP)"***, and verified that it describes how reachability events are implemented for each supported mobile platform. The implementation details are **Admin Portal Administrators can view whether an Android device is connected or not by viewing the Device & Users > Device page. Each Android device will display whether the device is Active or Not Active.** **This status information is as recent as the previous check-in (MDM Agent initiated). If an Administrator wants the current information they can select Action > Force Device Check-In for a specific device from the Admin Portal. This will force a check-in and determine if the device is reachable (MDM Server).**

The evaluator verifies that this description clearly indicates who (MDM Agent or MDM Server) initiates reachability events.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.7** titled **"FAU_NET_EXT.1 Network Reachability Review *(MDM_PP)"***, and verified that the description clearly indicates who (MDM Agent or MDM Server) initiates reachability events. The initiator of the reachability events is **Admin Portal Administrators can view whether an Android device is connected or not by viewing the Device & Users > Device page. Each Android device will display whether the device is Active or Not Active.** **This status information is as recent as the previous check-in (MDM Agent initiated). If an Administrator wants the current information they can select Action > Force Device Check-In for a specific device from the Admin Portal. This will force a check-in and determine if the device is reachable (MDM Server).days.**

Verdict:
PASS.

4.1.1.7.2 FAU_NET_EXT.1.1 AGD

The evaluator shall verify that the guidance instructs administrators on the method of determining the network connectivity status of an enrolled agent.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.2.5** titled **"QUERY CONNECTIVITY STATUS"**, and verified that it instructs administrators on the method of determining the network connectivity status of an enrolled agent. The AGD states that it instructs administrators to verify the network connectivity status of an enrolled agent via the Admin Portal by navigating Devices & Users > Devices. The device's current status is displayed in the STATUS column, typically indicating whether it is active. For deeper verification, administrators can access the device detail window by double-clicking the DISPLAY NAME, which reveals both the Status and Last Check-in timestamp providing a reliable method to assess connectivity and agent responsiveness within the evaluated configuration.

Verdict:
PASS.

4.1.1.8 FAU_SAR.1 AUDIT REVIEW *(MDM_PP)*

4.1.1.8.1 FAU_SAR.1.1 TSS

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).

Evaluator Findings:
The functionality is not invoked by the platform-provided functionality; hence, this activity is not applicable.

Verdict:
PASS.

4.1.1.8.2 FAU_SAR.1.1 AGD

None.

4.1.1.8.3 FAU_SAR.1.2 TSS

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).

Evaluator Findings:
The "invoke platform-provided functionality" is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.1.8.4 FAU_SAR.1.2 AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **10.2** titled **"VIEWING AUDITABLE EVENTS ON EPMM SERVER"**, and verified that it describes how the administrator accesses the audit data and the format of the audit record. The administrator to access audit data through the Admin Portal by navigating to Logs > Audit Logs, which opens the Audit Logs page in EPMM. The format of the audit records is well-defined in the information panel, displaying fields such as ACTION (event description), STATE (success or failure), PERFORMED BY (user login), ACTION DATE and COMPLETED AT (timestamps with full date and time zone), PERFORMED ON (system where the event occurred) and DETAILS (displays the outcome of the event). This structure ensures administrators can efficiently retrieve and interpret audit data.

Verdict:
PASS.

4.1.1.9 FAU_SEL.1(2) SECURITY AUDIT EVENT SELECTION *(AGT_PPM)*

4.1.1.9.1 FAU_SEL.1(2) SECURITY AUDIT EVENT SELECTION TSS

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).

Evaluator Findings:
The functionality is not invoked by the platform-provided functionality; hence, this activity is not applicable.

Verdict:
PASS.

4.1.1.9.2 FAU_SEL.1(2) SECURITY AUDIT EVENT SELECTION AGD

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).

Evaluator Findings:
The evaluator reviewed the AGD, section **10.3** titled **"CONFIGURING AUDIT EVENTS ON ANDROID DEVICES"**, and verified that the AGD contains instructions on how to define the set of auditable events as well as explains the syntax for multi-value selection, if applicable. The AGD states that the administrators are provided clear instructions for defining auditable events through the Samsung General Policy configuration in the Admin Portal. Administrators can enable Audit Collection Controls and configure event logging based on Severity Rule, Outcome Rule, and Audit Groups. Multi-value selection is supported for Audit Groups, allowing administrators to select multiple categories such as Security, System, Network, Events, and Applications.

The evaluator shall also verify that the operational guidance identifies those audit records that are always recorded, regardless of the selection criteria currently being enforced.

Evaluator Findings:
The evaluator reviewed the AGD, section **10.3** titled **"CONFIGURING AUDIT EVENTS ON ANDROID DEVICES"**, and verified that the AGD identifies those audit records that are always recorded, regardless of the selection criteria currently being enforced. The AGD states that regardless of the selection criteria applied critical events such as: i. Audit configuration change ii. TLS session failure iii. Transition to locked state iv. Enrollment in management v. Unenrollment from management vi. Policy validation failure vii. ID certificate import viii. ID certificate removal ix. Install application x. Remove application xi. Device OS upgrade xii. Server request client check-in xiii. Configure App Store quarantine compliance action Are considered security relevant and therefore not subject to the selection criteria. Additionally, events such as policy signature related audits and agent alerts are not stored in the platform audit log and these events are audited at all times meaning these cannot be deselected and are stored in the locally available dedicated storage until delivered to the MDM server.

Verdict:

PASS.

4.1.1.10 FAU_STG_EXT.1 EXTERNAL TRAIL STORAGE *(MDM_PP)*

4.1.1.10.1 FAU_STG_EXT.1.1 TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.10** titled **FAU_STG_EXT.1 External Trail Storage *(MDM_PP) and* FAU_STG_EXT.2 Audit Event Storage *(MDM_PP)***, and verified that it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. The details are **In summary, the MDM Server supports three TLS Client channels (Line E Figure 1** of ST**) (Bouncy Castle to Syslog), (Line F Figure 1** of ST**) (OpenSSL to Syslog) and (Line G Figure 1** of ST**) (OpenSSL to HTTPS) Figure 1** of ST**) to transfer audit data to the Audit Server. The Audit Server supports both HTTPS and Syslog to support the export of audit data.**

Verdict:
PASS.

4.1.1.10.2 FAU_STG_EXT.1.1 AGD

The evaluator shall 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.

Evaluator Findings:
The evaluator reviewed the AGD, section **6.2.1** titled **"FOR MDM SERVER"**, and verified that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. audit data is transmitted simultaneously to both the local EPMM system and the external Syslog servers configured by the administrator. This concurrent transmission makes the audit records accessible in both locations at the time of generation. Locally stored audit data remains available until it is removed through the system's default log rotation process. Administrators can access and review module-specific audit logs directly via the System Manager interface by navigating to Troubleshooting > Logs > View Module Logs.

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **6.2.1** titled **"FOR MDM SERVER"**, and verified that 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. The trusted channel to configure a SysLog server, first log into System Manager and navigate to Settings > Data

Export > SysLog Servers. Click Add to open the Add SysLog window, then modify the necessary fields based on the provided table for reference. Once all required changes have been made, click Apply, followed by OK, to save the configuration. The audit server is required to comply with the standards outlined in RFC 6125 Section 6 for all communications. To communicate with the audit server, the TOE must be configured with a valid Distinguished Name (DN) or Subject Alternative Name (SAN) that matches the identifier configured on the audit server. The TOE automatically performs this matching during certificate validation. The syslog server must use the correct DNS name as its identifier, and users are expected to upload a valid certificate chain for the server.

Verdict:

PASS.

4.1.1.11 FAU_STG_EXT.2 AUDIT EVENT STORAGE *(MDM_PP)*

4.1.1.11.1 FAU_STG_EXT.2.1 TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

If "implement functionality" is selected, the evaluator shall ensure that the TSS describes how the audit records are protected from unauthorized modification or deletion. The evaluator shall ensure that the TOE uses audit trail specific protection mechanisms.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.10** titled **“FAU_STG_EXT.1 External Trail Storage (MDM_PP) and FAU_STG_EXT.2 Audit Event Storage (MDM_PP)”**, and ensured that it describes how the audit records are protected from unauthorized modification or deletion. Audit records are protected from unauthorized modification or deletion, **The MDM Server component of the TOE provides the functions necessary for an administrator to review all of the collected audit records, while ensuring that the audit records cannot be modified. The MDM Server doesn’t offer any functions that allow the audit log or individual audit records therein to be modified or inserted. The MDM server does offer authorized Administrators a way to clear the audit data. This functionality is used to delete MDM Agent audit records, resident on the MDM Server, after they were exported to the Audit Server**.

Verdict:

PASS.

4.1.1.11.2 FAU_STG_EXT.2.1 AGD

None.

4.1.2 COMMUNICATION (FCO)

4.1.2.1 FCO_CPC_EXT.1 COMPONENT REGISTRATION CHANNEL DEFINITION

4.1.2.1.1 FCO_CPC_EXT.1.1 TSS

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). 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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.2.1.2 FCO_CPC_EXT.1.1 AGD

None.

4.1.2.1.3 FCO_CPC_EXT.1.2 TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.2.1.4 FCO_CPC_EXT.1.2 AGD

None.

4.1.2.1.5 FCO_CPC_EXT.1.3 TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.2.1.6 FCO_CPC_EXT.1.3 AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **5** titled “**DEVICE ENROLLMENT**”, section **7.2.2** titled ”**FULL WIPE OF PROTECTED DATA**”, and verified that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE. The AGD provides detailed instructions on how communication with an individual component of a distributed can be enabled and disabled.

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).

Evaluator Findings:
The evaluator reviewed the AGD, section **7.2.2** titled “**FULL WIPE OF PROTECTED DATA**”, and verified that the method of disabling communications ensures that all other components can be prevented from communicating with the component that is being removed from the TOE. The AGD states that **Once the device is unenrolled/wiped from the MDM Server, any communication from the MDM Server to the MDM Agent is disabled and visa versa.**

Verdict:
PASS.

4.1.3 CRYPTOGRAPHIC SUPPORT (FCS)

4.1.3.1 FCS_CKM.1 CRYPTOGRAPHIC KEY GENERATION *(MDM_PP)*

4.1.3.1.1 FCS_CKM.1.1 TSS

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 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).

Evaluator Findings:
The "invoke platform-provided functionality" is not selected; hence, this activity is not applicable.

If "implement functionality" is selected:

The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.1** titled "**FCS_CKM.1 Cryptographic Key Generation (MDM_PP)**", and ensured that it identifies the key sizes supported by the TOE. The TOE supports RSA key generation schemes as specified in FIPS PUB 186-5, with key sizes of 2048 bits, 3072 bits, and 4096. The TOE also supports ECC schemes using "NIST curves" P-256, P-384, and P-521 as specified in FIPS PUB 186-5, Appendix B.4. Additionally, if the ST specifies more than one scheme, the evaluator examined the TSS to verify that it identifies the usage for each scheme. The usage for each scheme is for key establishment in TLS communication channels, with the MDM Server acting as both sender and receiver, and the MDM Agent as sender.

Verdict:
PASS.

4.1.3.1.2 FCS_CKM.1.1 AGD

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation schemes and key sizes for all uses defined in this PP.

Evaluator Findings:
The evaluator reviewed the AGD, section **6.1.1** titled "**CONFIGURING CIPHER SUITES**", and verified that it instructs the administrator how to configure the TOE to use the selected key generation schemes and key sizes for all uses defined in this PP. The AGD states that The AGD guidance provides explicit instructions for configuring the TOE to utilize approved key generation schemes and key sizes in accordance with the Protection Profile. Administrators are directed to configure incoming and outgoing cipher suites via the System Manager interface under Security > Advanced settings. In Common Criteria mode, only TLSv1.2 is permitted, and the TOE enforces rejection of deprecated SSL/TLS versions. The guidance lists the exact cipher suites allowed in the evaluated configuration, including RSA- and ECC-based options defined in RFCs 5288 and 5289, ensuring that the TOE operates exclusively with cryptographic parameters aligned. The AGD states the following note "**The key generation parameters**

(such as scheme and key size) are determined by the cryptographic algorithm or cipher suite selected, and the keys are generated accordingly".

Verdict:
PASS.

4.1.3.2 FCS_CKM.2 CRYPTOGRAPHIC KEY ESTABLISHMENT *(MDM_PP)*

4.1.3.2.1 FCS_CKM.2.1 TSS

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 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).

Evaluator Findings:
The "invoke platform-provided functionality" is not selected; hence, this activity is not applicable.

If "implement functionality" is selected:

The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.2** titled "**FCS_CKM.2 Cryptographic Key Establishment (MDM_PP)**", and ensured that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. The supported key establishment schemes are the TOE supports RSA and ECC key generation schemes, which directly correspond to the key establishment schemes identified—RSAES-PKCS1-v1_5 and ECC-based key exchange per NIST SP 800-56A Rev. 2 Additionally, the ST specifies more than one scheme, the evaluator examined the TSS to verify that it identifies the usage for each scheme. The usage for each scheme is these schemes are used for TLS communication, with the MDM Server acting as both sender and receiver, and the MDM Agent as sender. The cryptographic libraries implementing these schemes are included within the TOE's physical boundary and are CAVP certified.

The evaluator shall ensure that the TSS describes how the TOE handles decryption errors. In accordance with NIST Special Publication 800-56B, the TOE must not reveal the particular error that occurred, either through the contents of any outputted or logged error message or through timing variations.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.2** titled "**FCS_CKM.2 Cryptographic Key Establishment (MDM_PP)**", and ensured that it describes how the TOE handles decryption errors. The decryption error handling mechanism is if a decryption error occurs, the TLS handshake fails silently—no connection is established, no error message is output, and no log entry is generated—ensuring that no sensitive information is leaked through observable behavior. The evaluator also ensured that the TOE does not reveal the particular error that occurred through any outputted or logged error message or through timing variations.

If KTS-OAEP is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.2.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each.

Evaluator Findings:
KTS-OAEP is not supported; hence, this assurance activity is not applicable.

If KTS-KEM-KWS is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.3.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each.

Evaluator Findings:
KTS-KEM-KWS is not supported; hence, this assurance activity is not applicable.

If Diffie-Hellman group 14 is selected from FCS_CKM.2.1, the TSS shall describe how the implementation meets RFC 3526 Section 3.

Evaluator Findings:
Diffie-Hellman group 14 is not selected; hence, this assurance activity is not applicable.

Verdict:
PASS.

4.1.3.2.2 FCS_CKM.2.1 AGD

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

Evaluator Findings:
The evaluator reviewed the AGD, section **6.1.1.1** titled "**KEY AGREEMENT PARAMETERS**", and verified that it instructs the administrator how to configure the TOE to use the selected key establishment scheme(s). The AGD states that "**The product does not require any additional configuration to utilize the ECDSA and RSA based key exchange algorithms, as these are implicitly applied when the respective cipher suites are selected**".

Verdict:
PASS.

4.1.3.3 FCS_CKM_EXT.4 CRYPTOGRAPHIC KEY DESTRUCTION *(MDM_PP)*

4.1.3.3.1 FCS_CKM_EXT.4.1 TSS

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).

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.3** titled "**FCS_CKM_EXT.4 Cryptographic Key Destruction (MDM_PP)**", and verified that it describes how the key destruction functionality is invoked for each supported platform. The invocation methods are The TOE securely destroys keys and Critical Security Parameters (CSPs) in both volatile and non-volatile memory via zeroization, in compliance with FIPS 140-3. For keys stored in platform key storage, such as certificate private keys, destruction is achieved by replacement when new certificates are received. Volatile memory keys, including TLS session keys, are overwritten with zeroes upon session or process termination. These operations are performed using cryptographic libraries OpenSSL and Bouncy Castle for the MDM Server, and OpenSSL for the MDM Agent which are included in the TOE's physical boundary and support key destruction functionality.

Verdict:
PASS.

4.1.3.3.2 FCS_CKM_EXT.4.1 AGD

None.

4.1.3.3.3 FCS_CKM_EXT.4.2 TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.3** titled "**FCS_CKM_EXT.4 Cryptographic Key Destruction (MDM_PP)**", and verified that it lists each type of plaintext key material and CSP, along with its origin and storage location. The TSS states as All keys within the TOE are securely destroyed as per the descriptions given in the following table titled "**Table 20: Keys and CSPs**".

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

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.3** titled "**FCS_CKM_EXT.4 Cryptographic Key Destruction (MDM_PP)**", and verified that it describes when each type of key material and CSP is no longer needed. The TSS states "**TLS private and public keys are destroyed when the associated X509v3 certificate is removed from the trust store of the TOE. All other keys are ephemeral, and are destroyed by the TOE when their associated session is terminated.".** A detailed information about the type of key and how they are destroyed is described in "**Table 20: Keys and CSP's**" of the TSS.

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).

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.3** titled "**FCS_CKM_EXT.4 Cryptographic Key Destruction (MDM_PP)**", and verified that it describes how the key releasing functionality is invoked for each supported platform. The TSS states as "**The MDM Agent invokes platform provided functionality for plaintext key destruction in volatile and non-volatile memory by overwriting with zeroes**. **The TOE meets all requirements specified in FIPS 140-3 for destruction of keys and Critical Security Parameters (CSPs). All keys within the TOE are securely destroyed as per the descriptions given in the table titled "Table 20: Keys and CSP's""**.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.3** titled "**FCS_CKM_EXT.4 Cryptographic Key Destruction (MDM_PP)**", and verified that for each type of key material and CSP, the type of clearing procedure that is performed is listed. The clearing procedures are as described in "**Table 20: Keys and CSPs**" of the TSS. The evaluator ensured that the TSS describes the clearing procedure for the keys stored in both

volatile and non-volatile memory are through zeroization and the type of key and CSP is as described in "**Table 20: Keys and CSPs**". The block erase commands are not part of the TOE selection, hence it would not be applicable.

Verdict:
PASS.

4.1.3.3.4 FCS_CKM_EXT.4.2 AGD

None.

4.1.3.4 FCS_COP.1(1) CRYPTOGRAPHIC OPERATION (CONFIDENTIALITY ALGORITHMS) *(MDM_PP)*

4.1.3.4.1 FCS_COP.1.1(1) TSS

If "invoke platform-provided functionality" is selected:

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).

Evaluator Findings:
The "invoke platform-provided functionality" is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.3.4.2 FCS_COP.1.1(1) AGD

None.

4.1.3.5 FCS_COP.1(2) CRYPTOGRAPHIC OPERATION (HASHING ALGORITHMS) *(MDM_PP)*

4.1.3.5.1 FCS_COP.1.1(2) TSS

If "invoke platform-provided functionality" is selected:

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this assurance activity is not applicable.

If "implement functionality" is selected:

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.5** titled “**FCS_COP.1(2) Cryptographic Operation (Hashing Algorithms) (MDM_PP)**”, and checked that the association of the hash function with other TSF cryptographic functions is documented. Upon investigation, the TSS states that **The TOE provides cryptographic hashing services as specified in FIPS Pub 180-5. Hashing is used for the following:** • **TLS communication,** • **Digital signing (MDM Server) and verification (MDM Agent) of Policies,** • **Digital signature verification by both TOE components as part of trusted update validation.** **The TSF performs cryptographic hashing with the following algorithms and message digest sizes:** • **SHA-256** • **SHA-384** • **SHA-512** • **and message digest sizes of 256 bits, 384 bits, and 512 bits respectively.** **The TOE uses a byte-oriented mode for hashing.** **Both the MDM Server and MDM Agent components of the TOE implement the hash functionality by including the cryptographic libraries used to generate hashes in the TOE’s physical boundary. The MDM Servers use the Ivanti MDM OpenSSL Component cryptography library for all TLS paths/channels except the initial TLS connection to the Syslog Server. For that connection, Ivanti MDM Bouncy Castle Cryptographic Library is used. The MDM Agents use the Ivanti MDM Android Client OpenSSL Component cryptography library.**

Verdict:
PASS.

4.1.3.5.2 FCS_COP.1.1(2) AGD

Note: An AGD activity was identified as part of the TSS Activity for FCS_COP.1.1(2).

If "implement functionality" is selected:

The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required hash sizes is present. The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF

only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs.edd

Evaluator Findings:
The evaluator reviewed the AGD, section **6.1.1.2** titled "**HASHING**", to ensure that any configuration required to configure the functionality for the required hash sizes is present. The AGD states "**The hash values used while establishing TLS communication, digital signing and certificate verification cannot be configured manually on the product. The hash algorithm used during these processes is determined at the time the certificate is created and issued by a Certificate Authority (CA). This algorithm is permanently embedded within the certificate in the signatureAlgorithm field**".

Verdict:
PASS.

4.1.3.6 FCS_COP.1(3) CRYPTOGRAPHIC OPERATION (SIGNATURE ALGORITHMS) *(MDM_PP)*

4.1.3.6.1 FCS_COP.1.1(3) TSS

If "invoke platform-provided functionality" is selected:

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).

Evaluator Findings:
The "invoke platform-provided functionality" is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.3.6.2 FCS_COP.1.1(3) AGD

None.

4.1.3.7 FCS_COP.1(4) CRYPTOGRAPHIC OPERATION (KEYED-HASH MESSAGE AUTHENTICATION) *(MDM_PP)*

4.1.3.7.1 FCS_COP.1.1(4) TSS

If "invoke platform-provided functionality" is selected:

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

If "implement functionality" is selected:

The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

Evaluator Findings:

The evaluator reviewed the TSS, section **6.3.7** titled “**FCS_COP.1(4) Cryptographic Operation (Keyed-Hash Message Authentication) (MDM_PP)**”and ensured that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used. The values specified are in accordance with FIPS PUB 198-1 and FIPS 180-4 mentioned in TSS is as below

Function	Hash Algorithm	Block Size	Key Length	Message Digest Size
HMAC-SHA-256	SHA-256	512 bits	256 bits	256 bits
HMAC-SHA-384	SHA-384	1024 bits	384 bits	384 bits

Verdict:
PASS.

4.1.3.7.2 FCS_COP.1.1(4) AGD

None.

4.1.3.8 FCS_IV_EXT.1 INITIALIZATION VECTOR GENERATION (PP_MDM_V4.0)

4.1.3.8.1 FCS_IV_EXT.1.1 TSS

If "invoke platform-provided functionality" is selected:

The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the IV generation is invoked for each mode 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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

If "implement functionality" is selected:

The evaluator shall examine the TSS to ensure that it details the encryption of user credentials, persistent secrets, and private keys and the generation of the IVs used for that encryption.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.9** titled "**FCS_IV_EXT.1 Initialization Vector Generation (MDM_PP)**", and ensured that it details the encryption of user credentials, persistent secrets, and private keys, and the generation of the IVs used for that encryption. The TSS states "**The TOE, using the library's RBG, generates AES-GCM IVs, that are non-repeating and 96-bits in length. The TOE uses IVs for the encryption/decryption algorithms used in TLS communication.** **Administrator and user credentials, certificates, and private keys are stored persistently in flat files with specific permission to limit access to only the necessary components or processes. These flat files are encrypted with AES-GCM as defined in NIST SP 800-38D. The TOE uses IV for the AES GCM algorithm.** **Refer to Section 6.3.12 FCS_STG_EXT.1 Cryptographic Key Storage (MDM_PP) and Section 6.3.13 FCS_STG_EXT.1(2) Cryptographic Key Storage (AGT_PPM) for the details of encryption of user credentials, persistent secrets, and private keys**.".

Verdict:
PASS.

4.1.3.8.2 FCS_IV_EXT.1.1 AGD

None.

4.1.3.9 FCS_HTTPS_EXT.1 HTTPS PROTOCOL *(MDM_PP)*

4.1.3.9.1 FCS_HTTPS_EXT.1.2 TSS

None.

4.1.3.9.2 FCS_HTTPS_EXT.1.2 AGD

None.

4.1.3.10 FCS_RBG_EXT.1/AGT EXTENDED: RANDOM BIT GENERATION *(MDM_PP)*

4.1.3.10.1 FCS_RBG_EXT.1.1/AGT TSS

If "invoke platform-provided functionality" is selected:

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not a selection; hence, this activity is not applicable.

Verdict:
PASS.

4.1.3.10.2 FCS_RBG_EXT.1.1/AGT AGD

None.

4.1.3.10.3 FCS_RBG_EXT.1.2/AGT TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.10** titled “**FCS_RBG_EXT.1/AGT Extended: Random Bit Generation (MDM_PP)**”, and verified that documentation is produced and the activities are performed in accordance with Appendix D: Entropy Documentation and Assessment and the "Clarification to the Entropy Documentation and Assessment Annex." The documentation and activities are **The MDM Agent provides an AES-256 CTR-DRBG (CAVP Cert #A6402) provided by the Ivanti MDM Android Client OpenSSL Component cryptographic library included in the TOE boundary.**

Verdict:
PASS.

4.1.3.10.4 FCS_RBG_EXT.1.2/AGT AGD

None.

4.1.3.11 FCS_RBG_EXT.1/SRV EXTENDED: RANDOM BIT GENERATION *(MDM_PP)*

4.1.3.11.1 FCS_RBG_EXT.1.1/SRV TSS

If "invoke platform-provided functionality" is selected:

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not a selection; hence, this activity is not applicable.

Verdict:
PASS.

4.1.3.11.2 FCS_RBG_EXT.1.1/SRV AGD

None.

4.1.3.11.3 FCS_RBG_EXT.1.2/SRV TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.11** titled “**FCS_RBG_EXT.1/SRV Extended: Random Bit Generation (MDM_PP)**”, and verified that documentation is produced and the activities are performed in accordance with Appendix D: Entropy Documentation and Assessment and the "Clarification to the Entropy Documentation and Assessment Annex. The documentation and activities are “**The MDM Server obtains entropy from two sources. It uses Intel RDRAND which is a hardware based entropy noise source and the Linux Kernel Random Number Generator (LKRNG) which is a platform-based RBG for random bit generation services. These are in accordance with NIST special Publications 800-90A.** **The MDM Server component of the TOE includes two cryptographic libraries: Ivanti MDM Bouncy Castle and Ivanti MDM OpenSSL Component. The OpenSSL library provides an AES-256 CTR_DRBG (CAVP #A6074) deterministic random bit generation. The Bouncy Castle library provides a SHA-256 Hash-DRBG (CAVP Cert #A6073)**”.

Verdict:
PASS.

4.1.3.11.4 FCS_RBG_EXT.1.2/SRV AGD

None.

4.1.3.12 FCS_STG_EXT.1 CRYPTOGRAPHIC KEY STORAGE *(MDM_PP)*

4.1.3.12.1 FCS_STG_EXT.1.1 TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.12** titled "**FCS_STG_EXT.1 Cryptographic Key Storage (MDM_PP) and FCS_STG_EXT.2 Encrypted Cryptographic Key Storage (MDM_PP)**", and verified 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 confirmed that the TSS lists the purpose it is used for and how it is stored. The items and their purposes and storage methods are all persistent secrets and private keys used by the MDM Server component of the TOE, detailing their purpose and storage method. Keys and certificates are stored in encrypted key stores, with access restricted via file permissions to authorized components and processes. The EPMM portal certificate key is the only exception stored on the Apache file system. The Bouncy Castle cryptographic library and platform-provided storage are used to protect these assets. Table 20: Keys and CSPs provides a complete listing of each key, its function, and its storage details.

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).

Evaluator Findings:
For MDM server, Persistent secrets and private keys are not manipulated by the TOE platform; hence, this activity is not applicable.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.12** titled **"FCS_STG_EXT.1 Cryptographic Key Storage (MDM_PP) and FCS_STG_EXT.2 Encrypted Cryptographic Key Storage (MDM_PP)**", and determined that for each item listed as being manipulated by the TOE, it is not written unencrypted to persistent memory, and the item is stored by the platform. The TSS states that "**The MDM Server stores its keys and certificates in files with specific permissions to limit access to only the necessary components or processes. Additionally, the files are encrypted with AES-GCM as defined in NIST SP 800-38D. Table**

20: Keys and CSPs can be referenced for a list of all keys resident on the MDM Server, what they are used for and how they are stored.".

Verdict:
PASS.

4.1.3.12.2 FCS_STG_EXT.1.1 AGD

None.

4.1.3.13 FCS_STG_EXT.1(2) CRYPTOGRAPHIC KEY STORAGE *(AGT_PPM)*

4.1.3.13.1 FCS_STG_EXT.1(2) CRYPTOGRAPHIC KEY STORAGE TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.13** titled "**FCS_STG_EXT.1(2) Cryptographic Key Storage (AGT_PPM)**", and verified that each persistent secret (credential, secret key) and private key needed to meet the requirements in the ST are listed. The TSS states that "**The MDM Agent stores mobile device user credentials, private keys and certificates persistently in the platform provided key storage (Android Key Store). This certificate is used to identify and authenticate the mobile device to the MDM Server on the TOE's internal TOE trusted channel (Line B Figure 1)".**

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.13** titled "**FCS_STG_EXT.1(2) Cryptographic Key Storage (AGT_PPM)**", and confirmed that for each persistent secret and private key, the TSS lists the purpose it is used for and how it is stored for each platform listed as supported in the ST. The purposes and storage methods are that each key and certificate is used to identify and authenticate the mobile device to the MDM Server via the TOE's internal trusted channel (Line B, Figure 1 of ST). For supported platforms specifically Android 13 and Android 14 the TSS specifies that these items are stored in the platform-provided Android Key Store.

The evaluator shall verify that the Agent calls a platform-provided API to store persistent secrets and private keys.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.13** titled "**FCS_STG_EXT.1(2) Cryptographic Key Storage (AGT_PPM)**", and verified that the Agent calls a platform-provided API to store persistent secrets and

private keys. The platform-provided APIs used are Samsung Boring SSL Android 1.7 or 1.8 for Android 13 and Android 14 respectively.

Verdict:
PASS.

4.1.3.13.2 FCS_STG_EXT.1(2) CRYPTOGRAPHIC KEY STORAGE AGD

None.

4.1.3.14 FCS_STG_EXT.2 ENCRYPTED CRYPTOGRAPHIC KEY STORAGE (PP_MDM_V4.0)

4.1.3.14.1 FCS_STG_EXT.2.1 TSS

The evaluator shall examine the TSS to ensure it describes in detail how user credentials, persistent secrets, and private keys are stored and encrypted.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.12** titled "**FCS_STG_EXT.1 Cryptographic Key Storage (MDM_PP) and FCS_STG_EXT.2 Encrypted Cryptographic Key Storage (MDM_PP)**", and verified that it describes in detail how user credentials, persistent secrets, and private keys are stored and encrypted. The TSS states that the MDM Server component of the TOE uses the Bouncy Castle cryptographic library in conjunction with platform-provided storage to securely store user credentials, persistent secrets, and private keys. Additionally AES-GCM is used for encrypting the files and details about storage of keys can be found in :**Table 20: Keys and CSPs**".

The evaluator shall review the TSS to determine that it makes a case that key material is not written unencrypted to persistent memory and that it identifies the mode of encryption.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.12** titled "**FCS_STG_EXT.1 Cryptographic Key Storage (MDM_PP) and FCS_STG_EXT.2 Encrypted Cryptographic Key Storage (MDM_PP)**", and verified that key material is not written unencrypted to persistent memory and that it identifies the mode of encryption. The TSS states that "**The MDM Server stores its keys and certificates in files with specific permissions to limit access to only the necessary components or processes. Additionally, the files are encrypted with AES-GCM as defined in NIST SP 800-38D. Table 20: Keys and CSPs can be referenced for a list of all keys resident on the MDM Server, what they are used for and how they are stored**".

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 encryption 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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.3.14.2 FCS_STG_EXT.2.1 AGD

None.

4.1.3.15 FCS_TLS_EXT.1 TLS PROTOCOL *(TLS_PKG)*

4.1.3.15.1 FCS_TLS_EXT.1.1 TSS

None.

4.1.3.15.2 FCS_TLS_EXT.1.1 AGD

The evaluator shall ensure that the selections indicated in the ST are consistent with selections in the dependent components.

Evaluator Findings:
The evaluator checked the AGD, and verified that the selections indicated in the ST are consistent with selections in the dependent components. This selection is satisfied by the SFR’s: CKM.1, CKM.2 and COP.1.1(1), COP.1.1(2), COP.1.1(3), COP.1.1(4), TLSC and TLSS and are consistent with the ciphers selected. This selection is also verified by the evaluator.

Verdict:
PASS.

4.1.3.16 FCS_TLSC_EXT.1 TLS CLIENT PROTOCOL *(TLS_PKG)*

4.1.3.16.1 FCS_TLSC_EXT.1.1 TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.15** titled "**FCS_TLSC_EXT.1 TLS Client Protocol (TLS_PKG)**",and verified that the cipher suites supported are specified. The supported cipher suites are mentioned as below • **TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,** • **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 evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.15** titled "**FCS_TLSC_EXT.1 TLS Client Protocol (TLS_PKG)**", and verified that the cipher suites specified include those listed for this component. The cipher suites listed for the components are mentioned as below • **TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,** • **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**

Verdict:
PASS.

4.1.3.16.2 FCS_TLSC_EXT.1.1 AGD

The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the product so that TLS conforms to the description in the TSS.

Evaluator Findings:
The evaluator checked the AGD, section **6.1.1** titled "**CONFIGURING CIPHER SUITES**", and verified that it contains instructions on configuring the product so that TLS conforms to the description in the TSS. The AGD states the administrators are instructed to configure TLS to use only TLSv1.2, rejecting older protocols. Cipher suites are managed via the System Manager, with outgoing suites set under Security > Advanced > Outgoing SSL Configuration, ensuring alignment with the TSS.

Verdict:
PASS.

4.1.3.16.3 FCS_TLSC_EXT.1.2 TSS

The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g. Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.15** titled "**FCS_TLSC_EXT.1 TLS Client Protocol (TLS_PKG)**", and verified that it describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g. Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject (Alternative Names) and whether IP addresses and wildcards are supported. The supported reference identifiers are The TSS specifies that reference identifiers for TOE components and trusted IT entities are administrator-configured via the web GUI and must be DNS Names, in accordance with RFC 6125 Section 6. During TLS certificate validation, the TOE client compares the configured reference identifier against the Subject Alternative Name (SAN) in the server's X.509 certificate; if the SAN is absent or mismatched, the connection is terminated. Wildcards are supported for user traffic but not for MDM Agent-to-Server communication, where exact CN matching is required. This validation process ensures secure identification of remote entities and aligns with RFC 6125 guidance. The following reference identifiers are not supported: The TOE doesn't support IP addresses as reference identifiers.

The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the product.

Evaluator Findings:
The evaluator reviewed section 6.3.15 of the ST titled "FCS_TLSC_EXT.1 TLS Client Protocol (TLS_PKG)" which states "The TOE does not implement certificate pinning.", satisfying the requirement to identify the manner in which certificate pinning is supported by the TOE.

Verdict:
PASS.

4.1.3.16.4 FCS_TLSC_EXT.1.2 AGD

The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.

Evaluator Findings:
The evaluator checked the AGD, section **6.2.1** titled "**FOR MDM SERVER**" and **6.2.2** titled "**FOR MDM AGENT**", and verified that it includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS. For the MDM Server, administrators set the DNS Name reference identifier via the Admin Portal under Settings > Data Export > Syslog Servers. For the MDM Agent, the reference identifier is established during initial registration by selecting "entering the company email or register with server URL" in the mobile application. These steps ensure that the configured DNS Name aligns with certificate validation requirements.

Verdict:
PASS.

4.1.3.16.5 FCS_TLSC_EXT.1.3 TSS

If the selection for authorizing override of invalid certificates is made, then the evaluator shall ensure that the TSS includes a description of how and when user or administrator authorization is obtained.

Evaluator Findings:
The selection for authorizing override of invalid certificates is not made; hence, this activity is not applicable.

The evaluator shall also ensure that the TSS describes any mechanism for storing such authorizations, such that future presentation of such otherwise-invalid certificates permits establishment of a trusted channel without user or administrator action.

Evaluator Findings:
The selection for authorizing override of invalid certificates is not made; hence, this activity is not applicable.

Verdict:
PASS.

4.1.3.16.6 FCS_TLSC_EXT.1.3 AGD

None.

4.1.3.17 FCS_TLSC_EXT.2 TLS CLIENT SUPPORT FOR MUTUAL AUTHENTICATION *(TLS_PKG)*

4.1.3.17.1 FCS_TLSC_EXT.2.1 TSS

The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.16** titled "**FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication (TLS_PKG)**", and verified that the description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication. The TSS states that The MDM Server component of the TOE supports mutual authentication when establishing secure communication with the Audit Server, which is the only Inter-TSF Trusted Channel (FPT_ITC.1(1)) implemented. Section 6.1.10 (FAU_STG_EXT.1) outlines three TLS Client channels that are capable of mutual authentication. Additionally, the MDM Agent component supports a mutual authentication TLS Client channel used for secure data transfer post-enrollment, leveraging the certificate received during the initial enrollment phase.

The evaluator shall also ensure that the TSS describes any factors beyond configuration that are necessary in order for the client to engage in mutual authentication using X.509v3 certificates.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.16** titled "**FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication (TLS_PKG)**", and verified that it describes any factors beyond configuration that are necessary in order for the client to engage in mutual authentication using X.509v3 certificates. **There are no factors beyond configuration that are necessary in order for the three client implementations to engage in mutual authentication using X.509 certificates.**

Verdict:
PASS.

4.1.3.17.2 FCS_TLSC_EXT.2.1 AGD

The evaluator shall ensure that the AGD guidance includes any instructions necessary to configure the TOE to perform mutual authentication.

Evaluator Findings:
The evaluator checked the AGD, section **6.3** titled "**MUTUAL AUTHENTICATION**", and verified that it includes any instructions necessary to configure the TOE to perform mutual authentication. The AGD states that the AGD guidance provides explicit instructions for configuring the TOE (EPMM) to perform mutual authentication. Section 6.3.1 outlines the steps to enable mutual authentication via the Admin Portal by navigating to Settings > Security > Certificate Authentication and selecting the appropriate checkbox. It also specifies that mutual authentication is enabled by default on new installations and cannot be disabled once activated.

The evaluator also shall verify that the AGD required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

Evaluator Findings:
The evaluator checked the AGD, section **6.3** titled "**MUTUAL AUTHENTICATION**", and verified that the guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication. The AGD states that Section 6.3.1 specifies that identity certificates are required for Android devices, and these are managed through the certificate enrollment settings of the MDM server. Section 6.3.2 further explains that MDM server generates certificate, or alternatively, administrators can import custom trusted root certificates via Services > Trusted Root certificates. This ensures that client-side certificates are properly configured and trusted for mutual authentication.

Verdict:
PASS.

4.1.3.18 FCS_TLSC_EXT.5 TLS CLIENT SUPPORT FOR SUPPORTED GROUPS EXTENSION *(TLS_PKG)*

4.1.3.18.1 FCS_TLSC_EXT.5.1 TSS

The evaluator shall verify that TSS describes the Supported Groups Extension.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.17** titled "**FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extensions (TLS_PKG)**", and verified that it describes the Supported Groups Extension. The TSS states that the Supported group extension in the Client are secp256r1, secp384r1, and secp521r1.

Verdict:
PASS.

4.1.3.18.2 FCS_TLSC_EXT.5.1 AGD

None.

4.1.3.19 FCS_TLSS_EXT.1 TLS SERVER PROTOCOL *(TLS_PKG)*

4.1.3.19.1 FCS_TLSS_EXT.1.1 TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.18** titled "**FCS_TLSS_EXT.1 TLS Server Protocol (TLS_PKG)**", and verified that the cipher suites supported are specified. The supported cipher suites are as mentioned as below • **TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,** • **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 evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.18** titled "**FCS_TLSS_EXT.1 TLS Server Protocol (TLS_PKG)**", and verified that the cipher suites specified include those listed for this component. The cipher suites listed for the components are as mentioned as below • **TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,** • **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**

Verdict:

PASS.

4.1.3.19.2 FCS_TLSS_EXT.1.1 AGD

The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

Evaluator Findings:
The evaluator checked the AGD, section **6.1.1** titled "**CONFIGURING CIPHER SUITES**", and verified that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS. The AGD states that administrators are instructed to configure TLS to use only TLSv1.2, rejecting older protocols. Cipher suites are managed via the System Manager, with incoming suites set under Security > Advanced > Incoming SSL Configuration, ensuring alignment with the TSS.

Verdict:

PASS.

4.1.3.19.3 FCS_TLSS_EXT.1.2 TSS

The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.18** titled "**FCS_TLSS_EXT.1 TLS Server Protocol (TLS_PKG)**", and verified that it describes the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2. The MDM Server enforces strict protocol compliance by supporting only TLS 1.2 for secure communications, in accordance with RFC 5246. Legacy SSL and earlier TLS versions are explicitly disabled, and any connection attempts using unsupported protocols are automatically rejected by the TOE.

Verdict:
PASS.

4.1.3.19.4 FCS_TLSS_EXT.1.2 AGD

The evaluator shall verify that the AGD guidance includes any configuration necessary to meet this requirement.

Evaluator Findings:
The evaluator checked the AGD, section **6.1.1** titled "**CONFIGURING CIPHER SUITES**", and verified that it contains instructions for any configurations necessary to meet this requirement. The AGD states that The TOE supports secure communications exclusively via TLS version 1.2, in strict accordance with RFC 5246. Operating in both TLS client and server roles, the TOE enforces protocol integrity by rejecting any connection attempts that utilize deprecated SSL or earlier TLS versions.

Verdict:
PASS.

4.1.3.19.5 FCS_TLSS_EXT.1.3 TSS

The evaluator shall verify that the TSS describes the key agreement parameters of the server's Key Exchange message.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.18** titled "**FCS_TLSS_EXT.1 TLS Server Protocol (TLS_PKG)**", and verified that it describes the key agreement parameters of the server's Key Exchange message. The AGD states "**The MDM Server generates key agreement parameters using NIST curves secp256r1, secp384r1, and secp521r1 when employing elliptic curve ciphers for the server's Key Exchange message and RSA key sizes of 2048, 3072, and 4096 bits for RSA key exchange. These key agreement parameters are dynamically generated during the TLS handshake process. The specific curve parameters correspond to the server certificate configured for client access. The selection of these curves ensures robust key agreement and cryptographic security for TLS sessions established between the MDM Server and clients, providing confidentiality and integrity for transmitted data**".

Verdict:
PASS.

4.1.3.19.6 FCS_TLSS_EXT.1.3 AGD

The evaluator shall verify that any configuration guidance necessary to meet the requirement must be contained in the AGD guidance.

Evaluator Findings:
The evaluator checked the AGD, section **6.1.1.1** titled "**KEY AGREEMENT PARAMETERS**", and verified that it contains instructions for any configurations necessary to meet this requirement. The AGD states that During the TLS handshake, the TOE dynamically generates key agreement parameters using NIST approved elliptic curves: secp256r1, secp384r1, and secp521r1. key sizes of 2048, 3072, and 4096 bits while using RSA schemes for this process. (RSA schemes meet FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Section 4. ECDSA Schemes meet FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Section 5). These parameters are consistently applied in both TLS client and server roles. As stated in the AGD "**The product does not require any additional configuration to utilize ECDSA key exchange algorithms, as these are implicitly applied when ECDHE cipher suites are selected"**.

Verdict:
PASS.

4.1.3.20 FCS_TLSS_EXT.2 TLS SERVER SUPPORT FOR MUTUAL AUTHENTICATION *(TLS_PKG)*

4.1.3.20.1 FCS_TLSS_EXT.2.1 TSS

None.

4.1.3.20.2 FCS_TLSS_EXT.2.1 AGD

None.

4.1.3.20.3 FCS_TLSS_EXT.2.2 TSS **[TD0770]**

The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.

Evaluator Findings:
The evaluator reviewed the TSS, **6.3.19** titled "**FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication (TLS_PKG)**", and verified that it describes the use of client-side certificates for TLS mutual authentication. The Client-side certificate of each mobile device is assigned to a unique UUID, which is linked to the serial number of its certificate. During the TLS handshake, when the device connects and presents its certificate, the UUID embedded within it is referenced to retrieve the corresponding serial number. This verification step ensures that the presented certificate matches the registered identifier, allowing only authenticated and authorized devices to establish secure TLS sessions with the MDM Server.

If error messages are provided prior to terminating a session, the evaluator shall ensure the error messages are described.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.19** titled "**FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication (TLS_PKG)**", and verified it describes error messages that are provided prior to terminating a session. Error messages detailing certificate validation failures are generated before the session is terminated, ensuring that administrators are promptly informed of authentication issues.

Verdict:

PASS.

4.1.3.20.4 FCS_TLSS_EXT.2.2 AGD

The evaluator shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

Evaluator Findings:
The evaluator checked the AGD, section **6.3** titled "**MUTUAL AUTHENTICATION**", and verified that it contains instructions for configuring the client-side certificates for TLS mutual authentication. The AGD states that the AGD guidance provides explicit instructions for configuring the TOE (EPMM) to perform mutual authentication. Section 6.3.1 outlines the steps to enable mutual authentication via the Admin Portal by navigating to Settings > Security > Certificate Authentication and selecting the appropriate checkbox. It also specifies that mutual authentication is enabled by default on new installations and cannot be disabled once activated.

The evaluator shall ensure that the AGD guidance includes instructions for configuring the server to require mutual authentication of clients using these certificates.

Evaluator Findings:
The evaluator checked the AGD, section **6.3** titled "**MUTUAL AUTHENTICATION**", and verified that it contains instructions for configuring the server to require mutual authentication of clients using these certificates. The AGD states that Section 6.3.1 specifies that identity certificates are required for Android devices, and these are managed through the certificate enrollment settings of the MDM server. Section 6.3.2 further explains that EPMM acts as a local Certificate Authority (CA) to generate these certificates, or alternatively, administrators can import custom trusted root certificates via Services > Trusted Root certificates. This ensures that client-side certificates are properly configured and trusted for mutual authentication.

Verdict:
PASS.

4.1.3.20.5 FCS_TLSS_EXT.2.3 TSS

If the product implements mutual authentication, the evaluator shall verify that the TSS describes how the DN and SAN in the certificate is compared to the expected identifier.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.3.19** titled "**FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication (TLS_PKG)**", and verified that it describes how the DN and SAN in the certificate is compared to the expected identifier. The TSS states that "**The comparison between the SAN (Subject Alternative Name) in the certificate and the expected identifier is crucial for certificate validation. The MDM Server and MDM Agent verify the certificate's SAN to ensure it matches the expected identifier, the fully qualified domain name (FQDN) of the remote network entity. Wildcards are not accepted. This comparison ensures that the certificate presented by the remote network entity is valid and authorized for communication, enhancing the security of TLS sessions between the MDM Server, MDM Agent, and external servers. Any discrepancies between the expected identifier and the information in the certificate are flagged, and TLS sessions are not established until validation is successful**. **In EPMM, wildcard certificate handling differs between the MDM server and MDM agent. The MDM Agent enforces strict identity validation by requiring an exact match of the Common Name (CN), as the CN is locally generated along with UUID of the agent and used for deterministic trust evaluation hence, wildcard certificates are not supported. On the other hand, the MDM server, when functioning as a client can accept wildcard certificates via the Subject Alternative Name (SAN) field, provided the TOE is designed only around Fully Qualified Domain Name (FQDN) resolution in SAN.** **For user portal communication, the TOE validates the SAN, otherName and rfc822Name".**

Verdict:
PASS.

4.1.3.20.6 FCS_TLSS_EXT.2.3 AGD

If the DN is not compared automatically to the domain name, IP address, username, or email address, the evaluator shall ensure that the AGD guidance includes configuration of the expected identifier or the directory server for the connection.

Evaluator Findings:
The evaluator checked the AGD, section **6.2** titled "**CONFIGURING REFERENCE IDENTIFIERS**", and verified that the DN is compared automatically.

Verdict:
PASS.

4.1.4 IDENTIFICATION AND AUTHENTICATION (FIA)

4.1.4.1 FIA_CLI_EXT.1 CLIENT AUTHORIZATION

4.1.4.1.1 FIA_CLI_EXT.1.1 TSS **[TD0754]**

The evaluator shall ensure that the TSS describes how the client is uniquely identified.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.4.1** titled "**FIA_CLI_EXT.1X.509 Unique Certificate (MDM_PP)**",and verified that it describes how the client is uniquely identified. The unique identification method is client identification through a certificate-based mechanism where the MDM Agent component maps the client identity certificate to a corresponding record in the registration database. During enrollment via a TLS trusted path, each mobile device is assigned a unique UUID linked to its certificate's serial number. Upon subsequent connections over the MA Inter-TSF trusted channel, the device presents its certificate, and the UUID is used to retrieve and validate the expected serial number, ensuring the presented certificate matches the certificate serial number and the client is uniquely and securely identified.

Verdict:
PASS.

4.1.4.1.2 FIA_CLI_EXT.1.1 AGD **[TD0754]**

None.

4.1.4.2 FIA_ENR_EXT.1 ENROLLMENT OF MOBILE DEVICE INTO MANAGEMENT (MDM_PP)

4.1.4.2.1 FIA_ENR_EXT.1.1 TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.4.2** titled "**FIA_ENR_EXT.1 Enrollment of Mobile Device into Management (MDM_PP)**", and verified that it describes the process of enrollment for each MDM Agent/platform listed as supported in the ST. The trusted path used for enrollment is TLS. The method of user authentication is For an MDM Agent to successfully enroll, the MDM Server Administrator needs to set up Device Registration parameters for the user. This setup requires assigning them a username/password or PIN, designating their role as a Mobile Device User, and linking the registration to their Android device's serial number. The method of authentication decision is remote authentication method. The actions performed on the MDM Server upon successful authentication are When the MD User downloads the MDM Agent app, they are prompted for a username/password or PIN based on the Device Registration settings configured by the MDM Server Administrator. The device initiates enrollment with a secure TLS connection, verifying only the MDM Server's certificate. After the connection is established, the user authenticates with their credentials. Enrollment is approved only if the device meets the MDM Server's criteria, such as device limits and restrictions. Once approved, the server's local CA generates a certificate, which is sent to the MDM Agent. The MDM Server then triggers a push notification to prompt the MDM Agent to complete enrollment. Upon receiving it, the MDM Agent establishes a new connection, requiring an X.509 certificate for mutual authentication. The device is assigned a UUID linked to its certificate serial number, ensuring secure identification during future connections.

Verdict:
PASS.

4.1.4.2.2 FIA_ENR_EXT.1.1 AGD

None.

4.1.4.2.3 FIA_ENR_EXT.1.2 TSS

The evaluator shall examine the TSS and verify that it implements a policy to limit the user's enrollment of devices.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.4.2** titled **"FIA_ENR_EXT.1 Enrollment of Mobile Device into Management (MDM_PP)**", and verified that it implements a policy to limit the user's enrollment of devices. The Enrollment is contingent on approval by the MDM Server, which verifies that the device complies with defined constraints such as not exceeding the configured per-user device limit and meeting any platform-specific restrictions. This limit is administratively set via the Admin Portal under Settings > Users & Devices > Registration.

Verdict:

PASS.

4.1.4.2.4 FIA_ENR_EXT.1.2 AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.3.2** titled "**CONFIGURE DEVICE LIMIT**" and verified that it describes the method(s) of restricting user enrollment. The AGD states that restricting of user device enrollment is by restricting the number of devices that he /she can register at a time. The evaluator also verified that the AGD instructs the administrator how to configure the restrictions. The AGD states that to configure device enrollment limits, access the Admin Portal and go to Settings > Users & Devices > Registration. Within the Per-User Device Limit section, you can set the maximum number of devices a user may enroll, using the available configuration function. The allowable range is between 1 and 50 devices, or the limit can be disabled entirely by selecting "none". Additionally, administrative guidance clearly outlines methods for restricting user enrollment and provides configuration instructions for two key parameters: passcode expiry and registration PIN code length. The Passcode Expiry setting defines the duration, in hours, for which a registration passcode remains valid, defaulting to 4 hours, with a configurable range between 1 and 4320 hours (up to 6 months). Additionally, the Registration PIN Code Length enforces a mandatory password entry during device registration, with a customizable length between 6 and 12 characters.

Verdict:
PASS.

4.1.4.3 FIA_ENR_EXT.2 AGENT ENROLLMENT OF MOBILE DEVICE INTO MANAGEMENT *(AGT_PPM)*

4.1.4.3.1 FIA_ENR_EXT.2 AGENT ENROLLMENT OF MOBILE DEVICE INTO MANAGEMENT TSS

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).

Evaluator Findings:
The evaluator reviewed the TSS, section **6.4.3** titled "**FIA_ENR_EXT.2 Agent Enrollment of Mobile Device into Management (AGT_PPM)**", and verified that it describes which types of reference identifiers are acceptable and how the identifier is specified. The types of reference identifiers and their specifications are in accordance with RFC 6125 Section 6, the reference identifier for MDM components within X.509 certificates are required to be a DNS Name. This constraint applies uniformly to inter-TOE communications between TOE components as well as interactions with non-TOE trusted IT entities. The TOE explicitly excludes support for IP address-based reference identifiers, ensuring consistent identity validation through domain-based naming conventions.

Verdict:
PASS.

4.1.4.3.2 FIA_ENR_EXT.2 AGENT ENROLLMENT OF MOBILE DEVICE INTO MANAGEMENT AGD

The evaluator shall examine the operational guidance to verify that it describes how to configure the 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.

Evaluator Findings:
The evaluator reviewed the AGD, section **6.2.2** titled "**FOR MDM AGENT**", and verified that the AGD describes how to configure the reference identifier of the MDM Server's certificate. The AGD states that the procedure for configuring the reference identifier of the MDM Server's certificate during the initial registration of the MDM Agent. Specifically, when prompted by the mobile application, users are instructed to bypass the default company email option and instead select "Or register with server URL," where they manually input the correct server URL. This URL functions as the reference identifier, typically represented as the DNS Name of the MDM Server.

Verdict:
PASS.

4.1.4.4 FIA_UAU.1 TIMING OF AUTHENTICATION *(MDM_PP)*

4.1.4.4.1 FIA_UAU.1.1 TSS

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).

Evaluator Findings:
The authentication functionality is not invoked by the platform-provided functionality; hence, this activity is not applicable.

Verdict:
PASS.

4.1.4.4.2 FIA_UAU.1.1 AGD

None.

4.1.4.4.3 FIA_UAU.1.2 TSS

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).

Evaluator Findings:
The authentication functionality is not invoked by the platform-provided functionality; hence, this activity is not applicable.

Verdict:
PASS.

4.1.4.4.4 FIA_UAU.1.2 AGD

None.

4.1.4.5 FIA_X509_EXT.1(1) X.509 CERTIFICATE VALIDATION *(MDM_PP)*

4.1.4.5.1 FIA_X509_EXT.1.1(1) TSS **[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.

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.4.5** titled “**FIA_X509_EXT.1(1) X.509 Certificate Validation (MDM_PP)**”, and verified that it describes where the check of validity of the certificates takes place and provides a description of the certificate path validation algorithm. The check of validity The TOE performs certificate validity checks within both its MDM Server and MDM Agent components. The

certificate validation includes "**Both components of the TOE validate X.509 certificates. Certificates are validated when they are loaded onto a TOE system and when they are received from a remote system during the TLS handshake**. **Certificate validation includes checking the validation path, basicConstraints, revocation, and extendedKeyUsage properties. Certificates must also chain to a trusted root that is configured by the administrator**" as quoted in TSS .

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.4.5** titled "**FIA_X509_EXT.1(1) X.509 Certificate Validation (MDM_PP)**", and verified that it describes when revocation checking is performed. The evaluator determined that MDM server and MDM agent's revocation checking behave the same way as stated in the TSS "**For MDM server**, **the revocation checking is performed when certificates are uploaded to the TOE also during the establishment of a TLS connection supporting both CRL and OCSP with OCSP takes precedence over CRL when EPMM acts as a server and only CRL is supported when EPMM acts as a client. For MDM agent, the revocation checking is performed when certificates are uploaded to the TOE also during the establishment of a TLS connection supporting both CRL and OCSP with OCSP takes precedence over CRL** ".

Verdict:
PASS.

4.1.4.5.2 FIA_X509_EXT.1.1(1) AGD **[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.

Evaluator Findings:
The "internal lookup of TOE-managed certificate status" is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.4.5.3 FIA_X509_EXT.1.2(1) TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.4.5.4 FIA_X509_EXT.1.2(1) AGD

None.

4.1.4.6 FIA_X509_EXT.2 X.509 CERTIFICATE AUTHENTICATION *(MDM_PP)*

4.1.4.6.1 FIA_X509_EXT.2.1 TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.4.6.2 FIA_X509_EXT.2.1 AGD

None.

4.1.4.6.3 FIA_X509_EXT.2.2 TSS

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.

Evaluator Findings:
"The evaluator reviewed the TSS, section 6.4.6 titled “FIA_X509_EXT.2 X.509 Certificate Authentication (MDM_PP)” and verified 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. The TSS states that “**The TOE maintains multiple X509 certificates namely Admin portal/System Manager certificate, mutual-authentication enrolment CA certificate, Audit server certificate and MDM Agent certificates. The TOE chooses an appropriate certificate based on the function that it performs. For example, when an administrator connects to the MDM server, the TOE uses Admin portal certificate for authentication**”. The evaluator reviewed AGD section **5** titled “**DEVICE ENROLLMENT**”, section **6.4** titled “**Configuring a client identity certificate for mutual**

authentication” and section **3.3.7** titled “**CONFIGURE SYSLOG**” provides a step-by-step instruction on how to configure the TOE.

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.4.6** titled “**FIA_X509_EXT.2 X.509 Certificate Authentication (MDM_PP)**”, and confirmed 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 behavior and distinctions are The TOE explicitly supports certificate validation for communications with Authorized IT Entities, notably the Audit Server via Syslog and HTTPS protocols, with MA being optional but supported upon server request. Internally, the MDM Server authenticates the MDM Agent’s certificate over the mutual authentication Internal TOE TSF Data Transfer channel, while MDM Clients validate the MDM Server’s certificate during both enrollment and subsequent data transfer operations. In all cases, failure to validate a certificate results in termination of the connection, ensuring integrity and trust. For all certificate authentication supported by the TOE, if the validation fails, the connection is not established.

Verdict:
PASS.

4.1.4.6.4 FIA_X509_EXT.2.2 AGD

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.

Evaluator Findings:
The requirement that the administrator is able to specify the default action is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.5 SECURITY MANAGEMENT (FMT)

4.1.5.1 FMT_MOF.1(1) MANAGEMENT OF FUNCTIONS BEHAVIOR *(MDM_PP)*

4.1.5.1.1 FMT_MOF.1.1(1) TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.1** titled **"FMT_MOF.1(1) Management of Functions Behaviour (MDM_PP)"**, and the user documents, and verified that they describe what security management functions are restricted to the administrator and what actions can be taken for each management function. Upon investigation the evaluator found that: **The security management functions within the MDM Server component of the TOE are exclusively restricted to authorized administrators accessing the server via the Admin Portal or the Service Manager GUI interfaces. This restriction is implemented through role-based access controls, wherein each function is assigned specific access rights.** Furthermore, the **Table 23: EPMM MDM Server Management Functions and Roles of ST** details the specific actions that 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.

Evaluator Findings:
The evaluator also reviewed the TSS, section **6.5.1** titled **"FMT_MOF.1(1) Management of Functions Behaviour (MDM_PP)"**, and confirmed that the security management functions are described as being restricted to authorized administrators and that the administrator is only able to take the actions documented in the user guidance. In addition, the evaluator performed testing to verify that management functions are not accessible to unauthorized users and that an administrator is limited to the set of actions as specified in the user documentation. The test results confirmed the claimed behavior.

The security management functions within the MDM Server component of the TOE are exclusively restricted to authorized administrators accessing the server via the Admin Portal or the Service Manager GUI interfaces. This restriction is implemented through role-based access controls, wherein each function is assigned specific access rights.
Furthermore, the **Table 23: EPMM MDM Server Management Functions and Roles of ST** details the specific actions that can be taken for each management function.

Verdict:
PASS.

4.1.5.1.2 FMT_MOF.1.1(1) AGD

None.

4.1.5.2 FMT_MOF.1(2) MANAGEMENT OF FUNCTIONS BEHAVIOR (ENROLLMENT) *(MDM_PP)*

4.1.5.2.1 FMT_MOF.1.1(2) TSS

The evaluator shall examine the TSS and verify that it describes how unauthorized users are prevented from enrolling in the MDM services.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.2** titled **"FMT_MOF.1(2) Management of functions Behaviour (Enrollment) (MDM_PP)"**, and verified that it describes how unauthorized users are prevented from enrolling in the MDM services. **MDM Services prevent unauthorized users from enrolling by requiring registration and authentication of MD Users. The MD Server Administrator configures Android devices on the MDM Server and issues a 6-12 digit PIN or password to the mobile device user. This PIN or password is necessary for enrolling the device, ensuring that only authorized users can access the MDM services.**

Verdict:
PASS.

4.1.5.2.2 FMT_MOF.1.1(2) AGD

None.

4.1.5.3 FMT_MOF.1(3) MANAGEMENT OF FUNCTIONS IN (MAS SERVER DOWNLOADS)

4.1.5.3.1 FMT_MOF.1.1(3)TSS

The evaluator shall examine the TSS to determine that all methods of initiating an application download or update push are specified.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.3** titled **"FMT_MOF.1(3) Management of Functions in (MAS Server Downloads) (MDM_PP)"**, and verified that it specifies all methods of initiating an application download or update push. **The Ivanti EPMM TOE supports Mobile Application Store (MAS) functionality. MAS hosts host applications for the enterprise and securely transmits applications to enrolled MDM Agents during its next check-in.** **Administrators can also configure the MDM Server with a list of available applications, enabling mobile device users to use the MDM Agent to query and install any available applications or updates applications from the list.** **An MD User can initiate an application update or an MDM Administrator or MDM Server can initiate an update.**

Verdict:
PASS.

4.1.5.3.2 FMT_MOF.1.1(3) AGD

The evaluator shall confirm that the operational guidance contains instructions on how to initiate an application download or update push.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.2.10** titled **"INSTALL APPLICATIONS"**, and verified that it contains instructions on how to initiate an application download or update push. The AGD states that the instructions for initiating an application download or update push within the Common Criteria evaluated configuration. It begins with the administrator accessing the Admin Portal to add and configure the desired application in the App Catalog, specifying metadata and deployment settings such as global configuration policies, catalog visibility, VPN settings, and Android enterprise installation. The app is then assigned to a label, which is applied to the target device through the portal. On the device, the user opens the Mobile@Work application, navigates to the "New Releases" section, and manually initiates installation by tapping the app.

Verdict:
PASS.

4.1.5.4 FMT_POL_EXT.1 TRUSTED POLICY UPDATE *(MDM_PP)*

4.1.5.4.1 FMT_POL_EXT.1.1 TSS **[TD0754]**

The evaluator shall verify that the ST describes how policies are signed, to include whether the private key used for signing is associated with an X509 certificate or public key, the method for distributing the policy verification material (a certificate or provisioned public key) to the agent, and the method for distinguishing whether a policy is appropriate for an agent.

Evaluator Findings:
The evaluator reviewed the ST, section **6.5.4** titled **"FMT_POL_EXT.1 Trusted Policy Update *(MDM_PP)* *and* FMT_POL_EXT.2 Agent Trusted Policy Update *(AGT_PPM)*"**, and verified that it describes how policies are signed. This includes whether the private key used for signing is associated with an X509 certificate or public key, the method for distributing the policy verification material to the agent, and the method for distinguishing whether a policy is appropriate for an agent. **The MDM Server provides digitally signed policies and policy updates to the MDM Agent. Policies and policy updates (including policy settings, configurations, and applications) are formatted using XML and the entire XML structure is hashed using SHA-256. The hash is then signed by the MDM Server using the configured portal certificate sent to the MDM Agent when the mutual authenticated channel was established during enrollment (FCS_COP.1(3)) defined algorithms: RSA-2048 or ECDSA-P384) and the signed hash is added to the policy structure prior to being sent to any agents via a secure TLS channel.** **When an MDM Agent checks in, the MDM Server returns a copy of the latest policies to the MDM Agent.** **When the MDM Agent receives the policy or policy update, it creates a SHA-256 hash of contents (less the signed hash). Then it verifies that the calculated hash matches with the hash was signed using the private key of the same MDM server portal certificate that was used to receive the policy or policy update. The MDM Agent will only install policies and policy updates if both the hash matches and the hash was signed by the expected certificate.**

If tokens are claimed in FMT_POL_EXT.1.3, the evaluator shall verify the ST describes how tokens are established and distributed to the agent.

Evaluator Findings:
Tokens are not claimed in FMT_POL_EXT.1.3; hence, this assurance activity is not applicable.

Verdict:

PASS.

4.1.5.4.2 FMT_POL_EXT.1.1 AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.2.4** titled **"INSTALL POLICIES"**, and verified that the AGD instructs administrators on configuring the Enterprise certificate to be used for signing policies or signing the policies before applying them. The evaluator verified the AGD, and it confirms the certificate used for signing policies is the same X.509 certificate employed for TLS communications. This certificate is validated by the MDM agent during the initial TLS handshake. Once validated, it is subsequently leveraged to sign policy bundles.

Verdict:

PASS.

4.1.5.5 FMT_POL_EXT.2 AGENT TRUSTED POLICY UPDATE *(AGT_PPM)*

4.1.5.5.1 FMT_POL_EXT.2 AGENT TRUSTED POLICY UPDATE TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.4** titled **"FMT_POL_EXT.1 Trusted Policy Update *(MDM_PP)* *and* FMT_POL_EXT.2 Agent Trusted Policy Update *(AGT_PPM)*"**, and verified that it 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 and unsuccessful cases are also described. The details are **When the MDM Agent receives the policy or policy update, it creates a SHA-256 hash of its contents (less the signed hash). Then it verifies that the calculated hash matches with the hash was signed using the private key of the same MDM server portal certificate that was used to receive the policy or policy update. The MDM Agent will only install policies and policy updates if both the hash matches and the hash was signed by the expected certificate. If the verification fails, the policy or policy update is discarded, an error notification is generated on the MDM Agent, and an alert is sent to the MDM Server, the agent will close the connection at the TCP layer, and will no longer send HTTP requests payloads to the server.** **This process is repeated during the next check-In since the policy current on the MDM Server has not been applied.** **The MDM Agent only accepts policies and policy updates that are digitally signed by the MDM Server. When the MDM Agent checks-in with the MDM Server, the MDM Server will send any policy or policy updates to the MDM Agent via the secure TLS channel if the policy or policy update has not yet been applied by the MDM Agent.** **Policies are sent when a MDM Agent checks-in with a MDM Servers. Check-ins occur the MDM Agent's Sync Interval expires. This parameter is in the Sync Policy. The TOE, in the Common Criteria Evaluated Configuration, is configured to check-in with the MDM Server at least every eight hours..**

The software components that are performing the processing must also be identified in the TSS and verified by the evaluators.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.4** titled **"FMT_POL_EXT.1 Trusted Policy Update *(MDM_PP)* *and* FMT_POL_EXT.2 Agent Trusted Policy Update *(AGT_PPM)*"**, and confirmed that the software components performing the processing are identified. The identified components are the **MDM Server** those signs and sends the policies, while the **MDM Agent** is the component that receives, processes, and verifies the policies. The evaluator verified that MDM server is the component that is performing the process by modifying the signature and the reaction of MDM agent when such modifications are made. The results of this activity can be found in the testing document for this SFR.

Verdict:
PASS.

4.1.5.5.2 FMT_POL_EXT.2 AGENT TRUSTED POLICY UPDATE AGD

None.

4.1.5.6 FMT_SMF.1(1) SPECIFICATION OF MANAGEMENT FUNCTIONS (SERVER CONFIGURATION OF AGENT) *(MDM_PP)*

4.1.5.6.1 FMT_SMF.1.1(1) TSS **[TD0479]**

The evaluator shall examine the TSS to ensure that it describes each management function claimed.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.5** titled **"FMT_SMF.1(1) Specification of Management Functions (Server configuration of Agent) *(MDM_PP)"***, and verified that it describes each management function claimed. **The MDM Server component of the TOE supports the following commands sent to the MDM Agents.** • **Transition to the locked state.** **Locking an Android device is performed by an administrator as performing the Action > Lock on a specific device from the Admin Portal. The command is sent immediately to the device. Upon receipt, the MDM Agent will lock the Android screen will appear and the login credentials will be required** • **Full wipe of protected data.** **Instructing an Android device to wipe (erase) protected data is performed by an administrator as performing the Action > Wipe on a specific device from the Admin Portal. The command is sent immediately to the device. Wiping a device is the same as unenrolling a device. Once wipe is initiated the device factory resets there by making the device to the preconfigured mode i.e. unenrolled.** • **Unenroll from management.** **Instructing an Android device to wipe (erase) protected data is performed by an administrator as performing the Action > Wipe on a specific device from the Admin Portal. The command is sent immediately to the device. Wiping a device is the same as unenrolling a device. Once wipe is initiated the device factory resets there by making the device to the preconfigured mode i.e. unenrolled.** • **Install policies.** **See below for policy updates.** • **Query status** ○ **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.

The above information is always present and can be viewed in the GUI screens when an administrator logs onto the MDM Server because it is information exchanged during a device check-in. The information is as current as the value of the Sync Interval parameter in the device's Sync Policy. An administrator may select to view current information by selecting the Action > Force Device Check-In for a specific device from the Admin Portal. This will force a device check-in and update the above parameters as well as all new policies.

- Import X.509v3 certificates into the Trust Anchor Database.
- Install applications.

  The MDM Server supports MAS functionality. MAS functionality enables the MDM Server to download applications and then make those applications available to a group of MDM Agents to download to their device.

- Update system software.

  Refer to Section 6.6.6 FPT_TUD_EXT.1 Trusted Update (MDM_PP) for a description of Update system software.

- Remove applications and Enterprise applications.

The MDM Server supports the following policies on the mobile devices that enable the following parameters to be configured.

- Password Policy:
  - minimum password length
  - minimum password complexity
  - maximum password lifetime

  The password parameters are updated by an administrator modifying fields in the Security Policy. The Administrator guidance provides detail information about the specific TSF Data being modified. When an Administrator modifies the Password Policy, it modifies the local copy of the MDM Servers policy. The new parameters will be sent to the MDM Agent when the MDM Agent next checks-in. Alternatively, if the Administrator wants to apply the policy immediately, the Administrator can select the Action > Force Device Check-In. This sends a message to the MDM Agent and forces the MDM Agent to update the new policies.

- Session locking Policy:
  - Screen-lock enabled/disabled.
  - Screen lock timeout
  - Configure the number of authentication failures.

  The session locking parameters are updated by an administrator modifying fields in the Security Policy and sending the new Policy update to the MDM Agent.

- Wireless networks (SSIDs) to which the MD may connect.

  The Wireless parameters are updated by an administrator modifying fields in the Wireless Network Policy. The Administrator guidance provides detail information about the specific TSF Data being modified. When an Administrator modifies the Wireless Network Policy, it

modifies the local copy of the MDM Servers policy. The new parameters will be sent to the MDM Agent when the MDM Agent next checks-in. Alternatively, if the Administrator wants to apply the policy immediately, the Administrator can select the Action > Force Device Check-In. This sends a message to the MDM Agent and forces the MDM Agent to update the new policies.

- **Security policy for each wireless network:**
  - **specify the CA(s) from which the MD will accept WLAN authentication server certificate(s)**
  - **ability to specify security type**
  - **ability to specify authentication protocol**
  - **specify the client credentials to be used for authentication**

  The Wireless parameters are updated by an administrator modifying fields in the Wireless Network Policy. The Administrator guidance provides detail information about the specific TSF Data being modified. When an Administrator modifies the Wireless Network Policy, it modifies the local copy of the MDM Servers policy. The new parameters will be sent to the MDM Agent when the MDM Agent next checks-in. Alternatively, if the Administrator wants to apply the policy immediately, the Administrator can select the Action > Force Device Check-In. This sends a message to the MDM Agent and forces the MDM Agent to update the new policies.
- **Application installation Policy by**
  - **specifying authorized application repository(s)**

  The application repository is specified by an administrator modifying a Services parameter and enabling/disabling Google Play or Apps@Work or both. Upon selecting Apply, the new configuration will be sent to the MDM Agent.
- **Enable/disable policy for the camera and microphone.**

  The camera and microphone parameters are updated by an administrator modifying fields in the Lockdown Policy. The Administrator guidance provides detail information about the specific TSF Data being modified. When an Administrator modifies the Lockdown Policy, it modifies the local copy of the MDM Servers policy. The new parameters will be sent to the MDM Agent when the MDM Agent next checks-in. Alternatively, if the Administrator wants to apply the policy immediately, the Administrator can select the Action > Force Device Check-In. This sends a message to the MDM Agent and forces the MDM Agent to update the new policies.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.5** titled **"FMT_SMF.1(1) Specification of Management Functions (Server configuration of Agent)** ***(MDM_PP)"***, and verified that it identifies the management functions implemented for each supported MDM Agent/platform. The TSS specifies that the TOE only

supports the Android platform, and the list of management functions provided applies to Android. The identified management functions for each platform are

The MDM Server component of the TOE supports the following commands sent to the MDM Agents.

- **Transition to the locked state.**
- **Full wipe of protected data.**
- **Unenroll from management.**
- **Install policies.**
- **Query status**
    - **Query connectivity status.**
    - **Query the current version of the MD firmware/software.**
    - **Query the current version of the hardware model of the device.**
    - **Query the current version of installed mobile applications.**
- **Import X.509v3 certificates into the Trust Anchor Database.**
- **Install applications.**
- **Update system software.**
- **Remove applications and Enterprise applications.**
- **Password Policy:**
    - **minimum password length**
    - **minimum password complexity**
    - **maximum password lifetime**
- **Session locking Policy:**
    - **Screen-lock enabled/disabled.**
    - **Screen lock timeout**
    - **Configure the number of authentication failures.**
- **Wireless networks (SSIDs) to which the MD may connect.**
- **Security policy for each wireless network:**
    - **specify the CA(s) from which the MD will accept WLAN authentication server certificate(s)**
    - **ability to specify security type**
    - **ability to specify authentication protocol**
    - **specify the client credentials to be used for authentication**
- **Application installation Policy by**
    - **specifying authorized application repository(s)**
- **Enable/disable policy for the camera and microphone.**

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.5** titled **"FMT_SMF.1(1) Specification of Management Functions (Server configuration of Agent) *(MDM_PP)"***, and verified that any differences between management functions and policies for each supported MDM Agent/platform are clearly indicated. The TSS consistently refers to "MDM Agents" and "Android devices" as a single, uniform entity. It does not mention other platforms or variations in functionality.

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.

Evaluator Findings:
The evaluator determined whether the Mobile Device has been evaluated. Upon confirmation of evaluation, the evaluator reviewed the TSS, section **6.5.5.2** titled **" Mobile Device Fundamentals FMT_SMF.1 Mappings"**, and verified that it clearly identifies which management functions match the selections and assignments of the evaluated Mobile Device and which management functions were not evaluated. The identified management functions that match the evaluated Mobile Device are the functions not evaluated are as specified in the identified TSS section.

Verdict:
PASS.

4.1.5.6.2 FMT_SMF.1.1(1) AGD

None.

4.1.5.7 FMT_SMF.1(2) SPECIFICATION OF MANAGEMENT FUNCTIONS (SERVER CONFIGURATION OF SERVER) *(MDM_PP)*

4.1.5.7.1 FMT_SMF.1.1(2) TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.6** titled **"FMT_SMF.1(2) Specification of Management Functions (Server configuration of Server) *(MDM_PP)"***, and verified that it describes each management function listed. For function c.4, the evaluator confirmed that the TSS explicitly states

the TOE does not collect any privacy-sensitive information from enrolled mobile devices, and therefore this function is not claimed in the ST.

The management functions listed for the MDM Server component of the TOE include:

- **Configuring devices which are identified by a unique device ID in the form of a serial number**

Managing device registration process: This function involves controlling the number of devices a user can enroll.

The MDM Servers do not collect any privacy-sensitive information from the enrolled mobile devices. (The ST does not claim function c.4.)

Verdict:

PASS.

4.1.5.7.2 FMT_SMF.1.1(2) AGD

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

Evaluator Findings:
The evaluator reviewed the AGD, section **5** titled **"DEVICE ENROLLMENT"** and section **7.3.2** titled "**CONFIGURE DEVICE LIMIT**", and verified that it includes detailed instructions of what options are available and how to configure each management functional capability listed. The AGD states that it provides detailed instructions on how an administrator can configure each management functional capability listed. The section clearly outlines the available configuration options for each function and includes step-by-step guidance to ensure proper implementation

Verdict:

PASS.

4.1.5.8 FMT_SMF.1(3) SPECIFICATION OF MANAGEMENT FUNCTIONS (MAS SERVER) *(MDM_PP)*

4.1.5.8.1 FMT_SMF.1.1(3) TSS

The evaluator shall examine the TSS to ensure that it describes each management function listed.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.7** titled **"FMT_SMF.1(3) Specification of Management Functions (MAS Server) *(MDM_PP)*"**, and verified that it describes each management function listed. **The MDM Server, which also functions as a MAS (Mobile Application Store) server and supports the configuration of application groups by the admin. Additionally, it enables the downloading of applications for deployment.**

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.7** titled **"FMT_SMF.1(3) Specification of Management Functions (MAS Server) *(MDM_PP)*"**, and determined if the MAS Server creates its own groups or relies on the groups specified by the MDM Server. The TSS states that **The MDM Server, which also functions as a MAS (Mobile Application Store) server and supports the configuration of application groups by the admin. Additionally, it enables the downloading of applications for deployment.** **Groups are implemented by the MDM Server as "Labels". The TOE has both default and custom labels that Administrator can create with specific criteria. Some of the default groups/labels include "All-Smartphones", "Android" and "Employee-Owned" etc. Admin can create custom labels both static and dynamic. The admin must manually assign these labels.**

Verdict:
PASS.

4.1.5.8.2 FMT_SMF.1.1(3) AGD

The evaluator shall confirm that the operational guidance contains instructions on how to create and define user groups and how to specify which applications are accessible by which group.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.3.4** titled **"CONFIGURING THE APPLICATION ACCESS GROUPS"**, and verified that it contains instructions on how to create and define user groups and how to specify which applications are accessible by which group. The AGD states that the process for creating and defining user groups through the use of Labels in Ivanti EPMM. Labels serve as dynamic tags that group users or devices based on specific criteria, enabling targeted access control. The documentation provides step-by-step instructions for creating labels either manually or via advanced search and applying them to devices. This mechanism allows administrators to manage application access by associating specific labels with app configurations, thereby controlling which applications are accessible to 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.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.3.4** titled **"CONFIGURING THE APPLICATION ACCESS GROUPS"**, and verified that it includes detailed instructions of what options are available and how to configure each management functional capability listed. The AGD states that "**Application Access Groups refer to collections of users or devices that are granted controlled access to specific mobile applications or app configurations. Ivanti EPMM allows configuring the application access groups using Labels. Note: Label is a dynamic tag used to group devices based on specific criteria, enabling targeted policy enforcement, app distribution, or configuration management."** it also provides detailed instructions for configuring each management functional capability listed in the requirement.

Specifically, the documentation outlines the use of “Labels” within Ivanti EPMM as a core mechanism for grouping users, devices, or applications based on defined criteria. This enables targeted policy enforcement and application access control. A step-by-step procedure for both creating labels instructions for configuring the management functions related to label creation and application. It details the available options such as label type, and character restrictions for label naming. Additionally, it explains how to apply labels to devices via the Admin Portal, ensuring that administrators can confidently navigate and configure these capabilities to enforce policies and manage access effectively. The evaluator reviewed the AGD, section **7.2.10** titled “**INSTALL APPLICATIONS**“ and verified that it includes a detailed instructions on how an application is made available to the mobile device user and how they can download it at the mobile end once made available.

Verdict:

PASS.

4.1.5.9 FMT_SMF_EXT.4 SPECIFICATION OF MANAGEMENT FUNCTIONS (AGT_PPM)

4.1.5.9.1 FMT_SMF_EXT.4 SPECIFICATION OF MANAGEMENT FUNCTIONS TSS

The evaluator shall verify that any assigned functions are described in the TSS and that these functions are documented as supported by the platform.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.8** titled **“FMT_SMF_EXT.4 Specification of Management Functions *(AGT_PPM)*”**, and verified that the assigned functions are described and documented as supported by the platform. The assigned functions are as mentioned below **The MDM Agent interacts with the Android platform to perform the following tasks:** • **Import the certificates to be used to authenticate the Servers it communicates with over TLS channels. The only server the MDM Agents communicate with is the MDM Server component of the TOE.**

The evaluator shall examine the TSS to verify that any differences between management functions and policies for each supported mobile device are listed.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.8** titled **“FMT_SMF_EXT.4 Specification of Management Functions *(AGT_PPM)*”**, and verified that any differences between management functions and policies for each supported mobile device are listed. The TSS consistently refers to "MDM Agents" and "Android devices" as a single, uniform entity. It does not mention other platforms or variations in functionality.

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

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.8** titled **"FMT_SMF_EXT.4 Specification of Management Functions *(AGT_PPM)*"**, and verified that it describes the methods in which the MDM Agent can be enrolled. Based on the TSS, • **Configure an Android device to enable the device to join an EPMM System. Specifically, usernames/passwords are assigned to a device.** **The method of use to support enrollment and configuration is the TOE supports two trusted channels that each support HTTP over TLS from the MDM Agents to the MDM Server. The MDM Server acts as the TLS Server and the MDM Agents act as the TLS Client. This remote enrollment and configuration is the only supported method.**

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).

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.8** titled **"FMT_SMF_EXT.4 Specification of Management Functions *(AGT_PPM)*"**, and verified that it makes clear if the MDM Agent supports multiple interfaces for enrollment and configuration. The TSS states that **The method of use to support enrollment and configuration is the TOE supports two trusted channels that each support HTTP over TLS from the MDM Agents to the MDM Server. The MDM Server acts as the TLS Server and the MDM Agents act as the TLS Client. This remote enrollment and configuration is the only supported method.**

Verdict:

PASS.

4.1.5.9.2 FMT_SMF_EXT.4 SPECIFICATION OF MANAGEMENT FUNCTIONS AGD

The evaluator shall verify the AGD guidance includes detailed instructions for configuring each function in this requirement.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.2.9** titled **"IMPORT X.509V3 CERTIFICATES"**, and verified that the AGD includes detailed instructions for configuring each function in this requirement. The AGD states that a detailed, step-by-step instructions for configuring each management functional capability listed in the requirement. The documentation clearly guides the administrator through the navigation process within the interface and outlines how to access, modify, and apply configuration settings for the function. The evaluator reviewed the AGD, section **5** titled "**DEVICE ENROLLMENT**", section 7.3.7 titled "**CONFIGURING THE SYNC INTERVAL FOR A DEVICE**" section **7.2.3** titled "**UNENROLL FROM MANAGEMENT**" provides a details step-by-step instructions on how an administrator can enroll a device with the TOE, instructions on how to setup periodic reachability between the server and the MDM agent. The AGD also confirms that mobile device users are not provided with the functionality to enroll form enterprise.

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **1.4** titled **"COMMON CRITERIA EVALUATED PLATFORMS"**, and verified, by consulting documentation for the claimed mobile device platforms, that the configurable functions listed for this Agent are supported by the platforms. The evaluator reviewed the AGD sections **7.2.9** titled "**IMPORT X.509V3 CERTIFICATES", section 5** titled "**DEVICE ENROLLMENT**" and section **7.3.7** titled "**CONFIGURING THE SYNC INTERVAL FOR A DEVICE**" provides a detailed a detailed, step-by-step instructions to perform the configurable functions listed for this agent and these functions 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.

Evaluator Findings:
The MDM Agent does not support multiple interfaces for configuration; hence, this activity is not applicable.

Verdict:
PASS.

4.1.5.10 FMT_SMR.1(1) SECURITY MANAGEMENT ROLES *(MDM_PP)*

4.1.5.10.1 FMT_SMR.1.2(1) TSS

The evaluator shall examine the TSS to verify that it describes the administrator role and the powers granted to and limitations of the role.

Evaluator Findings:

The evaluator reviewed the TSS, section **6.5.9** titled **"FMT_SMR.1(1) Security Management Roles *(MDM_PP)*"**, and verified that it describes the administrator role and the powers granted to and limitations of the role. The TSS states that **The Common Criteria evaluated MDM Server includes four default user roles that are configured at TOE installation and configuration of the TOE into the CC evaluated configuration. The TOE, in the CC evaluated configuration, will have one administrator assigned to each of the following default roles.**

- **Administrator - This is the initial administrator that has full control of all functions. Administrators have comprehensive control over the MDM Server functions and have no restrictions except that Administrators cannot get into the root shell.**
- **MAS Administrator - Application Access Group – This role is assigned to administrators for the MAS functionality role required by FMT_SMR.1(2). It's a limited admin role that provides authority to manage the MAS functionality from the server. The TOE allows "Labels" to be defined and apps to be assigned. Devices can also be assigned to labels and the a**
- **sociated services to restrict or allow access to corresponding apps. A MAS Administrator has permission to define and assign Labels.**
- **MD User - This is a use**
- **added in the Admin Portal, but not assigned any administrative roles.**
- **MAS MD User - Application Access Group – This is the agent side of the FMT_SMR.1(2) requirement. This role is assigned to MD Users but add permission for MAS functionality.**

Enrolled Mobile Device – This is a mobile device that has been enrolled by the MDM Server when a MD User or a MAS MD User is successfully enrolled.

The permissions are grouped into the following categories.

- **Device Management – includes permissions that enable an administrator to view and manage devices which include adding and deleting devices, applying and removing device labels, wiping devices, locking and unlocking devices, force a device check-in, push profiles to a device, and retire a device.**
- **Privacy Control - includes permissions that enable an administrator to view apps in device detail, View device IP and MAC addresses, and locate devices.**
- **Label Management - includes permissions that enable an administrator to view and manage labels assigned to devices.**
- **User Management - includes permissions that enable an administrator to view and manage users and user attributes.**
- **App Management - includes permissions that enable an administrator to view apps resident on the EPMM Server Host, distribute apps, import and edit apps, and manage app licenses.**
- **Configuration Management – view and manage configuration, apply and remove configuration label, and delete configuration.**
- **Policy Management - includes permissions that enable an administrator to view and manage policies, apply and remove the policy label, delete policies, and manage the compliance action policy.**

- **Compliance Policy Management – view and modify the Security Policy which contains compliance information.**
- **Logs and Events Management - includes permissions that enable an administrator to view audit logs, manage and view events, and manage and view certificates.**
- **Admin Management - includes permissions that enable an administrator to manage administrators ad device spaces.**

Verdict:
PASS.

4.1.5.10.2 FMT_SMR.1.2(1) AGD

The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE and which interfaces are supported.

Evaluator Findings:

The evaluator reviewed the AGD, section **2.2** titled **"ACCESSING ADMINISTRATIVE INTERFACES"**, and verified that it contains instructions for administering the TOE and which interfaces are supported. The AGD states that clear and complete instructions for administering the Target of Evaluation (TOE), including the creation and management of two distinct local administrative users, one for the Admin Portal and one for the System Manager. Although these users initially share the same credentials during setup, the guidance explicitly states that any changes made to the roles or passwords of the Admin Portal user do not affect the System Manager user, highlighting their independent management. Additionally, the document specifies the supported administrative interfaces, detailing access URLs for each the Admin Portal is accessed via https://<fully-qualified-domain-name>:443/mifs, and the System Manager via https://<fully-qualified-domain-name>:8443/mics. These interface descriptions are accompanied by configuration instructions, ensuring administrators can securely and effectively manage the TOE through both portals.

Verdict:
PASS.

4.1.5.11 FMT_SMR.1(2) SECURITY MANAGEMENT ROLES (MAS SERVER)

4.1.5.11.1 FMT_SMR.1.2(2) TSS

The evaluator shall examine the TSS to verify that it describes the administrator role and the powers granted to and limitations of the role.

Evaluator Findings:

The evaluator reviewed the TSS, section **6.5.10** titled **"FMT_SMR.1(1) Security Management Roles *(MDM_PP)*"**, and verified that it describes the administrator role and the powers granted to and limitations of the role. The TSS states that **The MAS Server functionality of the MDM Server is configured and maintained by a unique user role, the Application Access Group.**

- **Application Access Group – The TOE allows "Labels" to be defined and apps to be assigned. Devices can also be assigned to labels and the associated services to restrict or allow access to corresponding apps. An Application Access Group user has the permission to define and assign Labels.**
- **Enrolled Mobile Device – This is a mobile device that has been enrolled by the MDM Server when a MD User or a MAS MD User is successfully enrolled.**

Refer to Section 6.5.9 for a description of how the above roles are implemented in the TSF.

Verdict:
PASS.

4.1.5.11.2 FMT_SMR.1.2(2) AGD

The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE and which interfaces are supported.

Evaluator Findings:
The evaluator reviewed the AGD, section **2.2** titled **"ACCESSING ADMINISTRATIVE INTERFACES"**, and verified that it contains instructions for administering the TOE and which interfaces are supported. The AGD states that clear and complete instructions for administering the Target of Evaluation (TOE), including the creation and management of two distinct local administrative users, one for the Admin Portal and one for the System Manager. Although these users initially share the same credentials during setup, the guidance explicitly states that any changes made to the roles or passwords of the Admin Portal user do not affect the System Manager user, highlighting their independent management. Additionally, the document specifies the supported administrative interfaces, detailing access URLs for each the Admin Portal is accessed via https://<fully-qualified-domain-name>:443/mifs, and the System Manager via https://<fully-qualified-domain-name>:8443/mics. These interface descriptions are accompanied by configuration instructions, ensuring administrators can securely and effectively manage the TOE through both portals.

Verdict:
PASS.

4.1.5.12 FMT_UNR_EXT.1 USER UNENROLLMENT PREVENTION (AGT_PPM)

4.1.5.12.1 FMT_UNR_EXT.1 USER UNENROLLMENT PREVENTION TSS

The evaluator shall ensure that the TSS describes the mechanism used to prevent users from unenrolling or the remediation actions applied when unenrolled.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.5.11** titled **"FMT_UNR_EXT.1 User Unenrollment Prevention *(AGT_PPM)*"**, and verified that it describes the mechanism used to prevent users from unenrolling or the remediation actions applied when unenrolled. The mechanisms and remediation actions are **Because the TOE's use case is [USE CASE 1] an enterprise-owned device for general purpose enterprise use. The TSF does not allow a Mobile Device User (MAS MD User or MD User role) to unenroll from the enterprise. The TSF does not provide the functionality that enables a user to unenroll. Additionally, Mobile Device Users enrolled and using the EPMM System are in the evaluated configuration supporting non-Android administrators. This non-administrator privilege status means the MAS MD User and MD User cannot uninstall or remove the MDM Agent Ivanti application.**

Verdict:

PASS.

4.1.5.12.2 FMT_UNR_EXT.1 USER UNENROLLMENT PREVENTION AGD

The evaluator shall ensure that the administrative guidance instructs administrators in configuring the unenrollment prevention in each available configuration interface.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.3.3** titled **"CONFIGURE UNENROLLING DEVICES"**, and verified that the AGD instructs administrators in configuring the unenrollment prevention in each available configuration interface. The AGD states that Mobile Device Users (MAS MD User or MD User role) cannot unenroll from the enterprise, as the system does not provide any functionality for user-initiated unenrollment. In the evaluated configuration, these users are designated as non-administrators on Android devices, which prevents them from uninstalling or removing the Ivanti MDM Agent application. This built-in restriction ensures that unenrollment prevention is enforced by default across all configuration interfaces.

If any configuration allows users to unenroll, the guidance also describes the actions that unenroll the Agent.

Evaluator Findings:
The evaluator reviewed the AGD, section **7.2.3** titled **"UNENROLL FROM MANAGEMENT"**, and verified that the AGD describes the actions that unenroll the Agent if any configuration allows users to unenroll. The AGD states that the mobile device user is not provided with the functionality of unenrollment from the enterprise.

Verdict:

PASS.

4.1.6 PROTECTION OF THE TSF (FPT)

4.1.6.1 FPT_API_EXT.1 USE OF SUPPORTED SERVICES AND APIS *(MDM_PP)*

4.1.6.1.1 FPT_API_EXT.1.1 TSS

The evaluator shall verify that the TSS lists the platform APIs used by the MDM software.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.1** titled **"FPT_API_EXT.1 Use of Supported Services and API's (MDM_PP)"**, and verified that it lists the platform APIs used by the MDM software. The APIs used by the MDM software are as follows: The MDM Server, running on Oracle Linux, utilizes well-documented platform APIs including the Linux File System API for file read/write and permission management, the Linux Network API for packet transmission and interface configuration, standard POSIX time functions for time retrieval, and the RPM Package Manager for software integrity verification and updates. The MDM Agent, operating on Android 13 and 14, relies on Android's java.io package for file system access, java.time.chrono for time-of-day functions, the NetworkStack module for network operations, and the Android Keystore API for secure certificate storage.

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.

Evaluator Findings:
The evaluator compared the list with the supported APIs available through sources such as developer accounts and platform developer groups, and ensured that all APIs listed in the TSS are supported. The MDM Server, running on Oracle Linux, utilizes well-documented platform APIs including the Linux File System API for file read/write and permission management, the Linux Network API for packet transmission and interface configuration, standard POSIX time functions for time retrieval, and the RPM Package Manager for software integrity verification and updates all of which are supported and documented through Oracle's official developer resources such as the Oracle Linux Developer Portal and Oracle Linux Documentation. The MDM Agent, operating on Android 13 and 14, relies on Android's java.io package for file system access, java.time.chrono for time-of-day functions, the NetworkStack module for network operations, and the Android Keystore API for secure certificate storage all confirmed through the Android Developer Portal, which provides comprehensive documentation and API references and this ensures the API's listed are supported.

Verdict:

PASS.

4.1.6.1.2 FPT_API_EXT.1.1 AGD

None.

4.1.6.2 FPT_ITT.1(2) INTERNAL TOE TSF DATA TRANSFER (MDM AGENT) *(MDM_PP)*

4.1.6.2.1 FPT_ITT.1.1(2) TSS

The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.2** titled **"FPT_ITT.1(2) Internal TOE TSF Data Transfer (MDM Agent) (MDM_PP)"**, and verified that it describes the methods and protocols used to protect distributed TOE components. The methods and protocols used to protect distributed TOE components, is HTTP over TLS v1.2 with mutual authentication. TLS is implemented using CAVP-certified OpenSSL Crypto Libraries, and HTTP functionality is provided by Apache 2.4.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.2** titled **"FPT_ITT.1(2) Internal TOE TSF Data Transfer (MDM Agent) (MDM_PP)**"and verified 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. The protocol is HTTPS over TLS 1.2 compliant with RFC 2818 are consistent with those specified in the security functional requirements

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).

Evaluator Findings:
The "invoke platform-provided functionality" is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.6.2.2 FPT_ITT.1.1(2) AGD

The evaluator shall confirm that the operational guidance contains instructions for establishing the communication paths for each supported method.

Evaluator Findings:
The evaluator reviewed the AGD, section **3.3.5** titled **"IMPORT CERTIFICATES" 3.3.6 tilted "CONFIGURE TLS"** and section **4.2.2.3** titled **"IMPORT CERTIFICATES ON MDM AGENTS"**, and verified that it contains instructions for establishing the communication paths for each supported method. The AGD states that the procedures for importing and creating X.509 certificates on the EPMM server, which are essential

for enabling mutual authentication and securing communication channels. Additionally, it details the configuration of Transport Layer Security (TLS), ensuring encrypted communication between the MDM server, MDM agent, and external IT entities. The documentation also includes device-specific instructions for importing CA certificates on supported Android devices, particularly in scenarios involving self-signed certificates. These steps demonstrate that the guidance supports secure and reliable communication setup across all relevant components.

Verdict:
PASS.

4.1.6.3 FPT_LIB_EXT.1/AGT USE OF THIRD PARTY LIBRARIES *(MDM_PP)*

4.1.6.3.1 FPT_LIB_EXT.1.1/AGT TSS

The evaluator shall verify that the TSS lists the libraries used by the MDM software or the public libraries used by the cloud MDM services. The evaluator shall verify that libraries found to be packaged with or employed by the MDM software are limited to those in the assignment.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.3** titled **"FPT_LIB_EXT.1/AGT Use of Third Party Libraries (MDM_PP)"**, and verified that it lists the libraries used by the MDM software or the public libraries used by the cloud MDM services. The libraries used by the MDM software are as follows: Ivanti MDM Android Client OpenSSL Component Cryptographic Library, v2.2.1 and OTS libraries as listed in appendix B of the security target. The evaluator then verified the selection in the SFR and the content in the TSS are a match and are the ones employed by MDM Agent software.

Verdict:
PASS.

4.1.6.3.2 FPT_LIB_EXT.1.1/AGT AGD

None.

4.1.6.4 FPT_LIB_EXT.1/SRV USE OF THIRD PARTY LIBRARIES *(MDM_PP)*

4.1.6.4.1 FPT_LIB_EXT.1.1/SRV TSS

The evaluator shall verify that the TSS lists the libraries used by the MDM software or the public libraries used by the cloud MDM services. The evaluator shall verify that libraries found to be packaged with or employed by the MDM software are limited to those in the assignment.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.4** titled **"FPT_LIB_EXT.1/SRV Use of Third Party Libraries (MDM_PP)"**, and verified that it lists the libraries used by the MDM software. The libraries used by the MDM software are as follows: Apache 2.4 , Ivanti MDM Bouncy Castle v1.0.2.4, Ivanti MDM OpenSSL Component v1.1.1g, and the OTS libraries enumerated in Appendix B. The evaluator then verified the selection in the SFR and the content in the TSS are a match and are the ones employed by 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.

Evaluator Findings:
The evaluator also verified that the libraries found to be packaged with or employed by the MDM software are limited to those in the assignment. The libraries found to be packaged with or employed by the MDM software are strictly limited to those listed in the assignment. This includes the specific versions of Apache 2.4, Ivanti MDM Bouncy Castle v1.0.2.4, Ivanti MDM OpenSSL Component v1.1.1g, and the OTS libraries referenced in Appendix B. The evaluator then verified the selection in the SFR and the content in the TSS are a match and are the ones employed by MDM software.

Verdict:
PASS.

4.1.6.4.2 FPT_LIB_EXT.1.1/SRV AGD

None.

4.1.6.5 FPT_TST_EXT.1 FUNCTIONALITY TESTING *(MDM_PP)*

4.1.6.5.1 FPT_TST_EXT.1.2 TSS

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).

Evaluator Findings:
The "invoke platform-provided functionality" is not selected; hence, this activity is not applicable.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.5** titled **"FPT_TST_EXT.1 Functionality Testing (MDM_PP)"**, and ensured that it details the self-tests that are run by the TSF on start-up. "**Once the RPM integrity tests pass and the system has entered into the operational state, the TSF performs self-tests on all approved cryptographic algorithms. This includes both the Ivanti MDM Bouncy Castle crypto library and the Ivanti MDM OpenSSL Component crypto library. The tests run, include but are not limited to, the algorithms identified in Table 26: CAVP Algorithm Certificate References. During these tests, services are not available, and data output is inhibited during the conditional self-tests. Specifically, the tests run by each cryptographic library are:** • **Known Answer Tests – These tests verify the correct operation of cryptographic algorithms before they can be used in FIPS mode. They execute predefined tests vect** ○ **rs and compare outputs against known correct values.** • **Power-on Self-Test (POST) – This test runs automatically when and ensures that the crypto library is ready for FIPS compliance.** • **Hash Function Self-tests – These tests support multiple SHA variants using a unified testing framework.** • **Algorithm Size Calculation Functions – Each algorithm implements size calculation functions to determine memory requirements. These tests help ensure that the algorithms function correctly and are not susceptible to attacks.** The evaluator also ensured that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. The TSS states "**If any of these tests fail, an error message with be displayed on the console, the system will transition to the Error State, and halts processing. If this occurs, the Administrator should contact Ivanti support. Otherwise, if the tests are successful, the system transitions into the operational state. Successful completion of the self-tests and transitioning into the operational state demonstrates 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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.5** titled **"FPT_TST_EXT.1 Functionality Testing (MDM_PP)"**, and ensured that it describes how to verify the integrity of stored TSF executable code when it is loaded for execution. Integrity is verified the TOE utilizes the RPM Package Manager to verify the integrity of its executable files during system start-up. This platform-provided functionality checks each installed RPM, including those containing TSF code, against a secure database using standard Linux integrity verification mechanisms.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.5** titled **"FPT_TST_EXT.1 Functionality Testing (MDM_PP)"**, and ensured 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 TSS states that the RPM Package Manager's integrity checking functions invoked during system start-up are sufficient to guarantee the integrity of the stored TSF executable code. This is because RPM maintains a secure database of installed packages and performs cryptographic hash checks against each file. These checks are performed automatically and consistently at boot, and any discrepancies are logged and trigger a restart of the MDM Server without administrator intervention. Additionally, the use of FIPS-certified cryptographic modules (MDM OpenSSL) reinforces the reliability of the self-tests, ensuring that cryptographic operations are functioning correctly.

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.6.5** titled **"FPT_TST_EXT.1 Functionality Testing (MDM_PP)"**, and ensured 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. Successful cases When integrity is verified through RPM checks, the MDM Server proceeds with normal operation. In unsuccessful cases if the verification fails, the system logs the error to the console and initiates an automatic restart sequence. This behavior occurs without administrator intervention. And this RPM checks are sufficient to guarantee that no executable code is compromised.

Verdict:
PASS.

4.1.6.5.2 FPT_TST_EXT.1.2 AGD

None.

4.1.6.6 FPT_TUD_EXT.1 TRUSTED UPDATE *(MDM_PP)*

4.1.6.6.1 FPT_TUD_EXT.1.1 TSS

None.

4.1.6.6.2 FPT_TUD_EXT.1.1 AGD

The evaluator shall ensure that the administrator guidance includes instructions for determining the current version of the TOE.

Evaluator Findings:
The evaluator reviewed the AGD, section **8.2.1** titled **"DETERMINING THE CURRENT VERSION OF THE EPMM SERVER"** and section **8.3.1** titled **"DETERMINING THE CURRENT RUNNING VERSION OF THE ANDROID CLIENT"**, and verified that it includes instructions for determining the current version of the TOE. The current version of the MDM server is determined by the administrator to access the Admin Portal and navigate the Dashboard interface. From the top-right user menu, selecting the "About" option will trigger an information window displaying system metadata. Within this window, the value listed under the "Release Version" tag reflects the active build version of the product. It is critical that this version matches the designated Build Name specified in Table 3 of the MDM Server TOE Component Build of guidance documentation. Once confirmed, the administrator may close the window by clicking "OK.". And for the MDM client "**The current software of the EPMM client can be viewed via the Admin Portal of the EPMM server. For more information refer section Query Current Version of the Mobile Device Firmware/Software." For the MDM Agent, The current software of the MDM Agent component of the TOE (Ivanti MDM Agent for Android) installed by viewing the app's About Screen from their Android device.**

Verdict:
PASS.

4.1.6.6.1 FPT_TUD_EXT.1.2 TSS

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).

Evaluator Findings:
The "invoke platform-provided functionality" is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.6.6.2 FPT_TUD_EXT.1.2 AGD

None.

4.1.6.6.3 FPT_TUD_EXT.1.3 TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

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.

Evaluator Findings:
The evaluator reviewed the TSS section **6.6.6** titled “**FPT_TUD_EXT.1 Trusted Update (MDM_PP)**”, and verified that it describes the standards by which the updates are digitally signed and how the signature verification process is implemented. The standards and verification process are as stated in the TSS “**RPMs are signed with an Ivanti signing key which is an RSA 2048-bit SHA-256 key. When a package is installed, the signature on the package is validated by the RPM tool. The RPM includes digests of the files within the RPM. These digests are stored in a database on the system during package install. During TOE upgrade, the contents of each file are verified against the stored digests.** **To download, the Administrator selects the System Manager’s Download option. Once downloaded, the Administrator must select Validate, to validate the digital signature. The Validate option will validate the software using the RPM tool (Operational Environment). A message will be displayed reporting the success or failure of the signature.** **If the validation was successful, the Administrator must select the “Stage for Install” option to apply the MDM Server software. Only successfully signed packages will be allowed to be installed. Once the “Stage for Install” option is selected, the system is rebooted**”. For MDM agent, “**All releases are packaged in APK (Android Package) format and are signed with an Ivanti software certificate.** **To install the application, the Android User issues the adb install <downloaded .apk file> (Android Debug Bridge) command. Success or failure of the operation will be displayed on the screen. The upgrade is successfully performed only if the digital signature of the new build matches the digital signature of the current build”.**

Verdict:
PASS.

4.1.6.6.4 FPT_TUD_EXT.1.3 AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **8** titled **“UPDATING THE EPMM SYSTEM SOFTWARE”**, and verified 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. The current version of the TSF software the administrator to access the Admin Portal and navigate to the Dashboard interface. From the top-right user menu, selecting the "About" option will trigger an information window displaying system

metadata. Within this window, the value listed under the "Release Version" tag reflects the active build version of the product. Initiating updates involves upgrading of Ivanti EPMM software using System Manager, begin by logging into the platform and navigating to Maintenance > Software Updates. Under the Software repository configuration group, enter the credentials provided by Ivanti Support and apply the changes. Next, click Check Updates to view available updates, select the desired update, and either Download Now for later installation or proceed directly. Once downloaded, refresh the screen and verify the update status has changed to Downloaded. Before proceeding, validating the database is highly recommended, ensuring the structure and data integrity align with upgrade requirements. If validation fails, contact Ivanti Support before attempting an upgrade. When ready, click Stage for Install, ensuring the update is properly prepared. After staging, the status should reflect Reboot to Install, at which point the system must be rebooted for installation to proceed. If the update status does not show Reboot to Install, a reboot will not trigger the installation. Finally, navigate to Maintenance > Reboot to finalize the upgrade process. Checking the integrity of updates prior to installation occurs Before installing updates, the integrity of the system is verified through the Validate function in System Manager. This process ensures that the database structure aligns with upgrade requirements by either validating the schema alone or copying the database to a temporary instance for a more thorough check, which requires stopping EPMM services. Depending on the database size, validation with data can take up to four hours. The validation status can indicate whether the process is running, failed, or successful. If validation fails, the upgrade should not proceed, and Ivanti Support must be contacted for assistance. The evaluator reviewed the AGD, Section **8.3.1** titled "**DETERMINING THE CURRENT RUNNING VERSION OF THE ANDROID CLIENT**" and section **8.3.3** titled "**Updating the MDM Agent Software**" which states that the current version of MDM Agent component of the TOE can be verified by the Android User through the app's About screen on their device. Upon receiving an upgrade notification from Ivanti, the EPMM Server Administrator will inform the Android User of the availability of a new MDM Agent software build. If the Android User confirms that an upgrade is required, the installation is initiated by executing the following command "adb install <apk file> ". The result of the installation's success or failure is displayed on the console. The upgrade is considered successful only if the digital signature of the newly installed APK matches that of the currently deployed version, ensuring software integrity and continuity of the signing identity.

Verdict:
PASS.

4.1.7 TRUSTED PATH/CHANNELS (FTP)

4.1.7.1 FTP_ITC.1(1) INTER-TSF TRUSTED CHANNEL (AUTHORIZED IT ENTITIES) *(MDM_PP)*

4.1.7.1.1 FTP_ITC.1.1(1) TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.1.2 FTP_ITC.1.1(1) AGD

None.

4.1.7.1.3 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).

Evaluator Findings:
The “invoke platform-provided functionality” is not a selection; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.1.4 FTP_ITC.1.2(1) AGD

None.

4.1.7.1.5 FTP_ITC.1.3(1) TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.7.1** titled **“FTP_ITC.1(1) Inter-TSF Trusted Channel (Authorized IT Entities) (MDM_PP)”**, and verified that it indicates the methods of communication with authorized IT entities and how those communications are protected. The TSS states that the TOE communicates with one authorized IT entity, an Audit Server using mutually authenticated TLS 1.2 (RFC 5246). The MDM Server acts as a TLS Client, initiating secure connections via the Bouncy Castle and OpenSSL Crypto Libraries.

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.1.6 FTP_ITC.1.3(1) AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **6.2.1** titled **“FOR MDM SERVER”**, and verified that it contains instructions for configuring the communication channel between the MDM Server and authorized IT entities for each supported method. The AGD states that for external IT entity communication on the MDM Server, administrators use the Admin Portal by navigating to Settings > Data Export > Syslog Servers. Within this interface, selecting the Add utility allows entry of the server’s DNS name as the reference identifier. Additional parameters including protocol type, administrative state, template selection, and severity level can be configured to establish and tailor the Syslog Server connection.

Verdict:
PASS.

4.1.7.2 FTP_ITC_EXT.1 TRUSTED CHANNEL *(MDM_PP)*

4.1.7.2.1 FTP_ITC_EXT.1.1 TSS

The evaluator shall ensure that the TSS contains whether the MDM Server communication channel is internal or external to the TOE.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.7.2** titled **“FTP_ITC_EXT.1 Trusted Channel (MDM_PP)”**, and ensured that it specifies whether the MDM Server communication channel is internal or external to the TOE. The communication channel between the two TOE components is considered internal communication to the TOE.

Verdict:
PASS.

4.1.7.2.2 FTP_ITC_EXT.1.1 AGD

None.

4.1.7.3 FTP_TRP.1(1) TRUSTED PATH (FOR REMOTE ADMINISTRATION) *(MDM_PP)*

4.1.7.3.1 FTP_TRP.1.1(1) TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.3.2 FTP_TRP.1.1(1) AGD

None.

4.1.7.3.3 FTP_TRP.1.2(1) TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.3.4 FTP_TRP.1.2(1) AGD

None.

4.1.7.3.5 FTP_TRP.1.3(1) TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.1.7.2** titled **“FTP_TRP.1(1) Trusted Path (for Remote Administration) (MDM_PP)”**, and verified that it indicates the methods of remote TOE administration and how those communications are protected. The evaluator also confirmed 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. The methods, protections, and protocols are The MDM Server provides two distinct graphical user interfaces for remote administration: the Admin Portal and the System Manager. Network communication for the Admin Portal is established over standard HTTPS using TCP port 443, while the System Manager interface is accessible via TCP port 8443.

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.3.6 FTP_TRP.1.3(1) AGD

The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method.

Evaluator Findings:
The evaluator reviewed the AGD, section **2.2** titled **“ACCESSING ADMINISTRATIVE INTERFACES”**, and verified that it contains instructions for establishing the remote administrative sessions for each supported method. The AGD states that During the initial configuration phase, a local user accounts are provisioned to support remote administration for the Admin Portal and for the System Manager. They function independently; changes made to roles or authentication parameters (such as password updates) for the Admin Portal user do not affect the System Manager user. Remote access to both administrative interfaces is facilitated via HTTPS at the following endpoint: https://<fully-qualified-domain-name>:8443/mics and https://<fully-qualified-domain-name>:443/mifs, with each interface logically isolated to ensure secure and distinct management operations.

Verdict:
PASS.

4.1.7.4 FTP_TRP.1(2) TRUSTED PATH (FOR ENROLLMENT) *(MDM_PP)*

4.1.7.4.1 FTP_TRP.1.1(2) TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.4.2 FTP_TRP.1.1(2) AGD

None.

4.1.7.4.3 FTP_TRP.1.2(2) TSS

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.4.4 FTP_TRP.1.2(2) AGD

None.

4.1.7.4.5 FTP_TRP.1.3(2) TSS

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.

Evaluator Findings:
The evaluator reviewed the TSS, section **6.7.4** titled **“FTP_TRP.1(2) Trusted Path (for Enrollment) (MDM_PP)”**, and verified that it indicates the methods of remote enrollment and how those communications are protected. The evaluator also confirmed 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. The methods, protections, and protocols are a secure HTTPS interface that utilizes TLS encryption, allowing mobile devices to connect remotely through the MDM Agent. This interface ensures encrypted communication between the server and mobile devices, safeguarding sensitive data from unauthorized access and interception. By leveraging TLS security protocols, the MDM Server provides a reliable method for mobile device users to authenticate, enroll, and receive policy updates without compromising security.

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).

Evaluator Findings:
The “invoke platform-provided functionality” is not selected; hence, this activity is not applicable.

Verdict:
PASS.

4.1.7.4.6 FTP_TRP.1.3(2) AGD

The evaluator shall confirm that the operational guidance contains instructions for establishing the enrollment sessions for each supported method.

Evaluator Findings:
The evaluator reviewed the AGD, section **3.3.5** titled **“IMPORT CERTIFICATES” 3.3.6** tilted **“CONFIGURE TLS”** and **4.2.2.3** titled **“IMPORT CERTIFICATES ON MDM AGENTS”**, and verified that it contains instructions for establishing the enrollment sessions for each supported method. The AGD states that the procedures for importing and creating X.509 certificates on the EPMM server are essential for enabling mutual authentication and securing communication channels. Additionally, it details the configuration of Transport Layer Security (TLS), ensuring encrypted communication between the MDM server, MDM agent, and external IT entities. The documentation also includes device-specific instructions for importing CA certificates on supported Android devices, particularly in scenarios involving self-signed certificates.

Verdict:
PASS.

5 SECURITY ASSURANCE REQUIREMENTS

5.1 ADV: DEVELOPMENT

5.1.1 BASIC FUNCTIONAL SPECIFICATION (ADV_FSP.1)

5.1.1.1 EVALUATION ACTIVITY

There are no specific evaluation activities associated with these SARs, except ensuring the information is provided. The functional specification documentation is provided to support the evaluation activities described in Section 5 and the relevant appendices, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other evaluation activities being performed; if the evaluator is unable to perform an activity because there is insufficient interface information, then an adequate functional specification has not been provided.

Evaluator Findings:
The evaluator reviewed the functional specification documentation and verified that it is provided to support the evaluation activities described in Section 5 and the relevant appendices, as well as the activities described for AGD, ATE, and AVA SARs. The evaluator implicitly assessed the content of the functional specification information through the performance of other evaluation activities. No specific findings were noted, as the adequacy of the functional specification is determined by the ability to perform the other evaluation activities.

Verdict:
PASS.

5.2 AGD: GUIDANCE DOCUMENTATION

5.2.1 OPERATIONAL USER GUIDANCE (AGD_OPE.1)

5.2.1.1 EVALUATION ACTIVITY

Some of the contents of the operational guidance will be verified by the evaluation activities in Sections 4.2, 4.3, and 4.4 and evaluation of the TOE according to the CEM. The following additional information is also required:

Evaluator Findings:
The evaluator confirmed that some of the contents of the AGD will be verified by the evaluation activities in Section 5.1 Security Functional Requirements and evaluation of the OS 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.

Evaluator Findings:
The evaluator reviewed the AGD, section **6** titled "**CONFIGURING TRANSPORT LAYER SECURITY (TLS)**", and verified that the AGD contains instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It also provides 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:

Evaluator Findings:
The evaluator reviewed the AGD, section **8** titled "**UPDATING THE EPMM SYSTEM SOFTWARE**", and verified that the AGD describes the process for verifying updates to the TOE by verifying a digital signature – this may be done by the TOE or the underlying platform.

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).

Evaluator Findings:
The evaluator reviewed the AGD, section **8.2.2** titled "**UPDATING THE EPMM SERVER MANUALLY**", and verified that the AGD describes the instructions for obtaining the updates. The updates can be applied to the Ivanti EPMM update when in a Common Criteria evaluated configuration, by log into the EPMM System Manager and navigate to Maintenance > Software Updates. Scroll to the "Manually Update OS, Device and other EPMM data" section and click the provided link to access the latest update. Download the update file (zip format) and place it in the designated directory accessible to the TOE.

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **8.2.2** titled "**UPDATING THE EPMM SERVER MANUALLY**", and verified that it includes 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 updates procedure to initiate the update process for the Ivanti EPMM server, log into the System Manager and navigate to Maintenance > Software Updates. After selecting the desired update, click "Download Now" or "Stage for Install" to begin. During download, the system automatically verifies the update using a digital signature or hash to ensure authenticity and integrity. Once the update status

changes to “Reboot to Install,” reboot the system to apply the update. To confirm success, return to the Software Updates section post-reboot and verify that the current version reflects the intended update. If the status remains unchanged or validation fails, the update was unsuccessful and Ivanti Support should be contacted.

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.

Evaluator Findings:

The evaluator reviewed the TSS, section **1.5** titled “**Product Functionality not Included in the Scope of the Evaluation**”, and verified if there is any functionality excluded from the TOE. Section “Product Functionality not Included in the Scope of the Evaluation” in the ST states that The Common Criteria evaluated configuration for Ivanti EPMM excludes several components and functionalities. Ivanti Sentry is not part of the evaluation, nor are the CLI interface and the Self-Service User Portal. The configuration does not support Non-admin and Guest roles on Android devices, and it does not allow connection to the MDM Server via the SSH protocol, which typically provides command-line access. The evaluated EPMM Agent is the Federal Build, available only through Ivanti personnel, and differs from the commercial mobile@work agent found on Google Play and the Ivanti website. Additionally, Trusted Front End (TFE) functionality, SAML authentication, and ActiveSync are not enabled in the evaluated configuration are the ones excluded from the evaluation and only the default is supported. The evaluator then reviewed the AGD, section **1.5** titled “**PRODUCT FUNCTIONALITY NOT INCLUDED IN THE SCOPE OF THE EVALUATION**”, and verified that the evaluated configuration of Ivanti EPMM excludes several components and functionalities to maintain a controlled and secure assessment scope. Ivanti Sentry is not included in the evaluation, and neither the Command Line Interface (CLI) nor the Self-Service User Portal are part of the evaluated setup. The configuration does not support Non-admin and Guest roles on Android devices, and connectivity to the MDM Server via the SSH protocol—commonly used for CLI access—is explicitly excluded. The EPMM Agent assessed is the Federal Build, which is distributed exclusively through Ivanti personnel and differs from the commercial mobile@work agent available on Google Play and the Ivanti website. Additionally, Trusted Front End (TFE) functionality, Security Assertion Markup Language (SAML) integration, and ActiveSync support are all omitted from the Common Criteria (CC) evaluated configuration has been exclude from the Common Criteria evaluation and that only the default has been supported. The evaluator then reviewed the AGD, section **1.6** titled “**Security Functionalities covered in the CC evaluation**” which states the functionalities that are coved by the evaluation activity.

Verdict:
PASS.

5.2.2 PREPARATIVE PROCEDURES (AGD_PRE.1)

5.2.2.1 EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator reviewed the AGD, section **1.4** titled "**COMMON CRITERIA EVALUATED PLATFORMS**", and verified that the configuration requirements of the TOE platform are installed. The evaluator also reviewed the AGD, section **1.4** titled "**COMMON CRITERIA EVALUATED PLATFORMS**", and verified that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

Verdict:
PASS.

5.3 ALC: LIFE-CYCLE SUPPORT

5.3.1 LABELING OF THE TOE (ALC_CMC.1)

5.3.1.1 EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator reviewed the ST section **1.1** titled "**Security Target and TOE Reference**", to verify that the ST 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.

Evaluator Findings:
The evaluator reviewed the AGD section **1.4** titled "**COMMON CRITERIA EVALUATED PLATFORMS**", and ensured 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.

Evaluator Findings:
The evaluator reviewed the vendor's website and confirmed that the TOE is not advertised in website and ensured that it the information in the ST is sufficient to distinguish the product.

Verdict:
PASS.

5.3.2 TOE CM COVERAGE (ALC_CMS.1)

5.3.2.1 EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator confirmed 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 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.

Evaluator Findings:
The evaluator ensured 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.

Verdict:
PASS.

5.4 ATE: TESTS

5.4.1 INDEPENDENT TESTING - CONFORMANCE (ATE_IND.1)

5.4.1.1 EVALUATION ACTIVITY

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of this PP's Evaluation

Activities. While it is not necessary to have one test case per test listed in an evaluation activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered.

Evaluator Findings:
The evaluator prepared a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the [CEM] and the body of this PP's evaluation activities.

The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

Evaluator Findings:
The test plan covers all of the testing actions contained in the [CEM] and the body of this PP's evaluation activities.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).

Evaluator Findings:
The test plan covers all of the testing actions contained in the [CEM] and the body of this PP's evaluation activities.

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.

Evaluator Findings:
The test plan covers all of the testing actions contained in the [CEM] and the body of this PP's evaluation activities.

Verdict:
PASS.

5.5 AVA: VULNERABILITY ANALYSIS

5.5.1 VULNERABILITY SURVEY (AVA_VAN.1)

5.5.1.1 EVALUATION ACTIVITY

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.

Evaluator Findings:
The evaluator performed a Vulnerability Assessment for Endpoint Manager Mobile (EPMM) System 12, on October 13, 2025, and generated a report to document their findings with respect to this requirement. The evaluator performed a search of public information to find vulnerabilities that have been found in similar applications with a particular focus on network protocols the application uses and document formats it parses. The evaluator examined sources of publicly available information to identify potential vulnerabilities in the TOE. The sources of examined are as follows: • https://nvd.nist.gov/vuln/search#/nvd/home?resultType=records • https://www.ivanti.com/ • https://www.cve.org/ The evaluator examined public domain vulnerability searches by performing a keyword search. The terms used for this search were based on the vendor name, product name, key platform features leveraged by the product, and commonly known vulnerabilities for MDM products. As a result, the evaluator performed a search using the following keywords: • Ivanti Endpoint Manager Mobile (EPMM) 12 • Ivanti, Inc. • Oracle Linux 8.9 • Mobile Device Management • Mobile Management Software • MDM Server • MDM Agent • Galaxy S22 Ultra 5G Android 13 • Galaxy S22 Ultra 5G Android 14 • Mobile@Work • EPMM 12.3.1.0 • Ivanti MDM OpenSSL Component 1.1.1g • Ivanti MDM Bouncy Castle v1.0.2.4 • Ivanti Apache 2.4 • TLS/SSL MDM • Remote Wipe MDM • MDM enrollment vulnerability • MDM privilege escalation

The evaluator documents the sources consulted and the vulnerabilities found in the report.

Evaluator Findings:
The evaluator documented 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.

Evaluator Findings:
For each vulnerability found, the evaluator either provided a rationale with respect to its non-applicability, or the evaluator formulated a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable.

Verdict:
PASS.

6 DETAILED TEST CASES (TEST ACTIVITIES)

6.1 ENDPOINT MANAGER MOBILE (EPMM) SYSTEM 12

6.1.1 AUDIT TEST CASES

6.1.1.1 FAU_ALT_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator logs in to the MDM server as an administrator. • Then, the evaluator attempts to enroll the device with the MDM server. • The evaluator confirms and verifies that the enrollment is notified in the MDM server. • The evaluator attempts to retire the device with the MDM server. • The evaluator confirms and verifies that the retirement is notified in the MDM server.
Pass/Fail with Explanation	Pass, the evaluator is informed about the enrollment status of a device.

6.1.1.2 FAU_ALT_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator logs into the MDM server as an administrator. • The evaluator then configures an event setting that would generate alerts on MDM server.

	• The evaluator then creates a configuration that is not supported by MDM platform. • Then the evaluator verifies that an alert is generated on the MDM server.
Pass/Fail with Explanation	Pass, the administrator is alerted about the failed application of configuration.

6.1.1.3 FAU_ALT_EXT.1 TEST #3 – NA

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	N/A, no other alerts were selected for the TOE.

6.1.1.4 FAU_CRP_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	**Test 1:** Using the operational guidance, the evaluator shall demonstrate the ability to access the compliance reporting interface from an authorized entity and successfully obtain information about enrolled devices.
Test Steps	• The evaluator shall login to the TOE's reporting interface as an administrative user. • Then the evaluator shall execute command/instructions as per the operational guidance to list information on enrolled devices and confirms the same.
Pass/Fail with Explanation	Pass, the evaluator was able to obtain details about enrolled devices from compliance interface when logged in as authorized user.

6.1.1.5 FAU_CRP_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	**Test 2:** The evaluator shall attempt to access the compliance reporting interface from an unauthorized entity and demonstrate that the attempt is denied.
Test Steps	• The evaluator shall attempt to access the reporting interface as an unauthorized user. • Then the evaluator confirms that the access was unsuccessful.
Pass/Fail with Explanation	Pass, the evaluator was unable to access the reporting interface when attempted access as an unauthorized user.

6.1.1.6 FAU_GEN.1.1(1) TEST #1

Item	Data
Test Assurance Activity	The evaluator shall test the TOEs 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.
Test Steps	• The evaluator performs operations on the TOE that will generate auditable events for establishment and termination of communication channel for listed cryptographic protocol. • The evaluator then records the events that were generated as part of the above step.
Pass/Fail with Explanation	Pass, TOE was able to successfully able to generate audit data.

6.1.1.7 FAU_GEN.1.2(1) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator performs operations on the TOE that will generate auditable events. • The evaluator then records the events that were generated as part of the above step and confirms with the output specified in the AGD.
Pass/Fail with Explanation	Pass, Refer to test FAU_GEN1.1(1) Test#1.

6.1.1.8 FAU_GEN.1.1(2)/SRV TEST #1

Item	Data

Test Assurance Activity	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.
Test Steps	• The evaluator shall attempt to push or update an application. • The evaluator then attempts to install the application from the MDM agent end. • Finally, the evaluator verifies and confirms that the above steps make an audit entry as specified in the AGD.
Pass/Fail with Explanation	Pass, the TOE was generating logs for failed application installation as mentioned in the guidance documentation.

6.1.1.9 FAU_GEN.1.2(2) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator performs operations on the TOE that will generate auditable events. • The evaluator then records the events that were generated as part of the above step and confirms with the output specified in the AGD.
Pass/Fail with Explanation	Pass, the audit record generated is as per the guidance documentation.

6.1.1.10 FAU_NET_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator shall connect the TOE Agent with the server. • Then shall perform a ping test to see the devices are reachable. • The evaluator then shall disconnect the TOE agent from the network and again attempts to perform a ping test and confirms it fails.
Pass/Fail with Explanation	Pass, the evaluator confirms that the network reachability test works as expected.

6.1.1.11 FAU_SAR.1.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.1.12 FAU_SAR.1.2 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator shall access the server as an administrator. • Finally, the evaluator shall verify that he/she was able to view the audit record generated in step 1 and it was as defined in the AGD.
Pass/Fail with Explanation	Pass, the evaluator confirms that the audit records generated match the format mentioned in the AGD guide.

6.1.1.13 FAU_STG_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator shall configure an audit server for the TOE • The evaluator shall configure the audit server details in the TOE and attempt to establish connection. • The evaluator then shall perform activities that will generate audit records. • The evaluator shall then verify that a log of the above action is sent over to the syslog server. • The evaluator confirms that the data that is transmitted between the devices is not in plain text with the help of network capture.

Pass/Fail with Explanation	Pass, the audit data transferred between the TOE and the audit server was not transmitted in plain text.

6.1.1.14 FAU_STG_EXT.2.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: **Test 1:** The evaluator shall access the audit trail as an unauthorized user and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail.
Test Steps	• The evaluator shall attempt to access the TOE server as a read-only user. • Then the evaluator attempts to delete or modify the audit records and confirms unable to do so.
Pass/Fail with Explanation	Pass, the evaluator was not able to access the audit trail as an unauthorized user when access was attempted.

6.1.1.15 FAU_STG_EXT.2.1 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: **Test 2:** The evaluator shall access the audit trail as an authorized user and attempt to modify and delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records intended for modification and deletion are modified and deleted.
Test Steps	• The evaluator shall access the TOE server as an administrative user. • The evaluator shall access the TOE audit log interface. • Then the evaluator attempts to clear the audit log. • Finally, the evaluator shall verify and confirm the action performed in the above step is a success.
Pass/Fail with Explanation	Pass, the TOE allows deletion of audit records when logged in as an authorized user.

6.1.2 AUTH TEST CASES

6.1.2.1 FIA_ENR_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	**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.

Test Steps	• The evaluator attempts to enroll a TOE agent as device on the TOE server using incorrect credential. • The evaluator then verifies and confirms that the device is not enrolled.
Pass/Fail with Explanation	Pass, The TOE rejects the enrolment of a device when incorrect credentials are used.

6.1.2.2 FIA_ENR_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	**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.
Test Steps	• The evaluator attempts to enroll a TOE agent as device on the TOE server using correct credentials. • The evaluator then verifies and confirms that the device is enrolled
Pass/Fail with Explanation	Pass, the evaluator was able to enroll the device with the correct credential.

6.1.2.3 FIA_ENR_EXT.1.2 TEST #1

Item	Data
Test Assurance Activity	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.)
Test Steps	• The evaluator shall configure the TOE server's per-user device limit to prevent enrollment of devices as per the operational guidance. • The evaluator then shall attempt to enroll a device and confirms that it fails.
Pass/Fail with Explanation	Pass, The TOE rejects the registration of a device once the max device limit is reached.

6.1.2.4 FIA_UAU.1.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.2.5 FIA_UAU.1.2 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	EPMM Server: • The evaluator attempts to perform some prohibited actions without proper authentication. • The evaluator verifies that the above action step fails. MDM Agent: • The evaluator attempts to perform some prohibited actions without proper authentication. • The evaluator verifies that the above action step fails.
Pass/Fail with Explanation	Pass, performing actions was not allowed for failed authentication.

6.1.2.6 FIA_UAU.1.2 TEST #2

Item	Data
Test Assurance Activity	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.
Test Steps	EPMM Server: • The evaluator attempts to perform some prohibited configuration changes with proper authentication. • The evaluator verifies that the above action step is a success. MDM Agent: • The evaluator attempts to perform some prohibited configuration changes with proper authentication. • The evaluator verifies that the above action step is a success.
Pass/Fail with Explanation	Pass, the evaluator was able to perform operations post authenticating the device with the server.

6.1.2.7 FIA_X509_EXT.1.1(1) TEST #1 **[TD0641]**

Item	Data
Test Assurance Activity	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. **TD0641 has been applied.**
Test Steps	Client: Establish 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. • Configure the CA certificate lacking the basicConstraints extension. • Verify that the signing CA certificate does not contain the basicConstraints extension • Sign the certificate using CA certificate does not contain the basicConstraints extension. • Use Acumen-tlss tool to initiate a connection to the TOE with the chain does not contain the basicConstraints extension and verify the connections failed • Verify the connection failure via packet capture • Verify the connection failure via logs ○ By setting the basicConstraints field in an issuing certificate to have CA=False. • Configure the CA certificate with the flag in the basicConstraints extension set to FALSE. • Verify that the signing CA certificate has the cA flag in the basicConstraints extension set to FALSE • Sign the certificate using ICA with basic constraints set to FALSE. • Use Acumen-tlss tool to initiate a connection to the TOE with the chain with the basicConstraints set to FALSE and verify the connections failed • Verify the connection failure via packet capture • Verify the connection failure via logs

- By omitting the CA signing bit of the key usage field in an issuing certificate.
  - Configure the CA certificate lacking the CA signing bit in the Key usage field.
  - Load the certificate lacking the CA signing bit on the TLS server.
  - Sign the certificate using ICA with no certificate sign key usage.
  - Use Acumen-tlss tool to initiate a connection to the TOE without the CA signing bit of the key usage field in the chain and verify the connections failed
  - Verify the connection failure via packet capture
  - Verify the connection failure via logs

- By setting the path length field of a valid CA field to a value strictly less than the certificate path.
  - Configure the root CA certificate with the Path length of 1.
  - Configure the Intermediate CA1 certificate with the Path length of 0.
  - Configure the Intermediate CA2 certificate with the Path length of 0.
  - Sign the node certificate with ICA2.
  - Use Acumen-tlss tool to initiate a connection to the TOE with path length field of a valid CA field to a value strictly less than the certificate path and verify the connections failed
  - Verify the connection failure via packet capture
  - Verify the connection failure via logs

Valid certificate chain

- Create a full chain of certificates to connect to the TOE.
- Upload a complete certificate validation chain to the TOE.
- Attempt to connect to the TOE with the full chain of proper certificates
- Verify the connection is successful
- Verify the connection is successful via packet capture

Invalid certificate chain

- Delete the ICA2 certificate from the chain
- Attempt to connect to the TOE without the intermediate CA and verify the connection failed
- Verify the connection failure via packet capture
- Verify the connection failure via logs

Server:

- Establish 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.
  - Configure the CA certificate lacking the basicConstraints extension

	○ Verify that the signing CA certificate does not contain the basicConstraints extension • The evaluator verifies that the certificate upload fails with the help of system logs. • By setting the basicConstraints field in an issuing certificate to have CA=False. • The evaluator manipulates the ICA certificate with the flag in the basicConstraints extension set to FALSE • The evaluator verifies that the certificate upload fails with the help of system logs. • By omitting the CA signing bit of the key usage field in an issuing certificate. ○ Configure an ICA certificate lacking the CA signing bit in the Key usage field. • The evaluator verifies that the certificate upload fails with the help of system logs. • By setting the path length field of a valid CA field to a value strictly less than the certificate path. • Configure the root CA certificate with the Path length of 1. • Configure the Intermediate certificate with the Path length of 0. • The evaluator verifies that the certificate upload fails with the help of system logs. Valid certificate chain • Create a full chain of certificates to connect to the TOE. • Upload a complete certificate validation chain to the TOE. • Attempt to connect to the TOE with the full chain of proper certificates • Verify the connection is successful Invalid certificate chain • Delete the ICA certificate from the chain • Attempt to connect to the TOE without the intermediate CA and verify the connection failed • The evaluator verifies that the certificate upload fails with the help of system logs.
Pass/Fail with Explanation	Pass, TOE allows for successful TLS connection and certificate upload when a complete certificate chain is presented. And rejects TLS connection and certificate upload when an incomplete certificate chain is presented.

6.1.2.8 FIA_X509_EXT.1.1(1) TEST #2 **[TD0641]**

Item	Data
Test Assurance Activity	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. **TD0641 has been applied.**
Test Steps	Client:

	• The evaluator shall create and configure the TOE server with certificates of valid path length. • The evaluator makes sure that the end certificate in the validation path is expired. • The evaluator then verifies that a connection attempt with the expired certificate is not possible and verifies the same with system logs. • The evaluator confirms the above step with the help of a traffic capture. Server: • The evaluator attempts to upload a certificate chain where the end certificate is expired. • The evaluator verifies that the certificate upload fails with the help of system logs. • The evaluator shall create and configure the TOE server with certificates of valid path length. • The evaluator makes sure that the end certificate in the validation path is expired. • The evaluator then verifies that a connection attempt with the expired certificate is not possible and verifies the same with system logs. • The evaluator confirms the above step with the help of a traffic capture.
Pass/Fail with Explanation	Pass, Neither the TLS connection nor the certificate upload was successful with an expired certificate.

6.1.2.9 FIA_X509_EXT.1.1(1) TEST #3 **[TD0641]**

Item	Data
Test Assurance Activity	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. **TD0641 has been applied.**
Test Steps	Client:

	CRL • Create chain of certificates with CRL Distribution Point. • Attempt connection with valid certificate • Verify the successful connection • Verify the successful connection via packet capture. • Revoke the peer leaf certificate • Attempt to make a connection • Verify the connection failure • Verify the connection failure via packet capture • Verify the connection failure via logs • Unrevoke the peer's leaf certificate and revoke the server's intermediate certificate • Attempt to make a connection • Verify the connection failure via packet capture • Verify the connection failure via logs Server: CRL • Create chain of certificates with CRL Distribution Point. • Attempt to upload a valid certificate chain. • Verify the successful upload of certificates. • Verify the successful upload via packet capture and CRL server output. • Revoke the peer leaf certificate • Attempt to upload a certificate chain in which the end certificate is revoked and confirm failure. • Verify the upload failure via logs. • Verify the certificate upload failure via packet capture and CRL server output. • Revoke the intermediate certificate. • Attempt to upload a certificate chain in which the end certificate is revoked and confirm failure. • Verify the upload failure via logs. • Verify the certificate upload failure via packet capture and CRL server output. OCSP • Create chain of certificates with an OCSP endpoint specified. • Attempt to upload a valid certificate chain. • Verify the successful certificate upload. • Verify the OCSP responder and wireshark output. • Attempt to upload a certificate chain for which the end certificate is revoked. • Verify the certificate upload failure. • Verify the OCSP responder and wireshark output. • Confirm the certificate upload failuresystem logs. • Attempt to upload a certificate chain where the intermediate certificate is revoked.

	• Verify the certificate upload failure. • Verify the OCSP responder and Wireshark output. • Confirm the certificate upload failure system logs.
Pass/Fail with Explanation	Pass, The TOE will accept certificate upload and connection when the certificates are valid. The TOE will also reject the certificate upload and connection when either the end or the intermediate certificate of the certificate chain is revoked.

6.1.2.10 FIA_X509_EXT.1.1(1) TEST #4 **[TD0641]**

Item	Data
Test Assurance Activity	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. **TD0641 has been applied.**
Test Steps	Client: CRL: • The evaluator configures the TOE with the certificates with the ICA certificate lacking CRL sign Key usage. • The evaluator then verifies when the intermediate certificate is lacking the CRL sign key the connection attempt fails. • The evaluator confirms this with the help of network traffic. • The evaluator confirms the connection failure with audit log Server: CRL: • configure a certificate where the end certificate has been signed by an intermediate that's lacking CRL sign Key usage. • The evaluator then verifies the upload of a certificate chain whose ICA lacks CRL sign bit fails. • The evaluator confirms this with the help of system logs and network traffic capture. OCSP:

	• The evaluator configures the TOE server with the certificates where the end certificate lacking OCSP sign EKU set. • The evaluator then verifies when the above step is performed the connection attempt fails. • The evaluator then verifies the OCSP responder and Wireshark capture output. • The evaluator confirms the failure with the help TOE output and system logs.
Pass/Fail with Explanation	Pass. The TOE does not allow successful connections for certificates without the CRL sign key usage and OCSP signing Purpose.

6.1.2.11 FIA_X509_EXT.1.1(1) TEST #5 **[TD0641]**

Item	Data
Test Assurance Activity	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.) **TD0641 has been applied.**
Test Steps	Client: • The evaluator creates a chain of certificates of length three. • Start the server using the acumen-tls tool with a modified byte within the first 8 bytes of the certificate, the connection should fail. • Verify the error logs on the TOE showing failure due to the wrong tag. • Verify the unsuccessful connection with packet capture. Server: • Using ghex modify any byte within the first 8 bytes of the certificate. • The evaluator then attempts to upload the certificate and verifies failure. • The above step is confirmed with the help of system logs.
Pass/Fail with Explanation	Pass. Certificate validation fails upon upload and connection when a modified certificate is used.

6.1.2.12 FIA_X509_EXT.1.1(1) TEST #6 **[TD0641]**

Item	Data
Test Assurance Activity	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.) **TD0641 has been applied.**
Test Steps	Client: • The evaluator creates a chain of certificates of length three. • Start the server using the acumen-tls tool with a modified byte within the last byte of the certificate, the connection should fail. • Verify the error logs on the TOE showing failure due to the decrypt error • Verify the unsuccessful connection with packet capture. Server: • Using ghex modify any byte within the last 8 bytes of the certificate. • The evaluator then attempts to upload the certificate and verifies failure. • The above step is confirmed with the help of system logs.
Pass/Fail with Explanation	Pass, the TLS connection is not established and certificate was not successful when a modified certificate is used.

6.1.2.13 FIA_X509_EXT.1.1(1) TEST #7 **[TD0641]**

Item	Data
Test Assurance Activity	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.) **TD0641 has been applied.**
Test Steps	Client: • **The evaluator creates a chain of certificates of length three.** • Start the server using the acumen-tls tool with a modified byte of the certificate, the connection should fail. • Verify the error logs on the TOE showing failure due to the wrong tag. • Verify the unsuccessful connection with packet capture. Server: • Use ghex tool to a modify byte of the server certificate's public key. • The evaluator then attempts to upload the certificate and confirms it fails. • The above is verified with the help of system logs.
Pass/Fail with Explanation	Pass, TLS connection and certificate upload is not successful when a modified certificate is used.

6.1.2.14 FIA_X509_EXT.1.1(1) TEST #8A **[TD0641]**

Item	Data
Test Assurance Activity	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 four certificates: the node certificate to be tested, two Intermediate CAs, 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. **TD0641 has been applied.**
Test Steps	Client: • The evaluator attempts to configure the TOE with EC certificates of valid length. • The evaluator then verifies the connection attempt is a success and confirms with the help of a network capture. Server : • The evaluator attempts to configure the TOE with EC certificates of valid length. • The evaluator then verifies the connection attempt is a success and confirms with the help of a network capture.
Pass/Fail with Explanation	Pass, the evaluator was able to see successful establishment of connection and upload of certificate when valid certificates with named curve are used.

6.1.2.15 FIA_X509_EXT.1.1(1) TEST #8B **[TD0641]**

Item	Data
Test Assurance Activity	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 four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. **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.

	TD0641 has been applied.
Test Steps	Client: • The evaluator creates a chain of four certificates. • The evaluator attempts to connect with EC certificates of where ICA has explicit format version. • The evaluator makes sure that the attempted connection is failed. • The evaluator then verifies the certificate is invalid and confirms with the help of a network capture. Server: • The evaluator attempts to upload EC certificates where ICA has explicit format version. • The evaluator makes sure that the attempted connection is failed and confirms the same with system logs.
Pass/Fail with Explanation	Pass, the TLS connection is not established and certificate was not successful when a modified certificate with explicit curve is used.

6.1.2.16 FIA_X509_EXT.1.2(1) TEST #1

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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.
Test Steps	Client: • The evaluator attempts to upload a cert chain with the ICA cert without basicConstraint. • The evaluator verified that the connection attempt and upload fails. • The evaluator confirms the failure with the help of system logs and network capture. Server: • The evaluator attempts to upload a cert chain with the ICA cert without basicConstraint. • The evaluator verified that the upload fails and confirms with the help of system logs.
`	The evaluator confirms that certificate validation fails when the ICA cert does not contain basicConstarint.
Pass/Fail with Explanation	Pass, certificate validation fails when the ICA does not have set BasicConstraint.

6.1.2.17 FIA_X509_EXT.1.2(1) TEST #2

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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.
Test Steps	Client: • The evaluator attempts to connect with the ICA cert with basicConstraint CA set to false. • The evaluator verified that the connection attempt fails and confirms with the help of a system logs and network capture. Server: • The evaluator attempts to upload a cert chain with the ICA cert with basicConstraint CA set to false. • The evaluator verified that the upload fails and confirms with the help of a system logs.
Pass/Fail with Explanation	Pass, neither TLS connection nor certificate upload is successful when the certificate of the CA issuing the TOEs certificate has the cA flag in the basicConstraints extension not set.

6.1.2.18 FIA_X509_EXT.1.2(1) TEST #3

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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.
Test Steps	Client: • The evaluator attempts to upload a cert chain with the ICA cert with basicConstraint set to true. • The evaluator verified that the connection is successful.

	• The evaluator also confirms the successful connection with the help of network capture. Server: • The evaluator attempts to upload a cert chain with the ICA cert with basicConstraint set to true. • The evaluator verified that the upload succeeds and confirms with the help of system logs. • The evaluator shall verify the certificate contents uploaded to the TOE.
Pass/Fail with Explanation	Pass, the certificate validation was successful when the basicConstraints is set to TRUE.

6.1.2.19 FIA_X509_EXT.2.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.2.20 FIA_X509_EXT.2.2 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	Client: • The evaluator verifies that the CRL server is not operational • The evaluator shall then attempt to configure the syslog server to connect with the TLS server and confirm it fails • The evaluator shall confirm the connection failure with the help of system logs Server: • The evaluator verifies that the CRL server is not operational • The evaluator shall then attempt to upload the certificate to the TOE and confirm it fails • The evaluator shall verify the upload failure with the help of system logs
Pass/Fail with Explanation	Pass. When the TOE cannot establish a connection to determine the validity of a certificate, the TOE does not accept the certificate.

6.1.3 CRYPTO TEST CASES

6.1.3.1 FCS_CKM.1 RSA**[TD0951]**

Item	Data
Test Assurance Activity	**Key Generation for FIPS PUB 186-4 or FIPS PUB 186-5 RSA Schemes** The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: 1. Random Primes: a. Provable primes b. Probable primes 2. Primes with Conditions: a. Primes p1, p2, q1,q2, p and q shall all be provable primes b. Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes c. Primes p1, p2, q1,q2, p and q shall all be probable primes To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. If possible, the Random Probable primes method should also be verified against a known good implementation as described above. Otherwise, the evaluator shall have the TSF generate 10 keys pairs for each supported key length nlen and verify: • n = p*q, • p and q are probably prime according to Miller-Rabin tests, • GCD(p-1,e) = 1, • GCD(q-1,e) = 1, • 2^16 <= e <= 2^256 and e is an odd integer, • |p-q| > 2^(nlen/2 - 100), • p >= squareroot(2)*(2^(nlen/2 -1)), • q >= squareroot(2)*(2^(nlen/2 -1)), • 2^(nlen/2) < d < LCM(p-1,q-1), • e*d = 1 mod LCM(p-1,q-1). **TD0951 has been applied.**

Pass/Fail with Explanation	Algorithm: RSA KeyGen Key size / Modulus: 2048,3072,4096 CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.2 FCS_CKM.1 ECC **[TD0951]**

Item	Data
Test Assurance Activity	**FIPS 186-4 or FIPS 186-5 ECC Key Generation Test** For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation. **FIPS 186-4 or FIPS 186-5 Public Key Verification (PKV) Test** For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values. **TD0951 has been applied.**
Pass/Fail with Explanation	Algorithm: ECDSA KeyGen Curves: P-256, P-384, P-521 CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.3 FCS_CKM.2 SP800-56A

Item	Data
Test Assurance Activity	**SP800-56A Key Establishment Schemes** The evaluator shall verify a TOEs implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the recommendation. These components include the calculation of the primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag. *Function Test*

	The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested. The evaluator shall obtain the DKM, the corresponding TOEs public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE ID fields. If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret. The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values. If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm. *Validity Test* The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOEs public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE ID fields. The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOEs static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass). The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOEs results with the results using a known good implementation verifying that the TOE detects these errors.

Pass/Fail with Explanation	Algorithm: KAS-ECC-SSC SP800-56Ar3 Domain Parameter Generation Methods: P-256, P-384, P-521 CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.4 FCS_CKM.2 RSA

Item	Data
Test Assurance Activity	**RSA-based key establishment** The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1(1), FTP_TRP.1(2), FTP_TRP.1(3), FTP_ITC.1(1), FTP_ITC.1(2), FPT_ITT.1(1), and FPT_ITT.1(2) that uses RSAES-PKCS1-v1_5.
Pass/Fail with Explanation	Pass. This testing was performed in conjunction with FTP_TRP.1(1) Test #1, FTP_TRP.1(2) Test #1, FTP_TRP.1(3) Test #1, FTP_ITC.1(1) Test #1, FTP_ITC.1(2) Test #1, FPT_ITT.1(2) Test #1 and FCS_TLSS_EXT.1.3 Test#1 to demonstrate correct operation.

6.1.3.5 FCS_CKM_EXT.4.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.3.6 FCS_CKM_EXT.4.2 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	TLS Session Key: • The evaluator attempts to connect to the TLS server using OpenSSL from the TOE. • The evaluator verifies the TLS key information from the above step. • The evaluator then used the debugger to dump the memory image of the TOE whilst TLS connection is active and verified that information is present when searched. • The evaluator then closed the TLS connection and deleted the configuration for memory overwrite. • The evaluator then used the debugger to dump the memory image of the TOE and confirms that the key is not present in the memory image when searched. Certificate private key: • The evaluator uploads a portal certificate and verifies it is successful. • The evaluator then used the debugger to dump the memory file with the current certificate chain and • The evaluator replaces the portal certificate with a different one and verifies it succeeds. • The evaluator shall terminate the TOE session and restart, shall then confirm the uploaded certificate. • The evaluator then searches for the dumped binary image for key information that was captured with the initial upload and confirms it's not there.
Pass/Fail with Explanation	Pass, no copies of keys are found in memory.

6.1.3.7 FCS_CKM_EXT.4.2 TEST #2 – NA

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	N/A, TOE is not in a limited operating environment, and the evaluator was able to use debugger to perform the test.

6.1.3.8 FCS_COP.1.1(1) AES-GCM

Item	Data
Test Assurance Activity	**<u>If "implement functionality" is selected:</u>** **<u>AES-GCM Test</u>** **Test 1**: The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths: • 128 bit and 256 bit keys • Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported. • Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported. • Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested. The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known. The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5- tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
Pass/Fail with Explanation	Algorithm: AES-GCM Direction: Decrypt, Encrypt • Key Length: 128, 256 • IV Length: 96, 128 • Payload Length: 120, 128, 248, 256 • AAD Length: 0, 120, 128, 256

	CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.9 FCS_COP.1.1(2) HASHING

Item	Data
Test Assurance Activity	The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. **Short Messages Test Bit-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Short Messages Test Byte-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Selected Long Messages Test Bit-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Selected Long Messages Test Byte-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Pseudorandomly Generated Messages Test** • **Test 1:** This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1

	of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.
Pass/Fail with Explanation	Algorithm: SHA2-256, SHA2-384, SHA2-512 CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.10 FCS_COP.1.1(3) SIGGEN ECDSA **[TD0951]**

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** **ECDSA Algorithm Tests** **Test 1: ECDSA FIPS 186-4 or FIPS PUB 186-5 Signature Generation Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation. **Test 2: ECDSA FIPS 186-4 or FIPS PUB 186-5 Signature Verification Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values. **TD0951 has been applied.**
Pass/Fail with Explanation	Algorithm: ECDSA SigGen, ECDSA SigVer Key size / Modulus: P-256, P-384, P-521 CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.11 FCS_COP.1.1(3) SIGGEN RSA

Item	Data
Test Assurance Activity	**RSA Signature Algorithm Tests** **Test 1:** Signature Generation Test The evaluator shall verify the implementation of RSA Signature Generation by the TOE using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10

	messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator shall have the TOE use their private key and modulus value to sign these messages. The evaluator shall verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures. **Test 2:** Signature Verification Test The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's valid and invalid signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure. The evaluator shall use these test vectors to emulate the signature verification test using the corresponding parameters and verify that the TOE detects these errors.
Pass/Fail with Explanation	Algorithm: RSA SigGen, RSA SigVer Key size / Modulus: 2048, 3072, 4096 CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.12 FCS_COP.1.1(4) KEYED-HASH

Item	Data
Test Assurance Activity	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known good implementation.
Pass/Fail with Explanation	Algorithm: HMAC-SHA2-256, HMAC-SHA2-384 CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.13 FCS_IV_EXT.1.1

Item	Data
Test Assurance Activity	The evaluator shall ensure that the generation of IVs for each key encrypted by the same KEK meets Table 4.
Pass/Fail with Explanation	Pass, refer to FCS_COP.1.1(1) AES-GCM test case for CAVP details.

6.1.3.14 FCS_RBG_EXT.1.1/SRV

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** The evaluator shall perform the following tests. The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality. If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits, (3) generate a second block of random bits, and (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A). If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits, and (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. Entropy input: the length of the entropy input value must equal the seed length. Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length. Personalization string: The length of the personalization string must be less than or equal to seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

	Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.
Pass/Fail with Explanation	Algorithm: Hash DRBG & Counter DRBG Mode: Hash DRBG(SHA2-256, SHA2-512), Counter DRBG(AES-256) CAVP #: A6073, A6074 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.15 FCS_RBG_EXT.1.2/SRV

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.3.16 FCS_STG_EXT.1.1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.3.17 FCS_STG_EXT.2.1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.3.18 FCS_HTTPS_EXT.1.2 TEST #1

Item	Data
Test Assurance Activity	**Test 1:** The evaluator shall attempt to establish an HTTPS connection with a web server, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS. Other tests are performed in conjunction with the TLS evaluation activities.
Test Steps	Admin Portal: • The evaluator shall attempt a connection with the TOE server. • The evaluator then makes sure that the configuration in the above step is a success and TLS/HTTPS connection is established.

	• The evaluator then confirms the same with the help of a network capture System Manager: • The evaluator shall attempt a connection with the TOE server. • The evaluator then makes sure that the configuration in the above step is a success and TLS/HTTPS connection is established. • The evaluator then confirms the same with the help of a network capture User Portal: • The evaluator shall attempt a connection with the TOE server. • The evaluator then makes sure that the configuration in the above step is a success and TLS/HTTPS connection is established. • The evaluator then confirms the same with the help of a network capture
Pass/Fail with Explanation	Pass, the evaluator confirms the session established is HTTPS/TLS.

6.1.3.19 FCS_TLSC_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 1:** The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	• Configure the TOE to the Syslog server • Establish a connection with the TOE over TLS using the cipher TLS_RSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_RSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful

	• Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture
Pass/Fail with Explanation	Pass, the TOE was able to make TLS connection with each of the supported cipher suite.

6.1.3.20 FCS_TLSC_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 2:** The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation. The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established. The evaluator shall repeat this test using a different, but otherwise valid and trusted, certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension and ensure that a connection is not established. Ideally, the two certificates should be similar in structure, the types of identifiers used, and the chain of trust.
Test Steps	• Create a server certificate with the Server Authentication in the extendedKeyUsage field • Use openssl to initiate a successful TLS connection • Verify the connection is established • Verify the connection is established via packet capture • Create a server certificate without the Server Authentication in the extendedKeyUsage field • Use openssl to initiate a connection to the TOE with a server certificate without the Server Authentication field resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Pass/Fail with Explanation	Pass, the connection is established when the server certificate contains the Server Authentication purpose in the extendedKeyUsage extension, and the connection is not established when the server certificate lacks the Server Authentication purpose in the extendedKeyUsage extension.

6.1.3.21 FCS_TLSC_EXT.1.1 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 3:** The evaluator shall send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message.
Test Steps	• Use acumen-tls tool to initiate a connection to the TOE with an unsupported cipher suite server certificate when the TOE runs in RSA resulted in a failure. • Verify the connection failure via packet capture • Verify the connection failure is via logs
Pass/Fail with Explanation	Pass, the TOE rejected connection when ECDSA cipher is offered for TLS session when the TOE is operating in RSA.

6.1.3.22 FCS_TLSC_EXT.1.1 TEST #4

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the non-supported ciphersuite (NULL_WITH_NULL_NULL) • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, the TOE did not establish a TLS connection when NULL cipher is used for connection.

6.1.3.23 FCS_TLSC_EXT.1.1 TEST #5.1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.1:** Change the TLS version selected by the server in the Server Hello to an undefined TLS version (for example 1.5 represented by the two bytes 03 06) and verify that the client rejects the connection.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the undefined TLS version • Verify the connection failure via packet capture • Verify the connection failure via logs

Pass/Fail with Explanation	Pass. The TLS connection is rejected when an undefined TLS version is used.

6.1.3.24 FCS_TLSC_EXT.1.1 Test #5.2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.2:** Change the TLS version selected by the server in the Server Hello to the most recent unsupported TLS version (for example 1.1 represented by the two bytes 03 02) and verify that the client rejects the connection.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the unsupported TLS version • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejected the TLS connection when an unsupported version of TLS is used.

6.1.3.25 FCS_TLSC_EXT.1.1 TEST #5.3

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.3:** [conditional] If DHE or ECDHE cipher suites are supported, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the modified server nonce • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejects the connection when a modified server nonce is used.

6.1.3.26 FCS_TLSC_EXT.1.1 TEST #5.4

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.4:** Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message.

	The evaluator shall verify that the client does not complete the handshake and no application data flows.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with an unsupported ciphersuite • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection attempt is rejected when an unsupported cipher is used to make the connection.

6.1.3.27 FCS_TLSC_EXT.1.1 TEST #5.5

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.5:** [conditional] If DHE or ECDHE cipher suites are supported, modify the signature block in the server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with a modified signature in the Server Key Exchange • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection was not a success when a modified server signature is used.

6.1.3.28 FCS_TLSC_EXT.1.1 TEST #5.6

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.6:** Modify a byte in the Server Finished handshake message, and verify that the client does not complete the handshake and no application data flows.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with a modified Server Finished handshake message • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection is not established when a modified server finished handshake message is used.

6.1.3.29 FCS_TLSC_EXT.1.1 TEST #5.7

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.7:** Send a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message and verify that the client does not complete the handshake and no application data flows. The message must still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with a garbled message after the Change Cipher Spec message is issued • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. After the server has issued the Change Cipher Spec message, the client does not complete the handshake and no application data flows.

6.1.3.30 FCS_TLSC_EXT.1.2 TEST #1 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 1:** The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. Note that some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1. **TD0499 has been applied.**
Test Steps	FQDN • Create a Server Certificate with no SAN and CN that does not match the reference identifier. • The evaluator shall configure the TOE with the respective reference identifier and attempt to initiate a connection with the server. • Use OpenSSL initiate connection with the TOE using a certificate missing the SAN but with a CN that does not match the reference identifier. • Verify the connection failure. • Verify the connection failure via packet capture.

	• Verify the connection failure via logs.
Pass/Fail with Explanation	Pass, TOE rejects the connection when the server certificate has an invalid CN and contains no SAN.

6.1.3.31 FCS_TLSC_EXT.1.2 TEST #2 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 2:** The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type. **TD0499 has been applied.**
Test Steps	FQDN • Create a Server Certificate with a SAN does not match the reference identifier and a CN that does match the reference identifier • Use OpenSSL to initiate a connection to the TOE using a certificate the matched CN but with a SAN that does not match the reference identifier and verify the connection failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejects the connection when the server certificate contains a CN that matches the reference identifier and SAN that has an invalid reference identifier.

6.1.3.32 FCS_TLSC_EXT.1.2 TEST #3 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 3:** [conditional] If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

	TD0499 has been applied.
Pass/Fail with Explanation	Pass, the TOE mandates the presence of SAN in the certificate.

6.1.3.33 FCS_TLSC_EXT.1.2 TEST #4 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 4:** The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. **TD0499 has been applied.**
Test Steps	CN as FQDN • Create a Server Certificate with a SAN that does match the reference identifier and a CN that does not match the reference identifier • Use the OpenSSL to run as a server using a certificate the matched SAN but with a CN that does not match the reference identifier • Verify the connection succeeds • Verify the connection succeeds via packet capture
Pass/Fail with Explanation	Pass, The TOE accepts the connection when a server certificate contains a CN that mismatches the reference identifier and does contains the SAN extension that matches.

6.1.3.34 FCS_TLSC_EXT.1.2 TEST #5.1 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.1:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.

	TD0499 has been applied.
Test Steps	FQDN • Create a server certificate containing a wildcard that is not in the left-most label of SAN. • Configure the correct reference identifier in the TOE. • Use the OpenSSL to initiate a connection with the TOE using a certificate containing a wildcard that is not in the left-most label of SAN • Verify the connection failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejects the connection when the server presents a wildcard certificate that is not in the left most label.

6.1.3.35 FCS_TLSC_EXT.1.2 TEST #5.2 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.2:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.come) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	FQDN • Create a wildcard server certificate with wildcard in the left-most label • Configure the TOE with reference identifier with a single left-most label and attempt a connection with server • Using OpenSSL attempt to establish a TLS connection and confirm success • Verify the successful connection with the help of network capture • Configure the TOE with reference identifier without a left-most label and attempt a connection with server • Using OpenSSL attempt to establish a TLS connection and confirm the connection is rejected

	• Verify the rejected connection with the help of network capture • Verify the rejected connection with the help of system logs • Configure the TOE with reference identifier with two left-most labels and attempt a connection with server • Using OpenSSL attempt to establish a TLS connection and confirm the connection is rejected • Verify the rejected connection with the help of network capture • Verify the rejected connection with the help of system logs
Pass/Fail with Explanation	Pass, TOE accepted the connection when the server uses a wildcard certificate in the leftmost label and TOE is configured with reference identifier with single left-most label. And rejected connection when a server uses a wildcard certificate in the left most label and the reference identifier in the TOE is configured without a left-most label and with two left-most labels.

6.1.3.36 FCS_TLSC_EXT.1.2 TEST #5.3 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.3:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	FQDN • Create a server certificate with left-most label proceeding by a public suffix. • Configure the TOE with reference identifier with a single left most label and attempt connection with the server. • Using OpenSSL to initiate a connection with the TOE with a server certificate containing a wildcard in the left-most label proceeding the public suffix. • Verify the connection failure via packet capture • Verify the connection failure via logs • Configure the TOE with reference identifier with two single left-most labels and attempt connection with the server. • Using OpenSSL to initiate a connection with the TOE with a server certificate containing a wildcard in the left-most label proceeding the public suffix.

	• Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejects connection when server presents with wildcard certificate with leftmost label followed by a public suffix.

6.1.3.37 FCS_TLSC_EXT.1.2 TEST #5.4 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6 A pinned certificate must be added prior to performing Test 7 **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier The support for wildcards is intended to be optional If wildcards are supported, the first, second, and third tests below shall be executed If wildcards are not supported, then the fourth test below shall be executed • **Test 5 4:** [conditional]: If wildcards are not supported, the evaluator shall present a server certificate containing a wildcard in the left-most label (e g * example com) The evaluator shall configure the reference identifier with a single left-most label (e g foo example com) and verify that the connection fails **TD0499 has been applied**
Pass/Fail with Explanation	Pass, wildcards are supported by the TOE.

6.1.3.38 FCS_TLSC_EXT.1.2 TEST #6 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 6:** [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall

	repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	NA, TOE does not support URI or Service Name reference identifier.

6.1.3.39 FCS_TLSC_EXT.1.2 TEST #7 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 7:** [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	N/A, certificate pinning is not supported by the TOE.

6.1.3.40 FCS_TLSC_EXT.1.3 TEST #1A **[TD0513]**

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1a:** The evaluator shall demonstrate that a server using a certificate with a valid certification path successfully connects. **TD0513 has been applied.**
Test Steps	• Create a full chain of certificates to connect to the TOE

	• Use OpenSSL to run as a server with the certificates mentioned in the above step and verify the connection is successful • Verify the connection is successful via packet capture
Pass/Fail with Explanation	Pass, TOE accepts the connection when full chain is provided by the server.

6.1.3.41 FCS_TLSC_EXT.1.3 TEST #1B **[TD0513]**

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1b:** The evaluator shall modify the certificate chain used by the server in test 1a to be invalid and demonstrate that a server using a certificate without a valid certification path to a trust store element of the TOE results in an authentication failure. **TD0513 has been applied.**
Test Steps	• Create a certificate signed with an invalid certification path to TOE's trust store. • Attempt to connect to the TOE with the invalid intermediate CA and verify the connection failed • Verify that the connection failed via packet capture • Verify that the connection failed via logs
Pass/Fail with Explanation	Pass, TLS connection was not established when an invalid certificate path is used.

6.1.3.42 FCS_TLSC_EXT.1.3 TEST #1C **[TD0513]** - NA

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1c** [conditional]: If the TOE trust store can be managed, the evaluator shall modify the trust store element used in Test 1a to be untrusted and demonstrate that a connection attempt from the same server used in Test 1a results in an authentication failure. **TD0513 has been applied.**
Pass/Fail with Explanation	NA, TOE only supports upload of certificates and trust store cannot be managed on the TOE.

6.1.3.43 FCS_TLSC_EXT.1.3 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 2:** The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.
Test Steps	• Attempt to connect to the TOE with a revoked Certificate and verify the connection failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejected connection when server presents a revoked certificate.

6.1.3.44 FCS_TLSC_EXT.1.3 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 3:** The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.
Test Steps	• Attempt to connect to the TOE with an expired certificate and verify the connection failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when an expired certificate is used.

6.1.3.45 FCS_TLSC_EXT.1.3 TEST #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 4:** The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.
Test Steps	• Create a Server Certificate with an invalid reference identifier • Use OpenSSL to initiate a connection to the TOE and verify that the connection fails with an invalid reference identifier • Verify the connection failure via packet capture

	• Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejects the connection when server certificate uses invalid reference identifier.

6.1.3.46 FCS_TLSC_EXT.2.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 1:** The evaluator shall establish a connection to a server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE without the Server's Certificate Request and verify that the connection succeeds • Verify the connection succeeds via packet capture
Pass/Fail with Explanation	Pass, the TOE does not send certificate to a non-MA TLS server.

6.1.3.47 FCS_TLSC_EXT.2.1 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 2:** The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Test Steps	• Use the Acumen-tls tool to initiate a connection to the TOE with the Server's Certificate Request and verify that the connection succeeds • Verify the connection succeeds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message via packet capture
Pass/Fail with Explanation	Pass, The TOE confirms mutual authentication is taking place and TOE responds with non-empty client's certificate message and certificate verify.

6.1.3.48 FCS_TLSC_EXT.5.1 TEST #1

Item	Data

Test Assurance Activity	The evaluator shall also perform the following test: **Test 1:** The evaluator shall configure a server to perform key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.
Test Steps	• Initiate a connection from the TOE over TLS , where the server will run using the curve secp256r1, secp384r1, secp521r1 and verify the connection succeeds. • Verify with packet capture
Pass/Fail with Explanation	Pass, The TOE accepts TLS connection support for each curve.

6.1.3.49 FCS_TLSS_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 1:** The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	• Establish a connection with the TOE over TLS using TLS_RSA_WITH_AES_128_GCM_SHA256 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using TLS_RSA_WITH_AES_256_GCM_SHA256 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 and show the connection being successful. • Verify the packet capture.

Pass/Fail with Explanation	Pass, successful connection over TLS using claimed cipher suites.

6.1.3.50 FCS_TLSS_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 2:** The evaluator shall send a Client Hello to the server with a list of cipher suites that does not contain any of the cipher suites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the server denies the connection.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify that the connection fails with the non-supported cipher suite ○ RSA_WITH_3DES_EDE_CBC_SHA ○ NULL_WITH_NULL_NULL • Verify the connection failure via logs • Verify the connection failure via packet capture.
Pass/Fail with Explanation	Pass, the TOE rejected the NULL connection and non-claimed cipher suites.

6.1.3.51 FCS_TLSS_EXT.1.1 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 3:** If RSA key exchange is used in one of the selected ciphersuites, the evaluator shall use a client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake. The evaluator shall verify that the handshake is not completed successfully and no application data flows.
Test Steps	• Use acumen-tlss tool to initiate a connection with a modified EncryptedPreMasterSecret field to the TOE and verifies that the connection attempt fails. • Verify the above with the help of wireshark capture. • Verify with the help of the system logs.
Pass/Fail with Explanation	Pass, TOE rejects the connection when a modified EncryptedPreMasterSecret field is used when initiated.

6.1.3.52 FCS_TLSS_EXT.1.1 TEST #4.1 **[TD0469]**

Item	Data
Test Assurance Activity	TD0469 has been applied which removed this test.
Pass/Fail with Explanation	TD0469 has been applied which removed this test. Hence, this test assurance activity has been met.

6.1.3.53 FCS_TLSS_EXT.1.1 TEST #4.2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: • **Test 4.2:** Modify a byte in the data of the client's Finished handshake message, and verify that the server rejects the connection and does not send any application data.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify the connection fails when a byte is modified in the client finished handshake and verify the connection fails. • Verify the connection failure using Wireshark. • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, the TOE rejects the connection after it receives the modified Client Finished handshake message.

6.1.3.54 FCS_TLSS_EXT.1.1 TEST #4.3I [TD0779] - NA

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: **Test 4.3:** Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption):

	Test 4.3i [conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test: a. The evaluator shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. b. The evaluator shall verify the server does not send a NewSessionTicket handshake message (at any point in the handshake). c. The evaluator shall verify the Server Hello message contains a zero-length session identifier or passes the following steps: Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID. d. The evaluator shall complete the TLS handshake and capture the SessionID from the ServerHello. e. The evaluator shall send a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d). f. The evaluator shall verify the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0779 has been applied.**
Pass/Fail with Explanation	NA. The TOE does support session resumption based on session IDs or session tickets.

6.1.3.55 FCS_TLSS_EXT.1.1 TEST #4.3II **[TD0779]**

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: **Test 4.3:** Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption):

	Test 4.3ii [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246). b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0779 has been applied.**
Test Steps	Positive connection: • Use Acumen-TLS tool to initiate a connection to the TOE with the same session ID and confirm success. • Verifying the connection is successful with the help of Wireshark. Negative connection: • Use Acumen-TLS tool to initiate a connection to the TOE with the same session ID after forcing an alert message and confirms the TOE issues a new session ID. • Verify the new connection with a new session ID with the help of Wireshark. • Confirm the session ID's with the help of TOE logs.
Pass/Fail with Explanation	Pass, The TOE successfully accepted a TLS connection request when a previously used session ID was reused. Additionally, upon disruption of an ongoing TLS connection, the TOE issued a new session ID and correctly handled a reconnection attempt using the session ID from the disrupted session.

6.1.3.56 FCS_TLSS_EXT.1.1 TEST #4.3III **[TD0779]**

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: **Test 4.3:** Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption):

	Test 4.3iii [conditional]: If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): *a)* The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE successfully resumes the session in accordance with section 3.1 of RFC 5077. *b)* The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) by implicitly rejects the session ticket performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data. *c)* The evaluator shall send the TSF a Client Hello with a SessionTicket extension, and observe the TSF responds with a Server Hello with an empty SessionTicket extension. The evaluator shall then send the TSF a invalid Finished message, and observe that the TSF terminates the session without sending a valid newTicket message. Note: if the TSF sends a newTicket message prior to terminating the session, the evaluator shall confirm the ticket is invalid by attempting to use the ticket to renew the session and observe that the TSF either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data. **TD0779 has been applied.**
Test Steps	Positive connection: • Use Acumen-TLS tool to initiate a connection to the TOE that will perform the evaluation activity a) based on session ticket. • Verifying the connection is successful with the help of Wireshark. New session ticket: • Use Acumen-TLS tool to initiate a connection to the TOE that will perform the evaluation activity b) based on session Ticket. • Verifying the connection is successful with the help of Wireshark. • Verify the session tickets with the help of TOE logs. Empty SessionTicket: • Use Acumen-TLS tool to initiate a TLS connection that will enable the TOE to send a empty session ticket. • Verify the connection failure with the help of Wireshark. • Verify the connection failure using TOE logs.
Pass/Fail with Explanation	Pass, TOE was successfully able to handle previous issued session ticket successfully, initiated a new connection when a modified session ticket is used and rejected the connection when an invalid finished message is used.

6.1.3.57 FCS_TLSS_EXT.1.1 TEST #4.4

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: • **Test 4.4:** Send a message consisting of random bytes from the client after the client has issued the ChangeCipherSpec message and verify that the server denies the connection.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify that the connection fails. • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, The TOE rejected the connection after it receives random bytes after ChangeCipherSpec message.

6.1.3.58 FCS_TLSS_EXT.1.2 TEST #1

Item	Data
Test Assurance Activity	**Test 1:** The evaluator shall send a Client Hello requesting a connection with version SSL 2.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 3.0 and TLS 1.0, and TLS 1.1 if it is selected.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify the connection fails for all the non-supported SSL (SSLv2.0, SSLv3.0) and TLS versions (TLSv1.0, TLSv1.1) • Verify the packet capture. • Verify the failure logs on the device.
Pass/Fail with Explanation	Pass, the TOE rejects and logs the SSL v2.0, SSL v3.0, TLS v1.0, TLS v1.1 connection attempts but accepts TLS v1.2 connection attempt.

6.1.3.59 FCS_TLSS_EXT.1.3 TEST #1 **[TD0739]**

Item	Data
Test Assurance Activity	The evaluator shall conduct the following tests. The testing can be carried out manually with a packet analyzer or with an automated framework that similarly captures such empirical evidence. Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. **Test 1:** [conditional] If RSA-based key establishment is selected, the evaluator shall configure the TOE with a certificate containing a supported RSA size and attempt a connection. The evaluator shall verify that the size used matches that which is configured and that the connection is successfully established. The evaluator shall repeat this test for each supported size of RSA-based key establishment. **TD0739 has been applied**

Test Steps	2048: • The evaluator configures the TOE with RSA-based certificate. • The evaluator then attempts a TLS connection with the TOE. • The above established connection is verified with the help of Wireshark. 3072: • The evaluator configures the TOE with RSA-based certificate. • The evaluator then attempts a TLS connection with the TOE. • The above established connection is verified with the help of Wireshark. 4096: • The evaluator configures the TOE with RSA-based certificate. • The evaluator then attempts a TLS connection with the TOE. • The above established connection is verified with the help of Wireshark.
Pass/Fail with Explanation	Pass, TLS connection is accepted when RSA based certificate is used.

6.1.3.60 FCS_TLSS_EXT.1.3 TEST #2 – NA

Item	Data
Test Assurance Activity	The evaluator shall conduct the following tests. The testing can be carried out manually with a packet analyzer or with an automated framework that similarly captures such empirical evidence. Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. **Test 2:** [conditional] If finite-field (i.e. non-EC) Diffie-Hellman ciphers are selected, the evaluator shall attempt a connection using a Diffie-Hellman key exchange with a supported parameter size or supported group. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported parameter size or group.
Pass/Fail with Explanation	N/A, as DHE not claimed by the TOE.

6.1.3.61 FCS_TLSS_EXT.1.3 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall conduct the following tests. The testing can be carried out manually with a packet analyzer or with an automated framework that similarly captures such empirical evidence. Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. **Test 3:** [conditional] If ECDHE ciphers are selected, the evaluator shall attempt a connection using an ECDHE ciphersuite with a supported curve. The evaluator shall verify that the key

	agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported elliptic curve.
Test Steps	• Initiate a connection with the TOE over TLS using the curve secp256r1, secp384r1, secp521r1 and verify the connection succeeds. • Verify with packet capture.
Pass/Fail with Explanation	Pass, the TOE accepts all the supported curves connections.

6.1.3.62 FCS_TLSS_EXT.2.2 TEST #1 **[TD0770]**

Item	Data
Test Assurance Activity	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity. If error messages are provided prior to terminating a session, the evaluator shall configure the test client to be able to observe application data sent over the non-mutually authenticated channel, and in addition to observing that the TSF terminates the session, that the only data received under the channel is the error message as described in the TSS: **Test 1:** The evaluator shall configure the server to send a certificate request to the client. The client shall send a certificate_list structure which has a length of zero. The evaluator shall verify that no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated. **TD0770 has been applied.**
Test Steps	• Use Acumen-tlss tool to initiate a connection to the TOE with a certificate_list structure which has a length of zero and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection is not established when no certificate is returned to the TOE.

6.1.3.63 FCS_TLSS_EXT.2.2 TEST #2 **[TD0770]**

Item	Data
Test Assurance Activity	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall

	apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity. If error messages are provided prior to terminating a session, the evaluator shall configure the test client to be able to observe application data sent over the non-mutually authenticated channel, and in addition to observing that the TSF terminates the session, that the only data received under the channel is the error message as described in the TSS: **Test 2:** The evaluator shall configure the server to send a certificate request to the client. The client shall send no client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. The client is required to respond to the certificate request message, even if the certificate message is empty. The evaluator shall verify that the handshake is not finished successfully and no application data flows. **TD0770 has been applied.**
Test Steps	• Use Acumen-tlss tool to initiate a connection to the TOE without client certificate and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection is not established when no certificate is shared with the server.

6.1.3.64 FCS_TLSS_EXT.2.2 TEST #3 **[TD0770]**

Item	Data
Test Assurance Activity	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity. If error messages are provided prior to terminating a session, the evaluator shall configure the test client to be able to observe application data sent over the non-mutually authenticated channel, and in addition to observing that the TSF terminates the session, that the only data received under the channel is the error message as described in the TSS: **Test 3:** The evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated.

	TD0770 has been applied.
Test Steps	• Use Acumen-tlss tool to initiate a connection to the TOE with the unsupported_signature_algorithm and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejects the connection when an unsupported signature algorithm is used.

6.1.3.65 FCS_TLSS_EXT.2.2 TEST #4 **[TD0770]**

Item	Data
Test Assurance Activity	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity. If error messages are provided prior to terminating a session, the evaluator shall configure the test client to be able to observe application data sent over the non-mutually authenticated channel, and in addition to observing that the TSF terminates the session, that the only data received under the channel is the error message as described in the TSS: **Test 4:** The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, load the modified certificate path, and verify that no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated. **TD0770 has been applied.**
Test Steps	• The evaluator shall initiate a check-in from the device and confirm this to be successful. • The evaluator shall confirm the check-in success with the help of Wireshark capture. • Create an invalid chain of certificates to connect to the TOE. • Use Acumen-tlss tool to initiate a connection to the TOE without a valid certification path and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE accepts connection when valid certificate path is presented and rejects connection when invalid certificate path is presented.

6.1.3.66 FCS_TLSS_EXT.2.2 TEST #5 **[TD0770]**

Item	Data
Test Assurance Activity	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity. If error messages are provided prior to terminating a session, the evaluator shall configure the test client to be able to observe application data sent over the non-mutually authenticated channel, and in addition to observing that the TSF terminates the session, that the only data received under the channel is the error message as described in the TSS: **Test 5:** The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognized by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not in fact correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not in fact terminate in the claimed CA certificate). The evaluator shall verify that no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated. **TD0770 has been applied.**
Test Steps	• TOE CA details • Create a CA certificate whose CN matches with the CA certificate on the TOE but with different key. • Use Acumen-tlss tool to initiate a connection to the TOE with a client certificate signed by impostor CA and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TOE rejects connection when a client certificate is signed by an imposter CA.

6.1.3.67 FCS_TLSS_EXT.2.2 TEST #6 **[TD0770]**

Item	Data
Test Assurance Activity	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity.

	If error messages are provided prior to terminating a session, the evaluator shall configure the test client to be able to observe application data sent over the non-mutually authenticated channel, and in addition to observing that the TSF terminates the session, that the only data received under the channel is the error message as described in the TSS: **Test 6:** The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated. Ideally, the two certificates should be identical except for the Client Authentication purpose. **TD0770 has been applied.**
Test Steps	• The evaluator verifies the device certificate configured on the EPMM server. • The evaluator attempts to check-in from the mobile device and confirms that the operation is successful. • The evaluator confirms that the connection was successful and the presence of client authentication purpose in the client's certificate. • Create a server certificate without the Client Authentication in the extendedKeyUsage field • Use Acumen-tlss tool to initiate a connection to the TOE with a client certificate without the Client Authentication purpose and confirm the connection is a failure. • Verify the connection failure via packet capture • Verify the connection failure is via logs
Pass/Fail with Explanation	Pass, TOE accepted connection when client certificate has the correct extended key usage field and rejected when client certificate has incorrect extended key usage field.

6.1.3.68 FCS_TLSS_EXT.2.2 TEST #7 **[TD0770]**

Item	Data
Test Assurance Activity	The evaluator shall use TLS as a function to verify that the validation rules in FIA_X509_EXT.1.1 are adhered to and shall perform the following tests. The evaluator shall apply the AGD guidance to configure the server to require TLS mutual authentication of clients for the following tests, unless overridden by instructions in the test activity. If error messages are provided prior to terminating a session, the evaluator shall configure the test client to be able to observe application data sent over the non-mutually authenticated channel, and in addition to observing that the TSF terminates the session, that the only data received under the channel is the error message as described in the TSS:

	Test 7: The evaluator shall perform the following modifications to the traffic: a) Configure the server to require mutual authentication and then modify a byte in the client's certificate. The evaluator shall verify that the server rejects the connection. B) Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message. The evaluator shall verify that the server rejects the connection. **TD0770 has been applied.**
Test Steps	A: • Use Acumen-tlss tool to initiate a connection to the TOE with a modified client certificate resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs **B:** • Use Acumen-tlss tool to initiate a connection to the TOE with a client's modified signature block resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Pass/Fail with Explanation	Pass, TOE rejects connection when a modified client certificate and signature block is used.

6.1.3.69 FCS_TLSS_EXT.2.3 TEST #1

Item	Data
Test Assurance Activity	**Test 1:** The evaluator shall send a client certificate with an identifier that does not match any of the expected identifiers and verify that the server denies the connection. The matching itself might be performed outside the TOE (e.g. when passing the certificate on to a directory server for comparison).
Test Steps	• The evaluator verifies the current device enrolled in the MDM server and verifies the device certificate. • Attempt a check-in from the mobile device and verify success. • Verify the check-in process with the help of Wireshark. • Take a snapshot of the VM with the current enrolled device. • Wipe the device and verify success. • Re-enroll device with MDM server and verify success and confirm the device UUID and certificate details. • Restore the MDM server to the older snapshot that was performed in the previous step. • Once the MDM server is restored, confirm the device details enrolled in the MDM server. • Now attempt a check-in from the MDM agent that was enrolled latest and verify failure. • Confirm the connection failure with the help of Wireshark and TOE logs.

Pass/Fail with Explanation	Pass, TOE rejects the connection when a client certificate that does not match the reference identifier is used to make connection and the TOE generates a HTTP error code of 401(Unauthorized).

6.1.4 SYSTEM MANAGER TLSS TEST CASES

6.1.4.1 FCS_TLSS_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 1:** The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	• Establish a connection with the TOE over TLS using TLS_RSA_WITH_AES_128_GCM_SHA256 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using TLS_RSA_WITH_AES_256_GCM_SHA256 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 and show the connection being successful. • Verify the packet capture. • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 and show the connection being successful. • Verify the packet capture.
Pass/Fail with Explanation	Pass, successful connection over TLS using TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_256_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

6.1.4.2 FCS_TLSS_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 2:** The evaluator shall send a Client Hello to the server with a list of cipher suites that does not contain any of the cipher suites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the server denies the connection.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify that the connection fails with the non-supported cipher suite ○ RSA_WITH_3DES_EDE_CBC_SHA ○ NULL_WITH_NULL_NULL • Verify the connection failure via logs • Verify the connection failure via packet capture.
Pass/Fail with Explanation	Pass, the TOE will reject the NULL connection and non-claimed cipher suites.

6.1.4.3 FCS_TLSS_EXT.1.1 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 3:** If RSA key exchange is used in one of the selected ciphersuites, the evaluator shall use a client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake. The evaluator shall verify that the handshake is not completed successfully and no application data flows.
Test Steps	• Use acumen-tlss tool to initiate a connection with a modified EncryptedPreMasterSecret field tothe TOE and verifies that the connection attempt fails. • Verify the above with the help of wireshark capture. • Verify with help of the system logs.
Pass/Fail with Explanation	Pass, TLS connection attempt fails when a modified EncryptedPreMasterSecret field is used when initiated.

6.1.4.4 FCS_TLSS_EXT.1.1 TEST #4.1 **[TD0469]**

Item	Data
Test Assurance Activity	TD0469 has been applied which removed this test.
Pass/Fail with Explanation	TD0469 has been applied which removed this test. Hence, this test assurance activity has been met.

6.1.4.5 FCS_TLSS_EXT.1.1 TEST #4.2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: • **Test 4.2:** Modify a byte in the data of the client's Finished handshake message, and verify that the server rejects the connection and does not send any application data.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify the connection fails when a byte is modified in the client finished handshake and verify the connection fails. • Verify the connection failure using Wireshark. • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, the TOE rejects the connection after it receives the modified Client Finished handshake message.

6.1.4.6 FCS_TLSS_EXT.1.1 TEST #4.3I [TD0779] - NA

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: **Test 4.3:** Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption): **Test 4.3i [conditional]:** If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test: b. The evaluator shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. c. The evaluator shall verify the server does not send a NewSessionTicket handshake message (at any point in the handshake).

	d. The evaluator shall verify the Server Hello message contains a zero-length session identifier or passes the following steps: Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID. e. The evaluator shall complete the TLS handshake and capture the SessionID from the ServerHello. f. The evaluator shall send a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d). g. The evaluator shall verify the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0779 has been applied.**
Pass/Fail with Explanation	NA. The TOE does support session resumption based on session IDs or session tickets.

6.1.4.7 FCS_TLSS_EXT.1.1 TEST #4.3II **[TD0779]**

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: **Test 4.3:** Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption): **Test 4.3ii [conditional]**: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): c) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246).

	d) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0779 has been applied.**
Test Steps	Positive connection: • Use Acumen-TLS tool to initiate a connection to the TOE that will perform the evaluation activity a) based on session ID. • Verifying the connection is successful with the help of Wireshark. Negative connection: • Use Acumen-TLS tool to initiate a connection to the TOE that will perform the evaluation activity b) based on session ID. • Verify the failure of connection with the help of Wireshark. Confirm the connection failure with the help of TOE logs.
Pass/Fail with Explanation	Pass, the TOE accepted TLS connection request when a session ID is reused and a new session ID was provided by TOE when the ongoing connection is disrupted and provided with the session ID of the disrupted TLS session.

6.1.4.8 FCS_TLSS_EXT.1.1 TEST #4.3III **[TD0779]**

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: **Test 4.3:** Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption): **Test 4.3iii [conditional]:** If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): *d)* The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE successfully resumes the session in accordance with section 3.1 of RFC 5077. *e)* The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a

	full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data. *f)* The evaluator shall send the TSF a Client Hello with a SessionTicket extension, and observe the TSF responds with a Server Hello with an empty SessionTicket extension. The evaluator shall then send the TSF a invalid Finished message, and observe that the TSF terminates the session without sending a valid newTicket message. Note: if the TSF sends a newTicket message prior to terminating the session, the evaluator shall confirm the ticket is invalid by attempting to use the ticket to renew the session and observe that the TSF either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data. **TD0779 has been applied.**
Test Steps	Positive connection: • Use Acumen-TLS tool to initiate a connection to the TOE that will perform the evaluation activity a) based on session ticket. • Verifying the connection is successful with the help of Wireshark. New session ticket: • Use Acumen-TLS tool to initiate a connection to the TOE that will perform the evaluation activity b) based on session Ticket. • Verifying the connection is successful with the help of Wireshark. • Verify the connection failure with the help of TOE logs. Empty SessionTicket: • Use Acumen-TLS tool to initiate a TLS connection that will enable the TOE to send a empty session ticket. • Verify the connection failure with the help of Wireshark. Verify the connection failure using TOE logs.
Pass/Fail with Explanation	Pass. The TOE demonstrates the correct use of session tickets.

6.1.4.9 FCS_TLSS_EXT.1.1 TEST #4.4

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall perform the following modifications to the traffic: • **Test 4.4:** Send a message consisting of random bytes from the client after the client has issued the ChangeCipherSpec message and verify that the server denies the connection.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify that the connection fails. • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, The TOE rejected the connection after it receives random bytes after ChangeCipherSpec message.

6.1.4.10 FCS_TLSS_EXT.1.2 TEST #1

Item	Data
Test Assurance Activity	**Test 1:** The evaluator shall send a Client Hello requesting a connection with version SSL 2.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 3.0 and TLS 1.0, and TLS 1.1 if it is selected.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify the connection fails for all the non-supported SSL (SSLv2.0, SSLv3.0) and TLS versions (TLSv1.0, TLSv1.1) • Verify the packet capture. • Verify the failure logs on the device.
Pass/Fail with Explanation	Pass, the TOE rejects and logs the SSL v2.0, SSL v3.0, TLS v1.0, TLS v1.1 connection attempts but accepts TLS v1.2 connection attempt.

6.1.4.11 FCS_TLSS_EXT.1.3 TEST #1 **[TD0739]**

Item	Data
Test Assurance Activity	The evaluator shall conduct the following tests. The testing can be carried out manually with a packet analyzer or with an automated framework that similarly captures such empirical evidence. Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. **Test 1:** [conditional] If RSA-based key establishment is selected, the evaluator shall configure the TOE with a certificate containing a supported RSA size and attempt a connection. The evaluator shall verify that the size used matches that which is configured and that the connection is successfully established. The evaluator shall repeat this test for each supported size of RSA-based key establishment. **TD0739 has been applied**
Test Steps	2048: • The evaluator configures the TOE with RSA-based certificate. • The evaluator then attempts a TLS connection with the TOE. • The above established connection is verified with the help of Wireshark. 3072: • The evaluator configures the TOE with RSA-based certificate. • The evaluator then attempts a TLS connection with the TOE. • The above established connection is verified with the help of Wireshark. 4096: • The evaluator configures the TOE with RSA-based certificate. • The evaluator then attempts a TLS connection with the TOE.

	• The above established connection is verified with the help of Wireshark.
Pass/Fail with Explanation	Pass, TLS connection is established when RSA based certificate is used.

6.1.4.12 FCS_TLSS_EXT.1.3 TEST #2 – NA

Item	Data
Test Assurance Activity	The evaluator shall conduct the following tests. The testing can be carried out manually with a packet analyzer or with an automated framework that similarly captures such empirical evidence. Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. **Test 2:** [conditional] If finite-field (i.e. non-EC) Diffie-Hellman ciphers are selected, the evaluator shall attempt a connection using a Diffie-Hellman key exchange with a supported parameter size or supported group. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported parameter size or group.
Pass/Fail with Explanation	N/A, as DHE not claimed by the TOE.

6.1.4.13 FCS_TLSS_EXT.1.3 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall conduct the following tests. The testing can be carried out manually with a packet analyzer or with an automated framework that similarly captures such empirical evidence. Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. **Test 3:** [conditional] If ECDHE ciphers are selected, the evaluator shall attempt a connection using an ECDHE ciphersuite with a supported curve. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported elliptic curve.
Test Steps	• Initiate a connection with the TOE over TLS using the curve secp256r1, secp384r1, secp521r1 and verify the connection succeeds. • Verify with packet capture.
Pass/Fail with Explanation	Pass, the TOE accepts all the supported curves connections.

6.1.5 MANAGEMENT FUNCTIONS TEST CASES

6.1.5.1 FMT_MOF.1.1(1) TEST #1

Item	Data
Test Assurance Activity	**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.
Test Steps	• The evaluator shall login to the TOE as an unauthorized user. • The evaluator confirms that access to the TOE management portal was rejected.
Pass/Fail with Explanation	Pass, the evaluator was not able to access the admin portal when attempted access as an unauthorized user.

6.1.5.2 FMT_MOF.1.1(1) TEST #2

Item	Data
Test Assurance Activity	**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.
Test Steps	• The evaluator shall login to the TOE as an unauthorized user. • The evaluator confirms that access to the TOE management portal was rejected.
Pass/Fail with Explanation	Pass, the evaluator was not able to access the admin portal when attempted access as an unauthorized user.

6.1.5.3 FMT_MOF.1.1(2) TEST #1

Item	Data
Test Assurance Activity	The test of this function is performed in conjunction with FIA_ENR_EXT.1.
Test Steps	See FIA_ENR_EXT.1.
Pass/Fail with Explanation	Pass, the evaluator was able to enroll the mobile device successfully. This testing requirement is covered as part of FIA_ENR.EXT.1

6.1.5.4 FMT_MOF.1.1(3) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Ensure that the mobile device is enrolled in the MDM Server and is in a compliant state.

	• Verify that the application cannot be downloaded from the MAS Server prior to enrollment. • Partially enroll the device so that it is connected to the MDM Server, but ensure that the device is not in a compliant state. • Attempt to download an application in a connected but non-compliant state, verify that the action cannot be completed due to the partial enrollment.
Pass/Fail with Explanation	Pass, the testing activity is not possible as partial enrollment is not feasible with the Device-Admin enrollment method which is the only enrollment method we are evaluating the TOE.

6.1.5.5 FMT_POL_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Attempt a policy update in accordance with FMT_SMF.1(1). • Verify that the policy is updated in either the MDM Server, in transmission, or at the MDM agent. • Verify that the TSF signs the update and provides it to the MDM Agent.
Pass/Fail with Explanation	Pass, Policy updates are successfully performed on the device.

6.1.5.6 FMT_SMF.1.1(1) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	The MDM Server shall be capable of communicating the following commands to the MDM Agent: 1. transition to the locked state (MDF Function 6) 2. full wipe of protected data (MDF Function 7) 3. unenroll from management 4. install policies 5. query connectivity status 6. query the current version of the MD firmware/software 7. query the current version of the hardware model of the device 8. query the current version of installed mobile applications 9. import X.509v3 certificates into the Trust Anchor Database (MDF Function 11) 10. install applications (MDF Function 16) 11. update system software (MDF Function 15) 12. remove applications (MDF Function 14)

	and the following commands to the MDM Agent: [• no other management functions] and the following MD configuration policies: 25. password policy: 1. minimum password length 2. minimum password complexity 3. maximum password lifetime (MDF Function 1) 26. session locking policy: 1. screen-lock enabled/disabled 2. screen lock timeout 3. number of authentication failures (MDF Function 2) 27. wireless networks (SSIDs) to which the MD may connect (MDF Function 2) 28. security policy for each wireless network: 1. [▪ specify the FQDN(s) of acceptable WLAN authentication server certificate(s),] 2. ability to specify security type 3. ability to specify authentication protocol 4. specify the client credentials to be used for authentication 5. [no additional WLAN management functions] (WLAN Client Function 1) 29. application installation policy by [○ specifying authorized application repository(s),], (MDF Function 8) 30. enable/disable policy for [camera and microphone] across device, [• no other method], (MDF Function 5)
Pass/Fail with Explanation	Pass, the evaluator was able to perform all the management functions claimed by the TOE.

6.1.5.7 FMT_SMF.1.1(2) TEST #1

Item	Data
Test Assurance Activity	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)).
Test Steps	The evaluator shall demonstrate that the MDM Server is capable of the following: 1. The evaluator shall upload/import certificate that will be used by the TOE for connection establishment and verify its success.

	2. The evaluator then attempts to establish a connection with the TOE and verifies the TOE using the certificate uploaded in the above step for trusted connection. 3. The above connection is verified with the help of WireShark.
Pass/Fail with Explanation	Pass, the TOE uses the uploaded x509 certificates for its trusted established connections.

6.1.5.8 FMT_SMF.1.1(2) TEST #2

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator configures sync policy for different time intervals. • The evaluator verifies that MDM server was able to successfully apply the sync policy received a response from MDM agent as configured.
Pass/Fail with Explanation	Pass, the MDM agent was able to communicate with the server for the configured periodicity.

6.1.5.9 FMT_SMF.1.1(2) TEST #3

Item	Data
Test Assurance Activity	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.
Test Steps	The evaluator shall demonstrate that the MDM Server is capable of the following: • The evaluator shall attempt to enroll the device. • The evaluator verifies that after successful enrollment the device enrolled has a unique ID. • Shall limit the number of devices per enrollment.
Pass/Fail with Explanation	Pass, all of the required MDM server functionality were performed successfully when executed.

6.1.5.10 FMT_SMF.1.1(3) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Create a new user group. • The evaluator verified that the MAS client is excluded from the newly created group. • Create and verify that the application accessible to that group cannot be accessed by the excluded MAS client. • Include the MAS client user in the group. • Verify that the application can now be accessed by the MAS client.
Pass/Fail with Explanation	Pass, The MAS client can only access the applications specified for the group they are enrolled in.

6.1.5.11 FMT_SMR.1.2(1) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	Please refer to FMT_SMF.1.1(1) and FIA_ENR_EXT.1 for this test case.
Pass/Fail with Explanation	Pass, the evaluator was able to administer the TOE via available interfaces. This testing requirement is fulfilled by the test cases FMT_SMF.1.1(1) and FIA_ENR_EXT.1.

6.1.5.12 FMT_SMR.1.2(2) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	Please refer to FMT_SMF.1.1(1) and FIA_ENR_EXT.1 for this test case.
Pass/Fail with Explanation	Pass, the evaluator was able to administer the TOE via available interfaces. This testing requirement is fulfilled by the test cases FMT_SMF.1.1(1) and FIA_ENR_EXT.1.

6.1.6 DATA TRANSFER TEST CASES

6.1.6.1 FPT_API_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.6.2 FPT_LIB_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.6.3 FPT_ITT.1.1(2) TEST #1

Item	Data
Test Assurance Activity	**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.
Test Steps	• The evaluator confirms the SSL certificate used by the TOE server. • The evaluator then attempts to enroll the MDM agent with the server. • Verify the successful device enrollment with the help of server logs. • Finally, the evaluator confirms the enrollment was successfully established over a mutually authenticated TLS connection with the help of network capture.
Pass/Fail with Explanation	Pass, the data between the MDM server and MDM agent is transmitted in TLS as selected in the ST.

6.1.6.4 FPT_ITT.1.1(2) TEST #2

Item	Data
Test Assurance Activity	**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.
Test Steps	• The evaluator attempts to perform some random operations from the mobile device.

	• Finally, the evaluator confirms the data transmitted is not transmitted between the server and the agent is not in plain text.
Pass/Fail with Explanation	Pass, the TLS channel data is not transmitted in plain text between the MDM server and MDM Agent

6.1.7 UPDATE TEST CASES

6.1.7.1 FPT_TST_EXT.1.2 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Perform a reboot the TOE • Upon reboot verify the integrity of stored TSF executable code.
Pass/Fail with Explanation	Pass, the TOE performs an integrity check when a reboot is issued.

6.1.7.2 FPT_TST_EXT.1.2 TEST #2

Item	Data
Test Assurance Activity	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.
Test Steps	• Modify the TSF executable • Perform a reboot • Verify the integrity check fails during the reboot
Pass/Fail with Explanation	Pass, TOE's executable was modified and integrity check failed as the TOE was rebooted.

6.1.7.3 FPT_TUD_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	Server: • Query the current version of the TOE • Verify the displayed version matches the documented and installed version Client: • Query the current version of the TOE

	• Verify the displayed version matches the documented and installed version
Pass/Fail with Explanation	Pass, the current installed version of the TOE matches with the documents.

6.1.7.4 FPT_TUD_EXT.1.2 TEST #2

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.7.5 FPT_TUD_EXT.1.3 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	Server: • Attempt an update of the TOE. • Verify that the update is successful by showing the newly installed version. Client: • Attempt an update of the TOE. • Verify that the update is successful by showing the newly installed version.
Pass/Fail with Explanation	Pass, the evaluator was able to successfully upgrade the TOE to the latest version.

6.1.7.6 FPT_TUD_EXT.1.3 TEST #2

Item	Data
Test Assurance Activity	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.
Test Steps	Server: • Attempt an update of the TOE with an incorrect digital signature • Verify that the update attempt fails Client: • Attempt an update of the TOE with an incorrect digital signature • Verify that the update attempt fails

Pass/Fail with Explanation	Pass, The TOE does not proceed with system upgrade when a image not digitally signed by vendor is provided.

6.1.8 TRUSTED CHANNELS TEST CASES

6.1.8.1 FCO_CPC_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.2 FCO_CPC_EXT.1.2 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.3 FCO_CPC_EXT.1.3 TEST #1

Item	Data
Test Assurance Activity	**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)
Pass/Fail with Explanation	Pass, An IT entity cannot communicate with the TOE unless it is registered with the TOE by admin provided credentials.

6.1.8.4 FCO_CPC_EXT.1.3 TEST #2 **[TD0594]**

Item	Data
Test Assurance Activity	**Test 2:** 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. **TD0594 has been applied.**

Pass/Fail with Explanation	Pass, refer to test “FAU_ALT_EXT.1.1 Test #1” registration process will enable the IT entity to communicate only with the TOE component.

6.1.8.5 FCO_CPC_EXT.1.3 TEST #3 **[TD0594]**

Item	Data
Test Assurance Activity	**Test 3:** 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. **TD0594 has been applied.**
Pass/Fail with Explanation	Pass, refer to test “FAU_ALT_EXT.1.1 Test #1” wipe process will enable the IT entity to disable communication with the TOE component.

6.1.8.6 FTP_ITC_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	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).
Pass/Fail with Explanation	Pass. This testing is completed in conjunction with the testing for FPT_ITT.1(2).

6.1.8.7 FTP_ITC.1.1(1) TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.8 FTP_ITC.1.2(1) TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.9 FTP_ITC.1.3(1) TEST #1

Item	Data
Test Assurance Activity	**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.
Test Steps	Audit Server: • Attempt to configure the audit server. • verify that the successful connection is established via OpenSSL. • Confirm the established connection with network capture.
Pass/Fail with Explanation	Pass, the communication channel was able to be established post configuration as per administrative guide.

6.1.8.10 FTP_ITC.1.3(1) TEST #2

Item	Data
Test Assurance Activity	**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.
Test Steps	Audit Server: • Attempt to configure the device. • verify that the connection is successfully established. • Confirm that data is not sent in plaintext with network capture.
Pass/Fail with Explanation	Pass, the data is not transmitted in plain text. Refer to FTP_ITC.1.3(1) Test#1

6.1.8.11 FTP_ITC.1.3(1) TEST #3

Item	Data
Test Assurance Activity	**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.
Test Steps	Audit Server: • Attempt to configure the device. • verify that the connection is successfully established. • Confirm the established connection is in TLS v1.2 with network capture.
Pass/Fail with Explanation	Pass, all the communication that takes place is in TLS v1.2. Refer to FTP_ITC.1.3(1) Test#1

6.1.8.12 FTP_ITC.1.1(2) TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.13 FTP_ITC.1.2(2) TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.14 FTP_ITC.1.3(2) TEST #1

Item	Data
Test Assurance Activity	**Test 1:** The evaluators shall ensure that communications using each specified (in the operational guidance) Agent-Server 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.
Test Steps	See FTP_ITC.1.3(1) Test 1
Pass/Fail with Explanation	Pass, the communication channel was able to be established post configuration as per administrative guide. This testing is satisfied by FTP_ITC.1.3(1) Test 1

6.1.8.15 FTP_ITC.1.3(2) TEST #2

Item	Data
Test Assurance Activity	**Test 2:** The evaluator shall ensure, for each method of Agent-Server communication, the channel data is not sent in plaintext. Further evaluation activities are associated with the specific protocols.
Test Steps	See FTP_ITC.1.3(1) Test 2
Pass/Fail with Explanation	Pass, the data is not transmitted in plain text. This testing is satisfied by FTP_ITC.1.3(1) Test 1

6.1.8.16 FTP_ITC.1.3(2) TEST #3

Item	Data
Test Assurance Activity	**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.
Test Steps	See FTP_ITC.1.3(1) Test 3
Pass/Fail with Explanation	Pass, the data is transmitted in TLS v1.2. This testing is satisfied by FTP_ITC.1.3(1) Test 1

6.1.8.17 FTP_TRP.1.1(1) TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.18 FTP_TRP.1.2(1) TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.19 FTP_TRP.1.3(1) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	Admin Portal: • Start a web-based session and log into the MDM Server • Verify the successful HTTPS/TLS channel System Manager Portal: • Start a web-based session and log into the MDM Server • Verify the successful HTTPS/TLS channel
Pass/Fail with Explanation	Pass, the remote communication channel established is in HTTPS/TLS.

6.1.8.20 FTP_TRP.1.3(1) TEST #2

Item	Data
Test Assurance Activity	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.
Test Steps	• Start a web-based session and log into the MDM Server • Verify the only available interface from the MDM server is the expected HTTPS/TLS channel
Pass/Fail with Explanation	Pass, the communication channel established is in HTTPS/TLS. This testing is satisfied by FTP_TRP.1.3(1) Test#1.

6.1.8.21 FTP_TRP.1.3(1) TEST #3

Item	Data
Test Assurance Activity	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.
Test Steps	• Start a web-based session and log into the MDM Server • Verify the successful HTTPS/TLS channel is encrypted
Pass/Fail with Explanation	Pass, the communication channel established is in HTTPS/TLS and no data is transmitted in plain-text. This testing is satisfied by FTP_TRP.1.3(1) Test#1

6.1.8.22 FTP_TRP.1.1(2) TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.23 FTP_TRP.1.2(2) TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.1.8.24 FTP_TRP.1.3(2) TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Attempt to enroll the MDM Agent to the server in device owner-mode. • Verify that a successful TLS connection happens as the device is enrolled
Pass/Fail with Explanation	Pass, the TLS communication channel is successful between the MDM Server and the MDM Agent during the enrollment process

6.1.8.25 FTP_TRP.1.3(2) TEST #2

Item	Data
Test Assurance Activity	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.
Test Steps	
Pass/Fail with Explanation	Pass, the TLS communication channel is successful between the MDM Server and the MDM Agent during the enrollment process with the communication transferred not in plain-text and operated over TLS v1.2. This testing is satisfied by the FTP_TRP.1(2) Test 1.

6.1.8.26 FTP_TRP.1.3(2) TEST #3

Item	Data
Test Assurance Activity	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.
Test Steps	

Pass/Fail with Explanation	Pass, the TLS communication channel is successful between the MDM Server and the MDM Agent during the enrollment process with the communication transferred not in plain-text and operated over TLS v1.2. This testing is satisfied by the FTP_TRP.1(2) Test 1.

6.1.8.27

6.2 GALAXY S22 ULTRA 5G RUNNING ANDROID 13

6.2.1 MDM AGENT TEST CASES

6.2.1.1 FCS_STG_EXT.1(2) TEST #1 - NA

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.2.1.2 FAU_ALT_EXT.2 TEST #1

Item	Data
Test Assurance Activity	**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.
Test Steps	• Create a policy update • Push the update to the MDM Agent • Verify that the MDM Agent has accepted the update and applied the policy • Verify that the MDM Server successfully received acknowledgement about the policy update
Pass/Fail with Explanation	Pass. The MDM Server successfully pushes policy updates accepted by the MDM Agent.

6.2.1.3 FAU_ALT_EXT.2 TEST #2

Item	Data
Test Assurance Activity	**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.
Test Steps	• Create a policy update to the enrolled MDM Agent

	• Push the update to the MDM Agent • Verify that the MDM Agent has accepted the update and applied the policy • Verify that the MDM Server receives an alert from the MDM Agent for the successful policy update • Ensure the MDM Server and MDM Agent are connected • The MDM Server forces a check-in from the MDM server to determine reachability. • Verify that the check-in process was successful. • Attempt to check in from the MDM agent • Verify that check-in was successful on the MDM server. • Attempt to wipe the MDM Agent from the MDM server. • Verify that the MDM Server receives an alert from the MDM Agent about the status of the wipe operation.
Pass/Fail with Explanation	Pass. The MDM Agent alerts the MDM Server for the required events.

6.2.1.4 FAU_ALT_EXT.2 TEST #3

Item	Data
Test Assurance Activity	**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.
Test Steps	• Verify the MDM Agent and the MDM Server are connected. • Trigger a forced check-in on the MDM Server to determine if there is an active connection between the MDM Agent and the MDM Server • Verify the result on the MDM Server. • Disconnect the MDM Agent from the MDM Server and trigger a forced check-in on the MDM Server to determine the status of the connection • Reconnect the device to see the queued alert from the MDM Server
Pass/Fail with Explanation	Pass. MDM Agent successfully performs a network reachability test while both in a connected and disconnected state.

6.2.1.5 FAU_ALT_EXT.2 TEST #4

Item	Data
Test Assurance Activity	**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.
Test Steps	• Disconnect the MDM Agent from the network. • Trigger a forced check-in on the MDM Server to determine the status of the connection • Reconnect the device to the network and see the queued action from the MDM Server post connection is established.
Pass/Fail with Explanation	Pass. Please refer to FAU_ALT_EXT.2 Test #3.

6.2.1.6 FAU_GEN.1(2)/AGT TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator performs operations on the TOE that will generate auditable events. • The evaluator then records the events that were generated as part of the above step and confirms with the output specified in the TSS.
Pass/Fail with Explanation	Pass, TOE generates audit events as expected.

6.2.1.7 FAU_SEL.1(2) TEST #1

Item	Data
Test Assurance Activity	**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.
Test Steps	• Create policy that will enable selective attribute audit logging and apply the policy. • Perform operations that will generate auditable events. • Verify the generated auditable event with the help of TOE logs.
Pass/Fail with Explanation	Pass, the audit data were successfully generated by the TOE for the audit operation performed.

6.2.1.8 FAU_SEL.1(2) TEST #2 - NA

Item	Data
Test Assurance Activity	**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.
Pass/Fail with Explanation	NA, no other attributes are selected for the TOE.

6.2.1.9 FIA_ENR_EXT.2 TEST #1

Item	Data
Test Assurance Activity	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 theMDM Server and validate the MDM Server's certificate.
Test Steps	• The evaluator installs the MDM server's root certificate on the mobile device to establish trust. • Install the MDM client application in device-owner mode by using adb command via evaluator's laptop. • Attempt to enroll the MDM agent with the reference identifier of the MDM server. • Verify that the MDM Agent has stored the reference identifier of the MDM Server for future communication.
Pass/Fail with Explanation	Pass. The MDM Agent stores the reference identifier of the MDM Server during initial enrollment and its used for future communication.

6.2.1.10 FMT_POL_EXT.2 TEST #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Verify the agent is configured on the available configuration interface. • Create a policy update on the MDM Server. • Push the policy update to the MDM Agent. • Verify that the MDM Agent accepts the policy update.
Pass/Fail with Explanation	Pass. MDM agent successfully applies the digitally signed policy update from the MDM Server.

6.2.1.11 FMT_POL_EXT.2 TEST #2

Item	Data
Test Assurance Activity	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 theMDM Agent. The evaluator shall verify the MDM Agent does not accept the digitally signed policy.
Test Steps	• The evaluator makes modifications to the config file so no signature for MDM agent • Create a policy update and push the policy to the mobile device • Verify the policy application failure with the help of system logs

	• The evaluator makes modifications to the config file the policies are incorrectly signed for MDM agent on server • Create a policy update and push the policy to the mobile device • Verify the policy application failure with the help of system logs
Pass/Fail with Explanation	Pass, policies that were not signed and incorrectly signed were rejected by the TOE.

6.2.1.12 FMT_SMF_EXT.4 TEST #1

Item	Data
Test Assurance Activity	**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.
Test Steps	
Pass/Fail with Explanation	Pass. refer to FMT_SMF_EXT.1.1 Test#1 in the MDM Base PP Test Plan, as all the selections were also tested in the mentioned SFR.

6.2.1.13 FMT_SMF_EXT.4 TEST #2 [TD0755]

Item	Data
Test Assurance Activity	**Test 2:** [conditional: if "import the certificates to be used for authentication of MDM Agent communications" is selected in FMT_SMF_EXT.4.1] 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). **TD0755 has been applied.**
Test Steps	• Configure the MDM Agents authentication certificate • Attempt to enroll the MDM Agent • Verify the MDM Agent uses the certificate during the enrollment process
Pass/Fail with Explanation	Pass. Please refer to test FIA_ENR_EXT 2 Test#1, The MDM agent uses the configured authentication certificate for all of its communication with the server

6.2.1.14 FMT_SMF_EXT.4 TEST #3

Item	Data
Test Assurance Activity	**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.

Pass/Fail with Explanation	Pass. Refer to test FIA_ENR_EXT.2 Test#1. The evaluator shows the successful enrollment of the device showing the MDM Agents ability to communicate the MDM Server.

6.2.1.15 FMT_SMF_EXT.4 TEST #4

Item	Data
Test Assurance Activity	**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.
Test Steps	For 15 mins: • Configure the periodicity of reachability events • Verify, the MDM Server receives alerts according to the schedule configured by the MDM Agent For 20 mins: • Configure the periodicity of reachability events. • Verify, the MDM Server receives alerts according to the schedule configured by the MDM Agent.
Pass/Fail with Explanation	Pass. The MDM agent was able to reach the MDM server based on configured sync/periodicity interval.

6.2.1.16 FMT_SMF_EXT.4 TEST #5

Item	Data
Test Assurance Activity	**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.
Pass/Fail with Explanation	Pass. Please refer to test FMT_SMF_EXT.4 Test#4. The MDM agent was able to reach the MDM server based on configured sync/periodicity interval that is performed with the help of sync policy.

6.2.1.17 FMT_UNR_EXT.1 TEST #1

Item	Data
Test Assurance Activity	**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.
Test Steps	• Configure the MDM agent in device-owner mode. • Verify the MDM server and MDM agent are connected and attempt to unenroll the MDM Agent • Verify that the attempt fails
Pass/Fail with Explanation	Pass. The evaluator was not able to retire the device when enrolled in device-owner mode.

6.2.1.18 FMT_UNR_EXT.1 TEST #2 – NA

Item	Data
Test Assurance Activity	**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.
Pass/Fail with Explanation	N/A, there is no "apply remediation actions" section made for the TOE.

6.2.2 AUTH TEST CASES

6.2.2.1 FIA_X509_EXT.1.1(1) TEST #1 **[TD0641]**

Item	Data
Test Assurance Activity	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. **TD0641 has been applied.**
Test Steps	• Establish a certificate path in which one of the issuing certificates is not a CA certificate: Refer to : By setting the basicConstraints field in an issuing certificate to have CA=False. • By omitting the basicConstraints field in one of the issuing certificates. ○ Configure an ICA certificate lacking the basicConstraints extension ○ Use OpenSSL tool to initiate a connection to the TOE with the chain does not contain the basicConstraints extension and verify the connections failed ○ Verify the connection failure via packet capture ○ Verify the connection failure via logs • By setting the basicConstraints field in an issuing certificate to have CA=False. ○ The evaluator manipulates the ICA certificate with the flag in the basicConstraints extension set to FALSE ○ Use the Openssl tool to initiate a connection to the TOE with the chain with the basicConstraints set to FALSE and verify the connections failed. ○ Verify the connection failure via packet capture ○ Verify the connection failure via logs • By omitting the CA signing bit of the key usage field in an issuing certificate. ○ Configure an ICA certificate lacking the CA signing bit in the Key usage field. ○ Use the Openssl tool to initiate a connection to the TOE without the CA signing bit of the key usage field in the chain and verify the connections failed ○ Verify the connection failure via packet capture ○ Verify the connection failure via logs • By setting the path length field of a valid CA field to a value strictly less than the certificate path. ○ Configure the Intermediate certificate with the Path length of 0. ○ Use Openssl tool to initiate a connection to the TOE with path length field of a valid CA field to a value strictly less than the certificate path and verify the connections failed ○ Verify the connection failure via packet capture ○ Verify the connection failure via logs • Valid certificate chain ○ Create a full chain of certificates to connect to the TOE. ○ The evaluator then attempts to connect to the TLS server and verifies the connection succeeds. ○ The above step is verified with the help of Wireshark. • Invalid certificate chain ○ Attempt a connection from the TOE to the TLS server running with an untrusted certificate and confirm the connection fails. ○ Verify the connection fails with the help of system logs.

	○ Verify the connection failure with the help of network capture.
Pass/Fail with Explanation	Pass, TLS connection was successful when a valid certificate chain is presented and TLS connection is rejected when an incomplete certificate chain is presented.

6.2.2.2 FIA_X509_EXT.1.1(1) TEST #2 **[TD0641]**

Item	**Data**
Test Assurance Activity	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. **TD0641 has been applied.**
Test Steps	• The evaluator shall configure the TOE with server information. • The evaluator makes sure that the end certificate in the validation path is expired. • The evaluator then verifies that a connection attempt with the expired certificate is not possible and verifies the same with system logs. • The evaluator confirms the above step with the help of a traffic capture.
Pass/Fail with Explanation	Pass. The TLS connection is not established when an expired certificate is used.

6.2.2.3 FIA_X509_EXT.1.1(1) TEST #3 **[TD0641]**

Item	Data
Test Assurance Activity	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. **TD0641 has been applied.**
Test Steps	CRL • Create a chain of certificates with CRL Distribution Point • Attempt connection with a valid certificate chain • Verify the successful connection via packet capture • Revoke the peer leaf certificate • Attempt to make a connection • Verify the connection failure • Verify the connection failure via packet capture • Verify the connection failure via logs • Revoke the server's intermediate certificate • Attempt to make a connection • Verify the connection failure • Verify the connection failure with the help of logs OCSP: • Create a chain of certificates with OCSP • Attempt connection with a server running in valid certificate chain and confirm connection success

	• Verify OCSP responder's outputs • Verify the successful connection via packet capture • Revoke the peer leaf certificate • Attempt to make a connection from the TOE and verify the connection fails • Verify OCSP responder's outputs • Verify the connection failure with the help of system logs • Verify the connection failure via packet capture • Revoke the intermediate certificate • Attempt to make a connection from the TOE and verify the connection fails • Verify OCSP responder's outputs • Verify the connection failure with the help of system logs • Verify the connection failure via packet capture
Pass/Fail with Explanation	Pass, The TOE was able to handle certificate checking when presented with valid as well as revoked certificate in both CRL and OCSP Stapling methods TLS session was successful when a valid certificate chain is presented and TLS connection was rejected when a revoked certificate is presented.

6.2.2.4 FIA_X509_EXT.1.1(1) TEST #4 **[TD0641]**

Item	Data
Test Assurance Activity	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. **TD0641 has been applied.**

Test Steps	CRL: • The evaluator configures the TOE server with the certificates with the end certificate no having CRL sign Key usage. • The evaluator then verifies when the above step is performed the connection attempt fails and verifies the same with system logs. • The evaluator confirms this with the help of network traffic. OCSP: • The evaluator configure an OCSP responder certificate which does not have a OCSP signing EKU • Then the evaluator attempts to connect to the server from the TOE and verifies the attempt fails • The evaluator then verifies the TOE was able to fetch OCSP responses • The evaluator confirms connection failure with the help of network traffic and system logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when the certificate has no CRL sign and OCSP signing EKU present.

6.2.2.5 FIA_X509_EXT.1.1(1) TEST #5 **[TD0641]**

Item	Data
Test Assurance Activity	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.) **TD0641 has been applied.**
Test Steps	• Start the server using the acumen-tlsc-v2.2e tool with a modified byte within the first 8 bytes of the certificate, the connection should fail. • Verify the error logs on the TOE showing failure due to the wrong tag.

	• Verify the unsuccessful connection with packet capture.
Pass/Fail with Explanation	Pass. The connection was unsuccessful when a modified server certificate was used.

6.2.2.6 FIA_X509_EXT.1.1(1) TEST #6 **[TD0641]**

Item	Data
Test Assurance Activity	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.) **TD0641 has been applied.**
Test Steps	• Start the server using the acumen-tlsc-v2.2e tool with a modified byte within the last byte of the certificate, the connection should fail. • Verify the error logs on the TOE showing failure due to the wrong tag. • Verify the unsuccessful connection with packet capture.
Pass/Fail with Explanation	Pass, the TLS connection fails when a modified connection is used.

6.2.2.7 FIA_X509_EXT.1.1(1) TEST #7 **[TD0641]**

Item	Data

Test Assurance Activity	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.) **TD0641 has been applied.**
Test Steps	• Start the server using the acumen-tlsc-v2.2e tool with a modified byte of the certificate, the connection should fail. • Verify the error logs on the TOE showing failure due to the wrong tag. • Verify the unsuccessful connection with packet capture.
Pass/Fail with Explanation	Pass, TLS connection fails when a modified server certificate is used.

6.2.2.8 FIA_X509_EXT.1.1(1) TEST #8A **[TD0641]**

Item	Data
Test Assurance Activity	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 four certificates: the node certificate to be tested, two Intermediate CAs, 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. **TD0641 has been applied.**
Test Steps	• The evaluator attempts to configure the TOE with EC certificates of valid length. • The evaluator then verifies the connection attempt is a success and confirms with the help of a network capture.
Pass/Fail with Explanation	Pass, Pass, the connection was successful and all the certs in the chain are validated when a named curve is used.

6.2.2.9 FIA_X509_EXT.1.1(1) TEST #8B **[TD0641]**

Item	Data
Test Assurance Activity	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 four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. **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. **TD0641 has been applied.**
Test Steps	Client: • The evaluator attempts to connect with EC certificates of where ICA has explicit format version. • The evaluator makes sure that the attempted connection is failed and confirms the same via system logs.

	• The evaluator then confirms the above step with a network capture.
Pass/Fail with Explanation	Pass. The TLS connection did not take place when a modified ICA certificate explicit curve is used.

6.2.2.10 FIA_X509_EXT.1.2(1) TEST #1

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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.
Test Steps	Client: • The evaluator attempts to connect to a TLS server where the chain with the ICA cert is without basicConstraint. • The evaluator verified that connection attempt to a TLS server fails with the help of system logs. • The evaluator confirms the failure with the help of system logs and network capture.
Pass/Fail with Explanation	Pass. The TLS connection is rejected when the ICA does not have a Basic Constraints extension.

6.2.2.11 FIA_X509_EXT.1.2(1) TEST #2

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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.
Test Steps	• The evaluator attempts to connect with the ICA cert with basicConstraint CA set to false. • The evaluator verified that the connection fails and confirms with the help of a system logs and network capture.
Pass/Fail with Explanation	Pass. The TLS connection is rejected when the ICA certificate has a Basic Constraints extension where the cA flag is not set.

6.2.2.12 FIA_X509_EXT.1.2(1) TEST #3

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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.
Test Steps	• The evaluator attempts to connect to the OpenSSL server where a cert chain with the ICA cert with basicConstraint set to true. • The evaluator verified that the connection succeeds and confirms the connection with the help of network capture.
Pass/Fail with Explanation	Pass, the TLS connection is successful when the ICA certificate's basicConstraint is set to TRUE.

6.2.2.13 FIA_X509_EXT.2.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass, There are no test evaluation activities for this SFR.

6.2.2.14 FIA_X509_EXT.2.2 TEST #1

Item	Data
Test Assurance Activity	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.

Test Steps	• The evaluator shall confirm that OCSP server is not operational to check certificate's validity • The evaluator shall attempt to establish a TLS session with the TOE • The evaluator shall confirm the connection is rejected with the help of TOE output and logs • The connection rejection is also confirmed with the help of network capture
Pass/Fail with Explanation	Pass. The TLS rejected connection by the TOE when validity of a certificate cannot be determined.

6.2.3 TLSC TEST CASES

6.2.3.1 FCS_TLSC_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 1:** The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	• Configure the TOE to connect to a TLS server • Establish a connection with the TOE over TLS using the cipher TLS_RSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_RSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful

	• Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture
Pass/Fail with Explanation	Pass. The TLS connection was successful for each of the cipher selected in the ST.

6.2.3.2 FCS_TLSC_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 2:** The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation. The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established. The evaluator shall repeat this test using a different, but otherwise valid and trusted, certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension and ensure that a connection is not established. Ideally, the two certificates should be similar in structure, the types of identifiers used, and the chain of trust.
Test Steps	• Create a server certificate with the Server Authentication in the extendedKeyUsage field • Use openssl to initiate a successful TLS connection. • Verify the connection is established • Verify the connection is established via packet capture • Create a server certificate without the Server Authentication in the extendedKeyUsage field • Use openssl tool to initiate a connection to the TOE with a server certificate without the Server Authentication field resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Pass/Fail with Explanation	Pass, the connection is established when the server certificate contains the Server Authentication purpose in the extendedKeyUsage extension, and the connection is not established when the server certificate lacks the Server Authentication purpose in the extendedKeyUsage extension.

6.2.3.3 FCS_TLSC_EXT.1.1 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 3:** The evaluator shall send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message.
Test Steps	• Use openSSL to initiate a connection to the TOE to the TOE with an unsupported server certificate when the TOE runs in ECDSA resulted in a failure. • Verify the connection failure via packet capture • Verify the connection failure is via logs
Pass/Fail with Explanation	Pass. The TLS connection failed when a RSA certificate is passed when the TOE is operating in ECDSA ciphersuite.

6.2.3.4 FCS_TLSC_EXT.1.1 TEST #4

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the non-supported ciphersuite (NULL_WITH_NULL_NULL) • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. No TLS connection is established when NULL cipher is used by the server.

6.2.3.5 FCS_TLSC_EXT.1.1 TEST #5.1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.1:** Change the TLS version selected by the server in the Server Hello to an undefined TLS version (for example 1.5 represented by the two bytes 03 06) and verify that the client rejects the connection.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the undefined TLS version

	• Verify the connection failure via packet capture. • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS connection is rejected when an undefined TLS version is used.

6.2.3.6 FCS_TLSC_EXT.1.1 TEST #5.2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.2:** Change the TLS version selected by the server in the Server Hello to the most recent unsupported TLS version (for example 1.1 represented by the two bytes 03 02) and verify that the client rejects the connection.
Test Steps	• Use the OpenSSL tool to initiate a connection to the TOE and verify that the connection fails with the unsupported TLS version • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass the TLS connection is not established when TLS v1.1 is used by the server.

6.2.3.7 FCS_TLSC_EXT.1.1 TEST #5.3

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.3:** [conditional] If DHE or ECDHE cipher suites are supported, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE that will modify the server's nonce and verify that the connection fails with the modified server nonce • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS Connection is rejected when a modified server nonce is used.

6.2.3.8 FCS_TLSC_EXT.1.1 TEST #5.4

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.4:** Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator shall verify that the client does not complete the handshake and no application data flows.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE that will modify the cipher suite and verify that the connection fails with an unsupported ciphersuite • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS connection is not successful when a ciphersuite not part of the selection is used.

6.2.3.9 FCS_TLSC_EXT.1.1 TEST #5.5

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.5:** [conditional] If DHE or ECDHE cipher suites are supported, modify the signature block in the server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE that will modify the signature block of the server's key exchange and verify that the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS connection was not established when a byte in the server's signature block is modified.

6.2.3.10 FCS_TLSC_EXT.1.1 TEST #5.6

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.6:** Modify a byte in the Server Finished handshake message, and verify that the client does not complete the handshake and no application data flows.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE that will modify a byte in the server's finished handshake and verify that the connection fails

	• Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection was not successful when a byte in the server's finished handshake is modified.

6.2.3.11 FCS_TLSC_EXT.1.1 TEST #5.7

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5.7:** Send a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message and verify that the client does not complete the handshake and no application data flows. The message must still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE that will send a random message after the server's changed cipher spec and verify that the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. After the server has issued the Change Cipher Spec message, the client does not complete the handshake and no application data flows.

6.2.3.12 FCS_TLSC_EXT.1.2 TEST #1 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 1:** The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. Note that some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1. **TD0499 has been applied.**
Test Steps	CN as FQDN • TOE's settings and certificate details

	• Create a Server Certificate with no SAN and CN that does not match the reference identifier. • Use the OpenSSL to initiate a connection to the TOE using a certificate missing the SAN but with a CN that does not match the reference identifier and verify the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS connection is not established when no SAN is present.

6.2.3.13 FCS_TLSC_EXT.1.2 TEST #2 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 2:** The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type. **TD0499 has been applied.**
Test Steps	CN as FQDN • TOE's settings and certificate details • Create a Server Certificate with a SAN does not match the reference identifier and a CN that does match the reference identifier • Use the OpenSSL to initiate a connection to the TOE using a certificate the matched CN but with a SAN that does not match the reference identifier and verify the connection failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS connection is not established when incorrect SAN is present.

6.2.3.14 FCS_TLSC_EXT.1.2 TEST #3 **[TD0499]** - NA

Item	Data

Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 3:** [conditional] If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted. **TD0499 has been applied.**
Pass/Fail with Explanation	N/A, TOE mandates the presence of a SAN in the certificate, as suggested by the TAA testing is omitted.

6.2.3.15 FCS_TLSC_EXT.1.2 TEST #4 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 4:** The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. **TD0499 has been applied.**
Test Steps	CN as FQDN • TOE's settings and certificate details • Create a Server Certificate with a SAN that does match the reference identifier and a CN that does not match the reference identifier • Use the OpenSSL to initiate a connection to the TOE using a certificate the matched SAN but with a CN that does not match the reference identifier and verify the connection succeeds. • Verify the connection succeeds via packet capture
Pass/Fail with Explanation	Pass. The TOE accepts the connection when a server certificate contains a CN that mismatches the reference identifier and does contains the SAN extension that matches.

6.2.3.16 FCS_TLSC_EXT.1.2 TEST #5.1 **[TD0499]** - NA

Item	Data

Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.1:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	NA, Wildcards are not accepted by MDM agents

6.2.3.17 FCS_TLSC_EXT.1.2 TEST #5.2 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.2:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.come) and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	NA, Wildcards are not accepted by MDM agents

6.2.3.18 FCS_TLSC_EXT.1.2 TEST #5.3 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.3:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	NA, Wildcards are not accepted by MDM agents

6.2.3.19 FCS_TLSC_EXT.1.2 TEST #6 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 6:** [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails. **TD0499 has been applied.**

Pass/Fail with Explanation	NA, TOE does not support URI or Service Name reference identifier.

6.2.3.20 FCS_TLSC_EXT.1.2 TEST #7 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 7:** [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	N/A, certificate pinning is not supported by the TOE.

6.2.3.21 FCS_TLSC_EXT.1.2 TEST #5.4 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.4:** [conditional]: If wildcards are not supported, the evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	• Confirm the server is configured with left-most label reference identifier. • Create a server certificate containing a wildcard in the left-most identifier. • Use OpenSSL and initiate a connection from the TOE with a server certificate containing a wildcard in the left-most identifier resulting in failure • Verify the connection failure via packet capture • Verify the connection failure via logs

Pass/Fail with Explanation	Pass, The TLS connection was not successful when a server certificate with wildcard in the left-most label is presented.

6.2.3.22 FCS_TLSC_EXT.1.3 TEST #1A **[TD0513]**

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1a:** The evaluator shall demonstrate that a server using a certificate with a valid certification path successfully connects. **TD0513 has been applied.**
Test Steps	• Create a full chain of certificates to connect to the TOE • Configure TOE to connect to the TLS server • Attempt the connection from the TOE to the TLS server and verify the connection is successful • Verify the connection is successful via packet capture
Pass/Fail with Explanation	Pass. The TLS connection was successful when the correct cert path is used.

6.2.3.23 FCS_TLSC_EXT.1.3 TEST #1B **[TD0513]**

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1b:** The evaluator shall modify the certificate chain used by the server in test 1a to be invalid and demonstrate that a server using a certificate without a valid certification path to a trust store element of the TOE results in an authentication failure. **TD0513 has been applied.**
Test Steps	• Create a full chain of certificates to connect to the TOE and configure the correct chain on the TOE's trust store. • Create a certificate signed with an invalid certification path to TOE's trust store. • Attempt to connect to the TOE with the invalid intermediate CA and verify the connection failed • Verify that the connection failed via packet capture • Verify that the connection failed via logs
Pass/Fail with Explanation	Pass. The TLS connection is not successful when invalid cert path is used.

6.2.3.24 FCS_TLSC_EXT.1.3 TEST #1C **[TD0513]** - NA

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1c** [conditional]: If the TOE trust store can be managed, the evaluator shall modify the trust store element used in Test 1a to be untrusted and demonstrate that a connection attempt from the same server used in Test 1a results in an authentication failure. **TD0513 has been applied.**
Pass/Fail with Explanation	N/A. The TOE's trust store cannot be managed.

6.2.3.25 FCS_TLSC_EXT.1.3 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 2:** The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.
Test Steps	• Attempt to connect to the TOE with a revoked Certificate and verify the connection failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS connection is not be established when revoked cert is used

6.2.3.26 FCS_TLSC_EXT.1.3 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 3:** The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.
Test Steps	• Attempt to connect to the TOE with an expired certificate and verify the connection failed • Verify the connection failure via packet capture

	• Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS connection is not established when a expired certificate is used.

6.2.3.27 FCS_TLSC_EXT.1.3 TEST #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 4:** The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.
Test Steps	• Configure the valid identifier on TOE. • Create a Server Certificate with an invalid the reference identifier • Use OpenSSL to initiate a connection to the TOE and verify that the connection fails with an invalid the reference identifier • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. The TLS connection is not established when the server certificate is provided with invalid reference.

6.2.3.28 FCS_TLSC_EXT.2.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 1:** The evaluator shall establish a connection to a server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.
Test Steps	• Use openSSL to initiate a connection to the TOE without the Server's Certificate Request and verify that the connection succeeds • Verify the connection succeeds via packet capture
Pass/Fail with Explanation	Pass. The TLS server did not send a certificate request as configured and the connection was a success.

6.2.3.29 FCS_TLSC_EXT.2.1 TEST #2

Item	Data

Test Assurance Activity	The evaluator shall also perform the following tests: **Test 2:** The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Test Steps	• Use OpenSSL to initiate a connection to the TOE with the Server's Certificate Request and verify that the connection succeeds • Verify the connection succeeds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message via packet capture
Pass/Fail with Explanation	Pass. The server sends a certificate request as configured and client responds with certificate and a verify. Also, the connection is a success.

6.2.3.30 FCS_TLSC_EXT.5.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following test: **Test 1:** The evaluator shall configure a server to perform key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.
Test Steps	• Configure the TOE's server with each of the curve certificates. • Attempt a connection to the server from the TOE and confirm it's a success. • Verify the above connection with the help of Wireshark.
Pass/Fail with Explanation	Pass. The TLS communication was established for all selected curve.

6.2.4 CRYPTO TEST CASES

6.2.4.1 FCS_CKM.1 RSA[**TD0951**]

Item	Data
Test Assurance Activity	**Key Generation for FIPS PUB 186-4 or FIPS PUB 186-5 RSA Schemes** The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d.

	Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: 1. Random Primes: a. Provable primes b. Probable primes 2. Primes with Conditions: a. Primes p1, p2, q1,q2, p and q shall all be provable primes b. Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes c. Primes p1, p2, q1,q2, p and q shall all be probable primes To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. If possible, the Random Probable primes method should also be verified against a known good implementation as described above. Otherwise, the evaluator shall have the TSF generate 10 keys pairs for each supported key length nlen and verify: • n = p*q, • p and q are probably prime according to Miller-Rabin tests, • GCD(p-1,e) = 1, • GCD(q-1,e) = 1, • 2^16 <= e <= 2^256 and e is an odd integer, • |p-q| > 2^(nlen/2 - 100), • p >= squareroot(2)*(2^(nlen/2 -1)), • q >= squareroot(2)*(2^(nlen/2 -1)), • 2^(nlen/2) < d < LCM(p-1,q-1), • e*d = 1 mod LCM(p-1,q-1) **TD0951 has been applied.**
Pass/Fail with Explanation	Algorithm: RSA KeyGen Key size / Modulus: 2048,3072,4096 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.2 FCS_CKM.1 ECC **[TD0951]**

Item	Data
Test Assurance Activity	**FIPS 186-4 or FIPS 186-5 ECC Key Generation Test**

	For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation. **FIPS 186-4 or FIPS 186-5 Public Key Verification (PKV) Test** For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values. **TD0951 has been applied.**
Pass/Fail with Explanation	Algorithm: ECDSA KeyGen Curves: P-256, P-384, P-521 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.3 FCS_CKM.2 SP800-56A

Item	Data
Test Assurance Activity	**SP800-56A Key Establishment Schemes** The evaluator shall verify a TOEs implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the recommendation. These components include the calculation of the primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag. *Function Test* The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter

	values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested. The evaluator shall obtain the DKM, the corresponding TOEs public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE ID fields. If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret. The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values. If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm. *Validity Test* The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOEs public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE ID fields. The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOEs static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass). The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOEs results with the results using a known good implementation verifying that the TOE detects these errors.
Pass/Fail with Explanation	Algorithm: KAS-ECC-SSC SP800-56Ar3 Domain Parameter Generation Methods: P-256, P-384, P-521 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.4 FCS_CKM.2 RSA

Item	Data
Test Assurance Activity	**RSA-based key establishment** The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1(1), FTP_TRP.1(2), FTP_TRP.1(3), FTP_ITC.1(1), FTP_ITC.1(2), FPT_ITT.1(1), and FPT_ITT.1(2) that uses RSAES-PKCS1-v1_5.
Pass/Fail with Explanation	Pass. This testing was performed in conjunction with FCS_TLSC_EXT.1.1 Test#1.

6.2.4.5 FCS_CKM_EXT.4.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.2.4.6 FCS_CKM.4.2 TEST#1 – NA.

Item	Data
Test Assurance Activity	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: 8. Load the instrumented TOE build in a debugger. 9. Record the value of the key in the TOE subject to clearing. 10. Cause the TOE to perform a normal cryptographic processing with the key from #1. 11. Cause the TOE to clear the key. 12. Cause the TOE to stop the execution but not exit.

	13. Cause the TOE to dump the entire memory footprint of the TOE into a binary file. 14. 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.
Pass/Fail with Explanation	N/A, the selection for the TOE is “invoke platform-provided functionality”.

6.2.4.7 FCS_CKM.4.2 TEST #2 – NA.

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	N/A, the selection for the TOE is “invoke platform-provided functionality”.

6.2.4.8 FCS_COP.1.1(1) AES-GCM

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** **AES-GCM Test** **Test 1**: The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths: • 128 bit and 256 bit keys • Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported. • Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported. • Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

	The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known. The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5- tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
Pass/Fail with Explanation	Algorithm: AES-GCM Direction: Decrypt, Encrypt • Key Length: 128, 256 • IV Length: 96, 128 • Payload Length: 120, 128, 248, 256 • AAD Length: 0, 120, 128, 256 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.9 FCS_COP.1.1(2) HASHING

Item	Data
Test Assurance Activity	The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. **Short Messages Test Bit-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Short Messages Test Byte-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators

	compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Selected Long Messages Test Bit-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Selected Long Messages Test Byte-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. **Pseudorandomly Generated Messages Test** • **Test 1:** This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.
Pass/Fail with Explanation	Algorithm: SHA2-256, SHA2-384, SHA2-512 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.10 FCS_COP.1.1(3) SIGGEN ECDSA **[TD0951]**

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** **ECDSA Algorithm Tests** Test 1: ECDSA FIPS 186-4 or FIPS PUB 186-5 Signature Generation Test For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation. **Test 2: ECDSA FIPS 186-4 or FIPS PUB 186-5 Signature Verification Test**

	For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values. **TD0951 has been applied**
Pass/Fail with Explanation	Algorithm: ECDSA SigGen (FIPS 186-5), ECDSA SigVer (FIPS 186-5) Key size / Modulus: P-256, P-384, P-521 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.11 FCS_COP.1.1(3) SIGGEN RSA

Item	Data
Test Assurance Activity	**RSA Signature Algorithm Tests** **Test 1:** Signature Generation Test The evaluator shall verify the implementation of RSA Signature Generation by the TOE using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10 messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator shall have the TOE use their private key and modulus value to sign these messages. The evaluator shall verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures. **Test 2:** Signature Verification Test The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's valid and invalid signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure. The evaluator shall use these test vectors to emulate the signature verification test using the corresponding parameters and verify that the TOE detects these errors.
Pass/Fail with Explanation	Algorithm: RSA SigGen, RSA SigVer (FIPS 186-5) Key size / Modulus: 2048, 3072, 4096 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.12 FCS_COP.1.1(4) KEYED-HASH

Item	Data
Test Assurance Activity	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known good implementation.
Pass/Fail with Explanation	Algorithm: HMAC-SHA-256, HMAC-SHA-384 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.13 FCS_IV_EXT.1.1

Item	Data
Test Assurance Activity	The evaluator shall ensure that the generation of IVs for each key encrypted by the same KEK meets Table 4.
Pass/Fail with Explanation	Pass, refer to FCS_COP.1.1(1) AES-GCM test case for CAVP details.

6.2.4.14 FCS_RBG_EXT.1.1/AGT

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** The evaluator shall perform the following tests. The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality. If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits, (3) generate a second block of random bits, and (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random

	bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A). If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits, and (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. Entropy input: the length of the entropy input value must equal the seed length. Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length. Personalization string: The length of the personalization string must be less than or equal to seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied. Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.
Pass/Fail with Explanation	Algorithm: Counter DRBG Mode: Counter DRBG(AES-256) CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.2.4.15 FCS_RBG_EXT.1.2/AGT

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.3 GALAXY S23 ULTRA 5G RUNNING ANDROID 14

6.3.1 MDM AGENT CASES

6.3.1.1 FCS_STG_EXT.1(2) TEST #1 - NA

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR.
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR.

6.3.1.2 FAU_ALT_EXT.2 TEST #1

Item	Data
Test Assurance Activity	**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
Test Steps	• Create a policy update • Push the update to the MDM Agent • Verify that the MDM Agent has accepted the update and applied the policy • Verify that the MDM Server successfully received acknowledgement about the policy update
Pass/Fail with Explanation	Pass, MDM Server successfully pushes policy updates accepted by the MDM Agent

6.3.1.3 FAU_ALT_EXT.2 TEST #2

Item	Data
Test Assurance Activity	**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
Test Steps	• Create a policy update to the enrolled MDM Agent • Push the update to the MDM Agent • Verify that the MDM Agent has accepted the update and applied the policy • Verify that the MDM Server receives an alert from the MDM Agent for the successful policy update • Ensure the MDM Server and MDM Agent are connected • The MDM Server forces a check-in from the MDM server to determine reachability • Verify that the check-in process was successful • Attempt to check in from the MDM agent • Verify that check-in was successful on the MDM server
Pass/Fail with Explanation	Pass, MDM Agent alerts the MDM Server for the required events

6.3.1.4 FAU_ALT_EXT.2 TEST #3

Item	Data
Test Assurance Activity	**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
Test Steps	• Verify the MDM Agent and the MDM Server are connected • Trigger a forced check-in on the MDM Server to determine if there is an active connection between the MDM Agent and the MDM Server • Verify the result on the MDM Server • Disconnect the MDM Agent from the MDM Server and trigger a forced check-in on the MDM Server to determine the status of the connection • Reconnect the device to see the queued alert from the MDM Server
Pass/Fail with Explanation	Pass, MDM Agent successfully performs a network reachability test while both in a connected and disconnected state

6.3.1.5 FAU_ALT_EXT.2 TEST #4

Item	Data
Test Assurance Activity	**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
Test Steps	• Disconnect the MDM Agent from the network • Trigger a forced check-in on the MDM Server to determine the status of the connection • Reconnect the device to the network and see the queued action from the MDM Server post connection is established
Pass/Fail with Explanation	Pass, MDM agent performs as expected while disconnecting from the MDM Server during a network reachability test

6.3.1.6 FAU_GEN.1(2)/AGT TEST #1

Item	Data
Test Assurance Activity	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
Test Steps	• The evaluator performs operations on the TOE that will generate auditable events • The evaluator then records the events that were generated as part of the above step and confirms with the output specified in the TSS
Pass/Fail with Explanation	Pass, TOE generates audit events as expected

6.3.1.7 FAU_SEL.1(2) TEST #1

Item	Data
Test Assurance Activity	**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
Test Steps	• Create policy that will enable selective attribute audit logging and apply the policy • Perform operations that will generate auditable events • Verify the generated auditable event with the help of TOE logs
Pass/Fail with Explanation	Pass, The TOE generates audit record as expected for that audit operation

6.3.1.8 FAU_SEL.1(2) TEST #2 - NA

Item	Data
Test Assurance Activity	**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.
Pass/Fail with Explanation	NA, no other attributes are selected for the TOE.

6.3.1.9 FIA_ENR_EXT.2 TEST #1

Item	Data
Test Assurance Activity	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
Test Steps	• The evaluator installs the MDM server's root certificate on the mobile device to establish trust • Install the MDM client application in device-owner mode by using adb command via evaluator's laptop • Attempt to enroll the MDM agent with the reference identifier of the MDM server • Verify that the MDM Agent has stored the reference identifier of the MDM Server for future communication
Pass/Fail with Explanation	Pass, MDM Agent stores the reference identifier of the MDM Server during initial enrollment, and it's used for future communication

6.3.1.10 FMT_POL_EXT.2 TEST #1

Item	Data
Test Assurance Activity	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
Test Steps	• Verify the agent is configured on the available configuration interface • Create a policy update on the MDM Server • Push the policy update to the MDM Agent • Verify that the MDM Agent accepts the policy update
Pass/Fail with Explanation	Pass, MDM agent successfully applies the digitally signed policy update from the MDM Server

6.3.1.11 FMT_POL_EXT.2 TEST #2

Item	Data
Test Assurance Activity	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.
Test Steps	• The evaluator makes modifications to the config file so no signature signs policies for MDM agent • Create a policy update and push the policy to the mobile device • Verify application failure with the help of system logs • The evaluator makes modifications to the config file the policies for MDM agent on server • Create a policy update and push the policy to the mobile device • Verify the policy application failure with the help of system logs
Pass/Fail with Explanation	Pass, policies that were not signed and incorrectly signed were rejected by the TOE

6.3.1.12 FMT_SMF_EXT.4 TEST #1

Item	Data
Test Assurance Activity	**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
Test Steps	• See FMT_SMF_EXT 1.1
Pass/Fail with Explanation	Pass, refer to FMT_SMF_EXT 1.1 Test#1, as all the selections were also tested in the mentioned SFR

6.3.1.13 FMT_SMF_EXT.4 TEST #2 **[TD0755]**

Item	Data
Test Assurance Activity	**Test 2:** [conditional: if "import the certificates to be used for authentication of MDM Agent communications" is selected in FMT_SMF_EXT.4.1] 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) **TD0755 has been applied**
Test Steps	• Configure the MDM Agents authentication certificate • Attempt to enroll the MDM Agent • Verify the MDM Agent uses the certificate during the enrollment process
Pass/Fail with Explanation	Pass, refer to test FIA_ENR_EXT.2 Test#1, The MDM agent uses the configured authentication certificate for all of its communication with the server

6.3.1.14 FMT_SMF_EXT.4 TEST #3

Item	Data
Test Assurance Activity	**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
Test Steps	• Attempt to Enroll the MDM Agent in management • Verify that the MDM Agent can successfully enroll into management • Verify that MDM agent can manage the device by accessing the play store confirming communication with server
Pass/Fail with Explanation	Pass, the device was enrolled in management successfully and was able to manage and communicate with the MDM server

6.3.1.15 FMT_SMF_EXT.4 TEST #4

Item	Data
Test Assurance Activity	**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
Test Steps	For 15 mins: • Configure the periodicity of reachability events • Verify, the MDM Server receives alerts according to the schedule configured by the MDM Agent For 20 mins: • Configure the periodicity of reachability events • Verify, the MDM Server receives alerts according to the schedule configured by the MDM Agent

Pass/Fail with Explanation	Pass, the MDM agent was able to reach the MDM server based on configured sync/periodicity interval

6.3.1.16 FMT_SMF_EXT.4 TEST #5

Item	Data
Test Assurance Activity	**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
Test Steps	Refer to test FMT_SMF_EXT.4 Test#4
Pass/Fail with Explanation	Pass, refer to test FMT_SMF_EXT.4 Test#4. The MDM agent was able to reach the MDM server based on configured sync/periodicity interval that is performed with the help of sync policy

6.3.1.17 FMT_UNR_EXT.1 TEST #1

Item	Data
Test Assurance Activity	**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
Test Steps	• Configure the MDM agent in device-owner mode • Verify the MDM server and MDM agent are connected and attempt to unenroll the MDM Agent • Verify that the attempt fails
Pass/Fail with Explanation	Pass, the evaluator was not able to retire the device when enrolled in device-owner mode

6.3.1.18 FMT_UNR_EXT.1 TEST #2 – NA

Item	Data
Test Assurance Activity	**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.
Pass/Fail with Explanation	N/A, there is no "apply remediation actions" section made for the TOE.

6.3.2 AUTH TEST CASES

6.3.2.1 FIA_X509_EXT.1.1(1) TEST #1 **[TD0641]**

Item	Data
Test Assurance Activity	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 **TD0641 has been applied**
Test Steps	• Establish a certificate path in which one of the issuing certificates is not a CA certificate: Refer to : By setting the basicConstraints field in an issuing certificate to have CA=False • By omitting the basicConstraints field in one of the issuing certificates ○ Configure an ICA certificate lacking the basicConstraints extension ○ Use OpenSSL tool to initiate a connection to the TOE with the chain does not contain the basicConstraints extension and verify the connections failed ○ Verify the connection failure via packet capture ○ Verify the connection failure via logs • By setting the basicConstraints field in an issuing certificate to have CA=False ○ The evaluator manipulates the ICA certificate with the flag in the basicConstraints extension set to FALSE ○ Use the Openssl tool to initiate a connection to the TOE with the chain with the basicConstraints set to FALSE and verify the connections failed ○ Verify the connection failure via packet capture ○ Verify the connection failure via logs • By omitting the CA signing bit of the key usage field in an issuing certificate ○ Configure an ICA certificate lacking the CA signing bit in the Key usage field ○ Use the Openssl tool to initiate a connection to the TOE without the CA signing bit of the key usage field in the chain and verify the connections failed ○ Verify the connection failure via packet capture ○ Verify the connection failure via logs

	• By setting the path length field of a valid CA field to a value strictly less than the certificate path ○ Configure the root CA certificate with the Path length of 1 ○ Configure the Intermediate certificate with the Path length of 0 ○ Use Openssl tool to initiate a connection to the TOE with path length field of a valid CA field to a value strictly less than the certificate path and verify the connections failed ○ Verify the connection failure via packet capture ○ Verify the connection failure via logs • Valid certificate chain ○ Create a full chain of certificates to connect to the TOE ○ The evaluator then attempts to connect to the TLS server and verifies the connection succeeds ○ The above step is verified with the help of Wireshark • Invalid certificate chain ○ Attempt a connection from the TOE to the TLS server running with an untrusted certificate and confirm the connection fails ○ Verify the connection fails with the help of system logs ○ Verify the connection failure with the help of network capture
Pass/Fail with Explanation	Pass, TLS connection was successful when a valid certificate chain is presented and TLS connection is rejected when an incomplete certificate chain is presented

6.3.2.2 FIA_X509_EXT.1.1(1) TEST #2 **[TD0641]**

Item	Data
Test Assurance Activity	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 **TD0641 has been applied**
Test Steps	• The evaluator shall configure the TOE with server information • The evaluator makes sure that the end certificate in the validation path is expired • The evaluator then verifies that a connection attempt with the expired certificate is not possible and verifies the same with system logs • The evaluator confirms the above step with the help of a traffic capture
Pass/Fail with Explanation	Pass, the TLS connection is rejected when an expired certificate is used by the server

6.3.2.3 FIA_X509_EXT.1.1(1) TEST #3 **[TD0641]**

Item	Data
Test Assurance Activity	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 **TD0641 has been applied**
Test Steps	CRL • Create a chain of certificates with CRL Distribution Point • Attempt connection with a valid certificate chain • Verify the successful connection via packet capture • Revoke the peer leaf certificate • Attempt to make a connection • Verify the connection failure • Verify the connection failure via packet capture • Verify the connection failure via logs • Revoke the server's intermediate certificate • Attempt to make a connection • Verify the connection failure • Verify the connection failure with the help of logs OCSP: • Create a chain of certificates with OCSP • Attempt connection with a server running in valid certificate chain and confirm connection success • Verify OCSP responder's outputs • Verify the successful connection via packet capture • Revoke the peer leaf certificate

	• Attempt to make a connection from the TOE and verify the connection fails • Verify OCSP responder's outputs • Verify the connection failure with the help of system logs • Verify the connection failure via packet capture • Revoke the intermediate certificate • Attempt to make a connection from the TOE and verify the connection fails • Verify OCSP responder's outputs • Verify the connection failure with the help of system logs • Verify the connection failure via packet capture
Pass/Fail with Explanation	Pass, The TOE was able to handle certificate checking when presented with valid as well as revoked certificate in both CRL and OCSP Stapling methods TLS session was successful when a valid certificate chain is presented and TLS connection was rejected when a revoked certificate is presented

6.3.2.4 FIA_X509_EXT.1.1(1) TEST #4 **[TD0641]**

Item	Data
Test Assurance Activity	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 **TD0641 has been applied**
Test Steps	CRL: • The evaluator configures the TOE server with the certificates with the end certificate no having CRL sign Key usage • The evaluator then verifies when the above step is performed the connection attempt fails and verifies the same with system logs • The evaluator confirms this with the help of network traffic OCSP: • The evaluator configure an OCSP responder certificate which does not have a OCSP signing EKU • Then the evaluator attempts to connect to the server from the TOE and verifies the attempt fails • The evaluator then verifies the TOE was able to fetch OCSP responses

	• The evaluator confirms connection failure with the help of network traffic and system logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when the certificate has no CRL sign and OCSP signing EKU present

6.3.2.5 FIA_X509_EXT.1.1(1) TEST #5 **[TD0641]**

Item	Data
Test Assurance Activity	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) **TD0641 has been applied**
Test Steps	• Start the server using the acumen-tlsc tool with a modified byte within the first 8 bytes of the certificate, the connection should fail • Verify the error logs on the TOE showing failure • Verify the unsuccessful connection with packet capture
Pass/Fail with Explanation	Pass, TLS connection was rejected when a modified server certificate is used

6.3.2.6 FIA_X509_EXT.1.1(1) TEST #6 **[TD0641]**

Item	Data
Test Assurance Activity	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) **TD0641 has been applied**
Test Steps	• Start the server using the acumen-tls tool with a modified byte within the last byte of the certificate, the connection should fail • Verify the error logs on the TOE showing failure • Verify the unsuccessful connection with packet capture

Pass/Fail with Explanation	Pass, the TLS connection was rejected when a modified connection is used

6.3.2.7 FIA_X509_EXT.1.1(1) TEST #7 **[TD0641]**

Item	Data
Test Assurance Activity	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) **TD0641 has been applied**
Test Steps	• Start the server using the acumen-tls tool with a modified byte of the certificate, the connection should fail • Verify the error logs on the TOE showing failure • Verify the unsuccessful connection with packet capture
Pass/Fail with Explanation	Pass, TLS connection was rejected when a modified server certificate is used

6.3.2.8 FIA_X509_EXT.1.1(1) TEST #8A **[TD0641]**

Item	Data
Test Assurance Activity	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 four certificates: the node certificate to be tested, two Intermediate CAs, 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 **TD0641 has been applied**
Test Steps	• The evaluator attempts to configure the TOE with EC certificates of valid length • The evaluator then verifies the connection attempt is a success and confirms with the help of a network capture

Pass/Fail with Explanation	Pass, the connection was accepted and all the certs in the chain are validated when a named curve is used

6.3.2.9 FIA_X509_EXT.1.1(1) TEST #8B **[TD0641]**

Item	Data
Test Assurance Activity	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 four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA **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 **TD0641 has been applied**
Test Steps	• The evaluator attempts to connect with EC certificates of where ICA has explicit format version • The evaluator makes sure that the attempted connection is failed and confirms the same via system logs • The evaluator then confirms the above step with a network capture
Pass/Fail with Explanation	Pass, TLS connection was rejected when a modified ICA certificate with explicit curve is used

6.3.2.10 FIA_X509_EXT.1.2(1) TEST #1

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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

Test Steps	• The evaluator attempts to connect to a TLS server where the chain with the ICA cert is without basicConstraint • The evaluator verified that connection attempt to a TLS server fails with the help of system logs • The evaluator confirms the failure with the help of system logs and network capture
`	TLS connection should be rejected when the ICA cert does not contain basicConstarint
Pass/Fail with Explanation	Pass, TLS connection is rejected when the ICA does not have a Basic Constraints extension

6.3.2.11 FIA_X509_EXT.1.2(1) TEST #2

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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
Test Steps	• The evaluator attempts to connect with the ICA cert with basicConstraint CA set to false • The evaluator verified that the connection fails and confirms with the help of a system logs and network capture
Pass/Fail with Explanation	Pass, The TLS connection is rejected when the ICA certificate has a Basic Constraints extension where the cA flag is not set

6.3.2.12 FIA_X509_EXT.1.2(1) TEST #3

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** 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

Test Steps	• The evaluator attempts to connect to the TLS server where a cert chain with the ICA cert with basicConstraint set to true • The evaluator verified that the connection succeeds and confirms the connection with the help of network capture
Pass/Fail with Explanation	Pass, TLS connection was accepted when the ICA certificate's basicConstraint is set to TRUE

6.3.2.13 FIA_X509_EXT.2.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR
Pass/Fail with Explanation	Pass There are no test evaluation activities for this SFR

6.3.2.14 FIA_X509_EXT.2.2 TEST #1

Item	Data
Test Assurance Activity	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
Test Steps	• The evaluator shall confirm that OCSP server is not operational to check certificate's validity • The evaluator shall attempt to establish a TLS session with the TOE • The evaluator shall confirm the connection is rejected with the help of TOE output and logs • The connection rejection is also confirmed with the help of network capture
Pass/Fail with Explanation	Pass, TLS rejected connection by the TOE when validity of a certificate cannot be determined

6.3.3 TLSC TEST CASES

6.3.3.1 FCS_TLSC_EXT.1.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 1:** The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement This connection may be established as part of the establishment of a higher-level protocol, e g , as part of an EAP session It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES)
Test Steps	• Configure the TOE to connect to a TLS server • Establish a connection with the server over TLS using the cipher TLS_RSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the server over TLS using the cipher TLS_RSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the server over TLS using the cipher TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the server over TLS using the cipher TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the server over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the server over TLS using the cipher TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture
Pass/Fail with Explanation	Pass, TLS connection was accepted for each of the selected cipher suites

6.3.3.2 FCS_TLSC_EXT.1.1 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests:

	Test 2: The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X 509v3 server certificate validation The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established The evaluator shall repeat this test using a different, but otherwise valid and trusted, certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension and ensure that a connection is not established Ideally, the two certificates should be similar in structure, the types of identifiers used, and the chain of trust
Test Steps	• Create a server certificate with the Server Authentication in the extendedKeyUsage field • Use OpenSSL to initiate a TLS connection • Verify the connection is established • Verify the connection is established via packet capture • Create a server certificate without the Server Authentication in the extendedKeyUsage field • Use openssl to initiate a TLS connection with the TOE and verify connection fails • Verify the connection failure via packet capture • Verify the connection failure is via logs
Pass/Fail with Explanation	Pass, the connection is established when the server certificate contains the Server Authentication purpose in the extendedKeyUsage extension and the connection is not established when the server certificate lacks the Server Authentication purpose in the extendedKeyUsage extension.

6.3.3.3 FCS_TLSC_EXT.1.1 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 3:** The evaluator shall send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites) The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message
Test Steps	• Use Acumen-tls tool to initiate a connection with the TOE with an unsupported cipher when the TOE runs in RSA resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when an ECDSA cipher is used when the TOE is operating in RSA cipher suite

6.3.3.4 FCS_TLSC_EXT.1.1 TEST #4

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 4:** The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection
Test Steps	• Use the Acumen-tls tool to initiate a connection with the TOE and verify that the connection fails with the non-supported ciphersuite (NULL_WITH_NULL_NULL) • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when NULL cipher is used by the server

6.3.3.5 FCS_TLSC_EXT.1.1 TEST #5.1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5 1:** Change the TLS version selected by the server in the Server Hello to an undefined TLS version (for example 1 5 represented by the two bytes 03 06) and verify that the client rejects the connection
Test Steps	• Use the Acumen-tls tool to initiate a connection with the TOE and verify that the connection fails with the undefined TLS version • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection rejected when an undefined protocol is used

6.3.3.6 FCS_TLSC_EXT.1.1 TEST #5.2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5 2:** Change the TLS version selected by the server in the Server Hello to the most recent unsupported TLS version (for example 1 1 represented by the two bytes 03 02) and verify that the client rejects the connection
Test Steps	• Use the Acumen-tls tool to initiate a connection with the TOE and verify that the connection fails with the unsupported TLS version • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when an unsupported protocol (TLS v1 1) is used

6.3.3.7 FCS_TLSC_EXT.1.1 TEST #5.3

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5 3:** [conditional] If DHE or ECDHE cipher suites are supported, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows
Test Steps	• Use the Acumen-tls tool to initiate a connection with the TOE that will modify the server's nonce and verify that the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS Connection is rejected when a modified server certificate is used

6.3.3.8 FCS_TLSC_EXT.1.1 TEST #5.4

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5 4:** Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message The evaluator shall verify that the client does not complete the handshake and no application data flows
Test Steps	• Use the Acumen-tls tool to initiate a connection with the TOE that will modify the cipher suite and verify that the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when an unsupported cipher suite is used

6.3.3.9 FCS_TLSC_EXT.1.1 TEST #5.5

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5 5:** [conditional] If DHE or ECDHE cipher suites are supported, modify the signature block in the server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows This test does not apply to cipher suites using RSA key exchange If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted

Test Steps	• Use the Acumen-tls tool to initiate a connection with the TOE that will modify the signature block of server's key exchange and verify that the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when a modified server certificate is used

6.3.3.10 FCS_TLSC_EXT.1.1 TEST #5.6

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5 6:** Modify a byte in the Server Finished handshake message, and verify that the client does not complete the handshake and no application data flows
Test Steps	• Use the Acumen-tls tool to initiate a connection with the TOE that will modify a byte in server finished handshake and verify that the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when a modified server certificate is used

6.3.3.11 FCS_TLSC_EXT.1.1 TEST #5.7

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 5:** The evaluator shall perform the following modifications to the traffic: • **Test 5 7:** Send a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message and verify that the client does not complete the handshake and no application data flows The message must still have a valid 5-byte record header in order to ensure the message will be parsed as TLS
Test Steps	• Use the Acumen-tls tool to initiate a connection with the TOE that will send random message after server's changed cipher spec and verify that the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass. After the server has issued the Change Cipher Spec message, the client does not complete the handshake and no application data flows.

6.3.3.12 FCS_TLSC_EXT.1.2 TEST #1 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6 A pinned certificate must be added prior to performing Test 7 **Test 1:** The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension The evaluator shall verify that the connection fails Note that some systems might require the presence of the SAN extension In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier Both reasons are acceptable to pass Test 1 **TD0499 has been applied**
Test Steps	• Verify TOE's settings and reference identifier details • Create a Server Certificate with no SAN and CN that does not match the reference identifier • Use the OpenSSL to initiate a connection with the TOE using a certificate missing the SAN but with a CN that does not match the reference identifier and verify the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, the TLS was rejected when server certificate has no SAN is present and CN that does not match the reference identifier

6.3.3.13 FCS_TLSC_EXT.1.2 TEST #2 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6 A pinned certificate must be added prior to performing Test 7 **Test 2:** The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier The evaluator shall verify that the connection fails The evaluator shall repeat this test for each supported SAN type **TD0499 has been applied**
Test Steps	• Verify TOE's settings and reference identifier details • Create a Server Certificate with a SAN does not match the reference identifier and a CN that does match the reference identifier

	• Use the OpenSSL to initiate a connection with the TOE using a certificate the matched CN but with a SAN that does not match the reference identifier and verify the connection failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, the TLS connection was rejected when incorrect reference identifier is present in SAN of server certificate

6.3.3.14 FCS_TLSC_EXT.1.2 TEST #3 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6 .A pinned certificate must be added prior to performing Test 7 **Test 3:** [conditional] If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension The evaluator shall verify that the connection succeeds If the TOE does mandate the presence of the SAN extension, this Test shall be omitted **TD0499 has been applied**
Pass/Fail with Explanation	N/A, TOE mandates the presence of a SAN in the certificate, as suggested by the TAA testing is omitted

6.3.3.15 FCS_TLSC_EXT.1.2 TEST #4 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6 A pinned certificate must be added prior to performing Test 7 **Test 4:** The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches The evaluator shall verify that the connection succeeds **TD0499 has been applied**
Test Steps	• Verify TOE's settings and reference identifier details • Create a Server Certificate with a SAN that does match the reference identifier and a CN that does not match the reference identifier

	• Use the OpenSSL to initiate a connection with the TOE using a certificate the matched SAN but with a CN that does not match the reference identifier and verify the connection succeeds • Verify the connection succeeds via packet capture
Pass/Fail with Explanation	Pass, TLS connection is accepted when a server certificate contains a CN that does not matches the reference identifier and does contain the SAN extension that matches the reference identifier

6.3.3.16 FCS_TLSC_EXT.1.2 TEST #5.1 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.1:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	NA, Wildcards are not accepted by MDM agents

6.3.3.17 FCS_TLSC_EXT.1.2 TEST #5.2 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.2:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). The evaluator shall configure the

	reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.come) and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	NA, Wildcards are not accepted by MDM agents

6.3.3.18 FCS_TLSC_EXT.1.2 TEST #5.3 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed. • **Test 5.3:** [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	NA, Wildcards are not accepted by MDM agents

6.3.3.19 FCS_TLSC_EXT.1.2 TEST #5.4 **[TD0499]**

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6 A pinned certificate must be added prior to performing Test 7 **Test 5:** The evaluator shall perform the following wildcard tests with each supported type of reference identifier The support for wildcards is intended to be optional If wildcards are

	supported, the first, second, and third tests below shall be executed If wildcards are not supported, then the fourth test below shall be executed • **Test 5 4:** [conditional]: If wildcards are not supported, the evaluator shall present a server certificate containing a wildcard in the left-most label (e g * example com) The evaluator shall configure the reference identifier with a single left-most label (e g foo example com) and verify that the connection fails **TD0499 has been applied**
Test Steps	• Confirm the server is configured with left-most label reference identifier • Create a server certificate containing a wildcard in the left-most identifier • Use OpenSSL and initiate a connection with the TOE where the server certificate containing a wildcard in the left-most identifier resulting in failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, The TLS connection was not successful when a server certificate with wildcard in the left-most label is presented

6.3.3.20 FCS_TLSC_EXT.1.2 TEST #6 **[TD0499]** - NA

Item	**Data**
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 6:** [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	NA, TOE does not support URI or Service Name reference identifier.

6.3.3.21 FCS_TLSC_EXT.1.2 TEST #7 **[TD0499]** - NA

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. **Test 7:** [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails. **TD0499 has been applied.**
Pass/Fail with Explanation	N/A, certificate pinning is not supported by the TOE.

6.3.3.22 FCS_TLSC_EXT.1.3 TEST #1A **[TD0513]**

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1a:** The evaluator shall demonstrate that a server using a certifcate with a valid certification path successfully connects **TD0513 has been applied**
Test Steps	• Create a full chain of certificates to connect with the TOE • Configure TOE to connect to the TLS server • Attempt the connection from the TOE to the TLS server and verify the connection is successful • Verify the connection is successful via packet capture
Pass/Fail with Explanation	Pass, TLS connection was accepted when a valid certificate path is presented by the server

6.3.3.23 FCS_TLSC_EXT.1.3 TEST #1B **[TD0513]**

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1b:** The evaluator shall modify the certificate chain used by the server in test 1a to be invalid and demonstrate that a server using a certificate without a valid certification path to a trust store element of the TOE results in an authentication failure

	TD0513 has been applied
Test Steps	• Create a full chain of certificates to connect to the TOE and configure the correct chain on the TOE's trust store • Create a self-signed end server certificate • Attempt to connect with the TOE and verify the connection failed • Verify that the connection failed via packet capture • Verify that the connection failed via logs
Pass/Fail with Explanation	Pass, TLS connection was rejected when an invalid cert path is used by the server

6.3.3.24 FCS_TLSC_EXT.1.3 TEST #1C **[TD0513]**- NA

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 1c** [conditional]: If the TOE trust store can be managed, the evaluator shall modify the trust store element used in Test 1a to be untrusted and demonstrate that a connection attempt from the same server used in Test 1a results in an authentication failure **TD0513 has been applied**
Pass/Fail with Explanation	N/A, TOE's trust store cannot be managed

6.3.3.25 FCS_TLSC_EXT.1.3 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 2:** The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure
Test Steps	• Attempt to connect with the TOE using a revoked Certificate and verify the connection failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, the TLS connection was rejected when a revoked certificate is used by the server

6.3.3.26 FCS_TLSC_EXT.1.3 TEST #3

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 3:** The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure
Test Steps	• Attempt to connect with the TOE using an expired certificate and verify the connection failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, the TLS connection was rejected when an expired certificate is used by the server

6.3.3.27 FCS_TLSC_EXT.1.3 TEST #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: **Test 4:** The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure
Test Steps	• Configure the valid identifier on TOE • Create a Server Certificate with an invalid reference identifier • Use OpenSSL to initiate a connection with the TOE and verify that the connection fails • Verify the connection failure via packet capture • Verify the connection failure via logs
Pass/Fail with Explanation	Pass, TLS connection is rejected when the server certificate is provided with invalid reference

6.3.3.28 FCS_TLSC_EXT.2.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 1:** The evaluator shall establish a connection to a server that is not configured for mutual authentication (i e does not send Server's Certificate Request (type 13) message) The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake

Test Steps	• Use OpenSSL to initiate a connection with the TOE without the Server's Certificate Request and verify that the connection succeeds • Verify the connection succeeds via packet capture
Pass/Fail with Explanation	Pass, TLS connection was accepted and server does not send a certificate request as configured

6.3.3.29 FCS_TLSC_EXT.2.1 TEST #2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: **Test 2:** The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i e it sends Server's Certificate Request (type 13) message) The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message
Test Steps	• Use OpenSSL to initiate a connection with the TOE with the Server's Certificate Request and verify that the connection succeeds • Verify the connection succeeds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message via packet capture
Pass/Fail with Explanation	Pass, TLS connection was accepted by the TOE when server is operating as mutual authentication

6.3.3.30 FCS_TLSC_EXT.5.1 TEST #1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following test: **Test 1:** The evaluator shall configure a server to perform key exchange using each of the TOE's supported curves and/or groups The evaluator shall verify that the TOE successfully connects to the server
Test Steps	• Configure the TOE's server with each of the curve certificates • Attempt a connection to the server from the TOE and confirm it's a success • Verify the above connection with the help of Wireshark
Pass/Fail with Explanation	Pass, TLS communication was accepted for each of the selected curves

6.3.4 CRYPTO TEST CASES

6.3.4.1 FCS_CKM.1 RSA[**TD0951**]

Item	Data
Test Assurance Activity	**Key Generation for FIPS PUB 186-4 or FIPS PUB 186-5 RSA Schemes** The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: 3. Random Primes: a. Provable primes b. Probable primes 4. Primes with Conditions: a. Primes p1, p2, q1,q2, p and q shall all be provable primes b. Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes c. Primes p1, p2, q1,q2, p and q shall all be probable primes To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. If possible, the Random Probable primes method should also be verified against a known good implementation as described above. Otherwise, the evaluator shall have the TSF generate 10 keys pairs for each supported key length nlen and verify: • n = p*q, • p and q are probably prime according to Miller-Rabin tests, • GCD(p-1,e) = 1, • GCD(q-1,e) = 1, • 2^16 <= e <= 2^256 and e is an odd integer, • |p-q| > 2^(nlen/2 - 100), • p >= squareroot(2)*(2^(nlen/2 -1)), • q >= squareroot(2)*(2^(nlen/2 -1)), • 2^(nlen/2) < d < LCM(p-1,q-1), • e*d = 1 mod LCM(p-1,q-1) **TD0951 has been applied.**
Pass/Fail with Explanation	Algorithm: RSA KeyGen Key size / Modulus: 2048,3072,4096 CAVP #: A6402

	Pass. Based on these findings, this assurance activity is considered satisfied.

6.3.4.2 FCS_CKM.1 ECC **[TD0951]**

Item	Data
Test Assurance Activity	**FIPS 186-4 or FIPS 186-5 ECC Key Generation Test** For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation. **FIPS 186-4 or FIPS 186-5 Public Key Verification (PKV) Test** For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values. **TD0951 has been applied.**
Pass/Fail with Explanation	Algorithm: ECDSA KeyGen Curves: P-256, P-384, P-521 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied.

6.3.4.3 FCS_CKM.2 SP800-56A

Item	Data
Test Assurance Activity	**SP800-56A Key Establishment Schemes** The evaluator shall verify a TOEs implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the recommendation. These components include the calculation of the primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag

Function Test

The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested

The evaluator shall obtain the DKM, the corresponding TOEs public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE ID fields

If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret

The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values

If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOEs public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE ID fields

The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOEs static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass)

	The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOEs results with the results using a known good implementation verifying that the TOE detects these errors
Pass/Fail with Explanation	Algorithm: KAS-ECC-SSC SP800-56Ar3 Domain Parameter Generation Methods: P-256, P-384, P-521 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied

6.3.4.4 FCS_CKM.2 RSA

Item	Data
Test Assurance Activity	**RSA-based key establishment** The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1(1), FTP_TRP.1(2), FTP_TRP.1(3), FTP_ITC.1(1), FTP_ITC.1(2), FPT_ITT.1(1), and FPT_ITT.1(2) that uses RSAES-PKCS1-v1_5
Pass/Fail with Explanation	Pass. This testing was performed in conjunction with FCS_TLSC_EXT.1.1 Test#1

6.3.4.5 FCS_CKM_EXT.4.1 TEST #1

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR

6.3.4.6 FCS_CKM.4.2 TEST#1 – NA.

Item	Data
Test Assurance Activity	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: 15. Load the instrumented TOE build in a debugger. 16. Record the value of the key in the TOE subject to clearing. 17. Cause the TOE to perform a normal cryptographic processing with the key from #1. 18. Cause the TOE to clear the key. 19. Cause the TOE to stop the execution but not exit. 20. Cause the TOE to dump the entire memory footprint of the TOE into a binary file. 21. 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.
Pass/Fail with Explanation	N/A, the selection for the TOE is “invoke platform-provided functionality”.

6.3.4.7 FCS_CKM.4.2 TEST #2 – NA.

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	N/A, the selection for the TOE is “invoke platform-provided functionality”.

6.3.4.8 FCS_COP.1.1(1) AES-GCM

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** **AES-GCM Test** **Test 1**: The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

	• 128 bit and 256 bit keys • Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported. • Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported. • Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5- tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation
Pass/Fail with Explanation	Algorithm: AES-GCM Direction: Decrypt, Encrypt • Key Length: 128, 256 • IV Length: 96, 128 • Payload Length: 120, 128, 248, 256 • AAD Length: 0, 120, 128, 256 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied

6.3.4.9 FCS_COP.1.1(2) HASHING

Item	Data
Test Assurance Activity	The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP **<u>Short Messages Test Bit-oriented Mode</u>** • **Test 1:** The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range

	sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF **Short Messages Test Byte-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF **Selected Long Messages Test Bit-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF **Selected Long Messages Test Byte-oriented Mode** • **Test 1:** The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF **Pseudorandomly Generated Messages Test** • **Test 1:** This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF
Pass/Fail with Explanation	Algorithm: SHA2-256, SHA2-384, SHA2-512 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied

6.3.4.10 FCS_COP.1.1(3) SIGGEN ECDSA **[TD0951]**

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** **ECDSA Algorithm Tests** **Test 1: ECDSA FIPS 186-4 or FIPS PUB 186-5 Signature Generation Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation. **Test 2: ECDSA FIPS 186-4 or FIPS PUB 186-5 Signature Verification Test** For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values. **TD0951 has been applied**
Pass/Fail with Explanation	Algorithm: ECDSA SigGen, ECDSA SigVer Key size / Modulus: P-256, P-384, P-521I CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied

6.3.4.11 FCS_COP.1.1(3) SIGGEN RSA

Item	Data
Test Assurance Activity	**RSA Signature Algorithm Tests** **Test 1:** Signature Generation Test The evaluator shall verify the implementation of RSA Signature Generation by the TOE using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10 messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator shall have the TOE use their private key and modulus value to sign these messages The evaluator shall verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures **Test 2:** Signature Verification Test The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's valid and invalid signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing

	errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure The evaluator shall use these test vectors to emulate the signature verification test using the corresponding parameters and verify that the TOE detects these errors
Pass/Fail with Explanation	Algorithm: RSA SigGen, RSA SigVer Key size / Modulus: 2048, 3072, 4096 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied

6.3.4.12 FCS_COP.1.1(4) KEYED-HASH

Item	Data
Test Assurance Activity	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known good implementation
Pass/Fail with Explanation	Algorithm: HMAC-SHA-256, HMAC-SHA-384 CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied

6.3.4.13 FCS_IV_EXT.1.1

Item	Data
Test Assurance Activity	The evaluator shall ensure that the generation of IVs for each key encrypted by the same KEK meets Table 4.
Pass/Fail with Explanation	Pass, refer to FCS_COP.1.1(1) AES-GCM test case for CAVP details.

6.3.4.14 FCS_RBG_EXT.1.1/AGT

Item	Data
Test Assurance Activity	**If "implement functionality" is selected:** The evaluator shall perform the following tests. The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator

	shall also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits, (3) generate a second block of random bits, and (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A) If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits, and (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator Entropy input: the length of the entropy input value must equal the seed length Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length Personalization string: The length of the personalization string must be less than or equal to seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths
Pass/Fail with Explanation	Algorithm: Counter DRBG Mode: Counter DRBG(AES-256) CAVP #: A6402 Pass. Based on these findings, this assurance activity is considered satisfied

6.3.4.15 FCS_RBG_EXT.1.2/AGT

Item	Data
Test Assurance Activity	There are no test evaluation activities for this SFR
Pass/Fail with Explanation	Pass. There are no test evaluation activities for this SFR

7 CONCLUSION

The testing shows that all test cases required for conformance have passed testing.