Apple iPadOS 14: iPads Assurance Activity Report

Version:	1.2
Date:	2021-08-31
Status:	RELEASED
Classification:	Public
Filename:	VID11147_SER_AAR_Apple_iPadOS_14_v1.2
Product:	Apple iPadOS 14: iPads
Sponsor:	Apple Inc.
Evaluation Facility:	atsec information security corporation
Validation ID:	11147
Validation Body:	NIAP CCEVS
Author(s):	Trang Huynh
Quality Assurance:	King Ables

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
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
0.1	2021-05-12	Trang Huynh	First version	
1.0	2021-07-19	Trang Huynh	Added CAVP certs	
1.1	2021-08-27	King Ables	Add test configuration summary.	
1.2	2021-08-31	King Ables	Update AVA.	

Copyright © 2021 atsec information security corporation

Table of Contents

1 Evaluation Basis and Documents 13
2 Evaluation Results 15
2.1 CAVP Summary 15
2.2 Test configuration 21
2.3 Security Functional Requirements 21
2.3.1 Security audit (FAU) 21
2.3.1.1 Audit Data Generation (FAU_GEN.1) 21
TSS Assurance Activities 21
Guidance Assurance Activities 21
Test Assurance Activities 22
2.3.1.2 Audit Storage Protection (FAU_STG.1) 23
TSS Assurance Activities 23
Guidance Assurance Activities 23
Test Assurance Activities 23
2.3.1.3 Prevention of Audit Data Loss (FAU_STG.4) 24
TSS Assurance Activities 24
Guidance Assurance Activities 24
Test Assurance Activities 24
2.3.1.4 Agent Alerts (FAU_ALT_EXT.2) 24
TSS Assurance Activities 24
Guidance Assurance Activities 25
Test Assurance Activities 25
2.3.1.5 Audit Data Generation (FAU_GEN.1(2)) 26
TSS Assurance Activities 26
Guidance Assurance Activities 26
Test Assurance Activities 26
2.3.1.6 Security Audit Event Selection (FAU_SEL.1(2)) 26
TSS Assurance Activities 26
Guidance Assurance Activities 27
Test Assurance Activities 27
2.3.2 Cryptographic support (FCS) 28
2.3.2.1 Cryptographic key generation (FCS_CKM.1) 28
TSS Assurance Activities 28
Guidance Assurance Activities 28
Test Assurance Activities 29
2.3.2.2 Cryptographic key establishment (FCS_CKM.2(1)) 32
TSS Assurance Activities 32
Guidance Assurance Activities 33
Test Assurance Activities 33
2.3.2.3 Cryptographic key distribution (While device is locked) (FCS_CKM.2(2)) 36
TSS Assurance Activities 36
Guidance Assurance Activities 36
Test Assurance Activities 36
2.3.2.4 Extended: Cryptographic Key Support (REK) (FCS_CKM_EXT.1) 37

Copyright © 2021 atsec information security corporation

TSS Assurance Activities 37
Guidance Assurance Activities 38
Test Assurance Activities 38
2.3.2.5 Extended: Cryptographic Key Random Generation (FCS_CKM_EXT.2) 38
TSS Assurance Activities 38
Guidance Assurance Activities 39
Test Assurance Activities 39
2.3.2.6 Extended: Cryptographic Key Generation (FCS_CKM_EXT.3) 41
TSS Assurance Activities 41
Guidance Assurance Activities 42
Test Assurance Activities 43
2.3.2.7 Extended: Cryptographic Key Destruction (FCS_CKM_EXT.4) 45
TSS Assurance Activities 45
Guidance Assurance Activities 45
Test Assurance Activities 45
2.3.2.8 Extended: TSF Wipe (FCS_CKM_EXT.5) 46
TSS Assurance Activities 46
Guidance Assurance Activities 47
Test Assurance Activities 47
2.3.2.9 Extended: Salt Generation (FCS_CKM_EXT.6) 48
TSS Assurance Activities 48
Guidance Assurance Activities 48
Test Assurance Activities 48
2.3.2.10 Extended: Cryptographic Key Support (REK) (FCS_CKM_EXT.7) 48
TSS Assurance Activities 48
Guidance Assurance Activities 48
Test Assurance Activities 48
2.3.2.11 Cryptographic operation (Confidentiality Algorithms) (FCS_COP.1(1)) 48
TSS Assurance Activities 48
Guidance Assurance Activities 49
Test Assurance Activities 49
2.3.2.12 Cryptographic operation (Hashing Algorithms) (FCS_COP.1(2)) 52
TSS Assurance Activities 52
Guidance Assurance Activities 52
Test Assurance Activities 52
2.3.2.13 Cryptographic operation (Signature Algorithms) (FCS_COP.1(3)) 53
TSS Assurance Activities 53
Guidance Assurance Activities 53
Test Assurance Activities 53
2.3.2.14 Cryptographic operation (Keyed Hash Algorithms) (FCS_COP.1(4)) 54
TSS Assurance Activities 54
Guidance Assurance Activities 55
Test Assurance Activities 55
2.3.2.15 Cryptographic operation (Password-based Key Derivation Functions) (FCS_COP.1(5)) 55
TSS Assurance Activities 55
Guidance Assurance Activities 56

Copyright © 2021 atsec information security corporation

Test Assurance Activities 56
2.3.2.16 Extended: HTTPS Protocol (FCS_HTTPS_EXT.1) 56
TSS Assurance Activities 56
Guidance Assurance Activities 56
Test Assurance Activities 56
2.3.2.17 Extended: Initialization Vector Generation (FCS_IV_EXT.1) 57
TSS Assurance Activities 57
Guidance Assurance Activities 57
Test Assurance Activities 57
2.3.2.18 Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1) 57
TSS Assurance Activities 57
Guidance Assurance Activities 58
Test Assurance Activities 58
2.3.2.19 Extended: Cryptographic Algorithm Services (FCS_SRV_EXT.1) 59
TSS Assurance Activities 59
Guidance Assurance Activities 59
Test Assurance Activities 59
2.3.2.20 Extended: Cryptographic Key Storage (FCS_STG_EXT.1) 59
TSS Assurance Activities 59
Guidance Assurance Activities 60
Test Assurance Activities 60
2.3.2.21 Extended: Encrypted Cryptographic Key Storage (FCS_STG_EXT.2) 61
FCS_STG_EXT.2.1 61
FCS_STG_EXT.2.2 62
2.3.2.22 Extended: Integrity of encrypted key storage (FCS_STG_EXT.3) 62
TSS Assurance Activities 62
Guidance Assurance Activities 62
Test Assurance Activities 62
2.3.2.23 Extended: EAP TLS Protocol (FCS_TLSC_EXT.1) 62
FCS_TLSC_EXT.1.1 62
FCS_TLSC_EXT.1.2 66
FCS_TLSC_EXT.1.3 68
FCS_TLSC_EXT.1.4 68
2.3.2.24 Extended: TLS Protocol (FCS_TLSC_EXT.2) 70
TSS Assurance Activities 70
Guidance Assurance Activities 70
Test Assurance Activities 70
2.3.2.25 Cryptographic Key Storage (FCS_STG_EXT.4) 70
TSS Assurance Activities 70
Guidance Assurance Activities 71
Test Assurance Activities 72
2.3.2.26 Cryptographic Key Generation (IKE) (FCS_CKM.1(VPN)) 72
TSS Assurance Activities 72
Guidance Assurance Activities 72
Test Assurance Activities 72
2.3.2.27 IPsec (FCS_IPSEC_EXT.1) 72

Copyright © 2021 atsec information security corporation

FCS_IPSEC_EXT.1.1 72
FCS_IPSEC_EXT.1.2 75
FCS_IPSEC_EXT.1.3 76
FCS_IPSEC_EXT.1.4 77
FCS_IPSEC_EXT.1.5 78
FCS_IPSEC_EXT.1.6 79
FCS_IPSEC_EXT.1.7 80
FCS_IPSEC_EXT.1.8 81
FCS_IPSEC_EXT.1.9 82
FCS_IPSEC_EXT.1.11 83
FCS_IPSEC_EXT.1.12 85
FCS_IPSEC_EXT.1.13 86
FCS_IPSEC_EXT.1.14 86
2.3.2.28 Cryptographic Key Generation (Symmetric Keys for WPA2 Connections) (FCS_CKM.1/WLAN) 87
TSS Assurance Activities 87
Guidance Assurance Activities 88
Test Assurance Activities 88
2.3.2.29 Cryptographic Key Distribution (GTK) (FCS_CKM.2/WLAN) 89
TSS Assurance Activities 89
Guidance Assurance Activities 89
Test Assurance Activities 89
2.3.2.30 Extensible Authentication Protocol-Transport Layer Security (FCS_TLSC_EXT.1/WLAN) 90
TSS Assurance Activities 90
Guidance Assurance Activities 90
Test Assurance Activities 91
2.3.3 User data protection (FDP) 93
2.3.3.1 Extended: Security access control (FDP_ACF_EXT.1) 93
FDP_ACF_EXT.1.1 93
FDP_ACF_EXT.1.2 95
2.3.3.2 Extended: Protected Data Encryption (FDP_DAR_EXT.1) 96
TSS Assurance Activities 96
Guidance Assurance Activities 97
Test Assurance Activities 97
2.3.3.3 Extended: Sensitive Data Encryption (FDP_DAR_EXT.2) 98
FDP_DAR_EXT.2.1 98
FDP_DAR_EXT.2.2 99
FDP_DAR_EXT.2.3 100
FDP_DAR_EXT.2.4 101
2.3.3.4 Extended: Subset information flow control (FDP_IFC_EXT.1) 101
TSS Assurance Activities 101
Guidance Assurance Activities 102
Test Assurance Activities 102
2.3.3.5 Extended: Storage of Critical Biometric Parameters (FDP_PBA_EXT.1) 103
TSS Assurance Activities 103
Guidance Assurance Activities 104

Copyright © 2021 atsec information security corporation

Test Assurance Activities 104
2.3.3.6 Extended: User Data Storage (FDP_STG_EXT.1) 104
TSS Assurance Activities 104
Guidance Assurance Activities 105
Test Assurance Activities 105
2.3.3.7 Extended: Inter-TSF user data transfer protection (FDP_UPC_EXT.1) 105
TSS Assurance Activities 105
Guidance Assurance Activities 105
Test Assurance Activities 106
2.3.3.8 Full Residual Information Protection (FDP_RIP.2) 107
TSS Assurance Activities 107
Guidance Assurance Activities 108
Test Assurance Activities 108
2.3.3.9 Subset Information Flow Control (FDP_IFC_EXT.1(VPN)) 108
TSS Assurance Activities 108
Guidance Assurance Activities 108
Test Assurance Activities 109
2.3.4 Identification and authentication (FIA) 109
2.3.4.1 Extended: Authentication failure handling (FIA_AFL_EXT.1) 109
TSS Assurance Activities 109
Guidance Assurance Activities 111
Test Assurance Activities 112
2.3.4.2 Extended: Bluetooth User Authorization (FIA_BLT_EXT.1) 112
TSS Assurance Activities 112
Guidance Assurance Activities 113
Test Assurance Activities 113
2.3.4.3 Extended: Bluetooth Authentication (FIA_BLT_EXT.2) 114
TSS Assurance Activities 114
Guidance Assurance Activities 114
Test Assurance Activities 114
2.3.4.4 Rejection of Duplicate Bluetooth Connections (FIA_BLT_EXT.3) 115
TSS Assurance Activities 115
Guidance Assurance Activities 115
Test Assurance Activities 115
2.3.4.5 Extended: Secure Simple Pairing (FIA_BLT_EXT.4) 116
TSS Assurance Activities 116
Guidance Assurance Activities 116
Test Assurance Activities 116
2.3.4.6 Extended: Accuracy of Biometric Authentication (FIA_BMG_EXT.1) 117
FIA_BMG_EXT.1.1 117
FIA_BMG_EXT.1.2 117
2.3.4.7 Extended: Biometric Enrollment (FIA_BMG_EXT.2) 118
TSS Assurance Activities 118
Guidance Assurance Activities 119
Test Assurance Activities 120
2.3.4.8 Extended: Biometric Verification (FIA_BMG_EXT.3) 120
TSS Assurance Activities 120

Copyright © 2021 atsec information security corporation

Guidance Assurance Activities 121
Test Assurance Activities 121
2.3.4.9 Extended: Handling Unusual Biometric Templates (FIA_BMG_EXT.5) 121
TSS Assurance Activities 121
Guidance Assurance Activities 122
Test Assurance Activities 122
2.3.4.10 Extended: Password Management (FIA_PMG_EXT.1) 122
TSS Assurance Activities 122
Guidance Assurance Activities 122
Test Assurance Activities 123
2.3.4.11 Extended: Authentication Throttling (FIA_TRT_EXT.1) 123
TSS Assurance Activities 123
Guidance Assurance Activities 123
Test Assurance Activities 123
2.3.4.12 Multiple Authentication Mechanisms (FIA_UAU.5) 124
TSS Assurance Activities 124
Guidance Assurance Activities 125
Test Assurance Activities 125
2.3.4.13 Re-Authentication (FIA_UAU.6) 125
FIA_UAU.6.1 125
FIA_UAU.6.2 126
2.3.4.14 Protected authentication feedback (FIA_UAU.7) 127
TSS Assurance Activities 127
Guidance Assurance Activities 127
Test Assurance Activities 128
2.3.4.15 Extended: Authentication for Cryptographic Operation (FIA_UAU_EXT.1) 128
TSS Assurance Activities 128
Guidance Assurance Activities 129
Test Assurance Activities 129
2.3.4.16 Extended: Timing of Authentication (FIA_UAU_EXT.2) 129
TSS Assurance Activities 129
Guidance Assurance Activities 130
Test Assurance Activities 130
2.3.4.17 Extended: Validation of certificates (FIA_X509_EXT.1) 130
TSS Assurance Activities 130
Guidance Assurance Activities 131
Test Assurance Activities 131
2.3.4.18 Extended: X509 certificate authentication (FIA_X509_EXT.2) 133
TSS Assurance Activities 133
Guidance Assurance Activities 134
Test Assurance Activities 134
2.3.4.19 Extended: Request Validation of certificates (FIA_X509_EXT.3) 135
TSS Assurance Activities 135
Guidance Assurance Activities 135
Test Assurance Activities 135

Copyright © 2021 atsec information security corporation

2.3.4.20 Agent Enrollment of Mobile Device into Management (FIA_ENR_EXT.2) 136
TSS Assurance Activities 136
Guidance Assurance Activities 136
Test Assurance Activities 137
2.3.4.21 Port Access Entity Authentication (FIA_PAE_EXT.1) 137
TSS Assurance Activities 137
Guidance Assurance Activities 137
Test Assurance Activities 137
2.3.4.22 X.509 Certificate Validation (EAP-TLS) (FIA_X509_EXT.1/WLAN) 138
TSS Assurance Activities 138
Guidance Assurance Activities 138
Test Assurance Activities 139
2.3.4.23 X.509 Certificate Authentication (EAP-TLS) (FIA_X509_EXT.2/WLAN) 139
TSS Assurance Activities 139
Guidance Assurance Activities 140
Test Assurance Activities 140
2.3.5 Security management (FMT) 141
2.3.5.1 Extended: Management of security functions behavior (FMT_MOF_EXT.1) 141
FMT_MOF_EXT.1.1 141
FMT_MOF_EXT.1.2 141
2.3.5.2 Extended: Specification of Management Functions (FMT_SMF_EXT.1) 142
TSS Assurance Activities 142
Guidance Assurance Activities 149
Test Assurance Activities 153
2.3.5.3 Extended: Specification of Specification of Remediation Actions (FMT_SMF_EXT.2) 168
TSS Assurance Activities 168
Guidance Assurance Activities 168
Test Assurance Activities 168
2.3.5.4 Agent Trusted Policy Update (FMT_POL_EXT.2) 169
TSS Assurance Activities 169
Guidance Assurance Activities 169
Test Assurance Activities 169
2.3.5.5 Specification of Management Functions (FMT_SMF_EXT.4) 170
TSS Assurance Activities 170
Guidance Assurance Activities 171
Test Assurance Activities 171
2.3.5.6 User Unenrollment Prevention (FMT_UNR_EXT.1) 172
TSS Assurance Activities 172
Guidance Assurance Activities 172
Test Assurance Activities 173
2.3.5.7 Specification of Management Functions (VPN) (FMT_SMF.1(VPN)) 173
TSS Assurance Activities 173
Guidance Assurance Activities 173
Test Assurance Activities 174

Copyright © 2021 atsec information security corporation

2.3.5.8 Specification of Management Functions (Wireless LAN) (FMT_SMF_EXT.1/WLAN) 174
TSS Assurance Activities 174
Guidance Assurance Activities 174
Test Assurance Activities 175
2.3.6 Protection of the TSF (FPT) 175
2.3.6.1 Extended: Extended: Anti-Exploitation Services (ASLR) (FPT_AEX_EXT.1) 175
TSS Assurance Activities 175
Guidance Assurance Activities 175
Test Assurance Activities 175
2.3.6.2 Extended: Anti-Exploitation Services (Memory Page Permissions) (FPT_AEX_EXT.2) 176
TSS Assurance Activities 176
Guidance Assurance Activities 176
Test Assurance Activities 176
2.3.6.3 Extended: Anti-Exploitation Services (Overflow Protection) (FPT_AEX_EXT.3) 176
TSS Assurance Activities 176
Guidance Assurance Activities 177
Test Assurance Activities 177
2.3.6.4 Extended: Domain Isolation (FPT_AEX_EXT.4) 177
TSS Assurance Activities 177
Guidance Assurance Activities 179
Test Assurance Activities 179
2.3.6.5 Extended: JTAG Disablement (FPT_JTA_EXT.1) 180
TSS Assurance Activities 180
Guidance Assurance Activities 180
Test Assurance Activities 180
2.3.6.6 Extended: Key Storage (FPT_KST_EXT.1) 181
TSS Assurance Activities 181
Guidance Assurance Activities 182
Test Assurance Activities 182
2.3.6.7 Extended: No Key Transmission (FPT_KST_EXT.2) 182
TSS Assurance Activities 182
Guidance Assurance Activities 184
Test Assurance Activities 184
2.3.6.8 Extended: No Plaintext Key Export (FPT_KST_EXT.3) 184
TSS Assurance Activities 184
Guidance Assurance Activities 184
Test Assurance Activities 184
2.3.6.9 Extended: Self-Test Notification (FPT_NOT_EXT.1) 184
TSS Assurance Activities 184
Guidance Assurance Activities 185
Test Assurance Activities 185
FPT_NOT_EXT.1.1 185
2.3.6.10 Reliable time stamps (FPT_STM.1) 185
TSS Assurance Activities 185

Copyright © 2021 atsec information security corporation

Guidance Assurance Activities 186
Test Assurance Activities 186
2.3.6.11 Extended: TSF Cryptographic Functionality Testing (FPT_TST_EXT.1) 187
TSS Assurance Activities 187
Guidance Assurance Activities 187
Test Assurance Activities 187
2.3.6.12 Extended: TSF Integrity Checking (FPT_TST_EXT.2) 188
TSS Assurance Activities 188
Guidance Assurance Activities 189
Test Assurance Activities 189
2.3.6.13 Extended: TSF Integrity Testing (FPT_TST_EXT.3) 190
TSS Assurance Activities 190
Guidance Assurance Activities 190
Test Assurance Activities 190
2.3.6.14 Extended: Trusted Update: TSF version query (FPT_TUD_EXT.1) 190
TSS Assurance Activities 190
Guidance Assurance Activities 190
Test Assurance Activities 190
2.3.6.15 Extended: Trusted Update Verification (FPT_TUD_EXT.2) 191
FPT_TUD_EXT.2.3 191
FPT_TUD_EXT.2.4 193
2.3.6.16 Extended: Trusted Update Verification (FPT_TUD_EXT.3) 194
TSS Assurance Activities 194
Guidance Assurance Activities 194
Test Assurance Activities 194
2.3.6.17 Extended: Trusted Update Verification (FPT_TUD_EXT.4) 194
FPT_TUD_EXT.4.1 194
FPT_TUD_EXT.4.2 195
2.3.6.18 TSF Self Test (VPN) (FPT_TST_EXT.1(VPN)) 196
TSS Assurance Activities 196
Guidance Assurance Activities 198
Test Assurance Activities 200
2.3.6.19 TSF Cryptographic Functionality Testing (Wireless LAN) (FPT_TST_EXT.1/WLAN) 200
TSS Assurance Activities 200
Guidance Assurance Activities 201
Test Assurance Activities 202
2.3.7 TOE access (FTA) 202
2.3.7.1 Extended: TSF- and User-initiated locked state (FTA_SSL_EXT.1) 202
TSS Assurance Activities 202
Guidance Assurance Activities 202
Test Assurance Activities 203
2.3.7.2 Default TOE Access Banners (FTA_TAB.1) 203
TSS Assurance Activities 203
Guidance Assurance Activities 203
Test Assurance Activities 204
2.3.7.3 Wireless Network Access (FTA_WSE_EXT.1) 204

Copyright © 2021 atsec information security corporation

TSS Assurance Activities 204
Guidance Assurance Activities 204
Test Assurance Activities 204
2.3.8 Trusted path/channels (FTP) 205
2.3.8.1 Extended: Trusted channel Communication (FTP_ITC_EXT.1) 205
TSS Assurance Activities 205
Guidance Assurance Activities 206
Test Assurance Activities 207
2.3.8.2 Trusted Channel Communication (FTP_ITC_EXT.1(2)) 208
TSS Assurance Activities 208
Guidance Assurance Activities 208
Test Assurance Activities 209
2.3.8.3 Trusted Path (for Enrollment) (FTP_TRP.1(2)) 209
TSS Assurance Activities 209
Guidance Assurance Activities 210
Test Assurance Activities 210
2.3.8.4 Trusted Channel Communication (Wireless LAN) (FTP_ITC_EXT.1/WLAN) 211
TSS Assurance Activities 211
Guidance Assurance Activities 211
Test Assurance Activities 211
2.4 Security Assurance Requirements 213
2.4.1 Life-cycle support (ALC) 213
2.4.1.1 Labelling of the TOE (ALC_CMC.1) 213
2.4.1.2 TOE CM coverage (ALC_CMS.1) 214
2.4.1.3 Extension: Timely Security Updates (ALC_TSU_EXT.1) 215
2.4.2 Guidance documents (AGD) 217
2.4.2.1 Operational user guidance (AGD_OPE.1) 217
2.4.2.2 Preparative procedures (AGD_PRE.1) 218
2.4.3 Tests (ATE) 219
2.4.3.1 Independent testing - conformance (ATE_IND.1) 219
2.4.4 Vulnerability assessment (AVA) 219
2.4.4.1 Vulnerability survey (AVA_VAN.1) 219
A Appendixes 222
A.1 References 222
A.2 Glossary 228

Classification: Public
Copyright © 2021 atsec information security corporation

List of Tables

Table 1: SFRs to CAVP certificates for iPads 15
Table 2: Broadcom core supported algorithms and CAVP certificates for iPads 20
Table 3: HMAC Functions 54
Table 4: Summary of keys and persistent secrets used by the MDM Agent 71
Table 5: EAP-TLS Ciphersuites 91

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 defined in the following:

- [CFG_MDF-MDM_AGENT-VPNC_V1.0] : PP-Configuration for Mobile Device Fundamentals (MDF), Mobile Device Management (MDM) Agents, and Virtual Private Network (VPN) Clients, Version 1.0, dated 2020-02-28
- [PP_MD_V3.1] : Protection Profile for Mobile Device Fundamentals, Version 3.1, dated 2017-06-16
- [MOD_MDM_AGENT_V1.0] : PP-Module for MDM Agents Version 1.0, dated 2019-04-25
- [MOD_VPN_CLI_V2.1] : PP-Module for Virtual Private Network (VPN) Clients, Version 2.1, dated 2017-10-05
- [PP_WLAN_CLI_EP_V1.0] : General Purpose Operating Systems Protection Profile/ Mobile Device Fundamentals Protection Profile Extended Package (EP) Wireless Local Area Network (WLAN) Clients, dated 2016-02-08

This evaluation claims Exact Compliance with the above PP, PP-Modules, and Extended Packages.

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-TD0244]: "FCS_TLSC_EXT - TLS Client Curves Allowed", version as of 2017-11-16.
- [CCEVS-TD0301]: "Updates to Administrator Management and Biometric Authenication", 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.

Copyright © 2021 atsec information security corporation

- [CCEVS-TD0404]: "Cryptographic selections and updates for use with App PP v1.3", version as of 2019-08-30.
- [CCEVS-TD0413]: "Conformance Claim to allow for a PP-Module", version as of 2019-03-22.
- [CCEVS-TD0426]: "Correction of Inconsistency with RFC7748", version as of 2019-06-24.
- [CCEVS-TD0439]: "EAP-TLS Revocation Checking", version as of 2019-08-29.
- [CCEVS-TD0468]: "Bluetooth pairing vs connection", version as of 2019-11-22.
- [CCEVS-TD0470]: "Wireless Network Restrictions", version as of 2020-01-22.
- [CCEVS-TD0491]: "Update to FMT_SMF_EXT.4 Test 2", version as of 2020-01-15.
- [CCEVS-TD0492]: "TLS-EAP Ciphers and TLS versions for WLAN Client", version as of 2020-01-22.
- [CCEVS-TD0508]: "Conformance Claim to allow for a PP-Module", version as of 2020-02-28.
- [CCEVS-TD0517]: "WLAN Client Corrections for X509 and TLSC", version as of 2020-06-19.
- [CCEVS-TD0523]: "Updates to Certificate Revocation (FIA_X509_EXT.1)", version as of 2020-07-01.
- [CCEVS-TD0579]: "FAU_GEN.1, item 7", version as of 2021-03-02.

Copyright © 2021 atsec information security corporation

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.

2.1 CAVP Summary

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.

The CAVP operational environments (OEs) for the software and firmware modules are the following.

- iPadOS 14 with Apple processors:
  - A9, A9X, A10 Fusion, A10X Fusion, A12 Bionic, A12X Bionic, A12Z Bionic, A14 Bionic

Module (Mod) names in Table 1.

BC
Broadcom Core

HW
SEP Hardware v2.0

KRN
Apple corecrypto Module v11.0 [Apple silicon, Kernel, Software]

SKS
Apple corecrypto Module v11.0 [Apple silicon, Secure Key Store, Hardware]

USR
Apple corecrypto Module v11.0 [Apple silicon, User, Software]

It is noted that the Apple corecrypto Module is v11.0, but the SEP Hardware is v2.0.

Table 1: SFRs to CAVP certificates for iPads

SFR	Algorithm	Modes / Other	Mod	Implementation	CAVP
FCS_CKM.1(1)	ECDSA KeyGen and KeyVer [FIPS 186-4]	P-256, P-384	USR	vng_ltc	A996
			KRN	vng_ltc	A1001
			SKS	vng_ltc	A1334
FCS_CKM.1/VPN	ECDSA KeyGen and KeyVer [FIPS 186-4]	P-256, P-384	USR	vng_ltc	A996
FCS_CKM.2(1)	ECC Key Establishment (KAS-ECC-SSC)	P-256, P-384	USR	c_ltc	A997
			SKS	c_ltc	A1332

Copyright © 2021 atsec information security corporation

SFR	Algorithm	Modes / Other	Mod	Implementation	CAVP
	[SP800-56Ar3]				
FCS_COP.1(1)	AES [FIPS 197]	CBC 128-bit, 256-bit [SP800-38A] (CBC)	USR	asm_arm	A995
			KRN	asm_arm	A999
			SKS	asm_arm	A1329
		CBC 128-bit, 256-bit encrypt only [SP800-38A] (CBC)	HW	A9 (skg)	C314
				A9X (skg)	C315
				A10 Fusion (skg)	C317
				A10X Fusion (skg)	C318
				A12 Bionic (skg)	C320
				A12X Bionic (skg)	C322
				A12Z Bionic (skg)	A510
				A14 Bionic (skg)	A1469
		CCM 128-bit [SP800-38C] (CCM)	USR	vng_asm	A992
			KRN	vng_asm	A1002
			SKS	vng_asm	A1333
			BC	4350	AES 4035
				4355	AES 3678
				4377	AES 4791
				4378 Aux	AES 5926
				4378 Main	AES 5927
				4387 Aux	AES 5926
				4387 Main	AES 5927
		CCM 256-bit [SP800-38C] (CCM)	USR	vng_asm	A992
			KRN	vng_asm	A1002
			SKS	vng_asm	A1333
			BC	4378 Aux	AES 5952

Classification: Public
Copyright © 2021 atsec information security corporation

SFR	Algorithm	Modes / Other	Mod	Implementation	CAVP
				4378 Main	AES 5953
				4387 Aux	AES 5952
				4387 Main	AES 5953
		ECB 128-bit, 256-bit [SP800-38A] (ECB) (Used by AES-KW)	HW	A9 (skg)	C314
				A9X (skg)	C315
				A10 Fusion (skg)	C317
				A10X Fusion (skg)	C318
				A12 Bionic (skg)	C320
				A12X Bionic (skg)	C322
				A12Z Bionic (skg)	A510
				A14 Bionic (skg)	A1469
		ECB 256-bit encrypt only [SP800-38A] (ECB) (Prereq for the hardware CTR_DRBG(AES))	HW	A9	AES 5275
				A9X	AES 5272
				A10 Fusion	AES 5273
				A10X Fusion	AES 5274
				A12 Bionic	C323
				A12X Bionic	C324
				A12Z Bionic (trng)	A501
				A14 Bionic (trng)	A1362
		GCM 128-bit [SP800-38D] (GCM)	USR	vng_asm	A992
			KRN	vng_asm	A1002
			SKS	vng_asm	A1333
			BC	4377	AES 4791
				4378 Aux	AES 5926
				4378 Main	AES 5927

Classification: Public
Copyright © 2021 atsec information security corporation

SFR	Algorithm	Modes / Other	Mod	Implementation	CAVP
				4387 Aux	AES 5926
				4387 Main	AES 5927
		GCM 256-bit [SP800-38D] (GCM)	USR	vng_asm	A992
			KRN	vng_asm	A1002
			SKS	vng_asm	A1333
			BC	4378 Aux	AES 5926
				4378 Main	AES 5927
				4387 Aux	AES 5926
				4387 Main	AES 5927
		KW 128-bit, 256-bit [SP800-38F] (KW)	USR	c_asm	A994
			KRN	c_asm	A998
			SKS	c_asm	A1330
		XTS 128-bit, 256-bit [SP800-38E] (XTS)	USR	asm_arm	A995
			KRN	asm_arm	A999
			SKS	asm_arm	A1329
FCS_COP.1(2)	SHS Byte-oriented mode [FIPS 180-4]	SHA2-256	USR	vng_neon	A991
			KRN	vng_neon	A1000
			SKS	vng_neon	A1335
		SHA-1, SHA2-384, SHA2-512	USR	vng_ltc	A996
			KRN	vng_ltc	A1001
			SKS	vng_ltc	A1334
FCS_COP.1(3)	RSA SigGen PKCS 1.5 and PKCSPSS [FIPS 186-4]	Modulo: 2048, 3072, 4096 using SHA2-256, SHA2-384, SHA2-512	USR	vng_ltc	A996
			KRN	vng_ltc	A1001

Copyright © 2021 atsec information security corporation

SFR	Algorithm	Modes / Other	Mod	Implementation	CAVP
	RSA SigVer PKCS 1.5 and PKCSPSS [FIPS 186-4]	Modulo: 2048, 3072, 4096 using SHA-1, SHA2-256, SHA2-384, SHA2-512	USR	vng_ltc	A996
			KRN	vng_ltc	A1001
	ECDSA SigGen [FIPS 186-4]	P-256, P-384, P-521 using SHA2-256, SHA2-384, SHA2-512	USR	vng_ltc	A996
			KRN	vng_ltc	A1001
			SKS	vng_ltc	A1334
	ECDSA SigVer [FIPS 186-4]	P-256, P-384, P-521 using SHA-1, SHA2-256, SHA2-384, SHA2-512	USR	vng_ltc	A996
			KRN	vng_ltc	A1001
			SKS	vng_ltc	A1334
FCS_COP.1(4)	HMAC Byte-oriented mode [FIPS 198-1]	HMAC-SHA2-256	USR	vng_neon	A991
			KRN	vng_neon	A1000
			SKS	vng_neon	A1335
		HMAC-SHA-1, HMAC-SHA2-384, HMAC-SHA2-512	USR	vng_ltc	A996
			KRN	vng_ltc	A1001
			SKS	vng_ltc	A1334
FCS_RBG_EXT.1 (Kernel and User space)	CTR_DRBG(AES) [SP800-90A]	AES-128	USR	vng_asm	A992
			KRN	vng_asm	A1002
FCS_RBG_EXT.1 (SEP)	CTR_DRBG(AES) [SP800-90A]	AES-256	HW	A9	DRBG 2025
				A9X	DRBG 2022
				A10 Fusion	DRBG 2023
				A10X Fusion	DRBG 2024
				A12 Bionic	C323
				A12X Bionic	C324

Classification: Public
Copyright © 2021 atsec information security corporation

SFR	Algorithm	Modes / Other	Mod	Implementation	CAVP
				A12Z Bionic (trng)	A501
				A14 Bionic (trng)	A1362

Table 2 maps iPads to Broadcom cores and Broadcom CAVP certificates.

Table 2: Broadcom core supported algorithms and CAVP certificates for iPads

iPad	Broadcom Core # (Implementation)	Supported Algorithms	CAVP
iPad Pro 12.9-inch (A9X)	4350 (aes_core.vhd rev 9)	AES-CCM-128	AES 4035
iPad 9.7-inch (5th gen) (A9) iPad Pro 9.7-inch (A9X) iPad 9.7-inch (6th gen) (A10 Fusion) iPad 10.2-inch (7th gen) (A10 Fusion) iPad Pro 12.9 (2nd gen) (A10X Fusion) iPad Pro 10.5 (A10X Fusion) iPad 10.2-inch (8th gen) (A12 Bionic)	4355 (aes_core.vhd rev 10)	AES-CCM-128	AES 3678
iPad mini (5th gen) (A12 Bionic) iPad Air 10.5-inch (3rd gen) (A12 Bionic) iPad Pro 11-inch (A12X Bionic) iPad Pro 12.9-inch (3rd gen) (A12X Bionic)	4377 (aes_core_gcm.vhd rev 4)	AES-CCM-128 AES-GCM-128	AES 4791
iPad Pro 11-inch (2nd gen) (A12Z Bionic) iPad Pro 12.9-inch (4th gen) (A12Z Bionic)	4378 Aux (2.4 GHz) (aes_core_gcm.vhd rev 6)	AES-CCM-128 AES-GCM-128 AES-GCM-256	AES 5926
		AES-CCM-256	AES 5952
	4378 Main (5 GHz) (aes_core_gcm_simult_enc_mic.vhd rev 6)	AES-CCM-128 AES-GCM-128 AES-GCM-256	AES 5927
		AES-CCM-256	AES 5953
iPad Air (4th gen) (A14 Bionic)	4387 Aux (2.4 GHz) (aes_core_gcm.vhd rev 6)	AES-CCM-128 AES-GCM-128 AES-GCM-256	AES 5926
		AES-CCM-256	AES 5952
	4387 Main (5 GHz) (aes_core_gcm_simult_enc_mic.vhd rev 6)	AES-CCM-128 AES-GCM-128 AES-GCM-256	AES 5927
		AES-CCM-256	AES 5953

Copyright © 2021 atsec information security corporation

2.2 Test configuration

The TOE is part of a Commercial Off the Shelf System (COTS) and is distributed either as already installed software on a device via retail channels, or via download from Apple using a proprietary protocol. The TOE is installed and configured precisely as specified in [CCGUIDE] and verified to be in the evaluated configuration at the start of each test.

The TOE is connected to a private WLAN network which hosts also a Linux system as well as a macOS server (a macOS server is formed by using the macOS operating system and installing the macOS Server software on top of it that can be obtained from the Apple App store).

The Linux server provides:

- WLAN access point functionality,
- an uplink to the Internet (Internet access is required for the TOE to be initialized as well as for the Apple Push Notification Service (APNS) upon which the communication between the MDM server and the TOE rests),
- the hosting of network sniffer tools required for testing, and
- a VPN Gateway with the Strongswan IKE daemon and the Linux kernel IPsec support.

The macOS server provides:

- the MDM server (by using the Apple Profile Manager product part of the macOS Server software published by Apple), and
- the Apple Configurator 2 application.

2.3 Security Functional Requirements

2.3.1 Security audit (FAU)

2.3.1.1 Audit Data Generation (FAU_GEN.1)

TSS Assurance Activities

No assurance activities defined.

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

Copyright © 2021 atsec information security corporation

[CCGUIDE] chapter 6 *Security Audit* describes the audit function and provides Table 16 "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], [MOD_MDM_AGENT_V1.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1].

Section 6.1 *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 6.1 of [CCGUIDE], 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 [DEV_MAN], [PROFS_LOGS], and [LOGGING]. 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 [MOD_MDM_AGENT_V1.0] , [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:

- The audit records described in Table 16 provide the required information such as timestamps and subject identity.
- The SFRs covered in Table 15 match with Table 3 "Combined mandatory auditable events from [PP_MD_V3.1] and [PP_WLAN_CLI_EP_V1.0]" and Table 4 "Auditable events from [MOD_MDM_AGENT_V1.0]" of [ST] that together cover the mandatory auditable events required by [PP_MD_V3.1], [MOD_MDM_AGENT_V1.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1].
- Each SFR from Table 3 and Table 4 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

Copyright © 2021 atsec information security corporation

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.

[DTR] 5.1.1

2.3.1.2 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 *Audit Records* describes audit records performed by the TOE. This section specifies the storage location for the audit records which depends on operating system of the trusted workstation or MDM server, can be the following:

- 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

Test 1 & Test 2: 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 TOE devices where only the operating system can write to. The audit data is protected from modification attempts by user processes using the standard TOE OS 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 both modify and delete the log and verified that both the modification and deletion did not succeed.

[DTR] 5.1.2

Copyright © 2021 atsec information security corporation

2.3.1.3 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 *Audit Records* describes the audit function. This section states that the audit storage capacity is defined by the Configuration Profiles 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.

2.3.1.4 Agent Alerts (FAU_ALT_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FAU_ALT_EXT.2-MDMA-ASE-01

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

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

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

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

Summary

Section 8.10.2 *MDM Agent Alerts* describes the agent alerts generated and received by the TOE. This section states the following:

- The MDM agent generates and sends alerts in response to an MDM server such as a request for applying a policy or receiving a reachability event.
- When candidate policies (defined in configuration profile) are sent to an MDM agent, the agent responses in the form of "Alert". 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", CommandFormatError, "Idle", "NotNow".

Copyright © 2021 atsec information security corporation

- Periodic reachability events are initiated by the MDM Server using Push Notifications. The ST does not select "configure periodicity of reachability events" in FMT_SMF_EXT.4.2.
- In cases where the HTTPS channel is unavailable, for example because the device is out of range of a suitable network, an alert in regard to the successful installation of policies is queued until the device is able to communicate with the server again. The queue cannot become long, because if the device is out of communication with the MDM server no additional requests can be received. If the MDM Server does not receive the alert, the MDM Server should re-initiate the transfer until a response is received from the device. MDM Server requests may be processed, for example, when databases cannot be modified while the device is locked with Data Protection. When a device cannot perform a command due to these types of situations, it will send the NotNow status without performing the command. The server may send another command immediately after receiving this status, but chances are the following command will also be refused. After sending a NotNow status, the device will poll the server at some future time. The device will continue to 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-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.2.1 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.*
- ***Test 4:*** *The evaluator shall remove network connectivity from the MDM Agent and generate an alert/event as defined in FAU_ALT_EXT.2.1. The evaluator shall restore network connectivity to the MDM Agent and verify that the alert generated while the TOE was disconnected is sent by the MDM Agent upon re-establishment of the connectivity.*

Summary

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.

Test 4: The evaluator disconnected the device from the WLAN and re-ran Test 1 without the steps for verification of actions. He then verified a task is added to the list of Active Tasks and reconnected the device to the WLAN. Within a minute, the new password setting was found to be configured. The evaluator then verified on the Profile Manager web UI that the Completed Tasks list shows "Push Settings" with the date and time of the configuration.

[DTR] 6.1.1

Classification: Public
Copyright © 2021 atsec information security corporation

2.3.1.5 Audit Data Generation (FAU_GEN.1(2))

TSS Assurance Activities

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

Summary

Section 8.10.1 *Audit Records* states that specific audit record is determined via configuration profile, that at minimum includes the following information:

- Date and time the audit record was generated
- Process ID or information about the cause of the event
- Information about the intended operation
- Success or failure (where appropriate)

This section also states that audit record information is not available to TOE users or administrators on TOE devices and is only accessible externally on trusted workstations via the Apple Configurator 2 or to an MDM server on enrolled devices.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

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

> *The evaluator shall use the TOE to perform the auditable events defined in the Auditable Events table in FAU_GEN.1.1(2) and observe that accurate audit records are generated with contents and formatting consistent with those described in the TSS. Note that this testing can be accomplished in conjunction with the testing of the security mechanisms directly.*

Summary

This testing has been performed in conjunction with FAU_GEN.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.

[DTR] 5.1.1

2.3.1.6 Security Audit Event Selection (FAU_SEL.1(2))

TSS Assurance Activities

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

Classification: Public
Copyright © 2021 atsec information security corporation

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

Summary

The ST selects "implements functionality" in FAU_SEL.1(2), therefore this assurance activity is not applicable.

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 6.2 *Audit Storage* and section 6.3 *Configure the Auditable Items* provide guidance for how to select auditable events to be audited. These sections state the following:

- The TOE OS has a logging framework that is used to configure different logging levels for the various TOE OS subsystems. This framework is configured by creating and installing a logging configuration profile property list file (i.e., .plist file) into the appropriate directory. More information may be found 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 [PROFS_LOGS], additional logs can be specified by performing user actions on a device or through using a Configuration Profile.
- Table 17 "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

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

Test 2 is not applicable since the functionality is not claimed in [ST].

Copyright © 2021 atsec information security corporation

[DTR] 6.1.2

2.3.2 Cryptographic support (FCS)

2.3.2.1 Cryptographic key generation (FCS_CKM.1)

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 8.3.4 *Explanation of usage for cryptographic functions* in the TSS of [ST] provides table 9 "Explanation of usage for cryptographic functions in the cryptographic modules". Table entry "FCS_CKM.1(1) Cryptographic Key Generation" identifies the following schemes for asymmetric key pair generation:

- ECC scheme - ECDSA KeyGen and KeyVer with curves P-256, P-384, and Curve25519.
- FFC scheme - Diffie-Hellman Group 14 with key size 2048-bits.

The evaluator found this information to be consistent with the defintion of FCS_CKM.1(1).

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 iOS 14: iPhones and Apple iPadOS 14: iPads Common Criteria Configuration Guide as the main guidance document, hereafter shorthanded as [CCGUIDE].

Section 3 *Introduction* of this guidance document contains the following important statement:

> *This document is written for administrators and users of Apple mobile devices that are managed using a mobile device management (MDM) solution. The "Apple iOS 14: iPhones Security Target" [IOS_ST] and "Apple iPadOS 14: iPads Security Target" [IPADOS_ST] includes specifications for security where the mobile device operating environment includes a Wi-Fi network and includes evaluation of the Always-On virtual private network (VPN) provided by iOS/iPadOS. This configuration guide applies to both evaluations VID11146 (Apple iOS 14: iPhones) and VID11147 (Apple iPadOS 14: iPads). Where applicable, the guide shall point out the differences between iPhones and iPads.*

For this report, the evaluator will focus on information relevant to VID11147 (Apple iPadOS 14: iPads).

Copyright © 2021 atsec information security corporation

[CCGUIDE] section 3.7 *Security Functional Requirements in the ST requiring configuration* contains Table 4: "SFR Configuration Requirements". Table entry for FCS_CKM.1(1) {MDF} {VPN} indicates that there is no configuration needed. This table entry also references section 5.2.1 *Key Generation, Signature Generation and Verification* which states:

" *For the evaluated configuration, no configuration is required from the mobile device user.* "

and

" *For the evaluated configuration, no configuration is required from the mobile device administrator.* "

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 p1, p2, q1, q2, p and q shall all be provable primes*
   - *Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes*
   - *Primes p1, p2, q1, q2, p and q shall all be probable primes*

To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.

If possible, the Random Probable primes method should also be verified against a known good implementation as described above. Otherwise, the evaluator shall have the TSF generate 10 keys pairs for each supported key length nlen and verify:

- *n = p*q*
- *p and q are probably prime according to Miller-Rabin tests*
- *GCD(p-1,e) = 1*
- *GCD(q-1,e) = 1*
- *2^16 < e < 2^256 and e is an odd integer*
- *|p-q| > 2^(nlen/2 – 100)*
- *p >= squareroot(2)*(2^(nlen/2 -1))*
- *q >= squareroot(2)*(2^(nlen/2 -1))*
- *2^(nlen/2) < d < LCM(p-1,q-1)*
- *e*d = 1 mod LCM(p-1,q-1)*

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

Copyright © 2021 atsec information security corporation

[DTR] 5.2.1

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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.1

Assurance Activity AA-FCS_CKM.1-ATE-03

Key Generation for Curve25519

[TD0502] 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) and shall be written in little-endian order (least significant byte first). 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 (using little-endian order):

a. confirm the private and public keys are 32-byte values

b. confirm the three least significant bits of the first byte of the private key are zero

c. confirm the most significant bit of the last byte is zero

d. confirm the second most significant bit of the last 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 evaluator performed CAVS-like testing using a tool developed by the lab for Curve25519.

[DTR] 5.2.1

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:

Copyright © 2021 atsec information security corporation

- *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:

- *len(q) bit output of RBG where 1 <=x <= q-1*
- *len(q) + 64 bit output of RBG, followed by a mod q-1 operation where 1<= x<=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 != 0,1*
- *q divides p-1*
- *g^q mod p = 1*
- *g^x mod p = y*

for each FFC parameter set and key pair.

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.1

Assurance Activity AA-FCS_CKM.1-ATE-05

Diffie-Hellman Group 14 and FFC Schemes using "safe-prime" groups

Testing for FFC Schemes using Diffie-Hellman group 14 and/or "safe-prime" groups is done as part of testing in FCS_CKM.2(1).

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.1

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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

Copyright © 2021 atsec information security corporation

2.3.2.2 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 8.3.4 *Explanation of usage for cryptographic functions* in the TSS of [ST] provides table 9 "Explanation of usage for cryptographic functions in the cryptographic modules". Table entry "FCS_CKM.2(1)" identifies the following schemes for key establishment:

- RSA
- ECC
- FFC scheme - Diffie-Hellman group 14.

The evaluator found this information to correspond with the key generation schemes specified in FCS_CKM.1.1.

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 8.9.1 *EAP-TLS and TLS* in the TSS of [ST] states that the TOE supports RSA [SP800-56B] key establishment schemes as both a sender (e.g., when starting a TLS session) and a recipient (e.g., during a rekey triggered by the remote endpoint).

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 *Storage of Persistent Secrets and Private Keys by the Agent* in the TSS of [ST] states that the TOE handles all decryption errors in compliance with NIST Special Publication 800-56B.

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

[TD0504] For all key establishment schemes refer to the Assurance Activity for FCS_CKM.2(1) in the MDF PP. Note that because a TOE that conforms to this PP-Module must implement IPsec, the tested protocols shall include IPsec at minimum.

Summary

Copyright © 2021 atsec information security corporation

This assurance activity was performed in conjunction with the assurance activity for FCS_CKM.2(1) in the MDF PP, AA-FCS_CKM.2-1-ASE-01.

The ST does select Diffie-Hellman group 14 in FCS_CKM.2(1) for the VPN. Section 8.9.4.3 *IPsec Characteristics* states the following:

> "*The key generation and key establishment for VPN are handled in the user space, while the bulk encryption is handled in the kernel space. The VPN seeds its private copy of the CTR_DRBG present in the corecrypto instance used by the VPN client with 384 bits of entropy, invokes the corecrypto ECDH or DH APIs to generate the appropriate keys using key material generated by the CTR_DRBG, and follows the key establishment algorithms defined in [SP800-56A], ECDSA key generation algorithm in [FIPS 186-4], Diffie-Hellman Group (MODP) key generation [RFC 3526], and IKEv2 key derivation function in [RFC 5996].* "

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 4 "SFR Configuration Requirements" identifies which SFR can be or needs to be configured in the evaluated configuration. For FCS_CKM.2(1) {MDF} {VPN}, there is no configuration necessary for the selection of key establishment schemes by the TOE users or administrators. Table entry FCS_CKM.2(1) {MDF} {VPN} references section 5.2.2 which also states:

> " *For the evaluated configuration, no configuration is required from the mobile device user.* "

and

> " *For the evaluated configuration, no configuration is required from the mobile device administrator.* "

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,

Copyright © 2021 atsec information security corporation

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.

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.3

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

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 from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with our without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and

Copyright © 2021 atsec information security corporation

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

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.3

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

[TD0502] RSAES-PKCS1-v1_5 Key Establishment Schemes

The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_ITC_EXT.1 that uses RSAES-PKCS1-v1_5.

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.3

Assurance Activity AA-FCS_CKM.2-1-ATE-04

[TD0502] Diffie-Hellman Group 14

The evaluator shall 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 that uses Diffie-Hellman Group 14.

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.3

Assurance Activity AA-FCS_CKM.2-1-ATE-05

[TD0502] FFC Schemes using "safe-prime" groups

The evaluator shall verify the correctness of the TSF's implementation of "safe-prime" groups by using a known good implementation for each protocol selected in FTP_ITC_EXT.1 that uses "safe-prime" groups. This test must be performed for each "safe-prime" group that each protocol uses.

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

Copyright © 2021 atsec information security corporation

[DTR] 5.2.3

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

[TD0504] For all key establishment schemes refer to the Assurance Activity for FCS_CKM.2(1) in the MDF PP. Note that because a TOE that conforms to this PP-Module must implement IPsec, the tested protocols shall include IPsec at minimum.

Summary

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

2.3.2.3 Cryptographic key distribution (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

Copyright © 2021 atsec information security corporation

The evaluator performed CAVP-like testing using a tool developed by the lab for Curve25519.

[DTR] 5.2.4

2.3.2.4 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 *Hardware Protection Functions* and section 8.3 *Cryptographic Support* in the TSS of [ST] provide relevant description about key management.

Section 8.3.1 *Overview of Key Management* describes an overview of key management performed by the TOE. This description includes a description of a REK, the TOE UID used by the TOE.

Section 8.2.1 *The Secure Enclave Processor (SEP)* 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 TOE 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. 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 AES engine accessible only from the Secure Enclave. The UID is used to derive two other keys: 0x89B and 0x835. 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

Copyright © 2021 atsec information security corporation

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.

2.3.2.5 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 *Overview of Key Management* in the TSS of [ST] states that every time a file on the data partition is created, a new 256-bit AES key (the "per-file" key) is created using the hardware random number generator of the SEP. Files are encrypted using this key with AES in CBC mode where the IV is calculated with the block offset into the file, encrypted with the SHA-1 of the per-file key.

Assurance Activity AA-FCS_CKM_EXT.2-ASE-02

Copyright © 2021 atsec information security corporation

[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 and expansion step* in the TSS of [ST] describes that steps used for key derivation. The steps are 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] Table 4 "SFR Configuration Requirements" identifies which SFR can be or needs to be configured in the evaluated configuration. For FCS_CKM_EXT.2 {MDF}, there is no configuration necessary by the TOE users or administrators since generation and maintenance of DEK is hard coded.

This assurance activity is also performed in conjunction with the test assurance activities.

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	*l length*	*l 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).*

Classification: Public
Copyright © 2021 atsec information security corporation

- *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 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.*

Copyright © 2021 atsec information security corporation

- *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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.7

2.3.2.6 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 *Overview of Key Management* of the [ST] describes the key management.

Copyright © 2021 atsec information security corporation

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(SEP).

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 implemented by the TOE 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 ST selects "concatenating the keys and using a KDF (as described in (SP 800- 56C)" in FCS_CKM_EXT.3.

Section 8.3.3 *Randomness extraction and expansion step* in the TSS of [ST] describes the randomness extraction step. According to this section, the TOE implements the Key Derivation Function (KDF) 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 TOE's implementation of the KDF uses HMAC-SHA-256 for both the extraction and the expansion phase. Specifically, the evaluator verified that the extraction and expansion steps are explicitly described in this section which the evaluator determined to be a reproduction of sections 2.2 and 2.3 of RFC 5869. This section also states that the salt length and output key of the KDF are 256 bits.

Guidance Assurance Activities

No assurance activities defined.

Copyright © 2021 atsec information security corporation

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 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.*

Copyright © 2021 atsec information security corporation

- *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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.8

Copyright © 2021 atsec information security corporation

2.3.2.7 Extended: Cryptographic 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 *Overview of Key Management* in the TSS of [ST] how the TOE performs key management.

This section provides Table 7 "Summary of keys and persistent secrets in the TOE OS" and table 8 "Summary of keys and persistent secrets used by the Agent" which 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 (device key) and the EMF key (file system master 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 TOE system other than the SEP. All other keys are stored in non-volatile memory wrapped.

Also, section 8.3.2 *Storage of Persistent Secrets and Private Keys by the Agent* in the TSS states that all clearing operations of keys stored in volatile memory are performed by Apple corecrypto Kernel Cryptographic Module for Apple Silicon and Apple corecrypto Cryptographic Module for Apple Silicon or by the AES implementation within the Wi-Fi chip. Also, this section 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 KEKs 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.

Copyright © 2021 atsec information security corporation

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

This test is performed as part of the zeroization testing in FIPS 140-2 conformance testing which is also performed by the evaluator. As all software requiring cryptographic support use CoreCrypto, the zeroization tests performed as part of FIPS 140-2 cover this test. These test results are included in evidence.

[DTR] 5.2.9

2.3.2.8 Extended: TSF Wipe (FCS_CKM_EXT.5)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.5-ASE-01

Copyright © 2021 atsec information security corporation

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 *Overview of Key Management* in the TSS of [ST] describes how the TOE performs key management.

This section states 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 effectively wiped. The overwriting of the KEKs is done through a memset(0) operation that overwrites the memory as detailed in the assurance activity for FCS_CKM_EXT.4.

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 files). 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

When effaceable storage is reserved, a handle is provided for access. No pointer directly to the storage location is provided, therefore the memory cannot be read 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.

[DTR] 5.2.10

Copyright © 2021 atsec information security corporation

2.3.2.9 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 8.2.1 *The Secure Enclave Processor (SEP)* describes how the SEP performs salt generation. The Secure Enclave uses its own phyiscal noise source and RNG to generate the 128-bit salt value for the PBKDF (specially PBKDF2), which uses AES with the TOE device UID as the key for the psuedo-random function (PRF). This salt is regenerated every time the passcode changes. All other salt values used in the TOE OS are generated using the TRNG 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.

2.3.2.10 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

This assurance activity was performed in conjunction with FCS_CKM_EXT.1.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

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

TSS Assurance Activities

No assurance activities defined.

Classification: Public
Copyright © 2021 atsec information security corporation

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 <=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 <=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:

Copyright © 2021 atsec information security corporation

```
# 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.*

Copyright © 2021 atsec information security corporation

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 (KEKs)

  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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.13

Copyright © 2021 atsec information security corporation

2.3.2.12 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 *Cryptographic Support* in the TSS of [ST] describes cryprographic services supported by the TOE.

The TSS in section 8.3.4 *Explanation of usage for cryptographic functions* provides table 9 "Explanation of usage for cryptographic functions in the cryptographic modules" which describes the hashing functions specified in FCS_COP.1(2). It states that the hashing functions are implemented in byte-oriented mode.

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 4 "SFR Configuration Requirements" identifies which SFR can be or needs to be configured in the evaluated configuration. For FCS_COP.1(2) {MDF}, 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 5.2.3 *Hashing* which states the following:

- 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 in the VPN payload. 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

Copyright © 2021 atsec information security corporation

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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.14

2.3.2.13 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***

Copyright © 2021 atsec information security corporation

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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.15

2.3.2.14 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 8.3.4 *Explanation of usage for cryptographic functions* in the TSS of [ST] provides table 9 "Explanation of usage for cryptographic functions in the cryptographic modules". The table entry FCS_COP.1(4) specifies the following HMAC functions and associated information:

Table 3: HMAC Functions

HMAC Function	Key Length	Block Size	Output MAC Length
HMAC-SHA-1	greater than or equal to 112 bits	512 bits	160 bits
HMAC-SHA-256	greater than or equal to 112 bits	512 bits	256 bits

Classification: Public
Copyright © 2021 atsec information security corporation

HMAC Function	Key Length	Block Size	Output MAC Length
HMAC-SHA-384	greater than or equal to 112 bits	1024 bits	384 bits
HMAC-SHA-512	greater than or equal to 112 bits	1024 bits	512 bits

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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.16

2.3.2.15 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 *Password based key derivation* in the TSS of [ST] describes the password-based key derivation. It states the following:

Copyright © 2021 atsec information security corporation

- The TOE implements PBKDF2 to derive a 256-bit key from a user's passcode. The PBKDF2 is implemented as specified in [SP800-132] following “Option 2b” defined in section 5.4 of the standard. It uses HMAC-SHA-256 as the pseudorandom function (PRF).
- The input to the PBKDF2 is the 128-bit random salt generated by the SEP (as described in section 8.2.1), the user’s passcode without any pre-processing, and an iteration count of one. The output is the 256-bit key mentioned above.
- Next, the output of the PBKDF2 is repeatedly encrypted with the AES-CBC-256 hardware cipher using the 256-bit UID as the encryption key to generate 256 bits of data with each loop iteration. The loop is performed as often as needed to reach a duration between 100 and 150 milliseconds on that device.
- The output after all AES iterations have completed forms the 256-bit root encryption key used to unwrap the user’s keybag that holds the class keys for the file system data protection. Only when the unwrapping of the user’s keybag is successful is the user considered authenticated.
- The number of AES-CBC-256 iterations is calibrated to take at least 100 to 150 milliseconds and is a minimum of 50,000. The number of iterations is device-specific and may be greater than 50,000 on some devices.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.2.16 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

Copyright © 2021 atsec information security corporation

Test 1: The evaluator loaded a configuration profile on the TOE containing all the information necessary to connect to the WiFi Access Point (WAP). The evaluator started a TLS web server, connected the TOE to it, and verified using Wireshark that the connection used TLS 1.2 and was successful.

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

The evaluator disabled the trust for the certificate in
`Settings -> General -> About -> Certificate Trust Settings`
and re-performed Test 1. The evaluator verified that the warning about the certificate was displayed once again.

Test 1 (repeat): redo of Test 1 except that the CA is not trusted in the keychain. The tester notices that the connection to the webpage is interrupted by Safari prompting to either accept or reject the certificate before connecting.

[DTR] 5.2.18

2.3.2.17 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 *Overview of Key Management* in the TSS of [ST] describes key management. Per this section, every time a file on the data partition is created, a new 256-bit AES key (the "per-file" key) is created using the hardware random number generator of the SEP. Files are encrypted using this key with AES in cipher block chaining (CBC) mode where the initialization vector (IV) is calculated with the block offset into the file, encrypted with the SHA-1 hash of the per-file key. (FCS_RBG_EXT.1(SEP) is used for the data encryption keys).

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.2.18 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".*

Copyright © 2021 atsec information security corporation

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 the proprietary Entropy Assessment Report (EAR) 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 <= 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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.20

Copyright © 2021 atsec information security corporation

2.3.2.19 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] (subsection *Keys*) and verified that it 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

This test is satisfied by the CAVP certificates listed in the Assurance Activity Report.

[DTR] 5.2.21

2.3.2.20 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

The ST selects "software-based" in FCS_STG_EXT.1 as a key storage mechanism for asymmetric keys, symmetric keys, and persistent secrets.

Section 8.3.1 *Overview of Key Management* in the TSS of [ST] describes how the TOE peforms key managemnet. This section describes how keys and data are stored and encrypted (wrapped) on the TOE OS, and describes the different key classes. The evaluator reviewed section 8.3.1 and also figure 5 "Key Hierarchy in the TOE OS" in section 8.3.2 and determined that the TOE implements the required secure key storage.

Copyright © 2021 atsec information security corporation

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 5.2.5 *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 subsection *Keys* of section *Certificate Key, and Trust Services* 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.

Copyright © 2021 atsec information security corporation

Summary

Test 1: The evaluator generated keys using OpenSSL (external to the TOE), loaded them into the TOE's keychain, and verified that their use 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 WAP, and connected the TOE to the WLAN network. The evaluator then accessed the test web server and selected the RSA certificate when prompted by the TOE. The evaluator verified that the connection was successful. The evaluator re-performed the operation and verified that if the evaluator declines selecting the certificate, the connection fails.

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

[DTR] 5.2.22

2.3.2.21 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 *Overview of Key Management* in the TSS of [ST] describes key management. Additionally, figure 5 "Key Hierarchy in the TOE OS" in section 8.3.2 *Storage of Persistent Secrets and Private Keys by the Agent* describes the key hierarchy and key materials used by the TOE. This description specifies 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 TOE 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.

Copyright © 2021 atsec information security corporation

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 *Overview of Key Management* in the TSS of [ST] describes key management. Per this section, DEK is encrypted using AES key wrapping in accordance to RFC 3394 (AES Key Wrap) as defined in FCS_STG_EXT.2.2.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.2.22 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

Per FCS_STG_EXT.3, the following ecrypted key is integrity protected: DEKs, KEKs, WPA2 (PSKs), IPsec (client certificates), and Bluetooth keys. Per section 8.3.1 *Overview of Key Management* in the TSS of [ST] the integrity mechanism used for each encrypted key is the one provided by the Key Wrap cipher mode defined in RFC 3394 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.

2.3.2.23 Extended: EAP TLS Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.1-ASE-01

Copyright © 2021 atsec information security corporation

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 TSS of [ST] describes the EAP-TLS and TLS protocols and ciphersuites supported by the TOE. This section describes 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 ciphersuite:

- 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_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- 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 verified that the TSS information above is consistent with the definition of FCS_TLSC_EXT.1.

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 5.3.2 *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 TOE OS Security Framework, which uses the Apple corecrypto Module [Apple silicon, User, Software] which implements TLS 1.2 (along with TLS 1.0 and TLS 1.1 for EAP-TLS) supporting the cipher suites listed in Table 10 (and Table 9 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:

Copyright © 2021 atsec information security corporation

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- 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 5.3.2 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' in the Restrictions Payload; configure which server certificate common names and certificates will be accept by the TOE using the *TLSTrustedServerNames* and *PayloadCertificateAnchorUUID* keys in the Wi-Fi Payload; set the reference identifier as described in [CKTSREF] (chapter *Policies*, section "Obtain policies for establishing trust"); and if needed, add additional CAs using the *EAPClientConfiguration* , *PayloadCertificateArchorUUID*, and *TLSTrustedServerNames* dictionary keys in the Wi-Fi Payload.

Additionally, section 5.3.2 of [CCGUIDE] also references *Appendix: Configuration Profiles* for a sample configuration profile.

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.*

Copyright © 2021 atsec information security corporation

- ***Test 3:*** *The evaluator shall send a server certificate in the TLS connection that the 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.*
  - *(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

Test 1: The evaluator configured the WAP to use each cipher suite defined in [ST] and connected the TOE to the WAP, 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 WAP and verified that the connection was unsuccessful.

Test 3: The evaluator configured the WAP 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 WAP 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 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.

[DTR] 5.2.26

Copyright © 2021 atsec information security corporation

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 TSS of [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 Subject Alternative Name (SAN) contained within the peer certificate (specifically the SAN fields, IP Address, and Wildcard certificate if applicable) to the FQDN of the requested server. The Common Name (CN) is ignored. If the SAN in the certificate does not match the expected SAN for the peer, then the application cannot establish the connection.

Section 8.9.1 also states that 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. Also, WLAN also does 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 5.3.2 *TLS Configuration* provides related guidance for configuring the reference identifier. It states the following:

- Mobile device administrators can use the functions of the Certificate, Key, and Trust Services [CKTSREF] API to manage and manipulate certificates.
- The TOE device implements a set of X.509 policy checks that cannot be altered. If an application wants to enforce additional checks, it can use the API detailed in [CKTSREF].
- When interpreting the term "reference identifier" as the name of the remote peer whose certificate should be validated, the TOE TLS and IKE stacks set the FQDN of the remote peer with the X.509 protocol checker. This operation is hard-coded and cannot be influenced by the user via any API when using TLS or IKE.
- Guidance documentation for setting additional constraints in validating an X.509 certificate can be specified with the rule definitions in the "Certificate, Key, and Trust Services" section of the “Policies” chapter in [CKTSREF].

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.2-ATE-01

Copyright © 2021 atsec information security corporation

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

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 for confirmation or that the connection 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 is not applicable because the TOE mandates the presence of a SAN and does not process the CN.

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

Copyright © 2021 atsec information security corporation

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.

Test 6: The evaluator tested certificates using correct and incorrect URI and DNS names and got the expected results.

Test 7 is not applicable because the TOE does not support CN.

[DTR] 5.2.26

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

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 or denies the connection. When the evaluator performed this test with a trusted CA, the TOE connected without asking any confirmation.

[DTR] 5.2.26

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 *X.509v3 Certificates* describes X.509 certificates used by the TOE and how the TOE uses certificates for different services including for TLS and EAP-TLS authentication. This section also describes how client-side certificates can be installed and how they are used for TLS mutual authentication.

Copyright © 2021 atsec information security corporation

Section 8.9.1.1 *TLS mutual authentication* describes how the TOE uses client-side cerificate for TLS mutual authentication:

- Each application can have more than one client certificate.
- For X.509 certificates, the TOE allows applications to store client certificates in either a P12 file or in a keychain. A P12 file only holds one client certificate, but a keychain can hold multiple client certificates.
- For applications that use a P12 file, there is only one client certificate choice when performing TLS mutual authentication. The application passes this choice to the TLS API.
- For applications that use a keychain, there may be multiple client certificate choices. An application can select the appropriate client certificate from the keychain and pass the selected certificate to the TLS API or it can pass an array of client certificates to the TLS API. The TLS API uses the first certificate in the array and ignores the other certificates in the array when establishing a connection.

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 5.3.2 *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 [DEV_MAN].

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

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.

[DTR] 5.2.26

Copyright © 2021 atsec information security corporation

2.3.2.24 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 *EAP-TLS and TLS* in the TSS of [ST] describes the EAP-TLS and TLS protocols stating the the elliptic curve ciphersuites may use the following elliptic curve extentions by default: secp256r1 (P-256), and secp384r1 (P-384), and secp521r1 (P-521) (SigGen/SigVer only). Curve25519 is also enabled by the TOE OS.

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 5.3.2 *TLS Configuration* provides related guidance for configuring elliptic curves extension for TLS. It states that the supported elliptic curves (i.e., secp256r1, secp384r1, and secp521r1 where secp521r1 is for signature generation and verification only) 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 or administrator.

Also, curve x25519 is 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

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 curve supported by the TOE.

[DTR] 5.2.27

2.3.2.25 Cryptographic Key Storage (FCS_STG_EXT.4)

TSS Assurance Activities

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

Copyright © 2021 atsec information security corporation

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 *Storage of Persistent Secrets and Private Keys by the MDM Agent* in the TSS of [ST] provides table 8 "Summary of keys and persistent secrets used by the MDM Agent" outlining the keys and persistent secrets stored for the agent. The table is duplicated below:

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

This section also states that the MD Agent calls the TOE OS API on the device in order to store keys and persistent secrets in the Keychain.

Guidance Assurance Activities

No assurance activities defined.

Copyright © 2021 atsec information security corporation

Test Assurance Activities

No assurance activities defined.

2.3.2.26 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

Section 8.9.4 *VPN* in the TSS of [ST] describes the VPN functionality. According to this section, 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.

2.3.2.27 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 platforms 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

Copyright © 2021 atsec information security corporation

Section 8.9.4.2 *IPsec General* in the TSS of [ST] states the following:

- IPsec is implemented by the TOE natively, as part of the TOE OS. The TOE implements IPsec in accordance to RFC4301. 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 TOE OS' Core OS Layer.
- The SPD is implemented by the TOE, which is configured as a managed device using either a configuration profile (either manually, through the Apple Configurator 2, or through an MDM). This requires the use of the VPN payload (described in [DEV_MAN]) which 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. The available rules are:
  - PROTECT: allow traffic via tunnel
  - BYPASS: allow traffic outside tunnel
  - DISCARD: drop traffic
- The TOE enforces an Always-On VPN which requires all traffic entering and leaving the TOE platform interfaces to be protected via an IPsec VPN connection. The only exceptions are Captive Networking apps, Voice Mail, Cellular Services, and AirPrint.

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] section 5.3.3 *IPsec Configuration* provides related guidance for IPsec. It states the following:

- The mobile devices implement IPsec natively, as part of their operating system, so any processing of packets used in IPsec communication takes place on the mobile device. IPsec VPN tunnels are configured and controlled by the Network Extension Framework, which is a part of the Core OS Layer of the mobile devices' operating system.
- 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 (PROTECT in the SPD). Any deviations from the default routing behavior must be explicitly specified as exceptions in the Configuration Profile, using the Wi-Fi Payload.

Copyright © 2021 atsec information security corporation

- 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 in the Wi-Fi Payload 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).
- 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.
- To configure exceptions for Voicemail, AirPrint, and CellularServices, the mobile device administrator can specify a ServiceExceptions array in the AlwaysOn dictionary of the VPN payload. Each entry in a ServiceExceptions array lists a ServiceName key and a corresponding Action key. The allowed values for ServiceName and Action can be found in Table 12: Essential Keys for the VPN Payload. For each ServiceName, the corresponding Action can be set to 'Allow' (BYPASS in the SPD) or 'Drop' (DISCARD in the SPD).
- Likewise, exceptions for Captive Network Apps can be defined by using *AllowCaptiveWebSheet*, *AllowAllCaptiveNetworkPlugins*, and *AllowedCaptiveNetworkPlugins* keys in the Configuration Profile as described in [DEV_MAN].

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 evaluator implemented an automatic test for both tests in 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

Copyright © 2021 atsec information security corporation

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 should have been transmitted encrypted, the TOE discards the remote traffic.

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

- Passing traffic encrypted through the tunnel,
- Preventing 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.

[DTR] 7.1.2

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

Section 8.9.4.3 *IPsec Characteristics* in the TSS of [ST] states that the TOE operates IPsec in tunnel mode only, which is consistent with the selection in FCS_IPSEC_EXT.1.2. The evaluator to examine operational guidance for instructions on how to configure the IPsec connection in tunnel mode.

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 5.3.5 *VPN Configuration* and section 5.3.6 *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.*

Copyright © 2021 atsec information security corporation

- ***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

Test 1: The test suite developed covers multiple test aspects. Please see FCS_IPSEC_EXT.1.4 Test 1, which demonstrates that the TOE connects through the VPN with all supported symmetric key cryptographic algorithms and key sizes. Please see FCS_IPSEC_EXT.1.11 Test 2, which demonstrates that the TOE connect through the VPN with all supported asymmetric key cryptographic algorithms.

Test 2-4 are not applicable because [ST] specifies tunnel mode only.

[DTR] 7.1.2

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

Section 8.9.4.2 *IPsec General* in the TSS of [ST] states the following:

The TOE enforces an "always on" configuration meaning that all traffic entering and leaving the TOE platform interfaces is protected via an IPsec VPN connection. The TOE allows a limited number of services to be configured to either not allow (DISCARD) or be sent plaintext (BYPASS). These services include applications that make use of Captive Networking Identifiers, Voice Mail, Cellular Services and AirPrint. All other communications are always sent through the IPsec tunnel (PROTECT within the Security Policy Database (SPD)). In order to set a service to match a PROTECT rule in the SPD, select "Allow traffic via tunnel." "Drop Traffic" will cause that traffic to match a DISCARD rule. "Allow traffic outside tunnel" will create a BYPASS rule for that service.

Any other plaintext data that is received is ignored (discarded). Discarding happens automatically without the need to configure an explicit discard.

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.]

Copyright © 2021 atsec information security corporation

Summary

[CCGUIDE] section 5.3.3 *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 and key values described in table 12 "Essential Keys for the VPN Payload" of [CCGUIDE]. Additionally, the evaluator used the provided guidance to perform independent 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

Test 1: The evaluator established a VPN in tunnel mode to a remote VPN peer via WiFi or LTE and installed 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.

[DTR] 7.1.2

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

Section 8.9.4.3 *IPsec Characteristics* in the TSS of [ST] describes the IPsec characteristics enforced by the TOE in the the evaluated configuration. It specifies the following:

- Symmetric algorithms for IKE and ESP encryption: AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256
- Integrity mechanisms: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.

The evaluator found the above information to be consistent with FCS_IPSEC_EXT.1.4 as well as FCS_COP.1(4).

Copyright © 2021 atsec information security corporation

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] section 5.3.5 *VPN Configuration*, the administrator may use the VPN Payload in the Configuration Profile to configure a traditional system-wide VPN based on IPsec. Table 12 "Essential Keys for the VPN Payload" of [CCGUIDE] describes the related keys and key values for VPN Payload.

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

Test 1: The evaluator established a 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.

[DTR] 7.1.2

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

Section 8.9.4.3 *IPsec Characteristics* in the TSS of [ST] specifies that the TOE implements IKEv2 only which is consistent with FCS_IPSEC_EXT.1.5.

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.

Copyright © 2021 atsec information security corporation

Summary

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

[CCGUIDE] section 5.3.3 *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.*

In other words, IKEv2 is set when configuring Always-On as per instructions in section 5.3.5 *VPN Configuration* of [CCGUIDE]. This involves using the VPN Payload in a Configuration Profile and setting the *VPNType key* to 'AlwaysOn' and *ProtocolType* key to 'IKEv2'.

Test Assurance Activities

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

- ***Test 1:*** *[TD0303] 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 the TOE/platform supports IKEv1 with or without XAUTH, the evaluator shall verify that this test can be successfully repeated with XAUTH enabled and disabled in the manner specified by the operational guidance. If the TOE/platform only supports IKEv1 with XAUTH, the evaluator shall verify that connections not using XAUTH are unsuccessful. If the TOE/platform only supports IKEv1 without XAUTH, the evaluator shall verify that connections using XAUTH are unsuccessful.*
- ***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

Test 1: The evaluator established a VPN in tunnel mode to a remote VPN peer via WiFi and LTE using NAT.

Test 2 is not applicable because [ST] only specifies IKEv2.

[DTR] 7.1.2

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 *IPsec Characteristics* in the TSS of [ST] specifies that the algorithms used for encrypting IKE payload are AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256, which is consistent with FCS_IPSEC_EXT.1.6.

Copyright © 2021 atsec information security corporation

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, AES-GCM-128, and AES-GCM-256 for payload encryption.

[CCGUIDE] Table 12 "Essential Keys for the VPN Payload" specifies the algorithms 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

Test 1: The evaluator established a 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 [ST].

[DTR] 7.1.2

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

Copyright © 2021 atsec information security corporation

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 12 "Essential Keys for the VPN Payload" describes the relevant keys and key values to set in the VPN Payload in a Configuration Profile.

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

Test 1 is not applicable because [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 evaluator found the phase 1 SA (IKE SA) is rekeyed before 24 hours passed.

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 evaluator found the phase 2 SA (IPSEC SA or Child SA) is rekeyed before 6 hours passed.

Test 4 is not applicable because [ST] specifies IKEv2 only.

[DTR] 7.1.2

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.

Copyright © 2021 atsec information security corporation

Summary

Section 8.9.4.3 *IPsec Characteristics* in the TSS of [ST] lists the following DH groups supported by the TOE:

- 14(2048-bit MODP),
- 15(3072-bit MODP),
- 19(256-bit Random ECP), and
- 20 (384-bit Random ECP).

Section 8.9.4.2 *IPsec General* states that configuration of VPN connection setting, such as, authentication method and algorithm selection, is performed by the IPsec VPN client administrator. This implies that DH groups are negotiated based on the configuration chosen by such 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

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

[DTR] 7.1.2

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 estipulations in the requirement.

Summary

Section 8.9.4.5 *IKE* in the TSS of [ST] states that TOE generates the secret value ‘x’ and nonces used in the IKEv2 Diffie-Hellman key exchanges using the TOE platform CAVP validated DRBG specified (as specified in FCS_RBG_EXT.1). The possible lengths of ‘x’ and the nonces are 224, 256, or 384 bits.

Guidance Assurance Activities

No assurance activities defined.

Copyright © 2021 atsec information security corporation

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

Section 8.9.4.4 *Peer authentication* in the TSS of [ST] describes how the TOE performs peer authentication for an IPsec connection. It states the following:

- The TOE supports the use of RSA and ECDSA X.509 certificates for peer authentication.
- During the authentication process, a comparison is made of the distinguished name (DN) contained within the peer certificate (CN and SAN fields) to the DN of the requested server. If the DN in the certificate does not match the expected DN, then the session will not be established.
- If a SAN is present in the certificate, its contents takes precedence over the DN entry. If the SAN does not match that of the peer's identifier, the authentication process fails. If the SAN matches the peer's identifier, the authentication process is successful. Only if the SAN is not present, the DN of the certificate is applied.

The evaluator noted that the ST does not select "pre-shared keys" in FCS_IPSEC_EXT.1.11.

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

Classification: Public
Copyright © 2021 atsec information security corporation

[CCGUIDE] section 5.3.5 *VPN Configuration* provides related guidance on VPN configuration. The configuration is done (by an administrator) via the Configuration Profile using the VPN Payload. table 12 "Essential Keys for the VPN Payload" of [CCGUIDE] describes the keys that must be specified for the payload. Per table 12, the *CertificateType* key must be set to RSA (default), ECDSA P-256, or ECDSA P-384. 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:*** *[TD0379] The evaluator shall configure the TOE to use a private key and associated certificate signed by a trusted CA and shall establish an IPsec connection with the peer.*
- ***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]:*** *[Removed per TD0379]*
- ***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]:*** *[TD0379] 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 Identifer (OID) in the DN) and verify that the IKE authentication fails. To demonstrate a comparison of DN values, the evaluator shall change any one of the four DN values 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.*

Copyright © 2021 atsec information security corporation

- ***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

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 and 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 evaluator established a VPN in tunnel mode to a remote VPN peer via WiFi and LTE. The connection is established with the certificate types specified in [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 evaluator "successfully" failed 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 specified in [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 is not applicable because it is removed by TD0379.

Test 5 is not applicable because [ST] specifies certificate-based authentication only.

Test 6: As outlined in [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.

Test 7: The evaluator "successfully" failed 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 specified in [ST]. The authentication with all certificate types fails due to an incorrect DN.

Test 8 is not applicable because the TOE no longer supports CN.

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

Test 10 is not applicable because IP address identifier types are not supported by the TOE.

Test 11 is not applicable because the TOE does not support ID payload.

[DTR] 7.1.2

FCS_IPSEC_EXT.1.12

TSS Assurance Activities

No assurance activities defined.

Copyright © 2021 atsec information security corporation

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

Assurance Activities for this element are tested through Assurance Activities for FCS_IPSEC_EXT.1.11.

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

Assurance Activities for this element are tested through Assurance Activities for FCS_IPSEC_EXT.1.11.

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

Section 8.9.4.5 *IKE* in the TSS of [ST] states the following.

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.

Classification: Public
Copyright © 2021 atsec information security corporation

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].

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

Tests 1, 3, 4: The evaluator established 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 is not applicable because the TOE only allows specifications of ciphers once for a given VPN configuration.

[DTR] 7.1.2

2.3.2.28 Cryptographic Key Generation (Symmetric Keys for WPA2 Connections) (FCS_CKM.1/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

Copyright © 2021 atsec information security corporation

Section 8.9.3 *Wireless LAN (WLAN)* in the TSS of [ST] describes the WLAN protocol. It states the the TOE implements the WLAN protcol as defined in the IEEE 802.11 (2012) as follow:

- Tthe TOE implements the CTR with CBC-MAC protocol (CCMP) with AES (128-bit key) as defined in section 11.4.3 of IEEE 802.11 (2012)
- The TOE implements AES key wrapping as defined in [SP800-38F] to wrap the Group Temporal Key (GTK), which is sent in an Extensible Authentication Protocol (EAPOL) key frame in message three of the 4-way handshake defined in section 11.6.2 of IEEE 802.11 (2012). Newer device models support AES-CCMP-256 with 256-bit keys following IEEE 802.11ac-2013 as well as AES-GCMP-256 with 256-bit key sizes, respectively, following IEEE 802.11ac-2013
- AES key unwrapping used to unwrap the GTK is performed as described in [SP800-38F] section 6.1, Algorithm 2: W^-1(C), and in section 6.2, Algorithm 4: KW-AD(C).
- PRF-384 is implemented as defined in IEEE 802.11-2012, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", section 11.6.1.2. It is implemented in the TOE as part of the WPA implementation and is used for the key generation of AES keys when the Counter Mode CBC-MAC Protocol (CCMP) cipher (defined in section 11.4.3.1 of IEEE 802.11-2012) is used.

In addition, the TOE implements WPA2 Enterprise which received Wi-Fi Alliance certificates listed in Annex B "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.*

Copyright © 2021 atsec information security corporation

- *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

Test 1: The evaluator connected the TOE to a WiFi Access point (AP) for over one hour and verified with Wireshark that the 4-way EAPOL handshake 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.

[DTR] 5.2.2

2.3.2.29 Cryptographic Key Distribution (GTK) (FCS_CKM.2/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 *Wirelss LAN(WLAN)* in the TSS of [ST] describes the WLAN protocol. It states that the AES key unwrapping used to unwrap the GTK is performed as described in [SP800-38F] section 6.1, Algorithm 2: W^-1(C), and in section 6.2, Algorithm 4: KW-AD(C).

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.

Copyright © 2021 atsec information security corporation

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

Performed in conjunction with FCS_CKM.1.1/WLAN 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.

[DTR] 5.2.5

2.3.2.30 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 TSS of [ST] describes the EAP-TLS and TLS protocols. For EAP-TLS, this section states that the TOE supports TLS versions 1.0, 1.1. and 1.2 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 evaluator verified that these ciphersuites are consistent with those specified in FCS_TLSC_EXT.1/WLAN.

The evaluator checked the administrative guidence [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.

Copyright © 2021 atsec information security corporation

Summary

[CCGUIDE] section 5.3.1 *EAP-TLS Configuration* provides guidance related to EAP-TLS. It states that the supported ciphersuites listed in the Table 9 "EAP-TLS Ciphersuites" (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 5.3.2 *TLS Configuration* subsection *Mobile device administrators* provides guidance for configuring CAs by stating that additional CAs (beside those preinstalled with the TOE) can be added using a Configuration Profile with the *EAPClientConfiguration*, *PayloadCertificateArchorUUID*, and *TLSTrustedServerNames* dictionary keys in the Wi-Fi Payload.

Additionally, a sample Configuration Profile for configuring the WLAN is provided in *Appendix: Configuration Profiles* of [CCGUIDE]

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:*

Copyright © 2021 atsec information security corporation

- *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.*
- *[TD0492] [conditional] If DHE or ECDHE cipher suites are supported, modify the signature block in the Server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.*
- *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

Test 1: The evaluator configured the AP to use each cipher suite defined in [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, 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, attempted to connect the TOE to the AP, and verified that it failed.

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

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

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

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

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

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

[DTR] 5.2.28

Copyright © 2021 atsec information security corporation

2.3.3 User data protection (FDP)

2.3.3.1 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 *User Data Protection (FDP)* in the TSS of [ST] describes how the TOE protects user and TSF data.

This section 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 developer 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
- Tracking
- Contacts
- Calendar
- Reminders
- Photos
- Bluetooth
- Local Network
- Microphone
- Speech Recognition
- Cameras
- Health
- Research Sensor & Usage Data

Copyright © 2021 atsec information security corporation

- HomeKit
- Media & Apple Music
- Files and Folders
- Motion & Fitness
- Analytics & Improvements
- Apple Advertising

Futhermore, this section states 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] section 5.4.2 *Restrict Application Access to System Services* states the following:

- Access control to system services in the Core Services layer is hardcoded and thus not configurable by the mobile device 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 mobile device 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

Copyright © 2021 atsec information security corporation

Test 1 is not applicable because [ST] specifies that no services are accessible to any applications.

Test 2 is not applicable because [ST] 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 and was unsuccessful. 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://server using Chrome. 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 and was unsuccessful. 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://server using Safari. The evaluator selected the RSA certificate and verified that the cryptographic details were displayed on the web page, indicating a successful connection.

[DTR] 5.3.1

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 *User Data Protecction (FDP)* in the TSS of [ST] describes how the TOE protects user and TSF data.

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

Classification: Public
Copyright © 2021 atsec information security corporation

- ***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 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.

[DTR] 5.3.1

2.3.3.2 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 *User Data Protecction (FDP)* in the TSS of [ST] describes how the TOE implements data at rest (DAR).

Protection of files The TOE implements the following basic classes and policies:

- **Complete Protection** (a.k.a. class A): Files in this class can only be accessed when the device is unlocked.
- **Protected Unless Open** (a.k.a. class B): Files in this class may need to be written while the device is locked.
- **Protected Until First User Authentication** (a.k.a. class C): Files in this class are protected until the user has successfully authenticated.
- **No Protection** (a.k.a. class D): Files in this class can always be accessed. However, the files themselves are encrypted and decrypted

All data in files is considered private data, because all files are encrypted. Sensitive data is data protected with a class A or class B key because this data is not accessible when the device is locked.

Protection of keychain data

Section 8.4.6 describes how the keychain data is separated into four different categories (similar to the class structure of the file data protection above) in table 10 "Keychain to File-system Mapping". The securityd daemon determines which keychain items each process or application can access.

Copyright © 2021 atsec information security corporation

Table 8 "Summary of keys and persistent secrets used by the Agent" describes the keys and persistent secrets the TOE stores for the agent.

Further details on how the TOE protects data is provided in section 8.3.1 *Overview of Key Management*. In particular, the description after figure 5 "Key Hierarchy in the TOE OS" clearly indicates that all data including all file system items, metadata, files and keychains are encrypted.

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 5.4.1 *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 passcode by using the *forcePin* key in the Passcode Policy Payload via a Configuration Profile. Additionally, a sample Configuration Profile for configuring passcode restrictions is provided in *Appendix: Configuration Profiles* of [CCGUIDE].

Users can check that data protection is enabled on their device in the Settings at *Touch ID and Passcode* on devices without Face ID functionality (models with a Home button) and Face ID and Passcode on devices with Face ID functionality (models without a Home button). A passcode is required to access the device and this enables data protection on the device. No further configuration is required.

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

Test 1: Currently no tool is available that is able to bypass the encryption/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.

Copyright © 2021 atsec information security corporation

- 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, iOS 11, and iOS 12 evaluations. However, 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.

[DTR] 5.3.2

2.3.3.3 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 *Overview of Key Management* in the TSS of [ST] provides detailed description on how the TOE performs key management. This section states that the TOE treats all data equally and encrypts all data in the same way, that is 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.

Copyright © 2021 atsec information security corporation

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

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.

[DTR] 5.3.3

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 *Overview of Key Management* in the TSS of [ST] provides detailed description on how the TOE performs key management. In particular, subsection *NSFileProtectionCompleteUnlessOpen* describes how files can be retrieved on the TOE OS although the TOE 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. The TOE OS specifically uses curve 25519. The TOE OS generates a device-wide asymmetric key for the NSFileProtectionCompleteUnlessOpen file class within the secure enclave. When data is received by TOE OS in the locked state, the TOE OS 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

Copyright © 2021 atsec information security corporation

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 NSFileProtectionCompleetUnlessOpen 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

Please 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 [PP_MD_V3.1] Figure 4 is implemented by the class B key pair.

[DTR] 5.3.3

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 *Overview of Key Management* in the TSS of [ST] provides detailed description on how the TOE performs key management. In particular figure 5 illustrates how the key hierarchy is implemented within the TOE OS. The evaluator verified that the DEKs are encrypted by a key that is chained to the REK. Some keys, like the passcode key, 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.

Copyright © 2021 atsec information security corporation

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 *Overview of Key Management* in the TSS of [ST] provides detailed description on how the TOE performs 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 asymmetric cryptography whenever the device is locked. The asymmetric keys used to wrap these keys are 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 and re-wrapped with a different key.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.3.4 Extended: Subset information flow control (FDP_IFC_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDP_IFC_EXT.1-ASE-01

[TD0596] 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. The evaluator shall verify 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) or needed for the correct functioning of the TOE 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 *VPN* in the TSS of [ST] describes the VPN functionality which states the following:

- IPsec is implemented in the TOE natively, as part of the TOE OS, 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.

Classification: Public
Copyright © 2021 atsec information security corporation

- 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.
- The TOE enforces an “always on” (AlwaysOn) configuration by default meaning that all traffic entering and leaving the TOE platform interfaces is protected via an IPsec VPN connection.
- The TOE allows limited services such as VoiceMail, Cellular Services, and AirPrint to be configured to either not allow (DISCARD) or be sent plaintext (BYPASS).
- The TOE supports separate configurations for cellular and Wi-Fi.

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 5.3.5 *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 5.3.6 *Keys for Configuring Network Protocols* provides table 12 "Essential Keys for the VPN Payload" outlining the keys and their values that the administrator may specified in the VPN Policy Payload. Additionally, a sample Configuration Profile for configuring the VPN is provided in *Appendix: Configuration Profiles* of [CCGUIDE].

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

Test Assurance Activities

Assurance Activity AA-FDP_IFC_EXT.1-ATE-01

- ***Test 1:*** *[TD0596] 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 ensure the test network is capable of sending any traffic identified as exceptions. 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, as well as ensuring that all exception traffic is generated. The evaluator shall verify that the sniffing tool captures traffic generated by these actions, turn off the sniffing tool, and save the session data.*

Copyright © 2021 atsec information security corporation

- *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, modulo the exceptions identified in the SFR (if applicable). For each exception listed in the SFR, the evaluator shall verify that that traffic is allowed outside of the VPN tunnel. 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: (Conditional: If ICMP is not listed as part of the IP traffic needed for the correct functioning of the TOE) 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

Step 1: The evaluator connected the TOE to the WiFi WAP on the test laptop and accessed https://server using 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 WAP, and verified that the VPN connection is automatically established. The evaluator then accessed 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 WAP, 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 assigned to the TOE by the VPN connection) and verified that there was no reply from the TOE.

[DTR] 5.3.4 ([PP_MD_V3.1]) & 7.2.1 ([MOD_VPN_CLI_V2.1])

2.3.3.5 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 *Identification and Authentication (FIA)* describes user authentication including biometric authentication mechanisms supported by the TOE. It states the following:

Classification: Public
Copyright © 2021 atsec information security corporation

- Biometric authentication inputs do not produce feedback to the user unless an input is rejected.
- For Touch ID, 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 fingerprint samples are presented, then 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.
- The time between consecutive authentication attempts, including biometric authentication factors, is at least the time it takes the PBKDF2 function to execute. This is calibrated to be at least 80 milliseconds between consecutive attempts.
- For Touch ID, the TOE enforces a five second delay between repeated failed authentication attempts. When a user exceeds the number of consecutive failed passcode login attempts, the user's partition is erased (by erasing the encryption key). The OS partition is mounted READONLY upon boot and is never modified during the use of the TOE except during a software update or restore.
- Successful authentication attempts are achieved exclusively by a successful key derivation that decrypts the keybag in the SEP with the respective class keys.

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

2.3.3.6 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 *X.509v3 Certificates* in the TSS of [ST] describes how X.509 certificates are used and managed in the TOE. It states the following:

Copyright © 2021 atsec information security corporation

- The Apple certificate which is used for trusted update is installed in ROM during manufacturing. Other certificates used for trusted channels can be imported by a user (if allowed by the policy) or installed using Configuration Profiles.
- 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.
- Code signing certificates need to be assigned by Apple and can be imported into a TOE device.
- To use a root certificate that isn't preinstalled, such as a self-signed root certificate created by the organization managing the TOE, they can be distributed either when reviewed and accepted by the user, using a Configuration Profile, or download from a web site.
- The database containing trust anchors for all certificates is protected via integrity check and write protection.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.3.7 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 *Trusted Path/Channels (FTP)* in the TSS of [ST] is divided into several section where each section describes the protocol supported by the TOE which are EAP-TLS and TLS, Bluetooth, Wireless LAN, and VPN (IPsec). The evaluator verified that these protocols are included in the requirements in the ST (that is the ST does include FCS_HTTPS_EXT.1, FCS_IPSEC_EXT.1, FCS_TLSC_EXT.1, and FIA_BLT_EXT.*).

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

Copyright © 2021 atsec information security corporation

[CCGUIDE] provides table 4 "SFR Configuration Requirements" mapping SFRs and related guidance. Table entry FDP_UPC_EXT.1 {MDF} 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 5.3 *Network protocols*. Guidance for each supported protocol (in the evaluated configuration) is provided in the following subsections:

- TLS/EAL-TLS: [CCGUIDE] sections 5.3.1 and 5.3.2
- HTTPS: section 5.3.2
- IPsec: [CCGUIDE] sections 5.3.3, 5.3.5, 5.3.6
- Bluetooth: [CCGUIDE] sections 5.3.4, 5.6.7

These protocols may be configured via a Configuration Profile described in [DEV_MAN]. 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*: secure transport API for network communication protocols including TLS.
- [COREBT] *Core Bluetooth*: Core Bluetooth framework.

While performing guidance assurance activities pertaining to these protocols, the evaluator also verified that sufficient guidance is provided. 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 5.3.1 and 5.3.2
- HTTPS: section 5.3.2
- IPsec: [CCGUIDE] sections 5.3.3, 5.3.5, and 5.3.6
- Bluetooth: [CCGUIDE] sections 5.3.4 and 5.6.7

These protocols may be configured via a Configuration Profile described in [DEV_MAN].

While performing guidance assurance activities pertaining to these protocols, the evaluator also verified that sufficient guidance is provided. 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

Copyright © 2021 atsec information security corporation

While no explicit test is specified for FDP_UPC_EXT.1.1, the Application Note in [PP_MD_V3.1] states:

> " *The intent of this requirement is that one of the selected protocols is available for use by user applications running on the device for use in connecting to distant-end services that are not necessarily part of the enterprise infrastructure.* "

For the tests specified for FDP_UPC_EXT.1.2, it can be stated that:

- for Test 1, TLS/HTTPS is tested implicitly with all EAP-TLS and HTTPS tests. For Bluetooth, the evaluator made the BlueZ stack on Linux visible, enabled Bluetooth on the TOE and paired both devices.
- for Test 2, TLS/HTTPS is tested implicitly with all EAP-TLS and HTTPS tests. For Bluetooth, the evaluator ensured that the TOE was disconnected and then
  - 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

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 suing 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 displayed that Link Level Encryption is ON (0x01 or 0x02) and that the command "hcitool con" shows the connection as encrypted:

```
$ hcitool con
Connections:
      > ACL 90:8D:6C:5F:44:4D handle 256 state 1 lm MASTER AUTH ENCRYPT
```

[DTR] 5.3.7

2.3.3.8 Full Residual Information Protection (FDP_RIP.2)

TSS Assurance Activities

Assurance Activity AA-FDP_RIP.2-VPN-ASE-01

Requirement met by the platform

Copyright © 2021 atsec information security corporation

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* in the TSS of [ST] states the IPsec is implemented in the TOE natively, as part of the TOE OS, hence the packets are proccessed by the TOE. There is no separate client application that process packets speifically 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

Section 8.9.4.6 *Residual information protection and packet processing* of the TSS states that when a network packet is received, the TCP/IP stack allocates a new buffer in memory of the same size as the incoming packet, then copies the packet into the new buffer, thereby, overwriting the entire allocated buffer. The packet is only referred to by reference/address, not copied. The VPN encrypts and decrypts the packets in-place because the size of the data does not change when using symmetric ciphers.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.3.9 Subset Information Flow Control (FDP_IFC_EXT.1(VPN))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

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

Copyright © 2021 atsec information security corporation

[CCGUIDE] section 5.3.5 *VPN Configuration* provides related guidance on VPN configuration. It states the following:

- In the evaluated configuration, the VPN must be in its Always-On configuration. The Always-On VPN configuration enables 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 *VPNType key* to 'AlwaysOn' in the Configuration Profile.
- For the user, there is no configuration required.
- For the administrator, they may use the VPN Payload to configure systemwide VPN based on IPsec. Further instructions for the administrators to configure the VPN Payload is provided in section 5.3.6 *Keys for Configuring Network Protocols* of [CCGUIDE].

Test Assurance Activities

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

This test is identical to the test performed for FDP_IFC_EXT.1.

[DTR] 5.3.4 ([PP_MD_V3.1]) & 7.2.1 ([MOD_VPN_CLI_V2.1])

2.3.4 Identification and authentication (FIA)

2.3.4.1 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

Copyright © 2021 atsec information security corporation

Section 8.5 *Identification and Authentication (FIA)* in the TSS of [ST] describes the different authentication mechanisms supported by the TOE including a passcode, fingerprint (Touch ID), or facial authentication (Face ID) of which the latter two are biometric mechanisms. Also, this section states the following:

- TOE devices that support Touch ID do not support Face ID and vice versa.
- The passcode and the device's biometric authentication method cannot be combined for two-factor authentication.
- For passcode authentication, the maximum number of consecutive failed attempts is between 2 and 11, where 11 is the default.
- For biometric authentication, if three invalid biometric samples are presented the TOE will offer passcode entry. After five invalid biometric samples are presented passcode authentication is required.
- The number of failed passcode 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.
- Touch ID and Face ID use a separate failed attempt counter from the passcode counter that is not maintained over a power loss or reboot (i.e., the Touch ID and Face ID failed attempt counter is reset after a power loss or reboot).

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 *Identification and Authentication (FIA)* in the TSS of [ST] describes the user identification and authentication. Based on the description, the evaluator determined the following:

The TOE supports three different authentication mechanisms: passcode, fingerprint (Touch ID), and facial (Face ID).

For Touch ID, 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 fingerprint samples are presented, then 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

Devices that support Touch ID do not support Face ID and vice versa. The passcode and the device's biometric authentication method cannot be combined for two-factor authentication.

The number of failed passcode 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.

Copyright © 2021 atsec information security corporation

Touch ID and Face ID use a separate failed attempt counter from the passcode counter that is not maintained over a power loss or reboot (i.e., the Touch ID and Face ID failed attempt counter is reset after a power loss or reboot).

Entering the same incorrect passcode multiple times consecutively causes the passcode failed login counter to increment only once for those multiple attempts even though these are all passcode failed login attempts. Different passcodes must be entered in order for the passcode failed login counter to increment. (This description only applies to the passcode failed login counter and does not apply to biometric authentication.)

In addition, the following behavior applies to biometric authentication methods. A passcode must be supplied for additional security validation under any of the following conditions.

- 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/Medical ID.

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

This assurance activity was performed in conjunction with the previous work unit AA-FIA_AFL_EXT.1-ASE-02 where the evaluator assessed description of both successful and unsuccessful authentication attempts.

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 5.5.4 *Authentication Attempt Configuration* provides guiidance 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 TOE administrator is responsible for performing this function.

The administrator performs this configuration via a Configuration Profile by specifying in the Passcode Policy Payload the *maxFailedAttempts* key with the value between 2 and 11. Additional detail of the Passcode Policy Payload is provided in [CCGUIDE] section 5.5.1 *Passcode Authentication Configuration* . Also, a sample Configuration Profile for configuring passcode restrictions is provided in *Appendix: Configuration Profiles* of [CCGUIDE].

Copyright © 2021 atsec information security corporation

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

Test 1: The evaluator set the maximum number of failed authentication attempts to 3 and provided three different incorrect passcode. The tester 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 a third wrong password. The evaluator then unlocked the device and verified that the data was wiped.

[DTR] 5.4.1

2.3.4.2 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 *Bluetooth* in the TSS of [ST] describes the Bluetooth protocol.

The evaluator verified that this section specifically addresses how a user authorizes Bluetooth pairing through the Bluetooth interface in the settings of the TOE OS. This section also states that during the pairing time, another device (or the TOE OS) 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

Copyright © 2021 atsec information security corporation

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 5.3.4 *Bluetooth Configuration* provides related guidance for configuring Bluetooth which references the related guidance from the iPad user guide , [iPad_UG], 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 (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 device. Bluetooth can also be disassociated via the Control Center.

Additionally, [CCGUIDE] section 5.3.4 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

Copyright © 2021 atsec information security corporation

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 approved the PIN. The evaluator verified that the same PIN is shown no 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 "hcitool con" shows the connection as encrypted:

```
$ hcitool con
Connections:
      > ACL 90:8D:6C:5F:44:4D handle 256 state 1 lm MASTER AUTH ENCRYPT
```

[DTR] 5.4.2

2.3.4.3 Extended: Bluetooth 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 *Bluetooth* in the TSS of [ST] describes the Bluetooth protocol which states the following:

- Manual authorization is implicitly configured since pairing can only occur when explicitly authorized through the Setting>>Bluetooth interface.
- The TOE requires that remote Bluetooth devices support an encrypted connection. Devices that want to pair with the TOE via Bluetooth are required by Apple to use Secure Simple Pairing, which uses ECDH based authentication and key exchange.
- No data can be transferred via Bluetooth until pairing has been completed.
- The TOE supports the L2CAP through an API in the IOBluetoothDevice class.
- The TOE supports RFCOMM through an API in the IOBluetoothRFCOMMChannel class.

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:

Copyright © 2021 atsec information security corporation

- ***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

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. The evaluator started the Bluetooth configuration on the TOE to make the device visible, then made another device visible to the TOE. The evaluator started the pairing process without confirming the PIN and verified that the TOE did not share files (Photos). The evaluator repeated the process but confirmed the PIN and verified that the device was in a "Connected" state.

[DTR] 5.4.3

2.3.4.4 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 *Bluetooth* in the TSS of [ST] describes the Bluetooth protocol which states the following:

- A local database is kept of all Bluetooth device addresses for paired devices which is checked prior to any automatic connection attempt.
- Bluetooth devices may not 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

[TD0468] The evaluator shall ensure that the TSS describes how Bluetooth sessions are maintained such that two devices with the same Bluetooth device address are not simultaneously connected and such that the initial session is not superseded by any following session initialization attempts.

The evaluator shall perform the following test:

- *Step 1: Pair the TOE with a remote Bluetooth device (DEV1) with a known address BD_ADDR. Establish an active session between the TOE and DEV1 with the known address BD_ADDR.*

Classification: Public
Copyright © 2021 atsec information security corporation

- *Step 2: Attempt to pair a second remote Bluetooth device (DEV2) claiming to have a Bluetooth device address matching DEV1 BD_ADDR to the TOE. Using a Bluetooth protocol analyzer, verify that the pairing attempt by DEV2 is not completed by the TOE and that the active session to DEV1 is unaffected.*
- *Step 3: Attempt to initialize a session to the TOE from DEV2 containing address DEV1 BD_ADDR. Using a Bluetooth protocol analyzer, verify that the session initialization attempt by DEV2 is ignored by the TOE and that the initial session to DEV1 is unaffected. Using a Bluetooth protocol analyzer, verify that the session initialization attempt by DEV2 is ignored by the TOE and that the initial session to DEV1 is unaffected.*

Summary

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 Wireshark that the connection failed.

[DTR] 5.4.4

2.3.4.5 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

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.

Copyright © 2021 atsec information security corporation

[DTR] 5.4.5

2.3.4.6 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 *Biometric Authentication* of [ST] which describes biometric authentication. Specifically, sections 8.5.1.1 and 8.5.1.3 explain the accuracy of biometric authentication of 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

Classification: Public
Copyright © 2021 atsec information security corporation

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 *Biometric Authentication* of [ST] which describes biometric authentication of the TOE. The TOE supports fingerprint (Touch ID) or facial recognition authentication (Face ID) Specifically, sections 8.5.1.1 and 8.5.1.3 explain the accuracy of biometric authentication of the TOE. The SAFAR that the TOE is targeting is described in section 8.5.1.3.

Section 8.5 describes how the Face ID and Touch ID interact with the passcode set on the TOE device. A TOE device that supports Face ID does not support Touch ID and vice-versa. A passcode must be supplied in the following 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.
- There have been five unsuccessful attempts to match.
- After initiating power off/Emergency SOS/Medical ID.

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. Past 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.

2.3.4.7 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 *Biometric Sample Quality* of the TSS describes the biometric sample quality used on the TOE for deriving the authentication template enrolled. This includes the following.

For Touch ID:

Copyright © 2021 atsec information security corporation

- 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 TOE OS 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

If the sample quality passes verification during enrollment, the TOE saves it as an enrollment template. When a user authenticates, an authentication template is generated. If a properly formatted template contains unusual data properties, incorrect syntax, low quality, or unrealistic modality, the TOE rejects the template.

The validation of the discussed mechanism is performed regularly for each major TOE OS 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

Copyright © 2021 atsec information security corporation

[CCGUIDE] section 5.5.3 *Biometric Authentication Factors* describes how to enroll biometric authentication factors which explicitly references [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 menu. 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 evaluator enrolled a fingerprint and a face into the device and attempted to unlock the device with another finger and with the nose and mouth covered and verified that both attempts failed. The evaluator then unlocked the device with the correct finger or full face and verified that it succeeds.

[DTR] 5.4.7

2.3.4.8 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 8.5.1.4 *Biometric Sample Quality* 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.

Copyright © 2021 atsec information security corporation

- 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 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

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

Face ID: 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.

Touch ID: 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.

[DTR] 5.4.8

2.3.4.9 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.

Classification: Public
Copyright © 2021 atsec information security corporation

Summary

Section 8.5.1.4 *Biometric Sample Quality* of the TSS describes the biometric sample quality used on the TOE for deriving the authentication template enrolled. It states that if the sample quality passes verification during enrollment, the TOE saves it as an enrollment template. When a user authenticates, an authentication template is generated. If a properly formatted template contains unusual data properties, incorrect syntax, low quality, or unrealistic modality, the TOE rejects the template.

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

No test is specified in [PP_MD_V3.1] for this SFR.

2.3.4.10 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] section 5.5.1 *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 [DEV_MAN]. 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'; *RequireAlphanumberic* to 'Yes' and *minLength* to a value defined by the organization's policy (i.e., greater than zero). Also, a sample Configuration Profile for configuring passcode restrictions is provided in *Appendix: Configuration Profiles* of [CCGUIDE].

Classification: Public
Copyright © 2021 atsec information security corporation

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 evaluator configured the password policy rule using the Apple Configurator 2 to require at least 4 characters and at least 1 complex character. The evaluator loaded the profile onto the TOE, and attempted 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 a more complex value (e.g. "zaq1@") and verified that it is accepted. The evaluator then locked the TOE and unlocked it with the new passcode.

[DTR] 5.4.11

2.3.4.11 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 *Identification and Authentication (FIA)* in the TSS of the [ST] describes the user identification and authentication. Per this section all authentication mechanisms require authentication data (passcode or biometric samples) to be entered by the TOE user.

Also, this section states that the time between consecutive authentication attempts, including biometric authentication factors, is at least the time it takes the PBKDF2 function to execute which is calibrated to be at least 80 milliseconds between consecutive attempts. In addition, for Touch ID, the TOE enforces a delay of 5 seconds between repeated failed authentication.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

Classification: Public
Copyright © 2021 atsec information security corporation

2.3.4.12 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 *Identification and Authentication (FIA)* in the TSS of [ST] describes the three authentication mechanisms supported by the TOE including a passcode, fingerprint (Touch ID), or facial authentication (Face ID) of which the latter two are biometric mechanisms.

Passcode

All passcode entries are obscured by a dot symbol for each character as the user input occurs.

The TOE can enforce the following passcode policy for managed devices:

- The minimum length of the passcode
- The minimum number of special characters a valid passcode must contain
- The maximum number of consecutive failed attempts to enter the passcode (which can be value between 2 and 11, the default is 11)
- The number of minutes for which the device can be idle before it gets locked by the system
- The maximum number of days a passcode can remain unchanged
- The size of the passcode history (the maximum value is 50)

For biometric authentication, a passcode must be supplied for additional security validation under any of the following conditions:

- 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/Medical ID.

When a user exceeds the number of consecutive failed passcode login attempts, the user's partition is erased (by erasing the encryption key). The TOE OS partition is mounted READONLY upon boot and is never modified during the use of the TOE except during a software update or restore.

Touch ID

Devices that support Touch ID do not support Face ID.

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 fingerprint samples are presented, then the device will offer passcode entry. After five invalid biometric samples are presented, passcode authentication is required.

Face ID

Copyright © 2021 atsec information security corporation

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.

In addition, the passcode and the biometric authentication methods cannot be combined for two-factor authentication.

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], the TOE provides password (passcode), fingerprint (Touch ID), and face (Face ID) for user authentication.

[CCGUIDE] provides related guidance for each of the authentication mechanism as follows:

- Passcode: section 5.5.1 *Passcode Authentication Configuration*
- Touch ID: section 5.5.3 *Biometric Authentication Factors*
- Face ID: section 5.5.3 *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

Test 1 & Test 2: The test for the password mechanism is covered by testing for FIA_BMG_EXT.2 and FIA_BMG_EXT.3 since they require the devices to be unlocked with the password to verify the biometric test results. For both biometric mechanisms, Touch ID and Face ID, the evaluator added his appropriate biometric data, locked the device, and confirmed the biometric mechanism could be used to unlock the device.

[DTR] 5.4.13

2.3.4.13 Re-Authentication (FIA_UAU.6)

FIA_UAU.6.1

TSS Assurance Activities

No assurance activities defined.

Copyright © 2021 atsec information security corporation

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

Test 1: The evaluator changed the passcode on the TOE by going to
`Settings -> Touch ID & Passcode -> Change Passcode`
and verified that the old password is required.

Test 2: The evaluator tried to change to change both Touch ID using
`Settings -> Touch ID & Passcode`
and Face ID using
`Settings -> Face ID & Passcode`
and verified that the TOE prompted for a password in both cases.

Test 3 is not applicable because "hybrid" is not selected in FIA_UAU.5.1 in [ST].

[DTR] 5.4.14

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.*

Copyright © 2021 atsec information security corporation

- ***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

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 Touch ID (fingerprint) and then Face ID (face).

Test 3 is not applicable because "hybrid" is not selected in FIA_UAU.5.1 in [ST].

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 (Touch ID) or Face ID.

Test 6 is not applicable because "hybrid" is not selected in FIA_UAU.5.1 in [ST].

[DTR] 5.4.15

2.3.4.14 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 *Identification and Authentication (FIA)* in the TSS of [ST] describes the three authentication mechanisms supported by the TOE including a passcode, fingerprint (Touch ID), or facial authentication (Face ID) of which the latter two are biometric mechanisms.

This section specifically states the following:

- All passcode entries are obscured by a dot symbol for each character as the user input occurs.
- Biometric authentication inputs do not produce feedback to the user unless an input is rejected.

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

Classification: Public
Copyright © 2021 atsec information security corporation

[CCGUIDE] section 5.5.2 *Protected Authentication Feedback* provides related guidance on protected authentication feedback. It states the following:

- All passcode entries are obscured by iOS/iPadOS. 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

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 Touch ID and Face ID and verified that no information pertaining to either the fingerprint or the face is displayed.

[DTR] 5.4.16

2.3.4.15 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 *Overview of Key Management* in the TSS of the [ST] describes key management.

Copyright © 2021 atsec information security corporation

This section states that the user's passcode is used, combined with the UID and the salt (128-bit random salt generated by the SEP described in section 8.2.1 of the TSS), to derive the 256-bit passcode key which is then used to unwrap the user's keybag that holds the class keys for the file system data protection. Only when the unwrapping of the user's keybag is successful is the user considered authenticated. 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

Test 1: The suggested testing cannot be conducted on any Apple device due to the data encryption implementation as described in the test for FDP_DAR_EXT.1. The evaluator verified those steps in the source code of the TOE. This approach has been approved by NIAP in previous iOS evaluations.

Test 2 & Test 3: Please see the test for FCS_CKM_EXT.4. This approach has been approved by NIAP in previous iOS evaluations.

[DTR] 5.4.17

2.3.4.16 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.

Copyright © 2021 atsec information security corporation

Summary

Section 8.5 *Identification and Authentication (FIA)* in the TSS of the [ST] describes user authentication. It states that making emergency calls, answering calls, accessing Medical ID information, using the cameras (unless their use is generally disallowed), and using the flashlights, and using the control center and using the notification center are the functions that do not require user authentication.

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 evaluator performed this test twice, once waiting for the device to lock automatically and a second time by locking the device manually. In both instances, the evaluator verified that the screen is overwritten by non-sensitive data and that the only functions that can be performed are answering calls, making emergency calls, accessing Medical ID (iPhones only), using the camera (unless that use is generally disallowed), using the flashlight, and using Control Center or Notification Center.

[DTR] 5.4.18

2.3.4.17 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 *X.509v3 Certificates* in the TSS of the [ST] describes X.509v3 certificates.

The evaluator reviewed this section and confirmed that the verification of certificates takes place whenever the TOE tries to establish a connection (via IPsec, TLS, or EAP-TLS) using a remote certificate. The process for certificate validation is as follows:

- When attempting to establish a connection using a peer certificate (i.e., a certificate received from the other endpoint), the peer certificate is first checked to ensure it is valid as per RFC 5280. Certificates are validated against the Subject Alternative Name (SAN). Wildcards are supported. The Common Name (CN) is ignored. If the SAN 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

Classification: Public
Copyright © 2021 atsec information security corporation

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

- The TOE, excluding WLAN, uses OCSP for validating the revocation status of certificates. When a connection cannot be established to the OCSP server to determine the revocation status of a certificate, the TOE considers the certificate as not revoked.
- As part of the certificate chain validation, the validity period of each certificate in the chain is verified. If the certificate is marked as an extended validation certificate, the TOE performs an OCSP lookup to verify the validity (revocation status) of the certificate (except for WLAN certificate validation which does not support OCSP). The basicContraints extension and the CA flag are checked. CA certificates must have the basicContraints extension, the CA flag set to TRUE, and include the caSigning purpose. The extendedKeyUsage (EKU) is validated against the rules defined in FIA_X509_EXT.1 (which is a superset of the rules in FIA_X509_EXT.1/WLAN). Finally, the signature of the issuer of the certificate is verified. Only when all checks succeed, the certificate is considered valid and the next certificate in the certificate chain is checked.
- The certificate chain searches for the certificates in the trust store. The trust store is a combination of the trust store delivered with the TOE and the certificates stored in the key chain and marked as trustworthy. Certificates from the trusted store are validated using the previously described checks at the time that they are used. Certificate path validation terminates with a certificate in the trust store.
- TLS is implemented as a stack that can be utilized by third-party applications. The API informs the calling application that the certificate is not valid. For example, Safari (e.g., HTTPS connection) will display a message to the user that the peer certificate validation failed and allow the user to examine the certificate with the option to allow the connection or not.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.1-ATE-01

[TD0523] 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 demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn:*
  - *by establishing a certificate path in which one of the issuing certificates is not a CA certificate,*
  - *by omitting the basicConstraints field in one of the issuing certificates,*
  - *by setting the basicConstraints field in an issuing certificate to have CA=False,*
  - *by omitting the CA signing bit of the key usage field in an issuing certificate, and*
  - *by setting the path length field of a valid CA field to a value strictly less than the certificate path.*

  The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.
- ***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, OCSP, OSCP stapling, or OCSP multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method:*

Copyright © 2021 atsec information security corporation

- *The evaluator shall test revocation of the node certificate.*
- *The evaluator shall also test 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. If OCSP stapling per RFC 6066 is the only supported revocation method, this test is omitted.*
- *The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.*

- ***Test 4:*** *If any OCSP option is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.*
- ***Test 5:*** *The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate (the certificate will fail to parse correctly).*
- ***Test 6:*** *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 7:*** *The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate (the signature on the certificate will not validate).*
- ***Test 8a:*** *(Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.*
- ***Test 8b:*** *(Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall replace the intermediate certificate in the certificate chain for Test 8a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.*

Summary

Test 1: The evaluator verified that each of a number of improper settings in an issuing certificate in a certificate path cause a connection attempt to fail.

1. a certificate is not a CA certificate
2. a certificate with no basicConstraints field
3. setting the basicConstraints field to CA=False
4. omitting the CA signing bit of the key usage field
5. setting the path length field of a valid CA field to a value strictly less than the certificate path

The evaluator also confirmed that establishing a valid certificate path consisting of valid CA certificates allows the connection attempt to succeed. The test where an installed and otherwise trusted CA certificate has the trust removed to verify that the remote server is not authenticated is performed in FCS_HTTPS_EXT.1 Test 3.

Test 2: The evaluator generated a certificate for the server, advanced the date one day so that the certificate is 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 accessed the multiple URLs with Extended Validation (EV) certificates. The evaluator verified that the OCSP requests were found in the pcap files. The revocation test is performed as part of Test 4.

Test 4: The evaluator tested an OCSP server with a revoked certificate and found the expected result in the pcap file. The evaluator tested with an invalid OCSP responder certificate per the test and verified that the certificate was not determined to be revoked because the OCSP response was discarded.

Copyright © 2021 atsec information security corporation

Test 5: The evaluator attempted to deploy a configuration profile containing a modified certificate as described. The TOE returned an error pointing to the invalid certificate.

Test 6: The evaluator attempted to deploy a configuration profile containing a modified certificate as described. The TOE returned an error pointing to the invalid certificate.

Test 7: The evaluator attempted to deploy a configuration profile containing a modified certificate as described. The TOE returned an error pointing to the invalid certificate.

Test 8a: The evaluator created an EC certificate chain to a web server and was able to establish a TLS connection to it from the TOE via the web browser.

Test 8b is not applicable because ECDSA SigGen/SigVer is not selected in [ST].

[DTR] 5.4.19

2.3.4.18 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 *X.509v3 Certificates* in the TSS of the [ST] describes X.509v3 certificates. This section states that every certificate has a certificate type respective to the area in which this certificate is going to be used such as AppleX509Basic, AppleIPsec, and AppleCodeSigning, as described in the administrative guidance [CCGUIDE].

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 *X.509v3 Certificates* in the TSS of the [ST] describes certificates. This section describe the certificate validation as follows:

- When attempting to establish a connection using a peer certificate (i.e., a certificate received from the other endpoint), the peer certificate is first checked to ensure it is valid as per RFC 5280. Certificates are validated against the Subject Alternative Name (SAN). Wildcards are supported. The Common Name (CN) is ignored. If the SAN 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.

Copyright © 2021 atsec information security corporation

- The TOE, excluding WLAN, uses OCSP for validating the revocation status of certificates. When a connection cannot be established to the OCSP server to determine the revocation status of a certificate, the TOE considers the certificate as not revoked.
- As part of the certificate chain validation, the validity period of each certificate in the chain is verified. If the certificate is marked as an extended validation certificate, the TOE performs an OCSP lookup to verify the validity (revocation status) of the certificate (except for WLAN certificate validation which does not support OCSP). The basicContraints extension and the CA flag are checked. CA certificates must have the basicContraints extension, the CA flag set to TRUE, and include the caSigning purpose. The extendedKeyUsage (EKU) is validated against the rules defined in FIA_X509_EXT.1 (which is a superset of the rules in FIA_X509_EXT.1/WLAN). Finally, the signature of the issuer of the certificate is verified. Only when all checks succeed, the certificate is considered valid and the next certificate in the certificate chain is checked.
- The certificate chain searches for the certificates in the trust store. The trust store is a combination of the trust store delivered with the TOE and the certificates stored in the key chain and marked as trustworthy. Certificates from the trusted store are validated using the previously described checks at the time that they are used. Certificate path validation terminates with a certificate in the trust store.
- TLS is implemented as a stack that can be utilized by third-party applications. The API informs the calling application that the certificate is not valid. For example, Safari (e.g., HTTPS connection) will display a message to the user that the peer certificate validation failed and allow the user to examine the certificate with the option to allow 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

The TOE does not require the administrator to specify the default action. Therefore this assurance activity is not applicable.

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

Copyright © 2021 atsec information security corporation

Test 1: The evaluator re-performed the test for FCS_TLSC_EXT.1 to show the correct operation of the TOE connecting to a non-TOE entity. The evaluator then set up a packet filter to drop packets sent from the TOE to the OCSP server, thus preventing the TOE from completing a request. The evaluator observed the web browser established the connection without asking the user to trust the certificate, as expected.

[DTR] 5.4.20

2.3.4.19 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] section 5.5.6 *X.509 Certificate Configuration* references "Certificate, Key, and Trust Services Reference" [CKTSREF] as the API documentation related to certificate validation, e.g., section *Trust*.

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 evaluator used an application provided by the developer, namely Safari, which requests certificate validation by the TSF. Safari uses the certificate validation logic offered by the TSF via the NSURLSession API (https://developer.apple.com/documentation/foundation/nsurlsession).

The hostap EAP-TLS server is configured to accept only one cipher. OpenSSL supports all selected cipher modes. For each configured cipher, an access to the web server must be performed which must be successful. The evaluator generated a full CA hierarchy and connected the TOE to http://server and https://server for each supported ciphersuite. The evaluator verified that the web connection is successful and, in the case of TLS, that the web page displays the TLS information.

[DTR] 5.4.22

Copyright © 2021 atsec information security corporation

2.3.4.20 Agent Enrollment of Mobile Device into Management (FIA_ENR_EXT.2)

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 *MDM Server Reference ID* in the TSS of [ST] provides table 11 "MDM Server Reference Identifiers" describes 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 4.3.3 *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, 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).
- The MDM Agent communications uses the TOE OS Security Framework as described in section 5.3.2 *TLS Configuration*. Configuring the device's TLS protocol automatically configures the MDM Agent communications. If an additional CA certificate needs to be added to support the MDM Server, see subsection 5.3.2.3.

The evaluator examined these sections and determined that they provided sufficient guidance.

Copyright © 2021 atsec information security corporation

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 evaluation activities verify that the MDM Agent can connect to the MDM Server and validate the MDM Server's certificate.

Summary

The evaluator went on the TOE's
`Settings -> General -> Device Management -> Remote Management -> More Details`
and verified that the Device Identity Certificate contains the issuers certificate. In Device Management, the evaluator verified that the URL indeed points to the OSX server.

[DTR] 6.3.1

2.3.4.21 Port Access Entity Authentication (FIA_PAE_EXT.1)

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

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 restored 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 then created an incorrect certificate for the server and verified that the TOE could not connect to the network.

Classification: Public
Copyright © 2021 atsec information security corporation

[DTR] 5.4.10

2.3.4.22 X.509 Certificate Validation (EAP-TLS) (FIA_X509_EXT.1/WLAN)

TSS Assurance Activities

Assurance Activity AA-FIA_X509_EXT.1-WLAN-ASE-01

[TD0439] The evaluator shall ensure the TSS describes where the check of validity of the EAP-TLS certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.

Summary

Section 8.5.2 *X.509v3 Certificates* of [ST] describes X.509 certificates. It states the following:

- When attempting to establish a connection using a peer certificate (i.e., a certificate received from the other endpoint), the peer certificate is first checked to ensure it is valid as per RFC 5280. Certificates are validated against the Subject Alternative Name (SAN). Wildcards are supported. The Common Name (CN) is ignored. If the SAN 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.
- The TOE, excluding WLAN, uses OCSP for validating the revocation status of certificates. This includes EAP-TLS even though EAP-TLS is not required to support certificate revocation. When a connection cannot be established to the OCSP server to determine the revocation status of a certificate, the TOE considers the certificate as not revoked.
- As part of the certificate chain validation, the validity period of each certificate in the chain is verified. If the certificate is marked as an extended validation certificate, the TOE performs an OCSP lookup to verify the validity (revocation status) of the certificate (except for WLAN certificate validation which does not support OCSP). The basicContraints extension and the CA flag are checked. CA certificates must have the basicContraints extension, the CA flag set to TRUE, and include the caSigning purpose. The extendedKeyUsage (EKU) is validated against the rules defined in FIA_X509_EXT.1 (which is a superset of the rules in FIA_X509_EXT.1/WLAN). Finally, the signature of the issuer of the certificate is verified. Only when all checks succeed, the certificate is considered valid and the next certificate in the certificate chain is checked.
- The certificate chain searches for the certificates in the trust store. The trust store is a combination of the trust store delivered with the TOE and the certificates stored in the key chain and marked as trustworthy. Certificates from the trusted store are validated using the previously described checks at the time that they are used. Certificate path validation terminates with a certificate in the trust store.
- TLS is implemented as a stack that can be utilized by third-party applications. The API informs the calling application that the certificate is not valid. For example, Safari (e.g., HTTPS connection) will display a message to the user that the peer certificate validation failed and allow the user to examine the certificate with the option to allow the connection or not.

Guidance Assurance Activities

No assurance activities defined.

Copyright © 2021 atsec information security corporation

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.1-WLAN-ATE-01

[TD0439] The tests described must be performed in conjunction with the other Certificate Services assurance activities. 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), 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 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 4: 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 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate (the certificate will fail to parse correctly).*
- *Test 6: 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 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate (the signature on the certificate will not validate).*

Summary

This test is included in the testing for FIA_X509_EXT.1.

[DTR] 5.4.19

2.3.4.23 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 TSS of [ST] describes the EAP-TLS and TLS protocols including the ciphersuites supported by the TOE. This section 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. Additional administrative guidance is provided in [CCGUIDE].

Copyright © 2021 atsec information security corporation

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 5.5.6 *X.509 Certificate Configuration* states the following:

- X.509 certificates are configured by an administrator using the keys of the *Certificate Payload* in a Configuration Profile as described in [DEV_MAN].
- 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], the TOE has a iOS/iPadOS Trust Store that contains trusted root certificates presintalled with the iOS/iPadOS. These preinstalled trusted root certificates cannot be modified, are automatically trusted, and do not need to be included when creating a Configuration Profile. New certificates can be added to the Trust Anchor Database or currently installed certificates can be removed.
- WLAN certificate configuration can be found in [DeployRef] and subsection "EAPClientConfiguration Dictionary" of the Wi-Fi section of [DEV_MAN]. In particular, when configuring the TOE devices to utilize EAP-TLS as part of a WPA2 protected Wi-Fi network, 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 described in [DEV_MAN].
- To configure the TOE devices to reject untrusted certificates, the administrator can use the TLSAllowTrustExceptions dictionary key in the Wi-Fi Payload EAPClientConfiguration Dictionary of the Configuration Profile which enforces that untrusted certificates are not accepted and the authentication fails if such untrusted certificates are presented.
- To enforce the verification of the server name defined with the X.509 certificate during the WPA-EAP handshake between the TOE device and the remote access point, the policy must contain the server name to be expected in the certificate with the TLSTrustedServerNames dictionary key in the Wi-Fi Payload EAPClientConfiguration Dictionary of the Configuration Profile.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-WLAN-ATE-01

Classification: Public
Copyright © 2021 atsec information security corporation

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

This test is not required due to TD0517.

2.3.5 Security management (FMT)

2.3.5.1 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 *Configuration Profiles* in the TSS of [ST] describes the TOE management using configuration profiles.

The evaluator identified from table 5 "Management Functions" of [ST] that the functions 4, 5, 12, 13, 18, 19, 22, are restricted to the users only. 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 TOE administrators or MDM administrators. Depending on the behavior defined in the configuration profile, users might be unable to access, perform, or relax management functions defined in table 5 of [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

Copyright © 2021 atsec information security corporation

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 *Specification of Management Functions (FMT)* describes security management of the TOE. It states that administrative configuration can be specified via the Configuration Profiles deployed with either the Apple Configurator 2, an email message, a web site, over-the-air configuration (as described in [OTAConfig]), or over-the-air configuration via an MDM server. Settings or restrictions defined in these Configuration Profiles would restrict any options the user has. If the Configuration Profiles are protected against removal by the administrator, the user cannot circumvent the restrictions or settings by removing the respective profile. In other words, depending on the behavior defined in the Configuration Profile, users will be unable to access, perform, or relax management functions defined in table 5 "Management Functions" of [ST].

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

Test 1: Numerous previous tests have used Apple Configurator 2 to deploy policies.

Test 2: All configuration options are tested for FMT_SMF_EXT.1. For administrator-related configuration options, Apple Configurator 2 is used.

Test 3: Please see the test for FMT_SMF_EXT.1.

[DTR] 5.5.1

2.3.5.2 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.

Copyright © 2021 atsec information security corporation

Summary

Section 8.6 *Specification of Management Functions (FMT)* describes security management of the TOE.

This section references table 5 "Management Functions" of [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 TOE 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 5.

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 *Identification and Authentication (FIA)* in the TSS of [ST] describes the passcode policy including the length, complexity and lifetime of the passcode configurable by the user authenticated to the device, or by administrator 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 *Identification and Authentication (FIA)* in the TSS of [ST] describes the passcode policy stating that the maximum number of consecutive failed attempts to enter the right passcode can be set to, between 2 and 11 (with 11 being the default), 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 8.6.5 *Radios* in the TSS of [ST] lists the radios available on the TOE which includes:

- Cellular
- Wi-Fi
- Bluetooth

Classification: Public
Copyright © 2021 atsec information security corporation

For information about which radio can be enabled/disabled along with what role can do so, this section refers to table 5 "Management Functions" in FCS_SMF_EXT.1 of [ST] .

For information about frequency ranges at which each radio operates, this section refers to in Annex A "Devices Covered by this Evaluation" of [ST].

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 8.6.6 *Audio and Visual collection devices* in the TSS of [ST] which identifies the cameras and microphones as audio and visual collection devices. This section refers to table 5 "Management Functions" and according to this table, the cameras can be enabled/disabled across device or on a per-app basis and the microphone is enabled/disabled on a per-app basis. Also for both the cameras and microphones, either the user or administrator can enable/disable.

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

The ST selects "denying installation of applications" in function 8 of FMT_SMF_EXT.1.

Section 8.6.2 *Configuration Profiles* in the TSS [ST] describes that administrator can configure via a Configuration Profile to allow or disable installation of apps by a user.

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 8.3.2 *Storage of Persistent Secrets and Private Keys by the Agent* in the TSS of [ST] provides table 8 "Summary of keys and persistent secrets used by the Agent" summarizing the keys/secrets that can be imported and stored on the TOE 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

Copyright © 2021 atsec information security corporation

Section 8.5.2 *X.509v3 Certificates* in the TSS of [ST] describes the different certificates used by the TOE. According to this section X.509 certificates are used for the following purposes:

- The Apple certificate is used for verifying the integrity and authenticity of software updates.
- Other certificates are used for setting up trusted channels including for IPsec, TLS, and EAP-TLS connections.
- Certificates are also used for validating the integrity of Configuration Profiles.
- Certificates are used for code signing.

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 8.6.2 *Configuration Profiles* in the TSS of [ST] describes management functions including those that will be enforced by the enterprise once the device is enrolled. These management functions which include password policy, VPN policy, Wi-Fi policy, allow/disallow specific services (e.g., remote backup) are enforced using Configuration Profiles.

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 *Configuration Profiles* states that configuration profiles can also be deployed such that users are unable to override or remove restrictions set in place by Administrators or MDM Administrators. 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

The ST selects function 1 "specify minimum level of security for each pairing. (for BR/EDR and LE" for Function 18 in FMT_SMF_EXT.1.

Section 8.9.2 *Bluetooth* in the TSS of [ST] describes the Bluetooth protocol. This section states that the TOE supports Bluetooth 4.0, 4.2, and 5.0 with the following profiles:

- Hands-Free Profile (HFP 1.6)

Copyright © 2021 atsec information security corporation

- Phone Book Access Profile (PBAP)
- Advanced Audio Distribution Profile (A2DP)
- Audio/Video Remote Control Profile (AVRCP 1.4)
- Personal Area Network Profile (PAN)
- Human Interface Device Profile (HID)
- Message Access Profile (MAP)

Also, this section describes the pairing process which requires the TOE users to manually (via the TOE device's Bluetooth status menu) pair their device with a remote Bluetooth device. During the pairing time, another device (or the TOE itself) can send a pairing request which is commonly a six-digit number displayed on both sides which must be manually matched by the user. If one device does not support this automatic exchange of PIN, a window for entering a manual PIN is presented. Devices that want to pair with the TOE must use Secure Simple Pairing which uses ECDH-based authentication and key exchange. Connections via Bluetooth is secured using AES-CTR-128 with CCM.

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

The ST selects "Biometric Authentication Factor" for Function 23 in FMT_SMF_EXT.1.

Section 8.6.3 *Biometric Authentication Factors (BAFs)* describes biometric authentication mechanisms supported by the TOE. The TOE can support either fingerprint BAF (Touch ID) or facial recognition BAF (Face ID), but not both. The BAF can be set up (enable/disable) in the security options in the TOE settings, i.e., Touch ID & Passcode menu or Face ID & Passcode menu, depending on the TOE device model.

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

The ST does not include Function 24 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-15

Classification: Public
Copyright © 2021 atsec information security corporation

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

The ST does not include Function 25 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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

The ST does not include Function 29 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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

The ST does not include Function 31 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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

The ST does not include Function 34 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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

The ST does not include Function 35 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

Assurance Activity AA-FMT_SMF_EXT.1-ASE-20

Copyright © 2021 atsec information security corporation

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 *Lock Screen / Access Banner Display* in the TSS of [ST] states that the banner can be defined using an image that is presented during the lock screen. As this banner is an image there are no character limitations, information is restricted to what can be included in an image appropriate to the TOE 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

The ST does not include Function 39 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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

The ST does not include Function 40 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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

The ST does not include Function 41 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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 TSS of [ST] describes the Always-On VPN which states the following:

Copyright © 2021 atsec information security corporation

For managed and supervised devices, the TOE must be configured with an 'AlwaysOn' VPN where the organization has full control over device traffic by tunneling all IP traffic back to the organization using an Internet Key Exchange (IKE) v2 based IPsec tunnel. A specific set of configuration key values dedicated to the VPN type 'AlwaysOn' allows the specification of the interfaces (cellular and/or Wi-Fi) for which the VPN is 'AlwaysOn' (default is: for both cellular and Wi-Fi), the specification of exceptions from this service (only VoiceMail, AirPrint, and CellularServices can be listed as exceptions), and the definition of exceptions for Captive networking (if any). The TOE supports separate configurations for cellular and Wi-Fi.

For IKEv2, the configuration contains the following (among other items).

- The IP address or hostname of the VPN server
- The identifier of the IKEv2 client in one of the supported formats
- The authentication method (shared key or certificate))
- The server certificate (for certificate-based authentication)
- If extended authentication is enabled
- The encryption algorithm (allows for AES-CBC-128, AES-CBC-256 (default), AES-GCM-128, and AES-GCM-256).
- The integrity algorithm (allows for SHA1-160, SHA2-256 (default), SHA2-384, SHA2-512).

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

The ST does not include Function 46 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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

The ST does not include Function 47 in FMT_SMF_EXT.1 therefore this assurance activity is not applicable.

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

Copyright © 2021 atsec information security corporation

Guidance to perform each management function is provided throughout [CCGUIDE], notably in section 3.8 *Security Management Configuration* , which has been assessed by the evaluator while performing other guidance assurance activities. Also, [CCGUIDE] section 3.8 contains table 5 "Required Mobile Device Management Functions" which includes management functions from table 5 "Management Functions" of [ST] and management functions specific to Wi-Fi and VPN. For each of these management functions, table 5 provides pointer to the corresponding section in [CCGUIDE] 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 *Radios* of [ST]), the following radios are found in the TOE:

- Cellular
- Wi-Fi
- Bluetooth

[CCGUIDE] section 5.6.7 *Enable/Disable Cellular, Wi-Fi, Wi-Fi Hotspot, Bluetooth, NFC, UWB* 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] mainly refers to the device user guide [iPad_UG] for guidance on how to perform the enable/disable functions of the devices. For example, the user is instructed to enable/disable Wi-Fi by following the instructions provided in the *ipad_UG* in section "Set up and get started » Connect iPad to the Internet".

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], notably in section 3.8 *Security Management Configuration* , which has been assessed by the evaluator while performing other guidance assurance activities. In particular, [CCGUIDE] section 3.8 provides table 5 "Required Mobile Device Management Functions" lists for each management function claimed in Table 5 "Management Functions" of *ST* as well as for Wi-Fi and VPN functionality, the corresponding section in [CCGUIDE] where guidance can be found as well as the user type/rule that can perform it.

Some functions can be enabled/disabled on the devices themselves in accordance to the user guide [iPad_UG], e.g., enable/disable cameras and microphones as described in section 5.6.6 *Enable/Disable Cameras and Microphones* of [CCGUIDE]. Alternatively, a large number of these

Copyright © 2021 atsec information security corporation

functions can be (or must be) performed via a Configuration Profile which is clearly indicated throughout [CCGUIDE], e.g., the ability to enable/disable biometric authentication factors as described in section 5.5.3 *Biometric Authentication Factors* of [CCGUIDE].

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 5 "Management Functions" of [ST], 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] Table 5 "Required Mobile Device Management Functions" refers to section 5.5.6 *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 Database.
- 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, 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 Database.
- Administrator configures X.509 certificates using the *Certificate Payload* in a Configuration Profile.
- Administrators can remove any installed certificate via the MDM protocol using the *RequestType* key with the content “CertificateList” in a Configuration Profile.
- Administrators can modify the Trust Anchor Database 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 5 "Management Functions" of [ST] identifies the management functions that will be enforced by the enterprise once the devices is enrolled. [CCGUIDE] Table 5 "Required Mobile Device Management Functions" provides pointers to the corresponding guidance. While performing related assurance activities the evaluator verified that the information in the TSS is consistent with [CCGUIDE].

Assurance Activity AA-FMT_SMF_EXT.1-AGD-06

Function 14

Copyright © 2021 atsec information security corporation

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 5.6.1 *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 in described in [iPad_UG]. This is of course 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 *DeployRef* 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 be enable/disable display notifications, in the locked state for all notifications.

[CCGUIDE] section 5.6.2 *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 [iPad_UG].
- Administrator can restrict user from viewing past notifications by disabling Notification history setting the *allowLockScreenNotificationsView* key in the Restrictions Payload in a Configuration Profile.
- Likewise, administrator can disable displaying notifications for applications by setting the *ShowInLockScreen* key in a Notification Payload in a Configuration Profile.
- 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

Copyright © 2021 atsec information security corporation

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 tests covering the assurance activities from [PP_MD_V3.1], [MOD_MDM_AGENT_V1.0], [MOD_VPN_CLI_V2.1], and [PP_WLAN_CLI_EP_V1.0].

[DTR] 5.5.2

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:*

Copyright © 2021 atsec information security corporation

- *minimum password length*
- *minimum password complexity*
- *maximum password lifetime*

Summary

Test 1: The evaluator used the Apple Configurator 2 (AC2) to set up 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 in both length and complexity. The evaluator verified that the TOE enforced the password policy for each change.

[DTR] 5.5.2 Test 1

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

[DTR] 5.5.2 Test 2

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 3: For the first test, please see FDP_IFC_EXT.1(MDF). The second and third tests are not applicable because they are not claimed in [ST].

Assurance Activity AA-FMT_SMF_EXT.1-ATE-05

Function 4

Copyright © 2021 atsec information security corporation

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 I 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, set the radio frequency scanner to sweep a defined frequency, 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). The RF scanner was set to a sweeping frequency of 100ms

• The oscilloscope data shows the signal strength data in dBuV. Yet, the MDFPP specifies the data in dBm. The applied conversion formula is: dBm = dBuV - 90 - 20 * log10(sqrt(Z0)) where Z0 is the system impedance which is 50 Ohm. Thus, the conversion of the dBuV value is: dBm = dBuV-107.

[DTR] 5.5.2 Test 4

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.*

Copyright © 2021 atsec information security corporation

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 he 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 from 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.

[DTR] 5.5.2 Test 5

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 confirming that the device requests a password. The evaluator then enrolled the devices into iCloud lost mode and triggered a remote lock of the device. Again, the evaluator confirmed that the device was locked.

[DTR] 5.5.2 Test 6

Assurance Activity AA-FMT_SMF_EXT.1-ATE-08

Function 7

Copyright © 2021 atsec information security corporation

- ***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.

[DTR] 5.5.2 Test 7

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 applications. The evaluator loaded that profile and verified that the App Store was not present anymore.

[DTR] 5.5.2 Test 8

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

For Tests 9 & 10, please 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

For Test 11, please see tests for FCS_TLSC_EXT.1(MDFPP) and FCS_TLSC_EXT.1(WLAN).

Assurance Activity AA-FMT_SMF_EXT.1-ATE-13

Classification: Public
Copyright © 2021 atsec information security corporation

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.

[DTR] 5.5.2 Test 12

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 verified that the enrollment profile is loaded onto the TOE.

[DTR] 5.5.2 Test 13

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.

[DTR] 5.5.2 Test 14

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 using

Copyright © 2021 atsec information security corporation

`Settings -> General -> About -> Version`
and verified that it is an older OS release. The evaluator installed the update for 13.1.2 using
`Settings -> General -> Software Update`
After completion, checking the version again as before, the evaluator confirmed the device now runs 13.1.2.

[DTR] 5.5.2 Test 15

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 Test 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.

[DTR] 5.5.2 Test 17

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.*

Copyright © 2021 atsec information security corporation

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 obtain the device ID of the TOE. The evaluator then disabled Bluetooth on the TOE and used hcidump to watch for connections. The evaluator verified that the TOE was not seen. 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.

Tests 18c through 18j are not applicable because these functions are not selected in [ST].

[DTR] 5.5.2 Test 18

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.*

Copyright © 2021 atsec information security corporation

Summary

The evaluator used AC2 to disable 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.

[DTR] 5.5.2 Test 19

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

On a PC or Mac, the tester formats an external storage device using the APFS encrypted format, setting a password. Then create a directory called "Test Directory" and a text file called "Atsec-Tester-File.txt" on the external storage device. Remove the device from the PC or Mac, insert device into the TOE and open Files app on the TOE. Verify that a password must be entered before the storage device is accessible to the TOE and enter the password. Verify that the directory and file are now visible. Create a new directory called <device-name>. Move the text file from Test Directory to the new directory. Remove device from TOE. Insert device into PC or Mac. Verify that the encrypted file was moved to the new directory.

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

Test 21 is not applicable because Function 21 is not claimed in [ST].

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

Copyright © 2021 atsec information security corporation

Test 22a: The evaluator enabled the location services on the TOE, opened 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.

[DTR] 5.5.2 Test 22

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 Touch ID or Face ID), locked the device, unlocked the device, removed the biometric configuration, and verified that he could not unlock the device anymore using biometric authentication factors.

[DTR] 5.5.2 Test 23

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

Test 24 is not applicable because Function 24 is not claimed in [ST].

Assurance Activity AA-FMT_SMF_EXT.1-ATE-27

Function 25 [conditional]

Copyright © 2021 atsec information security corporation

- ***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

Test 25 is not applicable because Function 25 is not claimed in [ST].

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

Test 26 is not applicable because Function 26 is not claimed in [ST].

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

Test 27 is not applicable because Function 27 is not claimed in [ST].

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 for application data. For a general wipe, FCS_CKM_EXT.4 demonstrates that all data is wiped on the device.

[DTR] 5.5.2 Test 28

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:*

Classification: Public
Copyright © 2021 atsec information security corporation

- *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

Test 29 is not applicable because Function 29 is not claimed in [ST].

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

Please see the test for 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

Test 31 is not applicable because Function 31 is not claimed in [ST].

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

Test 32 is not applicable because Function 32 is not claimed in [ST].

Assurance Activity AA-FMT_SMF_EXT.1-ATE-35

Classification: Public
Copyright © 2021 atsec information security corporation

Function 33 [conditional]

- ***Test 33:*** *The test of this function is performed in conjunction with FPT_TUD_EXT.2.5.*

Summary

Please see the test for FPT_TUD_EXT.2.4.

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

Test 34 is not applicable because Function 34 is not claimed in [ST].

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

Test 35 is not applicable because Function 35 is not claimed in [ST].

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

Test 36: Please see the test for 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

Test 37: Please see the test for 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

Test 38 is not applicable because Function 38 is not claimed in [ST].

Classification: Public
Copyright © 2021 atsec information security corporation

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

Test 39 is not applicable because Function 39 is not claimed in [ST].

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

Test 40 is not applicable because Function 40 is not claimed in [ST].

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

Test 41 is not applicable because Function 41 is not claimed in [ST].

Assurance Activity AA-FMT_SMF_EXT.1-ATE-44

Copyright © 2021 atsec information security corporation

Function 42 [conditional]

- ***Test 42:*** *The test of this function is performed in conjunction with FDP_ACF_EXT.1.2.*

Summary

Test 42 is not applicable because Function 42 is not claimed in [ST].

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

Test 43 is not applicable because Function 43 is not claimed in [ST].

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

Test 44 is not applicable because Function 44 is not claimed in [ST].

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

For all Test 45 parts, please see the test and results for FDP_IFC_EXT.1.1.

Test 45b: 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: Enabling and disabling of WLAN connectivity is performed, including a review of automated reconnect.

Copyright © 2021 atsec information security corporation

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

Test 46 is not applicable because Function 46 is not claimed in [ST].

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

Test 47 is not applicable because Function 47 is not claimed in [ST].

2.3.5.3 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 *Unenrollment* in the TSS of [ST] describes the unenrollment process.

This section references the user guide *Configuration Profile Reference* , [DEV_MAN], 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

Classification: Public
Copyright © 2021 atsec information security corporation

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 evaluator used Profile Manager to go to
`Device Groups -> testgroup1 -> Settings`
and unselect "Prevent unenrollment". The evaluator wiped the device and reinstalled the device as explained in FMT_SMF_EXT.4. On the device, the evaluator went to
`Settings -> General -> Device Management -> Remote Management`
and removed the profile.

[DTR] 5.5.4

2.3.5.4 Agent Trusted Policy Update (FMT_POL_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FMT_POL_EXT.2-MDMA-ASE-01

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

Summary

Section 8.6.2 *Configuration Profiles* in the TSS of [ST] states that candidate policies (defined in configuration profiles) can be received using the Apple Configurator 2 tool, as a file (via an email message, a web page or a files folder), via OTA configuration, or OTA configuration via an MDM server. This section also states that to configuration profiles are digitally signed and encrypted. The configuration profile can specify if a digital signature of the MDM payload is required. When checked as required possible outcomes are:

- signature verified - the payload is then deployed.
- signature failed - the payload is then not deployed.
- no signature present or signature cannot be verified due to other reasons such as missing a CA certificate - the TOE then checks the origin of the payload (i.e., its connection). If the connection to the MDM payload origin (e.g., Apple Configurator 2) is trusted (i.e., the certificates can be validated and its signature checks out), the MDM payload is processed as trusted and deployed.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FMT_POL_EXT.2-MDMA-ATE-01

This evaluation activity is performed in conjunction with the evaluation activity for FIA_X509_EXT.1 and FIA_X509_EXT.2 as defined in the Base-PPs.

Copyright © 2021 atsec information security corporation

- ***Test 1:*** *The evaluator shall perform a policy update from an available configuration interface (such as through a test MDM Server). The evaluator shall verify the update is signed and is provided to the MDM Agent. The evaluator shall verify the MDM Agent accepts the digitally signed policy.*
- ***Test 2:*** *The evaluator shall perform a policy update from an available configuration interface (such as through a test MDM Server). The evaluator shall provide an unsigned and an incorrectly signed policy to the MDM Agent. The evaluator shall verify the MDM Agent does not accept the digitally signed policy.*

Summary

The assurance activities for this requirement are performed in conjunction with the assurance activities for FIA_X509_EXT.1 and FIA_X509_EXT.2 as defined in the base PP.

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.

[DTR] 6.4.1

2.3.5.5 Specification of Management Functions (FMT_SMF_EXT.4)

TSS Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.4-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.

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).

Summary

The ST specifies the following management functions for FMT_SMF_EXT.4:

- enroll in management
- configure whether users can unenroll from management

Since all the mobile devices specified in this [ST] use the TOE OS, there are no differences between supported management functions and policies between the different mobile devices. The supported management functions for the TOE OS are described in [PM_Help]. The evaluator verified that the functions provided in FMT_SMF_EXT.4.1 are all configurable through the configuration profiles defined by the administrator (or MDM administrator) of the TOE.

Section 8.5.3 *MDM Server Reference ID* in the TSS of [ST] describes how to set up the MDM server for enrollment. Section 8.6.1 *Enrollment* describes how a device enrolls with an MDM server.

Section 8.6.1 *Enrollment* 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

Classification: Public
Copyright © 2021 atsec information security corporation

- Device Enrollment Program (DEP), which is an automated and enforced method of automatically enrolling new devices.

Section 8.5.3 also describes that the TOE 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.4.2.

Guidance Assurance Activities

Assurance Activity AA-FMT_SMF_EXT.4-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], [MOD_MDM_AGENT_V1.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1] are provided throughout [CCGUIDE]. A high-level summary is provided in Table 5 "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-01.

Additionally, [CCGUIDE] provides Table 4 "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.4-MDMA-ATE-01

- ***Test 1:*** *In conjunction with the evaluation activities in the Base-PP, the evaluator shall attempt to configure each administrator-provided management function and shall verify that the mobile device executes the commands and enforces the policies.*
- ***Test 2:*** *[TD0491] The evaluator shall configure the MDM Agent authentication certificate in accordance with the configuration guidance. The evaluator shall verify that the MDM Agent uses this certificate in performing the tests for FPT_ITT.1(2) (MDM as Base-PP) or FTP_ITC_EXT.1(2) (MDF as Base-PP).*
- ***Test 3:*** *In conjunction with other evaluation activities, the evaluator shall attempt to enroll the MDM Agent in management with each interface identified in the TSS, and verify that the MDM Agent can manage the device and communicate with the MDM Server.*
- ***Test 4:*** *[conditional] In conjunction with the evaluation activity for FAU_ALT_EXT.2.1, the evaluator shall configure the periodicity for reachability events for several configured time periods and shall verify that the MDM Server receives alerts on that schedule.*
- ***Test 5:*** *[conditional] The evaluator shall design and perform tests to demonstrate that the assigned function may be configured and that the intended behavior of the function is enacted by the mobile device.*

Summary

Test 1: The evaluator re-performed all the tests from FMT_SMF_EXT.1 for all the management function marked as 'Y' in the "Administrator" column in Table 6 of [ST].

Classification: Public
Copyright © 2021 atsec information security corporation

Test 2: For each device, the evaluator deployed the RemoteManagement device management configuration profile and verified the device started its Remote Management configuration.

Test 3: Manual enrollment: The evaluator accessed a specific preset URL containing an enrollment page and clicked to enroll the device. The evaluator accepted and installed the enrollment profile and verified the profile contained all settings and configurations applicable.

Test 3: Manual enrollment with Apple Configurator 2: The evaluator used the Profile Manager web UI to set allow enrollment and then wiped the device. When the first configuration screen appears on the device, the evaluator used Apple Configurator 2 to apply a profile. On the device he stepped through the Setup Assistant process. Upon completion, the evaluator verified the enrollment profile contains all appropriate settings and configurations applicable.

Test 3:Business Manager enrollment: The evaluator used Profile Manager to enable enforced Business Manager enrollment. The evaluator wiped the device, followed the new setup procedure, and verified that username and password are required and cannot be bypassed. After the setup process, the evaluator verified that the TOE is enrolled.

Tests 4 & 5 are not applicable.

[DTR] 6.4.2

2.3.5.6 User Unenrollment Prevention (FMT_UNR_EXT.1)

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 TSS of [ST] describes the unenrollment process for the TOE. It states the to prevent unenrollment by users the key "PayloadRemovalDisallowed" must be specified in the configuration profile. If 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 mobile device administrator to ensure that this key is set appropriately.

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 4.3.4 *Device Un-enrollment Prevention* provides guidance on how configure unenrollment prevention as follows:

- To allow or disallow a device user from removing the MDM Payload from the device, device administrator can set the PayloadRemovalDisallowed key in as described in [DEV_MAN].
- The mobile device must Supervised Mode to lock the MDM Payload to the device.

Classification: Public
Copyright © 2021 atsec information security corporation

- 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

- ***Test 1:*** *If 'prevent the unenrollment from occurring' is selected: The evaluator shall configure the Agent according to the administrative guidance for each available configuration interface, shall attempt to unenroll the device, and shall verify that the attempt fails.*
- ***Test 2:*** *If 'apply remediation actions' is selected: If any configuration allows the user to unenroll, the evaluator shall configure the Agent to allow user unenrollment, attempt to unenroll, and verify that the remediation actions are applied.*

Summary

Test 1: The evaluator used the Profile Manager to select "Prevent unenrollment" and deployed the change to the device. The evaluator then wiped the device and reinstalled as done for Business Manager, tried to remove the enrollment profile, and verified that the device cannot be unenrolled.

Test 2 is not applicable since the functionality is not selected in [ST].

[DTR] 6.4.3

2.3.5.7 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

Section 8.6.7 *VPN Certificate Credentials* in the TSS of [ST] identifies X.509 certificates as client credentials used for IPsec authentication.

Section 8.9.4.4 *Peer authentication* in the TSS of [ST] describes how the TOE performs IPsec authentication using X.509 certificates. During the authentication process, a comparison is made of the DN contained within the peer certificate (CN and SAN fields) to the DN of the requested server. If the DN in the certificate does not match the expected DN, then the session will not be established. Also, 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-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.

Classification: Public
Copyright © 2021 atsec information security corporation

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 5 "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-01

Additionally, [CCGUIDE] provides Table 4 "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 evaluator executed all the test required by [MOD_VPN_CLI_V2.1] which covers all the management functions defined in FMT_SMF_EXT.1.

[DTR] 7.3.1

2.3.5.8 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 5 "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 4 "SFR Configuration Requirements" outlining for each function (expressed in the form of an SFR and its description) whether configuration is required.

Classification: Public
Copyright © 2021 atsec information security corporation

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 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 accessing http://10.0.0.1 is not when using a server certificate with the option "Server Authentication".

[DTR] 5.5.3

2.3.6 Protection of the TSF (FPT)

2.3.6.1 Extended: 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 *Domain Isolation* in the TSS of [ST] states that the 8 bits generated for ALSR (Address space layout randomization) are taken from the application processor 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.*

Copyright © 2021 atsec information security corporation

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 evaluator performed this test for three TOE OS applications, Safari, Mail, and Messenger. For each test, the evaluator did the following:

- start the app on two identical devices
- send a signal to each application to kill them and have them write a crash dump
- synchronize each device with iTunes
- retrieve the crash logs for the applications containing the memory addresses of the text segments and shared libraries

The evaluator verified the memory locations were different in each case.

[DTR] 5.6.1

2.3.6.2 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* in the TSS of [ST] describes memory management unit (MMU). It states that the MMU supports memory address translation using a translation table maintained by the TOE 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.

2.3.6.3 Extended: Anti-Exploitation Services (Overflow Protection) (FPT_AEX_EXT.3)

TSS Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.3-ASE-01

Copyright © 2021 atsec information security corporation

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 *Domain Isolation* in the TSS of [ST] states that stack-based buffer overflow protection is implemented in every sandbox. The protection includes the following:

- Automatic reference counting (ARC): a memory management system that handles the reference count of objects automatically at compile time
- Address space layout randomization (ASLR)
- 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 *Inventory of TSF Binaries and Libraries* in the TSS of [ST] states that the inventory of TSF binaries and libraries is provided in Annex D "Inventory of TSF Binaries and Libraries" of [ST]. Section 8.7.8 states, however, all user space binaries (apps as well as shared libraries) are subject to ASLR.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.6.4 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 *Secure Software Update* and section 8.7.5 *Domain Isolation* in the TSS of [ST] provide relevant description. Per these sections, the entire TOE OS is mounted as read-only which prevents any application or attacker from modifying the system. Unnecessary tools, such as remote login services, aren't included in the system software, and APIs do not allow apps to escalate their own

Copyright © 2021 atsec information security corporation

privileges to modify other apps or the TOE OS itself. Also, 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 *Domain Isolation* in the TSS of [ST] describes domain isolation. It states the following:

- All applications are executed in their own domain or 'sandbox' which isolates the application from other applications and the rest of the system.
- Application are restricted from accessing files stored by other applications or from making changes to the TOE device configuration.
- Each application has a unique home directory for its files, which is randomly assigned when the application is installed.
- If a third-party application needs to access information other than its own, it does so only by using services explicitly provided by the TOE OS.

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 *Domain Isolation* in the TSS of [ST] states the following:

- When a TOE device with cellular capabilities (i.e., a dialer) is in the lock state, only the Emergency Call dialer is available to make a call. The Emergency Call interface does not support the entering of Unstructured Supplementary Service Data (USSD or Man-Machine Interface (MMI) codes. On TOE devices, USSD/MMI codes start with one or more of the following characters: number sign (#), asterisk (*). The Emergency Call interface ignores (i.e., does not send when the Call button is pressed) codes that start with one or more of these characters; thereby, preventing the code actions from executing.
- The TOE also does not support auxiliary boot modes; therefore, the ability to enter codes using auxiliary boot modes does not exist.

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

Copyright © 2021 atsec information security corporation

Section 8.7.5 *Domain Isolation* in the TSS of [ST] explicitly states that the TOE does not support auxiliary boot modes, therefore this work unit is 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 *Secure Boot* in the TSS of [ST] states the following:

- Every step of the boot process contains components that are cryptographically signed by Apple to ensure integrity and that proceed only after verifying the chain of trust. This includes the bootloaders, kernel, kernel extensions, and baseband firmware. This secure boot chain helps ensure that the lowest levels of software aren't tampered with.
- When a TOE device is turned on, its application processor immediately executes code from read-only memory known as Boot ROM. This immutable code, known as the hardware root of trust, is laid down during chip fabrication, and is implicitly trusted. The Boot ROM code contains the Apple Root CA public key, which is used to verify that the iBoot bootloader is signed by Apple before allowing it to load. Because the Apple Root CA public key resides in immutable code, no software (including unverified or unauthorized software) can modify this public key. This is the first step in the chain of trust where each step ensures that the next is signed by Apple. When the iBoot finishes its tasks, it verifies and runs the TOE OS kernel. For devices with an A9 or earlier A-series processor, an additional Low-Level Bootloader (LLB) stage is loaded and verified by the Boot ROM and in turn loads and verifies iBoot.
- A failure of the Boot ROM to load LLB (on older devices) or iBoot (on newer devices) results in the device entering DFU mode. In the case of a failure in LLB or iBoot to load or verify the next step, startup is halted and the device displays the connect to iTunes screen. This is known as recovery mode. In either case, the device must be connected to iTunes through USB and restored to factory default settings.
- The Boot Progress Register (BPR) is used by the SEP to limit access to user data in different modes and is updated before entering the following modes:
    - Recovery Mode: Set by iBoot on devices with Apple A10 system on a chip (SoCs) and higher
    - DFU Mode: Set by Boot ROM on devices with Apple A12 SoC and higher

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.4-ATE-01

Classification: Public
Copyright © 2021 atsec information security corporation

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

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 and verified that no service is available to modify files.

[DTR] 5.6.4

2.3.6.5 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

Per FPT_JTA_EXT.1.1, the TSF must disable access through hardware to JTAG.

Section 8.7.2 *Joint Test Action Group (JTAG) Disablement* in the TSS of [ST] describes how JTAG can be disabled. This section states that the TOE does not implement the actual JTAG protocol but rather a "JTAG-like" interface that is only available on development-fused device via the special Lightning cable. In other words, typical JTAG ports (i.e., Data In, Data Out Clock, etc.) are not present for physical access on the consumer version of the Apple devices (i.e., the TOE devices subject to this evaluation). Thus, this work unit is 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

Classification: Public
Copyright © 2021 atsec information security corporation

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

iPhones and iPads use a 2-stage interface that resembles the functionality of JTAG but does not implement the JTAG protocol.

The Apple development environment provides a JTAG-like interface (technically it is a serial link and no JTAG). 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 is the JTAG-like interface 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. 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 to connect the regular production test devices and verified that the JTAG-like interface was not accessible.
- The evaluator used the same special Lightning cable to connect the development-fused devices and verified that the JTAG-like connection was accessible.

[DTR] 5.6.5

2.3.6.6 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.

Classification: Public
Copyright © 2021 atsec information security corporation

Summary

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

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 the TOE OS by providing the correct passcode). This section 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 TOE OS 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 *Biometric Authentication* in the TSS of [ST] describes the events during biometric authentication. Section 8.3.1 *Overview of Key Management* describes key management operations in the TOE OS, and how KEK and other keys are derived and unwrapped based on the passcode used to unlock the TOE device. According to section 8.5, authentication credentials including the password and biometric samples are not stored on the TOE device. The key derivation, combined with the salt and UID, derive 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.

2.3.6.7 Extended: No Key Transmission (FPT_KST_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FPT_KST_EXT.2-ASE-01

The evaluator shall consult the TSS section of the ST in performing the assurance activities for this requirement. The evaluator shall ensure that the TSS describes the TOE security boundary. The cryptographic module may very well be a particular kernel module, the Operating System, the Application Processor, or up to the entire Mobile Device.

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.

Copyright © 2021 atsec information security corporation

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 *Overview of Key Management* in the TSS of [ST] describes key management. The TOE is the mobile device running the TOE OS which is iPadOS, depending on the TOE device. In particular section 8.3.1.2 *No plaintext key transmission and export* states the following:

- The TOE security boundary is the entire mobile device
- The TOE does not transmit or export plaintext key material outside of the TOE security boundary.
- Plaintext key material is never logged.
- Biometric credential data is confined in the SEP.
- Biometric keying material, enrollment and authentication templates, the features an algorithm uses to perform biometric authentication for enrollment or verification, threshold values, intermediate calculations, and final match scores never leave the SEP.
- Plaintext keys such as plaintext DEKs, KEKs, and keys stored in the secure key storage are never exported.

In addition, evaluator found in section 8.3.1 a description of the power-up process and password authentication within the TOE OS 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 *Identification and Authentication (FIA)* in the TSS of [ST] describes user authentication including BAFs.

According to section 8.5, biometric samples used as authentication credentials are not stored anywhere on the TOE. Successful authentication attempts are achieved exclusively by a successful key derivation that decrypts the keybag in the secure enclave with the respective class key. No intermediate results are sent outside the TOE either; the key derivation process happens locally only. This is further explained in section 8.3.1.2 *No plaintext key transmission and export*, which states the following:

- The TOE does not transmit or export plaintext key material outside of the TOE security boundary.

Copyright © 2021 atsec information security corporation

- Plaintext key material is never logged.
- Biometric credential data is confined in the SEP.
- Biometric keying material, enrollment and authentication templates, the features an algorithm uses to perform biometric authentication for enrollment or verification, threshold values, intermediate calculations, and final match scores never leave the SEP.
- Plaintext keys such as plaintext DEKs, KEKs, and keys stored in the secure key storage are never exported.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.6.8 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 *The Secure Enclave Processor (SEP)* in the TSS of [ST] explicitly states that the main system has no access to other memory areas of the secure enclave and no keys or key material may be exported.

This section also states that The SEP execution environment shares the main random access memory (RAM) with the application processor. The memory controller provides memory separation between the two processors by distinguishing between the origin of memory fetch or store requests performed by the processors. In addition, the memory controller encrypts/obfuscates (using AES-XEX) all SEP data in RAM using a key shared only between the SEP and the memory controller. This adds an additional level of protection between the SEP memory and the application processor. If the memory separation between the two processors were violated to where the application processor could access the SEP RAM, the application processor would only see encrypted data.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.6.9 Extended: Self-Test Notification (FPT_NOT_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FPT_NOT_EXT.1-ASE-01

Classification: Public
Copyright © 2021 atsec information security corporation

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 *Secure Boot* and section 8.7.9 *Self-Tests* provides relevant information.

According to these sections, 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 TOE will display the connect to iTunes screen to transition in the recovery mode. In this mode, the TOE device would require a USB connection to access iTunes in order to perform a 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

Test 1: The evaluator booted the TOE correctly and then booted the TOE after modifying the integrity of the kernel space and user space binaries. The evaluator was then able to use specialized Apple equipment to perform this test and obtain the boot logs that show that the TOE was unable to boot when the binaries are modified, due to integrity test failures.

[DTR] 5.6.9

2.3.6.10 Reliable time stamps (FPT_STM.1)

TSS Assurance Activities

Assurance Activity AA-FPT_STM.1-ASE-01

Classification: Public
Copyright © 2021 atsec information security corporation

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 *Time* in the TSS of [ST] describes the time functionality and its usage. It states the following:

- The following SFRs make use of time:
  - FAU_GEN.1.2 (Audit record time stamp)
  - FIA_TRT_EXT.1 (Authentication throttling)
  - FIA_UAU.7 (Protected authentication feedback)
  - FIA_X509_EXT.1.1 (TOE certificate expiration validation)
  - FIA_X509_EXT.3.1 (Application certificate expiration validation)
  - FMT_SMF_EXT.1.1 Function 1 (Password lifetime)
  - FMT_SMF_EXT.1.1 Function 2 (Screen-lock timeout management)
  - FTA_SSL_EXT.1 (Lock state inactivity timeout)

This section also states that the TOE sets time using either GPS, Apple NTP server, NITZ or the cellular carrier time service. NTP is configured by the administrator.

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] section 5.6.4 *Timestamp Configuration* provides guidance for setting up the date and time. The TOE can use several time sources to automatically update the time: Network, Identity and Time Zone (NITZ), Global Positioning Satellites (GPS), Network Time Protocol (NTP) standards, or the cellular carrier time service. When configured and maintained using one of these time sources, the time may be considered reliable. Only the NTP is configurable by the mobile device administrator. To do this, the administrator uses a Configuration Profile with the Time Server Payload using the *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

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.

[DTR] 5.6.10

Copyright © 2021 atsec information security corporation

2.3.6.11 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 *Self-Tests* in the TSS of [ST] describes self-tests.

It states the following:

- Self-tests are performed by the three cryptographic modules of the TOE: Apple corecrypto Cryptographic Module for Apple Silicon, Apple corecrypto Kernel Cryptographic Module for Silicon, and Apple Secure Key Store Cryptographic Module.
- Self-tests are run by the TOE OS when the TOE device is powered up.
- The TOE performs the FIPS 140-3 power-on self-tests for its cryptographic algorithms, along with a software integrity test using HMAC-SHA-256.
- If any of the defined self-tests fails, the TOE device powers itself off automatically.

More information about the self-tests including the test procedures can be found in sections 8.7.9.1 through 8.7.9.3 where a section is dedicated to each of the three cryptographic modules.

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

Section 8.7.9 *Self-Tests* in the TSS of [ST] describes self-tests. Per this section, all three of the TOE's cryptographic modules perform known answer tests. This information is detailed in the following tables of [ST]:

- Table 12 "Apple corecrypto Cryptographic Module for Apple Silicon Cryptographic Algorithm Tests"
- Table 13 "Apple corecrypto Kernel Cryptographic Module for Apple Silicon Cryptographic Algorithm Tests"
- Table 14 "Apple Secure Key Store Cryptographic Algorithm Tests"

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

Classification: Public
Copyright © 2021 atsec information security corporation

2.3.6.12 Extended: TSF Integrity Checking (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 *Secure Boot* in the TSS of [ST] describes the boot process of the TOE. It states the following:

- Every step of the startup process contains components that are cryptographically signed by Apple to ensure integrity and that proceed only after verifying the chain of trust. This includes the bootloaders, kernel, kernel extensions, and baseband firmware.
- When a TOE device is turned on, its application processor immediately executes code from read-only memory known as Boot ROM. This immutable code, known as the hardware root of trust, is laid down during chip fabrication, and is implicitly trusted. The Boot ROM code contains the Apple Root CA public key, which is used to verify that the iBoot bootloader is signed by Apple before allowing it to load. Because the Apple Root CA public key resides in immutable code, no software (including unverified or unauthorized software) can modify this public key. This is the first step in the chain of trust where each step ensures that the next is signed by Apple. When the iBoot finishes its tasks, it verifies and runs the TOE OS kernel. For devices with an A9 or earlier A-series processor, an additional Low-Level Bootloader (LLB) stage is loaded and verified by the Boot ROM and in turn loads and verifies iBoot.
- A failure of the Boot ROM to load LLB (on older devices) or iBoot (on newer devices) results in the device entering DFU mode. In the case of a failure in LLB or iBoot to load or verify the next step, startup is halted and the device displays the connect to iTunes screen. This is known as recovery mode. In either case, the device must be connected to iTunes through USB and restored to factory default settings.
- The Boot Progress Register (BPR) is used by the SEP to limit access to user data in different modes and is updated before entering the following modes:
    - Recovery Mode: Set by iBoot on devices with Apple A10 system on a chip (SoCs) and higher
    - DFU Mode: Set by Boot ROM on devices with Apple A12 SoC and higher

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 *Secure Boot* in the TSS of [ST] contains the following information:

Copyright © 2021 atsec information security corporation

- Each step of the startup process contains components that are cryptographically signed by Apple to ensure integrity and that proceed only after verifying the chain of trust. This includes the bootloaders, kernel, kernel extensions, and baseband firmware. This secure boot chain helps ensure that the lowest levels of software aren't tampered with.
- Read-only memory known as Boot ROM. This immutable code, known as the hardware root of trust, is laid down during chip fabrication, and is implicitly trusted. The Boot ROM code contains the Apple Root CA public key, which is used to verify that the iBoot bootloader is signed by Apple before allowing it to load. Because the Apple Root CA public key resides in immutable code, no software (including unverified or unauthorized software) can modify this public key. This is the first step in the chain of trust where each step ensures that the next is signed by Apple. When the iBoot finishes its tasks, it verifies and runs the TOE OS kernel.

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

According to section 8.7.5 *Domain Isolation* in the TSS of [ST] the TOE does not support auxiliary boot modes. Therefore this assurance activity is 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

Test 1: The evaluator rebooted the TOE and verified that reboot was successful.

Copyright © 2021 atsec information security corporation

Test 2: The evaluator booted the TOE correctly and then booted the TOE after modifying the integrity of the kernel space and user space binaries. The evaluator was then able to use specialized Apple equipment to perform this test and obtain the boot logs that show that the TOE was unable to boot when the binaries are modified, due to integrity test failures.

Test 3 is not applicable.

[DTR] 5.6.13

2.3.6.13 Extended: TSF Integrity Testing (FPT_TST_EXT.3)

TSS Assurance Activities

Assurance Activity AA-FPT_TST_EXT.3-ASE-01

Testing for this element is performed in conjunction with the assurance activities for FPT_TST_EXT.2.1.

Summary

This assurance activity was performed in conjunction with FPT_TST_EXT.2.1.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.6.14 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.

Copyright © 2021 atsec information security corporation

Summary

The evaluator could query the version of TOE OS and the hardware model of the device in Settings and the version of all installed apps using
`Settings -> General -> iPhone Storage`
and then selecting a specific app.

[DTR] 5.6.15

2.3.6.15 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 *Secure Software Update* in the TSS of [ST] describes the secure update progress. When updates are available, the TOE users Finder application on macOS versions 10.15.0 (Catalina) and higher, through iTunes on macOS versions prior to 10.15.0 and on a PC. All updates are digitally signed with the Apple code signing certificate and users cannot downgrade to older TOE OS versions (prevented by the System Software Authorization process).

The TOE OS software updates can be installed using the Finder application on macOS version 10.15.0 and higher, using iTunes on macOS versions prior to 10.15.0 and on PCs, or over-the-air (OTA) on the device via HTTPS trusted channel. With iTunes and Finder, a full copy of the latest OS 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 macOS Server so that the TOE devices do not need to access Apple servers to obtain the necessary update data. Software updates may also be cached on macOS version 10.13.0 and higher running the built-in caching service (in the client software).

During the upgrade, the TOE device (or Finder/iTunes) connects to and sends the Apple installation authorization server cryptographic measurements to validate. 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. The boot-time chain-of-trust makes sure the signature of the code comes from Apple.

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.

Classification: Public
Copyright © 2021 atsec information security corporation

Summary

Section 8.7.3 *Secure Software Update* in the TSS of [ST] describes the secure update process. Software updates are digitally signed and verified using a hardware-protected asymmetric key used for signature verification. The startup process described in section 8.7.1 *Secure Boot* helps ensure that only Apple-signed code can be installed on a TOE device. The Apple Root CA public key, which resides in immutable code as described in section 8.7.1, is used to verify the signed updates.

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

The ST author does not indicate in the TSS that software updates to system software running on other processors is verified. The evaluator found this to be consistent with FPT_TUD_EXT.2.1 which selects "no other processor system software". Thus, this assurance activity is not applicable.

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

This work unit is a place holder. The tests for FPT_TUD_EXT.2.3 are specificed in the next work unit.

Note that [ST] does indicate that the public key is used for software update digital signature validation, but due to vendor isolation of the function, the certificate does not use the Code Signing purpose.

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.*

Classification: Public
Copyright © 2021 atsec information security corporation

- ***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

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 TOE OS image and verify that the boot is successful.

Test 2: The evaluator signed the boot loader binary with a development RSA key, verified that the newly generated key is listed in the manifest file accompanying the iBEC image, tried to load the image onto the device and observed that the image is rejected by the boot loader. Then the evaluator used an iBEC image that was signed with Apple key, loaded it onto the device, and verified that the image load succeeded.

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: Please see the tests for FPT_TUD_EXT.2.4.

Test 5 is not applicable.

[DTR] 5.6.16

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

This selection is not included in ST, therefore this assurance activity is not applicable.

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.*

Classification: Public
Copyright © 2021 atsec information security corporation

- ***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 digitally signature and shall verify that installation fails. The evaluator shall attempt to install a digitally signed application, and verify that installation succeeds.*

Summary

Tooling does not allow creation of unsigned applications. The evaluator can only test a validly signed app.

Test 1: 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 successfully pushed the application to the TOE using Xcode. 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.

No certificates can be installed/limited that authorize apps. Only the Apple Root CA is allowed.

[DTR] 5.6.16

2.3.6.16 Extended: Trusted Update Verification (FPT_TUD_EXT.3)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.3-ATE-01

Testing for this element is performed in conjunction with the assurance activities for FPT_TUD_EXT.2.3 and FPT_TUD_EXT.4.1.

Summary

This testing is performed in conjunction with the assurance activities for FPT_TUD_EXT.2.3 and FPT_TUD_EXT.4.1 as stated in [PP_MD_V3.1].

2.3.6.17 Extended: Trusted Update Verification (FPT_TUD_EXT.4)

FPT_TUD_EXT.4.1

TSS Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.4.1-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 by a code signing certificate.

Summary

Copyright © 2021 atsec information security corporation

Section 8.5.2 *X.509v3 Certificates* in the TSS of [ST] describes the use of X.509 certificates including as digital signature for code signing applications. It states that all apps must have a valid signature that can be verified by a code signing certificate before they are installed on a device. 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.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.4.1-ATE-01

- ***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 digitally signature and shall verify that installation fails. The evaluator shall attempt to install an application digitally signed with an appropriate certificate, and verify that installation succeeds.*
- ***Test 2:*** *The evaluator shall digitally sign the application with an invalid certificate and verify that application installation fails. The evaluator shall digitally sign the application with a certificate that does not have the Code Signing purpose and verify that application installation fails. This test may be performed in conjunction with the assurance activities for FIA_X509_EXT.1.*
- ***Test 3:*** *If necessary, the evaluator shall configure the device to limit the public keys that can sign application software according to the AGD guidance. The evaluator shall digitally sign the application with a certificate disallowed by the device or configuration and verify that application installation fails. The evaluator shall attempt to install an application digitally signed with an authorized certificate and verify that application installation succeeds.*

Summary

Test 1: Tooling does not allow creation of unsigned applications. The evaluator tested installation of a validly signed app by generating an IPA package, verifying the correct signature, and generating a new certificate using CodeSigning purpose using KeyChain. The evaluator located the binary and invoked a command to sign the application. The evaluator then zipped it back together and successfully pushed the application to the TOE using Xcode. The evaluator removed the application from the TOE and pushed the modified app on the TOE and verified that the app is not usable.

Test 2 is performed in conjunction with Test 1.

Test 3 is not applicable as the signing key cannot be managed by any user.

[DTR] 5.6.18

FPT_TUD_EXT.4.2

TSS Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.4.2-ASE-01

The evaluator shall verify that the TSS describes the mechanism that prevents the TSF from installing software updates that are an older version that the currently installed version.

Summary

Section 8.7.3 *Secure Software Update* in the TSS of [ST] describes the secure update process.

Copyright © 2021 atsec information security corporation

It states that to prevent TOE devices from downgrading to older TOE OS versions, the TOE OS uses the System Software Authorization process that is described in detailed in this section. Also, the SEP utilizes System Software Authorization to ensure the integrity of its software and prevent downgrade installations.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.4.2-ATE-01

The evaluator shall repeat the following tests to cover all allowed software update mechanisms as described in the TSS. For example, if the update mechanism replaces an entire partition containing many separate code files, the evaluator does not need to repeat the test for each individual file.

- ***Test 1:*** *The evaluator shall attempt to install an earlier version of software (as determined by the manufacturer). The evaluator shall verify that this attempt fails by checking the version identifiers or cryptographic hashes of the privileged software against those previously recorded and checking that the values have not changed.*
- ***Test 2:*** *The evaluator shall attempt to install a current or later version and shall verify that the update succeeds.*

Summary

Test 1: Support from Apple is required to push an older version of the OS to the device. The evaluator used Apple tools to push an older version of the OS to the device and verified that it was rejected by the device.

Test 2: The evaluator performed a standard software update on the TOE and verified the new version was downloaded.

[DTR] 5.6.18

2.3.6.18 TSF Self Test (VPN) (FPT_TST_EXT.1(VPN))

TSS Assurance Activities

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

Except for where it is explicitly noted, the evaluator is expected to check the following information regardless of whether the functionality is implemented by the TOE or by the TOE platform.

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. If some of the tests are performed by the TOE platform, the evaluator shall check the TSS to ensure that those tests are identified, and that the ST for each platform contains a description of those tests. Note that the tests that are required by this component are those that support security functionality in this PP-Module, which may not correspond to the set of all self-tests contained in the platform STs.

The evaluator shall examine the TSS to ensure that it describes how the integrity of stored TSF executable code is cryptographically verified 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 shall check to ensure that the cryptographic requirements listed are consistent with the description of the integrity verification process.

Classification: Public
Copyright © 2021 atsec information security corporation

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. 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. The TOE is an entire mobile device hence all self-tests are performed entirely by the TOE itself and not the TOE platform.

Apple corecrypto Module Cryptographic Module for Apple Silicon

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 the TOE OS 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 12 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. If the followings tests succeed, the device continues with its normal power-up process.
- Software / Firmware integrity tests: A software integrity test is performed on the runtime image of the Apple TOE OS corecrypto Cryptographic Module for Apple Silicon. 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. If the followings tests succeed, the device continues with its normal power-up process.
- Conditional tests covering pair-wise consistency tests and SP 800-90A assurance tests.

Apple corecrypto Kernel Cryptographic Module for Apple Silicon

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 TOE OS 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 TOE OS 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. If the tests succeed, the device continues with its normal power-up process.
- Cryptographic algorithm tests specified in Table 13 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. If the test succeeds, the device continues with its normal power-up process.

Copyright © 2021 atsec information security corporation

- 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. If the tests succeed, the device continues with its normal power-up process.
- Cryptographic algorithm tests specified in Table 14 of [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. If the test succeeds, the device continues with its normal power-up process.
- Conditional tests covering pair-wise consistency test and SP800-90 assurance test

In addtion, section 8.7.9 of the TSS provides the following rationale:

> *These tests are sufficient to demonstrate that the TSF is operating correctly since they include each of the cryptographic modules included in the TSF. If the self-tests fail then the TSF will not operate. In addition, the secure boot process begins in hardware and builds a chain of trust through software using the self-tested corecrypto modules, where each step ensures that the next is properly vetted before handing over control to that TSF executable. Secure boot of the devices ensures that the lowest levels of software are not tampered with and that only trusted operating system software from Apple loads at startup. In the devices, security begins in immutable code called the Boot ROM, which is laid down during chip fabrication and known as the hardware root of trust and continues through the loading of the TSF executables.*

The evaluator found the rationale sound and reasonable.

Furthermore, the evaluator verified that the cryptographic requirements listed are consistent with the description of the integrity verification process.

As seen above the TSS describes how each cryptographic module peforms software/firewmare integrity tests. For each cryptographic module if the software/firmware integrity test fails, then the TOE device powers off. If the test succeeds, the device continues with its normal power-up process.

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

Copyright © 2021 atsec information security corporation

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. The TOE is an entire mobile device hence all self-tests are performed entirely by the TOE itself and not the TOE platform and thus no platform-specific guidance is required.

Apple corecrypto Module v11.0 [Apple silicon, User, Software]

This 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 runs all required module self-tests. This application is invoked by the TOE OS startup process upon device power on.

Self-tests performed by this module includes:

- Power-up tests: these tests (as listed in Table 12 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 tests fails the device shuts down automatically. To rerun the self-tests on demand, the user must reboot the device. If the tests succeed, the device continues with its normal power-up process.
- Software / Firmware integrity tests: A software integrity test is performed on the runtime image of the crypto module. 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. If the test succeeds, the device continues with its normal power-up process.
- Conditional tests covering pair-wise consistency tests and SP 800-90A assurance tests.

Apple corecrypto Module v11.0 [Apple silicon, Kernel, Software]

This 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 runs all required module self-tests. This functionality is invoked by the TOE OS Kernel startup process upon device initialization. If the self-tests succeed, the Apple corecrypto Kernel Cryptographic Module for Apple Silicon instance is maintained in the memory of the TOE OS 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 tests fails the device shuts down automatically. To run the self-tests on demand, the user may reboot the device. If the tests succeed, the device continues with its normal power-up process.
- Cryptographic algorithm tests specified in Table 13 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. If the test succeeds, the device continues with its normal power-up process
- Conditional tests covering continuous RNG test, pair-wise consistency test, and SP800-90 assurance test

Apple corecrypto Module v11.0 [Apple silicon, Secure Key Store, Hardware]

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:

Copyright © 2021 atsec information security corporation

- 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 tests fails the device shuts down automatically. To run the self-tests on demand, the user must reboot the device. If the tests succeed, the device continues with its normal power-up process.
- Cryptographic algorithm tests specified in Table 14 of [ST].
- Software / Firmware integrity tests: this test is performed on the runtime image of the TOE OS. 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. If the test succeeds, the device continues with its normal power-up process.
- Conditional tests covering pair-wise consistency test and SP800-90 assurance test.

As stated above, the TSS describes how each cryptographic module peforms software/firewmare integrity tests. For each cryptographic module if the software/firmware integrity test fails, then the TOE device powers off. If the test succeeds, the device continues with its normal power-up process.

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 evaluator booted the TOE correctly and then booted the TOE after modifying the integrity of the kernel space and user space binaries. The evaluator was then able to use specialized Apple equipment to perform this test and obtain the boot logs that show that the TOE was unable to boot when the binaries are modified, due to integrity test failures.

[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.

[DTR] 7.4.1 & 7.4.2

2.3.6.19 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.

Copyright © 2021 atsec information security corporation

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 TSS of [ST] describes the self-tests performed by the TOE. Section 8.7.1 *Secure Boot* describes the secure boot process.

Section 8.7.9 splits into several sections describing the self-test implemented by the following three cryptographic modules of the TOE. Self-tests include the power tests, cryptographic algorithm tests (known-answer tests or pair-wise consistency test), software/firmware integrity tests, critical function tests, and conditional tests.

1. Apple corecrypto Cryptographic Module for Apple Silicon
2. Apple corecrypto Kernel Cryptographic Module for Apple Silicon
3. Apple Secure Key Store Cryptographic Module

These sections clearly explain how the integrity of the libraries (libcorecrypto.dylib, libSystem.dylib and libcommoncrypto.dylib and the runtime image of the TOE OS 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 TOE device recovery.

The evaluator found that section 8.7.9.1 provides an argument that the tests are sufficient:

> *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 the TOE OS.*

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

Section 8.7.9 *Self-Tests* in the [ST] describes the self-tests performed by the TOE. Section 8.7.1 *Secure Boot* describes the secure boot process.

Section 8.7.9 is split into several sections described the different self-tests implemented by the TOE. This includes power-up test, cryptographic algorithms tests (know-answer-tests or pair-wise consistency test), software/firmware integrity tests, and critical function tests. These self-tests are performed for the TOE's Apple corecrypto Module v11.0 [Apple silicon, User, Software], Apple corecrypto Module v11.0 [Apple silicon, Kernel, Software], and Apple corecrypto Module v11.0 [Apple silicon, Secure Key Store, Hardware]. These sections clearly explain how the integrity of the libraries (libcorecrypto.dylib, libSystem.dylib and libcommoncrypto.dylib and the runtime image of the TOE OS 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.

Copyright © 2021 atsec information security corporation

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.

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

For both tests, the evaluator booted the TOE correctly and then booted the TOE after modifying the integrity of the kernel space and user space binaries. The evaluator was then able to use specialized Apple equipment to perform this test and obtain the boot logs that show that the TOE was unable to boot when the binaries are modified, due to integrity test failures.

[DTR] 5.6.12

2.3.7 TOE access (FTA)

2.3.7.1 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 *Device Locking* in the TSS of [ST] states that when the TOE device is locked, the class key for the 'Complete Protection class' is wiped 10 seconds after locking, making 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 8.5 *Identification and Authentication (FIA)* in the TSS of [ST] states Except for answering calls, making emergency calls, accessing Medical ID information, using the cameras (unless their use is generally disallowed), using the flashlight, and using the control center, users need to authenticate using an authentication factor provided above.

Guidance Assurance Activities

Assurance Activity AA-FTA_SSL_EXT.1-AGD-01

Copyright © 2021 atsec information security corporation

The evaluation shall verify that the AGD guidance describes the method of setting the inactivity interval and of commanding a lock.

Summary

[CCGUIDE] section 5.6.3 *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 desirable time. Also, a sample Configuration Profile for configuring passcode restrictions is provided in *Appendix: Configuration Profiles* of [CCGUIDE].

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

Test 1: The evaluator waited for the device to lock automatically and verified that the screen is overwritten with non-sensitive data. The evaluator verified that only unlock, use of the camera, use of information center, display of information, or use of Control Center or Notification Center was possible.

Test 2: The evaluator re-performed Test 1 but locked the device manually instead with the same results.

[DTR] 5.7.1

2.3.7.2 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 *Lock Screen / Access Banner Display* in the TSS of the [ST] states that a banner can be set up to display when the TOE device is locked (displayed on the lock screen).

Guidance Assurance Activities

No assurance activities defined.

Classification: Public
Copyright © 2021 atsec information security corporation

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

Test 1: The evaluator used the Apple Configurator 2 to create a configuration profile with a specific lock screen message, loaded the profile on the device, and verified that the information was displayed on the lock screen.

[DTR] 5.7.2

2.3.7.3 Wireless Network Access (FTA_WSE_EXT.1)

TSS Assurance Activities

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

[TD0470] The evaluator shall examine the TSS to determine that it defines SSIDs as the attribute to specify acceptable networks.

Summary

Section 8.9.1 *EAP-TLS and TLS* in the TSS of [ST] describes the attributes that can be used to specify acceptable networks. This section states that when an application attempts to establish a trusted channel, the TOE will compare the Subject Alternative Name (SAN) contained within the peer certificate (specifically the SAN fields, IP Address, and Wildcard certificate if applicable) to the Fully Qualified Domain Name (FQDN) of the requested server. The Common Name (CN) is ignored. If the FQDN in the certificate does not match the expected SAN for the peer, then the application cannot establish the connection.

Guidance Assurance Activities

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

[TD0470] The evaluator shall examine the operational guidance to determine that it contains guidance for configuring the list of SSID that the WLAN Client is able to connect to.

Summary

Section 5.6.7 *Enable/Disable Cellular, Wi-Fi, Wi-Fi Hotspot, Bluetooth, NFC, UWB* of [CCGUIDE] provides guidance to the adminstrator to configure the list of SSID that the WLAN client is able to connect to. This involves using the *SSID_STR* key in the Wi-Fi Payload in a Configuration Profile and also make sure the *DomainName* key must not be set.

Test Assurance Activities

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

Classification: Public
Copyright © 2021 atsec information security corporation

[TD0470] The evaluator shall also perform the following test:

- *Test 1: The evaluator configures the TOE to allow a connection to a wireless network with a specific SSID. The evaluator also configures the test environment such that the allowed SSID and an SSID that is not allowed are both "visible" to the TOE. The evaluator shall demonstrate that they can successfully establish a session with the allowed SSID. The evaluator will then attempt to establish a session with the disallowed SSID, and observe that the attempt fails.*

Summary

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.

[DTR] 5.7.3

2.3.8 Trusted path/channels (FTP)

2.3.8.1 Extended: Trusted channel Communication (FTP_ITC_EXT.1)

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 *Trusted Path/Channels (FTP)* in the TSS of [ST] describes the trusted channel protocols supported by the TOE including EAP-TLS, TLS, HTTPS, and IPsec, Wi-Fi, and Bluetooth as outlined in table 16 *Protocols used for trusted channels* .

Section 8.9.1 *EAP-TLS and TLS* discusses the EAP-TLS and TLS protocols which states that the TOE supports EAP-TLS using TLS versions 1.0, 1.1, and 1.2 with 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

and TLS version 1.2 with ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,

Copyright © 2021 atsec information security corporation

- 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

Also, certificate pinning is supported by the TOE.

Section 8.9.2 *Bluetooth* describes the Bluetooth protocol which states that 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.

Section 8.9.3 *Wireless LAN* describes the wireless LAN protocol in accordance to IEEE 802.11 (2012) which includes the TOE implementing CRT with CCMP with 128-bit AES. Also, the TOE implements AES key wrapping in accordance to NIST SP 800-38F.

Section 8.9.3 also states that newer device models support AES-CCMP-256 with 256-bit keys following IEEE 802.11ac-2013 as well as AES-GCMP-256 with 256-bit key sizes, respectively, following IEEE 802.11ac-2013.

Section 8.9.4 *VPN* describes the VPN including description of the AlwaysOn VPN, the IPsec and IKE protocols. This section describes the specific modes and algorithms used by IKE2 and IPsec 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):
  - 14(2048-bit MODP),
  - 15(3072-bit MODP),
  - 19(256-bit Random ECP), and
  - 20(384-bit Random ECP)

For IKE, some of the characteristics include the TOE supports IKEv2 Phase 1 and Phase 2 SAs. Also, during negotiation the TOE will only negotiate the configured algorithms which must include an IKEv2/IKE_SA at least that of IKEv2 CHILD_SA.

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 *Secure Software Update* in the TSS of [ST] states that the TOE OS software updates can be performed using the Finder application on macOS version 10.15.0 and higher, using iTunes on macOS versions prior to 10.15.0 and on PCs, or OTA via HTTPS trusted channel.

Guidance Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-AGD-01

Copyright © 2021 atsec information security corporation

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] section 5.3 *Network Protocols* provided related guidance for configuring the following network protocols supported by the TOE:

- EAP-TLS
- TLS
- IPsec
- Bluetooth
- VPN

Guidance pertaining to EAP-TLS is provided in section 5.3.1 *EAP-TLS Configuration* of [CCGUIDE]. Such guidance is already assessed in related assurance activities.

Guidance pertaining to TLS is provided in section 5.3.2 *TLS Configuration* of [CCGUIDE]. Such guidance is already assessed in related assurance activities.

Guidance pertaining to IPsec is provided in partly in section 5.3.3 *IPsec Configuration* and furthermore in section 5.3.5 *VPN Configuration* of [CCGUIDE]. Such guidance is already assessed in related assurance activities.

Guidance pertaining to Bluetooth is provided in section 5.3.4 *Bluetooth Configuration* of [CCGUIDE]. Such guidance is already assessed in related assurance activities.

Guidance pertaining to VPN is provided in section 5.3.5 *VPN Configuration* of [CCGUIDE]. This section particularly 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 *VPNType* setting to 'AlwaysOn' and the key 'ProtocolType' to IKEv2. Section 5.3.6 *Keys for Configuring Network Protocols* outline in Table 12 "Essential Keys for the VPN Payload" the additional keys and key values that must or must not be used in the evaluated configuration.

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

Copyright © 2021 atsec information security corporation

Test 1: The evaluator deployed the the test configuration profile on the TOE and connected the TOE to the WLAN network. For IPsec, the evaluator verified that the TOE connected automatically to the VPN server 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 it 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 the traffic is encrypted.

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: For any other selected protocol (not tested in Test 1, 2, or 3), the tester ensured 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. Since the TOE implements TLS for MDM, VPN, HTTPS, WLAN, we believe that all protocols are covered.

[DTR] 5.8.2 ([PP_MD_V3.1]) & 7.5.1 ([MOD_VPN_CLI_V2.1])

2.3.8.2 Trusted Channel Communication (FTP_ITC_EXT.1(2))

TSS Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-2-MDMA-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

Per FTP_ITC_EXT.1(2), the TOE uses HTTPS to communicate with an MD agent.

Per section 8.5.3 *MDM Server Reference ID*, the MDM Agent-Server communication is over HTTPS. This description is supported by the description provided in section 8.10.2 *MDM Agent Alerts* of the TSS which states that the MDM Agent initiates communication with the MDM Server in response to a push notification by establishing an HTTPS connection to the MDM Server URL.

This description is also consistent with the definition of FTP_TRP.1(2) (trusted path for the MDM agent) which specifies HTTPS as the protocol used for protecting remote TOE administration.

Guidance Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-2-MDMA-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] section 4.3.3 *Configure MDM Agent and MDM Communications* provides information to configure the communications between the MDM Agent and MDM Server. It states as follows:

Copyright © 2021 atsec information security corporation

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 TOE OS Security Framework as described in section 5.3.2 *TLS Configuration* of [CCGUIDE]. Thus, configuring the TOE's TLS protocol as per section 5.3.2 automatically configures the MDM Agent communications. If an additional CA certificate needs to be added to support the MDM Server, information is provided in this section. The evaluator examined these sections and determined that they provide related guidance.

Test Assurance Activities

Assurance Activity AA-FTP_ITC_EXT.1-2-MDMA-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 evaluation activities are associated with the specific protocols.

Summary

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 identified the traffic as either IKE/IPSec or HTTPS/TLS.

[DTR] 6.5.1

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

TSS Assurance Activities

Assurance Activity AA-FTP_TRP.1-2-MDMA-ASE-01

The evaluator shall examine the TSS to determine that the methods of remote enrollment are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of enrollment are consistent with those specified in the requirement, and are included in the requirements in the ST.

Summary

Per FTP_TRP.1(2), The TOE uses HTTPS to provide trusted channel for remote enrollment.

Section 8.5.3 *MDM Server Reference ID* states that the enrollment is initiated by the TOE device and performed over an HTTPS connection. The evaluator verifies that the ST includes the respective requirements FCS_HTTPS_EXT.1 and FCS_TLSC_EXT.1 (as mandated by MDF PP).

Copyright © 2021 atsec information security corporation

Guidance Assurance Activities

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

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

Summary

Section 4.3 *Mobile Device Supervision and Configuration* provides guidance for device enrollment. It states that the methods include:

- The Apple Business Manager (ABM), which provides an automated and enforced method of automatically enrolling new devices
- Using Apple's Profile Manager, which provides a manual method of enrolling mobile devices
- Using the Apple Configurator 2, which provides both automated and manual methods of enrolling mobile devices
- Using Email or a Website, which provides a way to distribute an enrollment profile to a mobile device

Guidance for enrolling via the Apple Business Manager (ABM) is provided in subsection 4.3.1.1 *Apple Business Manager*. Additional guidance is provided in [ABM_Guide], [DEV_MAN].

Guidance for enrolling via the Apple Profile Manager is provided in subsection 4.3.1.2 *Apple Profile Manager*. Additional guidance is provided in [PM_Help].

Guidance for enrolling via the Apple Configurator is provided in section 4.3.1.3 *Apple Configurator*. Additional guidance is provided in subsection [AConfig].

Guidance for enrolling via other methods such as email or website is provided in subsection 4.3.1.4 *Other Methods*.

The evaluator examined these sections including the referenced supporting guides and determined that they provide sufficient guidance for establishing enrollment sessions for each supported method.

Test Assurance Activities

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

For each MDM Agent/platform listed as supported in the ST:

- ***Test 1:*** *The evaluators shall ensure that communications using each specified (in the operational guidance) enrollment method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- ***Test 2:*** *For each method of enrollment supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish enrollment sessions without invoking the trusted path.*
- ***Test 3:*** *The evaluator shall ensure, for each method enrollment, the channel data is not sent in plaintext.*

Further evaluation activities are associated with the specific protocols.

Summary

These tests are included in testing for FTP_ITC_EXT.1(2).

Classification: Public
Copyright © 2021 atsec information security corporation

2.3.8.4 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 (WLAN)* in the TSS of [ST] describes the WLAN protocol implemented by the TOE. Section 8.9.1 *EAP-TLS and TLS* describes the EAP-TLS and TLS protocols and ciphersuites supported by the TOE.

Section 8.9.3 specifies which protocols are implemented according to the IEEE 802.11 (2012) standard for WLAN and which algorithms are supported by the TOE. Also, this section states that newer device models support AES-CCMP-256 with 256-bit keys following IEEE 802.11ac-2013 as well AES-GCMP-256 with 256-bit key sizes, respectively, following IEEE 802.11ac-2013. Additionally, 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 5.5.6 *X.509 Certificate Configuration* 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 TLSTrustedServerNames dictionary key in the Wi-FI Payload EAPClientConfiguration Dictionary of the Configuration Profiles. Additionally, a sample Configuration Profile for configuring the WLAN is provided in chapter 10 *Appendix: Configuration Profiles* of [CCGUIDE].

Additionally, configuration for EAP-TLS is provided in [CCGUIDE] section 5.3.1 *EAP-TLS Configuration* which states that no additional configuration is needed if 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.*

Copyright © 2021 atsec information security corporation

- *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

All tests for FCS_TLSC_EXT.1/WLAN {WLAN} support this test.

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.

[DTR] 5.8.1

Copyright © 2021 atsec information security corporation

2.4 Security Assurance Requirements

2.4.1 Life-cycle support (ALC)

2.4.1.1 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 iPadOS 14: iPads" and references *Annex A: Devices Covered by this Evaluation* which lists the devices that are covered by this evaluation and gives the technical characteristics of each device. These devices use either A9 processor (iPad 9.7-inch), A9X processor (iPad Pro 9.7-inch, iPad Pro 12.9-inch), A10 Fusion processor (iPad 9.7-inch (6th gen), iPad 10.2-inch (7th gen)) A10X Fusion processor (iPad Pro 12.9-inch (2nd gen), iPad Pro 10.5-inch), A12 Bionic (iPad mini (5th gen), iPad Air 10.5-inch (3rd gen), iPad 10.2-inch (8th gen)), A12X Bionic (11-inch iPad Pro, 12.9-inch iPad Pro (3rd gen)), A12Z Bionic (11-inch iPad Pro (2nd gen), 12.9-inch iPad Pro (4th gen)) A14 Bionic (iPad Air (4th gen)).

The guidance documentation, for example, [CCGUIDE] contains TOE reference as Apple iPadOS 14 on iPad.

In addition, for the delivery of the evaluated TOE devices, [CCGUIDE] section 4 *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, 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

Classification: Public
Copyright © 2021 atsec information security corporation

2.4.1.2 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 Integrated Development Environment (IDE) 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 and iPadOS. 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 LLBD debugger consistent with the editor and complier and finding mistakes and offering Fix-its for the code.

The vendor provided "Secure Coding Guide" [SecCodeGuide] [1] 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.*

[1] It should be noted that the "Secure Coding Guide" only references iOS and not iPadOS. However in the [CCGUIDE], chapter 1, it states the following: Although the operating system names of iOS 13 and iPadOS 13 are different, there is no difference between the two operating systems other than the naming.

Copyright © 2021 atsec information security corporation

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], the TOE is Apple iPadOS 14: iPads and claims exact conformance to [CFG_MDF-MDM_AGENT-VPNC_V1.0] (which is comprised of the following 3 documents: [PP_MD_V3.1], [MOD_MDM_AGENT_V1.0], [MOD_VPN_CLI_V2.1]) and [PP_WLAN_CLI_EP_V1.0]. 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 iPad with iPadOS 14 (e.g., [iPad_UG] and [CCGUIDE]).

2.4.1.3 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] section 8.7.3 *Secure Software Update* and section 8.7.4 *Security Updates* describes the secure software update process including security updates 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 OS update notifications on the device and through iTunes, and updates are delivered wirelessly, encouraging rapid adoption of the latest security fixes.*

Copyright © 2021 atsec information security corporation

Mobile device users receive TOE OS update notifications on the mobile device, through the Finder application on macOS versions 10.15.0 (Catalina) and higher, through iTunes on macOS versions prior to 10.15.0 and on a PC. Note that the iTunes application is not available on macOS versions 10.15.0 and higher.

Updates are delivered wirelessly, encouraging rapid adoption of the latest security fixes, as well as downloadable through the aforementioned iTunes and Finder applications.

The startup process described in section 8.7.1 above helps ensure that only Apple-signed code can be installed on a device. The Apple Root CA public key, which resides in immutable code as described in section 8.7.1, is used in verifying the signed updates.

To prevent devices from being downgraded to older versions that lack the latest security updates, the TOE OS 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 OS and exploit a vulnerability that's been fixed in the newer version.

The SEP also utilizes System Software Authorization to ensure the integrity of its software and prevent downgrade installations.

The TOE OS software updates can be installed using the Finder application on macOS version 10.15.0 and higher, using iTunes on macOS versions prior to 10.15.0 and on PCs, or over-the-air (OTA) on the device via HTTPS trusted channel. With iTunes and Finder, a full copy of the latest OS 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 macOS Server so that the TOE devices do not need to access Apple servers to obtain the necessary update data. Software updates may also be cached on macOS version 10.13.0 and higher running the built-in caching service (in the client software).

During a TOE OS upgrade, Finder/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 Exclusive Chip Identification (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 OS 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.

Apple 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

Copyright © 2021 atsec information security corporation

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/

2.4.2 Guidance documents (AGD)

2.4.2.1 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] chapter 7 *Installed Apps* contains Table 18 "Built-in and Presintalled Apps" listing the apps that come with the TOE devices. Built-in apps cannot be removed while preinstalled apps are included with purchased devices but may be removed by the user or administrator.

It is noted that table 18 applies to both iPad and iPhone devices, however the evaluator was only concerned with the app listing for the iPad TOE for this evaluation

Cryptographic modules

[CCGUIDE] section 5.2 *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 corecrypto Module v11.0 [Apple silicon, User, Software] (User Space)
- Apple corecrypto Module v11.0 [Apple silicon, Kernel, Software] (Kernel Space)
- Apple corecrypto Module v11.0 [Apple silicon, Secure Key Store, Hardware]

Section 5.2 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 5.6.10 *Secure Software Updates* describes the procedure for obtaining and verifying updates to the TOE. The process is as follows:

Copyright © 2021 atsec information security corporation

- All TOE updates are digitally signed by Apple.
- The TOE's feature System Software Authorization prevents downgrading of the TOE OS version.
- Updates can be installed using iTunes or over-the-air (OTA) on the device. With iTunes, a full copy of the iOS/iPadOS is downloaded (via TLS) and installed. OTA software updates download only the components required to complete an update.
- Updates may also be installed manually using Finder on macOS versions 10.15.0 (Catalina) and higher, or manually using iTunes on macOS versions prior to 10.15.0 and on PCs. A USB connection between the computer and the device is necessary to perform updates using Finder or iTunes.
- The TOE OS updates can be cached on a local network server running the caching service on macOS server so that TOE devices do not need to access Apple servers to obtain the necessary update data.
- Users can update the TOE OS on their device by following the instructions provided in the user guide [iPad_UG] section *Restart, update, reset, and restore* subsection *Update iPadOS*.

Security functionality provided by the TOE:

[CCGUIDE] section 3.2 *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 5 and 6 are largely structured based on these security functionalities, for example, section 5.5 and section 5.6 are labeled as *Identification & Authentication* and *Security Management*, 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], [MOD_MDM_AGENT_V1.0], [PP_WLAN_CLI_EP_V1.0], and [MOD_VPN_CLI_V2.1] are covered in the guidance/assurance activities.

Furthermore, section 3.9 *Un-evaluated Functionalities* of [CCGUIDE] explicitly lists the security functionalities that are outside the scope of the evaluated configuration.

2.4.2.2 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

Copyright © 2021 atsec information security corporation

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].

2.4.3 Tests (ATE)

2.4.3.1 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 [DTR] specifies all the tests covering the assurance activities from [PP_MD_V3.1], [MOD_MDM_AGENT_V1.0], [MOD_VPN_CLI_V2.1], and [PP_WLAN_CLI_EP_V1.0]. The [DTR] is provided to the certification body but is not published.

2.4.4 Vulnerability assessment (AVA)

2.4.4.1 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.

Copyright © 2021 atsec information security corporation

Summary

The evaluator searched for publicly known vulnerabilities applicable to iPadOS using the following sources.

- Apple security content disclosure statements for releases of iPadOS 14.2 related to this evaluation:
  - https://support.apple.com/kb/HT212003, iPadOS 14.3, released December 14, 2020.
  - https://support.apple.com/kb/HT212146, iPadOS 14.4, released January 26, 2021.
  - https://support.apple.com/kb/HT212221, iPadOS 14.4.1, released March 8, 2021.
  - https://support.apple.com/kb/HT212256, iPadOS 14.4.2, released March 26, 2021.
  - https://support.apple.com/kb/HT212317, iPadOS 14.5, released April 26, 2021.
  - https://support.apple.com/kb/HT212336, iPadOS 14.5.1, released May 3, 2021.
  - https://support.apple.com/kb/HT212528, iPadOS 14.6, released May 24, 2021.
  - https://support.apple.com/kb/HT212601, iPadOS 14.7, released July 21, 2021.
  - https://support.apple.com/kb/HT212623, iPadOS 14.7.1, released July 26, 2021.

  Note: Not all releases of iPadOS contain security fixes for publicly known vulnerabilities. iPadOS release 14.2.1 contained no published security fixes.

- 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:

- ipados ipad
- ipados core tls
- ipados core crypto
- ipados common crypto
- ipados http
- ipados https
- ipados tcp
- ipados ip
- ipados bluetooth
- ipados ipsec
- ipados vpn
- ipados mdm
- ipados mobile
- ipados touchid
- ipados faceid
- broadcom wi-fi

In addition to the lists of fixes published by the vendor, the evaluator performed manual searches on multiple occasions between 2021-01-20 and 2021-02-19, and again on 2021-03-25, 2021-04-27, 2021-05-04, 2021-05-24, 2021-07-12 and 2021-08-30.

Classification: Public
Copyright © 2021 atsec information security corporation

The developer publishes security content disclosure statements providing information about vulnerabilities fixed in each release of iPadOS after 14.2. The developer also provides continuous updates to the TOE, therefore the TOE version tested in this evaluation, iPadOS 14.2, is no longer available. As of the date of this report, the current version is iPadOS 14.7.1. The evaluator's CVE search found no vulnerabilities apart from the ones listed in the security content disclosure statements, all of which have been fixed in subsequent releases of iPadOS.

Copyright © 2021 atsec information security corporation

A Appendixes

A.1 References

ABM_Guide **Apple Business Manager User Guide**
Date October 2020
Location https://support.apple.com/guide/apple-business-manager/what-is-apple-business-manager-apdd344cdd9d/web

AConfig **Apple Configurator 2 User Guide (online)**
Date received 2021
Location https://support.apple.com/guide/apple-configurator-2/welcome/mac

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=0194

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=0244

CCEVS-TD0301 **Updates to Administrator Management and Biometric Authenication**
Date 2018-04-04
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0301

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=0304

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=0305

Copyright © 2021 atsec information security corporation

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=0330

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=0346

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=0347

CCEVS-TD0351 **Additional methods for DEK formation**
Date 2018-09-20
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0351

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=0366

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=0369

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=0378

CCEVS-TD0379 **Updated FCS_IPSEC_EXT.1.11 Tests for VPN Client**
Date 2018-12-20
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0379

CCEVS-TD0404 **Cryptographic selections and updates for use with App PP v1.3**
Date 2019-08-30
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0404

CCEVS-TD0413 **Conformance Claim to allow for a PP-Module**
Date 2019-03-22
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0413

Copyright © 2021 atsec information security corporation

CCEVS-TD0426 **Correction of Inconsistency with RFC7748**
Date 2019-06-24
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0426

CCEVS-TD0439 **EAP-TLS Revocation Checking**
Date 2019-08-29
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0439

CCEVS-TD0468 **Bluetooth pairing vs connection**
Date 2019-11-22
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0468

CCEVS-TD0470 **Wireless Network Restrictions**
Date 2020-01-22
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0470

CCEVS-TD0491 **Update to FMT_SMF_EXT.4 Test 2**
Date 2020-01-15
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0491

CCEVS-TD0492 **TLS-EAP Ciphers and TLS versions for WLAN Client**
Date 2020-01-22
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0492

CCEVS-TD0508 **Conformance Claim to allow for a PP-Module**
Date 2020-02-28
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0508

CCEVS-TD0517 **WLAN Client Corrections for X509 and TLSC**
Date 2020-06-19
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0517

CCEVS-TD0523 **Updates to Certificate Revocation (FIA_X509_EXT.1)**
Date 2020-07-01
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0523

CCEVS-TD0579 **FAU_GEN.1, item 7**
Author(s) NIAP
Date 2021-03-02
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0579

Copyright © 2021 atsec information security corporation

CCGUIDE	**Apple iOS 14: iPhones and Apple iPadOS 14: iPads Common Criteria Configuration Guide** Version 1.0 Date May 2021 File name agd/st_vid11146-agd.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
CFG_MDF-MDM_AGENT-VPNC_V1.0	**PP-Configuration for Mobile Device Fundamentals (MDF), Mobile Device Management(MDM)Agents,and Virtual Private Network (VPN) Clients** Version 1.0 Date 2020-02-28 Location https://www.niap-ccevs.org/MMO/PP/CFG_MDF-MDM_AGENT-VPNC_V1.0.pdf
CKTSREF	**Certificate, Key, and Trust Services (API Reference)** Date 2021 Location https://developer.apple.com/documentation/security/certificate_key_and_trust_services
COREBT	**Core Bluetooth Framework** Date 2021 Location https://developer.apple.com/documentation/corebluetooth
DeployRef	**Deployment Reference for iPhone and iPad** Date 2021 Location https://support.apple.com/guide/deployment-reference-ios/welcome/web
DEV_MAN	**Device Management** Version April 2021 Date received 2021-05-04 Location https://developer.apple.com/documentation/devicemanagement
DTR	**Apple iPadOS 14: iPad Test Plan Detailed Test Report** Version 5.3 Date received 2021-08-31 File name ind/VID11147-iPadOS_14-DTR.pdf
iPad_UG	**iPad User Guide** Author(s) Apple Inc. Version iOS 14.0 Date 2020 Location https://support.apple.com/guide/ipad/welcome/ipados File name agd/210111_iPad_User_Guide.pdf
KEYCHAINPG	**Keychain Services Programming Guide** Date 2021 Location https://developer.apple.com/documentation/security/keychain_services

Classification: Public
Copyright © 2021 atsec information security corporation

LOGGING **Logging**
Date 2021
Location https://developer.apple.com/documentation/os/logging?language=objc

MOD_MDM_AGENT_V1.0 **PP-Module for MDM Agents**
Version 1.0
Date 2019-04-25
Location https://www.niap-ccevs.org/Profile/Info.cfm?PPID=441&id=441

MOD_VPN_CLI_V2.1 **PP-Module for Virtual Private Network (VPN) Clients**
Version 2.1
Date 2017-10-05
Location https://www.niap-ccevs.org/Profile/Info.cfm?PPID=419&id=419

NETEXT **Network Extension**
Date 2021
Location https://developer.apple.com/documentation/networkextension

NETWORK **Network Framework**
Date 2021
Location https://developer.apple.com/documentation/network

OTAConfig **Over-the-Air Profile Delivery and Configuration**
Date 2021
Location https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/iPhoneOTAConfiguration/Introduction/Introduction.html

PM_Help **Profile Manager User Guide**
Date 2021
Location https://support.apple.com/guide/profile-manager/welcome/mac

PP_MD_V3.1 **Protection Profile for Mobile Device Fundamentals**
Version 3.1
Date 2017-06-16
Location https://www.niap-ccevs.org/Profile/Info.cfm?PPID=417&id=417

PP_WLAN_CLI_EP_V1.0 **General Purpose Operating Systems Protection Profile/ Mobile Device Fundamentals Protection Profile Extended Package (EP) Wireless Local Area Network (WLAN) Clients**
Version 1.0
Date 2016-02-08
Location https://www.niap-ccevs.org/Profile/Info.cfm?PPID=386&id=386

PROFS_LOGS **Profiles and Logs**
Date 2021
Location https://developer.apple.com/bug-reporting/profiles-and-logs/?platforms=ios

SecCodeGuide **Secure Coding Guide**
Date 2016-09-13
Location https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Introduction.html

Copyright © 2021 atsec information security corporation

SECTRAN	**Secure Transport API**	
	Date	2021
	Location	https://developer.apple.com/documentation/security/secure_transport
ST	**Apple iPadOS 14: iPads Security Target**	
	Version	1.5
	Date	2021-08-31
	File name	ase/st-iPadOS14_PUBLIC_V1.5.pdf
TRUST_STORE	**List of available trusted root certificates in iOS 14, iPadOS 14, macOS11, watchOS 7, and tvOS 14**	
	Date	2021
	Location	https://support.apple.com/en-us/HT212140

Copyright © 2021 atsec information security corporation

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.

Copyright © 2021 atsec information security corporation

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.

Copyright © 2021 atsec information security corporation

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.

Copyright © 2021 atsec information security corporation

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.

Copyright © 2021 atsec information security corporation