



Assurance Activity Report

Version:	1.1
Date:	2019-03-12
Status:	RELEASED
Classification:	Public
Filename:	VID10937_SER_AAR_Apple_iPad_and_iPhone_with_iOS_12.0_v1.1
Product:	Apple iPad and iPhone Mobile Devices with iOS 12.0
Sponsor:	Apple Inc.
Evaluation Facility:	atsec information security corporation
Validation ID:	10937
Validation Body:	NIAP CCEVS
Author(s):	King Ables, Trang Huynh
Quality Assurance:	Fiona Pattinson

This report must not be used to claim product certification, approval, or endorsement by NIAP CCEVS, NVLAP, NIST, or any agency of the Federal Government.

atsec information security corporation
9130 Jollyville Road, Suite 260
Austin, TX 78759

Phone: +1 512-615-7300
Fax: +1 512-615-7301
www.atsec.com

Classification Note

Public Information (public)

This classification level is for information that may be made available to the general public. No specific security procedures are required to protect the confidentiality of this information. Information classified “public” may be freely distributed to anyone inside or outside of atsec.

Information with this classification shall be clearly marked “public”, except that it is not required to mark “public” on printed marketing material obviously intended for publication.

Revision History

Version	Date	Author(s)	Changes to Previous Revision	Application Notes
1.0	2019-02-26	King Ables	First version	
1.1	2019-03-12	Trang Huynh, King Ables	Update CAVS tables and address ECRs	

Table of Contents

1	Evaluation Basis and Documents	13
2	Evaluation Results	14
2.1	Security Functional Requirements	24
2.1.1	Security audit (FAU)	24
2.1.1.1	Audit Data Generation (FAU_GEN.1(1))	24
	TSS Assurance Activities	24
	Guidance Assurance Activities	24
	Test Assurance Activities	26
2.1.1.2	Audit Storage Protection (FAU_STG.1)	27
	TSS Assurance Activities	27
	Guidance Assurance Activities	27
	Test Assurance Activities	27
2.1.1.3	Prevention of Audit Data Loss (FAU_STG.4)	28
	TSS Assurance Activities	28
	Guidance Assurance Activities	28
	Test Assurance Activities	28
2.1.1.4	Audit Data Generation (FAU_GEN.1(2)(AGENT))	28
	FAU_GEN.1.1	28
	FAU_GEN.1.2	29
2.1.1.5	Extended: Agent Alerts (FAU_ALT_EXT.2(AGENT))	30
	FAU_ALT_EXT.2.1	30
	FAU_ALT_EXT.2.2	31
2.1.1.6	Security Audit Event Selection (FAU_SEL.1(2)(AGENT))	32
	TSS Assurance Activities	32
	Guidance Assurance Activities	32
	Test Assurance Activities	32
2.1.2	Cryptographic support (FCS)	33
2.1.2.1	Cryptographic Key Generation (FCS_CKM.1(1)/(2))	33
	TSS Assurance Activities	33
	Guidance Assurance Activities	33
	Test Assurance Activities	34
2.1.2.2	Extended: Cryptographic Key Support (REK) (FCS_CKM_EXT.1)	37
	TSS Assurance Activities	37
	Guidance Assurance Activities	38
	Test Assurance Activities	38
2.1.2.3	Cryptographic Key Establishment (FCS_CKM.2(1))	38
	TSS Assurance Activities	38
	Guidance Assurance Activities	39
	Test Assurance Activities	39
2.1.2.4	Cryptographic Key Establishment (While device is locked) (FCS_CKM.2(2))	41
	TSS Assurance Activities	41
	Guidance Assurance Activities	41
	Test Assurance Activities	41
2.1.2.5	Extended: Cryptographic Key Random Generation (FCS_CKM_EXT.2)	42

TSS Assurance Activities	42
Guidance Assurance Activities	43
Test Assurance Activities	43
2.1.2.6 Extended: Cryptographic Key Generation (FCS_CKM_EXT.3)	45
TSS Assurance Activities	45
Guidance Assurance Activities	47
Test Assurance Activities	47
2.1.2.7 Extended: Key Destruction (FCS_CKM_EXT.4)	49
TSS Assurance Activities	49
Guidance Assurance Activities	50
Test Assurance Activities	50
2.1.2.8 Extended: TSF Wipe (FCS_CKM_EXT.5)	51
TSS Assurance Activities	51
Guidance Assurance Activities	51
Test Assurance Activities	51
2.1.2.9 Extended: Salt Generation (FCS_CKM_EXT.6)	52
TSS Assurance Activities	52
Guidance Assurance Activities	52
Test Assurance Activities	52
2.1.2.10 Extended: Cryptographic Key Support (REK) (FCS_CKM_EXT.7)	53
TSS Assurance Activities	53
Guidance Assurance Activities	53
Test Assurance Activities	53
2.1.2.11 Cryptographic operation (Confidentiality Algorithms) (FCS_COP.1(1))	53
TSS Assurance Activities	53
Guidance Assurance Activities	53
Test Assurance Activities	53
2.1.2.12 Cryptographic operation (Hashing Algorithms) (FCS_COP.1(2))	56
TSS Assurance Activities	56
Guidance Assurance Activities	57
Test Assurance Activities	57
2.1.2.13 Cryptographic operation (Signature Algorithms) (FCS_COP.1(3))	58
TSS Assurance Activities	58
Guidance Assurance Activities	58
Test Assurance Activities	58
2.1.2.14 Cryptographic operation (Keyed Hash Algorithms) (FCS_COP.1(4))	59
TSS Assurance Activities	59
Guidance Assurance Activities	59
Test Assurance Activities	59
2.1.2.15 Cryptographic operation (Password-based Key Derivation Functions) (FCS_COP.1(5))	60
TSS Assurance Activities	60
Guidance Assurance Activities	60
Test Assurance Activities	60
2.1.2.16 Extended: HTTPS Protocol (FCS_HTTPS_EXT.1)	60
TSS Assurance Activities	60
Guidance Assurance Activities	60

Test Assurance Activities	60
2.1.2.17 Extended: Initialization Vector Generation (FCS_IV_EXT.1)	61
TSS Assurance Activities	61
Guidance Assurance Activities	61
Test Assurance Activities	61
2.1.2.18 Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)	61
TSS Assurance Activities	61
Guidance Assurance Activities	62
Test Assurance Activities	62
2.1.2.19 Extended: Cryptographic Algorithm Services (FCS_SRV_EXT.1)	63
TSS Assurance Activities	63
Guidance Assurance Activities	63
Test Assurance Activities	63
2.1.2.20 Extended: Cryptographic Key Storage (FCS_STG_EXT.1)	63
TSS Assurance Activities	63
Guidance Assurance Activities	64
Test Assurance Activities	64
2.1.2.21 Extended: Encrypted Cryptographic Key Storage (FCS_STG_EXT.2)	65
FCS_STG_EXT.2.1	65
FCS_STG_EXT.2.2	66
2.1.2.22 Extended: Integrity of Encrypted Key Storage (FCS_STG_EXT.3)	66
TSS Assurance Activities	66
Guidance Assurance Activities	66
Test Assurance Activities	66
2.1.2.23 Extended: TLS Client Protocol (FCS_TLSC_EXT.1)	67
FCS_TLSC_EXT.1.1	67
FCS_TLSC_EXT.1.2	69
FCS_TLSC_EXT.1.3	71
FCS_TLSC_EXT.1.4	72
2.1.2.24 Extended: TLS Protocol (FCS_TLSC_EXT.2)	73
TSS Assurance Activities	73
Guidance Assurance Activities	73
Test Assurance Activities	74
2.1.2.25 Cryptographic Key Storage (FCS_STG_EXT.4(AGENT))	74
TSS Assurance Activities	74
Guidance Assurance Activities	75
Test Assurance Activities	75
2.1.2.26 Cryptographic Key Generation (IKE) (FCS_CKM.1(VPN))	75
TSS Assurance Activities	75
Guidance Assurance Activities	75
Test Assurance Activities	75
2.1.2.27 IPsec (FCS_IPSEC_EXT.1)	76
FCS_IPSEC_EXT.1.1	76
FCS_IPSEC_EXT.1.2	78
FCS_IPSEC_EXT.1.3	79
FCS_IPSEC_EXT.1.4	80

FCS_IPSEC_EXT.1.5	81
FCS_IPSEC_EXT.1.6	82
FCS_IPSEC_EXT.1.7	83
FCS_IPSEC_EXT.1.8	85
FCS_IPSEC_EXT.1.9	86
FCS_IPSEC_EXT.1.11	86
FCS_IPSEC_EXT.1.12	89
FCS_IPSEC_EXT.1.13	90
FCS_IPSEC_EXT.1.14	90
2.1.2.28 Cryptographic Key Generation (FCS_CKM.1(2)(WLAN))	91
TSS Assurance Activities	91
Guidance Assurance Activities	92
Test Assurance Activities	92
2.1.2.29 Cryptographic Key Distribution (GTK) (FCS_CKM.2(3)(WLAN))	93
TSS Assurance Activities	93
Guidance Assurance Activities	93
Test Assurance Activities	93
2.1.2.30 Extensible Authentication Protocol-Transport Layer Security (FCS_TLSC_EXT.1(WLAN))	94
TSS Assurance Activities	94
Guidance Assurance Activities	94
Test Assurance Activities	95
2.1.3 User data protection (FDP)	97
2.1.3.1 Extended: Security Access Control (FDP_ACF_EXT.1)	97
FDP_ACF_EXT.1.1	97
FDP_ACF_EXT.1.2	99
2.1.3.2 Extended: Protected Data Encryption (FDP_DAR_EXT.1)	100
TSS Assurance Activities	100
Guidance Assurance Activities	100
Test Assurance Activities	101
2.1.3.3 Extended: Sensitive Data Encryption (FDP_DAR_EXT.2)	102
FDP_DAR_EXT.2.1	102
FDP_DAR_EXT.2.2	103
FDP_DAR_EXT.2.3	104
FDP_DAR_EXT.2.4	104
2.1.3.4 Extended: Subset Information Flow Control (FDP_IFC_EXT.1)	105
TSS Assurance Activities	105
Guidance Assurance Activities	105
Test Assurance Activities	106
2.1.3.5 Extended: Storage of Critical Biometric Parameters (FDP_PBA_EXT.1)	107
TSS Assurance Activities	107
Guidance Assurance Activities	108
Test Assurance Activities	108
2.1.3.6 Extended: User Data Storage (FDP_STG_EXT.1)	108
TSS Assurance Activities	108
Guidance Assurance Activities	109
Test Assurance Activities	109

2.1.3.7	1 Extended: Inter-TSF User Data Transfer Protection (FDP_UPC_EXT.1)	109
	TSS Assurance Activities	109
	Guidance Assurance Activities	109
	Test Assurance Activities	110
2.1.3.8	Full Residual Information Protection (FDP_RIP.2)	111
	TSS Assurance Activities	111
	Guidance Assurance Activities	112
	Test Assurance Activities	112
2.1.4	Identification and authentication (FIA)	112
2.1.4.1	Extended: Authentication Failure Handling (FIA_AFL_EXT.1)	112
	TSS Assurance Activities	112
	Guidance Assurance Activities	113
	Test Assurance Activities	114
2.1.4.2	Extended: Bluetooth User Authorization (FIA_BLT_EXT.1)	114
	TSS Assurance Activities	114
	Guidance Assurance Activities	115
	Test Assurance Activities	115
2.1.4.3	Extended: Bluetooth Mutual Authentication (FIA_BLT_EXT.2)	116
	TSS Assurance Activities	116
	Guidance Assurance Activities	116
	Test Assurance Activities	116
2.1.4.4	Rejection of Duplicate Bluetooth Connections (FIA_BLT_EXT.3)	117
	TSS Assurance Activities	117
	Guidance Assurance Activities	117
	Test Assurance Activities	117
2.1.4.5	Extended: Secure Simple Pairing (FIA_BLT_EXT.4)	118
	TSS Assurance Activities	118
	Guidance Assurance Activities	118
	Test Assurance Activities	118
2.1.4.6	Extended: Password Management (FIA_PMG_EXT.1)	118
	TSS Assurance Activities	118
	Guidance Assurance Activities	119
	Test Assurance Activities	119
2.1.4.7	Extended: Authentication Throttling (FIA_TRT_EXT.1)	119
	TSS Assurance Activities	119
	Guidance Assurance Activities	120
	Test Assurance Activities	120
2.1.4.8	Multiple Authentication Mechanisms (FIA_UAU.5)	120
	TSS Assurance Activities	120
	Guidance Assurance Activities	120
	Test Assurance Activities	121
2.1.4.9	Extended: Accuracy of Biometric Authentication (FIA_BMG_EXT.1)	121
	FIA_BMG_EXT.1.1	121
	FIA_BMG_EXT.1.2	122
2.1.4.10	Extended: Biometric Enrollment (FIA_BMG_EXT.2)	123
	TSS Assurance Activities	123

Guidance Assurance Activities	124
Test Assurance Activities	124
2.1.4.11 Extended: Biometric Verification (FIA_BMG_EXT.3)	125
TSS Assurance Activities	125
Guidance Assurance Activities	125
Test Assurance Activities	125
2.1.4.12 Extended: Handling Unusual Biometric Templates (FIA_BMG_EXT.5)	126
TSS Assurance Activities	126
Guidance Assurance Activities	126
Test Assurance Activities	126
2.1.4.13 Re-Authentication (FIA_UAU.6)	127
FIA_UAU.6.1	127
FIA_UAU.6.2	127
2.1.4.14 Protected Authentication Feedback (FIA_UAU.7)	128
TSS Assurance Activities	128
Guidance Assurance Activities	128
Test Assurance Activities	129
2.1.4.15 Extended: Authentication for Cryptographic Operation (FIA_UAU_EXT.1)	129
TSS Assurance Activities	129
Guidance Assurance Activities	130
Test Assurance Activities	130
2.1.4.16 Extended: Timing of Authentication (FIA_UAU_EXT.2)	131
TSS Assurance Activities	131
Guidance Assurance Activities	131
Test Assurance Activities	131
2.1.4.17 Extended: Validation of Certificates (FIA_X509_EXT.1)	131
TSS Assurance Activities	131
Guidance Assurance Activities	132
Test Assurance Activities	132
2.1.4.18 Extended: X509 Certificate Authentication (FIA_X509_EXT.2)	133
TSS Assurance Activities	133
Guidance Assurance Activities	134
Test Assurance Activities	134
2.1.4.19 Extended: Request Validation of certificates (FIA_X509_EXT.3)	135
TSS Assurance Activities	135
Guidance Assurance Activities	135
Test Assurance Activities	135
2.1.4.20 Extended: Enrollment of Mobile Device into Management (FIA_ENR_EXT.2(AGENT))	135
TSS Assurance Activities	135
Guidance Assurance Activities	136
Test Assurance Activities	136
2.1.4.21 Port Access Entity Authentication (FIA_PAE_EXT.1(WLAN))	137
TSS Assurance Activities	137
Guidance Assurance Activities	137
Test Assurance Activities	137

2.1.4.22	X.509 Certificate Authentication (EAP-TLS) (FIA_X509_EXT.2(WLAN))	138
	TSS Assurance Activities	138
	Guidance Assurance Activities	138
	Test Assurance Activities	139
2.1.5	Security management (FMT)	139
2.1.5.1	Extended: Management of Security Functions Behavior (FMT_MOF_EXT.1)	139
	FMT_MOF_EXT.1.1	139
	FMT_MOF_EXT.1.2	140
2.1.5.2	Extended: Specification of Management Functions (FMT_SMF_EXT.1)	141
	TSS Assurance Activities	141
	Guidance Assurance Activities	147
	Test Assurance Activities	150
2.1.5.3	Extended: Specification of Specification of Remediation Actions (FMT_SMF_EXT.2)	164
	TSS Assurance Activities	164
	Guidance Assurance Activities	165
	Test Assurance Activities	165
2.1.5.4	Trusted Policy Update (FMT_POL_EXT.2(AGENT))	165
	FMT_POL_EXT.2.1	165
	FMT_POL_EXT.2.2	166
2.1.5.5	Specification of Management Functions (FMT_SMF_EXT.3(AGENT))	167
	TSS Assurance Activities	167
	Guidance Assurance Activities	167
	Test Assurance Activities	167
	FMT_SMF_EXT.3.1	167
	FMT_SMF_EXT.3.2	169
2.1.5.6	User Unenrollment Prevention (FMT_UNR_EXT.1(AGENT))	170
	TSS Assurance Activities	170
	Guidance Assurance Activities	170
	Test Assurance Activities	171
2.1.5.7	Specification of Management Functions (VPN) (FMT_SMF.1(VPN))	171
	TSS Assurance Activities	171
	Guidance Assurance Activities	171
	Test Assurance Activities	172
2.1.5.8	Specification of Management Functions (Wireless LAN) (FMT_SMF_EXT.1(WLAN))	172
	TSS Assurance Activities	172
	Guidance Assurance Activities	172
	Test Assurance Activities	172
2.1.6	Protection of the TSF (FPT)	173
2.1.6.1	Extended: Anti-Exploitation Services (ASLR) (FPT_AEX_EXT.1)	173
	TSS Assurance Activities	173
	Guidance Assurance Activities	173
	Test Assurance Activities	173
2.1.6.2	Extended: Anti-Exploitation Services (Memory Page Permissions) (FPT_AEX_EXT.2)	174

TSS Assurance Activities	174
Guidance Assurance Activities	174
Test Assurance Activities	174
2.1.6.3 Extended: Anti-Exploitation Services (Overflow Protection) (FPT_AEX_EXT.3)	
.....	175
TSS Assurance Activities	175
Guidance Assurance Activities	175
Test Assurance Activities	175
2.1.6.4 Extended: Domain Isolation (FPT_AEX_EXT.4)	175
TSS Assurance Activities	175
Guidance Assurance Activities	177
Test Assurance Activities	177
2.1.6.5 Extended: JTAG Disablement (FPT_JTA_EXT.1)	178
TSS Assurance Activities	178
Guidance Assurance Activities	178
Test Assurance Activities	178
2.1.6.6 Extended: Key Storage (FPT_KST_EXT.1)	179
TSS Assurance Activities	179
Guidance Assurance Activities	180
Test Assurance Activities	180
2.1.6.7 Extended: No Key Transmission (FPT_KST_EXT.2)	180
TSS Assurance Activities	180
Guidance Assurance Activities	181
Test Assurance Activities	181
2.1.6.8 Extended: No Plaintext Key Export (FPT_KST_EXT.3)	181
TSS Assurance Activities	181
Guidance Assurance Activities	181
Test Assurance Activities	181
2.1.6.9 Extended: Self-Test Notification (FPT_NOT_EXT.1)	182
TSS Assurance Activities	182
Guidance Assurance Activities	182
Test Assurance Activities	182
FPT_NOT_EXT.1.1	182
2.1.6.10 Reliable Time Stamps (FPT_STM.1)	183
TSS Assurance Activities	183
Guidance Assurance Activities	183
Test Assurance Activities	183
2.1.6.11 Extended: TSF Cryptographic Functionality Testing (FPT_TST_EXT.1)	184
TSS Assurance Activities	184
Guidance Assurance Activities	184
Test Assurance Activities	185
2.1.6.12 Extended: TSF Integrity Testing (FPT_TST_EXT.2)	185
TSS Assurance Activities	185
Guidance Assurance Activities	185
Test Assurance Activities	185
2.1.6.13 Extended: Trusted Update: TSF Version Query (FPT_TUD_EXT.1)	186
TSS Assurance Activities	186

Guidance Assurance Activities	186
Test Assurance Activities	186
2.1.6.14 Extended: Trusted Update Verification (FPT_TUD_EXT.2)	187
FPT_TUD_EXT.2.3	187
FPT_TUD_EXT.2.4	189
2.1.6.15 TSF Self Test (VPN) (FPT_TST_EXT.1(VPN))	190
TSS Assurance Activities	190
Guidance Assurance Activities	190
Test Assurance Activities	191
2.1.6.16 TSF Cryptographic Functionality Testing (Wireless LAN) (FPT_TST_EXT.1(WLAN))	192
TSS Assurance Activities	192
Guidance Assurance Activities	192
Test Assurance Activities	194
2.1.7 TOE access (FTA)	194
2.1.7.1 Extended: TSF- and User-initiated Locked State (FTA_SSL_EXT.1)	194
TSS Assurance Activities	194
Guidance Assurance Activities	195
Test Assurance Activities	195
2.1.7.2 Default TOE Access Banners (FTA_TAB.1)	195
TSS Assurance Activities	195
Guidance Assurance Activities	196
Test Assurance Activities	196
2.1.7.3 Wireless Network Access (FTA_WSE_EXT.1(WLAN))	196
TSS Assurance Activities	196
Guidance Assurance Activities	196
Test Assurance Activities	197
2.1.8 Trusted path/channels (FTP)	197
2.1.8.1 Trusted Channel Communication (FTP_ITC_EXT.1(2)(AGENT))	197
TSS Assurance Activities	197
Guidance Assurance Activities	198
Test Assurance Activities	199
2.1.8.2 Trusted Channel Communication (Wireless LAN) (FTP_ITC_EXT.1(WLAN))	201
TSS Assurance Activities	201
Guidance Assurance Activities	201
Test Assurance Activities	201
2.2 Security Assurance Requirements	203
2.2.1 Life-cycle support (ALC)	203
2.2.1.1 Labelling of the TOE (ALC_CMC.1)	203
2.2.1.2 TOE CM coverage (ALC_CMS.1)	203
2.2.1.3 Extension: Timely Security Updates (ALC_TSU_EXT.1)	205
2.2.2 Guidance documents (AGD)	206
2.2.2.1 Operational user guidance (AGD_OPE.1)	206
2.2.2.2 Preparative procedures (AGD_PRE.1)	208
2.2.3 Tests (ATE)	208
2.2.3.1 Independent testing - conformance (ATE_IND.1)	208

2.2.4	Vulnerability assessment (AVA)	209
2.2.4.1	Vulnerability survey (AVA_VAN.1)	209
A	Appendixes	211
A.1	References	211
A.2	Glossary	216

List of Tables

Table 1: Apple CoreCrypto User Space and Kernel Space module certificates	14
Table 2: Apple SEP SKS and SEP Hardware module certificates	19
Table 3: Broadcom Wi-Fi CAVP certificates	23
Table 4: Summary of keys and persistent secrets used by the Agent	74
Table 5: EAP-TLS Ciphersuites	95

1 Evaluation Basis and Documents

This evaluation is based on the "Common Criteria for Information Technology Security Evaluation" Version 3.1 Revision 5 [CC], the "Common Methodology for Information Technology Security Evaluation" [CEM] and the additional assurance activities given in the Protection Profile for Mobile Device Fundamentals (MDFPP) Version 3.1 [PP_MD_V3.1], the Extended Package for Mobile Device Management Agents Version 3.0 [EP_MDM_AGENT_V3.0], the PP-Module for VPN Client Version 2.1 [MOD_VPN_CLI_V2.1], the Extended Package (EP) Wireless Local Area Network (WLAN) Clients Version 1.0 [PP_WLAN_CLI_EP_V1.0] and associated technical decisions.

This evaluation claims Exact Compliance with the above US Government PP and EP's.

The following scheme documents and interpretations have been considered:

- [CCEVS-TD0194]: "Update to Audit of FTP_ITC_EXT.1/WLAN", version as of 2017-04-11.
- [CCEVS-TD0237]: "FAU_GEN.1.1(2) - FMT_UNR_EXT.1 Audit Record Selection-Based", version as of 2017-09-26.
- [CCEVS-TD0244]: "FCS_TLSC_EXT - TLS Client Curves Allowed", version as of 2017-11-16.
- [CCEVS-TD0301]: "Updates to Administrator Management and Biometric Authentication", version as of 2018-04-04.
- [CCEVS-TD0304]: "Update to FCS_TLSC_EXT.1.2", version as of 2018-04-04.
- [CCEVS-TD0305]: "Handling of TLS connections with and without mutual authentication", version as of 2018-04-04.
- [CCEVS-TD0330]: "Curve25519 scheme moved to optional and FFC scheme using DH Group 14 added", version as of 2018-06-01.
- [CCEVS-TD0346]: "Revision of FMT_SMF_EXT.2 in MDF PP", version as of 2018-08-23.
- [CCEVS-TD0347]: "Update of Use Case 2 in MDF PP", version as of 2018-08-23.
- [CCEVS-TD0351]: "Additional methods for DEK formation", version as of 2018-09-20.
- [CCEVS-TD0366]: "Flexibility in Password Conditioning in FCS_COP.1(5)", version as of 2018-10-12.
- [CCEVS-TD0369]: "Long-term trusted channel key material", version as of 2018-10-17.
- [CCEVS-TD0378]: "TOE/TOE Platform Selection in FCS_IPSEC_EXT.1 SFRs", version as of 2018-12-13.
- [CCEVS-TD0379]: "Updated FCS_IPSEC_EXT.1.11 Tests for VPN Client", version as of 2018-12-20.

2 Evaluation Results

The evaluator work units have been performed, including: evaluator actions and analysis explicitly stated in the CEM; evaluator actions implicitly derived from developer action elements described in the CC Part 3; and evaluator confirmation that requirements for content and presentation of evidence elements described in the CC Part 3 have been met.

The evaluation was performed by informal analysis of the evidence provided by the sponsor.

The following table summarizes the Cryptographic Algorithm Validation Program (CAVP) certificates that apply to the TOE, including specific standards, options, and implementations for each algorithm of each SFR, and the applicable CAVP certificate for each.

Table 1: Apple CoreCrypto User Space and Kernel Space module certificates

SFR	Algorithm	Modes / Other	Implementation	Processor	User Space CAVP	Kernel Space CAVP
FCS_CKM.1(1)	RSA KeyGen [FIPS 186-4]	2048-bit, 3072-bit	Generic	A8	C 96	
				A8X	C 97	
				A9	C 98	
				A9X	C 100	
				A10 Fusion	C 99	
				A10X Fusion	C 102	
				A11 Bionic	C 101	
				A12 Bionic	RSA 3084	
				A12X Bionic	C 107	
	ECDSA KeyGen [FIPS 186-4]	P-256, P-384	Generic	A8	C 96	C 128
				A8X	C 97	C 129
				A9	C 98	C 130
				A9X	C 100	C 131
				A10 Fusion	C 99	C 132
				A10X Fusion	C 102	C 133
				A11 Bionic	C 101	C 134
				A12 Bionic	ECDSA 1567	C 135
				A12X Bionic	C 107	C 209
	DSA KeyGen [FIPS 186-4]	2048-bit	Generic	A8	C 96	
				A8X	C 97	
				A9	C 98	

SFR	Algorithm	Modes / Other	Implementation	Processor	User Space CAVP	Kernel Space CAVP
				A9X	C 100	
				A10 Fusion	C 99	
				A10X Fusion	C 102	
				A11 Bionic	C 101	
				A12 Bionic	DSA 1481	
				A12X Bionic	C 107	
FCS_CKM.1/VPN	RSA KeyGen [FIPS 186-4]	2048-bit, 3072-bit	Generic	A8	C 96	
				A8X	C 97	
				A9	C 98	
				A9X	C 100	
				A10 Fusion	C 99	
				A10X Fusion	C 102	
				A11 Bionic	C 101	
				A12 Bionic	RSA 3084	
				A12X Bionic	C 107	
	ECDSA KeyGen [FIPS 186-4]	P-256, P-384	Generic	A8	C 96	
				A8X	C 97	
				A9	C 98	
				A9X	C 100	
				A10 Fusion	C 99	
				A10X Fusion	C 102	
				A11 Bionic	C 101	
				A12 Bionic	ECDSA 1567	
				A12X Bionic	C 107	
FCS_CKM.2(1)	ECC Key Establishment (KAS-ECC) [SP800-56A]	P-256, P-384	Generic	A8	C 96	
				A8X	C 97	
				A9	C 98	
				A9X	C 100	
				A10 Fusion	C 99	
				A10X Fusion	C 102	

SFR	Algorithm	Modes / Other	Implementation	Processor	User Space CAVP	Kernel Space CAVP
	FFC Key Establishment (KAS-FFC) [SP800-56A]	2048-bit	Generic	A11 Bionic	C 101	
				A12 Bionic	CVL 2115	
				A12X Bionic	C 107	
				A8	C 96	
				A8X	C 97	
				A9	C 98	
				A9X	C 100	
				A10 Fusion	C 99	
				A10X Fusion	C 102	
				A11 Bionic	C 101	
				A12 Bionic	CVL 2115	
				A12X Bionic	C 107	
FCS_COP.1(1)	AES [FIPS 197]	128-bit, 256-bit CCM/CCMP, GCM [SP800-38C] (CCM, CCMP), [SP800-38D] (GCM)	Assembler_VNG	A8	AES 5725	AES 5750
				A8X	AES 5728	AES 5753
				A9	AES 5727	AES 5751
				A9X	AES 5731	AES 5755
				A10 Fusion	AES 5729	AES 5754
				A10X Fusion	AES 5732	AES 5757
				A11 Bionic	AES 5730	AES 5756
				A12 Bionic	AES 5880	AES 5884
				A12X Bionic	C 105	C 104
		128-bit, 256-bit CBC, XTS [SP800-38A] (CBC), [SP800-38E] (XTS)	Assembler_PAA	A8	AES 5717	AES 5758
				A8X	AES 5719	AES 5760
				A9	AES 5720	AES 5761
				A9X	AES 5722	AES 5762
				A10 Fusion	AES 5721	AES 5763
				A10X Fusion	AES 5723	AES 5764
				A11 Bionic	AES 5724	AES 5765
				A12 Bionic	AES 5881	AES 5885
				A12X Bionic	C 12	C 19

SFR	Algorithm	Modes / Other	Implementation	Processor	User Space CAVP	Kernel Space CAVP
		128-bit, 256-bit KW [SP800-38F] (KW)	Assembler	A8	AES 5701	AES 5742
				A8X	AES 5703	AES 5743
				A9	AES 5704	AES 5744
				A9X	AES 5705	AES 5745
				A10 Fusion	AES 5706	AES 5746
				A10X Fusion	AES 5707	AES 5747
				A11 Bionic	AES 5716	AES 5748
				A12 Bionic	AES 5879	AES 5883
				A12X Bionic	C 106	C 103
FCS_COP.1(2)	SHS [FIPS 180-4]	SHA-1, SHA-256, SHA-384, SHA-512	VNG	A8	SHS 4572	SHS 4580
				A8X	SHS 4574	SHS 4582
				A9	SHS 4571	SHS 4581
				A9X	SHS 4575	SHS 4583
				A10 Fusion	SHS 4576	SHS 4584
				A10X Fusion	SHS 4578	SHS 4586
				A11 Bionic	SHS 4577	SHS 4585
				A12 Bionic	SHS 4636	SHS 4637
				A12X Bionic	C 13	C 20
FCS_COP.1(3)	RSA SigGen and SigVer [FIPS 186-4]	2048-bit, 3072-bit using SHA-1 (SigVer only), SHA-256, SHA-384, SHA-512	Generic	A8	C 96	
				A8X	C 97	
				A9	C 98	
				A9X	C 100	
				A10 Fusion	C 99	
				A10X Fusion	C 102	
				A11 Bionic	C 101	
				A12 Bionic	RSA 3084	
				A12X Bionic	C 107	
	RSA SigVer [FIPS 186-4]	2048-bit, 3072-bit using	Generic	A8		C 128
				A8X		C 129

SFR	Algorithm	Modes / Other	Implementation	Processor	User Space CAVP	Kernel Space CAVP
		SHA-1, SHA-256, SHA-384, SHA-512		A9		C 130
				A9X		C 131
				A10 Fusion		C 132
				A10X Fusion		C 133
				A11 Bionic		C 134
				A12 Bionic		C 135
				A12X Bionic		C 209
	ECDSA SigGen and SigVer [FIPS 186-4]	P-256, P-384 using SHA-1 (SigVer only), SHA-256, SHA-384, SHA-512	Generic	A8	C 96	C 128
				A8X	C 97	C 129
				A9	C 98	C 130
				A9X	C 100	C 131
				A10 Fusion	C 99	C 132
				A10X Fusion	C 102	C 133
				A11 Bionic	C 101	C 134
				A12 Bionic	ECDSA 1567	C 135
				A12X Bionic	C 107	C 209
FCS_COP.1(4)	HMAC [FIPS 198-1]	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	VNG	A8	HMAC 3799	HMAC 3807
				A8X	HMAC 3801	HMAC 3809
				A9	HMAC 3798	HMAC 3808
				A9X	HMAC 3802	HMAC 3810
				A10 Fusion	HMAC 3803	HMAC 3811
				A10X Fusion	HMAC 3805	HMAC 3813
				A11 Bionic	HMAC 3804	HMAC 3812
				A12 Bionic	HMAC 3861	HMAC 3862
				A12X Bionic	C 13	C 20
FCS_RBG_EXT.1 (Kernel and User space)	CTR_DRBG(AES) [SP800-90A]	AES-128	Assembler_VNG	A8	DRBG 2320	DRBG 2345
				A8X	DRBG 2323	DRBG 2348
				A9	DRBG 2322	DRBG 2346
				A9X	DRBG 2326	DRBG 2350
				A10 Fusion	DRBG 2324	DRBG 2349

SFR	Algorithm	Modes / Other	Implementation	Processor	User Space CAVP	Kernel Space CAVP
				A10X Fusion	DRBG 2327	DRBG 2352
				A11 Bionic	DRBG 2325	DRBG 2351
				A12 Bionic	DRBG 2444	DRBG 2447
				A12X Bionic	C 105	C 104

Table 2: Apple SEP SKS and SEP Hardware module certificates

SFR	Algorithm	Modes / Other	Implementation	Processor	SEP SKS CAVP	SEP HW CAVP
FCS_CKM.1(1)	ECDSA KeyGen [FIPS 186-4]	P-256, P-384	Generic	A8	C 163	
				A8X	C 164	
				A9	C 165	
				A9X	C 166	
				A10 Fusion	C 167	
				A10X Fusion	C 168	
				A11 Bionic	C 169	
				A12 Bionic	C 258	
				A12X Bionic	C 260	
FCS_CKM.2(1)	ECC Key Establishment (KAS-ECC) [SP800-56A]	P-256, P-384	Generic	A8	C 163	
				A8X	C 164	
				A9	C 165	
				A9X	C 166	
				A10 Fusion	C 167	
				A10X Fusion	C 168	
				A11 Bionic	C 169	
				A12 Bionic	C 258	
				A12X Bionic	C 260	
FCS_COP.1(1)	AES [FIPS 197]	128-bit, 256-bit CCM, GCM [SP800-38C] (CCM),	Assembler_VNG	A8	C 262	
				A8X	C 263	
				A9	C 264	

SFR	Algorithm	Modes / Other	Implementation	Processor	SEP SKS CAVP	SEP HW CAVP
		[SP800-38D] (GCM)		A9X	C 266	
				A10 Fusion	C 265	
				A10X Fusion	C 267	
				A11 Bionic	C 275	
				A12 Bionic	C 274	
				A12X Bionic	C 273	
		128-bit, 256-bit CBC, XTS [SP800-38A] (CBC), [SP800-38E] (XTS)	Assembler_PAA	A8	C 200	
				A8X	C 203	
				A9	C 201	
				A9X	C 204	
				A10 Fusion	C 202	
				A10X Fusion	C 205	
				A11 Bionic	C 207	
				A12 Bionic	C 206	
				A12X Bionic	C 208	
		128-bit, 256-bit KW [SP800-38F] (KW)	Assembler	A8	C 87	
				A8X	C 88	
				A9	C 89	
				A9X	C 90	
				A10 Fusion	C 91	
				A10X Fusion	C 94	
				A11 Bionic	C 92	
				A12 Bionic	C 93	
				A12X Bionic	C 161	
		128-bit, 256-bit CBC [SP800-38A] (CBC)	SKG	A8		C 312
				A8X		C 313
				A9		C 314
				A9X		C 315
				A10 Fusion		C 317
				A10X Fusion		C 318

SFR	Algorithm	Modes / Other	Implementation	Processor	SEP SKS CAVP	SEP HW CAVP
				A11 Bionic		C 319
				A12 Bionic		C 320
				A12X Bionic		C 322
		256-bit ECB [SP800-38A] (ECB)	Hardware DRBG (For the CTR_DRBG(AES))	A8		AES 5270
				A8X		AES 5271
				A9		AES 5275
				A9X		AES 5272
				A10 Fusion		AES 5273
				A10X Fusion		AES 5274
				A11 Bionic		AES 5261
				A12 Bionic		C 323
				A12X Bionic		C 324
FCS_COP.1(2)	SHS [FIPS 180-4]	SHA-1, SHA-256, SHA-384, SHA-512	VNG	A8	C 270	
				A8X	C 268	
				A9	C 277	
				A9X	C 279	
				A10 Fusion	C 278	
				A10X Fusion	C 284	
				A11 Bionic	C 282	
				A12 Bionic	C 280	
				A12X Bionic	C 283	
FCS_COP.1(3)	ECDSA SigGen and SigVer [FIPS 186-4]	P-256, P-384 using SHA-1 (SigVer only), SHA-256, SHA-384, SHA-512	Generic	A8	C 163	
				A8X	C 164	
				A9	C 165	
				A9X	C 166	
				A10 Fusion	C 167	
				A10X Fusion	C 168	
				A11 Bionic	C 169	
				A12 Bionic	C 258	
				A12X Bionic	C 260	

SFR	Algorithm	Modes / Other	Implementation	Processor	SEP SKS CAVP	SEP HW CAVP
FCS_COP.1(4)	HMAC [FIPS 198-1]	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	VNG	A8	C 270	
				A8X	C 268	
				A9	C 277	
				A9X	C 279	
				A10 Fusion	C 278	
				A10X Fusion	C 284	
				A11 Bionic	C 282	
				A12 Bionic	C 280	
				A12X Bionic	C 283	
FCS_RBG_EXT.1 (SEP)	CTR_DRBG(AES) [SP800-90A]	AES-128	Assembler_VNG	A8	C 262	
				A8X	C 263	
				A9	C 264	
				A9X	C 266	
				A10 Fusion	C 265	
				A10X Fusion	C 267	
				A11 Bionic	C 275	
				A12 Bionic	C 274	
				A12X Bionic	C 273	
		AES-256	Hardware DRBG	A8		DRBG 2020
				A8X		DRBG 2021
				A9		DRBG 2025
				A9X		DRBG 2022
				A10 Fusion		DRBG 2023
				A10X Fusion		DRBG 2024
				A11 Bionic		DRBG 2014
				A12 Bionic		C 323
				A12X Bionic		C 324

Table 3: Broadcom Wi-Fi CAVP certificates

SFR	Algorithm	Modes / Other	iOS Device	Broadcom Core #	Core CAVP
FCS_COP.1(1)	AES [FIPS 197]	128-bit CCM [SP800-38C] (CCM)	iPhone 6 (A8)	4345	AES 4035
			iPhone 6 Plus (A8)	4345	AES 4035
			iPhone 6s (A9)	4350	AES 4035
			iPhone 6s Plus (A9)	4350	AES 4035
			iPhone SE (A9)	43452	AES 4035
			iPhone 7 (A10 Fusion)	4355	AES 3678
			iPhone 7 Plus (A10 Fusion)	4355	AES 3678
			iPhone 8 (A11 Bionic)	4357	AES 4152
			iPhone 8 Plus (A11 Bionic)	4357	AES 4152
			iPhone X (A11 Bionic)	4357	AES 4152
			iPhone Xs (A12 Bionic)	4377	AES 4791
			iPhone Xs Max (A12 Bionic)	4377	AES 4791
			iPhone X _R (A12 Bionic)	4377	AES 4791
			iPad mini 4 (A8)	4350	AES 4035
			iPad Air 2 (A8X)	4350	AES 4035
			iPad 9.7-inch (5th gen) (A9)	4355	AES 3678
			iPad Pro 9.7-inch (A9X)	4355	AES 3678
			iPad Pro 12.9-inch (A9X)	4350	AES 4035
			iPad 9.7-inch (6th gen) (A10 Fusion)	4355	AES 3678
			iPad Pro 12.9 (2nd gen) (A10X Fusion)	4355	AES 3678
			iPad Pro 10.5 (A10X Fusion)	4355	AES 3678
			iPad Pro 11-inch (A12X Bionic)	4377	AES 4791
			iPad Pro 12.9 (3rd gen) (A12X Bionic)	4377	AES 4791

2.1 Security Functional Requirements

2.1.1 Security audit (FAU)

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

TSS Assurance Activities

Assurance Activity AA-FAU_GEN.1-VPN-ASE-01

The evaluator shall examine the TSS to determine that it describes the auditable events and the component that is responsible for each type of auditable event.

Summary

Section 8.10.1 of the [ST] [\[1\]](#) specifies the *Audit Records*. The TOE collects logs specified by the configuration profile. The audit logs can be configured through a configuration profile, which are further explained in section 8.6.2 of the [ST] [\[1\]](#).

Guidance Assurance Activities

Assurance Activity AA-FAU_GEN.1-AGD-01

The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN.1.2.

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

Summary

[CCGUIDE] [\[1\]](#) chapter 4 *Security Audit* describes auditing and provides Table 15 "Audit Record Format" listing the audit events corresponding with the requirements specified in [ST] [\[1\]](#) with the associated audit events for those requirements from [PP_MD_V3.1] [\[1\]](#), [EP_MDM_AGENT_V3.0] [\[1\]](#), [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and [MOD_VPN_CLI_V2.1] [\[1\]](#).

Section 4.1 *Audit Logging* provides a sample audit record showing how the fields in this audit record map to the required fields specified in FAU_GEN.1. The sample outlines the basic format for each audit record showing that the audit records include at minimum the following information:

- date and time of the event;
- type of event (this is described as log level and log tag)
- subject identity (this is described as PID and PPID)
- the outcome (success or failure) of the event; and
- any applicable required additional information.

As stated at the beginning of section 4.1, the available commands and responses constitute audit records and must be configured by the TOE administrators using Configuration Profiles. The details for profile implementation and audit record collection are documented in [IOS_CFG] and [IOS_LOGS]. The evaluator examined these documents along with [ST] and [CCGUIDE] as well as through testing, the evaluator determined that the guidance contains all the administrative actions and their associated audit events that are relevant to [PP_MD_V3.1] and [EP_MDM_AGENT_V3.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1], and through use of the TOE. These administrative actions are found to be consistent with the actions specified in [ST].

The evaluator made the following observations:

- Every audit described in Table 15 provides the required information such as timestamps and subject identity.
- The SFRs covered in Table 15 match with Table 2 and Table 3 of [ST] that together cover the mandatory auditable events required by [PP_MD_V3.1], [EP_MDM_AGENT_V3.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1].
- Each SFR from Table 2 and Table 3 of [ST] is covered by at least one audit record and vice versa.

Assurance Activity AA-FAU_GEN.1-VPN-AGD-01

The evaluator shall check the operational guidance and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP-Module is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Table C-1 of the PP-Module.

In particular, the evaluator shall ensure that the operational guidance is clear in relation to the contents for failed cryptographic events. In Table C-1 of the PP-Module, information detailing the cryptographic mode of operation and a name or identifier for the object being encrypted is required. The evaluator shall ensure that name or identifier is sufficient to allow an administrator reviewing the audit log to determine the context of the cryptographic operation (for example, performed during a key negotiation exchange, performed when encrypting data for transit) as well as the non-TOE endpoint of the connection for cryptographic failures relating to communications with other IT systems.

The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP-Module. The TOE may contain functionality that is not evaluated in the context of this PP-Module because the functionality is not specified in an SFR. This functionality may have administrative aspects that are described in the operational guidance. Since such administrative actions will not be performed in an evaluated configuration of the TOE, the evaluator shall examine the operational guidance and make a determination of which administrative commands, including subcommands, scripts, and configuration files, 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-Module, which thus form the set of "all administrative actions". The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.

Summary

[CCGUIDE] chapter 4 *Audit* describes auditing and provides Table 15 "Audit Record Format" listing the audit events corresponding with the requirements specified in [ST] with the associated audit events for those requirements from [PP_MD_V3.1], [EP_MDM_AGENT_V3.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1].

Section 4.1 *Audit Logging* provides a sample audit record showing how the fields in this audit record map to the required fields specified in FAU_GEN.1. The sample outlines the basic format for each audit record showing that the audit records include at minimum the following information:

- date and time of the event;
- type of event (this is described as log level and log tag)
- subject identity (this is described as PID and PPID)

- the outcome (success or failure) of the event; and
- any applicable required additional information.

As stated at the beginning of section 4.1, the available commands and responses constitute audit records and must be configured by the TOE administrators using Configuration Profiles. The details for profile implementation and audit record collection are documented in [IOS_CFG] and [IOS_LOGS]. The evaluator examined these documents along with [ST] and [CCGUIDE] as well as through testing, the evaluator determined that the guidance contains all the administrative actions and their associated audit events that are relevant to [PP_MD_V3.1] and [EP_MDM_AGENT_V3.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1], and through use of the TOE. These administrative actions are found to be consistent with the actions specified in [ST].

The evaluator made the following observations:

- Every audit described in Table 15 provides the required information such as timestamps and subject identity.
- The SFRs covered in Table 15 match with Table 2 and Table 3 of [ST] that together cover the mandatory auditable events required by [PP_MD_V3.1], [EP_MDM_AGENT_V3.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1].
- Each SFR from Table 2 and Table 3 of [ST] is covered by at least one audit record and vice versa.

Test Assurance Activities

Assurance Activity AA-FAU_GEN.1-ATE-01

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the provided table and administrative actions. This should include all instances of an event. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields specified in FAU_GEN.1.2 are contained in each audit record.

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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator triggered all events listed in the provided table and administrative actions, and used the console provided by Xcode to view and save the devices' logs.

Assurance Activity AA-FAU_GEN.1-VPN-ATE-01

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records in accordance with the Assurance Activities associated with the functional requirements in this PP-Module. Additionally, the evaluator shall test that each administrative action applicable in the context of this PP-Module is auditable. When verifying the test results, 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Not applicable because not claimed in the [ST].

2.1.1.2 Audit Storage Protection (FAU_STG.1)

TSS Assurance Activities

Assurance Activity AA-FAU_STG.1-ASE-01

The evaluator shall ensure that the TSS lists the location of all logs and the access controls of those files such that unauthorized modification and deletion are prevented.

Summary

Section 8.10.1 of the [ST] describes the *Audit Records*.

The evaluator ensured that section 8.10.1 defines the storage location for the audit records. The TSS states that the storage location depends on the underlying platform used by the trusted workstation or MDM OS as follows.

- macOS: ~/Library/Logs/CrashReporter/MobileDevice/[Your_Device_Name]/
- Windows: C:\Users\[Your_User_Name]\AppData\Roaming\Apple Computer\Logs\CrashReporter\MobileDevice\[Your_Device_Name]\

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FAU_STG.1-ATE-01

- **Test 1:** The evaluator shall attempt to delete the audit trail in a manner that the access controls should prevent (as an unauthorized user) and shall verify that the attempt fails.
- **Test 2:** The evaluator shall attempt to modify the audit trail in a manner that the access controls should prevent (as an unauthorized application) and shall verify that the attempt fails.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: During the test for FAU_GEN.1, the evaluator observed that the interface to obtain the audit log information is read-only. This means that the interface is not capable of allowing a modify/write/delete operation. The audit trail is stored by the TOE such that a user application can

only access the provided interface allowing the read operation. The audit log is kept in a storage location on the iOS devices where only the operating system can write to. The audit data is protected from modification attempts by user processes using the standard iOS protection mechanisms. The reading is limited to the log framework. That log framework can be accessed using tools provided with Xcode. Thus, the tester uses the Xcode log framework tool to access the audit logs to verify the audit log entries. While obtaining the audit log, the evaluator tried to delete the log and verified that the modification did not succeed.

Test 2: see Test 1. While obtaining the audit log, the evaluator tried to delete the log and verified that the modification did not succeed.

2.1.1.3 Prevention of Audit Data Loss (FAU_STG.4)

TSS Assurance Activities

Assurance Activity AA-FAU_STG.4-ASE-01

The evaluator shall examine the TSS to ensure that it describes the size limits on the audit records, the detection of a full audit trail, and the action(s) taken by the TSF when the audit trail is full. The evaluator shall ensure that the action(s) results in the deletion or overwrite of the oldest stored record.

Summary

Section 8.10.1 of the [ST]  describes the *Audit Records* .

The audit storage capacity is defined by the profiles configured by the TOE administrators, as well as the default action when the storage capacity is reached.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.1.4 Audit Data Generation (FAU_GEN.1(2)(AGENT))

FAU_GEN.1.1


TSS Assurance Activities

Assurance Activity AA-FAU_GEN.1.1-2-MDMA-ASE-01

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.

Summary

Section 8.10.1, *Audit Records*, in the [ST]  describes how the TOE performs auditing.

Section 8.10.1 specifies how audit record storage is performed by the TOE. The evaluator checked that section 8.10.1 describes how audit records are performed by the TOE. The TOE administrators must use configuration profiles to configure the audit storage capacity and the action to take when that limit is reached. The format of the audit records is described in the operational guidance, the [CCGUIDE]  .

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FAU_GEN.1.1-2-MDMA-ATE-01

The evaluator shall use the TOE to perform the auditable events defined in Table 1 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator performed all audible actions and recorded the device logs through the Xcode console. The evaluator verified that accurate audit logs are generated with contents and formatting consistent with those described in the TSS.

FAU_GEN.1.2

TSS Assurance Activities

Assurance Activity AA-FAU_GEN.1.2-2-MDMA-ASE-01

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.

Summary

Section 8.10.1, *Audit Records*, in the [ST] describes how the TOE performs auditing.

Section 8.10.1 specifies how audit record storage is performed by the TOE. The evaluator checked that section 8.10.1 describes how audit records are performed by the TOE. The TOE administrators must use configuration profiles to configure the audit storage capacity and the action to take when that limit is reached. The format of the audit records is described in the operational guidance, the [CCGUIDE].

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FAU_GEN.1.2-2-MDMA-ATE-01

When verifying the test results, 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator performed all audible actions and recorded the device logs through the Xcode console. The evaluator verified that accurate audit logs are generated with contents and formatting matched the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

2.1.1.5 Extended: Agent Alerts (FAU_ALT_EXT.2(AGENT))

FAU_ALT_EXT.2.1

TSS Assurance Activities

Assurance Activity AA-FAU_ALT_EXT.2.1-MDMA-ASE-01

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

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

The evaluator also ensures that the TSS describes how reachability events are implemented, and if configurable are selected in FMT_SMF_EXT.3.2. The evaluator verifies that this description clearly indicates who (MDM Agent or MDM Server) initiates reachability events.

Summary

Section 8.10.2, *MDM Agent Alerts*, in the [ST] describes the agent alerts generated and received by the TOE.

Section 8.10.2 describes that the MDM agent generates and sends an alert in response to an MDM server request. Section 8.10.2.2 describes how the agent communicates to the server that a policy was (un)successfully applied as follows.

"When the application of policies to a mobile device is successful the MDM Agent replies with an MDM Result Payload with Status value "Acknowledged". If a policy update is not successfully installed then the MDM Agent replies with an MDM Result Payload with Status value "Error" or CommandFormatError, "Idle" and "NotNow"."

Section 8.10.2.3 describes alerts on receiving periodic reachability events and that periodic reachability events are initiated by the MDM Server using Push Notifications. The software component processing these alerts is the Apple MDM Agent described in [IOS_MDM].

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FAU_ALT_EXT.2.1-MDMA-ATE-01

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 2: The evaluator shall perform each of the actions listed in FAU_ALT_EXT. 1.1 and verify that the alert does in fact reach the MDM Server.

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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[1\]](#).

Test 1: the evaluator configured a passcode length policy and verified that the policy was correctly enforced by the TOE.

Test 2: the evaluator notes that the testing of the successful application of policies to the mobile device is tested with Test 1. This test covers the submission of an alert after the receipt of periodic reachability events.

Test 3: The evaluator re-performed Test 2, but prior to sending the notification to the client, the clients are disconnected from the WLAN. Upon reconnection, the notification were automatically sent.

FAU_ALT_EXT.2.2

TSS Assurance Activities

Assurance Activity AA-FAU_ALT_EXT.2.2-MDMA-ASE-01

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.

Summary

Section 8.10.2.1 of the [ST][\[1\]](#) describes the *Queuing of Alerts* .

Section 8.10.2.1 describes that if alerts cannot be sent by the device (for example, if the device is out of reach of a network), these alerts will be queued. This section explains that the queue cannot become long, since when the device is out of communication with the MDM, no additional MDM requests can be received. Whenever the device is not able to perform an MDM server request (for example, databases cannot be modified when the device is locked with Data Protection), the device will issue the NotNow status without performing the command. After issuing the NotNow status, the device will poll the server later. The device will poll the server until a successful transaction is completed.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FAU_ALT_EXT.2.2-MDMA-ATE-01

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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator disconnected the TOE from the network and ensured that the device had no connectivity. The evaluator then re-ran FAU_ALT_EXT.2.1 Test 1, and ensured that the actions were listed on the Active Tasks list on the web UI for the MDM. The evaluator reconnected the device to the WLAN, waited up to one minute, and could verify that the passcode policy was indeed applied. The evaluator also verified on the web UI that the task had been applied.

Test 2: the evaluator disconnected the TOE from the network and ensured that the device had no connectivity. The evaluator then re-ran FAU_ALT_EXT.2.1 Test 1, and ensured that shortly after the devices reconnected to the WLAN, the action on the Active Tasks list was cleared. The evaluator could then check on the web UI, in the Completed Tasks list, that the Update Info appears with the date and time the configuration occurred.

2.1.1.6 Security Audit Event Selection (FAU_SEL.1(2)(AGENT))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FAU_SEL.1-2-MDMA-AGD-01

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

Summary

[CCGUIDE] section 4.2 *Audit Storage* and section 4.3 *Configure the Auditable Items* provides guidance for how to select auditable events to be audited. It states the following:

- iOS implements a logging framework for logging different logging levels. This framework is configured by creating and installing a logging configuration profile property list file (i.e., .plist file) into the appropriate directory as described in [LOGGING].
- [AConfig] describes how to use the device console to see all logged information that occurs between the device, Apple Configurator 2, and possibly connections outside your network to your mobile device management (MDM) solution or Apple. Administrators can mark a selection, clear the window to view a specific event, or save the log for troubleshooting.
- Per [IOS_LOGS], additional logs can be specified by performing user actions on a device or through using a Configuration Profile.
- Table 16 "Additional Audit Logs" identifies all the logs that can be optionally selected and how they can be initiated where the majority of which requires a configuration profile.

Test Assurance Activities

Assurance Activity AA-FAU_SEL.1-2-MDMA-ATE-01

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 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator performed this testing as part of FAU_GEN.1.

Test 2: not applicable.

2.1.2 Cryptographic support (FCS)

2.1.2.1 Cryptographic Key Generation (FCS_CKM.1(1)/(2))

TSS Assurance Activities

Assurance Activity AA-FCS_CKM.1-ASE-01

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.

Summary

Section 8.1 of the [ST] specifies the *Mapping to the Security Functional Requirements*.

The table entry FCS_CKM.1.1(1) {MDF} {VPN} {AGENT} in section 8.1 specifies the key sizes for FCS_CKM.1.1, which points to section 8.3.4 and *Table 8: Explanation of usage for cryptographic functions*: The modules can generate RSA key pairs with modulus sizes of 2048 to 3072. ECC key pairs can be generated for NIST curves P-256, P-384.

Guidance Assurance Activities

Assurance Activity AA-FCS_CKM.1-AGD-01

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP.

Summary

The developer provided the following document as the main administrative guide for configuring and using the TOE in the evaluated configuration:

- Apple iPad and iPhone Mobile Devices with iOS 12 Common Criteria Configuration Guide, [CCGUIDE]

Section 1.3 *Supporting Apple Documentation* of [CCGUIDE] lists the supporting Apple documentation referenced in [CCGUIDE] and contains the following statement:

This document provides clarifications to the Apple documentation as related to configuring the mobile devices into the evaluated configuration. The official Apple documentation should be referred to and followed only as directed within this document. This document supplements and supersedes the Apple documentation.

[CCGUIDE] section 1.7 *Security Functional Requirements in the ST requiring configuration* contains Table 3: "SFR Configuration Requirements". Table entry for FCS_CKM.1(1) indicates that the the selected key size generation schemes and key sizes in [ST] cannot be configured in the evaluated configuration.

Test Assurance Activities

Assurance Activity AA-FCS_CKM.1-ATE-01

Assurance Activity Note: *The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.*

Key Generation for FIPS PUB 186-4 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:*
 - *Provable primes*
 - *Probable primes*
2. *Primes with Conditions:*
 - *Primes p_1 , p_2 , q_1 , q_2 , p and q shall all be provable primes*
 - *Primes p_1 , p_2 , q_1 , and q_2 shall be provable primes and p and q shall be probable primes*
 - *Primes p_1 , p_2 , q_1 , q_2 , 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
- $|p - q| > 2^{(nlen/2 - 100)}$
- $p \geq \text{squareroot}(2) * (2^{(nlen/2 - 1)})$
- $q \geq \text{squareroot}(2) * (2^{(nlen/2 - 1)})$
- $2^{(nlen/2)} < d < LCM(p-1, q-1)$
- $e * d = 1 \text{ mod } LCM(p-1, q-1)$

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

Assurance Activity AA-FCS_CKM.1-ATE-02

Key Generation for FIPS 186-4 Elliptic Curve Cryptography (ECC)

FIPS 186-4 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 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

Assurance Activity AA-FCS_CKM.1-ATE-03

Key Generation for Curve25519

The evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated as specified in RFC 7748 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.

Note: Assuming the PKV function of the good implementation will:

- (a) confirm the private and public keys are 32-byte values
- (b) confirm the three least significant bits of the most significant byte of the private key are zero
- (c) confirm the most significant bit of the least significant byte is zero
- (d) confirm the second most significant bit of the most significant byte is one
- (e) calculate the expected public key from the private key and confirm it matches the supplied public key

The evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify 5 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory. For the elliptic curve Curve25519, the evaluator implemented a tool that mimics what the CAVS tool provides for testing ciphers.

Assurance Activity AA-FCS_CKM.1-ATE-04

Key Generation for Finite-Field Cryptography (FFC)

The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p , the cryptographic prime q (dividing $p-1$), the cryptographic group generator g , and the calculation of the private key x and public key y .

The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p :

Cryptographic and Field Primes:

- Primes q and p shall both be provable primes
- Primes q and field prime p shall both be probable primes

and two ways to generate the cryptographic group generator g :

Cryptographic Group Generator:

- Generator g constructed through a verifiable process
- Generator g constructed through an unverifiable process

The Key generation specifies 2 ways to generate the private key x :

Private Key:

- $\text{len}(q)$ bit output of RBG where $1 \leq x \leq q-1$
- $\text{len}(q) + 64$ bit output of RBG, followed by a mod $q-1$ operation where $1 \leq x \leq q-1$

The security strength of the RBG must be at least that of the security offered by the FFC parameter set.

To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set.

For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and 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. Verification must also confirm

- $g \neq 0, 1$
- q divides $p-1$
- $g^q \bmod p = 1$
- $g^x \bmod p = y$

for each FFC parameter set and key pair.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

Assurance Activity AA-FCS_CKM.1-VPN-ATE-01

Refer to the Assurance Activity for FCS_CKM.1 in the MDF PP.

Testing for FFC Schemes using Diffie-Hellman group 14 is done as part of testing in CKM.2.1.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by CAVS testing.

2.1.2.2 Extended: Cryptographic Key Support (REK) (FCS_CKM_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.1-ASE-01

The evaluator shall review the TSS to determine that a REK is supported by the TOE, that the TSS includes a description of the protection provided by the TOE for a REK, and that the TSS includes a description of the method of generation of a REK.

The evaluator shall verify that the description of the protection of a REK describes how any reading, import, and export of that REK is prevented. (For example, if the hardware protecting the REK is removable, the description should include how other devices are prevented from reading the REK.) The evaluator shall verify that the TSS describes how encryption/decryption/derivation actions are isolated so as to prevent applications and system-level processes from reading the REK while allowing encryption/decryption/derivation by the key.

The evaluator shall verify that the description includes how the Rich OS is prevented from accessing the memory containing REK key material, which software is allowed access to the REK, how any other software in the execution environment is prevented from reading that key material, and what other mechanisms prevent the REK key material from being written to shared memory locations between the Rich OS and the separate execution environment.

If key derivation is performed using a REK, the evaluator shall ensure that the TSS description includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to FCS_CKM_EXT.3.2.

The evaluator shall verify that the generation of a REK meets the FCS_RBG_EXT.1.1 and FCS_RBG_EXT.1.2 requirements:

- *If REK(s) is/are generated on-device, the TSS shall include a description of the generation mechanism including what triggers a generation, how the functionality described by FCS_RBG_EXT.1 is invoked, and whether a separate instance of the RBG is used for REK(s).*
- *If REK(s) is/are generated off-device, the TSS shall include evidence that the RBG meets FCS_RBG_EXT.1. This will likely necessitate a second set of RBG documentation equivalent to the documentation provided for the RBG assurance activities. In addition, the TSS shall describe the manufacturing process that prevents the device manufacturer from accessing any REK(s).*

Summary

Section 8.2 of the [ST]  describes the *Hardware Protection Functions*. Section 8.3 of the [ST]  describes the *Cryptographic Support*.

Section 8.3.1 describes an overview of key management performed by the TOE. This description includes a description of a REK (Root Encryption Key, the TOE UID) used by the TOE. Section 8.2.1 describes that the TOE UID (the REK) is stored in hardware-protected memory in the Secure Enclave and is not accessible to any other part of the system. The UID is not even known by Apple. When the device starts up, an ephemeral key is created, entangled with the UID which is used to encrypt the Secure Enclave's portion of the device's memory space. The UID is never passed to any other part of the system than the Secure Enclave. The REK stored in the Secure Enclave is not accessible to the "regular" processor. Even the software within the Secure Enclave cannot read the REK. It can only request encryption/decryption operations performed by a dedicated Advanced Encryption Standard (AES) engine accessible only from the Secure Enclave. The UID is used to derive two other keys: the 0x89B and 0x835 keys. These keys are derived during boot by encrypting defined constants with the UID. These keys are used to wrap the EMF key, which is the file system master key, and the DKey, which is the device key. The AES Key Wrap algorithm is used to encrypt the 0x89B and 0x835 keys. The UID is not generated on the device, but rather at manufacturing time. It is generated in the production environment using a protected system with an RNG that complies with the requirements of NIST SP 800-90A (i.e., an Approved SP 800-90A DRBG implementation), as specified in FCS_RBG_EXT.1.1 and FCS_RBG_EXT.1.2.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.2.3 Cryptographic Key Establishment (FCS_CKM.2(1))

TSS Assurance Activities

Assurance Activity AA-FCS_CKM.2-1-ASE-01

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.

Summary

Section 6.2 in the [ST] [\[ST\]](#) defines the *Cryptographic Support (FCS)* SFRs implemented by the TOE. Section 8.3.1 in the [ST] [\[ST\]](#) describes an *Overview of Key Management* .

The evaluator reviewed FCS_CKM.1.1(1) {MDF} {VPN} {AGENT}, FCS_CKM.1.1/WLAN {WLAN}, FCS_CKM.1.1/VPN {VPN}, FCS_CKM.2.1(1) {MDF} {VPN} {AGENT}, FCS_CKM.2.1(2) {MDF} and FCS_CKM.2.1/WLAN {WLAN}, and verified that the supported key establishment schemes in FCS_CKM.2.1 correspond to the key generation schemes identified in FCS_CKM.1.1. Two key establishment schemes are defined: RSA based, FFC based, and elliptic curve based. Section 8.3.1 describes how the TOE performs key management. RSA, FCC, and elliptic curve based key establishment schemes can be used for Extensible Authentication Protocol (EAP)-TLS/TLS, Bluetooth, VPN or WLAN. Elliptic curve key establishment schemes are also used within the TOE key management.

Assurance Activity AA-FCS_CKM.2-1-ASE-02

SP800-56B Key Establishment Schemes

The evaluator shall verify that the TSS describes whether the TOE acts as a sender, a recipient, or both for RSA-based key establishment schemes.

Summary

Section 8.9 in the [ST] [\[ST\]](#) describes the *Trusted Path/Channels (FTP)* .

The evaluator reviewed section 8.9 and confirms that the TOE can act both as the sender and the receiver for RSA-based key establishment schemes.

Assurance Activity AA-FCS_CKM.2-1-ASE-03

SP800-56B Key Establishment Schemes

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.

Summary

Section 8.3.2 in the [ST] [\[ST\]](#) describes the *Storage of Persistent Secrets and Private Keys by the Agent*.

In the bulleted summary, it is explained that all decryption errors are handled in accordance with SP 800-56B.

Assurance Activity AA-FCS_CKM.2-1-VPN-ASE-02

For all key establishment schemes that conform to NIST SP 800-56A or 800-56B, refer to the assurance activity for FCS_CKM.2(1) in the MDF PP.

If "Key establishment scheme using Diffie-Hellman group 14..." is selected, the evaluator shall ensure that the TSS describes how the implementation meets RFC 3526 Section 3.

Summary

See AA-FCS_CKM.2-1-ASE-01.

Guidance Assurance Activities

Assurance Activity AA-FCS_CKM.2-1-AGD-01

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

Summary

[CCGUIDE] table 3 "SFR Configuration Requirements" identifies which SFR can be or needs to be configured in the evaluated configuration. For FCS_CKM.2, there is no configuration necessary for the selection of key establishment schemes by the TOE users or administrators. There are, however, API available to app developers which allow specifications of the requested key sizes and key types.

Test Assurance Activities

Assurance Activity AA-FCS_CKM.2-1-ATE-01

Assurance Activity Note: *The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.*

Key Establishment Schemes

The evaluator shall verify the implementation of the key establishment schemes supported by the TOE using the applicable tests below.

SP800-56A Key Establishment Schemes

The evaluator shall verify a TOE's 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 DLC 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 TOE's 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 TOE's 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 TOE's 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 TOE's results with the results using a known good implementation verifying that the TOE detects these errors.

SP800-56B Key Establishment Schemes

If the TOE acts as a sender, the following assurance activity shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:

To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with or without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTS-OAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA public key, the plaintext keying material, any additional input parameters if applicable, the MacKey and MacTag if key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform a key establishment encryption operation on the TOE with the same inputs (in cases where key confirmation is incorporated, the test shall use the MacKey from the test vector instead of the randomly generated MacKey used in normal operation) and ensure that the outputted ciphertext is equivalent to the ciphertext in the test vector.

If the TOE acts as a receiver, the following assurance activities shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:

To conduct this test the evaluator shall generate or obtain test vectors FCS_CKM.2.1(2) from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with or without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTS-OAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA private key, the plaintext keying material (KeyData), any additional input parameters if applicable, the MacTag in cases where key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform the key establishment decryption operation on the TOE and ensure that the outputted plaintext keying material (KeyData) is equivalent to the plaintext keying material in the test vector. In cases where key confirmation is incorporated, the evaluator shall perform the key confirmation steps and ensure that the outputted MacTag is equivalent to the MacTag in the test vector.

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. 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. If KTS-KEMKWS 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[4\]](#).

This test is covered by the CAVS testing performed by the laboratory.

Assurance Activity AA-FCS_CKM.2-1-VPN-ATE-02

For all key establishment schemes that conform to NIST SP 800-56A or 800-56B, refer to the Assurance Activity for FCS_CKM.2(1) in the MDF PP.

[If "Key establishment scheme using Diffie-Hellman group 14..." is selected, the evaluator shall ensure that the TSS describes how the implementation meets RFC 3526 Section 3.]

The evaluator shall also verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_ITC_EXT.1 in the MDF PP that uses Diffie-Hellman group 14. Note that because a TOE that conforms to this PP-Module must implement IPsec, the tested protocols shall include IPsec at minimum.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[4\]](#).

This test is covered by CAVS testing. Note, for RSA SP800-56B, we received an email from a validator on December 12, 2014 which implies that the RSA2VS CAVS testing is to be used.

2.1.2.4 Cryptographic Key Establishment (While device is locked) (FCS_CKM.2(2))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_CKM.2-2-ATE-01

The test for SP800-56A and SP800-56B key establishment schemes is performed in association with FCS_CKM.2.1(1).

Curve22519 Key Establishment Schemes

The evaluator shall verify a TOE's implementation of the key agreement scheme 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 specification. These components include the calculation of the shared secret K and the hash of K.

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 role and hash function combination, the tester shall generate 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

The evaluator shall obtain the shared secret value K, and the hash of K.

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 K and compare the hash generated from this value.

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results. To conduct this test, the evaluator generates a set of 30 test vectors consisting of data sets including the evaluator's public keys and the TOE's public/private key pairs.

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 K or the hash of K. 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 TOE's results with the results using a known good implementation verifying that the TOE detects these errors.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory. For the elliptic curve Curve25519, the evaluator implemented a tool that mimics what the CAVS tool provides for testing ciphers.

2.1.2.5 Extended: Cryptographic Key Random Generation (FCS_CKM_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.2-ASE-01

[TD0351]The evaluator shall examine the key hierarchy section of the TSS to ensure that the formation of all DEKs is described and that the key sizes match that described by the ST author. The evaluator shall examine the key hierarchy section of the TSS to ensure that each DEK is generated or combined from keys of equal or greater security strength using one of the selected methods.

- If the symmetric DEK is generated by an RBG, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT.1 or documentation available for the operational environment to determine that the key size being requested is greater than or equal to the key size and mode to be used for the encryption/decryption of the data.
- If the DEK is formed from a combination, the evaluator shall verify that the TSS describes the method of combination and that this method is either an XOR, or a KDF.
- If "concatenating the keys and using a KDF (as described in (SP 800-56C))" is selected, the evaluator shall ensure the TSS includes a description of the randomness extraction step.

The description must include how an approved untruncated MAC function is being used for the randomness extraction step and the evaluator must verify the TSS describes that the output length (in bits) of the MAC function is at least as large as the targeted security strength (in bits) of the parameter set employed by the key establishment scheme (see Tables 1-3 of SP 800-56C).

The description must include how the MAC function being used for the randomness extraction step is related to the PRF used in the key expansion and verify the TSS description includes the correct MAC function:

- If an HMAC-hash is used in the randomness extraction step, then the same HMAC-hash (with the same hash function hash) is used as the PRF in the key expansion step.
- If an AES-CMAC (with key length 128, 192, or 256 bits) is used in the randomness extraction step, then AES-CMAC with a 128-bit key is used as the PRF in the key expansion step.
- The description must include the lengths of the salt values being used in the randomness extraction step and the evaluator shall verify the TSS description includes correct salt lengths:
- If an HMAC-hash is being used as the MAC, the salt length can be any value up to the maximum bit length permitted for input to the hash function hash.
- If an AES-CMAC is being used as the MAC, the salt length shall be the same length as the AES key (i.e. 128, 192, or 256 bits).

Summary

Section 8.3.1 of the [ST][1](#) describes the *Overview of Key Management*.

Every time a file is created, a new 256-bit AES key is generated by the hardware random number generator and post-processed by a SP 800-90A CTR-based Deterministic Random Bit Generator (DRBG). This key is used to encrypt the file it was generated for using AES-256-CBC, where the IV is calculated with the block offset into the file and then hashed with the Secure Hash Algorithm (SHA)-1 algorithm.

Assurance Activity AA-FCS_CKM_EXT.2-ASE-02

[TD0351] (conditional) If a KDF is used, the evaluator shall ensure that the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108 or SP 800-56C.

Summary

Section 8.3.3 *Randomness extraction step* in the [ST][1](#) describes that steps used for key derivation. The steps are copied over from SP 800-56C and RFC5869.

Guidance Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.2-AGD-01

[TD0351] The evaluator uses the description of the RBG functionality in FCS_RBG_EXT.1 or documentation available for the operational environment to determine that the key size being generated or combined is identical to the key size and mode to be used for the encryption/decryption of the data.

Summary

[CCGUIDE][1](#) table 3 "SFR Configuration Requirements" identifies which SFR can be or needs to be configured in the evaluated configuration. For FCS_CKM_EXT.2, there is no configuration necessary by the TOE users or administrators since generation and maintenance of DEK is hard coded.

Test Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.2-ATE-01

[TD0351] If a KDF is used, the evaluator shall perform one or more of the following tests to verify the correctness of the key derivation function, depending on the mode(s) that are supported. The following table maps the data fields to the notations used in SP 800-108 and SP 800-56C.

Data Fields	Notations	
	SP 800-108	SP 800-56C
Pseudorandom function	PRF	PRF
Counter length	r	r
Length of output of PRF	h	h
Length of derived keying material	L	L
Length of input values	I length	I length
Pseudorandom input values I	$K1$ (key derivation key)	Z (shared secret)
Pseudorandom salt values	n/a	s
Randomness extraction MAC	n/a	MAC

Counter Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function: - One or more pseudorandom functions that are supported by the implementation (PRF).

- One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
- The length (in bits) of the output of the PRF (h). - Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h .
- Up to two values of L that are NOT evenly divisible by h .
- Location of the counter relative to fixed input data: before, after, or in the middle.
 - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
- The length (I _length) of the input values I .

For each supported combination of I _length, MAC, salt, PRF, counter location, value of r , and value of L , the evaluator shall generate 10 test vectors that include pseudorandom input values I , and pseudorandom salt values. If there is only one value of L that is evenly divisible by h , the evaluator shall generate 20 test vectors for it. For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output.

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.

Feedback Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h .
- Up to two values of L that are NOT evenly divisible by h .
- Whether or not zero-length IVs are supported.
- Whether or not a counter is used, and if so:
 - One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
 - Location of the counter relative to fixed input data: before, after, or in the middle.
 - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.

- Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.

- The length (I_length) of the input values I .

For each supported combination of I_length , MAC, salt, PRF, counter location (if a counter is used), value of r (if a counter is used), and value of L , the evaluator shall generate 10 test vectors that include pseudorandom input values I and pseudorandom salt values. If the KDF supports zero-length IVs, five of these test vectors will be accompanied by pseudorandom IVs and the other five will use zero-length IVs. If zero-length IVs are not supported, each test vector will be accompanied by a pseudorandom IV. If there is only one value of L that is evenly divisible by h , the evaluator shall generate 20 test vectors for it.

For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output. 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.

Double Pipeline Iteration Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h .
- Up to two values of L that are NOT evenly divisible by h .
- Whether or not a counter is used, and if so:
 - One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
 - Location of the counter relative to fixed input data: before, after, or in the middle.
 - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
 - The length (I_length) of the input values I .

For each supported combination of I_length , MAC, salt, PRF, counter location (if a counter is used), value of r (if a counter is used), and value of L , the evaluator shall generate 10 test vectors that include pseudorandom input values I , and pseudorandom salt values. If there is only one value of L that is evenly divisible by h , the evaluator shall generate 20 test vectors for it.

For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output. 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

2.1.2.6 Extended: Cryptographic Key Generation (FCS_CKM_EXT.3)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.3-ASE-01

[TD0366] The evaluator shall examine the key hierarchy section of the TSS to ensure that the formation of all KEKs is described and that the key sizes match that described by the ST author. The evaluator shall examine the key hierarchy section of the TSS to ensure that each key (DEKs, software-based key storage, and KEKs) is encrypted by keys of equal or greater security strength using one of the selected methods.

- The evaluator shall review the TSS to verify that it contains a description of the conditioning use to derive KEKs. This description must include the size and storage location of salts. This activity may be performed in combination with that for FCS_COP.1(5).
- If the symmetric KEK is generated by an RBG, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT.1 or documentation available for the operational environment to determine that the key size being requested is greater than or equal to the key size and mode to be used for the encryption/decryption of the data.
- If the KEK is generated according to an asymmetric key scheme, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_CKM.1 is invoked. The evaluator uses the description of the key generation functionality in FCS_CKM.1 or documentation available for the operational environment to determine that the key strength being requested is greater than or equal to 112 bits.
- If the KEK is formed from a combination, the evaluator shall verify that the TSS describes the method of combination and that this method is either an XOR, a KDF, or encryption.

Summary

Section 8.3.1 in the [ST] describes the *Overview of Key Management*.

The evaluator reviewed section 8.3.1 and verified that all keys used for data encryption are 256-bit AES keys. Key Encryption Keys (KEKs) are also 256-bit AES keys. The passcode key is derived from the UID, the passcode and the Salt using the Concatenation Key Derivation Function (Approved Alternative 1) as described in 5.8.1 of NIST SP 800-56A. The KEKs are generated using an Approved NIST SP 800-90A as described in FCS_RBG_EXT.1.

Assurance Activity AA-FCS_CKM_EXT.3-ASE-02

(conditional) If a KDF is used, the evaluator shall ensure that the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108.

Summary

Although KDF is used, the SP 800-56C KDF is used and not the SP 800-108 KDF. See AA-FCS_CKM_EXT.3-ASE-03

Assurance Activity AA-FCS_CKM_EXT.3-ASE-03

If “concatenating the keys and using a KDF (as described in (SP 800-56C))” is selected, the evaluator shall ensure the TSS includes a description of the randomness extraction step.

- The description must include how an approved untruncated MAC function is being used for the randomness extraction step and the evaluator must verify the TSS describes that the output length (in bits) of the MAC function is at least as large as the targeted security strength (in bits) of the parameter set employed by the key establishment scheme (see Tables 1-3 of SP 800-56C).
- The description must include how the MAC function being used for the randomness extraction step is related to the PRF used in the key expansion and verify the TSS description includes the correct MAC function:
 - If an HMAC-hash is used in the randomness extraction step, then the same HMAC-hash (with the same hash function hash) is used as the PRF in the key expansion step.
 - If an AES-CMAC (with key length 128, 192, or 256 bits) is used in the randomness extraction step, then AES-CMAC with a 128-bit key is used as the PRF in the key expansion step.
- The description must include the lengths of the salt values being used in the randomness extraction step and the evaluator shall verify the TSS description includes correct salt lengths:

- If an HMAC-hash is being used as the MAC, the salt length can be any value up to the maximum bit length permitted for input to the hash function hash.
- If an AES-CMAC is being used as the MAC, the salt length shall be the same length as the AES key (i.e. 128, 192, or 256 bits).

Summary

The evaluator notes that a Key Derivation Function (KDF) as defined in SP 800-56C is used, which is based on RFC5869. The KDF defined in the RFC complies with SP 800-56C, as far as the extraction and expansions steps. The RFC specifies the order of the concatenation. The implementation of the KDF uses HMAC-SHA-256.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.3-ATE-01

If a KDF is used, the evaluator shall perform one or more of the following tests to verify the correctness of the key derivation function, depending on the mode(s) that are supported. The following table maps the data fields to the notations used in SP 800-108 and SP 800-56C.

Data Fields	Notations	
	SP 800-108	SP 800-56C
Pseudorandom function	PRF	PRF
Counter length	r	r
Length of output of PRF	h	h
Length of derived keying material	L	L
Length of input values	I_length	I_length
Pseudorandom input values I	K_1 (key derivation key)	Z (shared secret)
Pseudorandom salt values	n/a	s
Randomness extraction MAC	n/a	MAC

Counter Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h .
- Up to two values of L that are NOT evenly divisible by h .
- Location of the counter relative to fixed input data: before, after, or in the middle.
 - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.

- The length (I_length) of the input values I .

For each supported combination of I_length , MAC, salt, PRF, counter location, value of r , and value of L , the evaluator shall generate 10 test vectors that include pseudorandom input values I , and pseudorandom salt values. If there is only one value of L that is evenly divisible by h , the evaluator shall generate 20 test vectors for it. For each test vector, the evaluator shall supply this data to the TO E in order to produce the keying material output.

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.

Feedback Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h .
- Up to two values of L that are NOT evenly divisible by h .
- Whether or not zero-length IVs are supported.
- Whether or not a counter is used, and if so:
 - One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
 - Location of the counter relative to fixed input data: before, after, or in the middle.
 - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
 - The length (I_length) of the input values I .

For each supported combination of I_length , MAC, salt, PRF, counter location (if a counter is used), value of r (if a counter is used), and value of L , the evaluator shall generate 10 test vectors that include pseudorandom input values I and pseudorandom salt values. If the KDF supports zero-length IVs, five of these test vectors will be accompanied by pseudorandom IVs and the other five will use zero-length IVs. If zero-length IVs are not supported, each test vector will be accompanied by an pseudorandom IV. If there is only one value of L that is evenly divisible by h , the evaluator shall generate 20 test vectors for it.

For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output. 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.

Double Pipeline Iteration Mode Tests:

The evaluator shall determine the following characteristics of the key derivation function:

- One or more pseudorandom functions that are supported by the implementation (PRF).
- The length (in bits) of the output of the PRF (h).
- Minimum and maximum values for the length (in bits) of the derived keying material (L). These values can be equal if only one value of L is supported. These must be evenly divisible by h .
- Up to two values of L that are NOT evenly divisible by h .
- Whether or not a counter is used, and if so:
 - One or more of the values {8, 16, 24, 32} that equal the length of the binary representation of the counter (r).
 - Location of the counter relative to fixed input data: before, after, or in the middle.
 - Counter before fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter after fixed input data: fixed input data string length (in bytes), fixed input data string value.
 - Counter in the middle of fixed input data: length of data before counter (in bytes), length of data after counter (in bytes), value of string input before counter, value of string input after counter.
- The length (I_length) of the input values I .

For each supported combination of I_length , MAC, salt, PRF, counter location (if a counter is used), value of r (if a counter is used), and value of L , the evaluator shall generate 10 test vectors that include pseudorandom input values I , and pseudorandom salt values. If there is only one value of L that is evenly divisible by h , the evaluator shall generate 20 test vectors for it.

For each test vector, the evaluator shall supply this data to the TOE in order to produce the keying material output. 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

2.1.2.7 Extended: Key Destruction (FCS_CKM_EXT.4)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.4-ASE-01

The evaluator shall check to ensure the TSS lists each type of plaintext key material (DEKs, software-based key storage, KEKs, trusted channel keys, passwords, etc.) and its generation and storage location.

The evaluator shall verify that the TSS describes when each type of key material is cleared (for example, on system power off, on wipe function, on disconnection of trusted channels, when no longer needed by the trusted channel per the protocol, when transitioning to the locked state, and possibly including immediately after use, while in the locked state, etc.).

The evaluator shall also verify that, for each type of key, the type of clearing procedure that is performed (cryptographic erase, overwrite with zeros, overwrite with random pattern, or block erase) 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.

Summary

Section 8.3.1 in the [ST] describes the Overview of Key Management .

Table 6: Summary of keys and persistent secrets in iOS 12 and Table 7: Summary of keys and persistent secrets used by the Agent Summary of keys and persistent secrets in iOS 12 and Summary of keys and persistent secrets used by the Agent in section 8.3.1 describe all the key material, their purposes and storage locations. Keys that are encrypted are not cleared unless the data they protect is erased. The DKey and the EMF key are stored wrapped, in block 0 of the flash memory (effaceable storage) and thus erasable very quickly if needed. The UID is stored unencrypted but is not accessible by any other part of the system than the Secure Enclave. All other keys are stored in non-volatile memory , wrapped. The TSS states that all clearing operations of keys stored in volatile memory are performed by Apple iOS CoreCrypto Kernel Module for ARM and Apple iOS CoreCrypto Module for ARM . The TSS specifies that this clearing is performed using the memset(0) function to overwrite the memory containing keys and sensitive parameters. That applies to symmetric keys used for TLS, HTTPS, or Wi-Fi. All other keys are stored encrypted and therefore do not need to be wiped. When a wipe is triggered, the KEK are erased and therefore all other encrypted keys do not need be zeroized.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.4-ATE-01

Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

For each software and firmware key clearing situation (including on system power off, on wipe function, on disconnection of trusted channels, when no longer needed by the trusted channel per the protocol, when transitioning to the locked state, and possibly including immediately after use, while in the locked state) the evaluator shall repeat the following tests.

For these tests the evaluator shall utilize appropriate development environment (e.g. a Virtual Machine) and development tools (debuggers, simulators, etc.) to test that keys are cleared, including all copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.

- **Test 1:** Applied to each key held as plaintext in volatile memory and subject to destruction by overwrite by the TOE (whether or not the plaintext value is subsequently encrypted for storage in volatile or non-volatile memory). In the case where the only selection made for the destruction method key was removal of power, then this test is unnecessary. The evaluator shall:

1. Record the value of the key in the TOE subject to clearing.
2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
3. Cause the TOE to clear the key.
4. Cause the TOE to stop the execution but not exit.
5. Cause the TOE to dump the entire memory of the TOE into a binary file.
6. Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.
7. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.

Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.

Step 7 ensures that partial key fragments do not remain in memory. If a fragment is found, there is a minuscule chance that it is not within the context of a key (e.g., some random bits that happen to match). If this is the case the test should be repeated with a different key in Step #1. If a fragment is found the test fails.

- **Test 2:** Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:

1. Record the value of the key in the TOE subject to clearing.
2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
3. Cause the TOE to clear the key.
4. Search the non-volatile memory the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.
5. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece. If a fragment is found then the test is repeated (as described for test 1 above), and if a fragment is found in the repeated test then the test fails.

- **Test 3:** Applied to each key held as non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use special tools (as needed), provided by the TOE developer if necessary, to view the key storage location:

1. Record the storage location of the key in the TOE subject to clearing.
2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
3. Cause the TOE to clear the key.
4. Read the storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized.

The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test has been performed as part of the FIPS 140-2 conformance testing performed in parallel on the TOE. As part of FIPS 140-2, the zeroization testing is required. As all software requiring cryptographic support use CoreCrypto, the zeroization tests performed as part of FIPS 140-2 cover this aspect. This approach was discussed with and approved by NIAP for the previous iOS 9, iOS 10 and iOS 11 evaluations.

2.1.2.8 Extended: TSF Wipe (FCS_CKM_EXT.5)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.5-ASE-01

The evaluator shall check to ensure the TSS describes how the device is wiped; and the type of clearing procedure that is performed (cryptographic erase or overwrite) and, if overwrite is performed, the overwrite procedure (overwrite with zeros, overwrite three or more times by a different alternating pattern, overwrite with random pattern, or block erase). If different types of memory are used to store the data 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, data stored on flash are cleared by overwriting once with zeros, while data stored on the internal persistent storage device are cleared by overwriting three times with a random pattern that is changed before each write).

Summary

Section 8.3.1 in the [ST] describes the *Overview of Key Management*.

This section specifies that

"when a wipe command is issued, protected data is wiped by erasing the top level KEKs. Since all data-at-rest is encrypted with one of those keys, the device is wiped."

The overwriting of the KEK is done through a `memset(0)` operation that overwrites the memory (see previous assurance activity).

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.5-ATE-01

Assurance Activity Note: The following test may require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

The evaluator shall perform one of the following tests. The test before and after the wipe command shall be identical. This test shall be repeated for each type of memory used to store the data to be protected.

Method 1 for File-based Methods:

- **Test 1:** The evaluator shall enable encryption according to the AGD guidance. The evaluator shall create a user data (protected data or sensitive data) file, for example, by using an application. The evaluator shall use a tool provided by the developer to examine this data stored in memory (for example, by examining a decrypted file). The evaluator shall initiate the wipe command according to the AGD guidance provided for FMT_SMF_EXT.1. The evaluator shall use a tool provided by the developer to examine the same data location in memory to verify that the data has been wiped according to the method described in the TSS (for example, the files are still encrypted and cannot be accessed).

Method 2 for Volume-based Methods:

- **Test 1:** *The evaluator shall enable encryption according to the AGD guidance. The evaluator shall create a unique data string, for example, by using an application. The evaluator shall use a tool provided by the developer to search decrypted data for the unique string. The evaluator shall initiate the wipe command according to the AGD guidance provided for FMT_SMF_EXT.1. The evaluator shall use a tool provided by the developer to search for the same unique string in decrypted memory to verify that the data has been wiped according to the method described in the TSS (for example, the files are still encrypted and cannot be accessed).*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[1\]](#).

Reading of the effaceable memory is not possible. The storage controller does not provide the ability to read that storage. Effaceable storage is implemented as follows: during first initialization time (or after complete erase of all data), the controller is asked to reserve a new effaceable storage area. The software will not receive the pointer to the memory, but rather a handle. XNU/SEP uses that handle for access. But the storage controller does not return a precise pointer to the storage location. Therefore, there is no way to read out the memory directly. The debugging of the AppleKeystore contents done for FCS_CKM_EXT.4 after a complete wipe of the device shows that new keys are generated by SEP and therefore loaded onto the new effaceable memory. This approach was approved by NIAP during the previous evaluations of iOS 9, iOS 10, and iOS 11.

2.1.2.9 Extended: Salt Generation (FCS_CKM_EXT.6)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.6-ASE-01

The evaluator shall verify that the TSS contains a description regarding the salt generation, including which algorithms on the TOE require salts. The evaluator shall confirm that the salt is generated using an RBG described in FCS_RBG_EXT.1. For PBKDF derivation of KEKs, this assurance activity may be performed in conjunction with FCS_CKM_EXT.3.2.

Summary

Section 8.2 in the [ST][\[1\]](#) describes the *Hardware Protection Functions*.

The salt used for the password-based key derivation function 2 (PBKDF2) uses the Secure Enclave physical noise source, which uses AES with device UID as the key for the pseudo-random function. This salt is regenerated every time that the password changes. All other salt values used in iOS are generated using the True Random Number Generator of the application processor, post-processed by a CTR-based SP 800-90A DRBG implementation.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.2.10 Extended: Cryptographic Key Support (REK) (FCS_CKM_EXT.7)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.7-ASE-01

The assurance activity for this element is performed in conjunction with the assurance activity for FCS_CKM_EXT.1.

Summary

Please refer to AA-FCS_CKM_EXT.1-ASE-01 .

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.2.11 Cryptographic operation (Confidentiality Algorithms) (FCS_COP.1(1))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_COP.1-1-ATE-01

Assurance Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

AES-CBC Tests

- **Test 1:** AES-CBC Known Answer Tests

There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. 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.

- **Test 1.1:** KAT-1. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.
To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.
- **Test 1.2:** KAT-2. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.
To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.

- **Test 1.3: KAT-3.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost $N-i$ bits be zeros, for i in $[1, N]$.
To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost $N-i$ bits be zeros, for i in $[1, N]$. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.
- **Test 1.4: KAT-4.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost $128-i$ bits be zeros, for i in $[1, 128]$.
To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.
- **Test 2: AES-CBC Multi-Block Message Test**
The evaluator shall test the encrypt functionality by encrypting an i -block message where $1 < i \leq 10$. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.
The evaluator shall also test the decrypt functionality for each mode by decrypting an i -block message where $1 < i \leq 10$. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.
- **Test 3: AES-CBC Monte Carlo Tests**
The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

```
# Input: PT, IV, Key
for i = 1 to 1000:
  if i == 1:
    CT[1] = AES-CBC-Encrypt(Key, IV, PT)
    PT = IV
  else:
    CT[i] = AES-CBC-Encrypt(Key, PT)
    PT = CT[i-1]
```

The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation. The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

AES-CCM Tests

- **Test 1:** The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:

128 bit and 256 bit keys

Two payload lengths. One payload length shall be the shortest supported payload length, greater than or equal to zero bytes. The other payload length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits).

Two or three associated data lengths. One associated data length shall be 0, if supported. One associated data length shall be the shortest supported payload length, greater than or equal to zero bytes. One associated data length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 2^{16} bytes, an associated data length of 2^{16} bytes shall be tested.

Nonce lengths. All supported nonce lengths between 7 and 13 bytes, inclusive, shall be tested.

Tag lengths. All supported tag lengths of 4, 6, 8, 10, 12, 14 and 16 bytes shall be tested.

To test the generation-encryption functionality of AES-CCM, the evaluator shall perform the following four tests:

- **Test 1.1:** For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.
- **Test 1.2:** For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.
- **Test 1.3:** For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator shall supply one key value and 10 associated data, payload and nonce value 3-tuples and obtain the resulting ciphertext.
- **Test 1.4:** For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

To determine correctness in each of the above tests, the evaluator shall compare the ciphertext with the result of generation-encryption of the same inputs with a known good implementation.

To test the decryption-verification functionality of AES-CCM, for EACH combination of supported associated data length, payload length, nonce length and tag length, the evaluator shall supply a key value and 15 nonce, associated data and ciphertext 3-tuples and obtain either a FAIL result or a PASS result with the decrypted payload. The evaluator shall supply 10 tuples that should FAIL and 5 that should PASS per set of 15.

AES-GCM Test

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.

- **Test 1:** 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.
- **Test 2:** 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.

XTS-AES Test

- **Test 1:** The evaluator shall test the encrypt functionality of XTS-AES for each combination of the following input parameter lengths:

256 bit (for AES-128) and 512 bit (for AES-256) keys

Three data unit (i.e., plaintext) lengths. One of the data unit lengths shall be a nonzero integer multiple of 128 bits, if supported. One of the data unit lengths shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 2^{16} bits, whichever is smaller.

using a set of 100 (key, plaintext and 128-bit random tweak value) 3-tuples and obtain the ciphertext that results from XTS-AES encrypt.

The evaluator may supply a data unit sequence number instead of the tweak value if the implementation supports it. The data unit sequence number is a base-10 number ranging between 0 and 255 that implementations convert to a tweak value internally.

- **Test 2:** The evaluator shall test the decrypt functionality of XTS-AES using the same test as for encrypt, replacing plaintext values with ciphertext values and XTS-AES encrypt with XTS-AES decrypt.

AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test

- **Test 1:** The evaluator shall test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths:

128 and 256 bit key encryption keys (KEs)

Three plaintext lengths. One of the plaintext lengths shall be two semi-blocks (128 bits). One of the plaintext lengths shall be three semi-blocks (192 bits). The third data unit length shall be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).

using a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator shall use the AES-KW authenticated-encryption function of a known good implementation.

- **Test 2:** The evaluator shall test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption.
- **Test 3:** The evaluator shall test the authenticated-encryption functionality of AES-KWP using the same test as for AES-KW authenticated-encryption with the following change in the three plaintext lengths:
 - One plaintext length shall be one octet. One plaintext length shall be 20 octets (160 bits).
 - One plaintext length shall be the longest supported plaintext length less than or equal to 512 octets (4096 bits).
- **Test 4:** The evaluator shall test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[1\]](#).

This test is covered by the CAVS testing performed by the laboratory.

2.1.2.12 Cryptographic operation (Hashing Algorithms) (FCS_COP.1(2))**TSS Assurance Activities****Assurance Activity AA-FCS_COP.1-2-ASE-01**

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.

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.

Summary

Section 8.3 in the [ST][\[1\]](#) describes the *Cryptographic Support*.

The Cryptographic Algorithm Validation System (CAVS) certificate entry specifies whether the implementation has been tested in byte-oriented or bit-oriented mode and which hash functions have been tested. The evaluator reviewed the CAVS certificates and verified that the mode tested is byte-oriented messages. The hash functions also accept empty length messages.

Guidance Assurance Activities

Assurance Activity AA-FCS_COP.1-2-AGD-01

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.

Summary

[CCGUIDE] table 3 "SFR Configuration Requirements" identifies which SFR can be or needs to be configured in the evaluated configuration. For FCS_COP.1(2), there is no configuration necessary by the TOE users or administrators for TLS and EAP-TLS connections, however, for VPN connections with IKEv2 configuration is possible via a Configuration Profile. Additional guidance is provided in [CCGUIDE] section 3.2.3 *Hashing* which states the following:

- For TLS and EAP-TLS connections, functions to perform hashing are provided as part of the Apple CoreCrypto libraries. The invoking function dictates which SHA function is used. Neither the mobile device user nor the mobile device administrator has the ability to configure this choice.
- Similarly, each TLS ciphersuite uses a specific and appropriate SHA function. Neither the mobile device user nor the mobile device administrator has the ability to configure this choice.
- For VPN connections with IKEv2, the integrity algorithm to be used is selectable by the mobile device administrator by setting the IntegrityAlgorithm key. Note that setting IntegrityAlgorithm to SHA1-96 is not allowed in the evaluated configuration.

Test Assurance Activities

Assurance Activity AA-FCS_COP.1-2-ATE-01

The TSF may implement either bit-oriented or byte-oriented; both implementations are not required. 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.

Assurance Activity Note: *The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.*

- **Test 1: Short Messages Test: Bit-oriented Mode**
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 ranges 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.
- **Test 2: Short Messages Test: Byte-oriented Mode**
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.
- **Test 3: Selected Long Messages Test: Bit-oriented Mode**
*The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the i^{th} message is $512 + 99*i$, where $1 \leq i \leq m$. 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.*
- **Test 4: Selected Long Messages Test: Byte-oriented Mode**
*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 i^{th} message is $512 + 8*99*i$, where $1 \leq i \leq m/8$. 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.*

- **Test 5: Pseudorandomly Generated Messages Test**
This test is for byteoriented 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

2.1.2.13 Cryptographic operation (Signature Algorithms) (FCS_COP.1(3))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_COP.1-3-ATE-01

Assurance Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

- **Test 1: ECDSA Algorithm Tests**
 - **Test 1.1: ECDSA FIPS 186-4 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 1.2: ECDSA FIPS 186-4 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.
- **Test 2: RSA Signature Algorithm Tests**
 - **Test 2.1: 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 2.2: 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

2.1.2.14 Cryptographic operation (Keyed Hash Algorithms) (FCS_COP.1(4))

TSS Assurance Activities

Assurance Activity AA-FCS_COP.1-4-ASE-01

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.

Summary

Section 8.3 in the [ST] describes the *Cryptographic Support*.

The TOE has received CAVS certificates for the Hash Message Authentication Code (HMAC) algorithms. The CAVS certificate entry specifies the size of the HMAC key, output MAC length, hash function used, and block size. The evaluator reviewed the CAVS Certificates received for the HMAC algorithm, and could find that the hash functions used are SHA-1, SHA-256, SHA-384, and SHA-512; the key size is < block size, = block size and > block size. The SFR specifically defines the size of the keys, output size. The block size is as defined by the HMAC and SHS standard.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_COP.1-4-ATE-01

Assurance Activity Note: *The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.*

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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

2.1.2.15 Cryptographic operation (Password-based Key Derivation Functions) (FCS_COP.1(5))

TSS Assurance Activities

Assurance Activity AA-FCS_COP.1-5-ASE-01

[TD0366] The evaluator shall check that the TSS describes the method by which the password is first encoded and then fed to the SHA algorithm and verify the SHA algorithm matches the first selection.

If a key stretching function, such as PBKDF2, is selected the settings for the algorithm (padding, blocking, etc.) shall be described. The evaluator shall verify that the TSS contains a description of how the output of the hash function or key stretching function is used to form the submask that will be input into the function and is the same length as the KEK as specified in FCS_CKM_EXT.3.

If any manipulation of the key is performed in forming the submask that will be used to form the KEK, that process shall be described in the TSS.

No explicit testing of the formation of the submask from the input password is required.

Summary

Section 8.3.1.1 in the [ST] describes the Password based key derivation .

The PBKDF2 function is used to derive the passcode key from the passcode, the UID and the salt. The number of iterations is set to take at least 100 to 150 milliseconds (ms), which means that the number of iterations performed is dependent upon how many iterations can be performed within this time frame. The description of how the password is encoded and then fed to the SHA-256 algorithm for the PBKDF2 function, as described in NIST SP 800-132, option 2.b.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.2.16 Extended: HTTPS Protocol (FCS_HTTPS_EXT.1)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_HTTPS_EXT.1-ATE-01

- **Test 1:** The evaluator shall attempt to establish an HTTPS connection with a webserver, 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 FCS_TLSC_EXT.1. Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test:

- **Test 2:** The evaluator shall demonstrate that using a certificate without a valid certification path results in an application notification. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database 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, and show that the application is notified of the validation failure.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator loaded a configuration profile on the TOE containing all the information necessary to connect to the WiFi Access Point (AP). The evaluator started a TLS web server and connected to TOE to it, and verified using Wireshark that the connection was successful. When opening a web browser and going to https://taun, the evaluator could verify that the openssl s_server status is displayed.

Test 2: the evaluator reproduced the procedure in Test 1, but accessed https://server instead. The evaluator was prompted the option to accept or deny the certificate.

Test 3: the evaluator disabled the trust for the certificate in Settings -> General -> About -> Certificate Trust Settings, and reperformed Test 1. The evaluator verified that the warning about the certificate is displayed once again.

2.1.2.17 Extended: Initialization Vector Generation (FCS_IV_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FCS_IV_EXT.1-ASE-01

The evaluator shall examine the key hierarchy section of the TSS to ensure that the encryption of all keys is described and the formation of the IVs for each key encrypted by the same KEK meets FCS_IV_EXT.1.

Summary

Section 8.3.1 in the [ST] describes the *Overview of Key Management*.

The evaluator reviewed section 8.3.1 and found AES key wrapping according to RFC3394 is used for key wrapping. AES key wrapping according to RFC3394 does not required the formation of Initialization Vectors (IVs).

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.2.18 Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FCS_RBG_EXT.1-ASE-01

Documentation shall be produced and the evaluator shall perform the activities in accordance with Appendix D, the "Clarification to the Entropy Documentation and Assessment".

The evaluator shall verify that the API documentation provided according to Section 5.2.2, includes the security functions described in FCS_RBG_EXT.1.3.

Summary

Section 2.1 in the [EAR] [\[1\]](#) addresses this assurance activity, as it describes the entropy source high level design, which includes the security functions described in FCS_RBG_EXT.1.3.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_RBG_EXT.1-ATE-01

Assurance Activity Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

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 (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 SP800-90A).

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 (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 \leq 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[1\]](#).

This test is covered by the CAVS testing performed by the laboratory.

2.1.2.19 Extended: Cryptographic Algorithm Services (FCS_SRV_EXT.1)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FCS_SRV_EXT.1-AGD-01

The evaluator shall verify that the API documentation provided according to Section 5.2.2 includes the security functions (cryptographic algorithms) described in these requirements.

Summary

The evaluator examined the provided API documentation [CKTSREF] (e.g., sections *Keys*, *SecKeyAlgorithm*) and verified that they includes the necessary description of the security functions/cryptographic algorithms.

Test Assurance Activities

Assurance Activity AA-FCS_SRV_EXT.1-ATE-01

The evaluator shall write, or the developer shall provide access to, an application that requests cryptographic operations by the TSF. The evaluator shall verify that the results from the operation match the expected results according to the API documentation. This application may be used to assist in verifying the cryptographic operation assurance activities for the other algorithm services requirements.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

This test is covered by the CAVS testing performed by the laboratory.

2.1.2.20 Extended: Cryptographic Key Storage (FCS_STG_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FCS_STG_EXT.1-ASE-01

The evaluator shall review the TSS to determine that the TOE's implements the required secure key storage. The evaluator shall ensure that the TSS contains a description of the key storage mechanism that justifies the selection of "mutable hardware" or "software-based".

Summary

Section 8.3.1 in the [ST] describes the *Overview of Key Management* . Section 8.3.2 describes the *Storage of Persistent Secrets and Private Keys by the Agent* .


The evaluator verified in FCS_STG_EXT.1 that the key storage is software-based for asymmetric keys, symmetric keys and persistent secrets. Section 8.3.1 describes how keys and data are stored and encrypted (wrapped) on iOS, and describes the different key classes. The evaluator reviewed figure 5 and determined that the TOE implements the required secure key storage.


Guidance Assurance Activities

Assurance Activity AA-FCS_STG_EXT.1-AGD-01

The evaluator shall review the AGD guidance to determine that it describes the steps needed to import or destroy keys/secrets. The evaluator shall also verify that the API documentation provided according to Section 5.2.2 includes the security functions (import, use, and destruction) described in these requirements. The API documentation shall include the method by which applications restrict access to their keys/secrets in order to meet FCS_STG_EXT.1.4.

Summary

[CCGUIDE]  section 3.2.5 *Keys/Secrets Import/Destruction* provides related guidance for managing keys and secrets. It states that cryptographic keys are stored in keychains and an application only has access to its own key chain items, so access restrictions are automatically enforced. No configuration is required by the TOE users or administrators.

This section also identifies the related API documentation [KEYCHAINPG] , which describes how keychain items are created, managed, and deleted. The evaluator found sections *Keys* and *Keychain Services* which covers topics including key generation, import and export, key usage.

Test Assurance Activities

Assurance Activity AA-FCS_STG_EXT.1-ATE-01

The evaluator shall test the functionality of each security function:

- **Test 1:** *The evaluator shall import keys/secrets of each supported type according to the AGD guidance. The evaluator shall write, or the developer shall provide access to, an application that generates a key/secret of each supported type and calls the import functions. The evaluator shall verify that no errors occur during import.*
- **Test 2:** *The evaluator shall write, or the developer shall provide access to, an application that uses an imported key/secret:*

- *For RSA, the secret shall be used to sign data.*
- *For ECDSA, the secret shall be used to sign data*

In the future additional types will be required to be tested:

- *For symmetric algorithms, the secret shall be used to encrypt data.*
- *For persistent secrets, the secret shall be compared to the imported secret.*

The evaluator shall repeat this test with the application-imported keys/secrets and a different application's imported keys/secrets. The evaluator shall verify that the TOE requires approval before allowing the application to use the key/secret imported by the user or by a different application:

- *The evaluator shall deny the approvals to verify that the application is not able to use the key/secret as described.*
- *The evaluator shall repeat the test, allowing the approvals to verify that the application is able to use the key/secret as described.*

If the ST Author has selected "common application developer", this test is performed by either using applications from different developers or appropriately (according to API documentation) not authorizing sharing.

- **Test 3:** *The evaluator shall destroy keys/secrets of each supported type according to the AGD guidance. The evaluator shall write, or the developer shall provide access to, an application that destroys an imported key/secret. The evaluator shall repeat this test with the application-imported keys/secrets and a different application's imported keys/secrets. The evaluator shall verify that the TOE requires approval before allowing the application to destroy the key/secret imported by the administrator or by a different application:*

- The evaluator shall deny the approvals and verify that the application is still able to use the key/secret as described.
- The evaluator shall repeat the test, allowing the approvals and verifying that the application is no longer able to use the key/secret as described.

If the ST Author has selected “common application developer”, this test is performed by either using applications from different developers or appropriately (according to API documentation) not authorizing sharing.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator generated an RSA and ECDSA key using OpenSSL externally to the TOE and loaded them into the TOE's keychain, and verified that it succeeded. The evaluator also used the CryptoTest application to locally generate keys, store them and fetch in/from the TOE.

Test 2: the evaluator installed .p12 key bundles on the TOE, started the test AP, and connected the TOE to the WLAN network. The evaluator then access https://taun and selected the RSA certificate when prompted by the TOE. The evaluator verified that the connection was successful. The evaluator then cleared the browser cache and data and re-performed the operation with the ECDSA certificate. The evaluator verified that once gain the connection was successful. The evaluator re-performed the operations and verified that if the evaluator denies selecting one of the certificates, the connection fails.

Test 3: the evaluator deleted the configuration profile containing the RSA/ECDSA keys and verified that the keys are only deleted when removal of the profile is authorized by the user of the TOE.

2.1.2.21 Extended: Encrypted Cryptographic Key Storage (FCS_STG_EXT.2)

FCS_STG_EXT.2.1

TSS Assurance Activities

Assurance Activity AA-FCS_STG_EXT.2.1-ASE-01

The evaluator shall review the TSS to determine that the TSS includes key hierarchy description of the protection of each DEK for data-at-rest, of software-based key storage, of long-term trusted channel keys, and of KEK related to the protection of the DEKs, long-term trusted channel keys, and software-based key storage. This description must include a diagram illustrating the key hierarchy implemented by the TOE in order to demonstrate that the implementation meets FCS_STG_EXT.2. The description shall indicate how the functionality described by FCS_RBG_EXT.1 is invoked to generate DEKs (FCS_CKM_EXT.2), the key size (FCS_CKM_EXT.2 and FCS_CKM_EXT.3) for each key, how each KEK is formed (generated, derived, or combined according to FCS_CKM_EXT.3), the integrity protection method for each encrypted key (FCS_STG_EXT.3), and the IV generation for each key encrypted by the same KEK (FCS_IV_EXT.1). More detail for each task follows the corresponding requirement.

Summary

Section 8.3.1 in the [ST] describes the *Overview of Key Management*.

The evaluator reviewed section 8.3.1, particularly, the diagram in figure 5, and determined that it describes the key hierarchy or all the key material used by iOS. This description defines the keys as 256-bit AES keys and explains how a new per-file key is generated, as well as how keys are wrapped and protected by the OS. The IV is determined on the memory location of the file (the IV is calculated with the block offset into the file).

Guidance Assurance Activities

No assurance activities defined.


Test Assurance Activities

No assurance activities defined.

FCS_STG_EXT.2.2**TSS Assurance Activities****Assurance Activity AA-FCS_STG_EXT.2.2-ASE-02**

The evaluator shall examine the key hierarchy description in the TSS section to verify that each DEK and software-stored key is encrypted according to FCS_STG_EXT.2.

Summary

Section 8.3.1 in the [ST] describes the *Overview of Key Management* .

The evaluator reviewed section 8.3.1 and determined that every Data Encryption Key (DEK) is encrypted using AES key wrapping according to RFC3394 (AES Key Wrap) as defined in FCS_STG_EXT.2.

Guidance Assurance Activities

No assurance activities defined.


Test Assurance Activities

No assurance activities defined.

2.1.2.22 Extended: Integrity of Encrypted Key Storage (FCS_STG_EXT.3)**TSS Assurance Activities****Assurance Activity AA-FCS_STG_EXT.3-ASE-01**

The evaluator shall examine the key hierarchy description in the TSS section to verify that each encrypted key is integrity protected according to one of the options in FCS_STG_EXT.3.

Summary

Section 8.3.1 in the [ST] describes the *Overview of Key Management* .

The evaluator reviewed FCS_STG_EXT.3 and determined that the integrity mechanism used for each encrypted key is the one provided by the Key Wrap cipher mode defined in RFC3394 and NIST SP 800-38F . This mode includes authentication as described in section 6.2 in SP 800-38F.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.2.23 Extended: TLS Client Protocol (FCS_TLSC_EXT.1)


FCS_TLSC_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.1-ASE-01

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified include those listed for this component.

Summary

Section 8.9.1, *EAP-TLS and TLS*, in the [ST]  describes the EAP-TLS and TLS protocols and ciphersuites supported by the TOE.

Section 8.9.1 specifies how these protocols can be used for WLAN. The TOE supports EAP-TLS with TLS v1.0, v1.1 and v1.2 with the following ciphersuites.

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC5246
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC5246
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC5246

The TOE supports TLS v1.2 with the following ciphersuites.



- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

Guidance Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.1-AGD-01

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.

Summary

[CCGUIDE]  section 3.3.2 *TLS Configuration* provides related guidance for configuring TLS. It states the supported cipher suites below are automatically selected by the TOE (i.e., the TOE does not support the individual selection of TLS cipher suites.) The TLS cipher suites available are defined by the TLS server where all cipher suites listed in the ST are always available. Thus, no additional configuration is required by the administrator. It also states that TLS is provided by the APIs of iOS Security Framework, which uses the Apple iOS CoreCrypto Cryptographic Module for ARM which implements TLS 1.2 (along with TLS 1.0 and TLS 1.1 for EAP-TLS) supporting the cipher suites listed in Table 9 (and Table 8 for EAP-TLS) of [CCGUIDE]  which are duplicated below.

For TLS 1.2, the following cipher suites are supported by the TOE in the evaluated configuration:

- TLS_RSA_WITH_AES_128_CBC_SHA

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

For TLS 1.0, 1.1, and 1.2, the TOE supports the following TLS cipher suites in the evaluated configuration:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256

Other configurations for TLS can be done via a Configuration Profile as described in section 3.3.2 including setting the TLS version to 1.2 (for the evaluated configuration). Additionally, the administrator must configure the TOE to reject untrusted HTTPS certificates by setting the *allowUntrustedTLSPrompt* key to 'false'; configure which server certificate common names and certificates will be accept by the TOE using the *TLSTrustedServerNames* and *PayloadCertificateAnchorUUID* keys; set the reference identifier as described in [CKTSREF][\[1\]](#); and if needed, add additional CAs using the the *EAPClientConfiguration*, *PayloadCertificateAnchorUUID*, and *TLSTrustedServerNames* keys.

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.1-ATE-01

The evaluator shall write, or the ST author shall provide, an application for the purposes of testing TLS. The evaluator shall also perform the following tests:

- **Test 1:** *The evaluator shall establish a TLS connection using each of the ciphersuites 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 ciphersuite 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 ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).*
- **Test 2:** *The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.*
- **Test 3:** *The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send an RSA certificate while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.*
- **Test 4:** *The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.*
- **Test 5:** *The evaluator shall perform the following modifications to the traffic:*
 - *Change the TLS version selected by the server in the Server Hello to a non-supported TLS version (for example, 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection.*
 - *Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.*
 - *Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.*

- (conditional) If an ECDHE or DHE ciphersuite is selected, modify the signature block in the Server's Key Exchange handshake message, and verify that the client rejects the connection after receiving the Server Key Exchange message.
- Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.
- Send a valid Server Finished message in plaintext and verify the client sends a fatal alert upon receipt and does not send any application data. The server's finished message shall contain valid verify_data and shall parse correctly using a network protocol analysis tool.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[4\]](#).

Test 1: the evaluator configured the AP to use each cipher suite defined in the [ST] [\[5\]](#) and connected the TOE to the AP, while monitoring the traffic using Wireshark.

Test 2: the evaluator created a certificate that did not have the "Server Authentication" in the extendedKeyusage field. The evaluator then attempted to connect the TOE to the AP and verified that the connection was unsuccessful.

Test 3: the evaluator configured the AP in such a way that the selected ciphersuite does not match the ciphersuite listed in the certificate. The evaluator attempted to connect the TOE to the AP and verified that it failed.

Test 4: the evaluator configured the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite by using a modified version of OpenSSL and verified that the client (the TOE) denied the connection.

Test 5.1: the evaluator changed the version of the version of TLS used by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.2: the evaluator modified the first byte of the random number sent in the server hello by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.3: the evaluator set the first byte of the cipher spec field in the server hello message to 0xff by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.4: the evaluator added one to the first byte of the RSA signature of the server key exchange message by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.5: the evaluator added one to the first byte of the PRF digest server finished message by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.6: the evaluator sent a garbled message after the change cipher spec by using a modified version of OpenSSL and verified that the TOE denies the connection.

FCS_TLSC_EXT.1.2

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.2-ASE-01

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

Summary

Section 8.9.1, *EAP-TLS and TLS*, in the [ST] describes the EAP-TLS and TLS protocols and ciphersuites supported by the TOE.

The evaluator reviewed section 8.9.1 and determined that the TOE will compare the Domain Name (DN) contained within the certificate (the Subject Common Name (CN), Subject Alternative Name (SAN) fields, IP address or wildcards, if applicable) to the DN of the requested server. If the DN does not match, then the application requesting to establish a trusted channel cannot establish the connection.

Certificate pinning is supported by the TOE. The user of the TLS framework can use certificate pinning. Note that TLS clients in the TOE (such as Safari) do not support certificate pinning.

Guidance Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.2-AGD-01

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. In particular, the AGD guidance should describe the API used by applications for configuring the reference identifier.

Summary

[CCGUIDE] section 3.3.2 *TLS Configuration* provides related guidance for setting the reference identifier used for certification validation in TLS. The guidance/API documentation for setting the reference identifier is provided in the "Obtaining policies for establishing trust" section of the 'Policies' chapter in [CKTSREF].

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.2-ATE-01

The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

- **Test 1:** *The evaluator shall present a server certificate that does not contain an identifier in either the Subject Alternative Name (SAN) or Common Name (CN) that matches the reference identifier. The evaluator shall verify that the connection fails.*
- **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.*
- **[TD0304] 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.*
- **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.*
- **Test 5:** *The evaluator shall perform the following wildcard tests with each supported type of reference identifier:*
 - *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.*
 - *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.com) and verify that the connection fails.*
- *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.*
 - **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.
 - **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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[4\]](#).

Test 1: the evaluator generated keys and certificates for the TLS server that did not contain the right Common Name (CN) or Subject Alternative Name (SAN). The evaluator connected the TOE to https://server and verified that either Safari asks confirmation of the certificate, or that the communication is rejected.

Test 2: the evaluator generated keys and certificates for the server side with a CN that matches the reference identifier, contains the SAN extension, but does not contain a SAN that matches the reference identifier. The evaluator connected the TOE to the TLS test server and verified that the connection failed. The evaluator performed this test for both Domain Name Server (DNS) and Uniform Resource Identifier (URI) SANs.

Test 3: the evaluator generated keys and certificates for the server that contained a CN that matched the reference identifier, but no SAN extension. The evaluator connected the TOE to the TLS server and verified that the connection succeeded.

Test 4: the evaluator generated keys and certificates for the server that contained a CN that did not match the reference identifier but contained SAN extension which did match. The evaluator connected the TOE to the TLS server and verified that the connection was successful.

Test 5: the evaluator generated keys and certificates for the server which contained *.domain.com as a wildcard in the SAN. The evaluator connected the TOE to https://subdomain.domain.com, which was successful. The evaluator then connected the TOE to https://domain.com and https://server.subdomain.domain.com and verified that the connection failed, or that Safari asks for confirmation of the certificate.

FCS_TLSC_EXT.1.3

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.3-ATE-01

The evaluator shall perform the following test:

- **Test 1:** 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 to the Trust Anchor Database 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, and show that the function fails.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator generated keys and certificates for the server for which the CA was not trusted by the TOE. When connecting to https://server, the evaluator verified that either Safari asks for confirmation of the certificate, or denies the connection. When the evaluator performed this test with a trusted CA, the TOE connected without asking any confirmation.

FCS_TLSC_EXT.1.4

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.4-ASE-01

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.

Summary

Section 8.5.2 of the [ST] describes the certificates used by the TOE, and how iOS uses certificates for different services, including TLS and EAP-TLS authentication.

Section 8.5.2 describes that certificates can be installed for TLS mutual authentication.

Guidance Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.4-AGD-01

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.

Summary

[CCGUIDE] sections 3.3.2 *TLS Configuration* provides related guidance for configuring client-side X.509 certificates which involves installing a client certificate and its keys on the device using the Certificate Payload in a Configuration Profile as described [IOS_CFG].

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.4-ATE-01

[TD0305] The evaluator shall also perform the following tests:

- **Test 1:** The evaluator shall establish a connection to a peer 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 2:** The evaluator shall establish a connection to a peer 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) messages."

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator generate keys and certificates for the test server and connected the TOE to it. The evaluator verified that the server was not configured to require client authentication, and analyzed the packet traffic using the tcpdump tool. The evaluator also verified the information on the web page displayed by the server.

Test 2: the evaluator re-performed Test 1, where the server is this time configured to require mutual authentication. The evaluator verified in the packet capture that the server requested the certificate from the client, and that the client sent it to the server.

2.1.2.24 Extended: TLS Protocol (FCS_TLSC_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.2.1-ASE-01

The evaluator shall verify that TSS describes the Supported Elliptic Curves Extension and whether the required behavior is performed by default or may be configured.

Summary

Section 8.9.1 of the ST specifies that the TOE uses the *EAP-TLS* and *TLS* protocols. It is specified that the elliptic curve cipher suites may use the following elliptic curve extensions: secp256r1 (default), secp384r1 (default), and x25519 (may be disabled in the operational environment).

Guidance Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.2.1-AGD-01

If the TSS indicates that the Supported Elliptic Curves Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves Extension.

Summary

[CCGUIDE] sections 3.3.2 *TLS Configuration* provides related guidance for configuring elliptic curves extension for TLS. It states that the available elliptic curves (i.e., secp256r1, secp384r1) are automatically selected by the TOE, that is the TOE does not support the individual selection of elliptic curves. The curves available are defined by the server where all curves listed in the Security Target are always available. Thus, no additional configuration is required by the end user.

Also, curve x25519 is also supported by the TOE and may be disabled in the operational environment.

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.2.1-ATE-01

- **Test 1:** [TD0244] The evaluator shall configure a server to perform ECDHE key exchange using each of the TOE's supported curves and shall verify that the TOE successfully connects to the server.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator configured the TLS server to only accept one curve and connected the TOE to the server. The evaluator performed this task for each supported curve by the TOE.

2.1.2.25 Cryptographic Key Storage (FCS_STG_EXT.4(AGENT))

TSS Assurance Activities

Assurance Activity AA-FCS_STG_EXT.4-MDMA-MDF-ASE-01

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

Summary

Section 8.3.2 of the [ST] specifies the *Storage of Persistent Secrets and Private Keys by the Agent*. The evaluator verified that *Table 7: Summary of keys and persistent secrets used by the Agent* in this section lists all persistent secrets in the iOS Keychain:

Table 4: Summary of keys and persistent secrets used by the Agent

Key / Persistent Secret	Purpose	Storage (for all devices)
TLS keys	Protecting MDM Protocol communications with the MDM Server	Stored on the device in wrapped form in persistent storage
Device Push Token	The device push token is received when registering with the Apple Push Notification Service (APNS) in order to have an unambiguous identifier in APNS.	The token is not stored on the device but sent to the MDM server. The MDM server stores it to be able to contact the device.
UDID	Unique Device ID	Stored in wrapped form in persistent storage
PushMagic	The magic string that must be included in the push notification message. This value is generated by the device.	Stored in wrapped form in persistent storage
Device identity certificate	The device presents its identity certificate for authentication when it connects to the check-in server.	Stored in wrapped form in persistent storage

Key / Persistent Secret	Purpose	Storage (for all devices)
Certificate Payload	https://developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/Introduction/Introduction.html#/apple_ref/doc/uid/TP40010206-CH1-SW248	Stored in wrapped form in persistent storage
Profile encryption key	A profile can be encrypted so that it can only be decrypted using a private key previously installed on a device.	Stored in wrapped form in persistent storage
Guid	Volume Purchase Program (VPP) Account Protection A random UUID should be standard 8-4-4-4-12 formatted UUID string and must be unique for each installation of your product	Stored in wrapped form in persistent storage

The evaluator verified that the TOE

"calls the Apple iOS API"

to store these keys in the Keychain, as specified in the first sentence of section 8.3.2.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.2.26 Cryptographic Key Generation (IKE) (FCS_CKM.1(VPN))

TSS Assurance Activities

Assurance Activity AA-FCS_CKM.1-VPN-ASE-01

The evaluator shall examine the TSS to verify that it describes how the key generation functionality is invoked.

Summary

The evaluator reviewed section 8.9.4 of the [ST] [\[1\]](#) and could verify that key generation is performed in accordance with the IKE/IPsec protocols, using the (EC) Diffie-Hellman key agreement protocol or RSA key transport.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.2.27 IPsec (FCS_IPSEC_EXT.1)

FCS_IPSEC_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.1-VPN-ASE-01

The evaluator shall examine the TSS and determine that it describes how the IPsec capabilities are implemented and how a packet is processed, e.g., what takes place at the platform and what takes place within the client. The TSS will detail the relationship between the client and the underlying platform, including which aspects are implemented by the client, and those that are provided by the underlying platform. The TSS describes how the client interacts with the platform's network stack (e.g., does the client insert itself within the stack via kernel mods, does the client simply invoke APIs to gain access to network services).

If the SPD is implemented by the client, then the TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes the rules that are available and the resulting actions available after matching a rule. The TSS describes how the available rules and actions form the SPD using terms defined in RFC 4301 such as BYPASS (e.g., no encryption), DISCARD (e.g., drop the packet), and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301.

As noted in section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied given the rule structure implemented by the TOE. For example, if the TOE allows specification of ranges, conditional rules, etc., the evaluator shall determine that the description of rule processing (for both inbound and outbound packets) is sufficient to determine the action that will be applied, especially in the case where two different rules may apply. This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular packet) as well as packets that are part of an established SA. If the SPD is implemented by the underlying platform, then the TSS describes how the client interacts with the platform to establish and populate the SPD, including the identification of the platform's interfaces that are used by the client.

Summary

Section 8.9.4.2 of the [ST] specifies that packets are processed by the TOE, in little-endian order. There is no separate client application, the VPN tunnels are configured and controlled by the Network Extension Framework, which is part of the host OS's Core OS Layer. IPsec on the TOE is implemented according to RFC4301.

The Security Policy Database (SPD) is implemented by the TOE, which is configured as a managed device using a configuration profile (either manually, through the Apple Configurator, or through an MDM).

The VPN payload, described in [IOS_CFG], specifies how a packet is processed against the SPD, and includes IPsec Dictionary Keys, IKEv2 Dictionary Keys, DNS Dictionary Keys, Proxies Dictionary Keys, and AlwaysOn Dictionary Keys.

Guidance Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.1-VPN-AGD-01

The evaluator shall examine the operational guidance to verify it describes how the SPD is created and configured. If there is an administrative interface to the client, then the guidance describes how the administrator specifies rules for processing a packet. The description includes all three cases - a rule that ensures packets are encrypted/decrypted, dropped, and allowing a packet to flow in plaintext. The evaluator shall determine that the description in the operational guidance is consistent with the description in the TSS, and that the level of detail in the operational guidance is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

If the client is configured by an external application, such as the VPN gateway, then the operational guidance should indicate this and provide a description of how the client is configured by the external application. The description should contain information as to how the SPD is established and set up in an unambiguous fashion. The description should also include what is configurable via the external application, how ordering of entries may be expressed, as well as the impacts that ordering of entries may have on the packet processing.

In either case, the evaluator ensures the description provided in the TSS is consistent with the capabilities and description provided in the operational guidance.

Summary

[CCGUIDE] [\[1\]](#) section 3.3.3 *IPsec Configuration* provides related guidance for IPsec. It states the following:

- IPsec is implemented by the TOE natively.
- The Security Policy Database (SPD) is created and configured by defining exceptions for IP traffic routing in a Configuration Profile. By default, all IP traffic is sent through a protected channel between the devices and the desired endpoint (denoted as "PROTECT" in the SPD). Any deviations from the default routing behavior must be explicitly specified as exceptions in the Configuration Profile.
- Packet processing exceptions can be created for applications which make use of Captive Networking Identifiers (Captive Networking Apps), as well as for VoiceMail, AirPrint, and CellularServices. The mobile device administrator will need to refer to their organization's security policies to determine whether exceptions should be created and how those exceptions should be configured.
- Exceptions for Captive Networking Apps can be configured to allow traffic for these apps to pass outside the tunnel (BYPASS in the SPD). Exceptions for VoiceMail, AirPrint, and CellularServices can allow traffic to pass unencrypted outside the tunnel (BYPASS in the SPD) or drop the traffic entirely (DISCARD in the SPD).
- Exceptions for VoiceMail, AirPrint, and CellularServices can be defined in the VPN payload by specifying the respective keys and key values described in table 11 of [CCGUIDE] [\[1\]](#).
- Likewise, exceptions for Captive Network Apps can be defined by specifying the keys and key values described in table 11 of [CCGUIDE] [\[1\]](#).

The evaluator found the provided guidance to be consistent with the description in the TSS.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.1-VPN-ATE-01

Depending on the implementation, the evaluator may be required to use a VPN gateway or some form of application to configure the client and platform. For Test 2, the evaluator is required to choose an application that allows for the configuration of the full set of capabilities of the VPN client (in conjunction with the platform). For example, if the client provides a robust interface that allows for specification of wildcards, subnets, etc., it is unacceptable for the evaluator to choose a VPN Gateway that only allows for specifying a single fully qualified IP addresses in the rule.

The evaluator shall perform the following tests:

- **Test 1:** *The evaluator shall configure an SPD on the client that is capable of the following: dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the client with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule. The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packets were dropped, allowed through without modification, was encrypted by the IPsec implementation.*
- **Test 2:** *The evaluator shall devise several tests that cover a variety of scenarios for packet processing. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and operational guidance. Potential areas to cover include rules with overlapping ranges and conflicting entries,*

inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the operational guidance.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator implemented an automatic test to test this assurance activity. The program will perform the activities specified in the assurance activity, and the traffic will be sniffed using a traffic analyzer. The result shows the Always-On-VPN configuration with a passing and a failing connection attempt. The test result is obtained by performing the test defined for FDP_IFC_EXT.1 Test 1. With the PCAP logs, the passing connection attempt shows that all traffic is encapsulated into ESP. The failing test result shows that if the IPsec server is not reachable, no traffic is generated. The test also demonstrates that if the remote end sends non-encrypted traffic that ought to be transmitted encrypted, the TOE discards the remote traffic.

The testing conducted with FCS_IPSEC_EXT.1.1 Test 1 covers all different use cases defined for the Always-On-VPN:

- Passing traffic encrypted through the tunnel
- reverting traffic generated from TOE to a remote entity if the VPN is not set up
- Dropping traffic generated from a remote entity that is not encrypted
- Enabling and disabling all defined exceptions for traffic to bypass the VPN tunnel
- Testing failing authentication during the VPN handshake

FCS_IPSEC_EXT.1.2

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.2-VPN-ASE-01

The evaluator shall check the TSS to ensure it states that the VPN can be established to operate in tunnel mode and/or transport mode (as selected). The evaluator shall confirm that the operational guidance contains instructions on how to configure the connection in each mode selected.

Summary

The evaluator reviewed the VPN sections (8.9.4 and below) in the [ST] and verified that the TOE operates IPsec in tunnel mode only, which is congruent with FCS_IPSEC.1.1, which specifies that tunnel mode only is supported by the TOE.

Guidance Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.2-VPN-AGD-01

If both transport mode and tunnel mode are implemented, the evaluator shall review the operational guidance to determine how the use of a given mode is specified.

Summary

Per [ST], the TOE only supports tunnel mode. Related guidance for configuring tunnel mode is provided in [CCGUIDE] section 3.3.5 *VPN Configuration* and section 3.3.6 *Keys for Configuration Network Protocol*. The configuration involves using a Configuration Profile with the VPN Payload.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.2-VPN-ATE-01

The evaluator shall perform the following test(s) based on the selections chosen:

- **Test 1 [conditional]:** If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE/platform to operate in tunnel mode and also configures a VPN gateway to operate in tunnel mode. The evaluator configures the TOE/platform and the VPN gateway to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the client to connect to the VPN GW peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.
- **Test 2 [conditional]:** If transport mode is selected, the evaluator uses the operational guidance to configure the TOE/platform to operate in transport mode and also configures an IPsec peer to accept IPsec connections using transport mode. The evaluator configures the TOE/platform and the endpoint device to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection from the TOE/platform to connect to the remote endpoint. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.
- **Test 3 [conditional]:** If both tunnel mode and transport mode are selected, the evaluator shall perform both Test 1 and Test 2 above, demonstrating that the TOE can be configured to support both modes..
- **Test 4 [conditional]:** If both tunnel mode and transport mode are selected, the evaluator shall modify the testing for FCS_IPSEC_EXT.1 to include the supported mode for SPD PROTECT entries to show that they only apply to traffic that is transmitted or received using the indicated mode.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: The test suite developed covers multiple test aspects: the CSV file contained in the test results archive marked with FCS_IPSEC_EXT.1.2 Test 1 establishes VPN in tunnel mode to a remote VPN peer via WiFi (the test log specifies the different cryptographic parameters used for the VPN connection); establishes VPN in tunnel mode to a remote VPN peer via cellular connection; shows that without the exception, AirPrint printers are not reachable as all communication will be transmitted via VPN. With the exception, AirPrint data is sent bypassing the VPN; shows that without the exception, Captive Portals are not reachable as all communication will be transmitted via VPN. With the exception, Captive Portal data is sent bypassing the VPN; and shows that without the exception, Voice Mail is not reachable as all communication will be transmitted via VPN. With the exception, Voice Mail data is sent bypassing the VPN.

Test 2: not applicable, because [ST] specifies tunnel mode only.

Test 3: not applicable, because [ST] specifies tunnel mode only.

Test 4: not applicable, because [ST] specifies tunnel mode only.

FCS_IPSEC_EXT.1.3

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.3-VPN-ASE-01

The evaluator shall examine the TSS to verify that the TSS provides a description of how a packet is processed against the SPD and that if no “rules” are found to match, that a final rule exists, either implicitly or explicitly, that causes the network packet to be discarded.

Summary

Please refer to [AA-FCS_IPSEC_EXT.1.1-VPN-ASE-01](#)

Guidance Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.3-VPN-AGD-01

The evaluator checks that the operational guidance provides instructions on how to construct or acquire the SPD [and uses the guidance to configure the TOE/platform for the following tests.]

Summary

[CCGUIDE] [\[1\]](#) section 3.3.3 *IPsec Configuration* provides related guidance for IPsec. It states that the SPD is created and configured by defining exceptions for IP traffic routing in a Configuration Profile using the VPN Payload and related keys described in table 11 of [CCGUIDE] [\[1\]](#). Additionally, the evaluator used the guidance to perform testing.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.3-VPN-ATE-01

[The evaluator checks that the operational guidance provides instructions on how to construct or acquire the SPD] and uses the guidance to configure the TOE/platform for the following tests.

The evaluator shall perform the following test:

- **Test 1:** *The evaluator shall configure the SPD such that it has entries that contain operations that DISCARD, PROTECT, and (if applicable) BYPASS network packets. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches a BYPASS entry and send that packet. The evaluator should observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a “TOE/platform created” final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was not permitted to flow to any of the TOE’s interfaces.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[1\]](#).

Test 1: The test establishes a VPN in tunnel mode to a remote VPN peer via WiFi or LTE, and installing a Captive Portal bypass setting. Using the packet sniffer logs, the evaluator was able to clearly identify the captive portal bypass for both WiFi and LTE.

FCS_IPSEC_EXT.1.4

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.4-VPN-ASE-01

The evaluator shall examine the TSS to verify that the algorithms AES-GCM-128 and AES-GCM-256 are implemented. If the ST author has selected either AES-CBC-128 or AES-CBC-256 in the requirement, then the evaluator verifies the TSS describes these as well. In addition, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1(4) Cryptographic Operations (for keyed-hash message authentication).

Summary

The evaluator examined section 8.9.4.3 and verified that AES-GCM-128 and 256 are implemented by the TOE, as well as AES-CBC-128 and 256. The evaluator reviewed FCS_COP.1(4) and verified that the HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 are also implemented by the TOE.

Guidance Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.4-VPN-AGD-01

The evaluator checks the operational guidance to ensure it provides instructions on how the TOE is configured to use the algorithms selected in this component and whether this is performed through direct configuration, defined during initial installation, or defined by acquiring configuration settings from an environmental component.

Summary

Per [CCGUIDE] [\[1\]](#) section 3.3.5.3 *Mobile device administrators*, the administrator may use the VPN Policy Payload in the Configuration Profile to configure a traditional systemwide VPN based on IPsec. Table 11 of [CCGUIDE] [\[1\]](#) describes the keys for VPN configuration

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.4-VPN-ATE-01

- **Test 1:** The evaluator shall configure the TOE/platform as indicated in the operational guidance configuring the TOE/platform to using each of the AES-GCM-128, and AES-GCM-256 algorithms, and attempt to establish a connection using ESP. If the ST Author has selected either AES-CBC-128 or AES-CBC-256, the TOE/platform is configured to use those algorithms and the evaluator attempts to establish a connection using ESP for those algorithms selected.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[1\]](#).

Test 1: The test establishes VPN in tunnel mode to a remote VPN peer via WiFi and LTE. The connection is established with AES 128 CBC, AES 256 CBC, AES 128 GCM, and AES 256 GCM.

FCS_IPSEC_EXT.1.5

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.5-VPN-ASE-01

The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented. If IKEv1 is implemented, the evaluator shall verify that the TSS indicates whether or not XAUTH is supported, and that aggressive mode is not used for IKEv1 Phase 1 exchanges (i.e. only main mode is used). It may be that these are configurable options.

Summary

The evaluator examined the TSS and verified that only IKEv2 is implemented by the TOE, which is consistent with FCS_IPSEC.1.

Guidance Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.5-VPN-AGD-01

The evaluator shall check the operational guidance to ensure it instructs the administrator how to configure the TOE/platform to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE/platform to perform NAT traversal for the test below. If XAUTH is implemented, the evaluator shall verify that the operational guidance provides instructions on how it is enabled or disabled.

If the TOE supports IKEv1, the evaluator shall verify that the operational guidance either asserts that only main mode is used for Phase 1 exchanges, or provides instructions for disabling aggressive mode.

Summary

Per [ST], the TOE only supports IKEv2 in the evaluated configuration. Also, the TOE does not support XAUTH.

[CCGUIDE] section 3.3.3 *IPsec Configuration* contains the following statement:

When the VPN is configured as Always-On, the mobile device uses IKEv2 for security association (SA) establishment. Since the mobile device must be configured with Always-On VPN in order to be in the evaluated configuration, the use of IKEv2 does not need to be configured separately.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.5-VPN-ATE-01

- **Test 1:** The evaluator shall configure the TOE/platform so that it will perform NAT traversal processing as described in the TSS and RFC 7296, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed. If XAUTH is supported, the evaluator shall verify that this test can be repeated with XAUTH both enabled and disabled in the manner specified by the operational guidance.
- **Test 2 [conditional]:** If the TOE supports IKEv1, the evaluator shall perform any applicable operational guidance steps to disable the use of aggressive mode and then attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator shall show that the TOE/platform will reject a VPN gateway from initiating an IKEv1 Phase 1 connection in aggressive mode. The evaluator should then show that main mode exchanges are supported.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: The test establishes a VPN in tunnel mode to a remote VPN peer via WiFi and LTE using NAT. The TOE log file shows the following content for enabling the NAT processing: neagent: (NetworkExtension) [com.apple.networkextension:] Detected remote NAT
neagent: (NetworkExtension) [com.apple.networkextension:] Detected local NAT

Test 2: not applicable, because the [ST] only specifies IKEv2.

FCS_IPSEC_EXT.1.6

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.6-VPN-ASE-01

The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms AES-CBC-128, AES-CBC-256 are specified, and if others are chosen in the selection of the requirement, those are included in the TSS discussion.

Summary

Section 8.9.4.3 specifies that the algorithms used for encrypting the IKE payload are AES-GCM-128 and 256, AES-CBC-128 and 256, which is consistent with FCS_IPSEC.1.

Guidance Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.6-VPN-AGD-01

The evaluator checks the operational guidance to ensure it provides instructions on how the TOE is configured to use the algorithms selected in this component and whether this is performed through direct configuration, defined during initial installation, or defined by acquiring configuration settings from an environmental component.

Summary

Per [ST], the TOE supports AES-CBC-128, AES-CBC-256, and AES-GCM-128 for payload encryption.

[CCGUIDE] Table 11 "Essential Keys for the VPN Payload" specifies the algorithms that must be set for the *EncryptionAlgorithm* key to be either AES-128, AES-256, AES-128-GCM, or AES-256-GCM. This section also contains a note explaining that AES-128 and AES-256 use the CBC mode.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.6-VPN-ATE-01

The evaluator shall use the operational guidance to configure the TOE/platform (or to configure the Operational Environment to have the TOE receive configuration) to perform the following test for each ciphersuite selected:

- **Test 1:** *The evaluator shall configure the TOE/platform to use the ciphersuite under test to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation. The evaluator will confirm that the connection is successful by confirming that data can be passed through the connection once it is established. For example, the evaluator may connect to a webpage on the remote network and verify that it can be reached.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: The test establishes VPN in tunnel mode to a remote VPN peer via WiFi and LTE. The connection is established with AES 128 CBC, AES 256 CBC, AES 128 GCM, and AES 256 GCM. The list of ciphers covers all ciphers selected in the ST.

FCS_IPSEC_EXT.1.7

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.7-VPN-AGD-01

The evaluator shall check the operational guidance to ensure it provides instructions on how the TOE configures the values for SA lifetimes. In addition, the evaluator shall check that the guidance has the option for either the Administrator or VPN Gateway to configure Phase 1 SAs if time-based limits are supported. Currently there are no values mandated for the number of packets or number of bytes, the evaluator shall simply check the operational guidance to ensure that this can be configured if selected in the requirement.

Summary

Per [ST], the TOE ensures that IKEv2 SA lifetimes can be configured by an Administrator based on length of time. If length of time is used, it must include at least one option that is 24 hours or less for Phase 1 SAs and 8 hours or less for Phase 2 SAs.

[CCGUIDE] Table 11 "Essential Keys for the VPN Payload" specifies the algorithms that must be set for the *EncryptionAlgorithm* key.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.7-VPN-ATE-01

When testing this functionality, the evaluator needs to ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

- **Test 1 [conditional]:** The evaluator shall configure a maximum lifetime in terms of the # of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets (or bytes) through this SA is exceeded, the connection is closed.
- **Test 2 [conditional]:** The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.
- **Test 3 [conditional]:** The evaluator shall perform a test similar to Test 2 for Phase 2 SAs, except that the lifetime will be 8 hours or less instead of 24 hours or less.
- **Test 4 [conditional]:** If a fixed limit for IKEv1 SAs is supported, the evaluator shall establish an SA and observe that the connection is closed after the fixed traffic and/or time value is reached.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: not applicable, the [ST] specifies time-based maximum lifetime only

Test 2: The test suite maintains a VPN communication channel for more than 24h and 30 minutes. The VPN is established via WiFi and LTE. The iOS logs show: The phase 1 SA (IKE SA) is rekeyed after 19 hours (see log entry containing "Exchange Type: Rekey IKE SA". The evaluator notes that each platform is tested either with WiFi or with cellular network, but not with both. This is due to the fact that each test requires 24 hours time. The test bed allows only one device with one network

connection to be tested at one time. This implies that testing both WiFi and cellular network on all devices requires an infeasible amount of time. Further, all other tests cover WiFi and cellular network demonstrating that the two physical link layers are always supported by all devices under test. For the rekey test, half of the devices are tested with WiFi and the other half with cellular network. This demonstrates that the rekey works irrespective of the physical link layer. Therefore, the evaluator concludes that the testing is suitable to demonstrate the proper rekey behavior on all devices under test.

Test 3: The test suite maintains a VPN communication channel for more than 24h and 30 minutes. The VPN is established via WiFi and LTE. The iOS logs show: The phase 2 SA (IPSEC SA or Child SA) is rekeyed after 6 hours (see log entry containing "Exchange Type: Rekey Child SA"). The evaluator notes that each platform is tested either with WiFi or with cellular network, but not with both. This is due to the fact that each test requires 24 hours time. The test bed allows only one device with one network connection to be tested at one time. This implies that testing both WiFi and cellular network on all devices requires an infeasible amount of time. Further, all other tests cover WiFi and cellular network demonstrating that the two physical link layers are always supported by all devices under test. For the rekey test, half of the devices are tested with WiFi and the other half with cellular network. This demonstrates that the rekey works irrespective of the physical link layer. Therefore, the evaluator concludes that the testing is suitable to demonstrate the proper rekey behavior on all devices under test.

Test 4: not applicable, the [ST] specifies IKEv2 only.

FCS_IPSEC_EXT.1.8

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.8-VPN-ASE-01

The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

Summary

The evaluator verified that section 8.9.4.3 of the [ST] lists the DH groups supported by the TOE as follows.

- 5(1536-bit MODP)
- 14(2048-bit MODP)
- 15(3072-bit MODP)
- 19(256-bit Random ECP)
- 20 (384-bit Random ECP)

Within the same section, the evaluator verified that DH groups are negotiated based on the configuration chosen by the administrator.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.8-VPN-ATE-01

The evaluator shall perform the following test:

- **Test 1:** For each supported DH group, the evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: The test establishes a VPN in tunnel mode to a remote VPN peer via WiFi. The connection is established with the Diffie-Hellman groups of 5, 14, 15, 19, 20 compliant to the ST.

FCS_IPSEC_EXT.1.9

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.9-VPN-ASE-01

The evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating "x" (as defined in FCS_IPSEC_EXT.1.9) and each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP-module is used, and that the length of "x" and the nonces meet the stipulations in the requirement.

Summary

The evaluator reviewed section 8.9.4.5 the [ST] and verified that this section describes how the value 'x' is generated.

The values 'x' and the nonce are generated using the TOE platform CAVP validated DRBG, according to FCS_RBG_EXT.1. The possible lengths for 'x' are 224, 256 or 384 bits.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

FCS_IPSEC_EXT.1.11

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.11-VPN-ASE-01

The evaluator ensures that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication.

If pre-shared keys are chosen in the selection, the evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The description in the TSS shall also indicate how pre-shared key establishment is accomplished for TOEs/platforms that can generate a pre-shared key as well as TOEs/platforms that simply use a pre-shared key.

The evaluator shall ensure that the TSS describes how the TOE compares the peer's presented identifier to the reference identifier. This description shall include whether the certificate presented identifier is compared to the ID payload presented identifier, which field(s) of the certificate are used as the presented identifier (DN, Common Name, or SAN),

and, if multiple fields are supported, the logical order comparison. If the ST author assigned an additional identifier type, the TSS description shall also include a description of that type and the method by which that type is compared to the peer's presented certificate.

Summary

The evaluator reviewed section 8.9.4.4 of the [ST] and verified that it describes that RSA or ECDSA X.509v3 certificates are used for authentication.

During the authentication process, a comparison is made between the DN within the certificate (CN and SAN name fields) to the DN of the requested server. If the DN in the certificate does not match the requested DN, the session will not be established.

The SAN takes precedence over the DN entry. If the SAN does not match the peer's reference identifier, the authentication fails. If the SAN matches, the authentication is successful. Only if the SAN is not present, the DN of the certificate is applied.

Guidance Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.11-VPN-AGD-01

The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established.

The evaluator ensures the operational guidance describes how to set up the TOE/platform to use the cryptographic algorithms RSA and/or ECDSA.

In order to construct the environment and configure the TOE/platform for the following tests, the evaluator will ensure that the operational guidance also describes how to configure the TOE/platform to connect to a trusted CA, and ensure a valid certificate for that CA is loaded into the TOE/platform as a trusted CA.

The evaluator shall also ensure that the operational guidance includes the configuration of the reference identifier(s) for the peer.

Summary

[CCGUIDE] section 3.3.5 *VPN Configuration* provides related guidance on VPN configuration. The configuration is done (by an administrator) via the Configuration Profile using the VPN Policy Payload. Table 11 of [CCGUIDE] describes the keys that must be specified for the payload. Per table 11, the *Certificate* key must be set to either RSA (default), ECDSA256, or ECDSA384. Also, the certificate to be used for authentication is specified via the *PayloadCertificateUUID* key. Additionally, reference identifier(s) must be set via the *LocalIdentifier* and *RemoteIdentifier* keys.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.11-VPN-ATE-01

For efficiency's sake, the testing that is performed here has been combined with the testing for FIA_X509_EXT.2.1 (for IPsec connections), FCS_IPSEC_EXT.1.12, FCS_IPSEC_EXT.1.13 and FIA_X509_EXT.2.3. The following tests shall be repeated for each peer authentication protocol selected in the FCS_IPSEC_EXT.1.11 selection above:

- **Test 1:** *The evaluator shall have the TOE/platform generate a public-private key pair, and submit a CSR (Certificate Signing Request) to a CA (trusted by both the TOE/platform and the peer VPN used to establish a connection) for its signature. The values for the DN (Common Name, Organization, Organizational Unit, and Country) will also be passed in the request. Alternatively, the evaluator may import to the TOE/platform a previously generated private key and corresponding certificate.*

- **Test 2:** The evaluator shall use a certificate signed using the RSA or ECDSA algorithm to authenticate the remote peer during the IKE exchange. This test ensures the remote peer has the certificate for the trusted CA that signed the TOE's certificate and it will do a bit-wise comparison on the DN. This bit-wise comparison of the DN ensures that not only does the peer have a certificate signed by the trusted CA, but the certificate is from the DN that is expected. The evaluator will configure the TOE/platform to associate a certificate (e.g., a certificate map in some implementations) with a VPN connection. This is what the DN is checked against.
- **Test 3:** The evaluator shall test that the TOE/platform can properly handle revoked certificates – conditional on whether CRL or OCSP is selected; if both are selected, and then a test is performed for each method. For this draft of the PP-Module, the evaluator has to only test one up in the trust chain (future drafts may require to ensure the validation is done up the entire chain). The evaluator shall ensure that a valid certificate is used, and that the SA is established. The evaluator then attempts the test with a certificate that will be revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the TOE/platform will not establish an SA..
- **Test 4 [conditional]:** The evaluator shall test that given a signed certificate from a trusted CA, that when the DN does not match – any of the four fields can be modified such that they do not match the expected value, that an SA does not get established.
- **Test 5 [conditional]:** The evaluator shall generate a pre-shared key and use it, as indicated in the operational guidance, to establish an IPsec connection with the VPN GW peer. If the generation of the pre-shared key is supported, the evaluator shall ensure that establishment of the key is carried out for an instance of the TOE/platform generating the key as well as an instance of the TOE/platform merely taking in and using the key.

For each supported identifier type (excluding DNs), the evaluator shall repeat the following tests.

- **Test 6:** For each field of the certificate supported for comparison, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's presented certificate and shall verify that the IKE authentication succeeds.
- **Test 7:** For each field of the certificate support for comparison, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to not match the field in the peer's presented certificate and shall verify that the IKE authentication fails.

The following tests are conditional.

- **Test 8 [conditional]:** If, according to the TSS, the TOE supports both Common Name and SAN certificate fields and uses the preferred logic outlined in the Application Note, the tests above with the Common Name field shall be performed using peer certificates with no SAN extension. Additionally, the evaluator shall configure the peer's reference identifier on the TOE to not match the SAN in the peer's presented certificate but to match the Common Name in the peer's presented certificate, and verify that the IKE authentication fails.
- **Test 9 [conditional]:** If the TOE supports DN identifier types, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the subject DN in the peer's presented certificate and shall verify that the IKE authentication succeeds. To demonstrate a bit-wise comparison of the DN, the evaluator shall change a single bit in the DN (preferably, in an Object Identifier (OID) in the DN) and verify that the IKE authentication fails.
- **Test 10 [conditional]:** If the TOE supports both IPv4 and IPv6 and supports IP address identifier types, the evaluator must repeat test 1 and 2 with both IPv4 address identifiers and IPv6 identifiers. Additionally, the evaluator shall verify that the TOE verifies that the IP header matches the identifiers by setting the presented identifiers and the reference identifier with the same IP address that differs from the actual IP address of the peer in the IP headers and verifying that the IKE authentication fails.
- **Test 11 [conditional]:** If, according to the TSS, the TOE performs comparisons between the peer's ID payload and the peer's certificate, the evaluator shall repeat the following test for each combination of supported identifier types and supported certificate fields (as above). The evaluator shall configure the peer to present a different ID payload than the field in the peer's presented certificate and verify that the TOE fails to authenticate the IKE peer.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: The TOE only supports the import of previously generated private key and corresponding certificates for VPN (although it is capable of generating a public-private key pair, it does not provide interfaces to generate a CSR and forward it to a CA). Thus, all previously conducted successful VPN connections require the import of a public certificate and the associated private key into the TOE

when configuring a VPN. In particular, the test steps outlined for FDP_IFC_EXT.1 {MDF} explain how to import a certificate and its key into the TOE using a configuration profile along with the VPN configuration details. All VPN configurations used for all tests import a configuration profile containing the certificate / key pair to establish a successful authentication and thus a successful VPN connection. To support that statement, all configuration profiles used for the VPN testing are provided which contain the certificate/key data as PKCS #12 data blob embedded in these configuration profiles.

Test 2: The test establishes a VPN in tunnel mode to a remote VPN peer via WiFi and LTE. The connection is established with the certificate types of RSA and ECDSA (P-256 and P-384) compliant to the ST. The authentication with all certificate types was successful using the remote ASA VPN server which demonstrates the correct verification of the DN.

Test 3: The test “successfully” fails to establish a VPN in tunnel mode to a remote VPN peer via WiFi and LTE. The connection attempt is performed with the certificate types of RSA and ECDSA (P-256 and P-384) compliant to the ST. The authentication failure is triggered with CRLs. The testing is repeated with the new certificates of the same types that are not revoked including certificates that are subjected to OCSP.


Test 4: The test “successfully” fails to establish a VPN in tunnel mode to a remote VPN peer via WiFi and LTE. The connection attempt is performed with the certificate types of RSA and ECDSA (P-256 and P-384) compliant to the ST. The authentication with all certificate types fails due to an incorrect DN.

Test 5: not applicable, the [ST]  specifies certificate-based authentication only.

Test 6: As outlined in the ST, the CN/DN must match the remote peer’s FQDN. All certificates used in the preceding tests covering successful VPN connections utilize a FQDN as it is visible in the certificate used for FCS_IPSEC_EXT.1.4 Test 1.

Test 7: The test “successfully” fails to establish a VPN in tunnel mode to a remote VPN peer via WiFi and LTE. The connection attempt is performed with the certificate types of RSA and ECDSA (P-256 and P-384) compliant to the ST. The authentication with all certificate types fails due to an incorrect DN.

Test 8: The test “successfully” fail to establish a VPN in tunnel mode to a remote VPN peer via WiFi. The connection attempt is performed with the certificate types of RSA and ECDSA (P-256 and P-384) compliant to the ST. The authentication with all certificate types fails due to an incorrect DN that is defined with the SAN.

Test 9: not applicable, the [ST]  specifies support for FQDN as part of the CN/DN only.

Test 10: not applicable, IP address identifier types are not supported by the TOE.

Test 11: not applicable, the TOE does not support ID payload.

FCS_IPSEC_EXT.1.12

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.12-VPN-ATE-01

Assurance activities for this element are tested through assurance activities for FCS_IPSEC_EXT.1.11.

Summary

FCS_IPSEC_EXT.1.13

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.13-VPN-ATE-01

Assurance activities for this element are tested through assurance activities for FCS_IPSEC_EXT.1.12.

Summary

FCS_IPSEC_EXT.1.14

TSS Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.14-VPN-ASE-01

The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation.

Summary

The evaluator reviewed section 8.9.4.5 of the [ST][\[ST\]](#) and verified that the

"The strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv2/IKE_SA connection and the strength of the symmetric algorithm negotiated to protect the IKEv2 CHILD_SA connection is configured using .xml configuration files. The administrator must explicitly choose the cryptographic algorithms (including key strength) used for each SA. Key strength must be one of 128 or 256 bits as specified in the IKEv2 Dictionary Keys, EncryptionAlgorithm Key. In the evaluated configuration, during negotiation the TOE will only negotiate the configured algorithms which must include an IKEv2/IKE_SA at least that of IKEv2 CHILD_SA."

The possible strengths of 'x' are 224, 256, or 384 bits. The key strength of the negotiated algorithm is explicitly specified by the administrator and can only be 128 or 256 bits. In the evaluated configuration, during negotiation the TOE will only negotiate the configured algorithms which must include an IKEv2/IKE_SA at least that of IKEv2 CHILD_SA. This configuration is specified in the [CCGUIDE][\[CCGUIDE\]](#).

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_IPSEC_EXT.1.14-VPN-ATE-01

The evaluator follows the guidance to configure the TOE/platform to perform the following tests.

- **Test 1:** This test shall be performed for each version of IKE supported. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.
- **Test 2 [conditional]:** This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.
- **Test 3:** This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an IKE SA using an algorithm that is not one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.
- **Test 4:** This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[4\]](#).

Test 1, 2, 4: The test establishes a VPN in tunnel mode to a remote VPN peer via WiFi and LTE. The connection is successfully established by using all of the following different cipher types:

- Symmetric ciphers: AES-128 CBC, AES-256 CBC, AES-128 GCM, AES-256 GCM
- Hashes: SHA-256, SHA384, SHA-512
- IKEv2
- All mentioned cipher combinations apply equally to the ISAKMP SA and the IPSEC SA.

Test 2: not applicable, the TOE only allows specifications of ciphers once for a given VPN configuration.

2.1.2.28 Cryptographic Key Generation (FCS_CKM.1(2)(WLAN))

TSS Assurance Activities

Assurance Activity AA-FCS_CKM.1-WLAN-ASE-01

The evaluator shall verify that the TSS describes how the primitives defined and implemented by this EP are used by the TOE in establishing and maintaining secure connectivity to the wireless clients. The TSS shall also provide a description of the developer's method(s) of assuring that their implementation conforms to the cryptographic standards; this includes not only testing done by the developing organization, but also any third-party testing that is performed.

Summary

Section 8.9.3, *Wireless LAN*, in the [ST][\[1\]](#) describes the WLAN protocol implemented by the TOE.

Section 8.9.3 describes that the WLAN protocol implements the protocol as defined in the IEEE 802.11 (2012). The TOE implements the AES-128-CTR block chaining mode along with the CBC-MAC authentication algorithm (CCMP protocol) required by IEEE 802.11 (2012). The TOE implements WPA2 Enterprise which received Wi-Fi Alliance certificates.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_CKM.1-WLAN-ATE-01

The evaluator shall perform the following tests:

- *Test 1: The evaluator shall configure the access point so the cryptoperiod of the session key is 1 hour. The evaluator shall successfully connect the TOE to the access point and maintain the connection for a length of time that is greater than the configured cryptoperiod. The evaluator shall use a packet capture tool to determine that after the configured cryptoperiod, a re-negotiation is initiated to establish a new session key. Finally, the evaluator shall determine that the renegotiation has been successful and the client continues communication with the access point.*
- *Test 2: The evaluator shall perform the following test using a packet sniffing tool to collect frames between the TOE and a wireless LAN access point:*
 - *Step 1: The evaluator shall configure the access point to an unused channel and configure the WLAN sniffer to sniff only on that channel (i.e., lock the sniffer on the selected channel). The sniffer should also be configured to filter on the MAC address of the TOE and/or access point.*
 - *Step 2: The evaluator shall configure the TOE to communicate with a WLAN access point using IEEE 802.11-2012 and a 256-bit (64 hex values 0-f) pre-shared key. The pre-shared key is only used for testing.*
 - *Step 3: The evaluator shall start the sniffing tool, initiate a connection between the TOE and the access point, and allow the TOE to authenticate, associate, and successfully complete the 4 way handshake with the client.*
 - *Step 4: The evaluator shall set a timer for 1 minute, at the end of which the evaluator shall disconnect the TOE from the wireless network and stop the sniffer.*
 - *Step 5: The evaluator shall identify the 4-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK from the 4-way handshake frames and pre-shared key as specified in IEEE 802.11-2012.*
 - *Step 6: The evaluator shall select the first data frame from the captured packets that was sent between the TOE and access point after the 4-way handshake successfully completed, and without the frame control value 0x4208 (the first 2 bytes are 08 42). The evaluator shall use the PTK to decrypt the data portion of the packet as specified in IEEE 802.11-2012, and shall verify that the decrypted data contains ASCII-readable text.*
 - *Step 7: The evaluator shall repeat Step 6 for the next 2 data frames between the TOE and access point and without frame control value 0x4208.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator connected the TOE to an WiFi Access point (AP) for over one hour, and verified that the 4-way EAPOL handshake using Wireshark was automatically triggered after one hour.

Test 2: the evaluator connected the TOE to the AP and used Wireshark to obtain the anonce, snonce and MAC address from both devices to derive the PTK, KCK, KEK and TK keys. The evaluator then verified that the keys were consistent with the ones found in the logs.

2.1.2.29 Cryptographic Key Distribution (GTK) (FCS_CKM.2(3)(WLAN))

TSS Assurance Activities

Assurance Activity AA-FCS_CKM.2-WLAN-ASE-01

The evaluator shall check the TSS to ensure that it describes how the GTK is unwrapped prior to being installed for use on the TOE using the AES implementation specified in this EP.

Summary

Section 8.9.3, *Wireless LAN*, in the [ST] [\[ST\]](#) describes the WLAN protocol implemented by the TOE. Paragraph three of this section specifically describes how the Group Temporal Key (GTK) is unwrapped as stated below.

The AES key wrapping algorithm is used and AES key unwrapping is performed as described in NIST SP 800-38F section 6.1 to unwrap the GTK, which is sent in an Extensible Authentication Protocol (EAPOL) key frame.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_CKM.2-WLAN-ATE-01

The evaluator shall perform the following test using a packet sniffing tool to collect frames between the TOE and a wireless access point (which may be performed in conjunction with the assurance activity for FCS_CKM.1.1/WLAN).

Step 1: The evaluator shall configure the access point to an unused channel and configure the WLAN sniffer to sniff only on that channel (i.e., lock the sniffer on the selected channel). The sniffer should also be configured to filter on the MAC address of the TOE and/or access point.

Step 2: The evaluator shall configure the TOE to communicate with the access point using IEEE 802.11-2012 and a 256-bit (64 hex values 0-f) pre-shared key, setting up the connections as described in the operational guidance. The pre-shared key is only used for testing.

Step 3: The evaluator shall start the sniffing tool, initiate a connection between the TOE and access point, and allow the TOE to authenticate, associate, and successfully complete the 4-way handshake with the TOE.

Step 4: The evaluator shall set a timer for 1 minute, at the end of which the evaluator shall disconnect the TOE from the access point and stop the sniffer.

Step 5: The evaluator shall identify the 4-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK and GTK from the 4-way handshake frames and pre-shared key as specified in IEEE 802.11-2012.

Step 6: The evaluator shall select the first data frame from the captured packets that was sent between the TOE and access point after the 4-way handshake successfully completed, and with the frame control value 0x4208 (the first 2 bytes are 08 42). The evaluator shall use the GTK to decrypt the data portion of the selected packet as specified in IEEE 802.11-2012, and shall verify that the decrypted data contains ASCII-readable text.

Step 7: The evaluator shall repeat Step 6 for the next 2 data frames with frame control value 0x4208.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[PP_MD_V3.1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[EP_MDM_AGENT_V3.0\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[PP_WLAN_CLI_EP_V1.0\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[MOD_VPN_CLI_V2.1\]](#).

Test 2: the evaluator connected the TOE to the AP and used Wireshark to obtain the anonce, snonce and MAC address from both devices to derive the PTK, KCK, KEK and TK keys. The evaluator then verified that the keys were consistent with the ones found in the logs.

2.1.2.30 Extensible Authentication Protocol-Transport Layer Security (FCS_TLSC_EXT.1(WLAN))

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1-WLAN-ASE-01

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified include those listed for this component. 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.

Summary

Section 8.9.1, *EAP-TLS and TLS*, in the [ST] describes the EAP-TLS and TLS protocols and ciphersuites supported by the TOE.

Section 8.9.1 describes which protocols are supported by the TOE (TLS and EAP-TLS), and specifies how these protocols can be used for WLAN. The TOE supports EAP-TLS with TLS v1.0, v1.1 and v1.2 with the following ciphersuites.

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC5246
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC5246 (optional)
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC5246 (optional)
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC5246 (optional)

The TOE supports TLS v1.2 with the following ciphersuites.

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC5289
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

The evaluator checked section 3.2.8 of the [CCGUIDE] and could verify that there is no configuration required for the TOE. They are automatically selected by the TOE.

Guidance Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1-WLAN-AGD-01

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 (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

The evaluator shall check that the OPE guidance contains instructions for the administrator to configure the list of Certificate Authorities that are allowed to sign certificates used by the authentication server that will be accepted by the TOE in the EAP-TLS exchange, and instructions on how to specify the algorithm suites that will be proposed and accepted by the TOE during the EAP-TLS exchange.

Summary

[CCGUIDE] section 3.3.1 *EAP-TLS Configuration* provides guidance related to EAP-TLS. It states that the supported ciphersuites listed in the Table 8 (reproduced below) are automatically selected by the TOE (i.e., the TOE does not support the individual selection of EAP-TLS cipher suites) when Wi-Fi Protected Access (WPA)-EAP is configured via (by an administrator) a Configuration Profile as follows:

- *EncryptionType* key must be set to 'WPA2'.
- *AcceptEAPTypes* key must be set to '13' (as the value representing EAP-TLS)

Ciphersuite Name
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256

Table 5: EAP-TLS Ciphersuites

[CCGUIDE] section 3.3.2.3 *Mobile device administrators* provides guidance for configuring CAs by stating that additional CAs (beside those preinstalled with iOS) can be added using a Configuration Profile with the *EAPClientConfiguration*, *PayloadCertificateAnchorUUID*, and *TLSTrustedServerNames* keys.

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1-WLAN-ATE-01

The evaluator shall write, or the ST author shall provide, an application for the purposes of testing TLS.

The evaluator shall also perform the following tests:

- *Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites 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 ciphersuite 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 ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).*
- *Test 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.*
- *Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA certificate while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.*
- *Test 4: The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.*
- *Test 5: The evaluator shall perform the following modifications to the traffic:*
 - *Change the TLS version selected by the server in the Server Hello to a non-supported TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection.*

- *Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.*
- *Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.*
- *Modify the signature block in the Server's Key Exchange handshake message, and verify that the client rejects the connection after receiving the Server Key Exchange message.*
- *Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.*
- *Send a garbled message from the Server after the Server has issued the ChangeCipherSpec message and verify that the client denies the connection.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator configured the AP to use each cipher suite defined in the [ST] and connected the TOE to the AP, while monitoring the traffic using Wireshark.

Test 2: the evaluator created a certificate that did not have the "Server Authentication" in the extendedKeyusage field. The evaluator then attempted to connect the TOE to the AP and verified that the connection was unsuccessful.

Test 3: the evaluator configured the AP in such a way that the selected ciphersuite does not match the ciphersuite listed in the certificate. The evaluator attempted to connect the TOE to the AP and verified that it failed.

Test 4: the evaluator configured the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite by using a modified version of OpenSSL and verified that the client (the TOE) denied the connection.

Test 5.1: the evaluator changed the version of the version of TLS used by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.2: the evaluator modified the first byte of the random number sent in the server hello by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.3: the evaluator set the first byte of the cipher spec field in the server hello message to 0xff by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.4: the evaluator added one to the first byte of the RSA signature of the server key exchange message by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.5: the evaluator added one to the first byte of the PRF digest server finished message by using a modified version of OpenSSL and verified that the TOE denies the connection.

Test 5.6: the evaluator sent a garbled message after the change cipher spec by using a modified version of OpenSSL and verified that the TOE denies the connection.

2.1.3 User data protection (FDP)

2.1.3.1 Extended: Security Access Control (FDP_ACF_EXT.1)

FDP_ACF_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FDP_ACF_EXT.1.1-ASE-01

The evaluator shall ensure the TSS lists all system services available for use by an application. The evaluator shall also ensure that the TSS describes how applications interface with these system services, and means by which these system services are protected by the TSF.

The TSS shall describe which of the following categories each system service falls in:

1. *No applications are allowed access*
2. *Privileged applications are allowed access*
3. *Applications are allowed access by user authorization*
4. *All applications are allowed access*

Privileged applications include any applications developed by the TSF developer. The TSS shall describe how privileges are granted to third-party applications. For both types of privileged applications, the TSS shall describe how and when the privileges are verified and how the TSF prevents unprivileged applications from accessing those services.

For any services for which the user may grant access, the evaluator shall ensure that the TSS identifies whether the user is prompted for authorization when the application is installed, or during runtime.

Summary

Section 8.4 in the [ST] [\[1\]](#) describes the *User Data Protection (FDP)* .

Section 8.4 describes what an application is allowed to access on the system and which files it can access. Each application is sandboxed which means that it can only access the files that belong to itself. Regarding the capabilities of the application to access system services, every application must declare device-specific capabilities they need to run. These capabilities are defined in an array or dictionary that contains keys identifying features that the application requires. Applications owned by the same developed account can share content if configured to be part of an application group on the device. Creating such an appropriate group is up to the developer. Moreover, the TOE allows a user to restrict the applications to access certain system services such as the following.

- Location Services
- Contacts
- Calendar
- Reminders
- Photos
- Microphones
- Cameras

Section 8.4.5 specifies that

"Applications prompt the mobile device user to grant permission for the application to use system services when they are installed."

Guidance Assurance Activities

Assurance Activity AA-FDP_ACF_EXT.1.1-AGD-01

The evaluator shall ensure that the operational user guidance contains instructions for restricting application access to system services

Summary

[CCGUIDE] [\[1\]](#) section 3.4.2 *Restrict Application Access to System Services* states the following:

- Access control to system services is hardcoded thus not configured by the TOE user or administrator.
- Access control for applications to system services can be restricted on a per-app basis.
- A list of system services can be obtained from the TOE devices directly.
- For TOE normal users, for each system service, the applications which have permissions to use that service can be inspected and changed.
- For TOE administrators, they cannot specify access control for applications to system services.

Test Assurance Activities

Assurance Activity AA-FDP_ACF_EXT.1.1-ATE-01

Assurance Activity Note: *The following tests require the vendor to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.*

The evaluator shall write, or the developer shall provide, applications for the purposes of the following tests.

- **Test 1:** *For each system service to which no applications are allowed access, the evaluator shall attempt to access the system service with a test application and verify that the application is not able to access that system service.*
- **Test 2:** *For each system service to which only privileged applications are allowed access, the evaluator shall attempt to access the system service with an unprivileged application and verify that the application is not able to access that system service. The evaluator shall attempt to access the system service with a privileged application and verify that the application can access the service.*
- **Test 3:** *For each system service to which the user may grant access, the evaluator shall attempt to access the system service with a test application. The evaluator shall ensure that either the system blocks such accesses or prompts for user authorization. The prompt for user authorization may occur at runtime or at installation time, and should be consistent with the behavior described in the TSS.*
- **Test 4:** *For each system service listed in the TSS that is accessible by all applications, the evaluator shall test that an application can access that system service.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[1\]](#).

Test 1: not applicable, because the [ST] [\[1\]](#) specifies that no services are accessible to any applications.

Test 2: not applicable, because the [ST] [\[1\]](#) specifies that no services grant special access to privileged applications.

Test 3: the evaluator opened the Edhita editor and created a directory and a file with some content. The evaluator then opened iEditor and attempted to locate the directory or file unsuccessfully. The evaluator reversed the process, created a directory or file in iEditor and verified that it could not be seen nor accessed by Edhita. The evaluator then installed configuration profiles on the TOE to supply it with certificates and WiFi credentials, connected the TOE to the WLAN network, and

accessed https://taun. The evaluator verified that Chrome does not accept the web certificate as trusted (even though the CA was imported by the profile and the CN applies), showing that the certificates imported for Safari are not usable).

Test 4: the evaluator opened the Edhita editor and created a directory and a file with some content. The evaluator then opened iEditor and attempted to locate the directory or file unsuccessfully. The evaluator reversed the process, created a directory or file in iEditor and verified that it could not be seen nor accessed by Edhita. The evaluator then installed configuration profiles on the TOE to supply it with certificates and WiFi credentials, connected the TOE to the WLAN network, and accessed https://taun through Safari. The evaluator selected the RSA certificate and verified that the cryptographic details were displayed on the web page, indicating a successful connection.

FDP_ACF_EXT.1.2

TSS Assurance Activities

Assurance Activity AA-FDP_ACF_EXT.1.2-ASE-01

The evaluator shall examine the TSS to verify that it describes which data sharing is permitted between applications, which data sharing is not permitted, and how disallowed sharing is prevented. It is possible to select both "application" and "groups of application", in which case the TSS is expected to describe the data sharing policies that would be applied in each case.

Summary

Section 8.4 in the [ST] describes the *User Data Protection (FDP)* .

Sections 8.4.1 and 8.4.2 describe that applications are sandboxed and can only access data within their own directory in the file tree structure, unless they are part of an app group set up by the application developer, as described in section 8.4.4. During the installation of the application, the installer creates app containers limited to the directories inside the application's sandbox which enforces which files the application can access.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDP_ACF_EXT.1.2-ATE-01

- **Test 1:** The evaluator shall write, or the developer shall provide, two applications, one that saves data containing a unique string and the other, which attempts to access that data. If "groups of applications" is selected, the applications shall be placed into different groups. If "application" is selected, the evaluator shall install the two applications. If "private data" is selected, the application shall not write to a designated shared storage area. The evaluator shall verify that the second application is unable to access the stored unique string. If "the user" is selected, the evaluator shall grant access as the user and verify that the second application is able to access the stored unique string. If "the administrator" is selected, the evaluator shall grant access as the administrator and verify that the second application is able to access the stored unique string. If "a common application developer" is selected, the evaluator shall grant access to an, application with a common application developer to the first, and verify that the application is able to access the stored unique string.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test: the evaluator verified that the iCalendar app exports an interface to the iMail app via the UIDocumentInteractionController to allow invitations received by the Mail system to the Calendar. The evaluator registered an account with iCloud, and ensured that Mail is active. He then created a calendar invite from a different system and invite the iCloud account. The evaluator waited for the email invite to appear with iMail, opened it, opened the attached ical file, and verified that it displayed the event. The evaluator then imported it, and verified in iCalendar that the event is listed. The evaluator then logged out of iCloud.

2.1.3.2 Extended: Protected Data Encryption (FDP_DAR_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.1-ASE-01

The evaluator shall verify that the TSS section of the ST indicates which data is protected by the DAR implementation and what data is considered TSF data. The evaluator shall ensure that this data includes all protected data.

Summary

Section 8.4.6 of the [ST] describes *Keychain Data Protection*.

Section 8.4.6 describes how the data is separated into four different categories in *Table 9: MDM Server Reference Identifiers*. MDM Server Reference Identifiers The security daemon determines which keychain items each process or application can access. *Table 6: Summary of keys and persistent secrets in iOS 12* describes the keys and persistent secrets stored in iOS. *Table 7: Summary of keys and persistent secrets used by the Agent* describes the keys and persistent storage stored by the Agent on iOS. In the summary after Figure 5, the evaluator verified that all file system items (files), file system metadata, files and key chains are encrypted and are considered TSF data protected as DAR.

Guidance Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.1-AGD-01

The evaluator shall review the AGD guidance to determine that the description of the configuration and use of the DAR protection does not require the user to perform any actions beyond configuration and providing the authentication credential. The evaluator shall also review the AGD guidance to determine that the configuration does not require the user to identify encryption on a per-file basis.

Summary

[CCGUIDE] section 3.4.1 *Data-At-Rest (DAR) Protection Configuration* provides related guidance on DAR. It states that data is always encrypted for protection which requires the use of a passcode on the device. Administrators must ensure that TOE users set a passport by using the *forcePin* key in the Passcode Policy Payload via a Configuration Profile.

Users can check that data protection is enabled on their device by looking at the passcode settings screen and also by seeing that a passcode is required to access the device. No further configuration is required.

Test Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.1-ATE-01

Assurance Activity Note: The following test requires the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- **Test 1:** The evaluator shall enable encryption according to the AGD guidance. The evaluator shall create user data (non-system) either by creating a file or by using an application. The evaluator shall use a tool provided by the developer to verify that this data is encrypted when the product is powered off, in conjunction with Test 1 for FIA_UAU_EXT.1.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: There is currently no tool available that is able to bypass the en/decryption functionality for data. The encryption/decryption is performed directly by the storage controller as follows:

- XNU provides the wrapped file meta data to SEP.
- SEP unwraps the meta data to access the plaintext encryption key.
- SEP re-wraps the meta data with a new key that was established during boot with the storage controller.
- SEP sends the wrapped meta data back to XNU.
- XNU sends the wrapped meta data to the storage controller together with the file access request.
- The storage controller unwraps the meta data and performs the file encryption/decryption with the unwrapped key.
- Storage controller receives/sends plaintext file contents from/to XNU.

Therefore, checking that data is stored encrypted is not possible as discussed and agreed with NIAP for the iOS 9, iOS 10, and iOS 11 evaluations. Though, the tester performs the following inspection: check the SEP code to verify the steps 2 through 4 in the SEP code path. Verify that the unwrapped file key is immediately zeroized after the re-wrapping operation. In addition, the developer explained that the following test is performed when new hardware is released:

- Test vectors using the cryptographic implementation of CoreCrypto are generated:
 - for devices up to and including A8:
 - Encryption: AES CBC 128 with key, IV and plaintext
 - Decryption: AES CBC 128 with key, IV and ciphertext
 - for devices starting with A9:
 - Encryption: AES XTS 128 with key, tweak key, IV and plaintext
 - Decryption: AES XTS 128 with key, tweak key, IV and ciphertext
 - for all test vectors, the expected ciphertext (encryption) or plaintext (decryption) is calculated
- The generated test vectors are sent to the storage controller to perform the respective cryptographic operation.
- The resulting data (ciphertext for encryption, plaintext for decryption) is compared to the expected data generated by CoreCrypto for the test vectors. If both match, the test is considered to be successful. Otherwise, the test fails and indicates a broken hardware implementation causing a rework of the hardware.

2.1.3.3 Extended: Sensitive Data Encryption (FDP_DAR_EXT.2)

FDP_DAR_EXT.2.1

TSS Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.2.1-ASE-01

The evaluator shall verify that the TSS includes a description of which data stored by the TSF (such as by native applications) is treated as sensitive. This data may include all or some user or enterprise data and must be specific regarding the level of protection of email, contacts, calendar appointments, messages, and documents.

The evaluator shall examine the TSS to determine that it describes the mechanism that is provided for applications to use to mark data and keys as sensitive. This description shall also contain information reflecting how data and keys marked in this manner are distinguished from data and keys that are not (for instance, tagging, segregation in a "special" area of memory or container, etc.).

Summary

Section 8.3.1 of the [ST]  describes an *Overview of Key Management* .

Section 8.3.1 describes how key management works on iOS. iOS treats all data equally and encrypts all data in the same way: every time a file is created, a new AES key is generated and used to encrypt this file in memory. This AES key is then wrapped by a group key.

Guidance Assurance Activities





No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.2.1-ATE-01

- **Test 1:** *The evaluator shall enable encryption of sensitive data and require user authentication according to the AGD guidance. The evaluator shall try to access and create sensitive data (as defined in the ST and either by creating a file or using an application to generate sensitive data) in order to verify that no other user interaction is required.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

Test 1: The storage encryption outlined for FDP_DAR_EXT.1 is the same that provides the extended storage encryption support by using the class keys. The calling application can select the respective class key to be used to wrap the session key used to protect the data at rest. Thus, the testing of FDP_DAR_EXT.1 covers the data storage part of the test requirement. In addition, the key life cycle discussion provided in FCS_CKM_EXT.4 demonstrates the key management side of the test requirement. Therefore, both analyses fully cover the test requirement of this SFR. The evaluator notes that the data protection mechanism with the encryption operation is always enforced, i.e. there is no configuration option that can enable or disable this functionality. In addition, the evaluator notes that the session key management with the use of all four class keys as illustrated in FCS_CKM_EXT.4 is only present if the user has set a PIN/password/passphrase with the TOE. The guidance outlines how such PIN/passwords/passphrases are set.

FDP_DAR_EXT.2.2

TSS Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.2.2-ASE-01

The evaluator shall review the TSS section of the ST to determine that the TSS includes a description of the process of receiving sensitive data while the device is in a locked state. The evaluator shall also verify that the description indicates if sensitive data that may be received in the locked state is treated differently than sensitive data that cannot be received in the locked state. The description shall include the key scheme for encrypting and storing the received data, which must involve an asymmetric key and must prevent the sensitive data-at-rest from being decrypted by wiping all key material used to derive or encrypt the data (as described in the application note). The introduction to this section provides two different schemes that meet the requirements, but other solutions may address this requirement.

Summary

Section 8.3.1 of the [ST] [\[1\]](#) describes an *Overview of Key Management*.

The paragraph *NSFileProtectionCompleteUnlessOpen* describes how files can be retrieved on iOS although the device is locked, for example by downloading an email attachment while the device is in the locked state. This is achieved using asymmetric cryptography and elliptic curves. iOS specifically uses curve 25519. iOS generates a device-wide asymmetric key for the *NSFileProtectionCompleteUnlessOpen* file class within the Secure Enclave. When data is received by iOS in the locked state, iOS generates another asymmetric key pair. The device-wide public key and the file object private key are used to generate a shared secret (one-pass Diffie-Hellman), which is post-processed by a KDF to obtain a symmetric key. This key is used to encrypt the data that is being received. Then, the file object private key is destroyed with the shared secret when the file is closed and only the public key is stored with the file. To access the file, the user has to provide the correct passcode to unwrap the device-wide private key. The files that have been encrypted with the above scheme have their key re-wrapped by the *NSFileProtectionCompleteUnlessOpen* class key. It is up to the application to check whether the device is locked or not and to re-wrap the file with the correct key.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.2.2-ATE-01

The evaluator shall perform the tests in FCS_CKM_EXT.4 for all key material no longer needed while in the locked state and shall ensure that keys for the asymmetric scheme are addressed in the tests performed when transitioning to the locked state.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[4\]](#).

Test: See the key life cycle testing performed as part of FCS_CKM_EXT.4 which covers the destruction of all class keys, including the private and public class B key referenced in this SFR. The evaluator notes that the asymmetric key handling with its Diffie-Hellman operation as described in the PP with figure 4 is implemented by the class B key pair.

FDP_DAR_EXT.2.3


TSS Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.2.3-ASE-01

The evaluator shall verify that the key hierarchy section of the TSS required for FCS_STG_EXT.2.1 includes the symmetric encryption keys (DEKs) used to encrypt sensitive data. The evaluator shall ensure that these DEKs are encrypted by a key encrypted with (or chain to a KEK encrypted with) the REK and password-derived or biometric-unlocked KEK.

The evaluator shall verify that the TSS section of the ST that describes the asymmetric key scheme includes the protection of any private keys of the asymmetric pairs. The evaluator shall ensure that any private keys that are not wiped and are stored by the TSF are stored encrypted by a key encrypted with (or chain to a KEK encrypted with) the REK and password-derived or biometric-unlocked KEK.

Summary

Section 8.3.1 of the [ST]  describes an *Overview of Key Management* .

Section 8.3.1, and specifically figure 5 , describe how key hierarchy is implemented within iOS. The evaluator verified that the DEKs are encrypted by a key that is chained to the REK. Some keys, like the passcodes, are actually password derived (using the REK as well). Regarding asymmetric cryptography keys, the private keys are wrapped whenever the device is locked. These keys will be unwrapped when the user provides the correct passcode for the device.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.


FDP_DAR_EXT.2.4

TSS Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.2.4-ASE-01

The evaluator shall verify that the TSS section of the ST that describes the asymmetric key scheme includes a description of the actions taken by the TSF for the purposes of DAR upon transitioning to the unlocked state. These actions shall minimally include decrypting all received data using the asymmetric key scheme and re-encrypting with the symmetric key scheme used to store data while the device is unlocked.

Summary

Section 8.3.1 of an [ST]  describes the *Overview of Key Management* .

Asymmetric cryptography is used for verifying signatures of the software, protocols (key establishment), and when the device receives data while it is locked. The private keys are wrapped by symmetric cryptography whenever the device is locked. The symmetric keys used to wrap these keys is destroyed and can only be re-generated whenever the user provides the correct passcode. When the device is unlocked, the keys protecting data received during the locked state are being unwrapped are re-wrapped with a different key.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.3.4 Extended: Subset Information Flow Control (FDP_IFC_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDP_IFC_EXT.1-ASE-01

The evaluator shall verify that the TSS section of the ST describes the routing of IP traffic through processes on the TSF when a VPN client is enabled. The evaluator shall ensure that the description indicates which traffic does not go through the VPN and which traffic does and that a configuration exists for each baseband protocol in which only the traffic identified by the ST author as necessary for establishing the VPN connection (IKE traffic and perhaps HTTPS or DNS traffic) is not encapsulated by the VPN protocol (IPsec). The evaluator shall verify that the TSS section describes any differences in the routing of IP traffic when using any supported baseband protocols (e.g. Wi-Fi or, LTE).

Summary

Section 8.9.4.2 IPsec General specifies that:

"IPsec is implemented in the TOE natively, as part of iOS, hence the packets are processed by the TOE. Packets are processed in little-endian order. There is no separate "client" application; the VPN tunnels are configured and controlled by Network Extension Framework, which is a part of the host operating system's Core OS Layer. The TOE implements the IPsec protocol as specified in RFC 4301. Configuration of VPN connection setting, such as, authentication method and algorithm selection, is performed by the IPsec VPN client administrator."

AlwaysOn VPN means that **all** IP traffic is routed through the VPN, without exceptions. It provides the organization with full-control over the device traffic.

Guidance Assurance Activities

Assurance Activity AA-FDP_IFC_EXT.1-AGD-01

The evaluator shall verify that one (or more) of the following options is addressed by the documentation:

- The description above indicates that if a VPN client is enabled, all configurations route all Data Plane traffic through the tunnel interface established by the VPN client
- The AGD guidance describes how the user and/or administrator can configure the TSF to meet this requirement.
- The API documentation includes a security function that allows a VPN client to specify this routing.

Summary

[CCGUIDE] [📄](#) section 3.3.5 *VPN Configuration* provides related guidance on VPN configuration.. It states that the VPN must be in its Always-On configuration in the evaluated configuration. This configuration enables the organization to have full control over supervised device traffic by tunneling all IP traffic back to the organization. For the TOE users, no configuration is required. The administrator may perform this configuration using the VPN Policy Payload in a Configuration Profile. Also, section 3.3.6 *Keys for Configuring Network Protocols* provides table 11 "Essential Keys for the VPN Payload" outlining the keys and their values that the administrator may specified in the VPN Policy Payload.

The evaluator also examined the provided API documentation [NETEXT] [📄](#) which covers configuration of VPN tunnels.

Assurance Activity AA-FDP_IFC_EXT.1-VPN-AGD-01

The evaluator shall verify that the following is addressed by the documentation:

- The description above indicates that if a VPN client is enabled, all configurations route all IP traffic (other than IP traffic required to establish the VPN connection) through the VPN client.
- The AGD guidance describes how the user and/or administrator can configure the TSF to meet this requirement.

Summary

[CCGUIDE][\[1\]](#) section 3.3.5 *VPN Configuration* provides related guidance for VPN configuration. It states the following:

- In the evaluated configuration, the VPN must be in its Always-On configuration which allows the organization to have full control over supervised device traffic by tunneling all IP traffic back to the organization. Always-On is enabled by setting the *VPNTType* key to 'AlwaysOn' in the Configuration Profile.
- For the user, there is no configuration required.
- For administrator, they may use the VPN Policy Payload to configure systemwide VPN based on IPsec. Further instructions for the administrators to configure the VPN Policy Payload is provided in section 3.3.6 *Keys for Configuring Network Protocols* of [CCGUIDE][\[1\]](#).

Test Assurance Activities

Assurance Activity AA-FDP_IFC_EXT.1-ATE-01

- **Test 1:** If the ST author identifies any differences in the routing between Wi-Fi and cellular protocols, the evaluator shall repeat this test with a base station implementing one of the identified cellular protocols.
 - *Step 1:* The evaluator shall enable a Wi-Fi configuration as described in the AGD guidance (as required by *FTP_ITC_EXT.1*). The evaluator shall use a packet sniffing tool between the wireless access point and an Internet-connected network. The evaluator shall turn on the sniffing tool and perform actions with the device such as navigating to websites, using provided applications, and accessing other Internet resources. The evaluator shall verify that the sniffing tool captures the traffic generated by these actions, turn off the sniffing tool, and save the session data.
 - *Step 2:* The evaluator shall configure an IPsec VPN client that supports the routing specified in this requirement, and if necessary, configure the device to perform the routing specified as described in the AGD guidance. The evaluator shall turn on the sniffing tool, establish the VPN connection, and perform the same actions with the device as performed in the first step. The evaluator shall verify that the sniffing tool captures traffic generated by these actions, turn off the sniffing tool, and save the session data.
 - *Step 3:* The evaluator shall examine the traffic from both step one and step two to verify that all Data Plane traffic is encapsulated by IPsec. The evaluator shall examine the Security Parameter Index (SPI) value present in the encapsulated packets captured in Step two from the TOE to the Gateway and shall verify this value is the same for all actions used to generate traffic through the VPN. Note that it is expected that the SPI value for packets from the Gateway to the TOE is different than the SPI value for packets from the TOE to the Gateway. The evaluator shall be aware that IP traffic on the cellular baseband outside of the IPsec tunnel may be emanating from the baseband processor and shall verify with the manufacturer that any identified traffic is not emanating from the application processor.
 - *Step 4:* The evaluator shall perform an ICMP echo from the TOE to the IP address of another device on the local wireless network and shall verify that no packets are sent using the sniffing tool. The evaluator shall attempt to send packets to the TOE outside the VPN tunnel (i.e. not through the VPN gateway), including from the local wireless network, and shall verify that the TOE discards them.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[1\]](#).

Test 1, Step 1: the evaluator connected the TOE to the WiFi AP on the test laptop, and access <https://server> through Safari on the TOE. The evaluator captured that traffic and disconnected from the WLAN.

Step 2: the evaluator loaded the IPsec configuration profile onto the TOE using the Apple Configurator 2, started Strongswan on the test laptop, started the WiFi AP, and verified that the VPN connection is automatically established. The evaluator then access <https://server>. The evaluator disconnected from the WLAN, and re-connected and verified that the VPN connection is automatically re-established.

Step 3: the evaluator recorded all the traffic from Step 2, and reviewed it. The evaluator verified that HTTP traffic is encapsulated into ESP, and verified the SPI for both directions. According to the developer, the always-on VPN implies that all traffic is always routed to the VPN channel. If two channels exist (one on WiFi and one on Cellular), the VPN channel on WiFi is preferred. If no VPN channel is found, no traffic other than the captive portal traffic and DNS traffic prior to establishing the VPN link is allowed.

Step 4: the evaluator connected the TOE to the AP, where Strongswan is not started, and verified that the VPN connection is not established. The evaluator pinged the TOE from the test laptop (to the IP address awarded to the TOE by the VPN connection) and verified that there was no reply from the TOE.

Assurance Activity AA-FDP_IFC_EXT.1-VPN-ATE-01

The evaluator shall perform the following test:

Step 1 - The evaluator shall use the platform to enable a network connection without using IPsec. The evaluator shall use a packet sniffing tool between the platform and an Internet-connected network. The evaluator shall turn on the sniffing tool and perform actions with the device such as navigating to websites, using provided applications, accessing other Internet resources (Use Case 1), accessing another VPN client (Use Case 2), or accessing an IPsec-capable network device (Use Case 3). The evaluator shall verify that the sniffing tool captures the traffic generated by these actions, turn off the sniffing tool, and save the session data.

Step 2 - The evaluator shall configure an IPsec VPN client that supports the routing specified in this requirement, and if necessary, configure the device to perform the routing specified as described in the AGD guidance. The evaluator shall turn on the sniffing tool, establish the VPN connection, and perform the same actions with the device as performed in the first step. The evaluator shall verify that the sniffing tool captures traffic generated by these actions, turn off the sniffing tool, and save the session data.

Step 3 - The evaluator shall examine the traffic from both step one and step two to verify that all IP traffic, aside from and after traffic necessary for establishing the VPN (such as IKE, DNS, and possibly HTTPS), is encapsulated by IPsec.

Step 4 - The evaluator shall attempt to send packets to the TOE outside the VPN connection and shall verify that the TOE discards them.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[1\]](#).

The test is identical to FDP_IFC_EXT.1 {MDF} and is thus covered by the test execution.

2.1.3.5 Extended: Storage of Critical Biometric Parameters (FDP_PBA_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDP_PBA_EXT.1-ASE-01

The evaluator shall determine that the TSS contains a description of the activities that happen during biometric authentication.

Summary

Section 8.5.1 of the [ST] [\[1\]](#) specifies what happens when a user tries to authenticate to a device using biometrics. Biometric authentication does not produce feedback unless the input is rejected. In that case, when an invalid fingerprint is provided or can't be authenticated, an error message is displayed to the user. For facial recognition, the device will vibrate. When three invalid biometric samples are presented the device will offer password/passcode authentication. After five invalid biometric samples are presented passcode authentication is required.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDP_PBA_EXT.1-ATE-01

The evaluator shall ensure that the authentication template is protected either using a PIN or by other secure means, as specified by the vendor.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[1\]](#).

The evaluator made sure that, when performing the other tests, the evaluator could not access the authentication settings without authenticating first to the TOE.

2.1.3.6 Extended: User Data Storage (FDP_STG_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDP_STG_EXT.1-ASE-01

The evaluator shall ensure the TSS describes the Trust Anchor Database implemented that contain certificates used to meet the requirements of this PP. This description shall contain information pertaining to how certificates are loaded into the store, and how the store is protected from unauthorized access (for example, UNIX permissions) in accordance with the permissions established in FMT_SMF_EXT.1 and FMT_MOF_EXT.1.1.

Summary

Section 8.5.2 in the [ST] [\[1\]](#) describes how certificates are used in iOS.

The Apple certificate is installed in ROM during manufacturing. Other certificates can be imported by a user (if allowed by the policy) or installed using configuration profiles. The trust anchor database is the database with the CA roots used in TLS and the other protocols. That is hard-coded and cannot be changed. The database is protected the same way as the binary code of the TOE, i.e., by the TOE integrity check and write protection.

Every certificate has a certificate type which defines what the certificate is used for. This ensures that only certificates defined for a specific application can be used for this application, and nothing else.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.3.7 1 Extended: Inter-TSF User Data Transfer Protection (FDP_UPC_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDP_UPC_EXT.1-ASE-01

The evaluator shall examine the TSS to determine that it describes that all protocols listed in the TSS are specified and included in the requirements in the ST.

Summary

Section 8.9 in the [ST] [\[ST\]](#) describes the *Trusted Path/Channels (FTP)* .

Section 8.9 is divided into several sections where each lists the protocol supported by the TOE: EAP-TLS, TLS, IPsec, Bluetooth and WLAN. These protocols are specified and included in the requirements in the [ST] [\[ST\]](#) .

Guidance Assurance Activities


Assurance Activity AA-FDP_UPC_EXT.1-AGD-01




The evaluator shall verify that the API documentation provided according to Section 5.2.2 includes the security functions (protection channel) described in these requirements, and verify that the APIs implemented to support this requirement include the appropriate settings/parameters so that the application can both provide and obtain the information needed to assure mutual identification of the endpoints of the communication as required by this component. The evaluator shall write, or the developer shall provide access to, an application that requests protected channel services by the TSF. The evaluator shall verify that the results from the protected channel match the expected results according to the API documentation. This application may be used to assist in verifying the protected channel assurance activities for the protocol requirements.

Summary

[CCGUIDE] [\[CCGUIDE\]](#) provides table 3 "SFR Configuration Requirements" mapping SFRs and related guidance. Table entry "FDP_UPC_EXT.1" states that depending on the network protocol used, configuration is possible (e.g., IPsec) while other options are not configurable (e.g., Bluetooth). General guidance for trusted channels is provided in section 3.3 *Network protocols*. Guidance for each supported protocol (in the evaluate configuration) is provided in the following subsections:

- TLS/EAL-TLS: [CCGUIDE] [\[CCGUIDE\]](#) sections 3.3.1 and 3.3.2
- HTTPS: section 3.3.2
- IPsec: [CCGUIDE] [\[CCGUIDE\]](#) sections 3.3.3, 3.3.5, 3.3.6
- Bluetooth: [CCGUIDE] [\[CCGUIDE\]](#) sections 3.3.4, 3.6.7

These protocols may be configured via a Configuration Profile described in [IOS_CFG] . Additionally, the evaluator examined the following provided API documentation:


- [NETWORK]  *Network Framework*: high-level description of network framework covering TLS and HTTPS.
- [SECTRAN]  *Secure Transport Reference*: secure transport API for network communication protocols including TLS.
- [COREBT]  *Core Bluetooth*: Core Bluetooth framework.




While performing guidance assurance activities for these protocols, the evaluator also verified that sufficient guidance is provided. Additionally, this was also supported by the related test assurance activities.

Assurance Activity AA-FDP_UPC_EXT.1-AGD-02

The evaluator shall confirm that the operational guidance contains instructions necessary for configuring the protocol(s) selected for use by the applications.

Summary

[CCGUIDE]  provides guidance for the following:

- TLS/EAL-TLS: [CCGUIDE]  sections 3.3.1 and 3.3.2
- HTTPS: section 3.3.2
- IPsec: [CCGUIDE]  sections 3.3.3, 3.3.5, 3.3.6
- Bluetooth: [CCGUIDE]  sections 3.3.4, 3.6.7

These protocols may be configured via a Configuration Profile described in [IOS_CFG] .





While performing guidance assurance activities for these related protocols, the evaluator also verified that sufficient guidance is provided. Additionally, this was also supported by the related test assurance activities.

Test Assurance Activities

Assurance Activity AA-FDP_UPC_EXT.1-ATE-01

[The evaluator shall write, or the developer shall provide access to, an application that requests protected channel services by the TSF. The evaluator shall verify that the results from the protected channel match the expected results according to the API documentation. This application may be used to assist in verifying the protected channel assurance activities for the protocol requirements.]

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

Test 1: TLS/HTTPS: tested implicitly with all EAP-TLS and HTTPS tests above and below. For Bluetooth, the evaluator made the BlueZ stack on Linux visible, enabled Bluetooth on the TOE and paired both devices.

Test 2: TLS/HTTPS: tested implicitly with all EAP-TLS and HTTPS tests above and below. For Bluetooth, the evaluator ensured that the TOE was disconnected invoked hcidump to capture the traffic, initiated the Bluetooth connection, opened the pcap file and inspected the traffic. The evaluator verified that the event displayed that Link Level Encryption is ON. The evaluator used the command hcitool con to show the connections as encrypted.

Assurance Activity AA-FDP_UPC_EXT.1-ATE-02

The evaluator shall also perform the following tests:

- **Test 1:** The evaluators shall ensure that the application is able to initiate communications with an external IT entity using each protocol specified in the requirement, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- **Test 2:** The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator used the BlueZ software to pair the test laptop with the TOE.

Test 2: TLS/HTTPS is implicitly tested through EAP-TLS and HTTPS tests. For Bluetooth, the evaluator connected the TOE to the laptop using BlueZ software and invoked hcidump to record the packets. The evaluator then used Wireshark to analyze the packets and searched for the string "Encryption Change". The evaluator then verified that HCI event displays that Link Level Encryption is ON (0x01 or 0x02) and that the command "hcidtool con" shows the connection as encrypted:

```
$ hcidtool con
```

Connections:

```
> ACL 90:8D:6C:5F:44:4D handle 256 state 1 lm MASTER AUTH ENCRYPT
```

2.1.3.8 Full Residual Information Protection (FDP_RIP.2)

TSS Assurance Activities

Assurance Activity AA-FDP_RIP.2-VPN-ASE-01

Requirement met by the platform

The evaluator shall examine the TSS to verify that it describes (for each supported platform) the extent to which the client processes network packets and addresses the FDP_RIP.2 requirement.

Summary

Section 8.9.4.2, *IPsec General*, mentions that IPsec is implemented as part of the TOE and part of iOS. The packets are hence processed by the TOE in little-endian order. There is no separate client application that processes packets specifically for the VPN.

Assurance Activity AA-FDP_RIP.2-VPN-ASE-02

Requirement met by the TOE

"Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

Summary

Please refer to AA-FDP_RIP.2-VPN-ASE-01.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.4 Identification and authentication (FIA)

2.1.4.1 Extended: Authentication Failure Handling (FIA_AFL_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FIA_AFL_EXT.1-ASE-01

The evaluator shall ensure that the TSS describes that a value corresponding to the number of unsuccessful authentication attempts since the last successful authentication is kept for each Authentication Factor interface. The evaluator shall ensure that this description also includes if and how this value is maintained when the TOE loses power, either through a graceful powered off or an ungraceful loss of power. The evaluator shall ensure that if the value is not maintained, the interface is after another interface in the boot sequence for which the value is maintained.

Summary


Section 8.5 of [ST]  describes the *Identification and Authentication (FIA)* .

A user can authenticate to the TOE using a passcode. The value of the number of unsuccessful authentication attempts since the last authentication is kept in a system file which persists in the event of a graceful or ungraceful loss of power to the TOE. This value can be set by the administrator of the device. After 3 attempts to authenticate through biometrics, the passcode is offered. After 5 attempts to authenticate through biometrics, the user is forced to use the passcode to authenticate.

Assurance Activity AA-FIA_AFL_EXT.1-ASE-02

If the TOE supports multiple authentication mechanisms, the evaluator shall ensure that this description also includes how the unsuccessful authentication attempts for each mechanism selected in FIA_UAU.5.1 is handled. The evaluator shall verify that the TSS describes if each authentication mechanism utilizes its own counter or if multiple authentication mechanisms utilize a shared counter. If multiple authentication mechanisms utilize a shared counter, the evaluator shall verify that the TSS describes this interaction.

Summary

Section 8.5 of the [ST]  describes the *Identification and Authentication (FIA)* .

iOS supports three different authentication modes claimed for the TOE: passcode, Face ID or Touch ID. The passcode and the biometric methods on the phone share different unsuccessful attempt counters. Section 8.5.1 describes the Face ID and Touch ID authentication mechanism, and section 8.5 describes the general authentication rules on iOS. The passcode counter is stored in a file on the file system, and as soon as iOS triggers an authentication failure, the counter is updated, which prevents an attacker from powering-off the phone before the counter is updated. After 3 attempts have been made with Face ID or Touch ID, the user is offered to use the passcode. After 5 attempts, the user must authenticate through using the passcode only. In addition, the following behavior applies to biometric authentication methods. A passcode must be supplied for additional security validation when:

- The device has just been turned on or restarted.
- The device hasn't been unlocked for more than 48 hours.
- The passcode hasn't been used to unlock the device in the last 156 hours (six and a half days) and Face ID hasn't unlocked the device in the last 4 hours.
- The device has received a remote lock command.
- After five unsuccessful attempts to match.
- After initiating power off/Emergency SOS.

Assurance Activity AA-FIA_AFL_EXT.1-ASE-03

The evaluator shall confirm that the TSS describes how the process used to determine if the authentication attempt was successful. The evaluator shall ensure that the counter would be updated even if power to the device is cut immediately following notifying the TOE user if the authentication attempt was successful or not.

Summary

Section 8.5 specifies that

"The number of failed authentication attempts is maintained in a system file which will persist in the event of graceful or ungraceful loss of power to the TOE. The counter maintaining the number of failed consecutive logon attempts is increased by one immediately once the TOE has identified that the passcode is incorrect. The increment of the counter is completed before the UI informs the user about the failed logon attempt."

Regarding the unsuccessful attempts, section 8.5 also specifies:


"Biometric authentication inputs do not produce feedback to the user unless an input is rejected. When an invalid fingerprint sample is given or cannot be authenticated, a simple error message is returned to the user to try again. If three invalid biometric samples are presented the device will offer passcode entry. After five invalid biometric samples are presented passcode authentication is required. For Face ID, when an invalid facial sample is given or cannot be authenticated, the user needs to swipe up before a second attempt can occur and passcode entry will be presented to the user as an option. After five invalid Face ID attempts, the device will vibrate and passcode entry must be used."


Guidance Assurance Activities

Assurance Activity AA-FIA_AFL_EXT.1-AGD-01

The evaluator shall verify that the AGD guidance describes how the administrator configures the maximum number of unique unsuccessful authentication attempts.

Summary

[CCGUIDE]  section 3.5.4 *Authentication Attempt Configuration* provides instructions for setting the authentication attempt. It states that in the evaluated configuration the TOE users are not allowed configure the maximum number of failed authentication attempts.

The administrator performs this configuration via Configuration Profile by specifying in the Passcode Policy Payload the key *maxFailedAttempts* with the value between 2 and 11. Additional detail of the Passcode Policy Payload is provided in [CCGUIDE]  section 3.5.1 *Passcode Authentication Configuration*.

Test Assurance Activities

Assurance Activity AA-FIA_AFL_EXT.1-ATE-01

- **Test 1:** The evaluator shall configure the device with all authentication mechanisms selected in FIA_UAU.5.1. The evaluator shall perform the following tests for each available authentication interface:
 - **Test 1a:** The evaluator shall configure the TOE, according to the AGD guidance, with a maximum number of unsuccessful authentication attempts. The evaluator shall enter the locked state and enter incorrect passwords until the wipe occurs. The evaluator shall verify that the number of password entries corresponds to the configured maximum and that the wipe is implemented.
 - **Test 1b:** [conditional] If the TOE supports multiple authentication mechanisms the previous test shall be repeated using a combination of authentication mechanisms confirming that the critical authentication mechanisms will cause the device to wipe and that when the maximum number of unsuccessful authentication attempts for a non-critical authentication mechanism is exceeded, the device limits authentication attempts to other available authentication mechanisms. If multiple authentication mechanisms utilize a shared counter, then the evaluator shall verify that the maximum number of unsuccessful authentication attempts can be reached by using each individual authentication mechanism and a combination of all authentication mechanisms that share the counter.
- **Test 2:** The evaluator shall repeat test one, but shall power off (by removing the battery, if possible) the TOE between unsuccessful authentication attempts. The evaluator shall verify that the total number of unsuccessful authentication attempts for each authentication mechanism corresponds to the configured maximum and that the critical authentication mechanisms cause the device to wipe. Alternatively, if the number of authentication failures is not maintained for the interface under test, the evaluator shall verify that upon booting the TOE between unsuccessful authentication attempts another authentication factor interface is presented before the interface under test.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[1\]](#).

Test 1: the evaluator set the maximum of authentication failed attempts to 3, and provided an incorrect passcode three times with three different passcodes. The tested unlocked the device and verified that the data was wiped.

Test 2: the evaluator tried to unlock the device with the wrong password twice, rebooted the device, and tried to unlock the device with another wrong password. The evaluator then unlocked the device and verified that the data was wiped.

2.1.4.2 Extended: Bluetooth User Authorization (FIA_BLT_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FIA_BLT_EXT.1-ASE-01

The evaluator shall examine the TSS to ensure that it contains a description of when user permission is required for Bluetooth pairing, and that this description mandates explicit user authorization via manual input for all Bluetooth pairing, including application use of the Bluetooth trusted channel and situations where temporary (non-bonded) connections are formed.

Summary

Section 8.9.2 of the [ST][\[1\]](#) describes the *Bluetooth* protocol.

The evaluator verified that section 8.9.2 specifically addresses how a user authorizes Bluetooth pairing through the Bluetooth interface in the settings of iOS. This section also has the following statement:

"During the pairing time, another device (or the iOS) can send a pairing request. Commonly, a six-digit number is displayed on both sides which must be manually matched by a user, i.e. the PIN is shown and the user must accept it before the pairing completes. If one device does not support this automatic exchange of a PIN, a window for entering a manual PIN is shown. That PIN must match on both sides."

Guidance Assurance Activities

Assurance Activity AA-FIA_BLT_EXT.1-AGD-01

The evaluator shall examine the API documentation provided according to Section 5.2.2 and verify that this API documentation does not include any API for programmatic entering of pairing information (e.g. PINs, numeric codes, or "yes/no" responses) intended to bypass manual user input during pairing.

Summary

The evaluator examined the provided API documentation for Bluetooth [COREBT] [Core Bluetooth framework](#) to ensure that there is no means for bypassing manual user input during pairing.

Assurance Activity AA-FIA_BLT_EXT.1-AGD-02

The evaluator shall examine the AGD guidance to verify that these user authorization screens are clearly identified and instructions are given for authorizing Bluetooth pairings.

Summary

[CCGUIDE] [section 3.3.4 Bluetooth Configuration](#) provides related guidance for configuring Bluetooth which references the iPhone and iPad User Guides [iPhone_UG] [and](#) [iPad_UG] [section Connect bluetooth devices](#) for how to turn Bluetooth on and off and how to pair and un-pair a Bluetooth device. This involves turning on the Bluetooth feature on the TOE device (go to Settings>>Bluetooth), choose the Bluetooth accessory, and then enter a passkey or PIN if prompted. When pairing is complete, user can use the Bluetooth accessory with their iOS device. Bluetooth can also be disassociated via the Control Center.

Additionally, [CCGUIDE] [section 3.3.4](#) states that prior to the pairing the mobile device must first be discovered. Two conditions must be met for the mobile device to become discoverable: Bluetooth must be enabled, and the Bluetooth configuration panel must be both active and in the foreground. If the Bluetooth configuration panel is not the active panel, or if Bluetooth is disabled, the mobile device is not discoverable. There is no other method to make the mobile device discoverable or not discoverable.

Test Assurance Activities

Assurance Activity AA-FIA_BLT_EXT.1-ATE-01

The evaluator shall perform the following test:

- **Test 1:** The evaluator shall perform the following steps:
 - Step 1: Initiate pairing with the TOE from a remote Bluetooth device that requests no man-in-the-middle protection, no bonding, and claims to have NoInputNoOutput input-output (IO) capability. (Such a device will attempt to evoke behavior from the TOE that represents the minimal level of user interaction that the TOE supports during pairing.)
 - Step 2: Verify that the TOE does not permit any Bluetooth pairing without explicit authorization from the user (e.g. the user must have to minimally answer "yes" or "allow" in a prompt).

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1, Step 1: the evaluator used the BlueZ software to pair the TOE and the test laptop. The evaluator verified that during the process, a PIN was displayed and the user is required to approve the PIN. The evaluator verified that the same PIN is shown on the remote Bluetooth test laptop.

Step 2: the evaluator ensured that the TOE was disconnected from the test laptop, invoked hcidump to capture the Bluetooth traffic, and initiated a Bluetooth connection between the TOE and the test laptop. The evaluator then opened the pcap file in Wireshark and search for string: "Encryption Change", and verified that HCI event displays that Link Level Encryption is ON (0x01), and that the command "hcidump con" shows the connection as encrypted:

```
$ hcidump con
```

```
Connections:
```

```
> ACL 90:8D:6C:5F:44:4D handle 256 state 1 lm MASTER AUTH ENCRYPT
```

2.1.4.3 Extended: Bluetooth Mutual Authentication (FIA_BLT_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FIA_BLT_EXT.2.1-ASE-01

The evaluator shall ensure that the TSS describes how data transfer of any type is prevented before the Bluetooth pairing is completed. The TSS shall specifically call out any supported RFCOMM and L2CAP data transfer mechanisms. The evaluator shall ensure that the data transfers are only completed after the Bluetooth devices are paired and mutually authenticated.

Summary

Section 8.9.2 of the [ST] describes the *Bluetooth* protocol.

The evaluator ensured that section 8.9.2 describes that the Bluetooth pairing has to be complete before data can be exchanged. The evaluator also verified that the TSS does not call out any Radio frequency communication (RFCOMM) or Logical link control and adaptation protocol (L2CAP) data transfer mechanism. Paragraph three of section 8.9.2 describes how the device has to authenticate for the pairing to successfully complete, and data to be exchanged.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_BLT_EXT.2.1-ATE-01

The evaluator shall perform the following test:

- **Test 1:** *The evaluator shall use a Bluetooth tool to attempt to access TOE files using the OBEX Object Push service and verify that pairing and mutual authentication are required by the TOE before allowing access. (If the OBEX Object Push service is unsupported on the TOE, a different service that transfers data over Bluetooth L2CAP and/or RFCOMM may be used in this test.)*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the test performed for FIA_BLT_EXT.1 demonstrates this functionality as well: data can only be communicated once a connection is established. Such connection can only be established when devices are either manually or automatically paired. Automatic pairing can only be enabled on the TOE after an initial manual pairing was successful. Open Bluetooth configuration on TOE General -> Bluetooth to make the device visible, make another device visible to the TOE, verify that the remote device is visible by the TOE on the Bluetooth screen, verify that the device is in a "Connected" state.

2.1.4.4 Rejection of Duplicate Bluetooth Connections (FIA_BLT_EXT.3)

TSS Assurance Activities

Assurance Activity AA-FIA_BLT_EXT.3.1-ASE-01

The evaluator shall ensure that the TSS describes how Bluetooth connections are maintained such that two devices with the same Bluetooth device address are not simultaneously connected and such that the initial connection is not superseded by any following connection attempts.

Summary

Section 8.9.2 in the [ST] describes the *Bluetooth* protocol within iOS.

Section 8.8.2 describes that a local database of all Bluetooth device addresses is kept for paired devices. This database is checked prior to any automatic connection attempt. Bluetooth cannot establish more than one connection. Multiple connection attempts from the same BD_ADDR for an established connection will be discarded.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_BLT_EXT.3.1-ATE-01

The evaluator shall perform the following test:

- **Test 1:** *The evaluator shall perform the following steps:*
 - *Step 1: Make a Bluetooth connection between the TOE and a remote Bluetooth device with address a known address (BD_ADDR1).*
 - *Step 2: Attempt a connection to the same TOE from a second remote Bluetooth device claiming to have a Bluetooth device address matching BD_ADDR1.*
 - *Step 3: Using a Bluetooth protocol analyzer, verify that the second connection attempt is ignored by the TOE and that the initial connection to the device with BR_ADDR1 is unaffected.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1, Step 1: the evaluator paired the TOE with another device and obtained the Bluetooth ID of the remote device.

Step 2: the evaluator set the Bluetooth dongle ID to the ID of the remote device, scanned the TOE, started the Bluetooth sniffer on the Linux test laptop, tried to connect to the TOE with the Bluetooth dongle and verified that the connection fails. The evaluator also verified that the Bluetooth connection remains unaltered.

Step 3: the evaluator verified with the pcap file that the connection failed.

2.1.4.5 Extended: Secure Simple Pairing (FIA_BLT_EXT.4)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_BLT_EXT.4-ATE-01

- **Test 1:** The evaluator shall perform the following steps:
 - Step 1: Initiate pairing with the TOE from a remote Bluetooth device that supports Secure Simple Pairing.
 - Step 2: During the pairing process, observe the packets in a Bluetooth protocol analyzer and verify that the TOE claims support for both "Secure Simple Pairing (Host Support)" and "Secure Simple Pairing (Controller Support)" during the LMP Features Exchange.
 - Step 3: Verify that Secure Simple Pairing is used during the pairing process.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[1\]](#).

Test 1, Step 1: on the TOE and test laptop, the evaluator removed the paired devices. The evaluator then started the Bluetooth sniffer, initiated and completed the pairing of the TOE with the test laptop.

Step 2: the evaluator opened the pcap file obtained in step 1 with Wireshark and searched for the packet "Read Remote Supported Features" and verified that the option for "Secure Simple Pairing" is set to True. The evaluator then opened the pcap file with Wireshark and searched for the packet "Read Remote Extended Features Complete" and verified that the option for "Secure Simple Pairing Host" is set to True.

Step 3: the evaluator searched in the pcap file for the packet "Encryption Change" which marks the successful Bluetooth connection.

2.1.4.6 Extended: Password Management (FIA_PMG_EXT.1)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FIA_PMG_EXT.1-AGD-01

The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length.

Summary

[CCGUIDE] [\[1\]](#) section 3.5.1 *Passcode Authentication Configuration* provides guidance to administrators on the composition of strong passcode which consists of:

- Passcode composing of any combination of upper and lower case letters, numbers, and special characters: "!", "@", "#", "\$", "%", "^", "&", "*", "(", ")", ";
- Passcode length up to 16 characters

Passcode policy is defined by administrator using the Passcode Policy Payload in a Configuration Profile described in [IOS_CFG] [\[1\]](#). The Passcode Policy Payload presents the administrator with an alphanumeric passcode entry mechanism, which allows for the entry of arbitrarily long and complex passcodes including the selection of special characters. The administrator must set the configuration keys *allowSimple* to 'false'; *RequireAlphanumeric* to 'Yes' and *minLength* to a value defined by the organization's policy (i.e., greater than zero).

Test Assurance Activities

Assurance Activity AA-FIA_PMG_EXT.1-ATE-01

The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case.

- **Test 1:** *The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[1\]](#).

Test 1: the evaluator configured the password policy rule using the APple Configurator 2 with at least 4 characters, and at least 1 complex character. The evaluator loaded the profile onto the TOE, and went to change the passcode (or waited until the change password request appeared). The evaluator set the passcode to "1234" and verified that it is rejected. The evaluator set the password to something complex (i.e. "zaq1@") and verified that it is accepted. The evaluator then locked the TOE and unlocked it with the new password.

2.1.4.7 Extended: Authentication Throttling (FIA_TRT_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FIA_TRT_EXT.1-ASE-01

The evaluator shall verify that the TSS describes the method by which authentication attempts are not able to be automated. The evaluator shall ensure that the TSS describes either how the TSF disables authentication via external interfaces (other than the ordinary user interface) or how authentication attempts are delayed in order to slow automated entry and shall ensure that this delay totals at least 500 milliseconds over 10 attempts for all authentication mechanisms selected in FIA_UAU.5.1.

Summary

Section 8.5 of the [ST] describes the *Identification and Authentication (FIA)* .

The evaluator reviewed this section and determined that whenever a passcode is entered, iOS forces the PBKDF2 function to iterate for 80 milliseconds. In addition, the TSF enforces a delay of 5 seconds between repeated failed authentication.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.4.8 Multiple Authentication Mechanisms (FIA_UAU.5)

TSS Assurance Activities

Assurance Activity AA-FIA_UAU.5-ASE-01

The evaluator shall ensure that the TSS describes each mechanism provided to support user authentication and the rules describing how the authentication mechanism(s) provide authentication.

Summary

Section 8.5 of the [ST] describes the *Identification and Authentication (FIA)*.

The evaluator reviewed this section and verified that the passcode, Face ID and Touch ID are the only ways that the user can authenticate, as described in section 8.5 and as claimed in FIA_UAU.5 . They cannot, however, be used as two-factor authentication. After 3 attempts have been made with Face ID or Touch ID, the user is offered to use the passcode. After 5 attempts, the user must authenticate through using the passcode only. In addition, the following behavior applies to biometric authentication methods. A passcode must be supplied for additional security validation when:

- The device has just been turned on or restarted.
- The device hasn't been unlocked for more than 48 hours.
- The passcode hasn't been used to unlock the device in the last 156 hours (six and a half days) and Face ID hasn't unlocked the device in the last 4 hours.
- The device has received a remote lock command.
- After five unsuccessful attempts to match.
- After initiating power off/Emergency SOS.

Guidance Assurance Activities

Assurance Activity AA-FIA_UAU.5-AGD-01

The evaluator shall verify that configuration guidance for each authentication mechanism is addressed in the AGD guidance.

Summary

Per the definition of FIA_UAU.5 from [ST] [\[ST\]](#), the TOE provides password (passcode), fingerprint (Touch ID), and face (Face ID) for user authentication.

[CCGUIDE] [\[CCGUIDE\]](#) provides related guidance for each of the authentication mechanism as follows:

- Passcode: section 3.5.1 *Passcode Authentication Configuration*
- Touch ID: section 3.5.3 *Biometric Authentication Factors*
- Face ID: section 3.5.3 *Biometric Authentication Factors*

In previous assurance activities, the evaluator already determined that sufficient guidance is provided for each of the authentication mechanism noted above.

Test Assurance Activities

Assurance Activity AA-FIA_UAU.5-ATE-01

- **Test 1:** For each authentication mechanism selected, the evaluator shall enable that mechanism and verify that it can be used to authenticate the user at the specified authentication factor interfaces.
- **Test 2:** For each authentication mechanism rule, the evaluator shall ensure that the authentication mechanism(s) behave accordingly.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[PP_MD_V3.1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[EP_MDM_AGENT_V3.0\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[PP_WLAN_CLI_EP_V1.0\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[MOD_VPN_CLI_V2.1\]](#).

Test 1: the password test is covered with FIA_BMG_EXT.1 Test 2 as these tests require to unlock the device and verify successful and failed login attempts. For the fingerprint testing, the evaluator added its fingerprint to the device, locked the device and unlocked it with the fingerprint. The evaluator then added their face to FaceID on the compatible TOE and locked and unlocked the device using facial recognition.

2.1.4.9 Extended: Accuracy of Biometric Authentication (FIA_BMG_EXT.1)

FIA_BMG_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.1.1-ASE-01

[TD0301] The evaluator shall verify that the TSS contains evidence supporting the testing and calculations completed to determine the FAR and FRR. Appendix H provides guidance to how this testing could be completed and to what error bars are expected when the Rule of 3 is applied. The evaluator shall consult Appendix H as a reference, but should not treat it as a mandate. The evaluator shall verify that the TSS contains evidence of whether online or offline testing was used. If offline testing was completed, evidence describing the differences between the biometric system used for testing and the TOE in the evaluated configuration, if any must be included.

The following documentation is not required to be part of the TSS - it may be submitted as a separate proprietary document. The evaluator shall verify the evidence includes how many imposters were used for testing and that the testing describes how imposters are compared to enrolled users, for example, if multiple devices for online testing or full cross-comparison for offline testing was used. Adequate documentation is required to demonstrate that testing was completed to support the claimed FAR and FRR.

Summary

The evaluator reviewed section 8.5.1 which contains the *Biometric Authentication* information from the TOE. Specifically, sections 8.5.1.1 and 8.5.1.3 explain the accuracy of biometric authentication for the TOE. The evaluator could find in these sections evidence of the calculation used to determine the FAR and FRR for the TOE. The evaluator verified that: the testing was performed offline; that the system used for testing is emulated on a different platform in a cloud computation infrastructure for efficiency reasons; a full cross-comparison for offline testing is used to compare imposters for Gen.1 sensors, and a partial full-cross comparison is used for Gen.2 and Gen.3 sensors.

Validation of Face ID follows the methodology established for Touch ID also using offline testing. The FAR was evaluated by full cross-comparison of all subjects in the datasets. The datasets contained fully labeled data. Data was collected from a wide range of subjects. Facial expression variations were collected from each subject as well as variations in environmental factors. Variations in subject age, ethnicity, and gender were also introduced into the dataset as well as subjects that exhibited familial relationships such as siblings. Offline testing was performed with data that simulates a normal presentation -- near frontal view, no obstructions, within nominal range (20-45 cm).

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

FIA_BMG_EXT.1.2

TSS Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.1.2-ASE-01

The evaluator shall verify that the TSS indicates which SAFAR the TOE is targeting and contains evidence supporting the calculations, per Appendix H.3, completed to determine the SAFAR. The evaluator shall verify that the TSS contains evidence of how the authentication factors interact, per FIA_UAU.5.2 and FIA_AFL_EXT.1. The evaluator shall verify that the TSS, contains the combination(s) of authentication factors needed to meet the SAFAR, and the number of attempts for each authentication factor the TOE is configured to allow. Adequate documentation is required to demonstrate the calculations completed to support the claimed SAFAR.

Summary

The evaluator reviewed section 8.5.1 of the [ST] which describes then biometric authentication supported by the TOE. The TOE supports fingerprint (Touch ID) or facial recognition authentication (Face ID). Sections 8.5.1.1 and 8.5.1.3 of the ST describes the *Accuracy of Biometric Authentication*.

Section 8.5 describes how the Face ID and Touch ID interact with the passcode set on the device. A device that supports Face ID does not support Touch ID and vice-versa. A passcode must be supplied in certain cases for additional security.

- The device has just been turned on or restarted.

- The device hasn't been unlocked for more than 48 hours.
- The passcode hasn't been used to unlock the device in the last 156 hours (six and a half days) and Face ID hasn't unlocked the device in the last 4 hours.
- The device has received a remote lock command.
- After five unsuccessful attempts to match.
- After initiating power off/Emergency SOS.

Section 8.5.1.3 describes that there can be at most only 5 attempts for the fingerprint and 5 attempts for the facial recognition. Passed these attempts, at most 10 attempts for the 6-digit passcode are provided.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.4.10 Extended: Biometric Enrollment (FIA_BMG_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.2-ASE-01

The evaluator shall verify that the TSS describes how the quality of samples used to create the authentication template at enrollment are verified. As well as the quality standard that the validation method uses to perform the assessment.

Summary

Section 8.5.1.4 of the TSS describes the biometric sample quality used on the TOE for deriving the authentication template enrolled. This includes the following.

- deciding what portion of the sensor is covered by the finger. Sensor regions containing very weak signal are considered not covered. Samples with high number of such regions are rejected
- assessing the level of residual fixed-pattern noise. Samples where the noise could significantly alter the fingerprint pattern are rejected
- detection and removal of regions affected by image discontinuity caused by finger motion. Samples with many regions affected by motion are rejected

Moreover, the testing mechanisms are repeated for every major iOS release. Section 8.5.1.1 describe the tests subjects chosen for the samples: various people were used to collect facial expressions with various eye wear, indoor/outdoor settings, age, ethnicity and gender. Family members were also chosen, such as siblings.

The method used to perform this assessment is an off-line test. Fingerprint data (Touch ID) were collected separately for each sensor. The data was passed to an emulator for efficiency reasons, and not on the production hardware. Note that a special testing step is performed in order to guarantee that the results on the emulator match the results on the production hardware. For Gen.1 sensors, a full-cross comparison scheme is used. For Gen.2 and Gen.3 sensors, a partial cross-comparison is used.

For Face ID:

- User is attending the device

- No significant depth holes in the depth map
- Anti-Spoofing network to reject physical spoofs
- For enrollment
 - No occlusions detected (e.g. hand covering face)
 - Face within certain pose angles
 - User attending device
 - No significant depth holes in the depth map
 - Anti-spoofing network to reject physical spoofs




The validation of the discussed mechanism is performed regularly for each major iOS release. The test is based on specialized datasets containing different levels of coverage and different artifacts. These samples are fed to the biometric system and it is confirmed whether the sample is correctly passed or rejected from the processing as expected. Additionally, the biometric system is tested by feeding artificially created images containing different geometric patterns.

Guidance Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.2-AGD-01

The evaluator shall verify that the AGD guidance describes how to enroll a user for each biometric modality supported.

Summary

[CCGUIDE]  section 3.5.3 *Biometric Authentication Factors* describes how to enroll biometric authentication factors which explicitly references [iPhone_UG]  and [iPad_UG] . This section also states that in the evaluated configuration, users cannot enable Touch ID or Face ID on their devices. This restriction must be performed by the administrator using a Configuration Profile. If enabled, the user can enroll Touch ID or Face ID as follows:





- Enrollment for Touch ID is typically accomplished during initial device configuration but can also be performed using the Settings>>Touch ID & Passcode menus. Multiple fingerprints may be enrolled, named, and deleted from this menu.
- Enrollment for Face ID is typically accomplished during initial device configuration but can also be performed using the Settings>>Face ID & Passcode menu by tapping the "Set up Face ID" option. Users can enroll an alternative appearance for Face ID, for a total of two enrollments per device. Face ID credential is established by providing biometric samples for enrollment.

Test Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.2-ATE-01

The evaluator shall input biometric samples for enrollment. Upon inputting biometric samples a fixed number of times as specified in the prompts, one or more authentication templates will be generated. The evaluator shall verify that the device only accepts samples of sufficient quality or requests additional samples if the authentication template is not of sufficient quality. For all quality metrics, the evaluator shall ensure that biometric samples achieving a worse quality score than the prescribed threshold are rejected.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

The evaluator enrolled a fingerprint and a face into the device and attempted to unlock the device with another finger, and the nose and mouth covered and verified that it failed. The evaluator then unlocked the device with the correct finger or face and verified that it succeeds.

2.1.4.11 Extended: Biometric Verification (FIA_BMG_EXT.3)

TSS Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.3-ASE-01

The evaluator shall verify that the TSS describes how the quality of samples used to verify authentication are verified. As well as the quality standard that the validation method uses to perform the assessment.

Summary

Section 8.4.1.4 of the TSS describes the biometric sample quality used on the TOE for deriving the authentication template enrolled. This includes:

For Touch ID:

- deciding what portion of the sensor is covered by the finger. Sensor regions containing very weak signal are considered not covered. Samples with high number of such regions are rejected
- assessing the level of residual fixed-pattern noise. Samples where the noise could significantly alter the fingerprint pattern are rejected
- detection and removal of regions affected by image discontinuity caused by finger motion. Samples with many regions affected by motion are rejected

For Face ID

- User is attending the device
- No significant depth holes in the depth map
- Anti-Spoofing network to reject physical spoofs
- For enrollment
 - No occlusions detected (e.g. hand covering face)
 - Face within certain pose angles
 - User attending device
 - No significant depth holes in the depth map
 - Anti-spoofing network to reject physical spoofs

Moreover, the TSS describe that:

"The test is based specialized datasets containing different levels of coverage and different artifacts. These samples are fed to the biometric system and it is confirmed whether the sample is correctly passed or rejected from the processing as expected. Additionally, the biometric system is tested by feeding artificially created images containing different geometric patterns."

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.3-ATE-01

The evaluator shall enroll a user for each biometric modality supported. The evaluator will then input biometric samples for verification and ensure that the device only accepts samples of sufficient quality. The evaluator shall ensure that biometric samples achieving a worse quality score than the prescribed threshold are rejected.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

FaceID: the evaluator registered a face that is partially covered and verified that the TOE does not accept the registration. The evaluator then registered a normal face and verified that the face is used.

TouchID: the evaluator attempted to enroll a finger partially covering the sensor and verified that the TOE refused to enroll this finger. The evaluator then enrolled a normal finger and verified that it succeeds. The evaluator then attempted to unlock the device 5 times with an incorrect finger and verified that it did not succeed.

2.1.4.12 Extended: Handling Unusual Biometric Templates (FIA_BMG_EXT.5)

TSS Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.5-ASE-01

The evaluator shall verify that the TSS how the matching algorithm addresses properly formatted templates with unusual data properties, incorrect syntax, or low quality.

Summary

See work unit AA-FIA_BMG_EXT.3-ASE-01 .

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_BMG_EXT.5-ATE-01

[The evaluator shall verify that the TSS how the matching algorithm addresses properly formatted templates with unusual data properties, incorrect syntax, or low quality.] The evaluator shall ensure that these claims are sound through appropriate testing based on test programs provided by the vendor.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

No testing required as defined in the PP for the current SFR.

2.1.4.13 Re-Authentication (FIA_UAU.6)

FIA_UAU.6.1

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_UAU.6.1-1-ATE-01

- **Test 1:** The evaluator shall configure the TSF to use the Password Authentication Factor according to the AGD guidance. The evaluator shall change Password Authentication Factor according to the AGD guidance and verify that the TSF requires the entry of the Password Authentication Factor before allowing the factor to be changed.
- **Test 2:** [conditional] For each BAF selected in FIA_UAU.5.1, the evaluator shall configure the TSF to use the BAF, which includes configuring the Password Authentication Factor, according to the AGD guidance. The evaluator shall change the BAF according to the AGD guidance and verify that the TSF requires the entry of the Password Authentication Factor before allowing the BAF to be changed.
- **Test 3:** [conditional] If "hybrid" is selected in FIA_UAU.5.1, the evaluator shall configure the TSF to use the BAF and PIN or password, which includes configuring the Password Authentication Factor, according to the AGD guidance. The evaluator shall change the BAF and PIN according to the AGD guidance and verify that the TSF requires the entry of the Password Authentication Factor before allowing the factor to be changed.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator changed the passcode on the TOE by going to Password at Settings -> TouchID -> Passcode -> Change Passcode and verified that the old password is required

Test 2: the evaluator tried to change to change the fingerprint (TouchID)/FaceID on the device and verified that the TOE prompted for a password.

Test 3: not applicable.

FIA_UAU.6.2

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_UAU.6.1-2-ATE-01

- **Test 1:** The evaluator shall configure the TSF to transition to the locked state after a time of inactivity (FMT_SMF_EXT.1) according to the AGD guidance. The evaluator shall wait until the TSF locks and then verify that the TSF requires the entry of the Password Authentication Factor before transitioning to the unlocked state.
- **Test 2:** [conditional] For each BAF selected in FIA_UAU.5.1, the evaluator shall repeat Test 1 verifying that the TSF requires the entry of the BAF before transitioning to the unlocked state.
- **Test 3:** [conditional] If “hybrid” is selected in FIA_UAU.5.1, the evaluator shall repeat Test 1 verifying that the TSF requires the entry of the BAF and PIN/password before transitioning to the unlocked state.
- **Test 4:** The evaluator shall configure user-initiated locking according to the AGD guidance. The evaluator shall lock the TSF and then verify that the TSF requires the entry of the Password Authentication Factor before transitioning to the unlocked state.
- **Test 5:** [conditional] For each BAF selected in FIA_UAU.5.1, the evaluator shall repeat Test 4 verifying that the TSF requires the entry of the BAF before transitioning to the unlocked state.
- **Test 6:** [conditional] If “hybrid” is selected in FIA_UAU.5.1, the evaluator shall repeat Test 4 verifying that the TSF requires the entry of the BAF and PIN/password before transitioning to the unlocked state.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator made sure that the lock out period is not set to never and waited for the device to auto-lock. The evaluator then authenticated to the device.

Test 2: the evaluator re-performed Test 1 with TouchID (fingerprint) and then FaceID (face).

Test 3: not applicable.

Test 4: the evaluator locked the device manually (power button) and authenticated to the TOE with the password as described in Test 1.

Test 5: the evaluator performed Test 4 but unlocked the device with fingerprint (TouchID) or FaceID.

Test 6: not applicable.

2.1.4.14 Protected Authentication Feedback (FIA_UAU.7)

TSS Assurance Activities

Assurance Activity AA-FIA_UAU.7-ASE-01

The evaluator shall ensure that the TSS describes the means of obscuring the authentication entry, for all authentication methods specified in FIA_UAU.5.1.

Summary

Section 8.5 in the [ST] describes the *Identification and Authentication (FIA)*.

The TSS describes that each character of the passcode, when entered, is replaced by a dot symbol. Moreover, section 8.5 states that biometric authentication inputs do not produce feedback to the user unless an input is rejected. When an invalid fingerprint sample is given or cannot be authenticated, a simple error message is returned to the user to try again. When an invalid facial sample is given or cannot be authenticated, the device will vibrate.

Guidance Assurance Activities

Assurance Activity AA-FIA_UAU.7-AGD-01

The evaluator shall verify that any configuration of this requirement is addressed in the AGD guidance and that the password is obscured by default.

Summary

[CCGUIDE] [section 3.5.2 Protected Authentication Feedback Configuration](#) provides related guidance on protected authentication feedback. It states the following:

- All passcode entries are obscured by iOS. This is done by displaying a dot symbol in place of each character as the passcode entry user input occurs. No configuration of this feature is required from the mobile device administrator.
- Biometric authentication inputs do not provide feedback to the user unless the input is rejected. Additionally, biometric authentication inputs do not relay authentication entry information and are inherently obscured. When an invalid fingerprint sample is given or a fingerprint sample cannot be authenticated, a simple error message is returned which prompts the user to try again. When an invalid facial sample is given or a facial sample cannot be authenticated, the mobile device will vibrate. If three invalid biometric samples are presented the mobile device will offer passcode entry. After five invalid biometric samples are presented passcode authentication is required.

Test Assurance Activities

Assurance Activity AA-FIA_UAU.7-ATE-01

- **Test 1:** *The evaluator shall enter passwords on the device, including at least the Password Authentication Factor at lockscreen, and verify that the password is not displayed on the device.*
- **Test 2:** *[conditional] For each BAF selected in FIA_UAU.5.1, the evaluator shall authenticate by producing a biometric sample at lockscreen. As the biometric algorithms are performed, the evaluator shall verify that sensitive images, audio, or other information identifying the user are kept secret and are not revealed to the user. Additionally, the evaluator shall produce a biometric sample that fails to authenticate and verify that the reason(s) for authentication failure (user mismatch, low sample quality, etc.) are not revealed to the user. It is acceptable for the BAF to state that it was unable to physically read the biometric sample, for example, if the sensor is unclean or the biometric sample was removed too quickly. However, specifics regarding why the presented biometric sample failed authentication shall not be revealed to the user.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[4\]](#).

Test 1: the evaluator locked the device, and verified that when unlocking, password characters are not displayed, and dots are displayed instead.

Test 2: the evaluator locked and unlocked the device using both TouchID and FaceID and verified that no information pertaining either the fingerprint or the face is displayed.


2.1.4.15 Extended: Authentication for Cryptographic Operation (FIA_UAU_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FIA_UAU_EXT.1-ASE-01

The evaluator shall verify that the TSS section of the ST describes the process for decrypting protected data and keys. The evaluator shall ensure that this process requires the user to enter a Password Authentication Factor and, in accordance with FCS_CKM_EXT.3, derives a KEK, which is used to protect the software-based secure key storage and (optionally) DEK(s) for sensitive data, in accordance with FCS_STG_EXT.2.

Summary

Section 8.3.1 of the [ST]  describes the *Overview of Key Management*.

Section 8.3.1 describes that the passcode is used, combined with the UID and the salt, to derive the passcode key (KEK) which is then used to decrypt the class keys for protected data. That passcode key is erased whenever the device is locked, and reconstructed whenever the device is unlocked with the right passcode.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities





Assurance Activity AA-FIA_UAU_EXT.1-ATE-01

The following tests may be performed in conjunction with FDP_DAR_EXT.1 and FDP_DAR_EXT.2.

Assurance Activity Note: *The following test require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.*

- **Test 1:** *The evaluator shall enable encryption of protected data and require user authentication according to the AGD guidance. The evaluator shall write, or the developer shall provide access to, an application that includes a unique string treated as protected data. The evaluator shall reboot the device, use a tool provided by developer to search for the unique string amongst the application data, and verify that the unique string cannot be found. The evaluator shall enter the Password Authentication Factor to access full device functionality, use a tool provided by the developer to access the unique string amongst the application data, and verify that the unique string can be found.*
- **Test 2:** *[conditional] The evaluator shall require user authentication according to the AGD guidance. The evaluator shall store a key in the software-based secure key storage. The evaluator shall lock the device, use a tool provided by developer to access the key amongst the stored data, and verify that the key cannot be retrieved or accessed. The evaluator shall enter the Password Authentication Factor to access full device functionality, use a tool provided by developer to access the key, and verify that the key can be retrieved or accessed.*
- **Test 3:** *[conditional] The evaluator shall enable encryption of sensitive data and require user authentication according to the AGD guidance. The evaluator shall write, or the developer shall provide access to, an application that includes a unique string treated as sensitive data. The evaluator shall lock the device, use a tool provided by developer to attempt to access the unique string amongst the application data, and verify that the unique string cannot be found. The evaluator shall enter the Password Authentication Factor to access full device functionality, use a tool provided by developer to access the unique string amongst the application data, and verify that the unique string can be retrieved.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

Test 1: The suggested testing cannot be conducted on any Apple device due to the approach for encryption data as documented for FDP_DAR_EXT.1 is implemented. The evaluator verified those steps in the source code of the TOE. This approach has been approved by NIAP during the previous iOS 11 evaluation.

Test 2: see FCS_CKM_EXT.4. This approach has been approved by NIAP during the previous iOS 11 evaluation.

Test 3: see FCS_CKM_EXT.4. This approach has been approved by NIAP during the previous iOS 11 evaluation.


2.1.4.16 Extended: Timing of Authentication (FIA_UAU_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FIA_UAU_EXT.2-ASE-01

The evaluator shall verify that the TSS describes the actions allowed by unauthorized users in the locked state.

Summary

Section 8.5 of the [ST]  describes the *Identification and Authentication (FIA)* .

The first paragraph of section 8.5 explains that except for making emergency calls, answering calls, using the camera and the flashlight, users need to authenticate to the device to perform any services.

Guidance Assurance Activities





No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_UAU_EXT.2-ATE-01

The evaluator shall attempt to perform some actions not listed in the selection while the device is in the locked state and verify that those actions do not succeed.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

Test 1: the evaluator waited for the device to lock automatically (or locked the device manually), and verified that the screen is overwritten by non-sensitive data. The evaluator verified that only the camera, information center, display of information, and control center can be used.

2.1.4.17 Extended: Validation of Certificates (FIA_X509_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FIA_X509_EXT.1-ASE-01

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.

Summary

Section 8.5.2 of the [ST]  describes *Certificates* .

The evaluator reviewed section 8.5.2 and verified that the verification of certificates takes place whenever the TOE tries to establish a connection using a remote certificate. The process for certificate validation is as follows.

"When attempting to establish a connection using a remote certificate, the certificate is first checked to ensure it is valid. Certificates are validated against the common name (CN) portion of the distinguished name (DN). If the CN does not match the corresponding domain name system (DNS) or IP Address of the server being accessed, validation and subsequently the connection will fail. If the certificate is valid, the attempt to establish the connection continues. If the certificate is invalid, the next step is up to the application. The application should provide an indication to the user that the certificate is invalid and options to accept or reject."

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.1-ATE-01

The tests described must be performed in conjunction with the other Certificate Services assurance activities, including the use cases in FIA_X509_EXT.2.1 and FIA_X509_EXT.3. 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 1:** *The evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function (e.g. application validation, trusted channel setup, or trusted software update), and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.*
- **Test 2:** *The evaluator shall demonstrate that validating an expired certificate results in the function failing.*
- **Test 3:** *The evaluator shall test that the TOE can properly handle revoked certificates-conditional on whether CRL or OCSP is selected; if both 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). For the test of the WLAN use case, only pre-stored CRLs are used. 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.*
- **Test 4:** *The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.*
- **Test 5:** *The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension not set. The validation of the certificate path fails.*
- **Test 6:** *The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.*
- **Test 7:** *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).*
- **Test 8:** *The evaluator shall modify any bit in the last byte of the signature algorithm of the certificate and demonstrate that the certificate fails to validate (the signature on the certificate will not validate).*
- **Test 9:** *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).*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: Test covered by tests performed for FCS_TLSC_EXT.1.1(MDFPP) and FCS_TLSC_EXT.1.2(MDFPP): valid certificates signed by a CA known to the TOE are trusted by the TOE. Valid certificates loaded onto the TOE are signed by a CA unknown to the TOE are treated as untrusted by the TOE.

Test 2: the evaluator generated a certificate for the server, advanced the date to one day to mark the certificate as expired, connected the TOE to https://server and verified that the connection is rejected.

Test 3: the evaluator cleared the Safari cache settings, connected the TOE to the network, started Wireshark, and access the following URLs with Extended Validation (EV) certificates. The evaluator verified that the OCSP requests were found in the pcap files.

Test 4: the evaluator generated a server CA which does not contain the basicConstraints extension, connected the TOE to https://server and verified that Safari either rejects the communication or prompts the user to either accept or reject the certificate.

Test 5: the evaluator generated a server certificate with the basicConstraints extension, but without the CA flag set, connected the TOE to https://server and verified that Safari either rejects the communication or prompts the user to either accept or reject the certificate.

Test 6: the evaluator generated a server certificate with the basicConstraints extension, with the CA flag set to TRUE, connected the TOE to https://server and verified that the connection succeeded and that the TLS information were displayed on the webpage.

Test 7: the evaluator created a new profile and added the CA certificate to it, opened the XML profile file and modified that the first character of the certificate within it. The evaluator then loaded the profile onto the device with the Apple Configurator 2 and verified that the operation failed because the certificate is broken.

Test 8: the evaluator performed the same as Test 7, but instead modified the last character of the last line of the certificate.

Test 9: the evaluator performed the same as Test 7, but instead modified the first character of the sixth line of the certificate.

2.1.4.18 Extended: X509 Certificate Authentication (FIA_X509_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-ASE-01

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.

Summary

Section 8.5.2 of the [ST] [\[1\]](#) describes the *Certificates* .

Section 8.5.2 describes that every certificate in iOS is assigned a certificate type respective to the area in which this certificate is going to be used (such as AppleX509Basic or AppleSSL for example), as described in the [CCGUIDE] [\[1\]](#) .

Assurance Activity AA-FIA_X509_EXT.2-ASE-02

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. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

Summary

Section 8.5.2 of the [ST] describes the *Certificates* .

Whenever a certificate cannot be validated, it is up to the application requesting a connection be established to decide whether to abort the connection or not.

Guidance Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-AGD-01

If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

Summary

According to [ST], the default setting is for the TOE to not accept untrusted certificates.

[CCGUIDE] section 3.5.6 *X.509 Certificate Configuration* provides related guidance for X.509 certificate validation which states that to configure the devices to reject untrusted certificates, the administrator can use the *TLSAllowTrustExceptions* key in the Wi-Fi Payload of the Configuration Profile to ensure that untrusted certificates are rejected and authentication fails by the TOE if such untrusted certificates are presented.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-ATE-01

The evaluator shall perform the following test for each trusted channel:

- **Test 1:** *The evaluator shall demonstrate that using a valid certificate that requires 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.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator re-performed FCS_TLSC_EXT.1.1 to show the correct operation of the TOE connecting to a non-TOE entity. The evaluator created a certificate without the "Server Authentication" field and verified that the connection failed when trying to connect the TOE to the test laptop. The evaluator then configured the server to use the TLS_RSA_WITH_AES_128_CBC_SHA while the TOE uses an ECDSA certificate, and verified that the connection fails.

Note that the evaluator performed that test both on Wi-Fi and cellular networks.

2.1.4.19 Extended: Request Validation of certificates (FIA_X509_EXT.3)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FIA_X509_EXT.3-AGD-01

The evaluator shall verify that the API documentation provided according to Section 5.2.2 includes the security function (certificate validation) described in this requirement. This documentation shall be clear as to which results indicate success and failure.

Summary

[CCGUIDE] [\[1\]](#) section 3.5.6 X.509 Certificate Configuration references "Certificate, Key, and Trust Services Reference" [CKTSREF] [\[1\]](#) as the API documentation related to certificate validation. This is covered by the function 'SetTrustEvaluate'.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.3-ATE-01

The evaluator shall write, or the developer shall provide access to, an application that requests certificate validation by the TSF. The evaluator shall verify that the results from the validation match the expected results according to the API documentation. This application may be used to verify that import, removal, modification, and validation are performed correctly according to the tests required by FDP_STG_EXT.1, FTP_ITC_EXT.1, FMT_SMF_EXT.1.1, and FIA_X509_EXT.1.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[1\]](#).

The hostap EAP-TLS server is configured to accept only one cipher. OpenSSL supports all selected cipher modes. For each configured cipher, the evaluator accessed to the web server must be performed which must be successful. The evaluator generated a full CA hierarchy, and connected the TOE for every cipher suites supported to http://server and https://server. The evaluator verified that the web connection is successful, and in the case of TLS, that the web page displays the TLS information.

2.1.4.20 Extended: Enrollment of Mobile Device into Management (FIA_ENR_EXT.2(AGENT))

TSS Assurance Activities

Assurance Activity AA-FIA_ENR_EXT.2-MDMA-ASE-01

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).

Summary

Section 8.5.3 of the [ST]  specifies the *MDM Server Reference ID* .

Table 10: MDM Server Reference Identifiers in section 8.5.3 specifies the reference identifiers used by the MDM server. They can be the following.

- `serve_name`: An identifiable name for the MDM Server
- `server_uuid`: A system-generated server identifier
- `admin_id`: Apple ID of the person who generated the current tokens that are in use
- `facilitator_id`: Legacy equivalent to the `admin_id` key. This key is deprecated and may not be returned in future responses
- `org_name`: The organization name
- `org_email`: The organization email address
- `org_phone`: The organization phone
- `org_address`: The organization address

The evaluator verified which type of reference identifiers are acceptable to the TOE, and they are configured by the MDM enrollment service through JavaScript Object Notation (JSON) tags on the mobile device.

Guidance Assurance Activities

Assurance Activity AA-FIA_ENR_EXT.2-MDMA-AGD-01

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

Summary

[CCGUIDE]  section 2.3.3 *Configure MDM Agent and MDM Communications* describes how to configure MDM Agent and MDM Server communications as follows:

- MDM Agent-Server communication is achieved securely using the MDM protocol which is built on top of HTTP, TLS, and push notifications that use HTTP PUT over TLS (SSL). A managed mobile device uses an identity to authenticate itself to the MDM server over TLS (SSL). This identity can be included in the profile as a Certificate payload, or can be generated by enrolling the device with Simple Certificate Enrollment Protocol (SCEP).
- The MDM Agent communications uses the iOS Security Framework as described in section 3.3.2 *TLS Configuration*. Thus, configuring the TOE's TLS protocol as per section 3.3.2 automatically configures the MDM Agent communications. If an additional CA certificate needs to be added to support the MDM Server, see section 3.3.2.3. Certificate Authority (CA) Configuration.

Test Assurance Activities

Assurance Activity AA-FIA_ENR_EXT.2-MDMA-ATE-01

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 Assurance Activities verify that the MDM Agent can connect to the MDM Server and validate the MDM Server's certificate.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator went on the TOE's Settings -> General -> Device Management -> Remote Management -> More Details, and could verify that the Device Identity Certificate contains the issuers certificate. In Device Management, the evaluator verified that the URL indeed points to the OSX server at osxserver.atsec.com.

2.1.4.21 Port Access Entity Authentication (FIA_PAE_EXT.1(WLAN))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_PAE_EXT.1-WLAN-ATE-01

The evaluator shall perform the following tests:

- *Test 1: The evaluator shall demonstrate that the TOE has no access to the test network. After successfully authenticating with an authentication server through a wire less access system, the evaluator shall demonstrate that the TOE does have access to the test network.*
- *Test 2: The evaluator shall demonstrate that the TOE has no access to the test network. The evaluator shall attempt to authenticate using an invalid client certificate, such that the EAP-TLS negotiation fails. This should result in the TOE still being unable to access the test network.*
- *Test 3: The evaluator shall demonstrate that the TOE has no access to the test network. The evaluator shall attempt to authenticate using an invalid authentication server certificate, such that the EAP-TLS negotiation fails. This should result in the TOE still being unable to access the test network.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator removed the profile from the TOE containing the WiFi information and verified that the device could not connect to the network. The evaluator then re-added the profile to the TOE and verified that the device could connect to the network by witnessing the EAPOL 4-way handshake using Wireshark.

Test 2: the evaluator removed the profile from the TOE containing the WiFi information and verified that the device could not connect to the network. The evaluator then created a similar profile but with a wrong certificate, loaded it on the TOE, and verified that the TOE could not connect to the network.

Test 3: the evaluator removed the profile from the TOE containing the WiFi information and verified that the device could not connect to the network. The evaluator created an incorrect certificate for the server and verified that the TOE could not connect to the network.

2.1.4.22 X.509 Certificate Authentication (EAP-TLS) (FIA_X509_EXT.2(WLAN))

TSS Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-WLAN-ASE-01

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.

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. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

Summary

Section 8.9.1, *EAP-TLS and TLS*, in the [ST] describes the EAP-TLS and TLS protocols and ciphersuites supported by the TOE.

Section 8.9.1 describes that when the TOE is configured to use EAP-TLS, the CA certificate(s) to which the server's certificate must chain can be configured using the PayloadCertificateAnchorUUID key in the Wi-Fi payload of the configuration profile. When the TLSAllowTrustExceptions key in the Wi-Fi payload is used, the administrator can enforce that untrusted certificates are not accepted and the authentication fails if such an untrusted certificate is presented. Please refer to the [CCGUIDE] for a description of the default action that the administrator is able to specify.

Guidance Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-WLAN-AGD-01

The evaluator shall check the administrative guidance to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions for configuring the operating environment so that the TOE can use the certificates.

Summary

[CCGUIDE] section 3.5.6 *X.509 Certificate Configuration* states the following:

- X.509 certificates are configured by an administrator using a Configuration Profile as described in [IOS_CFG].
- Certificates have a certificate type that defines their respective application area thus ensuring that only certificates defined for a specific application area are used. In addition, the database containing trust anchors for all certificates is protected via integrity check and write protection. The certificate types supported by the TOE are:
 - AppleX509Basic
 - AppleSSL
 - AppleSMIME
 - AppleEAP
 - AppleIPsec
 - AppleCodeSigning
 - AppleIDValidation
 - AppleTimeStamping

- Per [TRUST_STORE][1](#), the TOE has an iOS Trust Store that contains trusted root certificates presintalled with iOS.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-WLAN-ATE-01

The evaluator shall perform the following test for each trusted channel:

Test: The evaluator shall demonstrate using a valid certificate that requires 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][1](#), the extended package [EP_MDM_AGENT_V3.0][1](#), the extended package [PP_WLAN_CLI_EP_V1.0][1](#), and the PP module [MOD_VPN_CLI_V2.1][1](#).

Test 1: the evaluator configured the EAP server to only support one cipher, and OpenSSL supports all cipher modes. For each configured cipher, the evaluator connected the TOE to the server successfully.

Test 2: the evaluator then created a server certificate without the "Server Authentication" value in the extendedKeyUsage field and verified that the TOE could not connect successfully to the server.

Test 3: the evaluator used an ECDSA certificate while connecting to the server only supporting the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite, and verified that the TOE could not connect successfully to the server.

2.1.5 Security management (FMT)

2.1.5.1 Extended: Management of Security Functions Behavior (FMT_MOF_EXT.1)

FMT_MOF_EXT.1.1

TSS Assurance Activities


Assurance Activity AA-FMT_MOF_EXT.1.1-ASE-01

The evaluator shall verify that the TSS describes those management functions that may only be performed by the user and confirm that the TSS does not include an Administrator API for any of these management functions. This activity will be performed in conjunction with FMT_SMF_EXT.1.

Summary

Section 8.6.2 in the [ST][1](#) describes the *Configuration Profiles* .

The evaluator verified the functions 4, 5, 12, 13, 18 are only allowed by the user and not the administrator. The last paragraph of section 8.6.2 explains that configuration profiles can be deployed such that users are unable to override or remove restrictions set by the administrators or MDM

administrators. Depending on the behavior defined in the configuration profile, users might be unable to perform or access management functions defined in *Table 4: Management Functions* of the [ST] .

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

FMT_MOF_EXT.1.2


TSS Assurance Activities

Assurance Activity AA-FMT_MOF_EXT.1.2-ASE-01

The evaluator shall verify that the TSS describes those management functions that may be performed by the Administrator, to include how the user is prevented from accessing, performing, or relaxing the function (if applicable), and how applications/APIs are prevented from modifying the Administrator configuration. The TSS also describes any functionality that is affected by administrator-configured policy and how. This activity will be performed in conjunction with FMT_SMF_EXT.1.

Summary

Section 8.6 of the [ST]  describes the *Specification of Management Functions (FMT)* .

Administrator configuration is specified via the configuration profiles deployed with the Apple Configurator tool, an email message, a web page, over-the-air configuration (As described in [OTA_CFG] ), or over-the-air using the MDM server. All settings or restrictions defined in these profiles restrict any options the user has. If the profiles are protected against removal by the administrator, the user cannot circumvent the restrictions or settings by removing the respective profile.

Guidance Assurance Activities





No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FMT_MOF_EXT.1.2-ATE-01

- **Test 1:** *The evaluator shall use the test environment to deploy policies to Mobile Devices.*
- **Test 2:** *The evaluator shall create policies which collectively include all management functions which are controlled by the (enterprise) administrator and cannot be overridden/relaxed by the user as defined in FMT_MOF_EXT.1.2. The evaluator shall apply these policies to devices, attempt to override/relax each setting both as the user (if a setting is available) and as an application (if an API is available), and ensure that the TSF does not permit it. Note that the user may still apply a more restrictive policy than that of the administrator.*
- **Test 3:** *Additional testing of functions provided to the administrator are performed in conjunction with the testing activities for FMT_SMF_EXT.1.1.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

Test 1: See all tests before: they use Apple Configurator 2 to deploy policies.

Test 2: All configuration options are tested with FMT_SMF_EXT.1. For administrator-related configuration options, Apple Configurator 2 is used.

Test 3: See FMT_SMF_EXT.1

2.1.5.2 Extended: Specification of Management Functions (FMT_SMF_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.1-ASE-01

The evaluator shall verify that the TSS describes all management functions, what role(s) can perform each function, and how these functions are (or can be) restricted to the roles identified by FMT_MOF_EXT.1.

The following activities are organized according to the function number in the table. These activities include TSS assurance activities, AGD assurance activities, and test activities.

Summary

Section 8.6 of the [ST] describes the *Specification of Management Functions (FMT)* .

Section 8.6 references *Table 4: Management Functions* of the [ST] , which describes which roles can perform which functions. The last paragraph of section 8.6.2 explains that configuration profiles can be deployed such that users are unable to override or remove restrictions set by the administrators or MDM administrators. Depending on the behavior defined in the configuration profile, users might be unable to perform, or access, management functions defined in *Table 4: Management Functions* of the [ST] .

Assurance Activity AA-FMT_SMF_EXT.1-ASE-02

Function 1

The evaluator shall verify the TSS defines the allowable policy options: the range of values for both password length and lifetime, and a description of complexity to include character set and complexity policies (e.g., configuration and enforcement of number of uppercase, lowercase, and special characters per password).

Summary

Section 8.5 of the [ST] describes the *Identification and Authentication (FIA)* .

Section 8.5 describes that the length, complexity and lifetime of the password is configurable by the user authenticated to the device, or can also be set using a configuration profile if the device is under MDM.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-03

Function 2

The evaluator shall verify the TSS defines the range of values for both timeout period and number of authentication failures for all supported authentication mechanisms.

Summary

Section 8.5 in the [ST] describes the *Identification and Authentication* .

Section 8.5 describes that the maximum number of consecutive failed attempts to enter the right passcode can be set to, between 2 and 10, again, by either the user or through a configuration profile. This section also describes that the number of minutes for which the device can be idle before it locks itself automatically can be defined by the user or a configuration profile.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-04

Function 4

The evaluator shall verify that the TSS includes a description of each radio and an indication of if the radio can be enabled/disabled along with what role can do so. In addition the evaluator shall verify that the frequency ranges at which each radio operates is included in the TSS.

Summary

Section 8.8.2 of the [ST]  describes the *Restricting Access to Wireless Networks* .

Section 8.8.2 describes how users and administrators can restrict the wireless networks accessible by the TOE. This section references *Table 1: Devices Covered by the Evaluation* . The standards defined in these tables list the frequency ranges at which each radio operates.


Assurance Activity AA-FMT_SMF_EXT.1-ASE-05

Function 5

The evaluator shall verify that the TSS includes a description of each collection device and an indication of if it can be enabled/disabled along with what role can do so.

Summary

Section 8.6 of the [ST]  describes the *Specification of Management Functions (FMT)* .

Section 8.6 references *Table 4: Management Functions* of the [ST]  which describes that the cameras can be enabled/disabled on a per-app basis by the user the administrator, and that the microphone can be enabled/disabled on a per-app basis by the user only.


Assurance Activity AA-FMT_SMF_EXT.1-ASE-06

Function 8

The evaluator shall verify the TSS describes the allowable application installation policy options based on the selection included in the ST. If the application whitelist is selected, the evaluator shall verify that the TSS includes a description of each application characteristic upon which the whitelist may be based.

Summary

Section 8.6 of the [ST]  describes the *Specification of Management Functions (FMT)* .

Section 8.6 references table 5 which defines that the administrator can set iOS to disallow any application from being installed. The application white-list option has not been selected in the [ST]  .

Assurance Activity AA-FMT_SMF_EXT.1-ASE-07

Function 9 & Function 10

The evaluator shall verify that the TSS describes each category of keys/secrets that can be imported into the TSF's secure key storage.

Summary

Section 8.3.1 in the [ST][1](#) describes the *Overview of Key Management* . Section 8.3.2 in the [ST][1](#) describes the *Storage of Persistent Secrets and Private Keys by the Agent* .

Section 8.3.2 defines which keys/secrets can be imported and stored in the device, their purpose and how they are stored.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-08

Function 12

The evaluator shall verify that the TSS describes each additional category of X.509 certificates and their use within the TSF.

Summary

Section 8.9.4.4 in the [ST][1](#) describes the *Peer authentication*.

The aforementioned section specifies that X.509v3 certificates are being used for peer authentication, which is consistent with the selections made in the SFRs by the ST author.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-09

Function 13

The evaluator shall examine the TSS to ensure that it contains a description of each management function that will be enforced by the enterprise once the device is enrolled.

Summary

Section 8.6.1 of the [ST][1](#) describes the *Enrollment* management function. Section 8.6.2 of the [ST][1](#) describes the *Configuration Profiles*.

Function 13 only selects one management function: Enroll the TOE in management. This is described in section 8.6.1 of the [ST][1](#) . All the management functions that will be enforced by the enterprise once the device is enrolled are described in section 8.6.2. These functions are enforced by the configuration profile installed on the device.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-10

Function 14

The evaluator shall verify that the TSS includes an indication of what applications (e.g., user-installed applications, Administrator-installed applications, or Enterprise applications) can be removed along with what role can do so.

Summary

Section 8.6.2 in the [ST][1](#) describes the *Configuration Profiles* .

Section 8.6.2 explains in the last paragraph that configuration profiles can be defined to enable users to override or remove restrictions set in place by an administrator (or MDM administrator). This includes installation or removal of application of the TOE.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-12

Function 18

The evaluator shall ensure that the TSS includes a description of the Bluetooth profiles and services supported and the Bluetooth security modes and levels supported by the TOE. If function e is selected, the evaluator shall verify that the TSS describes any additional wireless technologies that may be used with Bluetooth, including Wi-Fi with Bluetooth High Speed and NFC as an Out of Band pairing mechanism. If function h is selected, the evaluator shall verify that all supported Bluetooth services are listed in the TSS as manageable and, if the TOE allows disabling by application rather than by service name, that a list of services for each application is also listed. If function i is selected, the evaluator shall verify that the TSS describes the method by which the level of security for pairings are managed, including whether the setting is performed for each pairing or is a global setting. If function j is selected, the evaluator shall verify that the TSS describes when Out of Band pairing methods are allowed and which ones are configurable.

Summary

Section 8.9.2 in the [ST]  describes the *Bluetooth* protocols.

In function 18, only function i is selected. Section 8.9.2 describes the Bluetooth profile security modes and levels supported by the TOE. The TOE supports Bluetooth 4.0, 4.2 and 5.0. All Bluetooth connections are encrypted using AES-CTR-128 with CCM mode.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-13

Function 23

- **Test 23:** The evaluator shall verify that the TSS states if the TOE supports a BAF and/or hybrid authentication. If the TOE does not include a BAF and/or hybrid authentication this test is implicitly met.
 - a. [conditional] If a BAF is selected the evaluator shall verify that the TSS describes the procedure to enable/disable the BAF. If the TOE includes multiple BAFs, the evaluator shall verify that the TSS describes how to enable/disable each BAF, specifically if the different modalities can be individually enabled/disabled.
 - b. [conditional] If “Hybrid” is selected the evaluator shall verify that the TSS describes the procedure to enable/disable the hybrid (biometric credential and PIN/password) authentication.

Assurance Activity Note: It should be noted that the following functions are optional capabilities, if the function is implemented, then the following assurance activities shall be performed. The notation of “[conditional]” beside the function number indicates that if the function is not included in the ST, then there is no expectation that the assurance activity be performed.

Summary

Section 8.6.3 of the ST describes the *Biometric Authentication Factors (BAFs)* .

The TOE can support either Touch ID or Face ID as a BAF, but not both. The BAF can be set up in the security options in the TOE settings (Touch ID & Passcode menu OR Face ID & Passcode menu, depending on the device), to enable/disable Touch ID/Face ID and/or the passcode.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-14

Function 24 [conditional]

The evaluator shall verify that the TSS includes a list of each externally accessible hardware port and an indication of if data transfer over that port can be enabled/disabled.

Summary

Function 24 is not selected in the [ST]  , (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-15

Function 25 [conditional]

The evaluator shall verify that the TSS describes how the TSF acts as a server in each of the protocols listed in the ST, and the reason for acting as a server.

Summary

Function 25 is not selected in the [ST] , (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-16

Function 29 [conditional]

The evaluator shall verify that the TSS describes how approval for an application to perform the selected action (import, removal) with respect to certificates in the Trust Anchor Database is accomplished (e.g., a pop-up, policy setting, etc.).

Summary

Function 29 is not selected in the [ST] , (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-17

Function 31 [conditional]

The evaluator shall ensure that the TSS describes which cellular protocols can be disabled.

Summary

Function 31 is not selected in the [ST] , (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-18

Function 34 [conditional]

The evaluator shall verify that the TSS describes how the approval for exceptions for shared use of keys/secrets by multiple applications is accomplished (e.g., a pop-up, policy setting, etc.).

Summary

Function 34 is not selected in the [ST] , (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-19

Function 35 [conditional]

The evaluator shall verify that the TSS describes how the approval for exceptions for destruction of keys/secrets by applications that did not import the key/secret is accomplished (e.g., a pop-up, policy setting, etc.).

Summary

Function 35 is not selected in the [ST] , (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-20

Function 36 [conditional]

The evaluator shall verify that the TSS describes any restrictions in banner settings (e.g., character limitations)

Summary

Section 8.8.3 in the [ST][1](#) describes the *Lock Screen / Access Banner Display*.

Section 8.8.3 describes how the banner can be set by displaying an image only when the screen is locked. Since the banner is an image there are no character limitations, information is restricted to what can be included in an image appropriate to the device display.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-21

Function 39 [conditional]

The evaluator shall verify that the TSS includes a description of how data transfers can be managed over USB.

Summary

Function 39 is not selected in the [ST][1](#), (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-22

Function 40 [conditional]

The evaluator shall verify that the TSS includes a description of available backup methods that can be enabled/disabled. If “selected applications or selected groups of applications are selected the TSS shall include which applications of groups of applications backup can be enabled/disabled.

Summary

Function 40 is not selected in the [ST][1](#), (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-23

Function 41 [conditional]

The evaluator shall verify that the TSS includes a description of Hotspot functionality and USB tethering to include any authentication for these.

Summary

Function 41 is not selected in the [ST][1](#), (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-24

Function 45 [conditional]

- **Test 45:** *The evaluator shall verify that the TSS contains guidance to configure the VPN as Always-On.*

Summary

Section 8.9.4.1 *AlwaysOn VPN* in the [ST][1](#) describes the functionalities offered by the TOE regarding AlwaysOn VPN. The configuration is provided in [CCGUIDE][1](#).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-25

Function 46 [conditional]

- **Test 46:** *The evaluator shall verify that the TSS describes the procedure to revoke a biometric credential stored on the TOE.*

Summary

Function 46 is not selected in the [ST][1](#), (Table 4).

Assurance Activity AA-FMT_SMF_EXT.1-ASE-26

Function 47

The evaluator shall verify that the TSS describes all assigned security management functions and their intended behavior.

Summary

This function has not been selected by the ST author.

Guidance Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.1-AGD-01

The evaluator shall consult the AGD guidance to perform each of the specified tests, iterating each test as necessary if both the user and administrator may perform the function. The evaluator shall verify that the AGD guidance describes how to perform each management function, including any configuration details. For each specified management function tested, the evaluator shall confirm that the underlying mechanism exhibits the configured setting.

Summary

Guidance to perform each management function is provided throughout [CCGUIDE][1](#), notably in section 1.8 *Security Management Configuration* and section 3.6 *Security Management*, which has been assessed by the evaluator while performing other guidance assurance activities. Also, [CCGUIDE][1](#) section 1.8 contains table 4 "Required Mobile Device Management Functions" which includes management functions from table 4 of [ST][1](#) and management functions specific to Wi-Fi and VPN. For each of these management functions, table 4 provides pointer to the corresponding section in [CCGUIDE][1](#) where guidance can be found. In addition, all these management functions were performed (in accordance to the provided guidance) as part of independent testing.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-02

Function 4

[The evaluator shall verify that the TSS includes a description of each radio and an indication of if the radio can be enabled/disabled along with what role can do so.]

The evaluator shall confirm that the AGD guidance describes how to perform the enable/disable function for each radio.

Summary

Per the TSS (section 8.6.5 of [ST][1](#)), the following radios are found in the TOE:

- Cellular
- Wi-Fi
- Bluetooth

[CCGUIDE][1](#) section 3.6.7 *EEnable/Disable Cellular, Wi-Fi, Wi-Fi Hotspot, Bluetooth, NFC* provides related guidance for these functions which states that the TOE devices contain a variety of radios which can be configured by the users or administrators according to the organization's policy.

For these functions with radios, [CCGUIDE][1](#) mainly refers to the device user guides [iPhone_UG][1](#) and [iPad_UG][1](#) for guidance on how to perform the enable/disable functions of the devices. For example, the user is instructed to enable/disable Bluetooth by following the instructions provided in the *iPhone_UG* the *iPad_UG*: "Use iPhone with other devices >> Connect Bluetooth devices to iPhone.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-03**Function 5**

[The evaluator shall verify that the TSS includes a description of each collection device and an indication of if it can be enabled/disabled along with what role can do so.] The evaluator shall confirm that the AGD guidance describes how to perform the enable/disable function.

Summary

Guidance to perform each management function (including enable/disable the function) is provided throughout [CCGUIDE] [\[1\]](#), notably in section 1.8 *Security Management Configuration* and section 3.6 *Security Management*, which has been assessed by the evaluator while performing other guidance assurance activities. In particular, [CCGUIDE] [\[1\]](#) section 1.8 provides table 4 "Required Mobile Device Management Functions" lists for each management function claimed in table 4 of ST and for Wi-Fi and VPN functionality, the corresponding section in [CCGUIDE] [\[1\]](#) where guidance can be found as well as the user type/rule that can perform it.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-04**Function 11**

The evaluator shall review the AGD guidance to determine that it describes the steps needed to import, modify, or remove certificates in the Trust Anchor database, and that the users that have authority to import those certificates (e.g., only administrator, or both administrators and users) are identified.

Summary

Per table 4 of [ST] [\[1\]](#), only administrator can import certificates to the Trust Anchor and both users and administrators can remove imported certificates and no other certificates in the Trust Anchor.

[CCGUIDE] [\[1\]](#) table 4 refers to section 3.5.6 X.509 *Certificate Configuration* for related guidance on certificate management which states the following:

- In the evaluated configuration, users are not allowed to import X.509 certificates into the Trust Anchor.
- If allowed (by the administrator) in the evaluated configuration, the TOE users can manually remove certificates installed on their device by choosing Settings >> General >> Profile & Device Management>>Profiles, select a profile, choose More Details, and then choose the appropriate certificate to remove.
- In the evaluated configuration, the user can only remove imported X.509v3 certificates but cannot remove other (pre-installed) X.509v3 certificates in the Trust Anchor.
- Administrator configures X.509 certificates using the *Certificate Payload* in a Configuration Profile.
- Administrators can remove any certificate installed via the MDM protocol using the *RequestType* key with the content "CertificateList" in a Configuration Profile.
- Administrators can modify the Trust Anchor via a Configuration Profile using either the Certificate Payload or SCEP Payload.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-05**Function 13**

[The evaluator shall examine the TSS to ensure that it contains a description of each management function that will be enforced by the enterprise once the device is enrolled.] The evaluator shall examine the AGD guidance to determine that this same information is present.

Summary

Table 4 of [ST] identifies the management functions that will be enforced by the enterprise once the device is enrolled. [CCGUIDE] table 4 "Required Mobile Device Management Functions" provides pointers to the corresponding guidance. While performing related assurance activities the evaluator, the evaluator verified that the information in the TSS is consistent with [CCGUIDE].

Assurance Activity AA-FMT_SMF_EXT.1-AGD-06

Function 14

The evaluator shall examine the AGD guidance to determine that it details, for each type of application that can be removed, the procedures necessary to remove those applications and their associated data. For the purposes of this assurance activity, "associated data" refers to data that are created by the app during its operation that do not exist independent of the app's existence, for instance, configuration data, or e-mail information that's part of an e-mail client. It does not, on the other hand, refer to data such as word processing documents (for a word processing app) or photos (for a photo or camera app).

Summary

[CCGUIDE] section 3.6.1 *Install/Remove Apps from the Device* provides related guidance on installing/removing apps. It states the following:

- For enrolled TOE devices, managed apps can be removed by an administrator remotely via the MDM system or when the user removes their own device from the MDM.
- TOE users can install or remove apps from their device as described in [iPhone_UG] and [iPad_UG]. This is of course also subject to the organization's policy and the settings of the Restriction Payload in a Configuration Profile.
- Administrator can install apps using an MDM system or Apple Configurator 2 as described in *IOS_MDM* and [AConfig], respectively.
- When a managed app is removed from a device, the associated data is removed with it.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-08

Function 19

The evaluator shall examine the AGD Guidance to determine that it specifies, for at least each category of information selected for Function 19, how to enable and disable display information for that type of information in the locked state.

Summary

Per [ST], the TOE can enable/disable display notifications, in the locked state for all notifications.

[CCGUIDE] section 3.6.2 *Configure Access and Notification in Locked State* provides related guidance which states the following:

- Certain display notifications can be set when the mobile device is locked. These notifications can be enable/disable via the device settings and as described in [iPhone_UG] and [iPad_UG].
- Administrator can restrict user from viewing past notifications by disabling Notification history setting the *allowLockScreenNotification* key in the Restrictions Payloads in a Configuration Profile.
- Likewise, administrator can disable displaying notifications for applications by setting the *ShowInLockScreen* key.

- Once the notification settings have been implemented by the mobile device administrator, the allowNotificationsModification key in the Restrictions Payload must be set to 'true' if the settings are not allowed to be modified.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-09

Function 24 [conditional]

[The evaluator shall verify that the TSS includes a list of each externally accessible hardware port and an indication of if data transfer over that port can be enabled/disabled.] AGD guidance will describe how to perform the enable/disable function.

Summary

This function is optional and not selected in [ST]📄, thus no guidance is required.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-10

Function 27 [conditional]

The evaluator shall examine the AGD guidance to determine that it describes how to enable and disable any "Forgot Password", password hint, or remote authentication (to bypass local authentication mechanisms) capability.

Summary

This function is optional and not selected in [ST]📄, thus no guidance is required.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-11

Function 29 [conditional]

The evaluator shall also verify that the API documentation provided according to Section 5.2.2 includes any security functions (import, modification, or destruction of the Trust Anchor Database) allowed by applications.

Summary

This function is optional and not selected in [ST]📄, thus no guidance is required.

Assurance Activity AA-FMT_SMF_EXT.1-AGD-12

Function 31 [conditional]

The evaluator shall confirm that the AGD guidance describes the procedure for disabling each cellular protocol identified in the TSS.

Summary

This function is optional and not selected in [ST]📄, thus no guidance is required.

Test Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.1-ATE-01

Test activities specified below shall take place in the test environment described in the Assurance Activity for FPT_TUD_EXT.1.1, FPT_TUD_EXT.1.2, and FPT_TUD_EXT.1.3. The evaluator shall consult the AGD guidance to perform each of the specified tests, iterating each test as necessary if both the user and administrator may perform the function.

The evaluator shall verify that the AGD guidance describes how to perform each management function, including any configuration details. For each specified management function tested, the evaluator shall confirm that the underlying mechanism exhibits the configured setting.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Assurance Activity AA-FMT_SMF_EXT.1-ATE-02

Function 1

- **Test 1:** *The evaluator shall exercise the TSF configuration as the administrator and perform positive and negative tests, with at least two values set for each variable setting, for each of the following:*
 - *minimum password length*
 - *minimum password complexity*
 - *maximum password lifetime*

Summary

Test 1: the evaluator used the Apple Configurator 2 (AC2) to setup the password policy for the TOE. The evaluator loaded the profile on the TOE and attempted to change the password to a password not fulfilling the password policy (either in length or complexity). The evaluator verified that the TOE enforced the password policy.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-03

Function 2

- **Test 2:** *The evaluator shall exercise the TSF configuration as the administrator. The evaluator shall perform positive and negative tests, with at least two values set for each variable setting, for each of the following.*
 - *screen-lock enabled/disabled*
 - *screen lock timeout*
 - *number of authentication failures (may be combined with test for FIA_AFL_EXT.1)*

Summary

Test 2: the evaluator used the AC2 to configure auto-lock on the TOE, loaded the profile on the TOE and verified that it was not possible to select a longer auto-lock timeout on the TOE. The evaluator then measured how long it took for the TOE to timeout and verified that it was consistent with the policy.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-04

Function 3

- **Test 3:** *The evaluator shall perform the following tests:*
 - *The evaluator shall exercise the TSF configuration to enable the VPN protection. These configuration actions must be used for the testing of the FDP_IFC_EXT.1.1 requirement.*
 - *[conditional] If “per-app basis” is selected, the evaluator shall create two applications and enable one to use the VPN and the other to not use the VPN. The evaluator shall exercise each application (attempting to access network resources; for example, by browsing different websites) individually while capturing*

- packets from the TOE. The evaluator shall verify from the packet capture that the traffic from the VPN-enabled application is encapsulated in IPsec and that the traffic from the VPN-disabled application is not encapsulated in IPsec.
- [conditional] If “per-groups of application basis” is selected, the evaluator shall create two applications and the applications shall be placed into different groups. Enable one application group to use the VPN and the other to not use the VPN. The evaluator shall exercise each application (attempting to access network resources; for example, by browsing different websites) individually while capturing packets from the TOE. The evaluator shall verify from the packet capture that the traffic from the application in the VPN-enabled group is encapsulated in IPsec and that the traffic from the application in the VPN-disabled group is not encapsulated in IPsec.

Summary

Test 3a: See FDP_IFC_EXT.1(MDF). Test 3b, 3c: not applicable.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-05

Function 4

The evaluator shall ensure that minimal signal leakage enters the RF shielded enclosure (i.e., Faraday bag, Faraday box, RF shielded room) by performing the following steps:

- Step 1: Place the antenna of the spectrum analyzer inside the RF shielded enclosure.
- Step 2: Enable “Max Hold” on the spectrum analyzer and perform a spectrum sweep of the frequency range between 300MHz – 6000MHz, in 1 KHz steps (this range should encompass 802.11, 802.15, GSM, UMTS, LTE and GPS). This range will not address NFC 13.56MHz, another test should be set up with similar constraints to address NFC.

If power above -90 dBm is observed, the Faraday box has too great of signal leakage and shall not be used to complete the test for Function 4.

- **Test 4:** The evaluator shall exercise the TSF configuration as all roles specified in the TSS to enable and disable the state of each radio (e.g. Wi-Fi, GPS, cellular, NFC, Bluetooth). Additionally, the evaluator shall repeat the steps below, booting into any auxiliary boot mode supported by the device. For each radio, the evaluator shall:
 - Step 1: Place the antenna of the spectrum analyzer inside the RF shielded enclosure. Configure the spectrum analyzer to sweep desired frequency range for the radio to be tested (based on range provided in the TSS). The ambient noise floor shall be set to - 110dBm. Place the TOE into the RF shielded enclosure to isolate them from all other RF traffic.
 - Step 2: The evaluator shall create a baseline of the expected behavior of RF signals. The evaluator shall power on the device, ensure the radio in question is enabled, power off the device, enable “Max Hold” on the spectrum analyzer and power on the device. The evaluator shall wait 2 minutes at each Authentication Factor interface prior to entering the necessary password to complete the boot process, waiting 5 minutes after the device is fully booted. The evaluator shall observe that RF spikes are present at the expected uplink channel frequency. The evaluator shall clear the “Max Hold” on the spectrum analyzer.
 - Step 3: The evaluator shall verify the absence of RF activity for the uplink channel when the radio in question is disabled. The evaluator shall complete the following test five times. The evaluator shall power on the device, ensure the radio in question is disabled, power off the device, enable “Max Hold” on the spectrum analyzer and power on the device. The evaluator shall wait 2 minutes at each Authentication Factor interface prior to entering the necessary password to complete the boot process, waiting 5 minutes after the device is fully booted. The evaluator shall clear the “Max Hold” on the spectrum analyzer. If a spike of RF activity for the uplink channel of the specific radio frequency band is observed at any time (either at an Authentication Factor interface or when the device is fully booted) it is deemed that the radio is enabled.

Summary

Test 4: the evaluator put the TOE in a shielded room and set the spectrum analyzer to "Max Hold" and verified that the power emanating from the TOE had a baseline measurement showing that a transmission spike is about 40dBm or higher (from -90dBm to -40dBm).

Assurance Activity AA-FMT_SMF_EXT.1-ATE-06

Function 5

- **Test 5:** *The evaluator shall perform the following test(s):*
 - a. *The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable and disable the state of each audio or visual collection devices (e.g. camera, microphone) listed by the ST author. For each collection device, the evaluator shall disable the device and then attempt to use its functionality. The evaluator shall reboot the TOE and verify that disabled collection devices may not be used during or early in the boot process. Additionally, the evaluator shall boot the device into each available auxiliary boot mode and verify that the collection device cannot be used.*
 - b. *[conditional] If “per-app basis” is selected, the evaluator shall create two applications and enable one to use access the A/V device and the other to not access the A/V device. The evaluator shall exercise each application attempting to access the A/V device individually. The evaluator shall verify that the enabled application is able to access the A/V device and the disabled application is not able to access the A/V device.*
 - c. *[conditional] If “per-groups of application basis” is selected, the evaluator shall create two applications and the applications shall be placed into different groups. Enable one group to access the A/V device and the other to not access the A/V device. The evaluator shall exercise each application attempting to access the A/V device individually. The evaluator shall verify that the application in the enabled group is able to access the A/V device and the application in the disabled group is not able to access the A/V device.*

Summary

Test 5a (camera): the evaluator disabled the camera app on the TOE and verified that the camera app disappeared from the main page. Also, the evaluator installed a QR scanner app and verified that the camera was not accessible through the QR scanner app.

Test 5a (microphone): the evaluator disabled the microphone on the TOE, installed Google Chrome, and tried to perform a search with the microphone. The evaluator verified that this operation failed.

Test 5a (aux boot): the evaluator powered off the devices and powered them on using a special combination of keys. The evaluator verified that the iTunes logo was displayed.

Test 5b (camera): the evaluator enabled the camera on the TOE, verified that the app is present, opened the QR scanner app and verified that it prompts for camera access permission. The evaluator accepted, and verified that the app was authorized by the OS in the settings to use the camera, and that the QR scanner app could make use of the camera. The evaluator then removed the camera usage permission or the QR scanner app, and verified that the app could not use the camera anymore.

Test 5b (microphone): the evaluator enabled the microphone on the TOE and used Chrome to search using the voice. The evaluator then verified that the Chrome app had the microphone access permission in the settings. The evaluator then removed this permission and verified, by using the Chrome app, that the microphone could not be used.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-07**Function 6**

- **Test 6:** *The evaluator shall use the test environment to instruct the TSF, both as a user and as the administrator, to command the device to transition to a locked state, and verify that the device transitions to the locked state upon command.*

Summary

Test 6: the evaluator pressed the power button and verified that the phone was locked, by pressing the home button and verifying that the device requests the password. The evaluator then enrolled the devices into iCloud and lost mode, and triggered a remote lock of the device. Again, the evaluator verified that the device was locked.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-08**Function 7**

- **Test 7:** *The evaluator shall use the test environment to instruct the TSF, both as a user and as the administrator, to command the device to perform a wipe of protected data. The evaluator must ensure that this management setup is used when conducting the assurance activities in FCS_CKM_EXT.5.*

Summary

Test 7: the evaluator went to Settings -> General -> Reset -> "Erase All Content and Settings", and verified that all the data was wiped from the device.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-09**Function 8**

- **Test 8:** *The evaluator shall exercise the TSF configuration as the administrator to restrict particular applications, sources of applications, or application installation according to the AGD guidance. The evaluator shall attempt to install unauthorized applications and ensure that this is not possible. The evaluator shall, in conjunction, perform the following specific tests:*
 - a. **Test 8a:** *[conditional] The evaluator shall attempt to connect to an unauthorized repository in order to install applications.*
 - b. **Test 8b:** *[conditional] The evaluator shall attempt to install two applications (one whitelisted, and one not) from a known allowed repository and verify that the application not on the whitelist is rejected. The evaluator shall also attempt to side-load executables or installation packages via USB connections to determine that the white list is still adhered to.*

Summary

Test 8: the evaluator created a profile through AC2 that disabled installation of application. The evaluator loaded that profile and verified that the App Store was not present anymore.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-11**Function 9 & Function 10**

- **Test 9 & Test 10:** *The test of these functions is performed in association with FCS_STG_EXT.1.*

Summary

See FCS_STG_EXT.1.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-12**Function 11**

- **Test 11:** *The evaluator shall import certificates according to the AGD guidance as the user and/or as the administrator, as determined by the administrative guidance. The evaluator shall verify that no errors occur during import. The evaluator should perform an action requiring use of the X.509v3 certificate to provide assurance that installation was completed properly.*

Summary

See tests for FCS_TLSC_EXT.1(MDFPP) and FCS_TLSC_EXT.1(WLAN).

Assurance Activity AA-FMT_SMF_EXT.1-ATE-13

Function 12

- **Test 12:** *The evaluator shall remove an administrator-imported certificate and any other categories of certificates included in the assignment of function 14 from the Trust Anchor Database according to the AGD guidance as the user and as the administrator.*

Summary

The evaluator installed a profile containing certificates and tried to delete/remove it. The TOE either refused or prompted for a password/passcode.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-14**Function 13**

- **Test 13:** *The evaluator shall verify that user approval is required to enroll the device into management.*

Summary

The evaluator obtained the root certificate signing the certificates for the Apple Profile manager ("atsec Open Directory Certificate Authority") and added it into a profile, deployed the profile to the TOE, and connected the TOE to the WLAN. The evaluator then connected Apple laptop with Profile Manager to WLAN, registered each TOE device's serial number with Apple Profile manager - this step is considered the approval process where a particular device is approved to be usable with the MDM solution. On the TOE, the evaluator started Safari, accessed "https://osxserver.atsec.com/" and selected "Profile Manager". The evaluator logged into Profile Manager with user "testuser" and password "test", and clicked on enroll. The evaluator finally verified that the enrollment profile is loaded onto the TOE.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-15**Function 14**

- **Test 14:** *The evaluator shall attempt to remove applications according to the AGD guidance and verify that the TOE no longer permits users to access those applications or their associated data.*

Summary

The evaluator used the App Store application to install Google Chrome and used the app to browse some website. The evaluator then remove the app from the device, and verified that the app was indeed removed and not usable.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-16**Function 15**

- **Test 15:** *The evaluator shall attempt to update the TSF system software following the procedures in the AGD guidance and verify that updates correctly install and that the version numbers of the system software increase.*

Summary

The evaluator checked the current version: Settings -> General -> About -> Version, verified that it is iOS 11.4, installed the update to iOS 12.0, checked the version again and verified that it was indeed at 12.0.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-17

Function 16

- **Test 16:** The evaluator shall attempt to install an application following the procedures in the AGD guidance and verify that the application is installed and available on the TOE.

Summary

Covered by Function 14.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-18**Function 17**

- **Test 17:** The evaluator shall attempt to remove any Enterprise applications from the device by following the administrator guidance. The evaluator shall verify that the TOE no longer permits users to access those applications or their associated data.

Summary

The evaluator compiled and archived a test application and generated an IPA archive file using Xcode. The evaluator deployed this archive on the TOE using the AC2, verified that the app was installed, and executed properly. The evaluator then used the AC2 to remove the app from the TOE.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-19**Function 18**

- **Test 18:** The evaluator shall use a Bluetooth-specific protocol analyzer to perform the following tests of each sub-function:
 - a. The evaluator shall disable the Discoverable mode and shall verify that other Bluetooth BR/EDR devices cannot detect the TOE. The evaluator shall use the protocol analyzer to verify that the TOE does not respond to inquiries from other devices searching for Bluetooth devices. The evaluator shall enable Discoverable mode and verify that other devices can detect the TOE and that the TOE sends response packets to inquiries from searching devices.
 - b. The evaluator shall examine Bluetooth traffic from the TOE to determine the current Bluetooth device name, change the Bluetooth device name, and verify that the Bluetooth traffic from the TOE lists the new name.
 - c. [conditional] The evaluator shall examine Bluetooth traffic from the TOE to determine the current Bluetooth device name for BR/EDR and LE. The evaluator shall change the Bluetooth device name for LE independently of the device name for BR/EDR. The evaluator shall verify that the Bluetooth traffic from the TOE lists the new name.
 - d. [conditional] The evaluator shall disable Bluetooth BR/EDR and enable Bluetooth LE. The evaluator shall examine Bluetooth traffic from the TOE to confirm that only Bluetooth LE traffic is present. The evaluator shall repeat the test with Bluetooth BR/EDR enabled and Bluetooth LE disabled, confirming that only Bluetooth BR/EDR is present.
 - e. [conditional] The evaluator shall disable additional wireless technologies for the TOE and verify that Bluetooth High Speed is not able to send Bluetooth traffic over Wi-Fi, and that NFC cannot be used for pairing. The evaluator shall enable additional wireless technologies and verify that Bluetooth High Speed uses Wi-Fi or that the device can pair using NFC.
 - f. [conditional] The evaluator shall enable Advertising for Bluetooth LE, verify that the advertisements are captured by the protocol analyzer, disable Advertising, and verify that no advertisements from the device are captured by the protocol analyzer.
 - g. [conditional] The evaluator shall enable Connectable mode and verify that other Bluetooth devices may pair with the TOE and (if the devices were bonded) re-connect after pairing and disconnection. For BR/EDR devices: The evaluator shall use the protocol analyzer to verify that the TOE responds to pages from the other devices and permits pairing and re-connection. The evaluator shall disable Connectable mode and verify that the TOE does not respond to pages from remote Bluetooth devices, thereby not permitting pairing or re-connection. For LE: The evaluator shall use the protocol analyzer to verify that the TOE sends

connectable advertising events and responds to connection requests. The evaluator shall disable Connectable mode and verify that the TOE stops sending connectable advertising events and stops responding to connection requests from remote Bluetooth devices.

- h. [conditional] The evaluator shall verify that all supported Bluetooth services and/or profiles are listed in the TSS as manageable and, if the TOE allows disabling by application rather than by service and/or profile name, that a list of services and/or profile for each application is also listed.
- i. [conditional] The evaluator shall allow low security modes/levels on the TOE and shall initiate pairing with the TOE from a remote device that allows only something other than Security Mode 4/Level 3 or Security Mode 4/Level 4 (for BR/EDR), or Security Mode 1/Level 3 (for LE). (For example, a remote BR/EDR device may claim Input/Output capability "NoInputNoOutput" and state that man-in-the-middle (MiTM) protection is not required. A remote LE device may not support encryption.) The evaluator shall verify that this pairing attempt succeeds due to the TOE falling back to the low security mode/level. The evaluator shall then remove the pairing of the two devices, prohibit the use of low security modes/levels on the TOE, then attempt the connection again. The evaluator shall verify that the pairing attempt fails. With the low security modes/levels disabled, the evaluator shall initiate pairing from the TOE to a remote device that supports Security Mode 4/Level 3 or Security Mode 4/Level 4 (for BR/EDR) or Security Mode 1/Level 3 (for LE). The evaluator shall verify that this pairing is successful and uses the high security mode/level.
- j. [conditional] The evaluator shall attempt to pair using each of the Out of Band pairing methods, verify that the pairing method works, iteratively disable each pairing method, and verify that the pairing method fails.

Summary

Test 18a: the evaluator used the hcitool to scan for Bluetooth devices while Bluetooth was disabled on the TOE. The evaluator verified that the TOE was not found. The evaluator then tried to force a connection to the TOE using its Bluetooth ID, and verified that the connection did not succeed, and that no Bluetooth packets were recorded by the packet sniffer tool.

Test 18b: the evaluator connected the TOE to the test laptop using Bluetooth and sniffed the traffic using hcidump. The evaluator then used Wireshark to verify that the name of the TOE was correctly displayed in the "Remote Name Request Complete" packet. The evaluator then modified the name of the TOE in the settings and re-performed the same operations. The evaluator could verify that the packet then showed the new name of the TOE.

Test 18d, e, f, g, h, u, j are not claimed. Test 18c is covered by Test 18b.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-20

Function 19

- **Test 19:** For each category of information listed in the AGD guidance, the evaluator shall verify that when that TSF is configured to limit the information according to the AGD, the information is no longer displayed in the locked state.

Summary

The evaluator used the AC2 to disabled notifications on the lock screen. The evaluator created a calendar event and ensured that no notification was displayed on the lock screen, but that the notification was indeed displayed once the device was unlocked.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-21

Function 20

- **Test 20:** The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable system-wide data-at-rest protection according to the AGD guidance. The evaluator shall ensure that all assurance activities for FDP_DAR are conducted with the device in this configuration.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-22

Function 21

- **Test 21:** *The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable removable media's data-at-rest protection according to the AGD guidance. The evaluator shall ensure that all assurance activities for DAR (see FDP_DAR) are conducted with the device in this configuration.*

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-23

Function 22

- **Test 22:** *The evaluator shall perform the following tests.*
 - a. *The evaluator shall enable location services device-wide and shall verify that an application (such as a mapping application) is able to access the TOE's location information. The evaluator shall disable location services device-wide and shall verify that an application (such as a mapping application) is unable to access the TOE's location information.*
 - b. *[conditional] If "per-app basis" is selected, the evaluator shall create two applications and enable one to use access the location services and the other to not access the location services. The evaluator shall exercise each application attempting to access location services individually. The evaluator shall verify that the enabled application is able to access the location services and the disabled application is not able to access the location services.*

Summary

Test 22a: the evaluator enabled the location services on the TOE, opens the Maps app and verified that the TOE could find its location. The evaluator disabled the location services and verified that the TOE was unable to find its location.

Test 22b: the evaluator enabled the location services, opened Maps, and verified that the app could find the location of the device. The evaluator then disabled the location services specifically for the Maps app, opened maps, and verified that Maps could not find the location.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-24

Function 23

- **Test 23:** *The evaluator shall verify that the TSS states if the TOE supports a BAF and/or hybrid authentication. If the TOE does not include a BAF and/or hybrid authentication this test is implicitly met.*
 - a. *[conditional] If a BAF is selected the evaluator shall verify that the TSS describes the procedure to enable/disable the BAF. If the TOE includes multiple BAFs, the evaluator shall verify that the TSS describes how to enable/disable each BAF, specifically if the different modalities can be individually enabled/disabled. The evaluator shall configure the TOE to allow each supported BAF to authenticate and verify that successful authentication can be achieved using the BAF. The evaluator shall configure the TOE to disable the use of each supported BAF for authentication and confirm that the BAF cannot be used to authenticate.*
 - b. *[conditional] If "Hybrid" is selected the evaluator shall verify that the TSS describes the procedure to enable/disable the hybrid (biometric credential and PIN/password) authentication. The evaluator shall configure the TOE to allow hybrid authentication to authenticate and confirm that successful authentication can be achieved using the hybrid authentication. The evaluator shall configure the TOE to disable the use of hybrid authentication and confirm that the hybrid authentication cannot be used to authenticate.*

Summary

Test 23a: the evaluator enabled the biometric configuration for the devices (either TouchID or FaceID), locked the device, unlocked the device, removed the biometric configuration, and verified that the evaluator could not unlock the device anymore using biometric authentication factors.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-25

Assurance Activity Note: It should be noted that the following functions are optional capabilities, if the function is implemented, then the following assurance activities shall be performed. The notation of “[conditional]” beside the function number indicates that if the function is not included in the ST, then there is no expectation that the assurance activity be performed.

Function 24 [conditional]

- **Test 24:** The evaluator shall exercise the TSF configuration to enable and disable data transfer capabilities over each externally accessible hardware ports (e.g. USB, SD card, HDMI) listed by the ST author. The evaluator shall use test equipment for the particular interface to ensure that no low-level signaling is occurring on all pins used for data transfer when they are disabled. For each disabled data transfer capability, the evaluator shall repeat this test by rebooting the device into the normal operational mode and verifying that the capability is disabled throughout the boot and early execution stage of the device.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-27

Function 25 [conditional]

- **Test 25:** The evaluator shall attempt to disable each listed protocol in the assignment. The evaluator shall verify that remote devices can no longer access the TOE or TOE resources using any disabled protocols.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-28

Function 26 [conditional]

- **Test 26:** The evaluator shall exercise the TSF configuration as the administrator and, if not restricted to the administrator, the user, to enable and disable any developer mode. The evaluator shall test that developer mode access is not available when its configuration is disabled. The evaluator shall verify the developer mode remains disabled during device reboot.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-29

Function 27 [conditional]

- **Test 27:** For each mechanism listed in the AGD guidance that provides a “Forgot Password” feature or other means where the local authentication process can be bypassed, the evaluator shall disable the feature and ensure that they are not able to bypass the local authentication process.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-30

Function 28 [conditional]

- **Test 28:** The evaluator shall attempt to wipe Enterprise data resident on the device according to the administrator guidance. The evaluator shall verify that the data is no longer accessible by the user.

Summary

There is no difference between enterprise applications and their data and regular apps. Thus Test 16 covers this test too for application data. For a general wipe, FCS_CKM_EXT.4 demonstrates that all data is wiped on the device.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-31**Function 29 [conditional]**

- **Test 29:** The evaluator shall perform one of the following tests:
 - a. [conditional] If applications may import certificates to the Trust Anchor Database, the evaluator shall write, or the developer shall provide access to, an application that imports a certificate into the Trust Anchor Database. The evaluator shall verify that the TOE requires approval before allowing the application to import the certificate:
 - The evaluator shall deny the approvals to verify that the application is not able to import the certificate. Failure of import shall be tested by attempting to validate a certificate that chains to the certificate whose import was attempted (as described in the Assurance Activity for FIA_X509_EXT.1).
 - The evaluator shall repeat the test, allowing the approval to verify that the application is able to import the certificate and that validation occurs.
 - b. [conditional] If applications may remove certificates in the Trust Anchor Database, the evaluator shall write, or the developer shall provide access to, an application that removes certificates from the Trust Anchor Database. The evaluator shall verify that the TOE requires approval before allowing the application to remove the certificate:
 - The evaluator shall deny the approvals to verify that the application is not able to remove the certificate. Failure of removal shall be tested by attempting to validate a certificate that chains to the certificate whose removal was attempted (as described in the Assurance Activity for FIA_X509_EXT.1)

The evaluator shall repeat the test, allowing the approval to verify that the application is able to remove/modify the certificate and that validation no longer occurs.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-32**Function 30 [conditional]**

- **Test 30:** The test of this function is performed in conjunction with FIA_X509_EXT.2.2.

Summary

See FIA_X509_EXT.2.2.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-33**Function 31 [conditional]**

- **Test 31:** The evaluator shall attempt to disable each cellular protocol according to the administrator guidance. The evaluator shall attempt to connect the device to a cellular network and, using network analysis tools, verify that the device does not allow negotiation of the disabled protocols.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-34

Function 32 [conditional]

- **Test 32:** The evaluator shall attempt to read any device audit logs according to the administrator guidance and verify that the logs may be read. This test may be performed in conjunction with the assurance activity of FAU_GEN.1.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-35

Function 33 [conditional]

- **Test 33:** The test of this function is performed in conjunction with FPT_TUD_EXT.2.5.

Summary

See AA-FPT_TUD_EXT.2.4-ATE-01.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-36

Function 34 [conditional]

- **Test 34:** The test of this function is performed in conjunction with FPT_TUD_EXT.4.1.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-37

Function 35 [conditional]

- **Test 35:** The test of this function is performed in conjunction with FCS_STG_EXT.1.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-38

Function 36 [conditional]

- **Test 36:** The test of this function is performed in conjunction with FTA_TAB.1.

Summary

See FTA_TAB.1.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-39

Function 37 [conditional]

- **Test 37:** The test of this function is performed in conjunction with FAU_SEL.1.

Summary

See FAU_SEL.1.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-40**Function 38 [conditional]**

- **Test 38:** The test of this function is performed in conjunction with FPT_NOT_EXT.1.2.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-41**Function 39 [conditional]**

- **Test 39:** The evaluator shall perform the following tests based on the selections made in the table:
 - a. [conditional] The evaluator shall disable USB mass storage mode, attach the device to a computer, and verify that the computer cannot mount the TOE as a drive. The evaluator shall reboot the TOE and repeat this test with other supported auxiliary boot modes.
 - b. [conditional] The evaluator shall disable USB data transfer without user authentication, attach the device to a computer, and verify that the TOE requires user authentication before the computer can access TOE data. The evaluator shall reboot the TOE and repeat this test with other supported auxiliary boot modes.
 - c. [conditional] The evaluator shall disable USB data transfer without connecting system authentication, attach the device to a computer, and verify that the TOE requires connecting system authentication before the computer can access TOE data. The evaluator shall then connect the TOE to another computer and verify that the computer cannot access TOE data. The evaluator shall then connect the TOE to the original computer and verify that the computer can access TOE data.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-42**Function 40 [conditional]**

- **Test 40:** If “all applications” is selected, the evaluator shall disable each selected backup location in turn and verify that the TOE cannot complete a backup. The evaluator shall then enable each selected backup location in turn and verify that the TOE can perform a backup.
If “selected applications” is selected, the evaluator shall disable each selected backup location in turn and verify that for the selected application the TOE prevents backup from occurring. The evaluator shall then enable each selected backup location in turn and verify that for the selected application the TOE can perform a backup.
If “selected groups of applications” is selected, the evaluator shall disable each selected backup location in turn and verify that for a group of applications the TOE prevents the backup from occurring. The evaluator shall then enable each selected backup location in turn and verify for the group of application the TOE can perform a backup.
If “configuration data” is selected, the evaluator shall disable each selected backup location in turn and verify that the TOE prevents the backup of configuration data from occurring. The evaluator shall then enable each selected backup location in turn and verify that the TOE can perform a backup of configuration data.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-43**Function 41 [conditional]**

- **Test 41:** The evaluator shall perform the following tests based on the selections in 0.
 - a. [conditional] The evaluator shall enable hotspot functionality with each of the of the support authentication methods. The evaluator shall connect to the hotspot with another device and verify that the hotspot functionality requires the configured authentication method.
 - b. [conditional] The evaluator shall enable USB tethering functionality with each of the of the support authentication methods. The evaluator shall connect to the TOE over USB with another device and verify that the tethering functionality requires the configured authentication method.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-44**Function 42 [conditional]**

- **Test 42:** The test of this function is performed in conjunction with FDP_ACF_EXT.1.2.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-45**Function 43 [conditional]**

- **Test 43:** The evaluator shall set a policy to cause a designated application to be placed into a particular application group. The evaluator shall then install the designated application and verify that it was placed into the correct group.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-46**Function 44 [conditional]**

- **Test 44:** The evaluator shall attempt to unenroll the device from management and verify that the steps described in FMT_SMF_EXT.2.1 are performed. This test should be performed in conjunction with the FMT_SMF_EXT.2.1 assurance activity.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-47**Function 45 [conditional]**

- **Test 45:** The evaluator shall configure the VPN as Always-On and perform the following test.
 - a. The evaluator shall verify that when the VPN is connected all traffic is routed through the VPN. This test is performed in conjunction with FDP_IFC_EXT.1.1.

- b. *The evaluator shall verify that when the VPN is not established, that no traffic leaves the device. The evaluator shall ensure that the TOE has network connectivity and that the VPN is established. The evaluator shall use a packet sniffing tool to capture the traffic leaving the TOE. The evaluator shall disable the VPN connection on the server side. The evaluator shall perform actions with the device such as navigating to websites, using provided applications, and accessing other Internet resources and verify that no traffic leaves the device.*
- c. *The evaluator shall verify that the TOE has network connectivity and that the VPN is established. The evaluator shall disable network connectivity (i.e. Airplane Mode) and verify that the VPN disconnects. The evaluator shall reestablish network connectivity and verify that the VPN automatically reconnects.*

Summary

Test 45a: See tests for FDP_IFC_EXT.1.1.

Test 45b: See test for FDP_IFC_EXT.1.1 – the traffic is generated by pinging the TOE demonstrating that the device will not return an ICMP echo reply, but only attempts to establish an IPsec tunnel (i.e. IKE traffic)

Test 45c: See test and results of FDP_IFC_EXT.1.1 – enabling and disabling of WLAN connectivity is performed including a review of automated reconnect.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-48

Function 46 [conditional]

The evaluator shall configure the TOE to use BAF and confirm that the biometric can be used to authenticate to the device. The evaluator shall revoke the biometric credential's ability to authenticate to the TOE and confirm that the same BAF cannot be used to authenticate to the device.

Summary

Not claimed.

Assurance Activity AA-FMT_SMF_EXT.1-ATE-49

Function 47

[The evaluator shall verify that the TSS describes all assigned security management functions and their intended behavior.]

- **Test 47:** *The evaluator shall design and perform tests to demonstrate that the function may be configured and that the intended behavior of the function is enacted by the TOE.*

Summary

Not claimed.

2.1.5.3 Extended: Specification of Specification of Remediation Actions (FMT_SMF_EXT.2)



TSS Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.2-ASE-01

The evaluator shall verify that the TSS describes all available remediation actions, when they are available for use, and any other administrator-configured triggers. The evaluator shall verify that the TSS describes how the remediation actions are provided to the administrator.

Summary

Section 8.6.4 in the [ST]  describes the *Unenrollment* .

Section 8.6.4 references [IOS_CFG]  and [IOS_MDM] , the Configuration Profile Key Reference, which describes the unenrollment options. The optional key PayloadRemovalDisallowed, if present and set to true, prevents the user from deleting the configuration profile, unless this profile has a password that the user can provide.

Guidance Assurance Activities





No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.2-ATE-01

The evaluator shall use the test environment to iteratively configure the device to perform each remediation action in the selection. The evaluator shall configure the remediation action per how the TSS states it is provided to the administrator. The test environment could be a MDM agent application, but can also be an application with administrator access.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

The evaluator went to Profile Manager -> testgroup1 -> Settings, and unselected "Prevent unenrollment". The evaluator wiped the device and reinstalled the device as explained in FPT_SMF_EXT.3.2 Test 1 - Business Manager. On the device, the evaluator went to Settings -> General -> Device Management -> Remote Management, and removed the profile.

2.1.5.4 Trusted Policy Update (FMT_POL_EXT.2(AGENT))



FMT_POL_EXT.2.1

TSS Assurance Activities

Assurance Activity AA-FMT_POL_EXT.2.1-MDMA-ASE-01

The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate policies are obtained by the MDM Agent; the processing associated with verifying the digital signature of the policy updates; and the actions that take place for successful (signature was verified) and unsuccessful (signature could not be verified) cases. The software components that are performing the processing must also be identified in the TSS and verified by the evaluators.

Summary

Section 8.6.2 of the [ST]  describes the *Configuration Profiles* . Section 8.1 in the [ST]  describes the *Mapping to the Security Functional Requirements* .

The evaluator ensured that the TSS describes that candidate policies obtained by the MDM agent can be digitally signed and encrypted, as described in paragraph three of section 8.6.2. The evaluator also ensured by looking at the entry FMT_POL_EXT.2 and verified that MDM policies can be signed against enterprise signatures.

Section 8.10.2.2 states that:

"Candidate policies are generated by the administrator and disseminated as a configuration profile using one of the methods already described in section 8.6.2 above."

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FMT_POL_EXT.2.1-MDMA-ATE-01

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.

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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[1\]](#).

Test 1: the evaluator deployed a password policy on the TOE through the Apple Profile Manager, and could verify that the added policy's signature was successfully verified by the TOE.

Test 2: the evaluator created a policy with one option, generated a certificate using the keychain tool, signed the policy using a certificate that the TOE would not accept. The evaluator then loaded the policy and verified that the TOE prompted the evaluator with the choice to accept the profile or not.

FMT_POL_EXT.2.2

TSS Assurance Activities

Assurance Activity AA-FMT_POL_EXT.2.2-MDMA-ASE-01

The assurance activity for this requirement is performed in conjunction with the assurance activity for FIA_X509_EXT.1 and FIA_X509_EXT.2.

Summary

Guidance Assurance Activities

Assurance Activity AA-FMT_POL_EXT.2.2-MDMA-AGD-01

The assurance activity for this requirement is performed in conjunction with the assurance activity for FIA_X509_EXT.1 and FIA_X509_EXT.2.

Summary

[CCGUIDE][\[1\]](#) This assurance activity is performed in conjunction with the assurance activity for FIA_X509_EXT.1 and FIA_X509_EXT.2.

Test Assurance Activities

Assurance Activity AA-FMT_POL_EXT.2.2-MDMA-ATE-01

The assurance activity for this requirement is performed in conjunction with the assurance activity for FIA_X509_EXT.1 and FIA_X509_EXT.2 as defined in the base PPs.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The assurance activity for this requirement is performed in conjunction with the assurance activity for FIA_X509_EXT.1 and FIA_X509_EXT.2 as defined in the base PPs.

2.1.5.5 Specification of Management Functions (FMT_SMF_EXT.3(AGENT))

TSS Assurance Activities

See assurance activities for SFR elements below.

Guidance Assurance Activities

See assurance activities for SFR elements below.

Test Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.3-ATE-01

[TD0301] The evaluator shall cause the TOE to be enrolled into management. The evaluator shall then invoke this mechanism and verify the ability to view that the device has been enrolled, view the management functions that the administrator is authorized to perform.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Not applicable because not claimed in the ST.

FMT_SMF_EXT.3.1

TSS Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.3.1-MDMA-ASE-01

The evaluator shall verify that the any assigned functions are described in the TSS and that these functions are documented as supported by the platform. The evaluator shall examine the TSS to verify that any differences between management functions and policies for each supported Mobile Device are listed.

Summary

Section 8.6.2 of the [ST] describes the *Configuration Profiles*.

Since all the Mobile Devices specified in this [ST] use iOS, there are no differences between supported management functions and policies between the different mobile devices. The supported management functions for iOS are described in [IOS_CFG]. The evaluator verified that the functions provided in FMT_SMF_EXT.3.1 are all configurable through the configuration profiles defined by the administrator (or MDM administrator) of the TOE.

Guidance Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.3.1-MDMA-AGD-01

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

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

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

Summary

The evaluator noted that administrator-provided management functions in [PP_MD_V3.1], [EP_MDM_AGENT_V3.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1] are provided throughout [CCGUIDE]. A high-level summary is provided in Table 4 "Required Mobile Device Management Functions" of [CCGUIDE] where for each management function, a pointer to the relevant guidance is provided. Also, these management functions are already assessed in prior work units of this report, in particular, the work units of AA-FMT_SMF_EXT.1-AGD.

Additionally, [CCGUIDE] provides Table 3 "SFR Configuration Requirements" outlining for each function (expressed in the form of an SFR and its description) whether configuration is needed and referencing the section where related guidance is provided, if any.

Test Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.3.1-MDMA-ATE-01

Test 1: In conjunction with the assurance 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 2: The evaluator shall configure the MDM Agent authentication certificate in accordance with the configuration guidance. The evaluator shall verify that the MDM Agent uses this certificate in performing the tests for FPT_ITT.1.

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 mobile device.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator re-performed all the tests from FMT_SMF_EXT.1.1 for all the management function marked as 'X' in the "Administrator when enrolled in MDM" column in table 3 of [ST].

Test 2: covered by FPT_ITT.1.

Test 3: not applicable.

FMT_SMF_EXT.3.2

TSS Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.3.2-MDMA-ASE-01

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

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

Additionally, the evaluator shall verify that the TSS describes any management functions of the MDM Agent, if assigned.

Summary

Section 8.5.3 of the [ST] describes the *MDM Server Reference ID*. Section 8.6.1, *Enrollment*, in the [ST] describes how a device enrolls with an MDM server.

Section 8.6.1 describes that a device can enroll in multiple ways with an MDM server as follows.

- Manually, using Apple's Profile Manager
- Manually, using Apple Configurator 2
- Distributing an enrollment profile via email, or a web site
- Device Enrollment Program (This is an automated and enforced method of automatically enrolling new devices.)

Section 8.5.3 also describes that iOS devices automatically connect to the MDM Server during setup if the device is enrolled into the Device Enrollment Program (DEP) and is assigned to an MDM Server. No other function was defined for the MDM agent than the pre-defined ones in FMT_SMF_EXT.3.2.

Guidance Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.3.2-MDMA-AGD-01

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

Summary

The evaluator noted that administrator-provided management functions in [PP_MD_V3.1], [EP_MDM_AGENT_V3.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1] are provided throughout [CCGUIDE]. A high-level summary is provided in Table 4 "Required Mobile Device Management Functions" of [CCGUIDE] where for each management function, a pointer to the relevant guidance is provided. Also, these management functions are already assessed in prior work units of this report, in particular, the work units of AA-FMT_SMF_EXT.1-AGD.

Additionally, [CCGUIDE] provides Table 3 "SFR Configuration Requirements" outlining for each function (expressed in the form of an SFR and its description) whether configuration is needed and referencing the section where related guidance is provided.

Test Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.3.2-MDMA-ATE-01

Test 1: In conjunction with other assurance 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 2: (conditional) In conjunction with the assurance 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 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 mobile device.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator enrolled the device manually; manual enrollment during setup, where the device is provisioned by the Apple Configurator 2; DEP (Device Enrollment Program) where the devices were registered according to the steps outlined at the Device Enrollment Program Guide (https://ssl.apple.com/business/docs/DEP_Guide.pdf). In all cases, the evaluator could verify that the TOE was correctly enrolled into the MDM.

Test 2 and 3: not applicable.

2.1.5.6 User Unenrollment Prevention (FMT_UNR_EXT.1(AGENT))

TSS Assurance Activities

Assurance Activity AA-FMT_UNR_EXT.1-MDMA-ASE-01

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

Summary

Section 8.6.4, *Unenrollment*, in the [ST] describes the unenrollment process for the TOE.

Section 8.6.4 describes that if the configuration profile key PayloadRemovalDisallowed isn't present and set to true, the user cannot delete the profile unless the profile has a removal password and the user provides it. It is up to the MDM server to set this key.

Guidance Assurance Activities

Assurance Activity AA-FMT_UNR_EXT.1-MDMA-AGD-01

The evaluator shall ensure that the administrative guidance instructs administrators in configuring the unenrollment prevention in each available configuration interface. If any configuration allows users to unenroll, the guidance also describes the actions that unenroll the Agent.

Summary

[CCGUIDE] section 2.3.4 *Device Un-enrollment Prevention* provides guidance on how configure unenrollment prevention as follows:

- A mobile device is enrolled with the MDM server via a Configuration Profile that includes an MDM Payload. If this payload is removed, the device is no longer enrolled with the MDM server and can no longer in the evaluated configuration.
- To allow or disallow a device user from removing the MDM Payload from the device, device administrator can set the PayloadRemovalDisallowed key as described in [IOS_CFG].

- The mobile device must be in Supervised Mode to lock the MDM Payload to the device.
- An MDM Payload can have a removal password associated with it. If the PayloadRemovalDisallowed key is set to prevent unenrollment and the MDM Payload has a removal password associated with it, the mobile device user can unenroll the mobile device only if the mobile device user knows the removal password.

Test Assurance Activities

Assurance Activity AA-FMT_UNR_EXT.1-MDMA-ATE-01

If 'prevent the unenrollment from occurring' is selected:

Test 1: 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.

If 'apply remediation actions' is selected:

Test 2: 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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1 and 2: the evaluator used the Profile Manager to go to Device Groups -> testgroup1 -> Settings, and unselected Unenrollment. The evaluator then saved the setting and wiped the devices and could verify that the profile was removed from the devices

2.1.5.7 Specification of Management Functions (VPN) (FMT_SMF.1(VPN))

TSS Assurance Activities

Assurance Activity AA-FMT_SMF.1-VPN-ASE-01

The evaluator shall check to ensure the TSS describes the client credentials and how they are used by the TOE.

Summary

Please refer to AA-FCS_IPSEC_EXT.1.11-VPN-ASE-01 for Peer Authentication.

Guidance Assurance Activities

Assurance Activity AA-FMT_SMF.1-VPN-AGD-01

The evaluator shall check to make sure that every management function mandated in the ST for this requirement are described in the operational guidance and that the description contains the information required to perform the management duties associated with each management function.

Summary

The evaluator noted that administrator-provided management functions defined in [PP_MD_V3.1] and [MOD_VPN_CLI_V2.1] are provided throughout [CCGUIDE]. A high-level summary is provided in [CCGUIDE] Table 4 "Required Mobile Device Management Functions" where for each management

function, a pointer to the related guidance is provided. Also, these management functions are already assessed in prior work units of this report, in particular, the work units of AA-FMT_SMF_EXT.1-AGD.

Additionally, [CCGUIDE] provides Table 3 "SFR Configuration Requirements" outlining for each function (expressed in the form of an SFR and its description) whether configuration is required.

Test Assurance Activities

Assurance Activity AA-FMT_SMF.1-VPN-ATE-01

The evaluator shall test the TOE's ability to provide the management functions by configuring the TOE according to the operational guidance and testing each management activity listed in the Security Target.

The evaluator shall ensure that all management functions claimed in the ST can be performed by completing activities described in the AGD. Note that this may be performed in the course of completing other testing.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator executed all the test required by [MOD_VPN_CLI_V2.1], which covers all the management functions defined in FMT_SMF_EXT.1.

2.1.5.8 Specification of Management Functions (Wireless LAN) (FMT_SMF_EXT.1(WLAN))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.1-WLAN-AGD-01

The evaluator shall check to make sure that every management function mandated by the EP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function.

Summary

The evaluator noted that administrator-provided management functions defined in [PP_MD_V3.1] and [PP_WLAN_CLI_EP_V1.0] are provided throughout [CCGUIDE]. A high-level summary is provided in [CCGUIDE] Table 4 "Required Mobile Device Management Functions" where for each management function, a pointer to the related guidance is provided. Also, these management functions are already assessed in prior work units of this report, in particular, the work units of AA-FMT_SMF_EXT.1-AGD.

Additionally, [CCGUIDE] provides Table 3 "SFR Configuration Requirements" outlining for each function (expressed in the form of an SFR and its description) whether configuration is required.

Test Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.1-WLAN-ATE-01

The evaluator shall test the TOE's ability to provide the management functions by configuring the TOE and testing each option listed in the requirement above.

Note that the testing here may be accomplished in conjunction with the testing of other requirements, such as FCS_TLSC_EXT and FTA_WSE_EXT.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator verified that the TOE provided each management function listed in FMT_SMF_EXT.1, and configured, tested, and verified that the TOE indeed provides these functions. The evaluator logged the evidence using screenshots on the TOE.

Test 1: for each configured cipher, the tester configured OpenSSL to only accept one cipher suite, and used the TOE to connect to http://server. The tester verified that the access was possible.

Test 2: the tester verified that accessing http://server is possible, and http://10.0.0.1 is not when using a server certificate with the option "Server Authentication".

2.1.6 Protection of the TSF (FPT)

2.1.6.1 Extended: Anti-Exploitation Services (ASLR) (FPT_AEX_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.1-ASE-01

The evaluator shall ensure that the TSS section of the ST describes how the 8 bits are generated and provides a justification as to why those bits are unpredictable.

Summary

Section 8.7.5 describes the *Domain Isolation*.

Paragraph five of section 8.7.5 describes that the 8 bits generated for ALSR are taken from the application processor True Random Number Generator (TRNG) involved in the randomization.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.1-ATE-01

Assurance Activity Note: The following test require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- **Test 1:** The evaluator shall select 3 apps included with the TSF. These must include any web browser or mail client included with the TSF. For each of these apps, the evaluator will launch the same app on two separate Mobile Devices of the same type and compare all memory mapping locations. The evaluator must ensure that no memory mappings are placed in the same location on both devices.

If the rare (at most 1/256) chance occurs that two mappings are the same for a single app and not the same for the other two apps, the evaluator shall repeat the test with that app to verify that in the second test the mappings are different.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator performed the following tests for 3 iOS apps (e.g Safari, Mail, Messenger)

- have two identical devices on which you spawn the app
- Spawn the app on both devices
- Send signal 6 to the application to kill them and have them written the crash dump
- Synchronize the devices with iTunes
- Obtain the crash logs for the applications which contain the memory addresses of the text segment and the used shared libraries
- Verify that the memory locations differ in at least one byte

2.1.6.2 Extended: Anti-Exploitation Services (Memory Page Permissions) (FPT_AEX_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.2-ASE-01

The evaluator shall ensure that the TSS describes of the memory management unit (MMU), and ensures that this description documents the ability of the MMU to enforce read, write, and execute permissions on all pages of virtual memory.

Summary

Section 8.7.5 *Domain Isolation* of the [ST] describes the memory management unit (MMU) in the last paragraph:

" In addition, the Memory Management Unit (MMU) supports memory address translation using a translation table maintained by the OS kernel. For each page, the MMU maintains flags that allow or deny the read, write or execution of data. Execution in this case allows the CPU to fetch instructions from a given page. "

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.6.3 Extended: Anti-Exploitation Services (Overflow Protection) (FPT_AEX_EXT.3)

TSS Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.3-ASE-01

The evaluator shall determine that the TSS contains a description of stack-based buffer overflow protections implemented in the TSF software which runs in the non-privileged execution mode of the application processor. The exact implementation of stack-based buffer overflow protection will vary by platform. Example implementations may be activated through compiler options such as "-fstack-protector-all", "-fstackprotector", and "/GS" flags.

Summary

Section 8.7.5 describes *Domain Isolation*.

Section 8.7.5 describes that

"Stack-based buffer overflow protection is implemented for every sandbox."


Section 8.7.5, *Domain isolation*, specifies the mechanisms the the OS utilizes to perform stack-based buffer overflow protection:

- Automatic reference counting (ARC); a memory management system that handles the reference count of objects automatically at compile time
- Address space layout randomization (ASLR); discussed below, and
- Stack-smashing protection; by utilizing a canary on the stack. Apple recommends that developers compile applications using the -fstack-protector-all compiler flag

Assurance Activity AA-FPT_AEX_EXT.3-ASE-02

The evaluator shall ensure that the TSS contains an inventory of TSF binaries and libraries, indicating those that implement stack-based buffer overflow protections as well as those that do not. The TSS must provide a rationale for those binaries and libraries that are not protected in this manner.

Summary

Section 8.7.8 of the [ST]  states that a list of binaries has been provided to NIAP, since that list is considered proprietary information. However, all user space binaries are subject to ASLR.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.



2.1.6.4 Extended: Domain Isolation (FPT_AEX_EXT.4)

TSS Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.4-ASE-01

The evaluator shall ensure that the TSS describes the mechanisms that are in place that prevents non-TSF software from modifying the TSF software or TSF data that governs the behavior of the TSF. These mechanisms could range from hardware-based means (e.g. "execution rings" and memory management functionality); to software-based means (e.g. boundary checking of inputs to APIs). The evaluator determines that the described mechanisms appear reasonable to protect the TSF from modification.

Summary

Section 8.7.3 in the [ST]  describes *Secure Software Update*. Section 8.7.5 in the [ST]  describes *Domain Isolation* .

The entire OS partition is mounted as read-only which prevents any application or attacker from modifying the system. System updates are signed by Apple, along with applications which are signed by their developers with their Apple developer certificate, which prevents any non-TSF software from modifying the TSF software or TSF data.

Assurance Activity AA-FPT_AEX_EXT.4-ASE-02

The evaluator shall ensure the TSS describes how the TSF ensures that the address spaces of applications are kept separate from one another.

Summary

Section 8.7.5 in the [ST]  describes *Domain Isolation* .

Section 8.7.5 describes the domain isolation for the apps in iOS. Every application is executed in its own domain (sandbox) which isolates the application from other applications. Applications are restricted from accessing other applications' files.

Assurance Activity AA-FPT_AEX_EXT.4-ASE-03

The evaluator shall ensure the TSS details the USSD and MMI codes available from the dialer at the locked state or during auxiliary boot modes that may alter the behavior of the TSF. The evaluator shall ensure that this description includes the code, the action performed by the TSF, and a justification that the actions performed do not modify user or TSF data. If no USSD or MMI codes are available, the evaluator shall ensure that the TSS provides a description of the method by which actions prescribed by these codes are prevented.

Summary

Section 8.7.5 in the [ST]  describes *Domain Isolation* .

The TOE does not support Unstructured Supplementary Service Data (USSD) or Man-Machine Interface (MMI) codes and also does not support auxiliary boot modes.

Assurance Activity AA-FPT_AEX_EXT.4-ASE-04

The evaluator shall ensure the TSS documents any TSF data (including software, execution context, configuration information, and audit logs) which may be accessed and modified over a wired interface in auxiliary boot modes. The evaluator shall ensure that the description includes data, which is modified in support of update or restore of the device. The evaluator shall ensure that this documentation includes the auxiliary boot modes in which the data may be modified, the methods for entering the auxiliary boot modes, the location of the data, the manner in which data may be modified, the data format and packaging necessary to support modification, and software and/or hardware tools, if any, which are necessary for modifying the data.

Summary


Section 8.1 in the [ST]  describes the *Mapping to the Security Functional Requirements* .

The entry FPT_AEX_EXT.4 in *Table 5: Mapping of SFR Assurance Activities to the TSS* in section 8.1 states that iOS does not support auxiliary boot modes. This assurance activity is therefore not applicable.

Assurance Activity AA-FPT_AEX_EXT.4-ASE-05

The evaluator shall ensure that the TSS provides a description of the means by which unauthorized and undetected modification (that is, excluding cryptographically verified updates per FPT_TUD_EXT.2) of the TSF data over the wired interface in auxiliary boots modes is prevented. The lack of publicly available tools is not sufficient justification. Examples of sufficient justification include auditing of changes, cryptographic verification in the form of a digital signature or hash, disabling the auxiliary boot modes, and access control mechanisms that prevent writing to files or flashing partitions.

Summary

Section 8.7.1 in the [ST]  describes the *Secure Boot* process of iOS.

Section 8.7.1 describes that every step of the boot process contains components that are signed, and which signature is verified during boot. This includes the bootloaders, kernel, kernel extensions and baseband firmware. When the device is turned on, it immediately executes code from ROM known as Boot ROM. This code, the hardware root of trust, is implicitly trusted as it is laid down during manufacture. This Boot ROM contains Apple's Root CA public key, which is used to verify the integrity and authenticity of the iBoot bootloader. In case of any failure, the boot process is halted and the device displays the iTunes screen.

Guidance Assurance Activities

No assurance activities defined.





Test Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.4-ATE-01

Assurance Activity Note: *The following tests require the vendor to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products. In addition, the vendor provides a list of files (e.g., system files, libraries, configuration files, audit logs) that make up the TSF data. This list could be organized by folders/directories (e.g., /usr/sbin, /etc), as well as individual files that may exist outside of the identified directories.*

- **Test 1:** *The evaluator shall create and load an app onto the Mobile Device. This app shall attempt to traverse over all file systems and report any locations to which data can be written or overwritten. The evaluator must ensure that none of these locations are part of the OS software, device drivers, system and security configuration files, key material, or another untrusted application's image/data. For example, it is acceptable for a trusted photo editor app to have access to the data created by the camera app, but a calculator application shall not have access to the pictures.*
- **Test 2:** *For each available auxiliary boot mode, the evaluator shall attempt to modify a TSF file of their choosing using the software and/or hardware tools described in the TSS. The evaluator shall verify that the modification fails.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

Test 1: the evaluator compiled and loaded the application Find-UI to the device, executed Find-UI, and verified that all locations marked as writable are not part of the OS (i.e. is only located within the applications home directory).

Test 2: the evaluator booted into iTunes/DFU mode from power off state (up to and including iPhone6S with physical home button: Power + Home button + USB cable connection; iPhone7: USB cable connection, press power, after 3 seconds also press Volume down and keep both buttons pressed; starting with iPhone8: USB cable connection, Volume up, Volume down, hold power button), and verified that no service is available to modify files.

2.1.6.5 Extended: JTAG Disablement (FPT_JTA_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FPT_JTA_EXT.1.1-ASE-01

If “disable access through hardware” is selected:

- The evaluator shall examine the TSS to determine the location of the JTAG ports on the TSF, to include the order of the ports (i.e., Data In, Data Out, Clock, etc.).

If “control access by a signing key” is selected:

- The evaluator shall examine the TSS to determine how access to the JTAG is controlled by a signing key.

Summary

FPT_JTA_EXT.1.1(MDF) has selected "disable access through hardware."

Section 8.7.2 in the [ST] [describes the Joint Test Action Group \(JTAG\) Disablement](#).

Section 8.7.2 explains that typical JTAG ports (i.e., Data In, Data Out, Clock, etc.) are not present for physical access on iOS devices. This work-unit is therefore not applicable and considered satisfied.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_JTA_EXT.1.1-ATE-01

If “disable access through hardware” is selected the evaluator shall perform the following test:

- The evaluator shall connect a packet analyzer to the JTAG ports. The evaluator shall query the JTAG port for its device ID and confirm that the device ID cannot be retrieved.

If “control access by a signing key” is selected the evaluator shall perform the following test:

- The evaluator shall test with an application that is not approved to access the JTAG verifying that access cannot be achieved.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [the extended package \[EP_MDM_AGENT_V3.0\]](#) [the extended package \[PP_WLAN_CLI_EP_V1.0\]](#) [and the PP module \[MOD_VPN_CLI_V2.1\]](#).

The mobile devices of iPhones and iPads use a 2-staged interface that resembles the functionality of JTAG but does not implement the JTAG protocol.

The Apple development environment that is JTAG-like (technically it is a serial link and no JTAG) is based on the following: To use this JTAG-like interface, a development-fused device is required. This implies that certain hardware fuses were not blown during the manufacturing process. Only with these development-interface related fuses intact, the JTAG-like interface is technically reachable.

When having a development-fused device, the Apple developers are given a special Lightning cable that contains some additional computing logic. This cable establishes a serial channel with the mobile device's JTAG-like interface reachable on development-fused devices. This Lightning cable connects to the development machine's USB port and allows subsequent access by development tools. The serial link allows access to the serial console of the mobile device. The serial console, however, does not allow access on a production fused device. On a development-fused device, the root account is enabled and an SSH server is listening. The SSH server is accessible via the serial link and allows the developer to access the root account for development including uploading of software or modifying of installed software.

The evaluator performed the following verification steps to ensure that the JTAG-like interface is not reachable:

- The evaluator used the special developer's Lightning cable and connected it to the production-fused test devices. Verify that the JTAG-like interface was inaccessible.
- The evaluator used the same special Lightning cable to connect to the development-fused devices. Verify that the JTAG-like connection is possible. This testing was done in conjunction with FPT_AEX_EXT.1 and FPT_AEX_EXT.4 test 2.

2.1.6.6 Extended: Key Storage (FPT_KST_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FPT_KST_EXT.1-ASE-01

The evaluator shall consult the TSS section of the ST in performing the assurance activities for this requirement.


In performing their review, the evaluator shall determine that the TSS contains a description of the activities that happen on power-up and password authentication relating to the decryption of DEKs, stored keys, and data.

The evaluator shall ensure that the description also covers how the cryptographic functions in the FCS requirements are being used to perform the encryption functions, including how the KEKs, DEKs, and stored keys are unwrapped, saved, and used by the TOE so as to prevent plaintext from being written to non-volatile storage. The evaluator shall ensure that the TSS describes, for each power-down scenario how the TOE ensures that all keys in non-volatile storage are not stored in plaintext.

The evaluator shall ensure that the TSS describes how other functions available in the system (e.g., regeneration of the keys) ensure that no unencrypted key material is present in persistent storage.

The evaluator shall review the TSS to determine that it makes a case that key material is not written unencrypted to the persistent storage.

Summary

Section 8.3.1 in the [ST]  describes the *Overview of Key Management* in iOS.

The evaluator verified that section 8.3.1 describes how the REK, KEKs and DEKs interact by hierarchically wrapping each other, and how these keys are generated or formed upon boot (or from the authentication of the user to iOS by providing the correct passcode). Section 8.2.1 also describes which keys are wrapped, by which algorithm (namely AES in Key Wrap cipher mode), and which keys are stored in plaintext, encrypted, and in which type of memory (flash, non-volatile, etc.). The evaluator verified that the only key material that is written unencrypted in iOS is the REK (which is not accessible to any part of the system except the Secure Enclave), and the 0x89B and 0x835 keys. The latter are, however, stored in block 0 of the flash memory and can be erased very quickly if necessary. These keys are created during boot time and destroyed when the device is powered-off. When the device is locked, the passcode key is erased and can only be regenerated when the user provides the correct password.

Assurance Activity AA-FPT_KST_EXT.1-ASE-02

For each BAF selected in FIA_UAU.5.1:

The evaluator shall determine that the TSS also contains a description of the activities that happen on biometric authentication, relating to the decryption of DEKs, stored keys, and data. In addition how the system ensures that the biometric keying material is not stored unencrypted in persistent storage.

Summary

Section 8.5.1 describes the events during biometric authentication. Section 8.3.1 in the ST specifies the key management operations in iOS, and how KEK and other keys are derived and unwrapped based on the passcode used to unlock the device. The device passcode and biometrics credentials are not stored on the device (according to section 8.5) --neither encrypted or unencrypted. The key derivation, combined with the salt and the UID, derives the passcode key used to unencrypt keys used to encrypt files on the system.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.6.7 Extended: No Key Transmission (FPT_KST_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FPT_KST_EXT.2-ASE-01

The evaluator shall determine that the TSS also contains a description of the activities that happen on biometric authentication, relating to the decryption of DEKs, stored keys, and data. In addition how the system ensures that the biometric keying material is not stored unencrypted in persistent storage.

In performing their review, the evaluator shall determine that the TSS contains a description of the activities that happen on power-up and password authentication relating to the decryption of DEKs, stored keys, and data.

The evaluator shall ensure that the TSS describes how other functions available in the system (e.g., regeneration of the keys) ensure that no unencrypted key material is transmitted outside the security boundary of the TOE.

The evaluator shall review the TSS to determine that it makes a case that key material is not transmitted outside the security boundary of the TOE.

Summary

Section 8.3.1 in the [ST] describes the *Overview of Key Management* in iOS.

The TOE is the mobile device running iOS as well as iOS itself. The cryptographic boundary is therefore the entire mobile device. The evaluator found in section 8.3.1 a description of the power-up process and password authentication within iOS relating to the generation, derivation, and decryption of DEKs, DEKs themselves, and stored data. Whenever a file is accessed, it gets decrypted by its respective class key. The file is then decrypted with its respective file key. Whenever a file is closed, its file key is stored encrypted in memory. At any time, no key is sent outside the security boundary of the TOE (i.e., outside the device).

Assurance Activity AA-FPT_KST_EXT.2-ASE-02

For each BAF selected in FIA_UAU.5.1:

n performing their review, the evaluator shall determine that the TSS contains a description of the activities that happen on biometric authentication, including how any plaintext material, including critical security parameters and results of biometric algorithms, are protected and accessed.

The evaluator shall ensure that the TSS describes how functions available in the biometric algorithms ensure that no unencrypted plaintext material, including critical security parameters and intermediate results, is transmitted outside the security boundary of the TOE or to other functions or systems that transmit information outside the security boundary of the TOE.

Summary

Section 8.5 of the ST describes *Identification and Authentication (FIA)*. Section 8.6.3 in the ST describes the *Biometric Authentication Factors (BAFs)* .

The evaluator found that authentication credentials are not stored in the TOE in any form. When the user authenticates successfully, the key derivation output decrypts the keybag in the Secure Enclave Processor with the respective key class. No intermediate results are sent outside the TOE either; the key derivation process happens locally only.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.6.8 Extended: No Plaintext Key Export (FPT_KST_EXT.3)

TSS Assurance Activities

Assurance Activity AA-FPT_KST_EXT.3-ASE-01

The ST author will provide a statement of their policy for handling and protecting keys. The evaluator shall check to ensure the TSS describes a policy in line with not exporting either plaintext DEKs, KEKs, or keys stored in the secure key storage.

Summary

Section 8.2.1 in the [ST] describes *The Secure Enclave* .

The evaluator verified that the second paragraph in section 8.2.1 contains a statement of the policy defining that

"no keys or key material may be exported."

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.6.9 Extended: Self-Test Notification (FPT_NOT_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FPT_NOT_EXT.1-ASE-01

The evaluator shall verify that the TSS describes critical failures that may occur and the actions to be taken upon these critical failures.

Summary

Section 8.7.1 in the [ST] describes *Secure Boot*. Section 8.7.9 in the [ST] describes the *Self Tests*.

The evaluator reviewed sections 8.7.1 and 8.7.9 and verified that if the TOE encounters a failure of the self-tests the TOE will power itself off and will require a reboot. In case of a failure of the integrity check, the device will display the "Connect to iTunes" screen and transition to the recovery mode. In this mode, the device has to be plugged into a USB connection, connected to iTunes and factory reset.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

See assurance activities for SFR elements below.

FPT_NOT_EXT.1.1

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_NOT_EXT.1.1-ATE-01

Assurance Activity Note: *The following test require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.*

- **Test 1:** *The evaluator shall use a tool provided by the developer to modify files and processes in the system that correspond to critical failures specified in the second list. The evaluator shall verify that creating these critical failures causes the device to take the remediation actions specified in the first list.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: FIPS 140-2 integrity and power-up test was tested with TOE during the FIPS 140-2 validation by the same tester performing this test. Therefore, it is not tested again as agreed with NIAP and the validators.

2.1.6.10 Reliable Time Stamps (FPT_STM.1)

TSS Assurance Activities

Assurance Activity AA-FPT_STM.1-ASE-01

The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time. The TSS provides a description of how the time is maintained and considered reliable in the context of each of the time related functions. This documentation must identify whether the TSF uses a NTP server or the carrier's network time as the primary time sources.

Summary

Section 8.7.7 in the [ST][\[1\]](#) describes the *Time*.

Section 8.7.7 defines which security functions in the TOE make usage of time as follows.

- ALC_TSU_EXT
- FAU_GEN.1.2
- FIA_UAU.7
- FIA_X509_EXT.1.1
- FIA_X509_EXT.3.1
- FMT_SMF_EXT.1.1 Function 1
- FMT_SMF_EXT.1.1 Function 2
- FPT_STM.1.1
- FTA_SSL_EXT.1
- For A7 and later platforms: GPS or NTP (if GPS is not available)
- Network, Identity and Time Zone (NITZ)

The TOE will first use GPS for time, then (Apple) NTP or the cellular carrier time service if the "Set Automatically" option is set on the device.

Guidance Assurance Activities

Assurance Activity AA-FPT_STM.1-AGD-01

The evaluator examines the operational guidance to ensure it describes how to set the time.

Summary

[CCGUIDE][\[1\]](#) section 3.6.4 *Timestamp Configuration* provides guidance for setting up the date and time. In the evaluated configuration, the TOE must be configured for automatic time update using NTP as the time source. To do this, the administrator uses a Configuration Profile with the Time Server Payload and *timeServer* and *timeZone* keys.

Test Assurance Activities

Assurance Activity AA-FPT_STM.1-ATE-01

- **Test 1:** The evaluator uses the operational guide to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: (setting of time is only possible when NTP is disabled) the evaluator disabled NTP, changed the time, and verified that the new time is displayed at the top of the screen.

2.1.6.11 Extended: TSF Cryptographic Functionality Testing (FPT_TST_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FPT_TST_EXT.1-ASE-01

The evaluator shall examine the TSS to ensure that it specifies the self-tests that are performed at start-up. This description must include an outline of the test procedures conducted by the TSF (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 TSS must include any error states that they TSF may enter when self-tests fail, and the conditions and actions necessary to exit the error states and resume normal operation. The evaluator shall verify that the TSS indicates these self-tests are run at start-up automatically, and do not involve any inputs from or actions by the user or operator.

Summary

Section 8.7.9 in the [ST] describes the Self-Tests .

Section 8.7.9 describes which self-tests are run by iOS when the device is powered-up. The TOE performs the FIPS 140-2 power-on self-tests for its cryptographic algorithms, along with a software integrity test using HMAC-SHA-256. More information about the self-tests can be found in sections 8.7.9.1 through 8.7.9.3 in the [ST] .

Assurance Activity AA-FPT_TST_EXT.1-ASE-02

The evaluator shall inspect the list of self-tests in the TSS and verify that it includes algorithm self-tests. The algorithm self-tests will typically be conducted using known answer tests.

Summary

Sections 8.7.9.1 and 8.7.9.3 in the [ST] describes the self-tests.

These sections describe that the TOE performs self-tests for its cryptographic algorithms using the known answer test method. The following algorithms are tested when the TOE is powered-on: AES, DRBG, (HMAC-)SHA-1, (HMAC-)SHA-256, (HMAC-)SHA-384, (HMAC-)SHA-512, ECDSA, RSA, DH and ECDH.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.1.6.12 Extended: TSF Integrity Testing (FPT_TST_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FPT_TST_EXT.2-ASE-01

The evaluator shall verify that the TSS section of the ST includes a description of the boot procedures, including a description of the entire bootchain, of the software for the TSF's Application Processor. The evaluator shall ensure that before loading the bootloader(s) for the operating system and the kernel, all bootloaders and the kernel software itself is cryptographically verified. For each additional category of executable code verified before execution, the evaluator shall verify that the description in the TSS describes how that software is cryptographically verified.

Summary

Section 8.7.1 in the [ST] describes *Secure Boot* .

Section 8.7.1 includes a description of the boot process, and how every component is signed and its signature is verified during boot time. This includes the kernel, kernel extensions, the bootloaders and baseband firmware.

Assurance Activity AA-FPT_TST_EXT.2-ASE-02

The evaluator shall verify that the TSS contains a justification for the protection of the cryptographic key or hash, preventing it from being modified by unverified or unauthenticated software. The evaluator shall verify that the TSS contains a description of the protection afforded to the mechanism performing the cryptographic verification.

Summary

Section 8.7.1 in the [ST] describes the *Secure Boot* .

Section 8.7.1 describes that the baseband firmware's integrity is verified at boot by a digital signature algorithm. The asymmetric key used for this verification is protected in hardware.

Assurance Activity AA-FPT_TST_EXT.2-ASE-03

The evaluator shall verify that the TSS describes each auxiliary boot mode available on the TOE during the boot procedures. The evaluator shall verify that, for each auxiliary boot mode, a description of the cryptographic integrity of the executed code through the kernel is verified before each execution.

Summary

No auxiliary boot mode is claimed, and this assurance activity is therefore not applicable.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TST_EXT.2-ATE-01

The evaluator shall perform the following tests:

- **Test 1:** The evaluator shall perform actions to cause TSF software to load and observe that the integrity mechanism does not flag any executables as containing integrity errors and that the TOE properly boots.

Assurance Activity Note: The following tests require the vendor to provide access to a test platform that provides the evaluator with tools that are typically not found on consumer Mobile Device products.

- **Test 2:** The evaluator shall modify a TSF executable that is integrity protected and cause that executable to be successfully loaded by the TSF. The evaluator observes that an integrity violation is triggered and the TOE does not boot. (Care must be taken so that the integrity violation is determined to be the cause of the failure to load the module, and not the fact that the module was modified so that it was rendered unable to run because its format was corrupt).
- **Test 3:** [conditional] If the ST author indicates that the integrity verification is performed using a public key, the evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1. The evaluator shall digitally sign the TSF executable with a certificate that does not have the Code Signing purpose in the extendedKeyUsage field and verify that an integrity violation is triggered. The evaluator shall repeat the test using a certificate that contains the Code Signing purpose and verify that the integrity verification succeeds. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator rebooted the TOE and verified that reboot was successful.

Test 2: FIPS 140-2 integrity and power-up test was tested with TOE during the FIPS 140-2 validation by the same tester performing this test. Therefore, it is not tested again.

Test 3: not applicable.

2.1.6.13 Extended: Trusted Update: TSF Version Query (FPT_TUD_EXT.1)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.1-ATE-01

The evaluator shall establish a test environment consisting of the Mobile Device and any supporting software that demonstrates usage of the management functions. This can be test software from the developer, a reference implementation of management software from the developer, or other commercially available software. The evaluator shall set up the Mobile Device and the other software to exercise the management functions according to the provided guidance documentation.

- **Test 1:** Using the AGD guidance provided, the evaluator shall test that the administrator and user can query:
 - the current version of the TSF operating system and any firmware that can be updated separately
 - the hardware model of the TSF
 - the current version of all installed mobile applications

The evaluator must review manufacturer documentation to ensure that the hardware model identifier is sufficient to identify the hardware which comprises the device.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator could query the version of iOS and the hardware model of the device in the Settings, and the version of all installed apps through the App Store (select "Purchased" to display the installed apps and their versions).

2.1.6.14 Extended: Trusted Update Verification (FPT_TUD_EXT.2)

FPT_TUD_EXT.2.3

TSS Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.2.3-ASE-01

The evaluator shall verify that the TSS section of the ST describes all TSF software update mechanisms for updating the system software. The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. The evaluator shall verify that all software and firmware involved in updating the TSF is described and, if multiple stages and software are indicated, that the software/firmware responsible for each stage is indicated and that the stage(s) which perform signature verification of the update are identified.

Summary

Section 8.7.3 in the [ST] describes *Secure Software Updates*.

Section 8.7.3 describes the secure software updates provided by iOS. Whenever updates are available, users on iOS receive a notification that an update is available. All updates are digitally signed and users cannot downgrade to older software versions. These updates happen over a secure channel protected using the HTTPS protocol. The boot-time chain-of-trust makes sure the signature of the code comes from Apple. Paragraph three of this section explains that the installation is only permitted if the verification process succeeds. The authorization server checks the presented list of measurements against versions for which installation is permitted. If the server finds a match, it adds the device unique ID to the measurement and signs the result. The server then passes a complete set of signed data to the device, which will then verify the signature and install the upgrades if the signature verification is successful.

Assurance Activity AA-FPT_TUD_EXT.2.3-ASE-02

The evaluator shall verify that the TSS describes the method by which the digital signature is verified and that the public key used to verify the signature is either hardware-protected or is validated to chain to a public key in the Trust Anchor Database. If hardware-protection is selected, the evaluator shall verify that the method of hardware-protection is described and that the ST author has justified why the public key may not be modified by unauthorized parties.

Summary

Section 8.7.3 in the [ST] describes the *Secure Software Updates*.

The software updates are digitally signed and verified using a hardware-protected asymmetric key used for signature verification. That asymmetric key is protected as a x509v3 certificate in the Secure Enclave (Apple's root certificate).

Assurance Activity AA-FPT_TUD_EXT.2.3-ASE-03

[conditional] If the ST author indicates that software updates to system software running on other processors is verified, the evaluator shall verify that these other processors are listed in the TSS and that the description includes the software update mechanism for these processors, if different than the update mechanism for the software executing on the Application Processor.

Summary

This selection is not included in the [ST] by the ST author.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.2.3-ATE-01

[conditional] If the ST author indicates that the public key is used for software update digital signature verification, the evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1 and a check for the Code Signing purpose in the extendedKeyUsage.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator fetched a valid, but unsigned iBEC image (2nd boot loader that is executed after the ROM boot loader is started), tried to load the image onto the device and observe that the image is rejected by the boot loader. The positive test is to verify that properly signed images are booted is simply to reboot the device with an official iOS image and verify that the boot is successful.

Test 2: the evaluator signed the boot loader binary with a development RSA key, verified that that the newly generated key is listed in the manifest file accompanying the iBEC image, tried to load the image onto the device and observe that the image is rejected by the boot loader, used an iBEC image that was signed with Apple key and load it onto the device, verified that the image load succeeds.

Test 3: The test specified in the PP is not applicable. Apple uses a CA that is dedicated for software signing. The CA and its keys is not used for any other purpose. The root certificate is hard coded into the ROM code (validated by the evaluator's code inspection). The certificate path validation verifies that the software certificate used to sign an image traces back to the root certificate. Therefore, the verification of the certificate extension is not required as all certificates that are deemed valid based on the certificate chain validation are only used for software signatures as defined by Apple policy.

Test 4: See tests for FPT_TUD_EXT.2.4.

Test 5: not applicable.

Assurance Activity AA-FPT_TUD_EXT.2.3-ATE-02

The evaluator shall verify that the developer has provided evidence that the following tests were performed for each available update mechanism:

- **Test 1:** The tester shall try to install an update without the digital signature and shall verify that installation fails. The tester shall attempt to install an update with digital signature, and verify that installation succeeds.

- **Test 2:** The tester shall digitally sign the update with a key disallowed by the device and verify that installation fails. The tester shall digitally sign the update with the allowed key and verify that installation succeeds.
- **Test 3:** [conditional] The tester shall digitally sign the update with an invalid certificate and verify that update installation fails. The tester shall repeat the test using a valid certificate and a certificate that contains the purpose and verify that the update installation succeeds.
- **Test 4:** [conditional] The tester shall digitally sign the application with a certificate that does not have the Code Signing purpose and verify that application installation fails. The tester shall repeat the test using a valid certificate and a certificate that contains the Code Signing purpose and verify that the application installation succeeds.
- **Test 5:** [conditional] The tester shall repeat this test for the software executing on each processor listed in the first selection. The tester shall attempt to install an update without the digital signature and shall verify that installation fails. The tester shall attempt to install an update with digital signature, and verify that installation succeeds.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 5: not applicable.

FPT_TUD_EXT.2.4

TSS Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.2.4-ASE-01

The evaluator shall verify that the TSS describes how mobile application software is verified at installation. The evaluator shall ensure that this method uses a digital signature.

Summary

Section 8.5.2 in the [ST] describes the *Certificates*.

Applications are signed and their signature is verified before installation by an Apple certificate. If the signature verification fails, the application is not installed:

" Code signing certificates need to be assigned by Apple and can be imported into a device. The issue of such a certificate can be by app developers or by enterprises that want to deploy apps from their MDM to managed devices. All apps must have a valid signature that can be verified by a code signing certificate before they are installed on a device. "

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.2.4-ATE-01

Assurance Activity Note: The following test does not have to be tested using the commercial application store.

- **Test 1:** The evaluator shall write, or the developer shall provide access to, an application. The evaluator shall try to install this application without a digital signature and shall verify that installation fails. The evaluator shall attempt to install a digitally signed application, and verify that installation succeeds.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: Tooling does not allow to create unsigned applications. Installation of validly signed app is tested with Test 2.

Test 2: the evaluator generated an IPA package, verified the correct signature and generated a new certificate using CodeSigning purpose using KeyChain. The evaluator located the CryptoTest binary, and invoked a command to sign the application. The evaluator then zipped it back together and pushed the application using Xcode on the TOE (it succeeds). The evaluator removed the application from the TOE and pushed the modified CryptoTest app on the TOE, and verified that the app is not usable.

Test 3: not applicable, no certificates can be installed/limited that authorize apps. Only the Apple Root CA is allowed.

2.1.6.15 TSF Self Test (VPN) (FPT_TST_EXT.1(VPN))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FPT_TST_EXT.1-VPN-AGD-01

If not present in the TSS, the evaluator ensures that the operational guidance describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases. For checks implemented entirely by the platform, the evaluator ensures that the operational guidance for the TOE references or includes the platform-specific guidance for each platform listed in the ST.

Summary

Description of self-tests is provided in the TSS of [ST] in section 8.7.9 *Self-Tests*. According to this section, self-tests are performed by the three cryptographic modules of the TOE as outlined below.

Apple iOS CoreCrypto Module Cryptographic Module for ARM

This cryptographic module performs self-tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start up. In addition, the random bit generator requires continuous verification. The FIPS Self-Tests application which is invoked by the iOS startup process upon device power on, runs all required module self-tests.

Self-tests performed by this module includes:

- Power-up tests: these tests (as listed in Table 11 of [ST]) are performed each time the module starts and must be completed successfully for the module to operate in the FIPS approved mode. If any of the required tests fail, the device powers itself off. To rerun the self-tests on demand, the user must reboot the device.
- Software / Firmware integrity tests: A software integrity test is performed on the runtime image of the Apple iOS CoreCrypto Cryptographic Module for ARM. The CoreCrypto's HMAC-SHA-256 is used as an approved algorithm for the integrity test. If the test fails, then the device powers itself off.
- Conditional tests covering pair-wise consistency tests and SP 800-90A assurance tests.

Apple iOS CoreCrypto Kernel Cryptographic Module for ARM

This cryptographic module performs self-tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start up. In addition, the DRBG requires continuous verification. The FIPS Self-Tests functionality, which is invoked by the iOS Kernel startup process upon device initialization, runs all required module self-tests. If the self-tests succeed, the module's instance is maintained in the memory of the iOS Kernel on the device and made available to each calling kernel service without reloading.

Self-tests performed by this module includes:

- Power-up tests: these tests are performed each time the module starts and must be completed successfully for the module to operate in the FIPS approved mode. If any of the following tests fails the device shuts down automatically. To run the self-tests on demand, the user may reboot the device.
- Cryptographic algorithm tests specified in Table 12 of [ST] [\[ST\]](#).
- Software / Firmware integrity test: this test is performed on the runtime image of the module. The CoreCrypto's HMAC-SHA256 is used as an approved algorithm for the integrity test. If the test fails, then the device powers itself off.
- Conditional tests covering continuous RNG test, pair-wise consistency test, and SP800-90 assurance test

Apple Secure Key Store Cryptographic Module

This module runs all required module self-tests pertaining to the firmware. These self-tests are invoked automatically when starting the module. Also during startup of the hardware, the hardware DRBG invokes its independent self-test. If the self-test fails in either the firmware or hardware DRBG the device immediately shuts down to prevent any operation.

Self-tests performed by this module includes:

- Power-up tests: these tests are performed each time the TOE devices start and must be completed successfully for the module to operate in FIPS approved mode. If any of the following tests fails the device powers itself off. To run the self-tests on demand, the user must reboot the device.
- Cryptographic algorithm tests specified in Table 13 of [ST] [\[ST\]](#).
- Software / Firmware integrity tests: this test is performed on the runtime image of the TOE devices. The module's HMAC-SHA-256 is used as an FIPS approved algorithm for the integrity test. If the test fails, then the device powers itself off.
- Conditional tests covering pair-wise consistency test and SP800-90 assurance test

Test Assurance Activities

Assurance Activity AA-FPT_TST_EXT.1-VPN-ATE-01

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 2:** *The evaluator modifies the TSF executable, performs the integrity check on the modified TSF executable and verifies that the check fails.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[PP_MD_V3.1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[EP_MDM_AGENT_V3.0\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[PP_WLAN_CLI_EP_V1.0\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[MOD_VPN_CLI_V2.1\]](#).

FIPS 140-2 integrity and power-up tests were tested with TOE during the FIPS 140-2 validation by the same tester performing this test. This test demonstrated that the integrity check of a known good application succeeds. Therefore, it is not tested again as agreed with NIAP and the validators. The ST states that the self test applies to the cryptographic support offered to the VPN component of the TOE. This cryptographic support is provided with CoreCrypto executing in user space (IKE) and XNU kernel space (IPsec) that is subject to the mentioned FIPS 140-2 validation.

2.1.6.16 TSF Cryptographic Functionality Testing (Wireless LAN) (FPT_TST_EXT.1(WLAN))

TSS Assurance Activities

Assurance Activity AA-FPT_TST_EXT.1-WLAN-ASE-01

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

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

Summary

Section 8.7.9, *Self-Tests*, in the [ST] describes the self-tests performed by the TOE. Section 8.7.1 of the [ST] describes the *Secure Boot* process.

Section 8.7.9 is split into several sections defining all the different self-tests implemented by the TOE. This includes the power-up test, cryptographic algorithms tests (know-answer-tests or pair-wise consistency test), software/firmware integrity tests, critical function tests, and conditional tests. These self-tests are performed both for the CoreCrypto Kernel Module for ARM (kernel space), for the CoreCrypto Module for ARM (user space), and for the Apple Secure Key Store. These sections clearly explain how the integrity of the libraries (libcorecrypto.dylib, libSystem.dylib and libcommoncrypto.dylib and the runtime image of the iOS kernel) is checked using the HMAC-SHA-256 algorithm. These sections also describe that if any of these tests should fail, the device shuts down automatically. Section 8.7.1 explains that if one step of the boot process, including the integrity verification of the stored TSF executable, fails, then the system does not boot and displays the "Connect to iTunes" screen, for device recovery. The evaluator verified that section 8.7.9.1 of the TSS provides an argument that the tests are sufficient:

" iOS ensures that there is only one physical instance of the library and maps it to all application linking to that library. In this way, the module always stays in memory. Therefore, the self-test during startup time is sufficient as it tests the module instance loaded in memory which is subsequently used by every application on iOS."

Guidance Assurance Activities

Assurance Activity AA-FPT_TST_EXT.1-WLAN-AGD-01

The evaluator shall ensure 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.

Summary

Description of self-tests is provided in the TSS of [ST] in section 8.7.9 *Self-Tests*. According to this section, self-tests are performed by the three cryptographic modules of the TOE as outlined below.

Apple iOS CoreCrypto Module Cryptographic Module for ARM

This cryptographic module performs self-tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start up. In addition, the random bit generator requires continuous verification. The FIPS Self-Tests application which is invoked by the iOS startup process upon device power on, runs all required module self-tests.

Self-tests performed by this module includes:

- Power-up tests: these tests (as listed in Table 11 of [ST]) are performed each time the module starts and must be completed successfully for the module to operate in the FIPS approved mode. If any of the required tests fail, the device powers itself off. To rerun the self-tests on demand, the user must reboot the device.
- Software / Firmware integrity tests: A software integrity test is performed on the runtime image of the Apple iOS CoreCrypto Cryptographic Module for ARM. The CoreCrypto's HMAC-SHA-256 is used as an approved algorithm for the integrity test. If the test fails, then the device powers itself off.
- Conditional tests covering pair-wise consistency tests and SP 800-90A assurance tests.

Apple iOS CoreCrypto Kernel Cryptographic Module for ARM

This cryptographic module performs self-tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start up. In addition, the DRBG requires continuous verification. The FIPS Self-Tests functionality, which is invoked by the iOS Kernel startup process upon device initialization, runs all required module self-tests. If the self-tests succeed, the module's instance is maintained in the memory of the iOS Kernel on the device and made available to each calling kernel service without reloading.

Self-tests performed by this module includes:

- Power-up tests: these tests are performed each time the module starts and must be completed successfully for the module to operate in the FIPS approved mode. If any of the following tests fails the device shuts down automatically. To run the self-tests on demand, the user may reboot the device.
- Cryptographic algorithm tests specified in Table 12 of [ST].
- Software / Firmware integrity test: this test is performed on the runtime image of the module. The CoreCrypto's HMAC-SHA256 is used as an approved algorithm for the integrity test. If the test fails, then the device powers itself off.
- Conditional tests covering continuous RNG test, pair-wise consistency test, and SP800-90 assurance test

Apple Secure Key Store Cryptographic Module

This module runs all required module self-tests pertaining to the firmware. These self-tests are invoked automatically when starting the module. Also during startup of the hardware, the hardware DRBG invokes its independent self-test. If the self-test fails in either the firmware or hardware DRBG the device immediately shuts down to prevent any operation.

Self-tests performed by this module includes:

- Power-up tests: these tests are performed each time the TOE devices start and must be completed successfully for the module to operate in FIPS approved mode. If any of the following tests fails the device powers itself off. To run the self-tests on demand, the user must reboot the device.
- Cryptographic algorithm tests specified in Table 13 of [ST].
- Software / Firmware integrity tests: this test is performed on the runtime image of the TOE devices. The module's HMAC-SHA-2565 is used as an FIPS approved algorithm for the integrity test. If the test fails, then the device powers itself off.
- Conditional tests covering pair-wise consistency test and SP800-90 assurance test

Test Assurance Activities

Assurance Activity AA-FPT_TST_EXT.1-WLAN-ATE-01

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 2: The evaluator modifies the TSF executable, performs the integrity check on the modified TSF executable and verifies that the check fails.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1 and Test 2 are covered by the FIPS validation of the TOE concurrent to this CC evaluation, tested by the same evaluator.

2.1.7 TOE access (FTA)

2.1.7.1 Extended: TSF- and User-initiated Locked State (FTA_SSL_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FTA_SSL_EXT.1-ASE-01

The evaluator shall verify the TSS describes the actions performed upon transitioning to the locked state.

Summary


Section 8.7.6 in the [ST] describes *Device Locking*.

When the TOE transitions into a locked state, the class keys 'Complete Protection' and 'Accessible when unlocked' are wiped 10 seconds after the device is wiped, making all files in that class inaccessible.

Assurance Activity AA-FTA_SSL_EXT.1-ASE-02

The evaluator shall verify that the TSS describes the information allowed to be displayed to unauthorized users.

Summary

Section 8.8.3 of the [ST]  describes the *Lock Screen / Access Banner Display* .


Unauthorized users are only capable of viewing the displayed banner on the device, making emergency calls, answering calls, using the flashlight and using the camera.

Guidance Assurance Activities

Assurance Activity AA-FTA_SSL_EXT.1-AGD-01

The evaluation shall verify that the AGD guidance describes the method of setting the inactivity interval and of commanding a lock.

Summary





[CCGUIDE]  section 3.6.3 *Device/Session Locking* provides related guidance. It states that the TOE device is locked after a configurable time of user inactivity or upon request of the user. This can be defined by an administrator using a Configuration Profile by setting the configuration key *maxInactivity* in the Passcode Policy Payload to the desired time.

Test Assurance Activities

Assurance Activity AA-FTA_SSL_EXT.1-ATE-01

- **Test 1:** The evaluator shall configure the TSF to transition to the locked state after a time of inactivity (FMT_SMF_EXT.1) according to the AGD guidance. The evaluator shall wait until the TSF locks and verify that the display is cleared or overwritten and that the only actions allowed in the locked state are unlocking the session and those actions specified in FIA_UAU_EXT.2.
- **Test 2:** The evaluator shall command the TSF to transition to the locked state according to the AGD guidance as both the user and the administrator. The evaluator shall wait until the TSF locks and verify that the display is cleared or overwritten and that the only actions allowed in the locked state are unlocking the session and those actions specified in FIA_UAU_EXT.2.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] , the extended package [EP_MDM_AGENT_V3.0] , the extended package [PP_WLAN_CLI_EP_V1.0] , and the PP module [MOD_VPN_CLI_V2.1] .

Test 1: the evaluator waited for the device to lock automatically, verified that the screen is overwritten with non-sensitive data. The evaluator verified that only the unlock is possible, or the use of camera, information center, display of information and control center.

Test 2: the evaluator re-performed Test 1, but locked the device manually instead. The same conclusion holds.

2.1.7.2 Default TOE Access Banners (FTA_TAB.1)

TSS Assurance Activities

Assurance Activity AA-FTA_TAB.1-ASE-01

The TSS shall describe when the banner is displayed.

Summary

Section 8.8.3 of the [ST]  describes the *Lock Screen / Access Banner Display* .

A banner can be set up by the device and displayed when the device is locked (displayed on the lock screen).

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FTA_TAB.1-ATE-01

The evaluator shall also perform the following test:

- **Test 1:** *The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall then start up or unlock the TSF. The evaluator shall verify that the notice and consent warning message is displayed in each instance described in the TSS.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[4\]](#).

Test 1: the evaluator changed the wallpaper on the locked screen which is intended to display the banner. The evaluator re-performed that action once more and verified that a different wallpaper is displayed.

2.1.7.3 Wireless Network Access (FTA_WSE_EXT.1(WLAN))

TSS Assurance Activities

Assurance Activity AA-FTA_WSE_EXT.1-WLAN-ASE-01

The evaluator shall examine the TSS to determine that all of the attributes that can be used to specify acceptable networks (access points) are specifically defined.

Summary

Section 8.9.1, *EAP-TLS and TLS*, in the [ST][\[1\]](#) describes the attributes that can be used to specify acceptable networks.

The evaluator examined section 8.9.1 and verified that the fifth to last paragraph in this section describes that when an application attempts to establish a trusted channel, the TOE will compare the DN contained within the peer certificate (CN, Subject Alternative Name fields, IP address or wild-cards) to the DN of the requested server. If the DN does not match the expected DN for the peer, the application cannot establish the connection.

Guidance Assurance Activities

Assurance Activity AA-FTA_WSE_EXT.1-WLAN-AGD-01

The evaluator shall examine the operational guidance to determine that it contains guidance for configuring each of the attributes identified in the TSS.

Summary

[CCGUIDE] [\[1\]](#) section 3.6.7 *Enable/Disable Cellular, Wi-Fi Hotspot, Bluetooth, NFC* provides related guidance for Wi-Fi configuration. It states that the TOE devices contain a variety of radios which can be configured by the users or administrators according to the organization's policy. Additionally, this section refers to [iPhone_UG] [\[1\]](#) and [iPad_UG] [\[1\]](#) for instructions for enabling/ disabling cellular, Bluetooth, Wi-Fi, Wi-Fi hotspot.

Test Assurance Activities

Assurance Activity AA-FTA_WSE_EXT.1-WLAN-ATE-01

The evaluator shall also perform the following test for each attribute:

- *Test 1: The evaluator configures the TOE to allow a connection with a specific access point. The evaluator also configures the test environment such that the allowed access point and an access point that is not allowed are both "visible" to the TOE. The evaluator shall demonstrate that they can successfully establish a session with the allowed access point. The evaluator will then attempt to establish a session with the disallowed access point, and observe that the access attempt fails.*
- *Test 2: The evaluator configures the TOE to allow a connection with a specific access point using EAP-TLS authentication (not only will the valid SSID be configured but the TOE will also be provided with certificates to complete the EAP-TLS authentication). The evaluator also configures the test environment such that an access point broadcasts the SSID the TOE has been configured to connect to but the authentication server does not have valid credentials. The evaluator will then attempt to establish a session with the valid SSID/invalid authentication server, and observe that the access attempt fails.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1] [\[1\]](#), the extended package [EP_MDM_AGENT_V3.0] [\[1\]](#), the extended package [PP_WLAN_CLI_EP_V1.0] [\[1\]](#), and the PP module [MOD_VPN_CLI_V2.1] [\[1\]](#).

Test 1: the evaluator deployed a configuration profile on the TOE that allows the TOE to only connect to WiFi networks installed by WiFi payloads. The evaluator could then connect the TOE to the test Access point (AP). The evaluator removed the profile and tried to force the TOE to connect to the same WiFi AP and verified that it failed.

Test 2: tested through FCS_TLSC_EXT.1 {WLAN}.

2.1.8 Trusted path/channels (FTP)

2.1.8.1 Trusted Channel Communication (FTP_ITC_EXT.1(2)(AGENT))

TSS Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-ASE-01

The evaluator shall examine the TSS to determine that it describes the details of the TOE connecting to access points, VPN Gateways, and other trusted IT products in terms of the cryptographic protocols specified in the requirement, along with TOE-specific options or procedures that might not be reflected in the specifications. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to access points, VPN Gateways, and other trusted IT products.

Summary


Section 8.9.4 in the [ST] [\[1\]](#) describes the VPN. It describes the AlwaysOn VPN, IPsec and IKE protocols. The ST specifies the specific mode and algorithms used by the IKEv2 and IPsec protocols in section 8.9.4.3 as follows.


"

- *IKEv2 (as defined in RFCs 7296 and 4307),*
- *Tunnel Mode,*
- *Symmetric algorithms for IKE and ESP encryption (AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256),*
- *Integrity mechanisms (HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512),*
- *Key Exchange (Diffie-Hellman Groups):*
 - *5(1536-bit MODP),*
 - *14(2048-bit MODP),*
 - *15(3072-bit MODP),*
 - *19(256-bit Random ECP), and*
 - *20 (384-bit Random ECP)*

"

The characteristics of IKE are specified in section 8.9.4.5.


Section 8.9.1 in the [ST]  describes *EAP-TLS and TLS*. The evaluator verified that this section specifies the versions of TLS supported by the TOE, along with the ciphersuites it supports. Certificate pinning is also supported by the TOE.

Section 8.9.2 in the [ST]  describes *Bluetooth* on the device. The TOE supports Bluetooth 4.0, 4.2 and 5.0. Additional specification about the Bluetooth protocol are specific in this section about how to set up and use the Bluetooth protocol.

Assurance Activity AA-FTP_ITC_EXT.1-ASE-02

If OTA updates are selected, the TSS shall describe which trusted channel protocol is initiated by the TOE and is used for updates.


Summary

Section 8.7.3 of the [ST]  describes that the OTA updates on iOS are performed using HTTPS via a trusted channel.

Assurance Activity AA-FTP_ITC_EXT.1-MDMA-MDF-ASE-01

The evaluator shall examine the TSS to determine that the methods of Agent-Server communication are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of remote TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Summary

Section 8.10.2, *MDM Agent Alerts*, in the [ST]  specifies that whenever the MDM Agent initiates a communication with the MDM Server it establishes a TLS connection to the MDM Server URL. The evaluator confirms that the TLS protocol is supported by the TOE and selected in FCS_TLSC_EXT.1.1.

Guidance Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-AGD-01

The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to access points, VPN Gateways, and other trusted IT products.

Summary

[CCGUIDE] [\[1\]](#) section 3.3 *Network Protocols* provided related guidance for configuring the following network protocols:

- EAP-TLS
- TLS
- IPsec
- Bluetooth
- VPN

In particular, [CCGUIDE] [\[1\]](#) section 3.3.5 *VPN Configuration* provides detailed guidance for VPN configuration. It states that in the evaluated configuration, the VPN must be in its Always-On configuration which is configured by the administrator using the VPN Policy Payload with the key *VPNTType* setting to 'AlwaysOn' and the key 'ProtocolType' to IKEv2. Section 3.3.6 *Keys for Configuring Network Protocols* outline in table 11 "Essential Keys for the VPN Payload" the additional keys and key values that must or must not be used in the evaluated configuration.

Guidance related to IPsec is provided in section 3.3.3 *IPsec Configuration*.

Assurance Activity AA-FTP_ITC_EXT.1-MDMA-MDF-AGD-01

The evaluator shall confirm that the operational guidance contains instructions for configuring the communication channel between the MDM Agent and the MDM Server and conditionally, the MAS Server for each supported method.

Summary

[CCGUIDE] [\[1\]](#) section 2.3.3 *Configure MDM Agent and MDM Communications* provides information to configure the communications between the MDM Agent and MDM Server. It states as follows:

MDM Agent-Server communication is achieved securely using the MDM protocol which is built on top of HTTP, transport layer security (TLS), and push notifications that use HTTP PUT over TLS (secure sockets layer (SSL)). A managed mobile device uses an identity to authenticate itself to the MDM server over TLS (SSL). This identity can be included in the profile as a Certificate Payload or can be generated by enrolling the mobile device with Simple Certificate Enrollment Protocol (SCEP).

Also, the MDM Agent communications uses the iOS Security Framework as described in section 3.3.2 *TLS Configuration* of [CCGUIDE] [\[1\]](#). Thus, configuring the TOE's TLS protocol as per section 3.3.2 automatically configures the MDM Agent communications. If an additional CA certificate needs to be added to support the MDM Server, information is provided in section 3.3.2.3 of [CCGUIDE] [\[1\]](#).

Test Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-ATE-01

The evaluator shall also perform the following tests for each protocol listed:

- **Test 1:** *The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext and that a protocol analyzer identifies the traffic as the protocol under testing.*

- **Test 2:***[conditional] If IPsec is selected (and the TSF includes a native VPN client), the evaluator shall ensure that the TOE is able to initiate communications with a VPN Gateway, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- **Test 3:***[conditional] If OTA updates are selected, the evaluator shall trigger an update request according to the operational guidance and shall ensure that the communication is successful.*
- **Test 4:** *For any other selected protocol (not tested in Test 1, 2, or 3), the evaluator shall ensure that the TOE is able to initiate communications with a trusted IT product using the protocol, setting up the connection as described in the operational guidance and ensuring that the communication is successful.*

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

Test 1: the evaluator deployed the "atsec test profile" on the TOE, and connected the TOE to the WLAN network. For IPsec, the evaluator verified that the TOE connected automatically to Strongswan on the test laptop, went to Safari and accessed https://10.0.0.1/, and analyzed the traffic using tcpdump to make sure that HTTP traffic is indeed encapsulated with ESP. For HTTPS/TLS, the evaluator started a TLS server and connected the TOE to https://tauon, and verified that the s_server status was displayed on the web page.

Test 2: the evaluator reproduced the steps on Test 1 and verified by analyzing the captured packets that HTTP traffic is encapsulated into ESP and verified the SPI in both directions.

Test 3: the evaluator connected the TOE to the WLAN test network and started a system update by going to Settings -> General -> Software update, and verified that the software was downloaded.

Test 4: not applicable.

Assurance Activity AA-FTP_ITC_EXT.1-MDMA-MDF-ATE-01

For each supported identifier type (excluding DNs), the evaluator shall repeat the following tests:

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.

Test 2: The evaluator shall ensure, for each method of Agent-Server communication, the channel data is not sent in plaintext.

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 assurance activities are associated with the specific protocols.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

The evaluator deployed a WAP (WiFi Access Point) from a test laptop and connected the TOE to it. For both IPsec and HTTPS/TLS, the evaluator connected the TOE to the laptop and used Wireshark to verify that the communication channels were indeed encrypted. Wireshark indeed identified the traffic as either IKE/IPSec or HTTPS/TLS.



2.1.8.2 Trusted Channel Communication (Wireless LAN) (FTP_ITC_EXT.1(WLAN))

TSS Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-WLAN-ASE-01

The evaluator shall examine the TSS to determine that it describes the details of the TOE connecting to an access point in terms of the cryptographic protocols specified in the requirement, along with TOE-specific options or procedures that might not be reflected in the specification. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST.

Summary

Section 8.9.3, *Wireless LAN*, in the [ST]  describes the WLAN protocol implemented by the TOE. Section 8.9.1, *EAP-TLS and TLS*, in the [ST]  describes the EAP-TLS and TLS protocols and ciphersuites supported by the TOE.


Section 8.9.3 specify which protocols are implemented according to the IEEE 802.11 (2012) standard for WLAN and which algorithms are supported by the TOE. Section 8.9.1 specifies additional information about the EAP-TLS and TLS protocols certificates and ciphersuites that can be requested by applications.


Guidance Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-WLAN-AGD-01

The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to the access point, and that it contains recovery instructions should a connection be unintentionally broken.

Summary

[CCGUIDE]  section 3.5.6.3 *Mobile device administrators* states that to enforce the verification of the server name defined with the X.509 certificate during the WPA-EAP handshake between the TOE and the remote access point, the policy must contain the server name to be expected in the certificate with the *TLSTrustedReaderNames* setting. This can be configured with the Apple Configurator when configuring the “Trust” of the certificates for Wi-Fi EAP configurations by adding the server name to the list of trusted servers.

[CCGUIDE]  section 3.3.1 *EAP-TLS Configuration* states that no recovery instructions is needed for the automatic recovery of a broken Wi-Fi connection.

Test Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-WLAN-ATE-01

The evaluator shall perform the following tests:

- *Test 1: The evaluators shall ensure that the TOE is able to initiate communications with an access point using the protocols specified in the requirement, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- *Test 2: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.*
- *Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, modification of the channel data is detected by the TOE.*

- *Test 4: The evaluators shall physically interrupt the connection from the TOE to the access point (e.g., moving the TOE host out of range of the access point, turning the access point off). The evaluators shall ensure that subsequent communications are appropriately protected, at a minimum in the case of any attempts to automatically resume the connection or connect to a new access point.*

Further assurance activities are associated with the specific protocols.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1][\[1\]](#), the extended package [EP_MDM_AGENT_V3.0][\[2\]](#), the extended package [PP_WLAN_CLI_EP_V1.0][\[3\]](#), and the PP module [MOD_VPN_CLI_V2.1][\[4\]](#).

Test 1: the evaluator deployed a configuration profile on the TOE and verified that the TOE could successfully connect to the test AP.

Test 2: the evaluator used the same profile as in Test 1, and used Wireshark to sniff the wireless traffic when the connection is established with the AP. The evaluator could verify that the traffic is encrypted using EAP-TLS.

Test 3: the evaluator verified that Test 2 uses the EAP-TLS protocol.

Test 4: the evaluator deployed the same configuration profile on the TOE, terminated the connection on the AP side, and verified that the connection was terminated on the TOE. The evaluator restarted the AP on the test laptop and verified that the TOE automatically resumes the connection.

2.2 Security Assurance Requirements

2.2.1 Life-cycle support (ALC)

2.2.1.1 Labelling of the TOE (ALC_CMC.1)

Assurance Activity AA-ALC_CMC.1-ALC-01

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

Summary

[ST] section 1.2 *TOE Reference* identifies the TOE as "Apple iPad and iPhone Mobile Devices with iOS 12" specified in section 1.4 *TOE Description* of [ST]. These devices use either A8/A8X processor (iPhone 6, iPhone 6 Plus, iPad mini 4, iPad Air 2), A9/A9X processor (iPhone 6s Plus, iPhone 6s, iPhone SE, iPad 9.7-inch, iPad Pro 9.7-inch, iPad Pro 12.9-inch), A10 Fusion/A10X Fusion processor (iPhone 7 Plus, iPhone 7, iPad 9.7-inch, iPad Pro 12.9-inch, iPad Pro 10.5-inch), A11 Bionic processor (iPhone 8, iPhone 8 Plus, iPhone X), A12 Bionic (iPhone Xs, iPhone Xs Max, iPhone XR), A12X Bionic (iPad Pro 11-inch, iPad Pro 12.9-inch).

The guidance documentation, for example, [CCGUIDE] contains TOE reference as iOS 12.

In addition, for the delivery of the evaluated TOE devices, [CCGUIDE] section 2 *Secure Delivery and Installation* states the following:

The evaluated mobile devices are intended for authorized mobile device users of entities such as business organizations and government agencies.

The normal distribution channels for a regular end user to obtain these devices include:

- The Apple Store (either a physical stores or online at <https://www.apple.com>)
- Apple retailers
- Service carriers (e.g., AT&T, Verizon)
- Resellers

Business specific distribution channel

There is a distinct online store for Business customers with a link from the "Apple Store" to Apple and Business: (<https://www.apple.com/business/>). Additionally, the following link to "Shop for Business" is provided (<https://www.apple.com/retail/business/>).

Government specific distribution channel

Government customers can use the link: <https://www.apple.com/r/store/government/>

Additional

Large customers can have their own Apple Store Catalog for their employees to purchase devices directly from Apple under their corporate employee purchase program

2.2.1.2 TOE CM coverage (ALC_CMS.1)

Assurance Activity AA-ALC_CMS.1-ALC-01

The evaluator shall ensure that the developer has identified (in public-facing development guidance for their platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler and linker flags). The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled.

Summary

The evaluator examined the lifecycle evidence provided by the developer which contains the following information:

The developer uses the Xcode IDE (Integrated Development Environment) is used by developers for building apps for Mac, iPhone, and iPad. Xcode IDE is tightly integrated with the Cocoa and Cocoa Touch frameworks of the iOS. Xcode comes with an "Assistant editor" that provides intuitive display of the source code. Additionally, Apple LLVM compiler parses the code, keeping every symbol in the LLDB debugger consistent with the editor and compiler and finding mistakes and offering Fix-its for the code.

The vendor provided "Secure Coding Guide" [SecCodeGuide] which provides in section *Avoiding Buffer Overflows and Underflows* advice on how to avoid buffer overflows and underflows. In particular, subsection *Security Features that Can Help* states that iOS provides two features that can make it harder to exploit stack and buffer overflows which are address space layout randomization (ASLR) and non-executable stack and heap which are described as follows.

Address Space Layout Randomization

Recent versions of macOS and iOS, where possible, choose different locations for your stack, heap, libraries, frameworks, and executable code each time you run your software. This makes it much harder to successfully exploit buffer overflows because it is no longer possible to know where the buffer is in memory, nor is it possible to know where libraries and other code are located.

Address space layout randomization requires some help from the compiler—specifically, it requires position-independent code.

- *If you are compiling an executable that targets macOS 10.7 and later or iOS 4.3 and later, the necessary flags are enabled by default. You can disable this feature, if necessary, with the `-no_pie` flag, but for maximum security, you should not do so.*
- *If you are compiling an executable that targets an earlier OS, you must explicitly enable position-independent executable support by adding the `-pie` flag.*

Non-Executable Stack and Heap

Recent processors support a feature called the NX bit that allows the operating system to mark certain parts of memory as non-executable. If the processor tries to execute code in any memory page marked as non-executable, the program in question crashes..

macOS and iOS take advantage of this feature by marking the stack and heap as non-executable. This makes buffer overflow attacks harder because any attack that places executable code on the stack or heap and then tries to run that code will fail.

Most of the time, this is the behavior that you want. However, in some rare situations (such as writing a just-in-time compiler), it may be necessary to modify that behavior.

There are two ways to make the stack and heap executable:

- *Pass the `-allow_stack_execute` flag to the compiler. This makes the stack (not the heap) executable.*

- Use the *mprotect* system call to mark specific memory pages as executable.

Assurance Activity AA-ALC_CMS.1-ALC-02

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.

Summary

According to [ST][\[1\]](#), the TOE is Apple iPad and iPhone with iOS 12 and claims exact conformance to [PP_MD_V3.1][\[1\]](#), [EP_MDM_AGENT_V3.0][\[1\]](#), [PP_WLAN_CLI_EP_V1.0][\[1\]](#), and [MOD_VPN_CLI_V2.1][\[1\]](#). The exact conformance requires the TOE to be capable of providing/supporting the TSF defined in the aforementioned PP, EP's and PP module. As assessed in the work units of ALC_CMS.1E, the evaluator found that the provided TOE documentation associated with the TOE contains unique identification which is iPhone/iPad with iOS 12 (e.g., [iPad_UG][\[1\]](#), [iPad_UG][\[1\]](#) and [CCGUIDE][\[1\]](#)).

2.2.1.3 Extension: Timely Security Updates (ALC_TSU_EXT.1)

Assurance Activity AA-ALC_TSU_EXT.1-ALC-01

The evaluator shall verify that the TSS contains a description of the timely security update process used by the developer to create and deploy security updates. The evaluator shall verify that this description addresses the TOE OS, the firmware, and bundled applications, each. The evaluator shall also verify that, in addition to the TOE developer's process, any carrier or other third-party processes are also addressed in the description. The evaluator shall also verify that each mechanism for deployment of security updates is described.

The evaluator shall verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the TOE patching this vulnerability, to include any third-party or carrier delays in deployment. The evaluator shall verify that this time is expressed in a number or range of days.

The evaluator shall verify that this description includes the publicly available mechanisms (including either an email address or website) for reporting security issues related to the TOE. The evaluator shall verify that the description of this mechanism includes a method for protecting the report either using a public key for encrypting email or a trusted channel for a website.

The evaluator shall verify that the description includes where users can seek information about the availability of new security updates including details of the specific public vulnerabilities corrected by each update. The evaluator shall verify that the description includes the minimum amount of time that the TOE is expected to be supported with security updates, and the process by which users can seek information about when the TOE is no longer expected to receive security updates.

Summary

The Security Target [ST][\[1\]](#) section 8.7.4 *Secure Updates* describes the secure software update process for the TOE as follows:

Software updates to the TOE are released regularly to address emerging security concerns and also provide new features; these updates are provided for all supported devices simultaneously. Users receive iOS update notifications on the device and through iTunes, and updates are delivered wirelessly, encouraging rapid adoption of the latest security fixes. The startup process described above helps ensure that only Apple-signed code can be installed on a device. To prevent devices from being downgraded to older versions that lack the latest security updates, iOS uses a process called System Software Authorization. If downgrades were possible, an attacker who gains possession of a device could install an older version of iOS and exploit a vulnerability that's been fixed in the newer version. The Secure Enclave coprocessor also utilizes System Software Authorization to ensure the integrity of its software and prevent downgrade installations. iOS software updates can be installed using iTunes or over-the-air (OTA) on the

device via HTTPS trusted channel. With iTunes, a full copy of iOS is downloaded and installed. OTA software updates download only the components required to complete an update, improving network efficiency, rather than downloading the entire OS. Additionally, software updates can be cached on a local network server running the caching service on OS X Server so that iOS devices do not need to access Apple servers to obtain the necessary update data. During an iOS upgrade, iTunes (or the device itself, in the case of OTA software updates) connects to the Apple installation authorization server and sends it a list of cryptographic measurements for each part of the installation bundle to be installed (for example, LLB, iBoot, the kernel, and OS image), a random anti-replay value (nonce), and the device's unique ID (ECID). The authorization server checks the presented list of measurements against versions for which installation is permitted and, if it finds a match, adds the ECID to the measurement and signs the result. The server passes a complete set of signed data to the device as part of the upgrade process. Adding the ECID "personalizes" the authorization for the requesting device. By authorizing and signing only for known measurements, the server ensures that the update takes place exactly as provided by Apple. The boot-time, chain-of-trust evaluation verifies that the signature comes from Apple and that the measurement of the item loaded, combined with the device's ECID, matches what was covered by the signature. These steps ensure that the authorization is for a specific device and that an old iOS version from one device can't be copied to another. The nonce prevents an attacker from saving the server's response and using it to tamper with a device or otherwise alter the system software. Note that this ensures the integrity and authenticity of software updates. A TLS trusted channel is provided for this process. distributes information about security issues in its products through security advisories. Advisories are provided through the security-announce mailing list. Resources include the following.

Contact Apple About Security Issues <https://support.apple.com/en-us/HT201220>

Apple Security Updates (Advisories) <https://support.apple.com/en-us/HT201222>

Security-Announce Mailing List (receive Apple security advisories through) <https://lists.apple.com/mailman/listinfo/security-announce/>

2.2.2 Guidance documents (AGD)

2.2.2.1 Operational user guidance (AGD_OPE.1)

Assurance Activity AA-AGD_OPE.1-AGD-01

Some of the contents of the operational guidance are verified by the assurance activities in Section 5.1 and evaluation of the TOE according to the [CEM]. The following additional information is also required.

The operational guidance shall contain a list of natively installed applications and any relevant version numbers. If any third party vendors are permitted to install applications before purchase by the end user or enterprise, these applications shall also be listed.

The operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.


The documentation must describe the process for verifying updates to the TOE by verifying a digital signature. The evaluator shall verify that this process includes the following steps:

- *Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).*
- *Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.*


The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

Summary

Installed apps

[CCGUIDE]  section 5 *Installed Apps* contains table 17 "Built-in and Presintalled Apps" listing the apps that come with the TOE devices. Built-in apps and cannot be removed while preinstalled apps are included with purchased devices but may be removed by the user or administrator.

Cryptographic modules


[CCGUIDE]  section 3.2 *Cryptographic Support Functions* provides information about the cryptographic modules that provide cryptographic support to the TOE. It states that the TOE comes with the following three cryptographic modules that provide the cryptographic support for the TOE:



- Apple iOS CoreCrypto Kernel Cryptographic Module ARM, v9.0 (User Space)
- Apple iOS CoreCrypto Cryptographic Module for ARM, v9.0 (Kernel Space)
- Apple Secure Key Store Cryptographic Module, v9.0

Section 3.2 also contains the following warning:


" Warning: *The use of other cryptographic engines beyond those listed above was neither evaluated nor tested during the mobile device's Common Criteria evaluation.*"

Secure software updates


[CCGUIDE]  section 3.6.9 *Secure Software Updates* describes the procedure for obtaining and verifying updates to the TOE. The process is as follows:

- All iOS updates are digitally signed.
- The TOE's feature System Software Authorization prevents downgrading of the iOS version.
- Updates can be installed using iTunes or over-the-air (OTA) on the device. With iTunes, a full copy of iOS is downloaded (via TLS) and installed. OTA software updates download only the components required to complete an update.
- iOS updates can be cached on a local network server running the caching service on OS X Server so that iOS devices do not need to access Apple servers to obtain the necessary update data.
- Users can update iOS on their device by following the instructions provided in the user guides [iPhone_UG]  and [iPad_UG]  section "Update iOS software on iPhone" and "Update iOS software on your iPad", respectively.

Security functionality provided by the TOE:

[CCGUIDE]  section 1.2 *TOE Security Functionality* lists at a high-level the security functionality provided by the TOE in the evaluated configuration which covers:

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TOE Security Functionality (TSF)
- TOE access
- Trusted path/channels

While assessing other assurance activities for the guidance documentation, the evaluator also verified that the guidance covers the security functionalities listed above. The evaluator noticed that in particular, [CCGUIDE]  chapters 3 and 4 are largely structured based on these security

functionalities, for example, section 3.5 and section 3.6 are labeled as *Identification & Authentication* and *Security Management Configuration*, respectively. This structure makes it easy for the reader to follow as well as for the evaluator to verify which security functionalities from [PP_MD_V3.1], [EP_MDM_AGENT_V3.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1] are covered in the guidance/assurance activities.

Furthermore, section 1.9 *Un-evaluated Functionalities* of [CCGUIDE] explicitly lists the security functionalities that are outside the scope of the evaluated configuration.

2.2.2.2 Preparative procedures (AGD_PRE.1)

Assurance Activity AA-AGD_PRE.1-AGD-01

As indicated in the introduction of section 5.2.3, 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.

Summary

While performing other assurance activities the evaluator determined that sufficient guidance was provided for the TOE to address all claimed platforms. The guidance provided for the TOE applies to all platforms claimed in [ST].

2.2.3 Tests (ATE)

2.2.3.1 Independent testing - conformance (ATE_IND.1)

Assurance Activity AA-ATE_IND.1-ATE-01

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

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.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as 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).

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.

Summary

The Detailed Test Report specifies all the tests covering the assurance activities from [PP_MD_V3.1], the extended package [EP_MDM_AGENT_V3.0], the extended package [PP_WLAN_CLI_EP_V1.0], and the PP module [MOD_VPN_CLI_V2.1].

2.2.4 Vulnerability assessment (AVA)

2.2.4.1 Vulnerability survey (AVA_VAN.1)

Assurance Activity AA-AVA_VAN.1-AVA-01

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 find vulnerabilities that have been found in mobile devices and the implemented communication protocols in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report

For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

Summary

The evaluator searched for publicly known vulnerabilities applicable to iOS using the following sources.

- Apple security content disclosure statements for releases of iOS 12 related to this evaluation:
 - <https://support.apple.com/en-us/HT209162>, iOS 12.0.1, released October 8, 2018.
 - <https://support.apple.com/en-us/HT209192>, iOS 12.1, released October 30, 2018.
 - <https://support.apple.com/en-us/HT209340>, iOS 12.1.1, released December 5, 2018.
 - <https://support.apple.com/en-us/HT209443>, iOS 12.1.3, released January 22, 2019.
 - <https://support.apple.com/en-us/HT209520>, iOS 12.1.4, released February 7, 2019.
- MITRE Common Vulnerabilities and Exposures (CVE) List:
 - https://cve.mitre.org/cve/search_cve_list.html
- NIST National Vulnerability Database (NVD):
 - <https://web.nvd.nist.gov/view/vuln/search>

The following search terms were used on the MITRE and NIST web sites:

- ios ipad
- ios iphone
- ios core tls
- ios core crypto
- ios common crypto
- ios http
- ios https
- ios tcp
- ios ip
- ios bluetooth

- ios ipsec
- ios vpn
- ios mdm
- ios mobile
- broadcom wi-fi

A Appendixes

A.1 References

AConfig	Apple Configurator 2 Help (online guidance) Version 2018 Date received 2018-09-26 Location https://help.apple.com/configurator/mac/
APIGuide	Apple Developer Documentation Date 2018-03-28 Location https://developer.apple.com/documentation
CC	Common Criteria for Information Technology Security Evaluation Version 3.1R5 Date April 2017 Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf Location http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf
CCEVS-TD0194	Update to Audit of FTP_ITC_EXT.1/WLAN Date 2017-04-11 Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=198
CCEVS-TD0237	FAU_GEN.1.1(2) - FMT_UNR_EXT.1 Audit Record Selection-Based Date 2017-09-26 Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=243
CCEVS-TD0244	FCS_TLSC_EXT - TLS Client Curves Allowed Date 2017-11-16 Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=250
CCEVS-TD0301	Updates to Administrator Management and Biometric Authentication Date 2018-04-04 Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=307
CCEVS-TD0304	Update to FCS_TLSC_EXT.1.2 Date 2018-04-04 Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=310
CCEVS-TD0305	Handling of TLS connections with and without mutual authentication Date 2018-04-04 Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=311

CCEVS-TD0330 Curve25519 scheme moved to optional and FFC scheme using DH Group 14 added

Date 2018-06-01

Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=336**CCEVS-TD0346 Revision of FMT_SMF_EXT.2 in MDF PP**

Date 2018-08-23

Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=356**CCEVS-TD0347 Update of Use Case 2 in MDF PP**

Date 2018-08-23

Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=357**CCEVS-TD0351 Additional methods for DEK formation**

Date 2018-09-20

Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=361**CCEVS-TD0366 Flexibility in Password Conditioning in FCS_COP.1(5)**

Date 2018-10-12

Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=376**CCEVS-TD0369 Long-term trusted channel key material**

Date 2018-10-17

Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=379**CCEVS-TD0378 TOE/TOE Platform Selection in FCS_IPSEC_EXT.1 SFRs**

Date 2018-12-13

Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=388**CCEVS-TD0379 Updated FCS_IPSEC_EXT.1.11 Tests for VPN Client**

Author(s) NIAP

Date 2018-12-20

Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=389**CCGUIDE Apple iPad and iPhone Mobile Devices with iOS 12 Common Criteria Configuration Guide**

Author(s) atsec

Version 1.7

Date 2019-03-12

File name [agd/VID10937_CC-guide_v1.7.pdf](http://www.commoncriteriaportal.org/files/ccfiles/CC-guide_v1.7.pdf)**CEM Common Methodology for Information Technology Security Evaluation**

Version 3.1R5

Date April 2017

Location <http://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R5.pdf>

CKTSREF	Certificate, Key, and Trust Services (API Reference) Version 2018 Date received 2018-09-18 Location https://developer.apple.com/documentation/security/certificate_key_and_trust_services
COREBT	Core Bluetooth Framework Version 2018 Date received 2018-10-25 Location https://developer.apple.com/documentation/corebluetooth
DTR	Test Plan and Detailed Test Report Apple iOS Version 12.0 Version 3.2.1 Date 2019-01-18 File name VID10937_TestPlan-v3.2.1.pdf
EAR	Apple iOS 12 Entropy Assessment Report Version 5.4 Date 2018-11-16 File name EAR_Apple_iOS12-v5.4.pdf
EP_MDM_AGENT_V3.0	Extended Package for Mobile Device Management Agents Version 3.0 Version 3.0 Date 2016-11-21 Location https://www.niap-ccevs.org/pp/ep_mdm_agent_v3.0.pdf File name ase/ep_mdm_agent_v3.0.pdf
FSPMapping	Apple iOS 12 FSP Mapping Date 2018-10-02 File name adv/fsp/Apple_iOS_12_FSP.xlsx
HWSPEC	Hardware spec Version 4 Date received 2018-12-31 File name alc/DeviceInformation-iOS12-v4.pdf
IOS_CFG	Date 2018-10-05 File name agd/Configuration-Profile-Reference.pdf
IOS_LOGS	Profiles and Logs Date 2018 Location https://developer.apple.com/bug-reporting/profiles-and-logs/?platforms=ios
IOS_MDM	Mobile Device Management Protocol Reference Date 2018-09-18 Location https://developer.apple.com/enterprise/documentation/MDM-Protocol-Reference.pdf File name agd/MDM-Protocol-Reference.pdf

IOS_SEC	iOS Security, iOS 12 Author(s) Apple Inc. Version September 2018 Date received 2018-09-30 File name agd/iOS_Security_Guide.pdf
iOSDeployRef	iOS Deployment Reference Version iOS 12 and tvOS 12 Date received 2018-09-18 Location https://help.apple.com/deployment/ios/ File name agd/iOS Deployment Reference.pdf
iPad_UG	iPad User Guide Version iOS 12.0 Date 2018-09-18 File name agd/iPad User Guide.pdf
iPhone_UG	iPhone User Guide Version iOS 12.0 Date received 2018-09-18 File name agd/iPhone User Guide.pdf
KEYCHAINPG	Keychain Services Programming Guide Author(s) Apple, Inc. Date 2018 Location https://developer.apple.com/documentation/security/keychain_services File name agd/keychainServConcepts.pdf
LC	Apple iOS 12.0 Life Cycle Version 1.0 Date received 2019-01-21 File name alc/Apple_iOS_12_LifeCycle.pdf
LOGGING	Logging Date received 2018-09-18 Location https://developer.apple.com/documentation/os/logging?language=objc File name agd/Logging.pdf
MANAGE_CARDS	Manage the cards that you use with Apple Pay Date November 6, 2018 Location https://support.apple.com/en-us/HT205583
MOD_VPN_CLI_V2.1	Author(s) NIAP Version 2.1 Date 2017-10-05 File name https://www.niap-ccevs.org/Profile/Info.cfm?PPID=419&id=419
NETEXT	Network Extension Version 2018 Date received 2018-10-25 Location https://developer.apple.com/documentation/networkextension

NETWORK	Network Framework Version 2018 Date received 2018-10-25 Location https://developer.apple.com/documentation/network
OTA_CFG	Over-the-Air Profile Delivery and Configuration Date 2014-04-17 Location https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/iPhoneOTAConfiguration/Introduction/Introduction.html File name agd/iPhoneOTAConfiguration.pdf
PAY_SETUP	Set up Apple Pay Date January 3, 2019 Location https://support.apple.com/en-us/HT204506 File name agd/Set up Apple Pay - Apple Support.pdf
PP_MD_V3.1	Protection Profile for Mobile Device Fundamentals, Version 3.1 Version 3.1 Date received 2017-06-16 Location https://www.niap-ccevs.org/pp/pp_md_v3.1.pdf
PP_WLAN_CLI_EP_V1.0	Extended Package for Wireless LAN Client Version 1.0 Version 1.0 Date February 11, 2016 Location https://www.niap-ccevs.org/pp/pp_wlan_cli_ep_v1.0.pdf File name ase/pp_wlan_cli_ep_v1.0.pdf
SecCodeGuide	Secure Coding Guide Date 2016-09-13 Location https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Introduction.html
SECTRAN	Secure Transport API Version 2018 Date received 2018-10-25 Location https://developer.apple.com/documentation/security/secure_transport
ST	Apple iPad and iPhone Mobile Devices with iOS 12 Security Target Version 1.6 Date 2019-03-12 File name ase/st-vid10937_PUBLIC_V1.6.pdf
TRUST_STORE	List of available trusted root certificates in iOS 12, macOS10.14, watchOS 5, and tvOS 12 Date received 2018-09-18 Location https://support.apple.com/en-us/HT209144 File name agd/Trust store iOS12.pdf

A.2 Glossary

Augmentation

The addition of one or more requirement(s) to a package.

Authentication data

Information used to verify the claimed identity of a user.

Authorised user

A user who may, in accordance with the SFRs, perform an operation.

Class

A grouping of CC families that share a common focus.

Component

The smallest selectable set of elements on which requirements may be based.

Connectivity

The property of the TOE which allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.

Dependency

A relationship between components such that if a requirement based on the depending component is included in a PP, ST or package, a requirement based on the component that is depended upon must normally also be included in the PP, ST or package.

Deterministic RNG (DRNG)

An RNG that produces random numbers by applying a deterministic algorithm to a randomly selected seed and, possibly, on additional external inputs.

Element

An indivisible statement of security need.

Entropy

The entropy of a random variable X is a mathematical measure of the amount of information gained by an observation of X.

Evaluation

Assessment of a PP, an ST or a TOE, against defined criteria.

Evaluation Assurance Level (EAL)

An assurance package, consisting of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale.

Evaluation authority

A body that implements the CC for a specific community by means of an evaluation scheme and thereby sets the standards and monitors the quality of evaluations conducted by bodies within that community.

Evaluation scheme

The administrative and regulatory framework under which the CC is applied by an evaluation authority within a specific community.

Exact conformance

a subset of Strict Conformance as defined by the CC, is defined as the ST containing all of the requirements in the Security Requirements section of the PP, and potentially requirements from Appendices of the PP. While iteration is allowed, no additional requirements (from the CC parts 2 or 3) are allowed to be included in the ST. Further, no requirements in the Security Requirements section of the PP are allowed to be omitted.

Extension

The addition to an ST or PP of functional requirements not contained in Part 2 and/or assurance requirements not contained in Part 3 of the CC.

External entity

Any entity (human or IT) outside the TOE that interacts (or may interact) with the TOE.

Family

A grouping of components that share a similar goal but may differ in emphasis or rigour.

Formal

Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.

Guidance documentation

Documentation that describes the delivery, preparation, operation, management and/or use of the TOE.

Identity

A representation (e.g. a string) uniquely identifying an authorised user, which can either be the full or abbreviated name of that user or a pseudonym.

Informal

Expressed in natural language.

Object

A passive entity in the TOE, that contains or receives information, and upon which subjects perform operations.

Operation (on a component of the CC)

Modifying or repeating that component. Allowed operations on components are assignment, iteration, refinement and selection.

Operation (on an object)

A specific type of action performed by a subject on an object.

Operational environment

The environment in which the TOE is operated.

Organisational Security Policy (OSP)

A set of security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment.

Package

A named set of either functional or assurance requirements (e.g. EAL 3).

PP evaluation

Assessment of a PP against defined criteria.

Protection Profile (PP)

An implementation-independent statement of security needs for a TOE type.

Random number generator (RNG)

A group of components or an algorithm that outputs sequences of discrete values (usually represented as bit strings).

Refinement

The addition of details to a component.

Role

A predefined set of rules establishing the allowed interactions between a user and the TOE.

Secret

Information that must be known only to authorised users and/or the TSF in order to enforce a specific SFP.

Secure state

A state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs.

Security attribute

A property of subjects, users (including external IT products), objects, information, sessions and/or resources that is used in defining the SFRs and whose values are used in enforcing the SFRs.

Security Function Policy (SFP)

A set of rules describing specific security behaviour enforced by the TSF and expressible as a set of SFRs.

Security objective

A statement of intent to counter identified threats and/or satisfy identified organisation security policies and/or assumptions.

Security Target (ST)

An implementation-dependent statement of security needs for a specific identified TOE.

Seed

Value used to initialize the internal state of an RNG.

Selection

The specification of one or more items from a list in a component.

Semiformal

Expressed in a restricted syntax language with defined semantics.

ST evaluation

Assessment of an ST against defined criteria.

Subject

An active entity in the TOE that performs operations on objects.

Target of Evaluation (TOE)

A set of software, firmware and/or hardware possibly accompanied by guidance.

TOE evaluation

Assessment of a TOE against defined criteria.

TOE resource

Anything useable or consumable in the TOE.

TOE Security Functionality (TSF)

A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs.

Transfers outside of the TOE

TSF mediated communication of data to entities not under control of the TSF.

True RNG (TRNG)

A device or mechanism for which the output values depend on some unpredictable source (noise source, entropy source) that produces entropy.

Trusted channel

A means by which a TSF and a remote trusted IT product can communicate with necessary confidence.

Trusted path

A means by which a user and a TSF can communicate with necessary confidence.

TSF data

Data created by and for the TOE, that might affect the operation of the TOE.

TSF Interface (TSFI)

A means by which external entities (or subjects in the TOE but outside of the TSF) supply data to the TSF, receive data from the TSF and invoke services from the TSF.

User

See external entity

User data

Data created by and for the user, that does not affect the operation of the TSF.