

**Assurance Activity Report for
JUNOS 24.4R2 for MX240/480/960 Routers and EX9204/9208/9214 Switches**

**Junos 24.4R2 for MX240/480/960 Routers and EX9204/9208/9214 Switches
Security Target Version 0.6**

**collaborative Protection Profile for Network Devices Version 3.0e
Functional Package for SSH, Version 1.0
PP-Module for MACsec Ethernet Encryption, Version 1.0**

AAR Version 0.5,
June 3, 2026

Evaluated by:



**2400 Research Blvd, Suite 395
Rockville, MD 20850**

Prepared for:



**National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme**

The Developer of the TOE:
HPE Juniper Networking

The Author of the Security Target:
Acumen Security, LLC.

The TOE Evaluation was Sponsored by:
HPE Juniper Networking

Evaluation Personnel:
Pratheek Menon
Manohar Negi
Acumen Security, LLC.

Common Criteria Version
Common Criteria Version 3.1 Revision 5

Common Evaluation Methodology Version
CEM Version 3.1 Revision 5

REVISION HISTORY

VERSION	DATE	CHANGES
0.1	July 01, 2025	Initial Release
0.2	January 12, 2026	Changes resulting from lead review
0.3	March 17, 2026	Changes resulting from peer review
0.4	May 1, 2026	Changes based on learnings from other Juniper evaluations
0.5	June 3, 2026	Changes to address ECR comments

CONTENTS

1	TOE OVERVIEW	12
2	ASSURANCE ACTIVITIES IDENTIFICATION	15
3	TEST EQUIVALENCY JUSTIFICATION	16
3.1	PLATFORM/HARDWARE DEPENDENCIES	16
3.2	MACSEC CRYPTOGRAPHIC ASIC DEPENDENCIES.....	16
3.3	TOE FUNCTIONAL DIFFERENCES	17
3.4	DIFFERENCES IN LIBRARIES USED TO PROVIDE TOE FUNCTIONALITY	17
3.5	TOE MANAGEMENT INTERFACE DIFFERENCES.....	17
3.6	DIFFERENCE COMPARISON.....	18
3.7	RECOMMENDATIONS/CONCLUSIONS	18
4	TEST BED DESCRIPTIONS	19
4.1	AUDIT/AUTH/CRYPTO/UPDATE/SSH	19
4.2	MACSEC	19
4.3	CONFIGURATION INFORMATION	19
4.4	TEST TIME AND LOCATION.....	20
5	DETAILED EVALUATION ACTIVITIES	21
5.1	TSS AND AGD ACTIVITIES	21
5.1.1	<i>Security Audit (FAU)</i>	21
5.1.1.1	FAU_GEN.1 Audit Data Generation (CPP_ND_V3.0E)	21
5.1.1.1.1	FAU_GEN.1 TSS	21
5.1.1.1.2	FAU_GEN.1 AGD	22
5.1.1.2	FAU_GEN.1/MACSEC Audit Data Generation (MACsec) (MOD_MACSEC_V1.0).....	23
5.1.1.2.1	FAU_GEN.1/MACSEC	23
5.1.1.3	FAU_GEN.2 User Identity Association (CPP_ND_V3.0E)	24
5.1.1.3.1	FAU_GEN.2 TSS	24
5.1.1.3.2	FAU_GEN.2 AGD	24
5.1.1.4	FAU_STG_EXT.1 Protected Audit Event Storage (CPP_ND_V3.0E).....	24
5.1.1.4.1	FAU_STG_EXT.1 TSS.....	24
5.1.1.4.2	FAU_STG_EXT.1 AGD	27
5.1.2	<i>Cryptographic Support (FCS)</i>	30
5.1.2.1	FCS_CKM.1 Cryptographic Key Generation (CPP_ND_V3.0E)	30
5.1.2.1.1	FCS_CKM.1 TSS [TD0921]	30
5.1.2.1.2	FCS_CKM.1 AGD [TD0921]	31
5.1.2.2	FCS_CKM.2 Cryptographic Key Establishment (CPP_ND_V3.0E).....	32
5.1.2.2.1	FCS_CKM.2 TSS	32
5.1.2.2.2	FCS_CKM.2 AGD.....	32
5.1.2.3	FCS_CKM.4 Cryptographic Key Destruction (CPP_ND_V3.0E).....	33
5.1.2.3.1	FCS_CKM.4 TSS	33
5.1.2.3.2	FCS_CKM.4 AGD.....	37
5.1.2.4	FCS_COP.1/CMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm) (MOD_MACSEC_V1.0)	38
5.1.2.4.1	FCS_COP.1/CMAC TSS.....	38
5.1.2.4.2	FCS_COP.1/CMAC AGD	38
5.1.2.5	FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption) (CPP_ND_V3.0E) ..	38
5.1.2.5.1	FCS_COP.1/DataEncryption TSS.....	38

5.1.2.5.2	FCS_COP.1/DataEncryption AGD	39
5.1.2.6	FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm) (CPP_ND_V3.0E)	40
5.1.2.6.1	FCS_COP.1/Hash TSS	40
5.1.2.6.2	FCS_COP.1/Hash AGD	40
5.1.2.7	FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm) (CPP_ND_V3.0E)	41
5.1.2.7.1	FCS_COP.1/KeyedHash TSS	41
5.1.2.7.2	FCS_COP.1/KeyedHash AGD	42
5.1.2.8	FCS_COP.1/MACSEC Cryptographic Operation (MACsec AES Data Encryption and Decryption) (MOD_MACSEC_V1.0)	42
5.1.2.8.1	FCS_COP.1/MACSEC TSS	42
5.1.2.8.2	FCS_COP.1/MACSEC AGD	43
5.1.2.9	FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification) (CPP_ND_V3.0E)	43
5.1.2.9.1	FCS_COP.1/SigGen TSS [TD0921]	43
5.1.2.9.2	FCS_COP.1/SigGen AGD [TD0921]	43
5.1.2.10	FCS_MACSEC_EXT.1 MACsec (MOD_MACSEC_V1.0)	44
5.1.2.10.1	FCS_MACSEC_EXT.1 TSS [TD0884]	44
5.1.2.10.2	FCS_MACSEC_EXT.1 AGD [TD0884]	46
5.1.2.11	FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality (MOD_MACSEC_V1.0)	46
5.1.2.11.1	FCS_MACSEC_EXT.2 TSS [TD1030]	46
5.1.2.11.2	FCS_MACSEC_EXT.2 AGD	47
5.1.2.12	FCS_MACSEC_EXT.3 MACsec Randomness (MOD_MACSEC_V1.0)	47
5.1.2.12.1	FCS_MACSEC_EXT.3 TSS	47
5.1.2.12.2	FCS_MACSEC_EXT.3 AGD	48
5.1.2.13	FCS_MACSEC_EXT.4 MACsec Key Usage (MOD_MACSEC_V1.0)	48
5.1.2.13.1	FCS_MACSEC_EXT.4 TSS	48
5.1.2.13.2	FCS_MACSEC_EXT.4 AGD [TD0803]	49
5.1.2.14	FCS_MKA_EXT.1 MACsec Key Agreement (MOD_MACSEC_V1.0)	51
5.1.2.14.1	FCS_MKA_EXT.1.1, FCS_MKA_EXT.1.2, FCS_MKA_EXT.1.3 TSS	51
5.1.2.14.2	FCS_MKA_EXT.1.1, FCS_MKA_EXT.1.2, FCS_MKA_EXT.1.3 AGD	51
5.1.2.14.3	FCS_MKA_EXT.1.4 TSS	51
5.1.2.14.4	FCS_MKA_EXT.1.4 AGD	51
5.1.2.14.5	FCS_MKA_EXT.1.5, FCS_MKA_EXT.1.6, FCS_MKA_EXT.1.7 TSS	51
5.1.2.14.6	FCS_MKA_EXT.1.5, FCS_MKA_EXT.1.6, FCS_MKA_EXT.1.7 AGD	53
5.1.2.15	FCS_NTP_EXT.1 NTP Protocol (CPP_ND_V3.0E)	53
5.1.2.15.1	FCS_NTP_EXT.1.1 TSS	53
5.1.2.15.2	FCS_NTP_EXT.1.2, FCS_NTP_EXT.1.3, and FCS_NTP_EXT.1.4 TSS	54
5.1.2.15.3	FCS_NTP_EXT.1.1 AGD	54
5.1.2.15.4	FCS_NTP_EXT.1.2 AGD	55
5.1.2.15.5	FCS_NTP_EXT.1.3 AGD	56
5.1.2.15.6	FCS_NTP_EXT.1.4 AGD	57
5.1.2.16	FCS_RBG_EXT.1 Random Bit Generation (CPP_ND_V3.0E)	57
5.1.2.16.1	FCS_RBG_EXT.1 TSS	57
5.1.2.16.2	FCS_RBG_EXT.1 AGD	57
5.1.2.17	FCS_SSH_EXT.1 SSH PROTOCOL (PKG_SSH_V1.0)	58
5.1.2.17.1	FCS_SSH_EXT.1.1 TSS	58
5.1.2.17.2	FCS_SSH_EXT.1.1 AGD	58
5.1.2.17.3	FCS_SSH_EXT.1.2 TSS	58
5.1.2.17.4	FCS_SSH_EXT.1.2 AGD	59
5.1.2.17.5	FCS_SSH_EXT.1.3 TSS	62
5.1.2.17.6	FCS_SSH_EXT.1.3 AGD	62
5.1.2.17.7	FCS_SSH_EXT.1.4 TSS	62

5.1.2.17.8	FCS_SSH_EXT.1.4 AGD	63
5.1.2.17.9	FCS_SSH_EXT.1.5 TSS	64
5.1.2.17.10	FCS_SSH_EXT.1.5 AGD	64
5.1.2.17.11	FCS_SSH_EXT.1.6 TSS	65
5.1.2.17.12	FCS_SSH_EXT.1.6 AGD	65
5.1.2.17.13	FCS_SSH_EXT.1.7 TSS	66
5.1.2.17.14	FCS_SSH_EXT.1.7 AGD	66
5.1.2.17.15	FCS_SSH_EXT.1.8 TSS	67
5.1.2.17.16	FCS_SSH_EXT.1.8 AGD	67
5.1.2.18	FCS_SSHS_EXT.1 SSH Protocol - Server (PKG_SSH_V1.0)	68
5.1.2.18.1	FCS_SSHS_EXT.1 TSS	68
5.1.2.18.2	FCS_SSHS_EXT.1 AGD	68
5.1.3	<i>Identification and Authentication (FIA)</i>	69
5.1.3.1	FIA_AFL.1 Authentication Failure Management (CPP_ND_V3.0E)	69
5.1.3.1.1	FIA_AFL.1 TSS	69
5.1.3.1.2	FIA_AFL.1 AGD	70
5.1.3.2	FIA_PMG_EXT.1 Password Management (CPP_ND_V3.0E)	71
5.1.3.2.1	FIA_PMG_EXT.1 TSS	71
5.1.3.2.2	FIA_PMG_EXT.1 AGD	72
5.1.3.3	FIA_PSK_EXT.1 Pre-Shared Key Composition (MOD_MACSEC_V1.0)	74
5.1.3.3.1	FIA_PSK_EXT.1 TSS	74
5.1.3.3.2	FIA_PSK_EXT.1 AGD	74
5.1.3.4	FIA_UIA_EXT.1 User Identification and Authentication (CPP_ND_V3.0E)	76
5.1.3.4.1	FIA_UIA_EXT.1 TSS	76
5.1.3.4.2	FIA_UIA_EXT.1 AGD	77
5.1.3.5	FIA_UAU.7 Protected Authentication Feedback (CPP_ND_V3.0E)	80
5.1.3.5.1	FIA_UAU.7 TSS	80
5.1.3.5.2	FIA_UAU.7 AGD	80
5.1.4	<i>Security Management (FMT)</i>	81
5.1.4.1	FMT_MTD.1/CryptoKeys Management of TSF Data (CPP_ND_V3.0E)	81
5.1.4.1.1	FMT_MTD.1/CryptoKeys TSS	81
5.1.4.1.2	FMT_MTD.1/CryptoKeys AGD	81
5.1.4.2	FMT_MTD.1/CoreData Management of TSF Data (CPP_ND_V3.0E)	83
5.1.4.2.1	FMT_MTD.1/CoreData TSS	83
5.1.4.2.2	FMT_MTD.1/CoreData AGD	84
5.1.4.3	FMT_MOF.1/Functions Management of Security Functions Behaviour (CPP_ND_V3.0E)	88
5.1.4.3.1	FMT_MOF.1/Functions TSS	88
5.1.4.3.2	FMT_MOF.1/Functions AGD	88
5.1.4.4	FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour (CPP_ND_V3.0E)	91
5.1.4.4.1	FMT_MOF.1/ManualUpdate TSS	91
5.1.4.4.2	FMT_MOF.1/ManualUpdate AGD	91
5.1.4.5	FMT_MOF.1/SERVICES MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR (CPP_ND_V3.0E)	94
5.1.4.5.1	FMT_MOF.1/Services TSS	94
5.1.4.5.2	FMT_MOF.1/Services AGD	95
5.1.4.6	FMT_SMF.1 Specification of Management Functions (CPP_ND_V3.0E)	96
5.1.4.6.1	FMT_SMF.1 TSS (containing also requirements on Guidance Documentation and Tests)	96
5.1.4.6.2	FMT_SMF.1 AGD	100
5.1.4.7	FMT_SMF.1/MACSEC Specification of Management Functions (MACsec) (MOD_MACSEC_V1.0)	100
5.1.4.7.1	FMT_SMF.1/MACSEC TSS	100
5.1.4.7.2	FMT_SMF.1/MACSEC AGD	101
5.1.4.8	FMT_SMR.2 Restrictions On Security Roles (CPP_ND_V3.0E)	103

5.1.4.8.1	FMT_SMR.2 TSS	103
5.1.4.8.2	FMT_SMR.2 AGD.....	103
5.1.5	<i>Protection of the TSF (FPT)</i>	105
5.1.5.1	FPT_APW_EXT.1 Protection of Administrator Passwords (CPP_ND_V3.0E)	105
5.1.5.1.1	FPT_APW_EXT.1 TSS	105
5.1.5.1.2	FPT_APW_EXT.1 AGD.....	106
5.1.5.2	FPT_CAK_EXT.1 Protection of CAK Data (MOD_MACSEC_V1.0)	106
5.1.5.2.1	FPT_CAK_EXT.1 TSS.....	106
5.1.5.2.2	FPT_CAK_EXT.1 AGD	106
5.1.5.3	FPT_FLS.1 Failure with Preservation of Secure State	107
5.1.5.3.1	FPT_FLS.1 TSS [TD0816].....	107
5.1.5.3.2	FPT_FLS.1 AGD	107
5.1.5.4	FPT_RPL.1 Replay Detection	108
5.1.5.4.1	FPT_RPL.1 TSS [TD0881]	108
5.1.5.4.2	FPT_RPL.1 AGD	108
5.1.5.5	FPT_SKP_EXT.1 Protection of TSF Data (for Reading of All Symmetric Keys) (CPP_ND_V3.0E)	108
5.1.5.5.1	FPT_SKP_EXT.1 TSS	108
5.1.5.5.2	FPT_SKP_EXT.1 AGD	109
5.1.5.6	FPT_STM_EXT.1 Reliable Time Stamps (CPP_ND_V3.0E)	109
5.1.5.6.1	FPT_STM_EXT.1 TSS.....	109
5.1.5.6.2	FPT_STM_EXT.1 AGD	110
5.1.5.7	FPT_TST_EXT.1 TSF Testing (CPP_ND_V3.0E).....	111
5.1.5.7.1	FPT_TST_EXT.1 TSS [TD0836]	111
5.1.5.7.2	FPT_TST_EXT.1 AGD.....	113
5.1.5.8	FPT_TUD_EXT.1 Trusted Update (CPP_ND_V3.0E)	114
5.1.5.8.1	FPT_TUD_EXT.1 TSS.....	114
5.1.5.8.2	FPT_TUD_EXT.1 AGD.....	116
5.1.6	<i>TOE Access (FTA)</i>	117
5.1.6.1	FTA_SSL_EXT.1 TSF-Initiated Session Locking (CPP_ND_V3.0E).....	117
5.1.6.1.1	FTA_SSL_EXT.1 TSS.....	117
5.1.6.1.2	FTA_SSL_EXT.1 AGD.....	118
5.1.6.2	FTA_SSL.3 TSF-Initiated Termination (CPP_ND_V3.0E).....	118
5.1.6.2.1	FTA_SSL.3 TSS	118
5.1.6.2.2	FTA_SSL.3 AGD.....	119
5.1.6.3	FTA_SSL.4 User-Initiated Termination (CPP_ND_V3.0E)	119
5.1.6.3.1	FTA_SSL.4 TSS	119
5.1.6.3.2	FTA_SSL.4 AGD.....	119
5.1.6.4	FTA_TAB.1 Default TOE Access Banners (CPP_ND_V3.0E)	120
5.1.6.4.1	FTA_TAB.1 TSS	120
5.1.6.4.2	FTA_TAB.1 AGD.....	121
5.1.7	<i>Trusted Path/Channels (FTP)</i>	122
5.1.7.1	FTP_ITC.1 Inter-TSF Trusted Channel (CPP_ND_V3.0E)	122
5.1.7.1.1	FTP_ITC.1 TSS.....	122
5.1.7.1.2	FTP_ITC.1 AGD	123
5.1.7.2	FTP_ITC.1/MACSEC Inter-TSF Trusted Channel (MACsec Communications).....	125
5.1.7.2.1	FTP_ITC.1/MACSEC	125
5.1.7.3	FTP_TRP.1/Admin Trusted Path (CPP_ND_V3.0E)	125
5.1.7.3.1	FTP_TRP.1/Admin TSS.....	125
5.1.7.3.2	FTP_TRP.1/Admin AGD	126

6 SECURITY ASSURANCE REQUIREMENTS 128

6.1	ASE: SECURITY TARGET EVALUATION	128
6.1.1	General Evaluation Activities for TOE Summary Specification (ASE_TSS.1) for All TOEs (CPP_ND_V3.0E)	128
6.1.1.1	Evaluation Activity.....	129
6.2	ADV: DEVELOPMENT	129
6.2.1	Basic Functional Specification (ADV_FSP.1) (CPP_ND_V3.0E)	129
6.2.1.1	Evaluation Activity.....	129
6.2.1.2	Evaluation Activity.....	130
6.2.1.3	Evaluation Activity.....	130
6.3	AGD: GUIDANCE DOCUMENTATION	131
6.3.1	Operational User Guidance (AGD_OPE.1) (CPP_ND_V3.0E)	131
6.3.1.1	Evaluation Activity.....	131
6.3.1.2	Evaluation Activity.....	131
6.3.1.3	Evaluation Activity.....	131
6.3.1.4	Evaluation Activity.....	132
6.3.1.5	Evaluation Activity.....	132
6.3.2	Preparative Procedures (AGD_PRE.1) (CPP_ND_V3.0E)	135
6.3.2.1	Evaluation Activity.....	135
6.3.2.2	Evaluation Activity.....	135
6.3.2.3	Evaluation Activity.....	136
6.3.2.4	Evaluation Activity.....	136
6.3.2.5	Evaluation Activity.....	137
6.4	AVA: VULNERABILITY ASSESSMENT.....	137
6.4.1	Vulnerability Survey (AVA_VAN.1) (CPP_ND_V3.0E)	137
6.4.1.1	Evaluation Activity (Documentation)	137
6.4.1.2	Evaluation Activity.....	138
7	DETAILED TEST CASES (TEST ACTIVITIES)	140
7.1	TEST ACTIVITIES.....	140
7.1.1	Security Audit (FAU).....	140
7.1.1.1	FAU_GEN.1 Test #1 (CPP_ND_V3.0E).....	140
7.1.1.2	FAU_GEN.1/MACSEC Test #1 (MOD_MACSEC_V1.0).....	140
7.1.1.3	FAU_GEN.1 Test #2 (CPP_ND_V3.0E).....	141
7.1.1.4	FAU_GEN.2 Test #1 (CPP_ND_V3.0E).....	141
7.1.1.5	FAU_GEN.2 Test #2 (CPP_ND_V3.0E).....	141
7.1.1.6	FAU_STG_EXT.1 Test #1 (CPP_ND_V3.0E).....	141
7.1.1.7	FAU_STG_EXT.1 Test #2 (CPP_ND_V3.0E).....	142
7.1.1.8	FAU_STG_EXT.1 Test #3 (a) (CPP_ND_V3.0E)	143
7.1.1.9	FAU_STG_EXT.1 Test #3 (b) (CPP_ND_V3.0E)	143
7.1.1.10	FAU_STG_EXT.1 Test #4 (a) (CPP_ND_V3.0E)	143
7.1.1.11	FAU_STG_EXT.1 Test #4 (b) (CPP_ND_V3.0E)	144
7.1.1.12	FAU_STG_EXT.1 Test #4 (c) (CPP_ND_V3.0E)	144
7.1.1.13	FAU_STG_EXT.1 Test #5 (CPP_ND_V3.0E)	145
7.1.1.14	FAU_STG_EXT.1 Test #6 (CPP_ND_V3.0E) [TD0886]	145
7.1.2	Cryptographic Support (FCS)	146
7.1.2.1	FCS_CKM.1 RSA (CPP_ND_V3.0E) [TD0921]	146
7.1.2.2	FCS_CKM.1 ECC (CPP_ND_V3.0E) [TD0921].....	147
7.1.2.3	FCS_CKM.1 FFC - FIPS PUB 186-4 (CPP_ND_V3.0E).....	149
7.1.2.4	FCS_CKM.1 FFC - “safe-prime” groups (CPP_ND_V3.0E)	150

7.1.2.5	FCS_CKM.2 RSA (CPP_ND_V3.0E)	150
7.1.2.6	FCS_CKM.2 SP800-56A - ECC (CPP_ND_V3.0E)	150
7.1.2.7	FCS_CKM.2 SP800-56A - FFC (CPP_ND_V3.0E).....	152
7.1.2.8	FCS_CKM.2 FFC safe-prime (CPP_ND_V3.0E).....	153
7.1.2.9	FCS_CKM.4 (CPP_ND_V3.0E)	154
7.1.2.10	FCS_COP.1/DataEncryption AES-CBC (CPP_ND_V3.0E)	154
7.1.2.11	FCS_COP.1/DataEncryption AES-GCM (CPP_ND_V3.0E)	156
7.1.2.12	FCS_COP.1/DataEncryption AES-CTR (CPP_ND_V3.0E)	157
7.1.2.13	FCS_COP.1/Hash (CPP_ND_V3.0E)	159
7.1.2.14	FCS_COP.1/KeyedHash (CPP_ND_V3.0E).....	160
7.1.2.15	FCS_COP.1/CMAC Test #1 (MOD_MACSEC_V1.0)	160
7.1.2.16	FCS_COP.1/CMAC Test #2 (MOD_MACSEC_V1.0)	161
7.1.2.17	FCS_COP.1/MACSEC Test #3 (MOD_MACSEC_V1.0)	162
7.1.2.18	FCS_COP.1/MACSEC Test #4 (MOD_MACSEC_V1.0) [TD0728]	163
7.1.2.19	FCS_COP.1/SigGen ECDSA (CPP_ND_V3.0E) [TD0921]	164
7.1.2.20	FCS_COP.1/SigGen RSA (CPP_ND_V3.0E) [TD0921]	165
7.1.2.21	FCS_MACSEC_EXT.1 Test #5 (MOD_MACSEC_V1.0)	166
7.1.2.22	FCS_MACSEC_EXT.1 Test #6 (MOD_MACSEC_V1.0) [TD0884]	167
7.1.2.23	FCS_MACSEC_EXT.2 Test #7 (MOD_MACSEC_V1.0)	167
7.1.2.24	FCS_MACSEC_EXT.2 Test #8 (MOD_MACSEC_V1.0)	168
7.1.2.25	FCS_MACSEC_EXT.3 Test #1 (MOD_MACSEC_V1.0)	168
7.1.2.26	FCS_MACSEC_EXT.4 Test #9 (MOD_MACSEC_V1.0)	168
7.1.2.27	FCS_MACSEC_EXT.4 Test #10 (MOD_MACSEC_V1.0)	169
7.1.2.28	FCS_MKA_EXT.1.3 Test #11 (MOD_MACSEC_V1.0)	169
7.1.2.29	FCS_MKA_EXT.1.3 Test #12 (MOD_MACSEC_V1.0)	170
7.1.2.30	FCS_MKA_EXT.1.4 Test #13a (MOD_MACSEC_V1.0) [TD0882]	170
7.1.2.31	FCS_MKA_EXT.1.4 Test #13b (MOD_MACSEC_V1.0) [TD0882]	171
7.1.2.32	FCS_MKA_EXT.1.4 Test #14a (MOD_MACSEC_V1.0) [TD0882]	171
7.1.2.33	FCS_MKA_EXT.1.4 Test #14b (MOD_MACSEC_V1.0) [TD0882]	172
7.1.2.34	FCS_MKA_EXT.1.7 Test #15 (MOD_MACSEC_V1.0) [TD0882]	172
7.1.2.35	FCS_MKA_EXT.1.7 Test #16 (MOD_MACSEC_V1.0) [TD0889]	173
7.1.2.36	FCS_NTP_EXT.1.1 Test #1 (CPP_ND_V3.0E)	174
7.1.2.37	FCS_NTP_EXT.1.2 Test #1 (CPP_ND_V3.0E)	175
7.1.2.38	FCS_NTP_EXT.1.3 Test #1 (CPP_ND_V3.0E)	176
7.1.2.39	FCS_NTP_EXT.1.4 Test #1 (CPP_ND_V3.0E)	177
7.1.2.40	FCS_NTP_EXT.1.4 Test #2 (CPP_ND_V3.0E)	178
7.1.2.41	FCS_RBG_EXT.1 (CPP_ND_V3.0E)	179
7.1.2.42	FCS_SSH_EXT.1.1 Test #1 (PKG_SSH_V1.0)	180
7.1.2.43	FCS_SSH_EXT.1.2 Test #1a (PKG_SSH_V1.0)	180
7.1.2.44	FCS_SSH_EXT.1.2 Test #1b (PKG_SSH_V1.0)	182
7.1.2.45	FCS_SSH_EXT.1.2 Test #2 (PKG_SSH_V1.0)	183
7.1.2.46	FCS_SSH_EXT.1.2 Test #3 (PKG_SSH_V1.0)	183
7.1.2.47	FCS_SSH_EXT.1.3 Test #1 (PKG_SSH_V1.0)	184
7.1.2.48	FCS_SSH_EXT.1.3 Test #2 (PKG_SSH_V1.0) [TD0732]	184
7.1.2.49	FCS_SSH_EXT.1.4 Test #1 (PKG_SSH_V1.0)	185
7.1.2.50	FCS_SSH_EXT.1.4 Test #2 (PKG_SSH_V1.0)	185
7.1.2.51	FCS_SSH_EXT.1.4 Test #3 (PKG_SSH_V1.0)	186
7.1.2.52	FCS_SSH_EXT.1.5 Test #1 (PKG_SSH_V1.0)	186
7.1.2.53	FCS_SSH_EXT.1.5 Test #2 (PKG_SSH_V1.0) [TD1023]	187
7.1.2.54	FCS_SSH_EXT.1.6 Test #1 (PKG_SSH_V1.0) [TD0967]	187
7.1.2.55	FCS_SSH_EXT.1.6 Test #2 (PKG_SSH_V1.0)	187

7.1.2.56	FCS_SSH_EXT.1.6 Test #3 (PKG_SSH_V1.0) [TD0967]	188
7.1.2.57	FCS_SSH_EXT.1.6 Test #4 (PKG_SSH_V1.0) [TD0967]	188
7.1.2.58	FCS_SSH_EXT.1.7 Test #1 (PKG_SSH_V1.0)	189
7.1.2.59	FCS_SSH_EXT.1.8 Test #1 (PKG_SSH_V1.0)	189
7.1.2.60	FCS_SSH_EXT.1.8 Test #2 (PKG_SSH_V1.0)	189
7.1.2.61	FCS_SSH_EXT.1.8 Test #3 (PKG_SSH_V1.0)	190
7.1.2.62	FCS_SSHS_EXT.1 Test #1 (PKG_SSH_V1.0) [TD0682]	190
7.1.2.63	FCS_SSHS_EXT.1 Test #2 (PKG_SSH_V1.0) [TD0682]	191
7.1.2.64	FCS_SSHS_EXT.1 Test #3 (PKG_SSH_V1.0) [TD0682]	192
7.1.3	<i>Identification and Authentication (FIA)</i>	192
7.1.3.1	FIA_AFL.1 Test #1 (CPP_ND_V3.0E)	192
7.1.3.2	FIA_AFL.1 Test #2a (CPP_ND_V3.0E)	193
7.1.3.3	FIA_AFL.1 Test #2b (CPP_ND_V3.0E)	193
7.1.3.4	FIA_PMG_EXT.1 Test #1 (CPP_ND_V3.0E)	194
7.1.3.5	FIA_PMG_EXT.1 Test #2 (CPP_ND_V3.0E)	195
7.1.3.6	FIA_PSK_EXT.1 Test #17 (MOD_MACSEC_V1.0)	195
7.1.3.7	FIA_PSK_EXT.1 Test #18 (MOD_MACSEC_V1.0)	197
7.1.3.8	FIA_PSK_EXT.1 Test #19 (MOD_MACSEC_V1.0)	197
7.1.3.9	FIA_UIA_EXT.1 Test #1 (CPP_ND_V3.0E)	197
7.1.3.10	FIA_UIA_EXT.1 Test #2 (CPP_ND_V3.0E)	198
7.1.3.11	FIA_UIA_EXT.1 Test #3 (CPP_ND_V3.0E)	199
7.1.3.12	FIA_UIA_EXT.1 Test #4 (CPP_ND_V3.0E)	199
7.1.3.13	FIA_UAU.7 Test #1 (CPP_ND_V3.0E)	200
7.1.4	<i>Security Management (FMT)</i>	200
7.1.4.1	FMT_MOF.1/ Functions Test #1a (CPP_ND_V3.0E)	200
7.1.4.2	FMT_MOF.1/ Functions Test #2a (CPP_ND_V3.0E)	201
7.1.4.3	FMT_MOF.1/ Functions Test #1b (CPP_ND_V3.0E)	201
7.1.4.4	FMT_MOF.1/ Functions Test #2b (CPP_ND_V3.0E)	202
7.1.4.5	FMT_MOF.1/ Functions Test #1c (CPP_ND_V3.0E)	202
7.1.4.6	FMT_MOF.1/ Functions Test #2c (CPP_ND_V3.0E)	203
7.1.4.7	FMT_MOF.1/ Functions Test #1d (CPP_ND_V3.0E)	203
7.1.4.8	FMT_MOF.1/ Functions Test #2d (CPP_ND_V3.0E)	204
7.1.4.9	FMT_MOF.1/ManualUpdate Test #1 (CPP_ND_V3.0E)	204
7.1.4.10	FMT_MOF.1/ManualUpdate Test #2 (CPP_ND_V3.0E)	204
7.1.4.11	FMT_MOF.1/Services Test #1 (CPP_ND_V3.0E)	205
7.1.4.12	FMT_MOF.1/Services Test #2 (CPP_ND_V3.0E)	205
7.1.4.13	FMT_MTD.1/CoreData Test #1 (CPP_ND_V3.0E)	205
7.1.4.14	FMT_MTD.1/CryptoKeys Test #1 (CPP_ND_V3.0E)	206
7.1.4.15	FMT_MTD.1/CryptoKeys Test #2 (CPP_ND_V3.0E)	206
7.1.4.16	FMT_SMF.1 Test #1 (CPP_ND_V3.0E) [TD0880]	206
7.1.4.17	FMT_SMF.1/MACSEC Test #20 (MOD_MACSEC_V1.0) [TD0803]	207
7.1.4.18	FMT_SMF.1/MACSEC Test #21 (MOD_MACSEC_V1.0) [TD0889]	209
7.1.4.19	FMT_SMF.1/MACSEC Test #22 (MOD_MACSEC_V1.0) [TD0840]	209
7.1.4.20	FMT_SMF.1/MACSEC Test #23 (MOD_MACSEC_V1.0)	210
7.1.4.21	FMT_SMR.2 Test #1 (CPP_ND_V3.0E)	211
7.1.5	<i>Protection of the TSF (FPT)</i>	211
7.1.5.1	FPT_APW_EXT.1 Test #1 (CPP_ND_V3.0E)	211
7.1.5.2	FPT_CAK_EXT.1 Test #1 (MOD_MACSEC_V1.0)	211
7.1.5.3	FPT_FLS.1 Test #24 (MOD_MACSEC_V1.0) [TD0816]	212
7.1.5.4	FPT_RPL.1 Test #25 (MOD_MACSEC_V1.0) [TD0746, TD0881]	212
7.1.5.5	FPT_RPL.1 Test #26 (MOD_MACSEC_V1.0) [TD0881]	213

7.1.5.6	FPT_SKP_EXT.1 Test #1 (CPP_ND_V3.0E).....	213
7.1.5.7	FPT_STM_EXT.1 Test #1 (CPP_ND_V3.0E).....	214
7.1.5.8	FPT_STM_EXT.1 Test #2 (CPP_ND_V3.0E).....	214
7.1.5.9	FPT_STM_EXT.1 Test #3 (CPP_ND_V3.0E).....	214
7.1.5.10	FPT_TST_EXT.1 Test #1 (CPP_ND_V3.0E) [TD0836].....	215
7.1.5.11	FPT_TUD_EXT.1 Test #1 (CPP_ND_V3.0E)	216
7.1.5.12	FPT_TUD_EXT.1 Test #2 (a) (CPP_ND_V3.0E)	216
7.1.5.13	FPT_TUD_EXT.1 Test #2 (b) (CPP_ND_V3.0E)	217
7.1.5.14	FPT_TUD_EXT.1 Test #2 (c) (CPP_ND_V3.0E)	218
7.1.6	TOE Access (FTA).....	219
7.1.6.1	FTA_SSL_EXT.1 Test #1 (CPP_ND_V3.0E)	219
7.1.6.2	FTA_SSL.3 Test #1 (CPP_ND_V3.0E)	219
7.1.6.3	FTA_SSL.4 Test #1 (CPP_ND_V3.0E).....	220
7.1.6.4	FTA_SSL.4 Test #2 (CPP_ND_V3.0E).....	220
7.1.6.5	FTA_TAB.1 Test #1 (CPP_ND_V3.0E).....	221
7.1.7	Trusted Path/Channels (FTP)	221
7.1.7.1	FTP_ITC.1 Test #1 (CPP_ND_V3.0E)	221
7.1.7.2	FTP_ITC.1 Test #2 (CPP_ND_V3.0E)	222
7.1.7.3	FTP_ITC.1 Test #3 (CPP_ND_V3.0E)	222
7.1.7.4	FTP_ITC.1 Test #4 (CPP_ND_V3.0E)	223
7.1.7.5	FTP_ITC.1/MACSEC Test #1 (MOD_MACSEC_V1.0)	224
7.1.7.6	FTP_TRP.1/Admin Test #1 (CPP_ND_V3.0E)	224
7.1.7.7	FTP_TRP.1/Admin Test #2 (CPP_ND_V3.0E)	225
8	CAVP CERTIFICATE TABLE	226
9	CONCLUSION.....	229

1 TOE OVERVIEW

The TOE is HPE Juniper Networking Junos 24.4R2 for MX240/480/960 Routers and EX9204/9208/9214 Switches with MACsec on the MPC10E/EX9200/MPC3E Series Line Cards.

The following appliance models constitute the variations of the TOE:

Model	Routing Engine (RE)	RE-CPU	MACsec Line card	MIC-3D-10GE-SFP-E within MPC3E-3D-NG
MX240	RE-S-X6-128G	HASWELL-EP, Intel Xeon E5-2608L V3	MPC10E-10C and MPC10E-15C, with JNPR Penta Silicon MACsec ASIC	BCM82756 MACsec ASIC with CAVP num AES4550
MX480	RE-S-X6-64G			
MX960				
EX9204	EX9200-RE2	HASWELL-EP, Intel Xeon E5-2608L V3	EX9200-15C with JNPR Penta Silicon MACsec ASIC	Not supported
EX9208				
EX9214				

Note that the RE-S-X6 and EX9200-RE2 routing engines have identical hardware and are simply marketed with different names.

The TOE is a secure network device that protects itself largely by offering only a minimal logical interface to the network and attached nodes. It includes the Junos OS firmware, JUNOS OS 24.4R2, which is a special purpose OS offering no general-purpose computing capabilities. Junos OS implements both management and control functions as well as all IP routing.

All of the appliance's primary functions are to support the definition and enforcement of information flow policies among network nodes. Each information flow from one network node to other passes through an instance of the TOE. Information flow is controlled based on network node addresses and protocol. The TOE's also ensure that security-relevant activity is audited and implements the necessary tools to manage all security functions.

The TOE implements a variety of high-speed interfaces (only Ethernet is in the scope of the evaluation) for enterprise branch, campus, and data center networks. The appliances are physically self-contained, housing the firmware and hardware necessary to perform all routing functions.

The architecture components of the MX240, MX480, and MX960 are:

- **Modular Control Plane:** The platforms utilize a redundant Routing Engine (RE) for the control plane, with available models including the RE-S-X6-64G and RE-S-X6-128G, both featuring a 6-core 2.0 GHz CPU with 64 GB and 128 GB of memory, respectively.
- **Unified Operating System:** The architecture runs on the modular Junos OS, which provides routing, switching, security, and service features, as well as supporting automation tools and telemetry.
- **Programmable Forwarding:** At the core of the hardware is the programmable Trio chipset, which enables wire-rate forwarding, service agility, and allows for the implementation of inline services and telemetry.
- **Inline MACsec Technology:** The platforms support robust data plane security through features like integrated Layer 2 MACsec, with line cards such as the MPC-10E providing inline AES-256 MACsec line-side encryption.

The architecture components of the EX9204, EX9208, EX9214 are:

- **Modular Chassis Design:** The EX9204 (4-slot), EX9208 (8-slot), and EX9214 (14-slot) are modular chassis that share a common set of line cards, Routing Engines, and Switch Fabric modules.
- **Centralized Control Plane:** The architecture is built for high availability with support for redundant EX9200-RE2 Routing Engines. The RE2 features a six-core, 2 GHz Intel processor with 64 GB of DRAM to run the Junos OS and manage all control plane functions.
- **Programmable Forwarding ASIC:** The platforms are based on "Juniper One," a custom-designed ASIC that provides a programmable Packet Forwarding Engine (PFE), allowing new networking protocols and features to be added via software updates.
- **High-Density Cryptographic-Capable Line Cards:** The chassis support various line cards, including the EX9200-15C, which provides 15 ports of 100GbE/40GbE with Media Access Control Security (MACsec) capability, each of which can house 10-gigabit small form-factor pluggable plus (SFP+) transceivers and includes hardware support for cryptographic functions.
- **Integrated MACsec Technology:** These cryptographic capabilities are exposed via MACsec technology (IEEE 802.1ae) with AES-256-bit encryption, supported on line cards like the EX9200-15C to provide link-layer data confidentiality, integrity, and authentication.
- **Secure OS and Logging:** The EX9200-RE2 runs the carrier-class Junos OS, which stores its images and system logs on dual 64 GB SSDs. For secure management, the OS provides secure access protocols, including SSHv2.
- **Separated Management and Control:** The EX9200-RE2 modules communicate with the line cards over a dedicated internal out-of-band Gigabit Ethernet control interface, separating management traffic from the data plane.

The appliance supports numerous routing and switching standards for flexibility and scalability. Juniper's Virtual Chassis technology allows multiple interconnected switches to operate as a single, logical unit, enabling users to manage all platforms as one virtual device. The functions of the appliances can all be managed through the Junos firmware, either from a connected terminal console or via a network connection. Network management is secured using the SSH protocol. All management, whether from a user connecting to a terminal or from the network,

requires successful authentication. In the evaluated deployment the TOE is managed and configured via Command Line Interface, either via a directly connected console or over the network secured using the SSH protocol.

The TOE implements MACsec between adjacent devices. All traffic communicated between the devices including frames for LLDP (Link Layer Discovery Protocol), DHCP (Dynamic Host Configuration Protocol), ARP (Address Resolution Protocol), STP (Spanning Tree Protocol), Ethernet Control frames, etc. (the exceptions to this protection are Destination MAC and Source MAC addresses in MACsec and MKA frames).

MACsec can be deployed in point-to-point mode or shared mode with multiple stations. In the evaluated configuration MACsec must be configured individually on each point-to-point Ethernet link, such that a pair of MACsec devices (connected by a physical medium) protect Ethernet frames switched or routed from one device to the other. The two MACsec devices are provided with a Connectivity Association Key (CAK) and utilize the MACsec Key Agreement (MKA) protocol to create a secure tunnel. MKA is used by the two MACsec devices to agree upon MACsec keys. MACsec must be configured to protect all traffic between the devices, with the exception of the MKA or Ethernet control traffic such as EAP over LAN (EAPOL) frames. The devices will first exchange MKA frames, which serve to determine if the peer is an authorized peer and agree upon a shared key and MACsec cipher suite used to set up a transmit (Tx) Security Association (SA) and a receive (Rx) SA. Once the SAs are set up, MACsec-protected frames traverse the unprotected link.

2 ASSURANCE ACTIVITIES IDENTIFICATION

The Assurance Activities contained within this document include all those defined within the NDcPP 3.0e, PKG_SSH_V1.0, and MOD_MACSEC_V1.0 based upon the core SFRs and those implemented based on selections within the PP.

3 TEST EQUIVALENCY JUSTIFICATION

3.1 PLATFORM/HARDWARE DEPENDENCIES

The TOE boundary is inclusive of all hardware required by the TOE. Hardware platforms don't directly provide TSF functionality, instead they serve as the foundational execution environment for the TOE software. Hardware differences within the TOE are primarily related to form factor, capacity, and specific networking components.

Model	RE	RE-CPU	Analysis
MX240	RE-S-X6-64G	HASWELL-EP, Xeon E5-2608L V3	All platforms use the exact same processor (HASWELL-EP, Xeon E5-2608L V3) across different Routing Engines. This processor is critical from an NDcPP, MOD_MACsec and PKG_SSH scope perspective, as it executes the core security functions of the TOE's operating system.
MX480	RE-S-X6-128G		
MX960			
EX9204	EX9200-RE2	HASWELL-EP, Xeon E5-2608L V3	
EX9208			
EX9214			

Result: All RE platforms are considered equivalent from an NDcPP, MOD_MACsec and PKG_SSH scope perspective due to the consistent use of the exact same processor (HASWELL-EP, Xeon E5-2608L V3) across all Routing Engines.

3.2 MACSEC CRYPTOGRAPHIC ASIC DEPENDENCIES

MACsec ASIC	MACSec Linecard	Supported Platforms (for this ASIC)	Analysis
JNPR Penta Silicon	MPC10E-10C, MPC10E-15C, EX9200-15C	MX240, MX480, MX960, EX9204, EX9208, EX9214	All listed linecards utilize the identical JNPR Penta Silicon ASIC. As this ASIC handles MACsec cryptographic functions, linecards employing it are functionally equivalent for MACsec testing.
BCM82756	MIC-3D-10GE-SFP-E (within MPC3E-3D-NG)	MX240, MX480, MX960	This ASIC is distinct from the JNPR Penta Silicon. It is supported on specific MICs within MX linecards. EX models do not support this specific MIC/ASIC combination for MACsec.

Result: The JNPR Penta Silicon MACsec ASIC, found in the MPC10E-10C, MPC10E-15C, and EX9200-15C line cards are considered equivalent for MACsec testing because the ASIC itself performs the MACsec cryptographic functions. However, the BCM82756 MACsec ASIC is a separate and distinct cryptographic component; therefore, the JNPR Penta Silicon ASIC and the BCM82756 ASIC are not equivalent to each other and will require separate MACsec testing to ensure complete coverage of all cryptographic implementations in scope.

3.3 TOE FUNCTIONAL DIFFERENCES

TOE Model Type	Description	Analysis
MX series (MX240/480/960)	Routers	The fundamental functional difference between MX (router) and EX (switch) series devices is in their primary networking roles. However, these differences are related to network forwarding functionality and are not security relevant in the context of the NDcPP, MOD_MACsec and PKG_SSH testing scope.
EX series (EX9204/9208/9214)	Switches	

Result: While the MX and EX series have different primary functionalities (router vs. switch), these differences do not impact the security functionality under evaluation for NDcPP, MOD_MACsec and SSH Package testing.

3.4 DIFFERENCES IN LIBRARIES USED TO PROVIDE TOE FUNCTIONALITY

All software binaries and included libraries within the TOE are identical across models as far as the CC scope is considered, operating under the same JUNOS OS version. This ensures a consistent set of cryptographic modules for all security functions (including key management, encryption, and hashing) and uniform adherence to standards for SSHv2 and MACsec implementations across platforms.

Result: There are no security-relevant differences in the included libraries or cryptographic modules used across platforms. Both models are equivalent.

3.5 TOE MANAGEMENT INTERFACE DIFFERENCES

The TOE is managed via either a remote CLI session or a directly connected local console CLI. These management options are available on all hardware platforms regardless of the configuration. The remote CLI sessions are secured using SSHv2. There is no difference in the management interface or the protocols used for management across any platform.

Result: The management interfaces and protocols are consistent across all platforms. Both models are equivalent.

3.6 DIFFERENCE COMPARISON

This section summarizes the primary differences identified and their implications for testing.

- **Processor Equivalence:** All devices share the exact same CPU, making them **equivalent** for general security testing (NDcPP, MOD_MACsec and SSH Package).
- **MACsec ASIC Diversity:** Two distinct MACsec ASICs (JNPR Penta Silicon and BCM82756) are in scope. These are not equivalent and necessitate separate validation for MACsec cryptographic functions.
- **Library Consistency:** All software binaries and included libraries within the TOE are identical across models, operating under the same JUNOS-OS version. This ensures a consistent set of cryptographic modules for all security functions (including key management, encryption, and hashing) and consistent compliance with SSHv2 and MACsec standards. Both models are equivalent.
- **Management Interface Consistency:** The TOE is managed via either a remote CLI session or a directly connected local console CLI, with remote sessions secured using SSHv2. These management options and protocols are consistent across all platforms. Both models are equivalent.
- **Device Functionality:** The difference between routers (MX) and switches (EX) is not security-relevant and does not affect TSF behavior.

3.7 RECOMMENDATIONS/CONCLUSIONS

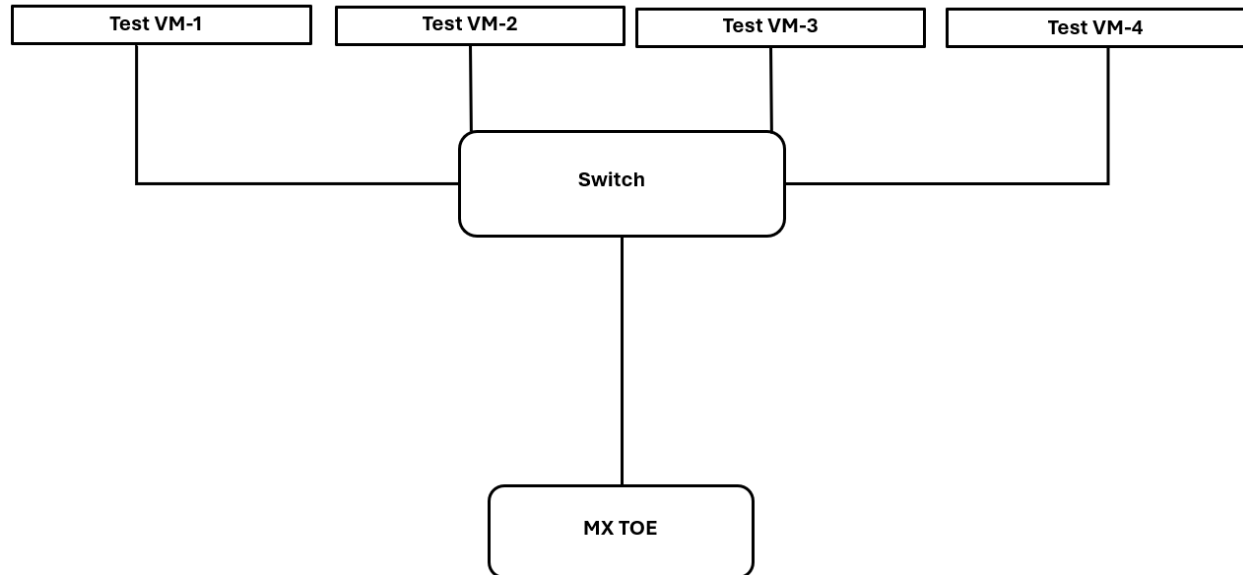
Based on the equivalency rationale outlined, testing will be performed on the following subset to ensure all differences are covered:

- **MX480 with MPC10E-15C (JNPR Penta Silicon MACsec ASIC):** This device is designated for full NDcPP and SSH Package testing. Additionally, it will undergo one round of MACsec testing, specifically for the JNPR Penta Silicon MACsec ASIC installed within the MPC10E-15C line card.
- **MX240 with BCM82756 MACsec ASIC:** This device will be utilized for a separate round of MACsec testing, focusing on the BCM82756 MACsec ASIC in the MIC-3D-10GE-SFP-E within the MPC3E-3D-NG line card.

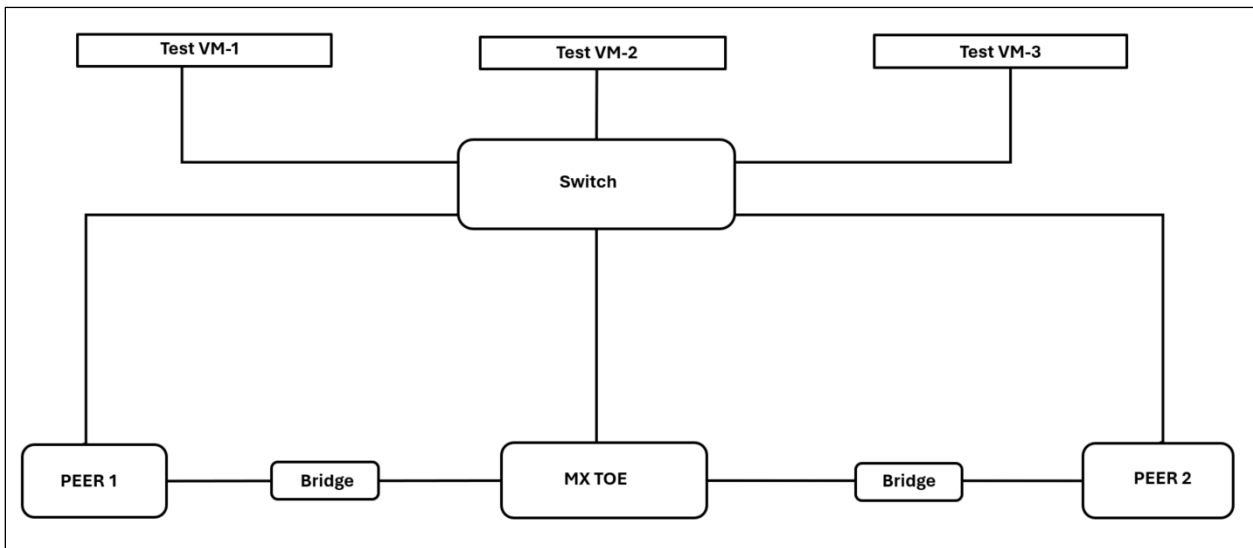
This ensures comprehensive testing of all security-critical components, including the processor and both types of MACsec ASICs.

4 TEST BED DESCRIPTIONS

4.1 AUDIT/AUTH/CRYPTO/UPDATE/SSH



4.2 MACSEC



4.3 CONFIGURATION INFORMATION

Device Details Table		
Device Details	Network Details	System Details

Device Name	Function	Protocols	OS, including version	Timing Source	Software & Tools, including version
MX TOE	TOE	SSH	Junos OS 24.4R2-S2.2	Manual	NA
PEER 1	Peer Device for TOE	SSH	Junos OS 24.4R2-S2.2	Manual	NA
PEER 2	Peer Device for TOE	SSH	Junos OS 23.4R2-S5.8	Manual	NA
Test VM-1	For TOE Configuration, Packet Capture and Audit Server	SSH	Ubuntu 22.04.5	Manual	OpenSSH (8.9p1) tcpdump (4.99.1)
Test VM-2	For PEER Configuration and Packet Capture	SSH	Ubuntu 22.04.5	Manual	OpenSSH (8.9p1) tcpdump (4.99.1)
Test VM-3	For acting as NTP server	SSH	Ubuntu 22.04.5	Manual	OpenSSH (7.6p1) tcpdump (4.9.3)
Test VM-4	For TOE and PEER Configuration	SSH	Ubuntu 18.04.6	Manual	OpenSSH (7.6p1) tcpdump (4.9.3)
Bridge	For Packet Capture	NA	Ubuntu 20.04.6	Manual	acumen-macsec(1.3) tcpdump (4.99.1)
Switch	Lab Switch	N/A	NA	NA	N/A

4.4 TEST TIME AND LOCATION

All testing was carried out at Acumen Security offices located at 2400 Research Blvd Suite #395, Rockville, MD 20850. Testing occurred from June 2025 to June 2026.

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

5 DETAILED EVALUATION ACTIVITIES

5.1 TSS AND AGD ACTIVITIES

5.1.1 SECURITY AUDIT (FAU)

5.1.1.1 FAU_GEN.1 AUDIT DATA GENERATION (CPP_ND_V3.0E)

5.1.1.1.1 FAU_GEN.1 TSS

For the administrative task of generating/importing, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the TSS should identify what information is logged to identify the relevant key.

Evaluator Findings:

The evaluator examined the TSS row **FAU_GEN.1** and ensured that it identifies what information is logged to identify the relevant cryptographic key during generating/import, changing, or deleting.

The relevant information is found in the following section: **TOE Summary Specification** row **FAU_GEN.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE records the following with each log entry:

- **date and time of the event and/or reaction**
- **type of event and/or reaction**
- **subject identity (where applicable)**
- **the outcome (success or failure) of the event (where applicable).**

In order to identify the key being operated on, the following details are recorded for all administrative actions relating to cryptographic keys (generating, importing, changing and deleting keys):

- **CAK – MACsec CA name is recorded in syslog**
- **SAK – TOE interface associated with the SAK is recorded in syslog**
- **KEK, SAK, ICV – key references provided by process id**
- **Username and key type - The key type (rsa/ecdsa) and the username of the associated user will be recorded when importing an SSH user public key**
- **SSH key configured for SSH public key authentication –the hash of the public key that is to be used for authentication is recorded in syslog**

For distributed TOEs the evaluator shall examine the TSS and ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components.

Evaluator Findings:

The TOE is not a distributed TOE; hence this assurance activity is not applicable.

The evaluator shall ensure that this mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (where applicable to the overall TOE). This includes that the evaluator shall confirm that all components defined as generating audit information for a

particular SFR should also contribute to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component cover all the SFRs that it implements.

Evaluator Findings:

The TOE is not a distributed TOE; hence this assurance activity is not applicable.
--

Verdict:

PASS.

5.1.1.1.2 FAU_GEN.1 AGD

The evaluator shall check the guidance documentation and ensure that it provides an example of each auditable event required by FAU_GEN.1 (i.e. at least one instance of each auditable event, comprising the mandatory, optional and selection-based SFR sections as applicable, shall be provided from the actual audit record).

Evaluator Findings:

The evaluator checked the guidance documentation and ensured that it provides an example of each auditable event required by FAU_GEN.1 (i.e. at least one instance of each auditable event, comprising the mandatory, optional and selection-based SFR sections as applicable, was provided from the actual audit record).
--

The relevant information is found in the following section(s): Event Logging Overview
--

Upon investigation, the evaluator found that Table 6: Auditable Events in the claimed AGD section provides an example of each auditable event required by FAU_GEN.1.

The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP.

Evaluator Findings:

The evaluator made a determination of the administrative actions related to TSF data related to configuration changes. The evaluator examined the guidance documentation and made a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP.

Upon investigation, the evaluator found that the following are applicable:
--

Administrative Activity	Method (Command/GUI Configuration)	Section
Start-up and shutdown of the audit functions	Note: There is no manual startup/shutdown of the local audit function, which is tied to startup/shutdown of the TOE itself, logs for which implicitly indicate the audit function stopping and starting as well. Command Line Interface	Section: Configure Audit Log Options in FIPS Mode
Administrative login and logout	Command Line Interface	Section(s): Sample Output for Remote Administrative Session Termination and Sample Output for User Initiated Termination
Changes to TSF data related to configuration changes	Command Line Interface	Section(s): Configure Administrative Credentials and Privileges, Configure SSH and Console Connection, Configure the Remote Syslog Server, Configure Audit Log Options and Configure MACsec
Generating/import of, changing, or deleting of cryptographic keys	Command Line Interface	Section(s): Delete SSH Keys and Sample Output for User Initiated Termination and Configure a Network Device Collaborative Protection Profile Authorized Administrator
Resetting passwords	Command Line Interface	Section: Configure a Network Device Collaborative Protection Profile Authorized Administrator

The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

Evaluator Findings:

The evaluator documented the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator performed this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

Verdict:

PASS.

5.1.1.2 FAU_GEN.1/MACSEC AUDIT DATA GENERATION (MACSEC) (MOD_MACSEC_V1.0)

5.1.1.2.1 FAU_GEN.1/MACSEC

The evaluator shall complete the evaluation activity for FAU_GEN.1 as described in the NDcPP for the auditable events defined in the PP-Module in addition to the applicable auditable events that are defined in the NDcPP. The evaluator shall also ensure that the administrative actions defined for this PP-Module are appropriately audited.

Evaluator Findings:

The evaluator completed the evaluation activity for FAU_GEN.1 as described in the NDcPP, covering the applicable auditable events defined in both the NDcPP and the MACsec PP-Module. These events are detailed in the FAU_GEN.1 section of the ST.

The relevant information is found in the following section: **TOE Summary Specification** row **FAU_GEN.1/MACSEC**

Upon investigation, the evaluator found that the TSS states that:

The Auditable events are specified in Table 12 – Security Functional Requirements and Auditable Events and Table 13 – Security Functional Requirements and Auditable Events/MACsec (in the ST). The TOE records the following with each log entry:

- **date and time of the event and/or reaction**
- **type of event and/or reaction**
- **subject identity (where applicable)**
- **the outcome (success or failure) of the event (where applicable).**

The evaluator also verified that the administrative actions defined for this PP-Module are appropriately audited. The relevant information was found in **Table 6: Auditable Events** of section **Event Logging Overview** in the AGD, where the entries for **FCS_MACSEC_EXT.1, FCS_MACSEC_EXT.3, FCS_MACSEC_EXT.4, and FPT_RPL.1** demonstrate the audit records corresponding to the administrative actions.

Verdict:

PASS.

5.1.1.3 FAU_GEN.2 USER IDENTITY ASSOCIATION (CPP_ND_V3.0E)

5.1.1.3.1 FAU_GEN.2 TSS

The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

5.1.1.3.2 FAU_GEN.2 AGD

The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

5.1.1.4 FAU_STG_EXT.1 PROTECTED AUDIT EVENT STORAGE (CPP_ND_V3.0E)

5.1.1.4.1 FAU_STG_EXT.1 TSS

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

Evaluator Findings:

The evaluator examined the TSS row **FAU_STG_EXT.1** and ensured that it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

The relevant information is found in the following section: **TOE Summary Specification** row **FAU_STG_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

Additionally, syslog can be configured to transmit audit data securely via Netconf over an SSH tunnel where the TOE is the SSH server. Audit data is transmitted to the configured syslog server in real time.

The evaluator shall examine the TSS to ensure it describes whether the TOE is a standalone TOE that stores audit data locally or a distributed TOE that stores audit data locally on each TOE component or a distributed TOE that contains TOE components that cannot store audit data locally on themselves but need to transfer audit data to other TOE components that can store audit data locally.

Evaluator Findings:

The TOE is not a distributed TOE. The TSS row **FAU_STG_EXT.1** states that: **The TOE stores audit logs locally in an audit file. The TOE is standalone. The logs are of persistent nature and these local audit logs are stored in /var/log/ in the underlying filesystem.**

The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally.

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

The evaluator shall examine the TSS to ensure that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components.

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

The evaluator shall examine the TSS to ensure that it details whether the transmission of audit data to an external IT entity can be done in real-time, periodically, or both. In the case where the TOE is capable of performing transmission periodically, the evaluator shall verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible acceptable frequency for the transfer of audit data.

Evaluator Findings:

The evaluator examined the TSS row **FAU_STG_EXT.1** and ensured that it details whether the transmission of audit information to an external IT entity can be done in real-time, periodically, or both. In the case where the TOE is capable of performing transmission periodically, the evaluator verified that the TSS provides details about what event stimulates the transmission to be made as well as the possible acceptable frequency for the transfer of audit data.

The relevant information is found in the following section: **TOE Summary Specification** row **FAU_STG_EXT.1**

Upon investigation, the evaluator found that the TSS states that: **Audit data is transmitted to the configured syslog server in real time.**

For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how audit data transfer to the external audit server is implemented among the different TOE components (e.g. every TOE components does its own transfer or the data is sent to another TOE component for central transfer of all audit events to the external audit server).

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

The evaluator shall examine the TSS to ensure it describes the amount of audit data that can be stored locally and how these records are protected against unauthorized modification or deletion.

Evaluator Findings:
The evaluator examined the TSS row FAU_STG_EXT.1 and ensured it describes the amount of audit data that are stored locally and how these records are protected against unauthorized modification or deletion. The relevant information is found in the following section: TOE Summary Specification row FAU_STG_EXT.1 Upon investigation, the evaluator found that the TSS states that: The TOE uses an active log file and archive files whose sizes are configurable from 64kB to 1GB, default being 1MB. Only authorized Security Administrators are allowed to read or delete log files. These log files can be accessed using command-line interface (CLI) or via direct filesystem access, but only after successful authentication as a Security Administrator.

The evaluator shall examine the TSS to ensure it describes the method implemented for local logging, including format (e.g. buffer, log file, database) and whether the logs are persistent or non-persistent.

Evaluator Findings:
The evaluator examined the TSS row FAU_STG_EXT.1 and ensured it describes the method implemented for local logging, including format (e.g. buffer, log file, database) and whether the logs are persistent or non-persistent. The relevant information is found in the following section: TOE Summary Specification row FAU_STG_EXT.1 Upon investigation, the evaluator found that the TSS states that: The logs are of persistent nature and these local audit logs are stored in /var/log/ in the underlying filesystem. The TOE uses an active log file and archive files whose sizes are configurable from 64kB to 1GB, default being 1MB.

The evaluator shall examine the TSS to ensure it describes the conditions that must be met for authorized deletion of audit records.

Evaluator Findings:

The evaluator examined the TSS row **FAU_STG_EXT.1** and ensured it describes the conditions that must be met for authorized deletion of audit records.

The relevant information is found in the following section: **TOE Summary Specification** row **FAU_STG_EXT.1**

Upon investigation, the evaluator found that the TSS states that: **Only authorized Security Administrators are allowed to read or delete log files.**

The evaluator shall examine the TSS to ensure it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

Evaluator Findings:

The evaluator examined the TSS row **FAU_STG_EXT.1** and ensured that it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE is detailed in the TSS.

The relevant information is found in the following section: **TOE Summary Specification** row **FAU_STG_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

These audit logs are automatically overwritten by newer logs as per the administrative configurable limits on storage volume.

When the active log file reaches its maximum size, it's compressed and archived (logfile.0.gz), and a new active log file is created thereby overwriting the old logs.

Upon reaching the maximum archive file limit, the oldest archive log file is purged to accommodate the newly archived log file.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes which TOE components are storing audit information locally and which components are buffering audit information and forwarding the information to another TOE component for local storage. For every component the TSS shall describe the behaviour when local storage space or buffer space is exhausted.

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:

PASS.

5.1.1.4.2 FAU_STG_EXT.1 AGD

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

Evaluator Findings:

The evaluator examined the guidance documentation section '**Sample Syslog Server Configuration on a Linux System**' and ensured it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as

well as configuration of the TOE needed to communicate with the audit server.

Upon investigation, the evaluator found that the claimed AGD section '**Sample Syslog Server Configuration on a Linux System**' provides necessary CLI commands to establish the trusted channel to the audit server as well as configuration of the TOE needed to communicate with the audit server. The AGD states that:

A syslog server receives the syslog messages streamed from the device. The syslog server must have an SSHv2 client with NETCONF 1.0 support configured to receive the streamed syslog messages. Junos OS supports NETCONF as defined by RFC6241 (NETCONF) and RFC6242 (Using NETCONF protocol over SSH). Use the configuration details and establish a session between the target of evaluation (TOE) and the audit server.

To configure event logging to a remote server when the SSH connection to the TOE is initiated from the remote system log server.

1. Generate an RSA public key on the remote syslog server. The TOE's administrator can also use ECDSA keys on the syslog server.

```
$ ssh-keygen -b 2048 -t rsa -C 'syslog-monitor key pair' -f ~/.ssh/syslog-monitor
```

The TOE's administrator will be prompted to enter the desired pass phrase. The storage locations for the `syslog-monitor` key pair is displayed.

2. On the TOE, create a class named monitor that has permission to trace events.

```
[edit system login]
```

```
security-administrator@host:fps# set class monitor permissions trace
```

3. Create a user named syslog-mon with the class monitor, and with public key-based or password-based authentication.

For public key-based authentication:

```
[edit system login]
```

```
security-administrator@host:fps# set user syslog-mon class monitor authentication ssh-rsa "public-key"
```

For password-based authentication:

```
[edit system login]
```

```
security-administrator@host:fps# set user syslog-mon class monitor authentication (plaintext-  
password | encrypted-password)
```

4. Set up NETCONF with SSH.

```
[edit system services]
```

```
security-administrator@host:fps# set netconf ssh
```

5. Configure syslog to log all the messages at `/var/log/file-name`.

```
[edit system]
```

```
security-administrator@host:fps# set syslog file file-name any any  
commit
```

6. On the remote system log server, start up the SSH agent `ssh-agent`. The start up is required to simplify the handling of the `syslog-monitor` key.

```
$ eval `ssh-agent -s`
```

7. On the remote syslog server, add the `syslog-monitor` key pair to the `ssh-agent`.

```
$ ssh-add ~/.ssh/syslog-monitor
```

The TOE's administrator will be prompted to enter the desired passphrase. Enter the same passphrase used in Step 1.

8. After logging in to the `external_syslog_server` session, establish a tunnel to the device and start NETCONF.

```
security-administrator@host:fips# $ssh syslog-mon@NDcPP_TOE -s netconf > test.out
```

9. After NETCONF is established, configure a system log events message stream. This RPC will cause the NETCONF service to start transmitting messages over the SSH connection that is established.

```
<rpc><get-syslog-events><stream>messages</stream></get-syslog-events></rpc>
```

Based on these findings, this assurance activity is considered satisfied.

The evaluator shall also examine the guidance documentation to ensure it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and “cleared” periodically by sending the data to the audit server.

Evaluator Findings:

The evaluator also examined the guidance documentation section ‘**Sample Syslog Server Configuration on a Linux System**’ and determined that it describes the relationship between the local audit data and the audit data that are sent to the audit log server.

Upon investigation, the evaluator found that the claimed AGD section states that:

A secure Junos OS environment requires auditing of events and storing them in a local audit file. The recorded events are simultaneously sent to an external syslog server. A syslog server receives the syslog messages streamed from the device.

Based on these findings, this assurance activity is considered satisfied.

The evaluator shall examine the guidance documentation to ensure it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

Evaluator Findings:

The evaluator examined the guidance documentation section ‘**Configure Audit Log Options on Device in FIPS Mode**’ and ensured it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

The evaluator confirmed that no specific configuration is required for this protection, as the guidance states: **Only authenticated administrators are authorized to delete locally stored audit data.**

The guidance further demonstrates this by providing the commands used to **Clear the local audit data:**

[edit]

```
security-administrator@host:fips# clear log syslog
```

If the storage size is configurable, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying the required parameters.

Evaluator Findings:

Since the audit storage size is configurable, the evaluator reviewed the guidance Documentation section '**Configure Audit Log Options on Device in FIPS Mode**' to ensure it contains instructions on specifying the required parameters.

The evaluator verified that the guidance provides instructions on configuring the audit log storage size by specifying the number and size of archived log files using the following commands:

To configure audit log options:

1. Specify the number of files to be archived in the system logging facility.

[edit system syslog]

security-administrator@host:fips# set archive files 2

2. Specify the file in which to log data.

[edit system syslog]

security-administrator@host:fips# set file syslog any any

3. Specify the size of files, in bytes, to be archived. The size can also be specified in shorthand form, for example, `10m` for 10 MB and `1g` for 1 GB.

[edit system syslog]

security-administrator@host:fips# set file syslog archive size 10000000

Additionally, the AGD states that:

The local logs are automatically overwritten according to configurable limits on storage volume. The default maximum size is 1 MB, which can be modified by the user, using the "size" argument for the "set system syslog file <filename> archive" CLI command.

The Junos OS defines an active log file and a number of "archive" files (10 by default, but configurable from 1 to 1000). When the active log file reaches its maximum size, the logging utility closes the file, compresses it, and names the compressed archive file 'logfile.0.gz'. The logging utility then opens and writes to a new active log file. When the new active log file reaches the configured maximum size 'logfile.0.gz' is renamed 'logfile.1.gz', and the active log file is closed, compressed, and renamed 'logfile.0.gz'. When the maximum number of archive files is reached and when the size of the active file reaches the configured maximum size, the contents of the oldest archived file are deleted so the current active file can be archived.

If more than one selection is made for FAU_STG_EXT.1.5, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying which action is performed when the local storage space is full.

Evaluator Findings:

Only one selection is made for FAU_STG_EXT.1.5; therefore, this activity is not applicable.

Verdict:

PASS.

5.1.2 CRYPTOGRAPHIC SUPPORT (FCS)

5.1.2.1 FCS_CKM.1 CRYPTOGRAPHIC KEY GENERATION (CPP_ND_V3.0E)

5.1.2.1.1 FCS_CKM.1 TSS [TD0921]

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

Evaluator Findings:				
The evaluator ensured that the TSS row FCS_CKM.1 identifies the key sizes supported by the TOE. The evaluator examined the TSS to verify that it identifies the usage for each scheme.				
The relevant information is found in the following section: TOE Summary Specification row FCS_CKM.1				
Upon investigation, the evaluator found that the TSS states that: The TOE uses several cryptographic key generation schemes with different key sizes for each specific operation. The table below details the same:				
Key Generation Scheme	SFR	Key Size(s)	Algorithm	Usage
SSH Key Generation (RSA)	FCS_CKM.1, FCS_SSHS_EXT.1 FCS_SSH_EXT.1	2048-bit, 3072-bit, 4096-bit	RSA	Asymmetric key generation with respective hashing algorithms and curves.
SSH Key Generation (ECDSA)	FCS_CKM.1, FCS_SSHS_EXT.1 FCS_SSH_EXT.1	P-256, P-384, P-521	ECDSA	

Verdict:

PASS.

5.1.2.1.2 FCS_CKM.1 AGD [TD0921]

The evaluator shall verify that the AGD instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

Evaluator Findings:
The evaluator verified that the AGD section 'Configure SSH on the Evaluated Configuration' instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target. The relevant information is found in the following section(s): Generate a New SSH Host Key, Configure SSH on the Evaluated Configuration Upon investigation, the evaluator found that in the section Generate a New SSH Host Key, the AGD instructs the administrator that to generate a new SSH host key, issue one of the following commands from the shell prompt, selecting the appropriate key size: • <code>ssh-keygen -t ecdsa -b [256 384 521] -f /etc/ssh/ssh_host_ecdsa_key</code> • <code>ssh-keygen -t rsa -b [2048 3072 4096] -f /etc/ssh/ssh_host_rsa_key</code> In the section Configure SSH on the Evaluated Configuration, the AGD specifies how to configure the SSH hostkey algorithms. The relevant configuration commands are: [edit] <pre>security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256 security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384 security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521 security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa</pre>

NOTE: Although the last hostkey algorithm option mentioned above only mentions RSA, it covers all the claimed RSA algorithms, that is SSH-RSA, RSA-SHA2-256, and RSA-SHA2-512.

Verdict:

PASS.

5.1.2.2 FCS_CKM.2 CRYPTOGRAPHIC KEY ESTABLISHMENT (CPP_ND_V3.0E)

5.1.2.2.1 FCS_CKM.2 TSS

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. It is sufficient to provide the scheme, SFR, and service in the TSS. The intent of this activity is to be able to identify the scheme being used by each service. This would mean, for example, one way to document scheme usage could be as shown in the table below. The information provided in this example does not necessarily have to be included as a table but can be presented in other ways as long as the necessary data is available.

Evaluator Findings:

The evaluator ensured that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. The evaluator examined the TSS section row **FCS_CKM.2** to verify that it identifies the usage for each scheme. It is sufficient to provide the scheme, SFR, and service in the TSS.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_CKM.2**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports several key establishment and key generation schemes. The table below details the same:

Scheme	SFR(s)	Usage/Service
SSH EC Key Agreement	FCS_CKM.2	SSH key exchange

Verdict:

PASS.

5.1.2.2.2 FCS_CKM.2 AGD

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

Evaluator Findings:

The evaluator verified that the AGD guidance section '**Configure SSH on the Evaluated Configuration**' instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

The relevant information is found in the following section: **Configure SSH on the Evaluated Configuration**

Upon investigation, the evaluator found that the AGD activity states that:

To configure SSH on the TOE:

2. Specify the SSH key-exchange for Elliptic Curve Diffie-Hellman keys for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256
```

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384
```

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521
```

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.2.3 FCS_CKM.4 CRYPTOGRAPHIC KEY DESTRUCTION (CPP_ND_V3.0E)

5.1.2.3.1 FCS_CKM.4 TSS

The evaluator shall examine the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target. The evaluator shall confirm that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator shall check that this is consistent with the operation of the TOE.

Evaluator Findings:

The evaluator examined the TSS row **FCS_CKM.4** to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_CKM.4**

Upon investigation, the evaluator found that the TSS states that:

The TOE stores plaintext keys in volatile and non-volatile storage. Encrypted keys are stored on disk. The TOE satisfies all requirements for destruction of CSPs, plaintext secret and private cryptographic keys as specified in Table 18 – Zeroization of Keys and CSP (of the ST) as shown below.

Keys/CSPs	Purpose	Method of storage	Storage Location	Method of Zeroization
SSH Private Host Key	The first time SSH is configured, the key is generated. Used to identify the host.	Plaintext	File format on SDD	single overwrite consisting of zeroes using the CLI command: "request vmhost zeroize no-forwarding"
	Loaded into memory to complete session establishment	Plaintext	Memory	Memory free() operation is performed by Junos upon

				session termination
SSH Session Key	Session keys used with SSH, AES 128, 256, hmac-sha2-256 or hmac-sha2-512 key (256 or 512), DH Private Key (elliptic curve 256/384/521-bits). Derived dynamically during each SSH session via SSH KDF.	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
User Password	Plaintext value as entered by user	Plaintext as entered	Processed in Memory	Memory free() operation is performed by Junos upon completion of authentication
		Hashed when stored (SHA256 and SHA512)	Stored on disk	single overwrite consisting of zeroes using the CLI command: "request vmhost zeroize no-forwarding"
RNG State	Internal state and seed key of RNG. Originates during instantiation when the DRBG initializes its internal state variables using seed material harvested from the noise source.	Plaintext	Memory	Handled by kernel, overwritten with zero's at reboot.
MACsec CAK	Pre-shared, static Connectivity Association Key. Defined statically through manual administrative configuration as a PSK.	Encrypted using GCM-AES-256 using System Master Password	Stored in config file	single overwrite consisting of zeroes using the CLI command: "request vmhost zeroize no-forwarding"
MACsec SAK	Security Association Key. Generated using AES-CMAC-128 or	Plaintext	Memory	Memory free() operation is performed by Junos upon

	AES-CMAC-256 KDF function based on the configured cipher suite during MKA session initialization.			session termination
MACsec KEK	Key Encryption Key. Derived from the CAK during the MKA session initialization.	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
MACsec ICK	Integrity Check Key. Derived from CAK using AES_CMAC KDF during MKA session initialization.	Plaintext	Memory	Memory free() operation is performed by Junos upon session termination
System Master Password	Password used to derive encryption key used for protecting MACsec CAK. Originates during initial system provisioning.	Plaintext	disk	single overwrite consisting of zeroes using the CLI command: "request vmhost zeroize no-forwarding"
The evaluator confirmed that the description of keys and storage locations is consistent with the functions carried out by the TOE.				

The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

Evaluator Findings:

The evaluator ensured that the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_CKM.4**

Upon investigation, the evaluator found that the TSS states that:

The full mapping of which keys are destroyed, how they are stored, and how they are zeroized is provided in **Table 18 – Zeroization of Keys and CSP** (of the ST).

Table 18 further specifies the zeroization method used by the TOE, indicating that keys stored in non-volatile memory are destroyed through a **single overwrite consisting of zeroes**, invoked via the TOE's CLI interface using the command **"request vmhost zeroize no-forwarding"** option.

Note that where selections involve 'destruction of reference' (for volatile memory) or 'invocation of an interface' (for non-volatile memory) then the relevant interface definition is examined by the evaluator to ensure that the interface supports the selection(s) and description in the TSS. In the case of nonvolatile memory, the evaluator includes in their examination the relevant interface description for each media type on which plaintext keys are stored. The presence of OS-level and storage device-level swap and cache files is not examined in the current version of the Evaluation Activity.

Evaluator Findings:

The evaluator ensured that the selections involved 'destruction of reference' (for volatile memory) and 'invocation of an interface' (for non-volatile memory), and that the relevant interface definition was examined to ensure that the interface supports the selections and description in the TSS. For non-volatile memory, the evaluator also examined the relevant interface description for each media type on which plaintext keys are stored.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_CKM.4**

Upon investigation, the evaluator found that the TSS states that:

The TOE satisfies all requirements for destruction of CSPs, plaintext secret and private cryptographic keys as specified in Table 18 – Zeroization of Keys and CSP (of the ST).

Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator shall check that the TSS identifies the encryption method and the key-encrypting-key used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

Evaluator Findings:

Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator checked that the TSS row **FCS_CKM.4** identifies the encryption method and the key-encrypting-key used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_CKM.4**

Upon investigation, the evaluator found that the TSS states that:

The TOE satisfies all requirements for destruction of CSPs, plaintext secret and private cryptographic keys as specified in Table 18 – Zeroization of Keys and CSP (of the ST).

Table 18 further specifies that the **MACsec CAK, a pre-shared static Connectivity Association Key, is encrypted using GCM-AES-256 using the System Master Password.** The table also specifies that the **System Master Password, which is used to derive the encryption key used for protecting the MACsec CAK, is zeroized by a single overwrite consisting of zeroes using the CLI command “request vmhost zeroize no-forwarding.”**

The evaluator shall check that the TSS identifies any configurations or circumstances that may not conform to the key destruction requirement (see further discussion in the Guidance Documentation section below). Note that reference may be made to the Guidance Documentation for description of the detail of such cases where destruction may be prevented or delayed.

Evaluator Findings:

The evaluator checked that the TSS row **FCS_CKM.4** identifies any configurations or circumstances that may not conform to the key destruction requirement (see further discussion in the Guidance Documentation section below).

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_CKM.4**

Upon investigation, the evaluator found that the TSS states that:

There are no configurations or circumstances that may not conform to the key destruction requirements.

Where the ST specifies the use of “a value that does not contain any CSP” to overwrite keys, the evaluator shall examine the TSS to ensure that it describes how that pattern is obtained and used, and that this justifies the claim that the pattern does not contain any CSPs.

Evaluator Findings:

The ST does not specify the use of “a value that does not contain any CSP.” Therefore, this assurance activity is NA.

Verdict:

PASS.

5.1.2.3.2 FCS_CKM.4 AGD

A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer. For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command[3] and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Evaluator Findings:

The evaluator checked that the guidance documentation section ‘**Zeroize the System For FIPS Mode**’ and ‘**Overview of Zeroization to Clear System Data for FIPS Mode**’ identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used).

The relevant information is found in the following section(s): **Zeroize the System For FIPS Mode** and **Overview of Zeroization to Clear System Data for FIPS Mode**

The evaluator examined AGD section 'Zeroize the System For FIPS Mode' and found that:
The entire operation can take considerable time depending on the size of the media, but the system removes all the critical security parameters (CSPs) within a few seconds. The physical environment must remain secure until the zeroization process is complete.

Additionally, the guidance document section 'Overview of Zeroization to Clear System Data for FIPS Mode' includes a note stating that **There are no configurations that do not conform to the key destruction requirement.**

Verdict:

PASS.

5.1.2.4 FCS_COP.1/CMAC CRYPTOGRAPHIC OPERATION (AES-CMAC KEYED HASH ALGORITHM) (MOD_MACSEC_V1.0)

5.1.2.4.1 FCS_COP.1/CMAC TSS

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

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_COP.1/CMAC** and verified that the TSS specifies the following values used by the AES-CMAC function: key length, hash function used, block size, and output MAC length.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_COP.1/CMAC**

Upon investigation, the evaluator found that the TSS states that:

The TOE performs keyed-hash message authentication in accordance with AES-CMAC. The cryptographic key sizes are 128 and 256 bits. The hash function used is AES. The block size is 128 bits. The message digest size (output MAC length) is 128 bits for MACsec cryptographic operations.

Verdict:

PASS.

5.1.2.4.2 FCS_COP.1/CMAC AGD

There are no guidance evaluation activities (EAs) for this component.

5.1.2.5 FCS_COP.1/DATAENCRYPTION CRYPTOGRAPHIC OPERATION (AES DATA ENCRYPTION/DECRYPTION) (CPP_ND_V3.0E)

5.1.2.5.1 FCS_COP.1/DATAENCRYPTION TSS

The evaluator shall examine the TSS to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption.

Evaluator Findings:

The evaluator examined the TSS row **FCS_COP.1/DataEncryption** to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_COP.1/DataEncryption**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports AES encryption and decryption used in CBC, CTR and GCM modes as specified in ISO 18033-3, ISO 10116 and ISO 19772 respectively. The AES key sizes supported are 128 and 256 bits. AES is implemented in the following protocols: SSH, MACsec.

Verdict:

PASS.

5.1.2.5.2 FCS_COP.1/DATAENCRYPTION AGD

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected mode(s) and key size(s) defined in the Security Target supported by the TOE for data encryption/decryption.

Evaluator Findings:

The evaluator verified that the AGD guidance sections '**Configure SSH on the Evaluated Configuration**' and '**Configuring MACsec with keychain using Layer 3 Traffic**' instructs the administrator how to configure the TOE to use the selected mode(s) and key size(s) defined in the Security Target supported by the TOE for data encryption/decryption.

The relevant information is found in the following section(s): **Step 4 of Configure SSH on the Evaluated Configuration** and **Step 4 of Configuring MACsec with keychain using Layer 3 Traffic**

Step 4 of the section **Configure SSH on the Evaluated Configuration** includes commands to explicitly set the allowed ciphers for SSH protocol version 2:

4. Specify the ciphers allowed for protocol version 2.

[edit]

```
security-administrator@host:fips# set system services ssh ciphers aes128-cbc
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-cbc
```

```
security-administrator@host:fips# set system services ssh ciphers aes128-ctr
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-ctr
```

```
security-administrator@host:fips# set system services ssh ciphers aes128-gcm@openssh.com
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-gcm@openssh.com
```

Similarly, **Step 4** of the section **Configuring MACsec with keychain using Layer 3 Traffic** instructs:

4. Associate the preshared keychain name with the connectivity association.

[edit]

```
security-administrator@host:fips# set security macsec connectivity-association CA1 pre-shared-key-chain macsec-kc1
```

```
security-administrator@host:fips# set security macsec connectivity-association CA1 offset 50
```

```
security-administrator@host:fips# set security macsec connectivity-association CA1 cipher-suite gcm-aes-256
```

NOTE: The cipher value can also be set as cipher-suite gcm-aes-128.

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.2.6 FCS_COP.1/HASH CRYPTOGRAPHIC OPERATION (HASH ALGORITHM) (CPP_ND_V3.0E)

5.1.2.6.1 FCS_COP.1/HASH TSS

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

Evaluator Findings:

The evaluator checked that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS row **FCS_COP.1/Hash**.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_COP.1/Hash**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports cryptographic hashing services in accordance with SHA-1, SHA2-256, SHA2- 384 and SHA2-512 for SSHv2 protocol with message digest size of 160, 256, 384, 512.

Additionally, the TOE also performs hashing while performing image integrity check during update, running self-tests and storing password. Trusted Update signature verification uses ECDSA on P-256 w/SHA-256. Cryptographic hashing for password conditioning, password hashing, and self-testing (verifying integrity of system files) is performed using SHA256 and SHA512.

Verdict:

PASS.

5.1.2.6.2 FCS_COP.1/HASH AGD

The evaluator shall check the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

Evaluator Findings:

The evaluator checked the AGD to determine that any configuration that is required to configure the required hash sizes is present.

The relevant information is found in the following section(s): **Step 2** of the section **Configure a Network Device Collaborative Protection Profile Authorized Administrator** and **Configure SSH on the Evaluated Configuration**

The evaluator found that in the AGD, **Step 2** of the section **Configure a Network Device Collaborative Protection Profile Authorized Administrator** states that:

2. Configure the hashed algorithm SHA256 or SHA512 for plain-text passwords. SHA512 is the default hashing algorithm.

[edit]

security-administrator@host:fips# set system login password format sha512

Lastly the evaluator examined the AGD section **Configure SSH on the Evaluated Configuration** and found that:

To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa
```

NOTE: Although the last hostkey algorithm option mentioned above only mentions RSA, it covers all the claimed RSA algorithms, that is SSH-RSA, RSA-SHA2-256, and RSA-SHA2-512.

The SHA algorithm used with an SSH host-key algorithm is determined by the host-key algorithm itself and does not require separate configuration.

2. Specify the SSH key-exchange for Diffie-Hellman keys for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256
```

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384
```

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521
```

The SHA2 function used with an ECDH key exchange algorithm is determined by the selected elliptic curve group and does not require separate configuration.

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.2.7 FCS_COP.1/KEYEDHASH CRYPTOGRAPHIC OPERATION (KEYED HASH ALGORITHM) (CPP_ND_V3.0E)

5.1.2.7.1 FCS_COP.1/KEYEDHASH TSS

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

Evaluator Findings:

The evaluator examined the TSS row **FCS_COP.1/KeyedHash** 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.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_COP.1/KeyedHash**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports keyed-hash message authentication in accordance with HMAC-SHA-256, HMAC-SHA-512 with key sizes 256 and 512 bits and message digest sizes 256, 512 bits. The key length, hash function used, block size, and output MAC lengths are identified in the table below.

Algorithm	Key Length	Hash function	Block Size	Output MAC length
-----------	------------	---------------	------------	-------------------

HMAC-SHA-256	256 bits	SHA-256	512 bits	256 bits
HMAC-SHA-512	512 bits	SHA-512	1024 bits	512 bits

Verdict:

PASS.

5.1.2.7.2 FCS_COP.1/KEYEDHASH AGD

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the values used by the HMAC function: key length, hash function used, block size, and output MAC length used defined in the Security Target supported by the TOE for keyed hash function.

Evaluator Findings:
The evaluator verified that the AGD instructs the administrator how to configure the TOE to use the values used by the HMAC function: key length, hash function used, block size, and output MAC length used defined in the Security Target supported by the TOE for keyed hash function. The relevant information is found in the following section: Step 3 of the section Configure SSH on the Evaluated Configuration Upon investigation, the evaluator found that in the AGD, Step 3 of the section Configure SSH on the Evaluated Configuration states that: 3. Specify all the permissible message authentication code algorithms for SSHv2 [edit] security-administrator@host:fips# set system services ssh macs hmac-sha2-256 security-administrator@host:fips# set system services ssh macs hmac-sha2-512 Supported MACs algorithm: hmac-sha2-256 Hash-based MAC using Secure Hash Algorithm (SHA2) hmac-sha2-512 Hash-based MAC using Secure Hash Algorithm (SHA2)

Verdict:

PASS.

5.1.2.8 FCS_COP.1/MACSEC CRYPTOGRAPHIC OPERATION (MACSEC AES DATA ENCRYPTION AND DECRYPTION) (MOD_MACSEC_V1.0)

5.1.2.8.1 FCS_COP.1/MACSEC TSS

The evaluator shall verify that the TSS describes the supported AES modes that are required for this PP- Module in addition to the ones already required by the NDcPP in FCS_COP.1/DataEncryption.

Evaluator Findings:
The evaluator reviewed the TSS row FCS_COP.1/MACSEC and verified that the TSS describes the supported AES modes that are required for this PP-Module in addition to the ones already required by the NDcPP in FCS_COP.1/DataEncryption. The supported AES mode that is required for this PP-Module is GCM as specified in ISO 19772. The supported AES modes required by the NDcPP in FCS_COP.1/DataEncryption are CBC, CTR and GCM as specified in ISO 10116-3, ISO 10116 and ISO 19772 respectively.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_COP.1/MACSEC**

Upon investigation, the evaluator found that the TSS states that:

The TOE performs key wrap encryption and decryption in accordance with the AES Key Wrap mechanism specified in NIST SP 800-38F, using AES with key sizes of 128 and 256 bits. The TOE also performs data encryption and decryption using AES in Galois/Counter Mode (GCM) as specified in ISO 19772, with key sizes of 128 and 256 bits.

Verdict:

PASS.

5.1.2.8.2 FCS_COP.1/MACSEC AGD

There are no guidance EAs for this component.

5.1.2.9 FCS_COP.1/SIGGEN CRYPTOGRAPHIC OPERATION (SIGNATURE GENERATION AND VERIFICATION) (CPP_ND_V3.0E)

5.1.2.9.1 FCS_COP.1/SIGGEN TSS [TD0921]

The evaluator shall examine the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services.

Evaluator Findings:

The evaluator examined the TSS row **FCS_COP.1/SigGen** to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_COP.1/SigGen**

Upon investigation, the evaluator found that the TSS states that:

The TOE provides cryptographic signature generation and verification services in accordance with the following cryptographic algorithms:

- **RSA: Key sizes of 2048, 3072, or 4096 bits**
- **ECDSA: Key sizes of 256, 384 or 521 bits**

Verdict:

PASS.

5.1.2.9.2 FCS_COP.1/SIGGEN AGD [TD0921]

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services.

Evaluator Findings:

The evaluator verified that the AGD instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services.

The relevant information is found in the following section(s): **Configure SSH on the Evaluated Configuration, SSH Host Keys, Generate a New SSH Host Key and Configure a Network Device Collaborative Protection Profile Authorized Administrator**

Upon investigation, the evaluator found that the AGD section **Configure SSH on the Evaluated Configuration** states that:

To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521
security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa
```

NOTE: Although the last hostkey algorithm option mentioned above only mentions RSA, it covers all the claimed RSA algorithms, that is SSH-RSA, RSA-SHA2-256, and RSA-SHA2-512.

In addition, the section **SSH Host Keys** explains that the TOE generates the SSH host keys automatically when the TOE is in FIPS mode and reboots. The SSH host keys located in the following files:

- /etc/ssh/fips_ssh_host_ecdsa_key
- /etc/ssh/fips_ssh_host_rsa_key

Upon investigation, the evaluator found that in the section **Generate a New SSH Host Key**, the AGD instructs the administrator that **to generate a new SSH host key, issue one of the following commands from the shell prompt, selecting the appropriate key size:**

- ssh-keygen -t ecdsa -b [256|384|521] -f /etc/ssh/ssh_host_ecdsa_key
- ssh-keygen -t rsa -b [2048|3072|4096] -f /etc/ssh/ssh_host_rsa_key

In Addition, the evaluator found that the AGD section **Configure a Network Device Collaborative Protection Profile Authorized Administrator** states that:

To configure public key authentication:

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin
authentication ssh-ecdsa <ecdsa-public-key>
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin
authentication ssh-rsa <rsa-public-key>
```

The algorithms and keys configured as seen above in turn effectively configure the corresponding signature functionalities.

Verdict:

PASS.

5.1.2.10 FCS_MACSEC_EXT.1 MACSEC (MOD_MACSEC_V1.0)

5.1.2.10.1 FCS_MACSEC_EXT.1 TSS [TD0884]

The evaluator shall examine the TSS to verify that it describes the ability of the TSF to implement MACsec in accordance with IEEE 802.1AE-2018.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MACSEC_EXT.1** and verified that the TSS describes the ability of the

TSF to implement MACsec in accordance with IEEE 802.1AE-2018.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MACSEC_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE's MACsec functionality is implemented in full accordance with the IEEE 802.1AE-2018 standard, encompassing all mandatory and recommended requirements.

The evaluator shall also determine that the TSS describes the ability of the TSF to derive SCI values from peer MAC address and port data and to reject traffic that does not have a valid SCI.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MACSEC_EXT.1** and verified that the TSS describes the ability of the TSF to derive SCI values from peer MAC address and port data and to reject traffic that does not have a valid SCI.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MACSEC_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

Secure channel is identified by Secure Channel Identifier (SCI) that is comprised of a globally unique peer MAC address and its 16-bit Port Identifier, unique within the system that has been allocated that address. SCI (8 octets) is appended to every MKPDU packet and the TOE can be configured to enforce SCI tagging such that packets are rejected if they do not have a valid SCI.

Finally, the evaluator shall check the TSS for an assertion that only the claimed EtherTypes are accepted by the MACsec interface. Where the SFR selection has provided additional EtherType support, the TSS must describe the usage of each of those frame types within the context of MKA, MACsec, and MAC control frame support.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MACSEC_EXT.1** and verified that the TSS asserts that only the claimed EtherTypes are accepted by the MACsec interface.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MACSEC_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TSF enforces a strict filter on all interfaces where MACsec is enabled. The TSF is configured to permit only the following three EtherTypes:

- 1. EAPOL (0x888E):** The TSF permits EAPOL frames to facilitate the MKA protocol. These frames are used for peer discovery, authentication, and the distribution of SAKs.
- 2. MACsec (0x88E5):** The TSF permits MACsec frames, which represent the encrypted and integrity-protected data traffic between authorized peers.
- 3. MAC Control (0x8808):** the TSF permits MAC control frames to allow essential link-layer functions such as Ethernet Flow Control. These frames are processed by the MAC layer to ensure link stability.

The TSF evaluates the EtherType field of every incoming frame at the ingress port. If a frame's EtherType does not match one of the three permitted values listed above, the TSF identifies the frame as unauthorized and immediately discards it. Also, a filter in PFE traps the packets to RE with ether type 88-8E.

Verdict:

PASS.

5.1.2.10.2 FCS_MACSEC_EXT.1 AGD [TD0884]

If the TOE requires enabling or disabling specific EtherTypes to operate within the evaluated configuration, the guidance documentation must provide this information to the administrator.

Evaluator Findings:

The evaluator reviewed the AGD and verified that it documents the instructions for enabling or disabling specific EtherTypes.

The relevant information is found in the following section: **Configure MACsec**

Upon investigation, the evaluator found that the AGD section **Configure MACsec** states that:

The TOE only allows the following MACsec ethertypes by default and this requires no configuration: 88-8E, 88-E5, 88-08.

Verdict:

PASS.

5.1.2.11 FCS_MACSEC_EXT.2 MACSEC INTEGRITY AND CONFIDENTIALITY (MOD_MACSEC_V1.0)

5.1.2.11.1 FCS_MACSEC_EXT.2 TSS [TD1030]

The evaluator shall examine the TSS to verify that it describes the methods that the TOE implements to provide assurance of MACsec integrity. This should include any confidentiality offsets used, the use of an ICV (including the supported length), and ICV generation with the SAK, using the SCI (or SSCI, when XPN is used) as the most significant bits of the initialization vector (IV) and an IV appropriate to the supported ciphersuites (least significant 32 bits as the IV in a non-XPN context, and a 32-bit SSCI and 64-bit packet number as the IV when XPN is used).

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MACSEC_EXT.2** and verified that the TSS describes the methods that the TOE implements to provide assurance of MACsec integrity, including any confidentiality offsets used, the use of an ICV (including the supported length), and ICV generation with the SAK, using the SCI as the most significant bits of the initialization vector (IV) and the 32 least significant bits of the PN as the IV.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MACSEC_EXT.2**

Upon investigation, the evaluator found that the TSS states that:

Each MACsec Key Agreement protocol data unit (MKPDU) transmitted is integrity protected by a 128-bit Integrity Check value (ICV), generated by AES- CMAC using the Integrity Check value Key (ICK). The ICV Key (ICK) is derived from CAK (using AES_CMAC).

The Integrity Check Value (ICV) of MACsec protocol data units (MPDUs) is calculated using the SAK over the destination address, source address, SecTAG, and user data (after encryption, if applicable) and is encoded in the last eight to sixteen octets of the MPDU. The length of the ICV is between 8 and 16 octets, depending on the Cipher Suite. The 64 most significant bits of the 96-bit IV used in generating the ICV are the octets of the SCI, and the 32 least significant bits of the 96-bit IV are the octets of the PN.

MACsec allows IPv4/v6 and TCP/UDP headers to be unencrypted while the rest of the frame is encrypted. The offset value for MACsec protected frames are:

Offset 0 – Default; Encrypts the entire MPDU payload in the frame

Offset 30 – IPv4 & TCP/UDP headers are unencrypted and rest of the payload is encrypted

Offset 50 – IPv6 & TCP/UDP headers are unencrypted and rest of the payload is encrypted

Verdict:

PASS.

5.1.2.11.2 FCS_MACSEC_EXT.2 AGD

If any integrity verifications are configurable, such as any confidentiality offsets used or the mechanism used to derive an ICK, the evaluator shall verify that instructions for performing these functions are documented.

Evaluator Findings:

The evaluator reviewed the AGD and verified that the AGD documents the instructions for any configurable integrity verifications.

The relevant information is found in the following section: **Configuring MACsec on a Device Running Junos OS**

Upon investigation, the evaluator found that the AGD section **Configuring MACsec on a Device Running Junos OS** states that:

To configure MACsec on a device running Junos OS:

1. Configure the MACsec security mode as for the connectivity association.

[edit]

security-administrator@host:fips# set security macsec connectivity-association connectivity-association-name exclude-protocol protocol-name

security-administrator@host:fips# set security macsec connectivity-association connectivity-association-name include-sci

security-administrator@host:fips# set security macsec connectivity-association connectivity-association-name mka key-server-priority priority-number

security-administrator@host:fips# set security macsec connectivity-association connectivity-association-name mka transmit-interval interval

security-administrator@host:fips# set security macsec connectivity-association connectivity-association-name offset 30

NOTE: Based on your requirement you can configure the offset offset-number value at the set security macsec connectivity-association connectivity-association-name hierarchy level to 0 , 30 , or 50.

In addition, the guidance also notes that **The ICK is automatically derived from the CAK using AES_CMAC and this requires no separate configuration.**

Verdict:

PASS.

5.1.2.12 FCS_MACSEC_EXT.3 MACSEC RANDOMNESS (MOD_MACSEC_V1.0)

5.1.2.12.1 FCS_MACSEC_EXT.3 TSS

The evaluator shall examine the TSS to verify that it describes the method used to generate SAKs and nonces and that the strength of the CAK and the size of the CAK's key space are provided.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MACSEC_EXT.3** and verified that the TSS describes the method used to generate SAKs and nonces and provides the strength of the CAK and the size of the CAK's key space.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FCS_MACSEC_EXT.3**

Upon investigation, the evaluator found that the TSS states that:

SAK is generated using KDF function AES-CMAC-128 or AES-CMAC-256 based on the cipher suite configured using the following transform function:

SAK = KDF (Key, Label, KS-nonce | MI-value list | KN, SAKlength)

where,

- **Key= CAK**
- **Label= "IEEE8021 SAK"**
- **KS-nonce = a nonce of the same size as the required SAK, obtained from an RNG each time an SAK is generated.**
- **MI-valuelist = a concatenation of MI values from all live participants**
- **KN = four octets, the Key Number assigned by the Key Server as part of the KI**
- **SAKlength = two octets representing an integer value (128 for a 128 bit SAK, 256 for a 256 bit SAK) with the most significant octet first.**

The strength of the session protected by the CAK is determined by the strength of the underlying AES algorithm and its standard key length. Using AES-128 means a 128-bit CAK and a 2^{128} key space. Using AES-256 means a 256-bit CAK and a 2^{256} key space.

The KDF function for generation of SAK is in compliance with IEEE 802.1x-2020 (Section 9.8.1). For CAK length and strength, it is based on the cipher suite configured on the TOE (AES_GCM 128/256). RNG is used for nonces on the TOE.

Verdict:

PASS.

5.1.2.12.2 FCS_MACSEC_EXT.3 AGD

There are no guidance EAs for this component.

5.1.2.13 FCS_MACSEC_EXT.4 MACSEC KEY USAGE (MOD_MACSEC_V1.0)

5.1.2.13.1 FCS_MACSEC_EXT.4 TSS

The evaluator shall check the TSS to ensure that it describes how the SAK is wrapped prior to being distributed using the AES implementation specified in this PP-Module.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MACSEC_EXT.4** and verified that the TSS describes how the SAK is wrapped prior to being distributed using the AES implementation specified in this PP-Module.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MACSEC_EXT.4**

Upon investigation, the evaluator found that the TSS states that:

Before this key is sent to the peer, the TSF protects the SAK's by applying the AES Key Wrap function as specified in FCS_COP.1/MACSEC. The encryption key for this wrap operation is a KEK derived from the CAK (which is associated with a specific CKN).

Verdict:

PASS.

5.1.2.13.2 FCS_MACSEC_EXT.4 AGD [TD0803]

If the method of peer authentication is configurable, the evaluator shall verify that the guidance provides instructions on how to configure this.

Evaluator Findings:

The evaluator reviewed the AGD and verified that the AGD provides instructions for configuring the method of peer authentication.

The relevant information is found in the following section(s): **Configuring MACsec on a Device Running Junos OS**

Upon investigation, the evaluator found that the AGD section **Configuring MACsec on a Device Running Junos OS** states that:

2. Create the pre-shared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

[edit]

security-administrator@host:fips# prompt security macsec connectivity-association connectivity-association-name pre-shared-key cak

New cak (secret):

Retype new cak (secret):

security-administrator@host:fips# set security macsec connectivity-association connectivity-association-name pre-shared-key ckn hexadecimal-number

security-administrator@host:fips# set security macsec connectivity-association connectivity-association-name replay-protect replay-window-size number-of-packets

The evaluator shall also verify that the method of specifying a lifetime for CAKs is described.

Evaluator Findings:

The evaluator reviewed the AGD and verified that the AGD describes the method of specifying a lifetime for CAKs.

Upon investigation, the evaluator found that the AGD section **Configuring MACsec with keychain using Layer 3 Traffic** states that:

2. Create the secret password to use. It is a string of hexadecimal digits up to 64 characters long. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret-data is used as the Connectivity Association Key (CAK).

The TOE's administrator can configure up to 64 keys within a keychain. In addition to specifying the key material, the TOE's administrator can also configure start-time based on TOE's current time to precisely control the key's lifetime.

The keys can be configured with a start-time in YYYY-MM-DD.HH:MM format to define when the corresponding key becomes active.

When multiple keys are configured with sequential start times, new keys keep becoming active when their start time is reached, effectively configuring the lifetime for each key.

[edit]

security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551

```

security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 0 start-
time 2018-03-20.20:35
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 1 key-
name 2345678922334455667788992223334445556667778889992222333344445552
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 1 start-
time 2018-03-20.20:37
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 2 key-
name 2345678922334455667788992223334445556667778889992222333344445553
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 2 start-
time 2018-03-20.20:39
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 3 key-
name 2345678922334455667788992223334445556667778889992222333344445554
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 3 start-
time 2018-03-20.20:41
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 4 key-
name 2345678922334455667788992223334445556667778889992222333344445555
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 4 start-
time 2018-03-20.20:43

```

Use the prompt command to enter a secret key value. For example, the secret key value is 2345678922334455667788992223334123456789223344556677889922233341.

The TOE's administrator can configure up to 64 secret keys. For example, the TOE's administrator can refer the following secret keys:

[edit]

```

security-administrator@host:fips# prompt security authentication-key-chains key-chain macsec-kc1 key 0
secret

```

New cak (secret):

Retype new cak (secret):

```

security-administrator@host:fips# prompt security authentication-key-chains key-chain macsec-kc1 key 1
secret

```

New cak (secret):

Retype new cak (secret):

```

security-administrator@host:fips# prompt security authentication-key-chains key-chain macsec-kc1 key 2
secret

```

New cak (secret):

Retype new cak (secret):

```

security-administrator@host:fips# prompt security authentication-key-chains key-chain macsec-kc1 key 3
secret

```

New cak (secret):

Retype new cak (secret):

```

security-administrator@host:fips# prompt security authentication-key-chains key-chain macsec-kc1 key 4
secret

```

(conditional, the length of the CKN is configurable) The evaluator shall verify that the guidance describes how to set the CKN length to 1-32 octets.

Evaluator Findings:

The length of the CKN is not configurable; hence, this assurance activity is not applicable.

Verdict:

PASS.

5.1.2.14 FCS_MKA_EXT.1 MACSEC KEY AGREEMENT (MOD_MACSEC_V1.0)

5.1.2.14.1 FCS_MKA_EXT.1.1, FCS_MKA_EXT.1.2, FCS_MKA_EXT.1.3 TSS

The evaluator shall examine the TSS to verify that it describes the methods that the TOE implements to provide assurance of MKA integrity, including the use of an ICV and the ability to use a KDF to derive an ICK.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MKA_EXT.1** and verified that the TSS describes the methods that the TOE implements to provide assurance of MKA integrity, including the use of an ICV and the ability to use a KDF to derive an ICK.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MKA_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

**Each MACsec Key Agreement (MKA) protocol data unit (MKPDU) transmitted is integrity protected by an 128 bit Integrity Check value (ICV), generated by AES- CMAC using the Integrity Check value Key (ICK).
The ICK Key (ICK) is derived from CAK (using AES_CMAC).**

Verdict:

PASS.

5.1.2.14.2 FCS_MKA_EXT.1.1, FCS_MKA_EXT.1.2, FCS_MKA_EXT.1.3 AGD

There are no guidance EAs for this element.

5.1.2.14.3 FCS_MKA_EXT.1.4 TSS

There are no TSS EAs for this element.

5.1.2.14.4 FCS_MKA_EXT.1.4 AGD

There are no guidance EAs for this element.

5.1.2.14.5 FCS_MKA_EXT.1.5, FCS_MKA_EXT.1.6, FCS_MKA_EXT.1.7 TSS

The evaluator shall verify that the TSS describes the TOE's compliance with IEEE 802.1X-2010 and 802.1Xbx2014 for MKA, including the values for MKA and Hello timeout limits and support for data delay protection.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MKA_EXT.1**, and verified that the TSS describes the TOE's compliance with IEEE 802.1X-2020 and 802.1Xbx2014 for MKA, including the values for MKA and Hello timeout limits and support for data delay protection.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MKA_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE is in compliance with IEEE 802.1X-2020 and 802.1Xbx2014 for MKA, including the values for MKA and Hello timeout limits and support for data delay protection.
The TOE has a mka-hello-timeout of 2 seconds. The TOE has a MKA lifetime of 6 seconds. The TOE has a sak-rekey-interval between 60 and 86400 seconds.

The evaluator shall also verify that the TSS describes the ability of the PAE of the TOE to establish unique CAs with individual peers and group CAs using a group CAK such that a new group SAK is distributed every time the group's membership changes.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MKA_EXT.1**, and verified that the TSS describes the ability of the PAE of the TOE to establish unique CAs with individual peers and group CAs using a group CAK such that a new group SAK is distributed every time the group's membership changes.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MKA_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

Group CAK is not supported. Pairwise CAKs which are PSKs are supported by the TOE. The TSF actively manages membership changes within the Pairwise CA. The TSF monitors the "Live" status of its peer using two primary timers: MKA Hello and MKA Lifetime. If the TSF detects a membership change the elected Key Server automatically distributes a fresh SAK. This ensures that the data-plane encryption is refreshed immediately upon any change to the secure session's participants.

The evaluator shall also verify that the TSS describes the invalid MKPDUs that are discarded automatically by the TSF in a manner that is consistent with the SFR, and that valid MKPDUs are decoded in a manner consistent with IEEE 802.1X-2020 section 11.11.4.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_MKA_EXT.1**, and verified that the TSS describes the invalid MKPDUs that are discarded automatically by the TSF in a manner that is consistent with the SFR, and that valid MKPDUs are decoded in a manner consistent with IEEE 802.1X-2020 section 11.11.4.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_MKA_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE is in compliance with IEEE 802.1X-2020 and 802.1Xbx2014 for MKA, including the values for MKA and Hello timeout limits and support for data delay protection.

For MKPDU's to be valid, they must meet the following criteria:

- **Must be from a recognized peer**
- **Must match current CAK/CKN**
- **Must pass integrity check (ICV)**
- **Must not be replayed (sequence number check)**

Invalid MKPDU's can be any of the following:

- **Come from unknown/untrusted peers**
- **Have bad syntax**
- **Have expired lifetimes or fail ICV check**
- **Are duplicate or replayed packets**

Verdict:

PASS.

5.1.2.14.6 FCS_MKA_EXT.1.5, FCS_MKA_EXT.1.6, FCS_MKA_EXT.1.7 AGD

The evaluator shall verify that the guidance documentation provides instructions on how to configure the TOE to act as the key server in an environment with multiple MACsec-capable devices.

Evaluator Findings:

The evaluator reviewed the AGD and verified that the AGD provides instructions on how to configure the TOE to act as the key server in an environment with multiple MACsec-capable devices.

The relevant information is found in the following section: **Configuring Static MACsec for Layer 2 Traffic**

Upon investigation, the evaluator found that the section **Configuring Static MACsec for Layer 2 Traffic** states that, to configure the TOE to operate as the key server, the administrator must set the MKA key server priority, as shown below:

[edit]

security-administrator@host:fips# set security macsec connectivity-association CA1 mka key-server-priority 1

The AGD further notes that **The configurable range for the key server priority values is 0-255 with 0 meaning highest priority. Also, the TOE must be configured with the lowest priority number to configure it to act as a key server.**

Verdict:

PASS.

5.1.2.15 FCS_NTP_EXT.1 NTP PROTOCOL (CPP_ND_V3.0E)

5.1.2.15.1 FCS_NTP_EXT.1.1 TSS

The evaluator shall examine the TSS to ensure it identifies the version of NTP supported, how it is implemented and what approach the TOE uses to ensure the timestamp it receives from an NTP timeserver (or NTP peer) is from an authenticated source and the integrity of the time has been maintained.

Evaluator Findings:

The evaluator examined the TSS row **FCS_NTP_EXT.1** to ensure it identifies the version of NTP supported, how it is implemented and what approach the TOE uses to ensure the timestamp it receives from an NTP timeserver (or NTP peer) is from an authenticated source and the integrity of the time has been maintained.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_NTP_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE uses NTP v4 and uses SHA1 and SHA256 for authenticating time stamps received.

NTPv4 is implemented on the TOE using the libntp and libntptpts libraries.

The TOE uses the combination of the configured IP address alongside the SHA1/SHA256 hash algorithm to properly authenticate the NTP server. If either of these values are not correct, the NTP server will not synchronize with the TOE until the proper authentication configuration is set.

The TOE must support at least one of the methods or may use multiple methods, as specified in the SFR element 1.2. The evaluator shall ensure that each method selected in the ST is described in the TSS, including the version of NTP supported in element 1.1, the message digest algorithms used to verify the authenticity of the timestamp and/or the protocols used to ensure integrity of the timestamp.

Evaluator Findings:

The TOE must support at least one of the methods or may use multiple methods, as specified in the SFR element 1.2. The evaluator shall ensure that each method selected in the ST is described in the TSS row **FCS_NTP_EXT.1**, including the version of NTP supported in element 1.1, the message digest algorithms used to verify the authenticity of the timestamp and/or the protocols used to ensure integrity of the timestamp.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_NTP_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE uses NTP v4 and uses SHA1 and SHA256 for authenticating time stamps received. The NTP sources are defined by the Security Administrator. Up to three sources can be configured. NTPv4 is implemented on the TOE using the libntp and libntptpts libraries. Time updates from a broadcast and multicast address are not supported by the TOE. The TOE uses the combination of the configured IP address alongside the SHA1/SHA256 hash algorithm to properly authenticate the NTP server. If either of these values are not correct, the NTP server will not synchronize with the TOE until the proper authentication configuration is set.

These descriptions align with the selections in elements FCS_NTP_EXT.1.1 and FCS_NTP_EXT.1.2.

Verdict:

PASS.

5.1.2.15.2 FCS_NTP_EXT.1.2, FCS_NTP_EXT.1.3, AND FCS_NTP_EXT.1.4 TSS

None.

5.1.2.15.3 FCS_NTP_EXT.1.1 AGD

The evaluator shall examine the AGD to ensure it provides the Security Administrator instructions as how to configure the version of NTP supported, how to configure multiple NTP servers for the TOE's time source and how to configure the TOE to use the method(s) that are selected in the ST.

Evaluator Findings:

The evaluator examined the AGD to ensure it provides the Security Administrator instructions as how to configure the version of NTP supported, how to configure multiple NTP servers for the TOE's time source and how to configure the TOE to use the method(s) that are selected in the ST.

The relevant information is found in the following section: **Configuring Network Time Protocol**

Upon investigation, the evaluator found that the AGD section **Configuring Network Time Protocol** states that:

The device can be configured to sync with a Network Time Protocol (NTP) server. It supports time updates using NTP version 4. The device authenticates updates using an administrator-configured SHA-1 or SHA-256 symmetric key. The device rejects broadcast and multicast time updates by default without requiring any configuration. It does not place a limit on the number of NTP time sources that can be configured. To configure the device in client mode, include the server statement and other statements at the [edit system ntp] hierarchy level:

[edit system ntp]

```
security-administrator@hostname:fips# server address key key-number version 4 prefer
security-administrator@hostname:fips# authentication-key key-number type sha1|sha256 value <key value>
security-administrator@hostname:fips# trusted-key trusted <key-number>
```

The above commands specify the address of the system acting as the time server. One specifies an address, not a hostname. The same command can be used to configure multiple NTP servers.

It also configures the authentication key sent in all messages sent to the time server, using the key option.

The key corresponds to the key number specified in the authentication-key statement. By default, the device sends NTP version 4 packets to the time server. If more than one time server is configured, one server can be marked as preferred by including the prefer option.

Verdict:

PASS.

5.1.2.15.4 FCS_NTP_EXT.1.2 AGD

For each of the secondary selections made in the ST, the evaluator shall examine the AGD to ensure it instructs the Security Administrator how to configure the TOE to use the algorithms that support the authenticity of the timestamp and/or how to configure the TOE to use the protocols that ensure the integrity of the timestamp.

Evaluator Findings:

The evaluator examined the AGD and ensured that, for each of the secondary selections made in the ST, it instructs the Security Administrator how to configure the TOE to use the algorithms that support the authenticity of the timestamp and/or how to configure the TOE to use the protocols that ensure the integrity of the timestamp.

The relevant information is found in the following section: **Configuring Network Time Protocol**

Upon investigation, the evaluator found that the AGD section **Configuring Network Time Protocol** states that:

The device authenticates updates using an administrator-configured SHA-1 or SHA-256 symmetric key.

It also configures the authentication key sent in all messages sent to the time server, using the key option.

The key corresponds to the key number specified in the authentication-key statement.

[edit system ntp]

```
security-administrator@hostname:fips# server address key key-number version 4 prefer
```

```
security-administrator@hostname:fips# authentication-key key-number type sha1|sha256 value key value
security-administrator@hostname:fips# trusted-key trusted key-number
```

For common criteria compliance, use trusted authentication using SHA1 or SHA256 as the message digest algorithm(s) to make sure that the NTP peer is trusted.

Assurance Activity Note: Each primary selection in the SFR contains selections that specify a cryptographic algorithm or cryptographic protocol. For each of these secondary selections made in the ST, the evaluator shall examine the guidance documentation to ensure that the documentation instructs the Security Administrator how to configure the TOE to use the chosen option(s).

Evaluator Findings:

Each primary selection in the SFR contains selections that specify a cryptographic algorithm or cryptographic protocol. For each of these secondary selections made in the ST, the evaluator examined the guidance documentation and ensured that the documentation instructs the Security Administrator how to configure the TOE to use the chosen option(s).

The relevant information is found in the following section: **Configuring Network Time Protocol**

Upon investigation, the evaluator found that the AGD section **Configuring Network Time Protocol** states that:
The device authenticates updates using an administrator-configured SHA-1 or SHA-256 symmetric key. To configure the device in client mode, include the server statement and other statements at the [edit system ntp] hierarchy level:
[edit system ntp]

```
security-administrator@hostname:fips# server address key key-number version 4 prefer
security-administrator@hostname:fips# authentication-key key-number type sha1|sha256 value <key value>
security-administrator@hostname:fips# trusted-key trusted <key-number>
```

It also configures the authentication key sent in all messages sent to the time server, using the key option.

For common criteria compliance, use trusted authentication using SHA1 or SHA256 as the message digest algorithm(s) to make sure that the NTP peer is trusted.

Verdict:

PASS.

5.1.2.15.5 FCS_NTP_EXT.1.3 AGD

The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the TOE to not accept broadcast and multicast NTP packets that would result in the timestamp being updated.

Evaluator Findings:

The evaluator examined the AGD to ensure it provides the Security Administrator instructions as how to configure the TOE to not accept broadcast and multicast NTP packets that would result in the timestamp being updated.

The relevant information is found in the following section: **Configuring Network Time Protocol**

Upon investigation, the evaluator found that the AGD section **Configuring Network Time Protocol** states that:
The device rejects broadcast and multicast time updates by default without requiring any configuration.

Verdict:

PASS.

5.1.2.15.6 FCS_NTP_EXT.1.4 AGD

None.

5.1.2.16 FCS_RBG_EXT.1 RANDOM BIT GENERATION (CPP_ND_V3.0E)

5.1.2.16.1 FCS_RBG_EXT.1 TSS

The evaluator shall examine the TSS to determine that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min- entropy contained in the combined seed value.

Evaluator Findings:

The evaluator examined the TSS row **FCS_RBG_EXT.1** and determined that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min- entropy contained in the combined seed value.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_RBG_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

All random number generation by the TOE is performed in accordance with NIST Special Publications SP 800-90A, SP 800-90B and ISO/IEC 18031:2011 using HMAC_DRBG (using HMAC-SHA-512) implemented in the OpenSSL library and kernel library. The HMAC_DRBG is seeded using a software-based entropy source (bits 2-9 of the timestamp of the software interrupts associated with the clock0 (RANDOM_SWI_CLOCK0) as the primary entropy source) implemented in accordance with NIST Special Publication SP 800-90B producing 448 bits of calculated min-entropy.

The appliance is to be operated with FIPS mode enabled.

Verdict:

PASS.

5.1.2.16.2 FCS_RBG_EXT.1 AGD

The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality.

Evaluator Findings:

The evaluator confirmed that the guidance documentation section **Supported Cryptographic Algorithms** contains appropriate instructions for configuring the RNG functionality.

The relevant information is found in the following section: **Supported Cryptographic Algorithms**

Upon investigation, the evaluator found that the AGD section **Supported Cryptographic Algorithms** states that:

Deterministic Random Bit Generator. The DRBG implemented is not configurable. It is always the HMAC DRBG using SHA-512. The DRBG is seeded with 448 bits of entropy, which exceed the 384 required by the FIPS standard. The DRBG is seeded with multiple entropy sources generated within the Junos OS kernel. The designated primary entropy source for FIPS purposes that determines when the DRBG is considered fully seeded and operational is bits 2-10 of the timestamp associated with kernel software interrupts.

NOTE: DRBG requires no configuration and runs automatically at boot.

Verdict:

PASS.

5.1.2.17 FCS_SSH_EXT.1 SSH PROTOCOL (PKG_SSH_V1.0)

5.1.2.17.1 FCS_SSH_EXT.1.1 TSS

The evaluator shall ensure that the selections indicated in the ST are consistent with selections in this and subsequent components. Otherwise, this SFR is evaluated by activities for other SFRs.

Evaluator Findings:

This SFR is evaluated by activities for subsequent SFRs.

Verdict:

PASS.

5.1.2.17.2 FCS_SSH_EXT.1.1 AGD

There are no guidance evaluation activities for this component. This SFR is evaluated by activities for other SFRs.

5.1.2.17.3 FCS_SSH_EXT.1.2 TSS

The evaluator shall check to ensure that the authentication methods listed in the TSS are identical to those listed in this SFR component; and, ensure if password-based authentication methods have been selected in the ST then these are also described; and, ensure that if keyboard-interactive is selected, it describes the multifactor authentication mechanisms provided by the TOE.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the authentication methods listed in the TSS are identical to those listed in this SFR component.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE can be configured to accept public-key based authentication, keyboard-interactive or password-

based authentication.

In accordance with RFC 4252, the TOE supports password-based authentication. The TSF prompts the user for a case-sensitive password. The input is masked during entry and compared against the stored credentials using a secure hash-based comparison.

The TOE supports public key authentication as defined in RFC 4252. The TSF verifies the digital signature provided by the SSH client against the authorized public keys stored on the TOE. The following public key algorithms are supported:

- RSA: ssh-rsa (RFC 4253), rsa-sha2-256, and rsa-sha2-512 (RFC 8332).
- ECDSA: ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521 (RFC 5656).

The TOE supports the Keyboard-interactive authentication method by default and requires no configuration to be enabled. Since AAA server that would be required to facilitate multifactor authentication for this method is not in scope, this method is functionally similar to password-based authentication, and hence only requires a password to be configured for the user. The TOE does not require multiple authentication mechanisms for users.

Verdict:

PASS.

5.1.2.17.4 FCS_SSH_EXT.1.2 AGD

The evaluator shall check the guidance documentation to ensure the configuration options, if any, for authentication mechanisms provided by the TOE are described.

Evaluator Findings:

The evaluator verified that the AGD section '**Configure a Network Device Collaborative Protection Profile Authorized Administrator**' instructs the administrator on the configuration options for authentication mechanisms provided by the TOE.

The relevant information is found in the following sections: **Configure a Network Device Collaborative Protection Profile Authorized Administrator** and **Supported Cryptographic Algorithms**

Upon investigation, the evaluator found that the AGD section **Configure a Network Device Collaborative Protection Profile Authorized Administrator** gives the steps for authentication configuration. The AGD states that:

To configure an authorized administrator:

4. Define the NDcPPv3.0e user authorized administrator using the command below.

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication encrypted-password
```

or

[edit]

```
security-administrator@host:fips# set system login user NDcPPv2-user class security-admin authentication plain-text-password
```

To configure public key authentication:

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh ecdsa <ecdsa-public-key>
```

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh rsa <rsa-public-key>
```

NOTE: ssh-ed25519 is not supported in FIPS mode although it is shown as a configurable option.

To delete a configured login credential, use the following command:

[edit]

```
security-administrator@host:fips# delete system login user NDcPPv3-user class security-admin authentication
```

The set commands used to configure the credentials as shown above can be repeated to overwrite the currently configured credentials.

The Keyboard-Interactive Based authentication for SSH is supported by default and needs no additional configuration apart from a password being configured for the user. Providing multifactor authentication mechanism would require the use of an external AAA server, which is outside the CC scope, as a result of which the keyboard-interactive authentication method works similarly to the password-based method in the evaluated configuration.

Table 1 in the AGD section **Supported Cryptographic Algorithms** also lists the public key algorithms supported by the TOE for client (user) authentication:

Client (user):

- ECDSA P-256
- ECDSA P-384
- ECDSA P-521
- SSH-RSA

- RSA-SHA2-256
- RSA-SHA2-512

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.2.17.5 FCS_SSH_EXT.1.3 TSS

The evaluator shall check that the TSS describes how “large packets” are detected and handled.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the TOE describes how “large packets” are detected and handled.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

“Large packets” are defined as the packets that are greater than 256K bytes in an SSH transport connection. SSH packets greater than 256k bytes, when detected by the TOE are dropped and the connection is terminated.

Verdict:

PASS.

5.1.2.17.6 FCS_SSH_EXT.1.3 AGD

None.

5.1.2.17.7 FCS_SSH_EXT.1.4 TSS

The evaluator will check the description of the implementation of SSH in the TSS to ensure the encryption algorithms supported are specified.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the encryption algorithms supported by the implementation of SSH are specified.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports the following algorithms for its SSH implementations:

- AES-CBC, AES-GCM@openssh.com and AES-CTR encryption algorithms with key sizes of 128 bits and 256 bits. The TOE does not support the “none” cipher.

The evaluator will check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the encryption algorithms listed in the TSS are identical to those selected in FCS_SSH_EXT.1.4.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports the following algorithms for its SSH implementations:

- AES-CBC, AES-GCM@openssh.com and AES-CTR encryption algorithms with key sizes of 128 bits and 256 bits. The TOE does not support the “none” cipher which is consistent with the selections indicated in the ST.

Therefore, the evaluator concludes that the TSS satisfies this objective.

Verdict:

PASS.

5.1.2.17.8 FCS_SSH_EXT.1.4 AGD

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Evaluator Findings:

The evaluator verified that the AGD section ‘**Configure SSH on the Evaluated Configuration**’ instructs the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

The relevant information is found in the following section: **Configure SSH on the Evaluated Configuration**

Upon investigation, the evaluator found that the claimed AGD section **Configure SSH on the Evaluated Configuration** states that:

4. Specify the ciphers allowed for protocol version 2.

[edit]

```
security-administrator@host:fips# set system services ssh ciphers aes128-cbc
security-administrator@host:fips# set system services ssh ciphers aes256-cbc
security-administrator@host:fips# set system services ssh ciphers aes128-ctr
security-administrator@host:fips# set system services ssh ciphers aes256-ctr
security-administrator@host:fips# set system services ssh ciphers aes128-gcm@openssh.com
security-administrator@host:fips# set system services ssh ciphers aes256-gcm@openssh.com
```

Supported SSH ciphers algorithm:

aes128-cbc	128-bit AES with Cipher Block Chaining
aes128-ctr	128-bit AES with Counter Mode
aes256-cbc	256-bit AES with Cipher Block Chaining
aes256-ctr	256-bit AES with Counter Mode
aes128-gcm@openssh.com	128-bit AES with Galois/Counter Mode
aes256-gcm@openssh.com	256-bit AES with Galois/Counter Mode

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.2.17.9 FCS_SSH_EXT.1.5 TSS

The evaluator will check the description of the implementation of SSH in the TSS to ensure the hashing algorithms supported are specified.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the hashing algorithms supported by the implementation of SSH are specified.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports the following algorithms for its SSH implementations:

- **HMAC-SHA-256, HMAC-SHA-512 and AES_GCM (implicit) for keyed-hashing.**

The evaluator will check the TSS to ensure that the hashing algorithms specified are identical to those listed for this component.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the encryption algorithms listed in the TSS are identical to those listed in FCS_SSH_EXT.1.5.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports the following algorithms for its SSH implementations:

- **HMAC-SHA-256, HMAC-SHA-512 and AES_GCM (implicit) for keyed-hashing** which is consistent with the selections indicated in the ST.

Therefore, the evaluator concludes that the TSS satisfies this objective.

Verdict:

PASS.

5.1.2.17.10 FCS_SSH_EXT.1.5 AGD

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Evaluator Findings:

The evaluator verified that the AGD section '**Configure SSH on the Evaluated Configuration**' instructs the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

The relevant information is found in the following section: **Configure SSH on the Evaluated Configuration**

Upon investigation, the evaluator found that the claimed AGD section states that:

3. Specify all the permissible message authentication code algorithms for SSHv2
[edit]

security-administrator@host:fips# set system services ssh macs hmac-sha2-256

security-administrator@host:fips# set system services ssh macs hmac-sha2-512

Supported MACs algorithm:

hmac-sha2-256	Hash-based MAC using Secure Hash Algorithm (SHA2)
hmac-sha2-512	Hash-based MAC using Secure Hash Algorithm (SHA2)

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.2.17.11 FCS_SSH_EXT.1.6 TSS

The evaluator will check the description of the implementation of SSH in the TSS to ensure the shared secret establishment algorithms supported are specified.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the shared secret establishment algorithms supported by the implementation of SSH are specified.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

Key exchange is performed only using one of the supported key exchange algorithms, which are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 (all specified in RFC 5656).

The evaluator will check the TSS to ensure that the shared secret establishment algorithms specified are identical to those listed for this component.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the encryption algorithms listed in the TSS are identical to those listed in FCS_SSH_EXT.1.6.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

Key exchange is performed only using one of the supported key exchange algorithms, which are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 (all specified in RFC 5656) which is consistent with the selections indicated in the ST.

Therefore, the evaluator concludes that the TSS satisfies this objective.

Verdict:

PASS.

5.1.2.17.12 FCS_SSH_EXT.1.6 AGD

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Evaluator Findings:

The evaluator verified that the AGD section '**Configure SSH on the Evaluated Configuration**' instructs the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

The relevant information is found in the following section: **Configure SSH on the Evaluated Configuration**

Upon investigation, the evaluator found that the claimed AGD section states that:

2. Specify the SSH key-exchange for Elliptic Curve Diffie-Hellman keys for the system services.

[edit]

security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256

security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384

security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521

Supported SSH key-exchange algorithm:

ecdh-sha2-nistp256 The EC Diffie-Hellman on nistp256 with SHA2-256

ecdh-sha2-nistp384 The EC Diffie-Hellman on nistp384 with SHA2-384

ecdh-sha2-nistp521 The EC Diffie-Hellman on nistp521 with SHA2-512

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.2.17.13 FCS_SSH_EXT.1.7 TSS

The evaluator will check the description of the implementation of SSH in the TSS to ensure the KDFs supported are specified.

The evaluator will check the TSS to ensure that the KDFs specified are identical to those listed for this component.

Evaluator Findings:

The evaluator reviewed the TSS row **FCS_SSH_EXT.1** and verified that the KDFs supported by the implementation of SSH are specified.

The relevant information is found in the following section: **TOE Summary Specification** row **FCS_SSH_EXT.1**.

Upon investigation, the evaluator found that the TSS states that:

Key exchange is performed only using one of the supported key exchange algorithms, which are ordered as follows: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 (all specified in RFC 5656).

The TSF uses the SSH KDF, as specified in RFC 5656 (Section 4), to derive the session keys from the shared secret value (K) established during the Key Exchange (KEX) process.

These KDFs specified are consistent with the selections listed in FCS_SSH_EXT.1.7.

Verdict:

PASS.

5.1.2.17.14 FCS_SSH_EXT.1.7 AGD

None.

5.1.2.17.15 FCS_SSH_EXT.1.8 TSS

The evaluator shall check the TSS to ensure that if the TOE enforces connection rekey or termination limits lower than the maximum values that these lower limits are identified.

In cases where hardware limitation will prevent reaching data transfer threshold in less than one hour, the evaluator shall check the TSS to ensure it contains:

- a. An argument describing this hardware-based limitation and
- b. Identification of the hardware components that form the basis of such argument.

For example, if specific Ethernet Controller or Wi-Fi radio chip is the root cause of such limitation, these subsystems shall be identified.

Evaluator Findings:
The evaluator reviewed the TSS row FCS_SSH_EXT.1 and verified that the lower limits for connection rekey or termination enforced by the TOE are identified. The relevant information is found in the following section: TOE Summary Specification row FCS_SSH_EXT.1 Upon investigation, the evaluator found that the TSS states that: SSH rekeying is triggered by exceeding either time or data thresholds. Administrators can configure the specific thresholds that initiate rekeying. The time limit for rekeying can be set from 1 to 1440 minutes (24 hours). The data limit for rekeying can be set from 51200 to 4294967295 bytes (0.5MB to 4GB). There are no hardware limitations that prevent reaching the data transfer threshold in less than one hour; hence, this part of the assurance activity is not applicable.

Verdict:

PASS.

5.1.2.17.16 FCS_SSH_EXT.1.8 AGD

The evaluator shall check the guidance documentation to ensure that if the connection rekey or termination limits are configurable, it contains instructions to the administrator on how to configure the relevant connection rekey or termination limits for the TOE.

Evaluator Findings:

The evaluator verified that the AGD section '**Limit the Number of User Login Attempts for SSH Sessions**' instructs the administrator on how to configure the relevant connection rekey or termination limits for the TOE, if these limits are configurable.

The relevant information is found in the following section: **Limit the Number of User Login Attempts for SSH Sessions**

Upon investigation, the evaluator found that the AGD claimed section **Limit the Number of User Login Attempts for SSH Sessions** states that:

The SSH2 protocol provides secure terminal sessions utilizing the secure encryption. The SSH2 protocol enforces running the key-exchange phase and changing the encryption and integrity keys for the session. Key exchange is done periodically, after specified seconds or after specified bytes of data have passed over the connection. The TSF ensures that within the SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of the transmitted data. When either of the thresholds are reached, a rekey must be performed.

[edit system]

security-administrator@host:fips# set services ssh rekey time-limit number
Time limit before renegotiating session keys is 1 through 1440 minutes.

[edit system]

security-administrator@host:fips# set services ssh rekey data-limit number
Data limit before renegotiating session keys is 51200 through 4294967295 byte.

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.2.18 FCS_SSHS_EXT.1 SSH PROTOCOL - SERVER (PKG_SSH_V1.0)**5.1.2.18.1 FCS_SSHS_EXT.1 TSS**

No activities.

5.1.2.18.2 FCS_SSHS_EXT.1 AGD

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Evaluator Findings:

The evaluator verified that the AGD section '**Configure SSH on the Evaluated Configuration**' instructs the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

The relevant information is found in the following section(s): **Configure SSH on the Evaluated Configuration**

Upon investigation, the evaluator found that the AGD section **Configure SSH on the Evaluated Configuration**

states that:

To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms for the system services.

[edit]

security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256

security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384

security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521

security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa

NOTE: Although the last hostkey algorithm option mentioned above only mentions RSA, it covers all the claimed RSA algorithms, that is SSH-RSA, RSA-SHA2-256, and RSA-SHA2-512

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.3 IDENTIFICATION AND AUTHENTICATION (FIA)

5.1.3.1 FIA_AFL.1 AUTHENTICATION FAILURE MANAGEMENT (CPP_ND_V3.0E)

5.1.3.1.1 FIA_AFL.1 TSS

The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

Evaluator Findings:

The evaluator shall examine the TSS row **FIA_AFL.1** to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

The relevant information is found in the following section: **TOE Summary Specification** row **FIA_AFL.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports SSH for remote administrative actions. Each unsuccessful authentication attempt is detected and tracked by the TOE. SSH login retries are enabled after the first failed authentication attempt for a username. The maximum number of authentication attempts permitted before SSH connection termination and lockout enforcement is configurable via tries-before-disconnect parameter. This parameter supports a range of 1 to 10, as defined in the CLI. Exceeding this threshold results in user account lockout, preventing further SSH login attempts until a TOE's administrator unlocks it or the administrator-configured "lockout-period" (1 to 43,200 minutes) expires.

Even when an account is locked for remote access to the TOE, an administrator is always able to login locally through the serial console.

The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

Evaluator Findings:

The evaluator shall examine the TSS row **FIA_AFL.1** to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

The relevant information is found in the following section: **TOE Summary Specification** row **FIA_AFL.1**

Upon investigation, the evaluator found that the TSS states that:

Even when an account is locked for remote access to the TOE, an administrator is always able to login locally through the serial console.

Verdict:

PASS.

5.1.3.1.2 FIA_AFL.1 AGD

The evaluator shall examine the guidance documentation to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period (if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each “action” specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

Evaluator Findings:

The evaluator examined the guidance documentation section ‘**Limiting the Number of User Login Attempts for SSH Sessions**’ to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each “action” specified. SSH is the only claimed secure protocol for TOE administration.

The relevant information is found in the following section: **Limit the Number of User Login Attempts for SSH Sessions**

Upon investigation, the evaluator found that the claimed AGD section states that:

The TOE’s administrator can configure the device to limit the number of attempts to enter a password while logging through SSH. Using the following command,

[edit system login]

security-administrator@host:fips# set retry-options tries-before-disconnect number

Here, tries-before-disconnect is the number of times a user can attempt to enter a password when logging in. The connection closes if a user fails to log in after the number specified. The range is from 1 through 10, and the default value is 10.

The lockout-period must be greater than zero. The range at which the TOE’s administrator can configure the lockout-period is one through 43,200 minutes.

[edit system login]

security-administrator@host:fips# set retry-options lockout-period number

When the interval expires, authentication attempts are again accepted.

To unlock a locked user account, use the following command from operational mode:
security-administrator@host:fps> clear system login lockout user username
An administrator uses the above command to manually unlock a user before their lockout period expires.

The evaluator shall examine the guidance documentation to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

Evaluator Findings:

The evaluator examined the guidance documentation section '**Limiting the Number of User Login Attempts for SSH Sessions**' to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

The relevant information is found in the following section: **Limit the Number of User Login Attempts for SSH Sessions**

Upon investigation, the evaluator found that the claimed AGD section states that:

The local administrator's access will be maintained even if the remote administration is made permanently or temporarily unavailable due to the multiple failed login attempts. The console login for local administration will be available to the users during the lockout period.

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.3.2 FIA_PMG_EXT.1 PASSWORD MANAGEMENT (CPP_ND_V3.0E)

5.1.3.2.1 FIA_PMG_EXT.1 TSS

The evaluator shall check that the TSS:

- lists the supported special character(s) for the composition of administrator passwords.
- to ensure that the minimum_password_length parameter is configurable by a Security Administrator.
- lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

Evaluator Findings:

The evaluator checked that the TSS row **FIA_PMG_EXT.1**:

- lists the supported special character(s) for the composition of administrator passwords.
- to ensure that the minimum_password_length parameter is configurable by a Security Administrator.
- lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

The relevant information is found in the following section: **TOE Summary Specification** row **FIA_PMG_EXT.1**

The TOE supports password management capabilities for administrative passwords. Passwords may be composed of any combination of upper and lower-case letters, numbers, specific special characters which include: ["!", "@", "#", "\$", "%", "^", "&", "*", "(", ")", "~", ", ", ".", "/", ": ", ";", "_", "+", "-", "=", "{", "}", "[", "]", "|", "<", ">", " ", "'", "\ ", "`"]. The minimum password length is configurable by the Administrator with a range of 06 to 20 characters and a default value of 10.

PASS.

The evaluator shall examine the guidance documentation to determine that it:

- a. identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and
- b. provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

The evaluator examined the guidance documentation sections ‘**Configure Password Rules for an Authorized Administrator**’ and ‘**Table 5: Guidelines for Strong Passwords and Characteristics of Weak Passwords**’ to determine that it:

- a. identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and
- b. provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

Upon investigation, the evaluator found that the claimed AGD section states that:

Ensure that the device is in FIPS mode before the TOE's administrator configure the security administrator or any users. All passwords established for users by the security administrator must conform to Junos OS in FIPS mode requirements in Table 4 on page 30 (of the AGD). Attempts to configure passwords that do not conform to the following specifications result in an error.

Requirements	Details
Length	Passwords must contain at least 15 characters.
Character set	Passwords must contain at least three of the following five character sets: • Uppercase letters • Lowercase letters • Digits • Punctuation marks • Keyboard characters are not included in the other four sets. These characters are the percent sign (%) and the ampersand (&). NOTE: Avoid control characters in passwords.
Password encryption	To change the default encryption method (SHA512), include the format statement at the [edit system login password] hierarchy level.

Authentication data for fixed password authentication is a case-sensitive, value containing a combination of

	phrases from sports, sayings, movies, or television shows. Permutations of any words or phrases mentioned above. For example, a dictionary word with letters replaced with digits (r00t) or with digits added at the end of the word.
The TOE's administrator must not disclose the passwords to anyone To protect user credentials, the login system masks characters during password input, obscuring feedback effectively. This is the default behavior and requires no additional configuration.	Any machine-generated password. Algorithms reduce the search space of password-guessing programs, and the TOE's administrator must not use machine-generated password.

Verdict:

PASS.

5.1.3.3 FIA_PSK_EXT.1 PRE-SHARED KEY COMPOSITION (MOD_MACSEC_V1.0)

5.1.3.3.1 FIA_PSK_EXT.1 TSS

The evaluator shall examine the TSS to ensure it describes the process by which the bit-based PSKs are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

Evaluator Findings:

The TOE does not support generation of bit-based pre-shared keys; therefore, this requirement is not applicable.

Verdict:

PASS.

5.1.3.3.2 FIA_PSK_EXT.1 AGD

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong PSKs, and (if the selection indicates keys of various lengths can be entered) that it provides information on the range of lengths supported.

Evaluator Findings:

The evaluator reviewed the AGD sections '**Media Access Control Security (MACsec) in FIPS Mode Overview**' and '**Configuring MACsec with keychain using Layer 3 Traffic**' and verified that the AGD provides guidance to administrators on the composition of strong PSKs.

The relevant information is found in the following section: **Media Access Control Security (MACsec) in FIPS Mode Overview** and **Configuring MACsec with keychain using Layer 3 Traffic**

Upon investigation, the evaluator found that the claimed AGD section **Media Access Control Security (MACsec) in FIPS Mode Overview** states that:

A pre-shared key is exchanged between directly-connected links to establish a MACsec-secure link. The pre-shared-key includes the CKN and the CAK. The CKN is a 64-digit hexadecimal number and the CAK is a 32-digit hexadecimal number. The CKN and CAK must match on both ends of a link to create a MACsec-secured link.

When a 128-bit cipher suite is configured, the CAK and CKN shall each be specified as a hexadecimal number with a minimum length of 1 digit and a maximum length of 32 digits. When a 256-bit cipher suite is configured, the CAK and CKN shall each be specified as a hexadecimal number with a minimum length of 2 digits and a maximum length of 64 digits.

NOTE: To maximize security, the TOE's administrator should configure all 64 digits of a CKN and all 32 digits of a CAK. If you do not configure all 64 digits of a CKN or all 32 digits of a CAK, the system auto-configures all the remaining digits to 0. However, you will receive a warning message when you commit the configuration.

Additionally, the AGD section **Configuring MACsec with keychain using Layer 3 Traffic** states that:

2. Create the secret password to use. It is a string of hexadecimal digits up to 64 characters long. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret-data is used as the Connectivity Association Key (CAK).

The TOE's administrator can configure up to 64 keys within a keychain. For example, we can refer the following keys:

[edit]

```
security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
```

Keys of various lengths can be entered on the TOE. On this, the AGD section **Media Access Control Security (MACsec) in FIPS Mode Overview** also states that:

When a 128-bit cipher suite is configured, the CAK and CKN shall each be specified as a hexadecimal number with a minimum length of 1 digit and a maximum length of 32 digits. When a 256-bit cipher suite is configured, the CAK and CKN shall each be specified as a hexadecimal number with a minimum length of 2 digits and a maximum length of 64 digits.

The evaluator shall confirm the operational guidance contains instructions for either entering bit-based PSKs for each protocol identified in the requirement, generating a bit-based PSK, or both.

Evaluator Findings:

The evaluator reviewed the AGD section '**Configuring MACsec with keychain for Layer 2 Traffic**' and verified that the AGD provides instructions for entering bit-based PSKs for each protocol identified in the requirement.

The relevant information is found in the following section: **Configuring MACsec with keychain for Layer 2 Traffic**

Upon investigation, the evaluator found that the AGD claimed section states that:

2. Create the secret password to use. It is a string of hexadecimal digits up to 64 characters long. The password can include spaces if the character string is enclosed in quotation marks. The keychain's secret-data is used as a CAK.

The TOE's administrator can configure up to 64 keys. For example, the TOE's administrator can refer the following keys:

[edit]

```
security-administrator@hostname:fips# set security authentication-key-chains key-chain macsec-kc1 key 0 key-name 2345678922334455667788992223334445556667778889992222333344445551
```

Verdict:

PASS.

5.1.3.4 FIA_UIA_EXT.1 USER IDENTIFICATION AND AUTHENTICATION (CPP_ND_V3.0E)

5.1.3.4.1 FIA_UIA_EXT.1 TSS

The evaluator shall examine the TSS to determine that it describes the logon process for remote authentication mechanism (e.g. SSH public key, Web GUI password, etc.) and optional local authentication mechanisms supported by the TOE. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a “successful logon”.

Evaluator Findings:

The evaluator examined the TSS row **FIA_UIA_EXT.1** to determine that it describes the logon process for remote authentication mechanism (e.g. SSH public key, Web GUI password, etc.) and optional local authentication mechanisms supported by the TOE. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a “successful logon”.

The relevant information is found in the following section: **TOE Summary Specification** row **FIA_UIA_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

For remote administration, the TOE supports public key authentication and password-based authentication over SSHv2.

For local TOE administration through console, administrators are presented with password-based authentication and are only granted access to the TOE after entering correct user credentials.

If the remote user uses public key-based authentication, the SSH daemon would look up for the user’s public key in the authorized keys file located in the directory ‘.ssh’ in the user’s home directory (i.e. ‘~/ssh/’) and this authentication method will be attempted before any other if the user has a key available. If the key is found in the authorized keys file, the user is granted access to the TOE. If the user uses password-based authentication and they provide valid username and password, then user is granted access to the TOE. If the user enters invalid user credentials, they will not be granted access.

The evaluator shall examine the TSS to determine that it describes which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

Evaluator Findings:

The evaluator examined the TSS row **FIA_UIA_EXT.1** and determined that it describes which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

The relevant information is found in the following section: **TOE Summary Specification** row **FIA_UIA_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

No TOE administrative access is permitted until an administrator is successfully identified and authenticated. Every user that attempts to authenticate is first provided non-administrative access with limited viewing functionalities.

Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality.

For remote administration, the TOE supports public key authentication and password-based authentication over SSHv2.

For local TOE administration through console, administrators are presented with password-based authentication and are only granted access to the TOE after entering correct user credentials.

The only available actions before remote administrator authentication is displaying the warning banner in accordance with FTA_TAB.1 and responding to ICMP echo requests. The only available action before local administrator's action is to display the warning banner.

For distributed TOEs the evaluator shall examine that the TSS details how Security Administrators are authenticated and identified by all TOE components. If not, all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur.

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

For distributed TOEs, the evaluator shall examine the TSS to determine that it describes for each TOE component which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for remote TOE administration and optionally for local TOE administration if claimed by the ST author. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 the TSS shall describe any unauthenticated services/services that are supported by the component.

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:

PASS.

5.1.3.4.2 FIA_UIA_EXT.1 AGD

The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services.

Evaluator Findings:

The evaluator examined the guidance documentation section ' Configure a Network Device Collaborative Protection Profile Authorized Administrator ', ' Configure SSH on the Evaluated Configuration ', ' Sample Syslog Server Configuration on a Linux System ', ' Services Available Prior to Authentication ' and determined that any necessary preparatory steps to logging in are described. For each supported login method, the evaluator ensured that the guidance documentation provides clear instructions for successfully logging on.

No configuration is necessary to ensure the services provided before login are limited.

The relevant information is found in the following section(s): **Configure a Network Device Collaborative Protection Profile Authorized Administrator, Configure SSH on the Evaluated Configuration, Sample Syslog Server Configuration on a Linux System and Services Available Prior to Authentication**

Upon investigation, the evaluator found that the AGD section '**Configure a Network Device Collaborative Protection Profile Authorized Administrator**' states that:

An NDcPPv3.0e authorized administrator must have all permissions, including the ability to change the device configuration.

To configure an authorized administrator:

1. Create a login class named security-admin with all permissions.

[edit]

security-administrator@host:fips# set system login class security-admin permissions all

2. Configure the hashed algorithm SHA256 or SHA512 for plain-text passwords. SHA512 is the default hashing algorithm.

[edit]

security-administrator@host:fips# set system login password format sha512

3. Commit the changes.

[edit]

security-administrator@host:fips# commit

4. Define the NDcPPv3.0e user authorized administrator using the command below. The same command is used to change the password.

[edit]

security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication encrypted-password

or

[edit]

security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication plain-text-password

To configure public key authentication:

[edit]

security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh ecdsa <ecdsa-public-key>

security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh rsa <rsa-public-key>

NOTE: ssh-ed25519 is not supported in FIPS mode although it is shown as a configurable option.

Further, the the evaluator found that the AGD section **Sample Syslog Server Configuration on a Linux System** and found that it gives steps for public key authentication. The AGD states that:

3. Create a user named syslog-mon with the class monitor, and with public key–based authentication that uses the syslogmonitor key pair from the key pair file located on the remote syslog server.

[edit system login]

security-administrator@host:fips# set user syslog-mon class monitor authentication ssh-rsa "public-key"

The evaluator also found that the AGD section '**Configure SSH on the Evaluated Configuration**' states that:

To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521
security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa
```

NOTE: Although the last hostkey algorithm option mentioned above only mentions RSA, it covers all the claimed RSA algorithms, that is SSH-RSA, RSA-SHA2-256, and RSA-SHA2-512

2. Specify the SSH key-exchange for Elliptic Curve Diffie-Hellman keys for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521
```

3. Specify all the permissible message authentication code algorithms for SSHv2

[edit]

```
security-administrator@host:fips# set system services ssh macs hmac-sha2-256
security-administrator@host:fips# set system services ssh macs hmac-sha2-512
```

4. Specify the ciphers allowed for protocol version 2.

[edit]

```
security-administrator@host:fips# set system services ssh ciphers aes128-cbc
security-administrator@host:fips# set system services ssh ciphers aes256-cbc
security-administrator@host:fips# set system services ssh ciphers aes128-ctr
security-administrator@host:fips# set system services ssh ciphers aes256-ctr
security-administrator@host:fips# set system services ssh ciphers aes128-gcm@openssh.com
security-administrator@host:fips# set system services ssh ciphers aes256-gcm@openssh.com
```

5. Commit the changes:

[edit]

```
security-administrator@host:fips# commit
```

The evaluator ensured that the guidance documentation provides clear instructions for successfully logging on for each method:

Sample output of Logging in via SSH (password):

```
This is a LOGIN message for MX 480. Authorized users only !!!
```

```
FreeBSD/amd64 (MX480_TOE) (ttyu0)
```

```
login: acumensec
```

```
Password:
```

```
Last login: Thu Sep 4 11:13:14 on ttyu0
```

```
--- JUNOS 24.4R1.9 Kernel 64-bit JNPR-15.0-20241104.1ed86e6_buil acumensec@MX480_TOE:fips>
```

Sample output of Logging in via SSH (pubkey):

```
acumensec@mx480-1:~$ ssh tester@10.1.2.158 -i ~/.ssh/id_rsa_256
```

```
This is a LOGIN message for MX 480.
```

```
Authorized users only !!!
```

```
Last login: Mon Sep 8 09:44:44 2025 from 10.1.5.84
--- JUNOS 24.4R1.9 Kernel 64-bit JNPR-15.0-20241104.1ed86e6_buil
tester@MX480_TOE:fips>
```

Sample output of Logging in via local authentication:

```
This is a LOGIN message for MX 480. Authorized users only !!!
FreeBSD/amd64 (MX480_TOE) (ttyu0)
login: acumensec
Password:
Last login: Thu Sep 4 11:13:14 on ttyu0
--- JUNOS 24.4R1.9 Kernel 64-bit JNPR-15.0-20241104.1ed86e6_buil
acumensec@MX480_TOE:fips>
```

Also, by referring to the section **Services Available Prior to Authentication** the evaluator found that no configuration is necessary to ensure the services provided before login are limited and the AGD states that:

The Junos OS in FIPS mode of operation allows a wide range of capabilities for users, and authentication is identity-based.
Prior to authentication, the only Junos OS managed responses provided to the administrator are:

- Display of the access banner
- ICMP echo responses (for remote management terminals).

Verdict:

PASS.

5.1.3.5 FIA_UAU.7 PROTECTED AUTHENTICATION FEEDBACK (CPP_ND_V3.0E)

5.1.3.5.1 FIA_UAU.7 TSS

None.

5.1.3.5.2 FIA_UAU.7 AGD

The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed.

Evaluator Findings:

The evaluator examined the guidance documentation section **Sample Output for Local Administrative Session Termination** to determine that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed.

The relevant information is found in the following section: **Sample Output for Local Administrative Session Termination**

Upon investigation, the evaluator found that the claimed AGD section states that:
For password authentication (both remote and local), login interacts with a user to request a username and password to establish and verify the user's identity. The username entered by the administrator at the username prompt is reflected to the screen, but no feedback to screen is provided while the entry made by the administrator at the password prompt until the Enter key is pressed, following which either access to the

CLI is provided (correct password) or the password prompt is presented again (incorrect password). This is the default behavior and does not require any configuration.

NOTE: Local console sessions display the banner FreeBSD/amd64 (hostname) (ttyu0) before the login prompt. This message is not shown during remote SSH access. Administrators should use this difference to verify whether the session is local or remote.

Verdict:

PASS.

5.1.4 SECURITY MANAGEMENT (FMT)

5.1.4.1 FMT_MTD.1/CRYPTOKEYS MANAGEMENT OF TSF DATA (CPP_ND_V3.0E)

5.1.4.1.1 FMT_MTD.1/CRYPTOKEYS TSS

For distributed TOEs see Section 2.4.1.1.

Evaluator Findings:

The TOE is not distributed; hence, this requirement is not applicable.

For non-distributed TOEs, the evaluator shall ensure the TSS lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

Evaluator Findings:

The evaluator examined the TSS row **FMT_MTD.1/CryptoKeys** and ensured that, for non-distributed TOEs, it lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FMT_MTD.1/CryptoKeys**

Upon investigation, the evaluator found that the TSS states that:

The Security Administrator can manage the following keys, with the specified operations available:

- **SSH Keys (ECDSA, RSA):** Import, modification, and deletion. These SSH Key operations are performed using the system services ssh and system login user <username> authentication hierarchies.
- **Pre-shared CAKeys:** Configuration (which includes configuration, modification, and deletion). These Pre-Shared CAK operations are performed using the security macsec connectivity-association <ca name> pre-shared-key cak hierarchy.

Verdict:

PASS.

5.1.4.1.2 FMT_MTD.1/CRYPTOKEYS AGD

For distributed TOEs see Section 2.4.1.2.

Evaluator Findings:

The TOE is not distributed; hence, this requirement is not applicable.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the operations are performed on the keys the Security Administrator is able to manage.

Evaluator Findings:

The evaluator ensured the Guidance Documentation describes how the operations are performed on the keys the Security Administrator is able to manage.

The relevant information is found in the following section(s): **Configure a Network Device Collaborative Protection Profile Authorized Administrator, Configuring MACsec with keychain using Layer 3 Traffic, Media Access Control Security (MACsec) in FIPS Mode Overview, Generate a New SSH Host Key and Delete SSH Keys**

Upon investigation, the evaluator found that the AGD section **Configure a Network Device Collaborative Protection Profile Authorized Administrator** states that:

4. To configure public key authentication:

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh ecdsa <ecdsa-public-key>
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh rsa <rsa-public-key>
```

To delete a configured login credential, use the following command:

[edit]

```
security-administrator@host:fips# delete system login user NDcPPv3-user class security-admin authentication
```

5. Set the log-key-changes configuration statement to log when SSH authentication keys are added or removed.

[edit]

```
security-administrator@host:fips# set system services ssh log-key-changes
```

NOTE: When the log-key-changes configuration statement is enabled and committed (with the commit command in configuration mode), Junos OS logs the changes to the set of authorized SSH keys for each user (including the keys that were added or removed). Junos OS logs the differences since the last time the log-key-changes configuration statement was enabled. If the log-key-changes configuration statement was never enabled, then Junos OS logs all the authorized SSH keys.

6. Commit the changes.

[edit]

```
security-administrator@host:fips# commit
```

Further the evaluator examined AGD section 'Configuring MACsec on a Device Running Junos OS' and found that:

2. Create the pre-shared key by configuring the connectivity association key name (CKN) and connectivity association key (CAK).

[edit]

```
security-administrator@host:fips# prompt security macsec connectivity-association connectivity-association-
```

```

name pre-shared-key cak
New cak (secret):
Retype new cak (secret):
security-administrator@host:fps# set security macsec connectivity-association connectivity-association-
name pre-shared-key ckn hexadecimal-number
security-administrator@host:fps# set security macsec connectivity-association connectivity-association-
name replay-protect replay-window-size number-of-packets

```

Also, the evaluator examined the sections **Configuring MACsec with keychain using Layer 3 Traffic** and **Media Access Control Security (MACsec) in FIPS Mode Overview** and found that it has the following commands:

To delete an existing CAK, use the following command:

```

security-administrator@host:fps# delete security macsec connectivity association ca_name pre-shared-key
cak

```

To delete an existing key from a keychain, use the following command:

```

[edit]
security-administrator@host:fps# delete security authentication-key-chains key-chain
macsec-kc1 key 0

```

Upon further investigation, the evaluator determined that in the section **Generate a New SSH Host Key**, the AGD instructs the administrator that **to generate a new SSH host key, issue one of the following commands from the shell prompt, selecting the appropriate key size:**

- `ssh-keygen -t ecdsa -b [256|384|521] -f /etc/ssh/ssh_host_ecdsa_key`
- `ssh-keygen -t rsa -b [2048|3072|4096] -f /etc/ssh/ssh_host_rsa_key`

In addition, the **Delete SSH Keys** section states that **If it becomes necessary to delete one or both of SSH host keys, issue the command from the shell prompt for the selected key.**

```

rm /etc/ssh/<key-name>

```

Verdict:

PASS.

5.1.4.2 FMT_MTD.1/COREDATA MANAGEMENT OF TSF DATA (CPP_ND_V3.0E)

5.1.4.2.1 FMT_MTD.1/COREDATA TSS

For each administrative function identified in the guidance documentation that is accessible through an interface prior to administrator log-in, the evaluator shall confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

Evaluator Findings:

The evaluator confirmed that the TSS row **FMT_MTD.1/CoreData** details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FMT_MTD.1/CoreData**

Upon investigation, the evaluator found that the TSS states that:

The TOE restricts the ability to manage the TOE to Security Administrators. Administrative users are required to login before being provided with access to any administrative functions. Non-security administrators are not allowed to modify any TOE functions. No interface is available to an unauthenticated user except the login prompt. Any commands used to modify TOE functions are not made available to non-administrative users and its attempt to use them will result in an invalid action error.

If the TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator shall examine the TSS to determine that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted.

Evaluator Findings:

The TOE does not support this functionality, hence this part of the objective is not applicable.
--

Verdict:

PASS.

5.1.4.2.2 FMT_MTD.1/COREDATA AGD

The evaluator shall review the guidance documentation to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

Evaluator Findings:

The evaluator reviewed the guidance documentation and determined that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

The relevant information is found in the following section(s):
--

Administrative Activity	Method (Command/GUI Configuration)	Section
Start-up and shutdown of the audit functions	user@host> request vmhost reboot Note: There is no manual startup/shutdown of the local audit function, which is tied to startup/shutdown of the TOE itself, logs for which implicitly indicate the audit function stopping and starting as well.	Install Junos OS Software Packages
Creating Users	[edit] security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication encrypted-password or [edit] security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication plain-text-password To configure public key authentication: [edit] security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh ecdsa <ecdsa-public-key> security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh rsa <rsa-public-key>	Configure a Network Device Collaborative Protection Profile Authorized Administrator
Setting the time	Set the date and time from the operational mode. Date and time format is YYYYMMDDHHMM.ss. security-administrator@hostname:fips> set date 201803202034.00	Customize Time in FIPS Mode
Configuring Admin Timeout	[edit] security-administrator@host:fips# set system login class security-admin idle-timeout 2	Configure Session Termination
Configuring Access Banner	[edit] security-administrator@host:fips# set system login message login-message-banner-text	Configure System Login Message and Announcement
Setting Password Length	[edit] security-administrator@host:fips# set system login password minimum-length 15 [edit] security-administrator@host:fips# set system login password minimum-upper-case 1 security-administrator@host:fips# set system login password minimum-lower-case 1 security-administrator@host:fips# set system login password minimum-numeric 1 security-administrator@host:fips# set system login password minimum-punctuations 1	Configure Password Rules for an Authorized Administrator
Configuring SSH	[edit] security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256	Configure SSH on the Evaluated

	security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384 security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521 security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa [edit] security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256 security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384 security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521 [edit] security-administrator@host:fips# set system services ssh macs hmac-sha2-256 security-administrator@host:fips# set system services ssh macs hmac-sha2-512 [edit] security-administrator@host:fips# set system services ssh ciphers aes128-cbc security-administrator@host:fips# set system services ssh ciphers aes256-cbc security-administrator@host:fips# set system services ssh ciphers aes128-ctr security-administrator@host:fips# set system services ssh ciphers aes256-ctr security-administrator@host:fips# set system services ssh ciphers aes128-gcm@openssh.com security-administrator@host:fips# set system services ssh ciphers aes256-gcm@openssh.com	Configuration
Configure audit	[edit system syslog] security-administrator@host:fips# set archive files 2 [edit system syslog] security-administrator@host:fips# set file syslog any any [edit system syslog] security-administrator@host:fips# set file syslog archive size 10000000 [edit system syslog] security-administrator@host:fips# set file syslog explicit-priority [edit system syslog] security-administrator@host:fips# set file syslog structured-data [edit] security-administrator@host:fips# commit	Configure Audit Log Options on Device in FIPS Mode
Manage a PSK-based CAK and install it in the	security-administrator@host:fips# prompt security macsec connectivity-association ca_name pre-shared-key cak New cak (secret): Retype new cak (secret):	Media Access Control Security (MACsec) in

device		FIPS Mode Overview
Manage the key server to create, delete, and activate MKA participants	<pre> [edit] security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name exclude-protocol protocol- name security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name include-sci security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name mka key-server-priority priority-number security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name mka transmit-interval interval security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name offset 30 [edit] security-administrator@host:fips# prompt security macsec connectivity- association connectivity-association-name pre-shared-key cak New cak (secret): Retype new cak (secret): security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name pre-shared-key ckn hexadecimal-number security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name replay-protect replay-window- size number-of-packets [edit] security-administrator@host:fips# set security macsec connectivity- association CA1 security-mode security-mode [edit] security-administrator@host:fips# set security macsec interfaces interface-name connectivity-association connectivity-association-name user@host# deactivate security macsec user@host# activate security macsec </pre>	Configuring MACsec on a Device Running Junos OS, Disable and Restart MACsec Sessions
Specify the lifetime of a CAK	<pre> [edit] security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 0 key-name 23456789223344556677889922233344455566677788899922233334444 5551 security-administrator@host:fips# set security authentication-key-chains key-chain macsec-kc1 key 0 start-time 2018-03-20.20:35 </pre>	Configuring MACsec with keychain using Layer 3 Traffic
Enable, disable, or delete a PSK-based CAK	<pre> security-administrator@host:fips# prompt security macsec connectivity- association ca_name pre-shared-key cak New cak (secret): Retype new cak (secret): security-administrator@host:fips# delete security macsec connectivity association ca_name pre-shared-key cak </pre>	Media Access Control Security (MACsec) in FIPS Mode Overview

If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to configure and maintain the trust store in a secure way. If the TOE supports loading of CA certificates, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to securely load CA certificates into the trust store. The evaluator shall also review the guidance documentation to determine that it explains how to designate a CA certificate a trust anchor.

Evaluator Findings:

The ST does not select the TOE handling of X.509v3 certificates and implementation of a trust store hence this part of the objective is not applicable.

Verdict:

PASS.

5.1.4.3 FMT_MOF.1/FUNCTIONS MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR (CPP_ND_V3.0E)

5.1.4.3.1 FMT_MOF.1/FUNCTIONS TSS

For distributed TOEs see Section 2.4.1.1.

Evaluator Findings:

The TOE is not distributed; hence, this requirement is not applicable.

For non-distributed TOEs, the evaluator shall ensure the TSS for each administrative function identified the TSS details how the Security Administrator determines or modifies the behaviour of (whichever is supported by the TOE) transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full (whichever is supported by the TOE).

Evaluator Findings:

The evaluator examined the TSS row **FMT_MOF.1/Functions** and ensured that, for non-distributed TOEs, it details how the Security Administrator determines or modifies the behaviour of transmitting audit data to an external IT entity and handling of audit data.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FMT_MOF.1/Functions**

Upon investigation, the evaluator found that the TSS states that:

The TOE allows administrators to modify the behaviour of the following administrative functions using CLI commands:

- transmission of audit data to an external IT entity – using the ‘system services netconf’ configuration hierarchy.
- handling of local audit data– using the ‘system syslog’ configuration hierarchy.

Verdict:

PASS.

5.1.4.3.2 FMT_MOF.1/FUNCTIONS AGD

For distributed TOEs see Section 2.4.1.2.

Evaluator Findings:

The TOE is not distributed; hence, this requirement is not applicable.
--

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the Security Administrator determines or modifies the behaviour of (whichever is supported by the TOE) transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full (whichever is supported by the TOE) are performed to include required configuration settings.

Evaluator Findings:

The evaluator examined the Guidance Documentation section ' Sample Syslog Server Configuration on a Linux System ' and ' Configure Audit Log Options on Device in FIPS Mode ' and ensured that, for non-distributed TOEs, it describes how the Security Administrator modifies the behaviour of transmitting audit data to an external IT entity and handling of audit data are performed to include required configuration settings.

The relevant information is found in the following section(s): **Sample Syslog Server Configuration on a Linux System** and **Configure Audit Log Options on Device in FIPS Mode**

Upon investigation, the evaluator found that the AGD section **Sample Syslog Server Configuration on a Linux System** states that:

A secure Junos OS environment requires auditing of events and storing them in a local audit file. The recorded events are simultaneously sent to an external syslog server. A syslog server receives the syslog messages streamed from the device. The syslog server must have an SSHv2 client with NETCONF 1.0 support configured to receive the streamed syslog messages. Junos OS supports NETCONF as defined by RFC6241 (NETCONF) and RFC6242 (Using NETCONF protocol over SSH).

Use the configuration details and establish a session between the target of evaluation (TOE) and the audit server. Examine the traffic that passes between the audit server and the TOE during several activities, and the generated audit data to be transferred to the audit server.

To configure event logging to a remote server when the SSH connection to the TOE is initiated from the remote system log server.

1. Generate an RSA public key on the remote syslog server. The TOE's administrator can also use ECDSA keys on the syslog server.

```
$ ssh-keygen -b 2048 -t rsa -C 'syslog-monitor key pair' -f ~/.ssh/syslog-monitor
```

The TOE's administrator will be prompted to enter the desired pass phrase. The storage locations for the syslog-monitor key pair is displayed.

2. On the TOE, create a class named monitor that has permission to trace events.

[edit system login]

```
security-administrator@host:fips# set class monitor permissions trace
```

3. Create a user named syslog-mon with the class monitor, and with public key-based or password-based authentication.

For public key-based authentication:

[edit system login]

```
security-administrator@host:fips# set user syslog-mon class monitor authentication ssh-rsa "public-key"
```

For password-based authentication:

[edit system login]

```
security-administrator@host:fips# set user syslog-mon class monitor authentication (plain-text-password | encrypted-password)
```

4. Set up NETCONF with SSH.

[edit system services]

```
security-administrator@host:fips# set netconf ssh
```

5. Configure syslog to log all the messages at /var/log/file-name.

[edit system]

```
security-administrator@host:fips# set syslog file file-name any any
commit
```

6. To configure syslog in year format:

[edit system]

```
security-administrator@host:fips# syslog time-format year
commit
```

7. On the remote system log server, start up the SSH agent ssh-agent. The start up is required to simplify the handling of the syslog-monitor key.

```
$ eval `ssh-agent -s`
```

8. On the remote syslog server, add the syslog-monitor key pair to the ssh-agent.

```
$ ssh-add ~/.ssh/syslog-monitor
```

The TOE's administrator will be prompted to enter the desired passphrase. Enter the same passphrase used in Step 1.

9. After logging in to the external_syslog_server session, establish a tunnel to the device and start NETCONF.

```
Security-administrator@host:fips# $ssh syslog-mon@NDcPP_TOE -s netconf > test.out
```

10. After NETCONF is established, configure a system log events message stream. This RPC will cause the NETCONF service to start transmitting messages over the SSH connection that is established.

```
<rpc><get-syslog-events><stream>messages</stream></get-syslog-events></rpc>
```

Further the evaluator found that the AGD section **Configure Audit Log Options on Device in FIPS Mode** states that:

To configure audit log options:

1. Specify the number of files to be archived in the system logging facility.

[edit system syslog]

```
security-administrator@host:fips# set archive files 2
```

2. Specify the file in which to log data.

[edit system syslog]

```
security-administrator@host:fips# set file syslog any any
```

3. Specify the size of files, in bytes, to be archived. The size can also be specified in shorthand form, for example, `10m` for 10 MB and `1g` for 1 GB.

[edit system syslog]

```
security-administrator@host:fips# set file syslog archive size 10000000
```

4. Specify the priority and facility in messages for the system logging facility.

[edit system syslog]

```
security-administrator@host:fips# set file syslog explicit-priority
```

5. Commit the changes:

[edit]

```
security-administrator@host:fips# commit
```

The above section also further explains the overwriting behavior as follows:

The local logs are automatically overwritten according to configurable limits on storage volume. The default maximum size is 1 MB, which can be modified by the user, using the “size” argument for the “set system syslog file <filename> archive” CLI command.

The Junos OS defines an active log file and a number of “archive” files (10 by default, but configurable from 1 to 1000). When the active log file reaches its maximum size, the logging utility closes the file, compresses it, and names the compressed archive file ‘logfile.0.gz’. The logging utility then opens and writes to a new active log file. When the new active log file reaches the configured maximum size ‘logfile.0.gz’ is renamed ‘logfile.1.gz’, and the active log file is closed, compressed, and renamed ‘logfile.0.gz’. When the maximum number of archive files is reached and when the size of the active file reaches the configured maximum size, the contents of the oldest archived file are deleted so the current active file can be archived.

Verdict:

PASS.

5.1.4.4 FMT_MOF.1/MANUALUPDATE MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR (CPP_ND_V3.0E)

5.1.4.4.1 FMT_MOF.1/MANUALUPDATE TSS

For distributed TOEs see Section 5.1.4.1. There are no specific requirements for non-distributed TOEs.

5.1.4.4.2 FMT_MOF.1/MANUALUPDATE AGD

The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described. The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).

Evaluator Findings:

The evaluator examined the guidance documentation sections ‘**Download Software Packages from Juniper Networks**’ and ‘**Install Junos OS Software Packages**’ and determined that any necessary steps to perform manual update are described. The guidance documentation also provides warnings regarding functions that may cease operating during the update.

The relevant information is found in the following section(s): **Download Software Packages from Juniper Networks** and **Install Junos OS Software Packages**

Upon investigation, the evaluator found that the AGD section **Download Software Packages from Juniper Networks** that:

The TOE’s administrator can download the Junos OS software package from the Juniper Networks website. Before the TOE’s administrator begin to download the software, ensure that the TOE’s administrator have a Juniper Networks Web account and a valid support contract. To obtain an account, complete the registration form at the Juniper Networks website: <https://userregistration.juniper.net/>

To download software packages from Juniper Networks:

- 1. Using a Web browser, follow the links to the download URL on the Juniper Networks webpage. <https://support.juniper.net/support/downloads/>.**
- 2. Log in to the Juniper Networks authentication system using the username (generally your e-mail address) and password supplied by Juniper Networks representatives.**
- 3. Download the software.**

Further the evaluator found that the AGD section **Install Junos OS Software Packages** states that:

The administrator can use this procedure to upgrade Junos OS on the device.

NOTE: Junos OS is delivered in signed packages with manifests which validate the immutability of the contents. There is also a signature of the manifest that can be validated with public keys that are packaged. During install of the software packages, Junos OS validates the signature and the hashes of the contents present in the manifest.

If the image fails the signature check, then the image is deleted from the device and no upgrade occurs, with the appropriate error visible on the CLI. Successful update is only initiated when the signature verification of the TOE passes. Logs are generated in both cases. Software update performed in FIPS mode is similar to that in non-FIPS mode for the functionalities available on the device. TOE functionalities are unavailable only during the reboot following the upgrade. Junos OS does not support delayed activation while performing software updates.

To install software upgrades on the device:

1. Download the software package as described in "Download Software Packages from Juniper Networks" on page 20. (of the AGD)

2. If the TOE's administrator has not already done so, connect to the console port on the device from your management device, and log in to the Junos OS CLI.

3. Copy the software package to the device. The best practice is to use SFTP to copy the file to the /var/tmp/ directory.

4. Install the new package on the device:

```
user@host> request vmhost software add package
```

Replace package with local directory on the device where the software package is copied. For example, /var/tmp/package.tgz.

5. Reboot the device to load the installation:

```
user@host> request vmhost reboot
```

6. Once the system completes rebooting, log in and use the show version command to verify the successful installation of the new software version.

```
user@host> show version
```

Hostname: hostname

Model: mx480

Family: junos

Junos: 24.4R2.9

JUNOS OS Kernel 64-bit [20250529.3e76435_builder_bsd15_244]

JUNOS OS libs [20250529.3e76435_builder_bsd15_244]

JUNOS OS libs compat32 [20250529.3e76435_builder_bsd15_244]

JUNOS OS vmguest [20250529.3e76435_builder_bsd15_244]

JUNOS OS pkgrs [20250529.3e76435_builder_bsd15_244]

JUNOS modules [20250613.010225_builder_junos_244_r2]

JUNOS OS 32-bit compatibility [20250529.3e76435_builder_bsd15_244]

JUNOS OS runtime [20250529.3e76435_builder_bsd15_244]

JUNOS OS crypto [20250529.3e76435_builder_bsd15_244]

JUNOS OS boot-ve files [20250529.3e76435_builder_bsd15_244]

JUNOS mx libs [20250613.010225_builder_junos_244_r2]

JUNOS OS time zone information [20250529.3e76435_builder_bsd15_244]

JUNOS network stack and utilities [20250613.010225_builder_junos_244_r2]

JUNOS SDN Software Suite [20250613.010225_builder_junos_244_r2]

JUNOS libs [20250613.010225_builder_junos_244_r2]

JUNOS py extensions [20250613.010225_builder_junos_244_r2]

JUNOS py base [20250613.010225_builder_junos_244_r2]

JUNOS OS package [20250407.212945_builder_main]

JUNOS jail runtime [20250529.3e76435_builder_bsd15_244]

JUNOS libs compat32 [20250613.010225_builder_junos_244_r2]

JUNOS daemons [20250613.010225_builder_junos_244_r2]
JUNOS vmguest (MTX) [20250613.010225_builder_junos_244_r2]
JUNOS mtx network modules [20250613.010225_builder_junos_244_r2]
JUNOS mx daemons [20250613.010225_builder_junos_244_r2]
JUNOS OS network modules [20250529.3e76435_builder_bsd15_244]
JUNOS vmguest [20250613.010225_builder_junos_244_r2]
JUNOS mx libs compat32 [20250613.010225_builder_junos_244_r2]
JUNOS runtime [20250613.010225_builder_junos_244_r2]
JUNOS mx runtime [20250613.010225_builder_junos_244_r2]
JUNOS mx modules [20250613.010225_builder_junos_244_r2]
JUNOS Services Telemetry [20250613.010225_builder_junos_244_r2]
JUNOS Services TCP-LOG [20250613.010225_builder_junos_244_r2]
JUNOS Services SSL [20250613.010225_builder_junos_244_r2]
JUNOS Services SOFTWARE [20250613.010225_builder_junos_244_r2]
JUNOS Services Stateful Firewall [20250613.010225_builder_junos_244_r2]
JUNOS Services RTCOM [20250613.010225_builder_junos_244_r2]
JUNOS Services RPM [20250613.010225_builder_junos_244_r2]
JUNOS Services PCEF package [20250613.010225_builder_junos_244_r2]
JUNOS Services NAT [20250613.010225_builder_junos_244_r2]
JUNOS Services Mobile Subscriber Service Container package
[20250613.010225_builder_junos_244_r2]
JUNOS Services MobileNext Software package [20250613.010225_builder_junos_244_r2]
JUNOS Services Logging Report Framework package [20250613.010225_builder_junos_244_r2]
JUNOS Services Jflow Container package [20250613.010225_builder_junos_244_r2]
JUNOS Services Deep Packet Inspection package [20250613.010225_builder_junos_244_r2]
JUNOS Services IPSec [20250613.010225_builder_junos_244_r2]
JUNOS Services IDS [20250613.010225_builder_junos_244_r2]
JUNOS Services HTTP Content Management package [20250613.010225_builder_junos_244_r2]
JUNOS Services DNS Filter package (i386) [20250613.010225_builder_junos_244_r2]
JUNOS Services Crypto [20250613.010225_builder_junos_244_r2]
JUNOS Services Captive Portal and Content Delivery Container package
[20250613.010225_builder_junos_244_r2]
JUNOS Services COS [20250613.010225_builder_junos_244_r2]
JUNOS Appld Services [20250613.010225_builder_junos_244_r2]
JUNOS Services Application Level Gateways [20250613.010225_builder_junos_244_r2]
JUNOS Extension Toolkit [20250613.010225_builder_junos_244_r2]
JUNOS Phone-home [20250613.010225_builder_junos_244_r2]
JUNOS Packet Forwarding Engine Support (wrlinux9) [20250613.010225_builder_junos_244_r2]
JUNOS Packet Forwarding Engine Support (MX/EX92XX Common)
[20250613.010225_builder_junos_244_r2]
JUNOS Packet Forwarding Engine FIPS Support [24.4R2.9]
JUNOS Packet Forwarding Engine Support (M/T Common) [20250613.010225_builder_junos_244_r2]
JUNOS Packet Forwarding Engine Support (aft) [20250613.010225_builder_junos_244_r2]
JUNOS Packet Forwarding Engine Support (MX Common) [20250613.010225_builder_junos_244_r2]
JUNOS Juniper Malware Removal Tool (JMRT) [1.0.0+24.4R2.9]
JUNOS J-Insight [20250613.010225_builder_junos_244_r2]
JUNOS jfirmware [20250529.211134_builder_junos_244_r2]
JUNOS Online Documentation [20250613.010225_builder_junos_244_r2]
JUNOS FIPS mode utilities [20250613.010225_builder_junos_244_r2]
JUNOS dsa [20250613.010225_builder_junos_244_r2]
JUNOS SSH Tunnel Daemon [20250613.010225_builder_junos_244_r2]
JUNOS sflow mx [20250613.010225_builder_junos_244_r2]
JUNOS na telemetry [24.4R2.9]

JUNOS full sysmond [20250613.010225_builder_junos_244_r2]
 JUNOS support scripts [20250613.010225_builder_junos_244_r2]
 JUNOS Security Intelligence [20250613.010225_builder_junos_244_r2]
 Junos scheduler tracing [20250613.010225_builder_junos_244_r2]
 JUNOS RPD Telemetry Application [24.4R2.9]
 JUNOS python routing scripts for consistency check in RIB/FIB/PFE [20250613.010225_builder_junos_244_r2]
 JUNOS Routing mpls-oam-basic [20250613.010225_builder_junos_244_r2]
 JUNOS Routing mpls-oam-advanced [20250613.010225_builder_junos_244_r2]
 JUNOS Routing lsys [20250613.010225_builder_junos_244_r2]
 JUNOS Routing controller-internal [20250613.010225_builder_junos_244_r2]
 JUNOS Routing controller-external [20250613.010225_builder_junos_244_r2]
 JUNOS Routing 32-bit Compatible Version [20250613.010225_builder_junos_244_r2]
 JUNOS Routing aggregated [20250613.010225_builder_junos_244_r2]
 JUNOS probe utility [20250613.010225_builder_junos_244_r2]
 JUNOS common platform support [20250613.010225_builder_junos_244_r2]
 JUNOS Openconfig [24.4R2.9]
 JUNOS L2 RSI Scripts [20250613.010225_builder_junos_244_r2]
 JUNOS Key Manager [20250613.010225_builder_junos_244_r2]
 JUNOS SQL Sync Daemon [20250613.010225_builder_junos_244_r2]
 JUNOS mtx Data Plane Crypto Support [20250613.010225_builder_junos_244_r2]
 JUNOS Broadband Edge User Plane Apps [24.4R2.9]
 Junos Remote Attestation package [1.0.0+20250613.010225_builder_junos_244_r2]
 JUNOS appidd-mx application-identification daemon [20250613.010225_builder_junos_244_r2]
 JUNOS Common BIOS [20250613.010225_builder_junos_244_r2]
 JUNOS Services URL Filter package [20250613.010225_builder_junos_244_r2]
 JUNOS Services TLB Service PIC package [20250613.010225_builder_junos_244_r2]
 JUNOS Juniper Malware Removal Tool (JMRT) Test [1.0.0+24.4R2.9]

For distributed TOEs the guidance documentation shall describe all steps how to update all TOE components. This shall contain description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable).

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:

PASS.

5.1.4.5 FMT_MOF.1/SERVICES MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR (CPP_ND_V3.0E)

5.1.4.5.1 FMT_MOF.1/SERVICES TSS

For distributed TOEs see Section 2.4.1.1.

Evaluator Findings:

The TOE is not distributed; hence, this requirement is not applicable.

For non-distributed TOEs, the evaluator shall ensure the TSS lists the services the Security Administrator is able to start and stop and how that operation is performed.

Evaluator Findings:
The evaluator examined the TSS row FMT_MOF.1/ Services and ensured that, for non-distributed TOEs, it lists the services the Security Administrator is able to start and stop and how that operation is performed. The relevant information is found in the following section(s): TOE Summary Specification row FMT_MOF.1/Services Upon investigation, the evaluator found that the TSS states that: The Security Administrator has the capability to: • Start/stop and modify the behaviour of the trusted communication channel to external syslog server using the 'system services netconf' hierarchy command. • Start/stop and modify the behaviour of the trusted communication path for remote administrative sessions (SSH) using the 'system services ssh' hierarchy command. • Start/Stop and modify the behaviour of the trusted communication channel with MACsec peers using the 'security macsec' hierarchy commands.

Verdict:

PASS.

5.1.4.5.2 FMT_MOF.1/SERVICES AGD

For distributed TOEs see Section 2.4.1.2.

Evaluator Findings:
The TOE is not distributed; hence, this requirement is not applicable.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the TSS lists the services the Security Administrator is able to start and stop and how that operation is performed.

Evaluator Findings:
The evaluator examined the AGD sections 'Configure SSH on the Evaluated Configuration' and 'Disable and Restart MACsec Sessions' and ensured that, for non-distributed TOEs, it describes how the TSS lists the services the Security Administrator is able to start and stop and how that operation is performed. The relevant information is found in the following section(s): Configure SSH on the Evaluated Configuration and Disable and Restart MACsec Sessions Upon investigation, the evaluator found that the AGD section Configure SSH on the Evaluated Configuration states that: NOTE: To disable SSH service, you can deactivate and commit the SSH configurations: security-administrator@host:fips# deactivate system services ssh NOTE: To disable Netconf service, you can deactivate and commit the netconf configurations: security-administrator@host:fips# deactivate system services netconf ssh The evaluator also confirmed that the operational guidance specifies that a secure Junos OS environment requires auditing of events with storage in a local audit file and simultaneous transmission of audit records to an external syslog server. The guidance further specifies that the syslog server must support SSHv2 with NETCONF 1.0 in order to receive streamed syslog messages, and that Junos OS supports NETCONF as defined by RFC 6241 and RFC 6242. Based on this information, the evaluator determined that the remote audit (syslog) transmission

channel operates over NETCONF secured by SSH. Therefore, the NETCONF over SSH service is directly tied to the establishment and maintenance of the remote syslog session used for audit data transfer.

Also, the evaluator found that the AGD section **Disable and Restart MACsec Sessions** states that:

To disable and restart the MACsec sessions use the following configurations:

• **To disable the MACsec session:**

user@host# deactivate security macsec

• **To restart the MACsec session:**

user@host# run restart dot1x-protocol

or

user@host# activate security macsec

Verdict:

PASS.

5.1.4.6 FMT_SMF.1 SPECIFICATION OF MANAGEMENT FUNCTIONS (CPP_ND_V3.0E)

5.1.4.6.1 FMT_SMF.1 TSS (CONTAINING ALSO REQUIREMENTS ON GUIDANCE DOCUMENTATION AND TESTS)

The evaluator shall examine the TSS, the Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE.

Evaluator Findings:

The evaluator examined the TSS row **FMT_SMF.1**, the Guidance Documentation section '**Overview of Roles and Services for Junos OS**' and the TOE as observed during all other testing and confirmed that the management functions specified in FMT_SMF.1 are provided by the TOE.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FMT_SMF.1** and AGD sections **Overview of Roles and Services for Junos OS** and **Security Administrator Role and Responsibilities**

Upon investigation, the evaluator found that the TSS states that:

The available management functions are listed below and these can be accessed via the SSH remotely or locally via console:

- *Ability to administer the TOE remotely*
- *Ability to configure the access banner*
- *Ability to configure the remote session inactivity time before session termination*
- *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates*
- *Ability to start and stop services*
- *Ability to configure local audit behaviour*
- *Ability to modify the behaviour of the transmission of audit data to an external IT entity*
- *Ability to manage the cryptographic keys*
- *Ability to configure the cryptographic functionality*
- *Ability to configure thresholds for SSH rekeying*
- *Ability to re-enable an Administrator account*
- *Ability to configure the local session inactivity time before session termination or locking*
- *Ability to configure the authentication failure parameters for FIA_AFL.1*
- *Ability to set the time which is used for timestamps*

- *Ability to configure NTP*
- *Ability to administer the TOE locally*
- *Ability to manage the trusted public keys database*

Further the evaluator found that the AGD section **Overview of Roles and Services for Junos OS** states that:

Security Administrator roles and responsibilities are as follows:

- 1. Ability to administer the TOE remotely**
- 2. Ability to configure the access banner**
- 3. Ability to configure the remote session inactivity time before session termination**
- 4. Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates**
- 5. Ability to start and stop services**
- 6. Ability to configure local audit behaviour (changes to storage locations for audit; changes to behaviour when local audit storage space is full, changes to local audit storage size)**
- 7. Ability to modify the behaviour of the transmission of audit data to an external IT entity**
- 8. Ability to manage the cryptographic keys**
- 9. Ability to configure the cryptographic functionality**
- 10. Ability to configure thresholds for SSH rekeying**
- 11. Ability to re-enable an Administrator account**
- 12. Ability to configure the local session inactivity time before session termination or locking**
- 13. Ability to configure the authentication failure parameters for FIA_AFL.1**
- 14. Ability to set the time which is used for time-stamps**
- 15. Ability to configure NTP**
- 16. Ability to administer the TOE locally**
- 17. Ability to manage the trusted public keys database.**

The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

Evaluator Findings:

The evaluator confirmed that the TSS row **FMT_SMF.1** details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

The relevant information is found in the following section(s): **TOE Summary Specification** row **FMT_SMF.1**

Upon investigation, the evaluator found that the TSS states that:

The available management functions are listed below and these can be accessed via the SSH remotely or locally via console:

- *Ability to administer the TOE remotely*
- *Ability to configure the access banner*
- *Ability to configure the remote session inactivity time before session termination*
- *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates*
- *Ability to start and stop services*
- *Ability to configure local audit behaviour*
- *Ability to modify the behaviour of the transmission of audit data to an external IT entity*
- *Ability to manage the cryptographic keys*
- *Ability to configure the cryptographic functionality*
- *Ability to configure thresholds for SSH rekeying*
- *Ability to re-enable an Administrator account*
- *Ability to configure the local session inactivity time before session termination or locking*
- *Ability to configure the authentication failure parameters for FIA_AFL.1*
- *Ability to set the time which is used for timestamps*
- *Ability to configure NTP*
- *Ability to administer the TOE locally*
- *Ability to manage the trusted public keys database*

The evaluator shall examine the TSS and the Guidance Documentation to verify they both describe the local administrative interface.

Evaluator Findings:

The evaluator examined the TSS row **FMT_SMF.1** and the Guidance Documentation section **Management Interfaces Overview** to verify they both describe the local administrative interface.

The relevant information is found in the following section(s): **TOE Summary Specification** section row **FMT_SMF.1** and AGD section **Management Interfaces Overview**

Upon investigation, the evaluator found that the TSS states that:

The local interface of the TOE can be accessed directly through a serial port connection. The presence of a system variable called "SSH_CONNECTION" is used to determine whether the connection to the TOE is local (via serial) or remote (via SSH). This variable is present during SSH connections and absent during console access.

Also, the evaluator found that the claimed AGD section **Management Interfaces Overview** states that:

The following management interfaces can be used in the evaluated configuration:

- **Local Management Interfaces—The RJ-45 console port on the device is configured as RS-232 data terminal equipment (DTE). The TOE's administrator can use the command-line interface (CLI) over this port to configure the device from a terminal.**

The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

Evaluator Findings:
The evaluator ensured the Guidance Documentation sections Management Interfaces Overview and Sample Output for Local Administrative Session Termination includes appropriate warnings for the administrator to ensure the interface is local. The relevant information is found in the following section: Management Interfaces Overview and Sample Output for Local Administrative Session Termination Upon investigation, the evaluator found that the claimed AGD section states that: The following management interfaces can be used in the evaluated configuration: • Local Management Interfaces—The RJ-45 console port on the device is configured as RS-232 data terminal equipment (DTE). The TOE's administrator can use the command-line interface (CLI) over this port to configure the device from a terminal. Further, the AGD section Sample Output for Local Administrative Session Termination includes a note which states that: Local console sessions display the banner FreeBSD/amd64 (hostname) (ttyu0) before the login prompt. This message is not shown during remote SSH access. Administrators should use this difference to verify whether the session is local or remote.

For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and the Guidance Documentation.

Evaluator Findings:
The TOE is not a distributed TOE; hence, this assurance activity is not applicable.

The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation.

Evaluator Findings:
The evaluator checked that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS row FMT_SMF.1 and the Guidance Documentation. This was verified throughout the testing process during the evaluation.

(If configure local audit is selected) The evaluator shall examine the TSS and Guidance Documentation to ensure that a description of the logging implementation is described in enough detail to determine how log files are maintained on the TOE.

Evaluator Findings:
(If configure local audit is selected) The evaluator examined the TSS row FMT_SMF.1 and Guidance Documentation section Configure Audit Log Options on Device in FIPS Mode to ensure that a description of the logging implementation is described in enough detail to determine how log files are maintained on the TOE. The relevant information is found in the following section: TOE Summary Specification row FMT_SMF.1 Upon investigation, the evaluator found that the TSS states that: The TOE logs all events and stores them in a local file using syslog configuration. For more details about the TOE's logging system (refer to TSS subsection: FAU_STG_EXT.1).

The relevant information is found in the following section of the AGD: **Configure Audit Log Options on Device in FIPS Mode**

Upon investigation, the evaluator found that the claimed AGD section states that:

To configure audit log options:

1. Specify the number of files to be archived in the system logging facility.

[edit system syslog]

security-administrator@host:fips# set archive files 2

2. Specify the file in which to log data.

[edit system syslog]

security-administrator@host:fips# set file syslog any any

3. Specify the size of files, in bytes, to be archived. The size can also be specified in shorthand form, for example, `10m` for 10 MB and `1g` for 1 GB.

[edit system syslog]

security-administrator@host:fips# set file syslog archive size 10000000

4. Specify the priority and facility in messages for the system logging facility.

[edit system syslog]

security-administrator@host:fips# set file syslog explicit-priority

5. Log system messages in a structured format.

[edit system syslog]

security-administrator@host:fips# set file syslog structured-data

6. Commit the changes:

[edit]

security-administrator@host:fips# commit

Verdict:

PASS.

5.1.4.6.2 FMT_SMF.1 AGD

See Section 5.1.4.6.1.

5.1.4.7 FMT_SMF.1/MACSEC SPECIFICATION OF MANAGEMENT FUNCTIONS (MACSEC) (MOD_MACSEC_V1.0)

5.1.4.7.1 FMT_SMF.1/MACSEC TSS

The evaluator shall verify that the TSS describes the ability of the TOE to provide the management functions defined in this SFR.

Evaluator Findings:

The evaluator reviewed the TSS row **FMT_SMF.1/MACSEC** and verified that the TSS describes the ability of the TOE to provide the management functions defined in this SFR.

The relevant information is found in the following section: **TOE Summary Specification** row **FMT_SMF.1/MACSEC**

Upon investigation, the evaluator found that the TSS states that:

The available management functions are listed below, and these can be accessed via the SSH remotely or locally via console:

- **Ability to generate a PSK and install it in the device**

- Ability to manage the Key Server to create, delete, and activate MKA participants using the CLI commands
- Ability to specify the lifetime of a CAK
- Ability to enable, disable, or delete a PSK-based CAK using CLI commands

Verdict:

PASS.

5.1.4.7.2 FMT_SMF.1/MACSEC AGD

The evaluator shall examine the operational guidance to determine that it provides instructions on how to perform each of the management functions defined in this SFR.

Evaluator Findings:

The evaluator reviewed the AGD section **Configuring MACsec with keychain using Layer 3 Traffic** and verified that the AGD provides instructions on how to perform each of the management functions defined in this SFR.

The relevant information is found in the following section(s):

Administrative Activity	Method (Command/GUI Configuration)	Section
Manage a PSK-based CAK and install it in the device	security-administrator@host:fips# prompt security macsec connectivity-association ca_name pre-shared-key cak New cak (secret): Retype new cak (secret):	Media Access Control Security (MACsec) in FIPS Mode Overview

Manage the key server to create, delete, and activate MKA participants	<pre> [edit] security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name exclude-protocol protocol-name security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name include-sci security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name mka key-server-priority priority-number security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name mka transmit-interval interval security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name offset 30 [edit] security-administrator@host:fips# prompt security macsec connectivity-association connectivity-association-name pre-shared- key cak New cak (secret): Retype new cak (secret): security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name pre-shared-key ckn hexadecimal-number security-administrator@host:fips# set security macsec connectivity- association connectivity-association-name replay-protect replay- window-size number-of-packets [edit] security-administrator@host:fips# set security macsec connectivity- association CA1 security-mode security-mode [edit] security-administrator@host:fips# set security macsec interfaces interface-name connectivity-association connectivity-association- name user@host# deactivate security macsec user@host# activate security macsec </pre>	Configuring MACsec on a Device Running Junos OS, Disable and Restart MACsec Sessions
Specify the lifetime of a CAK	<pre> [edit] security-administrator@host:fips# set security authentication-key- chains key-chain macsec-kc1 key 0 key-name 2345678922334455667788992223334445556667778889992223333 44445551 security-administrator@host:fips# set security authentication-key- chains key-chain macsec-kc1 key 0 start-time 2018-03-20.20:35 </pre>	Configuring MACsec with keychain using Layer 3 Traffic
Enable, disable, or delete a PSK-based CAK	<pre> security-administrator@host:fips# prompt security macsec connectivity-association ca_name pre-shared-key cak New cak (secret): Retype new cak (secret): security-administrator@host:fips# delete security macsec connectivity association ca_name pre-shared-key cak </pre>	Media Access Control Security (MACsec) in FIPS Mode Overview

Verdict:

PASS.

5.1.4.8 FMT_SMR.2 RESTRICTIONS ON SECURITY ROLES (CPP_ND_V3.0E)

5.1.4.8.1 FMT_SMR.2 TSS

The evaluator shall examine the TSS to determine that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE (e.g. if local administrators and remote administrators have different privileges or if several types of administrators with different privileges are supported by the TOE).

Evaluator Findings:
The evaluator examined the TSS row FMT_SMR.2 and determined that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE. The relevant information is found in the following section(s): TOE Summary Specification row FMT_SMR.2 Upon investigation, the evaluator found that the TSS states that: The TOE supports a Security Administrator role, administrable locally or remotely, responsible for provisioning user accounts. The TOE also supports a self-explanatory 'ready-only' role, which only has view permissions for the TOE configuration and operational state (show commands). Only Security Administrator accounts, associated with the "security-admin" login class and its requisite permissions, can manage the TOE. Administrators can manually create other custom login classes and allocate admin privileges to it.

Verdict:

PASS.

5.1.4.8.2 FMT_SMR.2 AGD

The evaluator shall review the Guidance Documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.

Evaluator Findings:
The evaluator reviewed the AGD sections 'Configure a Network Device Collaborative Protection Profile Authorized Administrator' and 'Configure SSH on the Evaluated Configuration' and ensured that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. The relevant information is found in the following section(s): Configure a Network Device Collaborative Protection Profile Authorized Administrator, Configure SSH on the Evaluated Configuration and Limit the Number of User Login Attempts for SSH Sessions Upon investigation, the evaluator found that the AGD section Configure a Network Device Collaborative Protection Profile Authorized Administrator states that: To configure an authorized administrator: 1. Create a login class named security-admin with all permissions. [edit] security-administrator@host:fips# set system login class security-admin permissions all

2. Configure the hashed algorithm SHA256 or SHA512 for plain-text passwords. SHA512 is the default hashing algorithm.

[edit]

```
security-administrator@host:fips# set system login password format sha512
```

3. Commit the changes.

[edit]

```
security-administrator@host:fips# commit
```

4. Define your NDcPPv3.0e user authorized administrator using the command below.

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin
authentication encrypted-password
or
```

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin
authentication plain-text-password
```

To configure public key authentication:

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin
authentication ssh ecdsa <ecdsa-public-key>
```

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin
authentication ssh rsa <rsa-public-key>
```

5. Set the log-key-changes configuration statement to log when SSH authentication keys are added or removed.

[edit]

```
security-administrator@host:fips# set system services ssh log-key-changes
```

NOTE: When the log-key-changes configuration statement is enabled and committed (with the commit command in configuration mode), Junos OS logs the changes to the set of authorized SSH keys for each user (including the keys that were added or removed). Junos OS logs the differences since the last time the log-key-changes configuration statement was enabled. If the log-key-changes configuration statement was never enabled, then Junos OS logs all the authorized SSH keys.

6. Commit the changes.

[edit]

```
security-administrator@host:fips# commit
```

Further, the evaluator found that the AGD section **Configure SSH on the Evaluated Configuration** states that:

To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa
```

NOTE: Although the last hostkey algorithm option mentioned above only mentions RSA, it covers all the claimed RSA algorithms, that is SSH-RSA, RSA-SHA2-256, and RSA-SHA2-512

2. Specify the SSH key-exchange for Elliptic Curve Diffie-Hellman keys for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256
```

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384
```

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521
```

3. Specify all the permissible message authentication code algorithms for SSHv2

[edit]

```
security-administrator@host:fips# set system services ssh macs hmac-sha2-256
```

```
security-administrator@host:fips# set system services ssh macs hmac-sha2-512
```

4. Specify the ciphers allowed for protocol version 2.

[edit]

```
security-administrator@host:fips# set system services ssh ciphers aes128-cbc
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-cbc
```

```
security-administrator@host:fips# set system services ssh ciphers aes128-ctr
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-ctr
```

```
security-administrator@host:fips# set system services ssh ciphers aes128-gcm@openssh.com
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-gcm@openssh.com
```

5. Commit the changes:

[edit]

```
security-administrator@host:fips# commit
```

In Addition, the section **Limit the Number of User Login Attempts for SSH Sessions** states that:

The SSH2 protocol provides secure terminal sessions utilizing the secure encryption. The SSH2 protocol enforces running the key-exchange phase and changing the encryption and integrity keys for the session. Key exchange is done periodically, after specified seconds or after specified bytes of data have passed over the connection. The TSF ensures that within the SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of the transmitted data. When either of the thresholds are reached, a rekey must be performed.

[edit system]

```
security-administrator@host:fips# set services ssh rekey time-limit number
```

Time limit before renegotiating session keys is 1 through 1440 minutes.

[edit system]

```
security-administrator@host:fips# set services ssh rekey data-limit number
```

Data limit before renegotiating session keys is 51200 through 4294967295 byte.

Verdict:

PASS.

5.1.5 PROTECTION OF THE TSF (FPT)

5.1.5.1 FPT_APW_EXT.1 PROTECTION OF ADMINISTRATOR PASSWORDS (CPP_ND_V3.0E)

5.1.5.1.1 FPT_APW_EXT.1 TSS

The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

Evaluator Findings:

The evaluator examined the TSS row **FPT_APW_EXT.1** to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS also detailed passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FPT_APW_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE stores all password authentication data in a secure directory that is not accessible to any TOE's user. Passwords are obscured from the user from both local and remote CLI interfaces. The TSF protects administrative passwords by ensuring they are never stored in plaintext. When a password is created or updated, the TSF processes it using a Modular Crypt Format (MCF), which incorporates a unique salt and a SHA256 or SHA512 hash algorithms.

Verdict:

PASS.

5.1.5.1.2 FPT_APW_EXT.1 AGD

None.

5.1.5.2 FPT_CAK_EXT.1 PROTECTION OF CAK DATA (MOD_MACSEC_V1.0)

5.1.5.2.1 FPT_CAK_EXT.1 TSS

The evaluator shall examine the TSS to determine that it details how CAKs are stored and that they are unable to be viewed through an interface designed specifically for that purpose. If these values are not stored in plaintext, the TSS shall describe how they are protected or obscured.

Evaluator Findings:

The evaluator reviewed the TSS row **FPT_CAK_EXT.1** and verified that the TSS details how CAKs are stored and that they are unable to be viewed through an interface designed specifically for that purpose.

The relevant information is found in the following section: **TOE Summary Specification** row **FPT_CAK_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE protects each CAK by wrapping it with AES-256 in GCM mode using a System Master Key derived from the System Master Password. The TOE stores all this information in a secure directory (config file) that is not accessible to administrators or other non-admin roles.

Verdict:

PASS.

5.1.5.2.2 FPT_CAK_EXT.1 AGD

There are no guidance EAs for this component.

5.1.5.3 FPT_FLS.1 FAILURE WITH PRESERVATION OF SECURE STATE

5.1.5.3.1 FPT_FLS.1 TSS [TD0816]

The evaluator shall examine the TSS to determine that it indicates that the TSF will attain a secure/safe state (e.g., shutdown) if a self-test failure is detected.

Evaluator Findings:

The evaluator reviewed the TSS row **FPT_FLS.1**, and verified that the TSS indicates that the TSF will attain a secure/safe state (e.g., shutdown) if a self-test failure is detected.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FPT_FLS.1**

Upon investigation, the evaluator found that the TSS states that:

If one of the self-tests fail, the device panics and reboots continuously. The TOE's administrator can recover the device by performing a fresh installation of Junos using a bootable USB drive created from a USB-compatible Junos image.

For TOEs with redundant failover capability, the evaluator shall examine the TSS to determine that it indicates that the failed components will attain a secure/safe state (e.g., shutdown) if a self-test failure is detected.

Evaluator Findings:

The TOE does not have a redundant failover capability; hence, this assurance activity is not applicable.

Verdict:

PASS.

5.1.5.3.2 FPT_FLS.1 AGD

The evaluator shall examine the operational guidance to verify that it describes the behavior of the TOE following a self-test failure and actions that an administrator should take if it occurs.

Evaluator Findings:

The evaluator reviewed the AGD section '**FIPS Self-Tests Overview**' and verified that the AGD describes the behavior of the TOE following a self-test failure and actions that an administrator should take if it occurs.

The relevant information is found in the following section: **FIPS Self-Tests Overview**

Upon investigation, the evaluator found that the AGD claimed section states that:

If one of the self-tests fail, the device panics and reboot continuously. The TOE's administrator can recover the device using USB install as follows:

1. **Download the Junos USB installation image and create a bootable USB drive.**
2. **Insert the bootable USB drive into the Routing Engine USB port.**
3. **Reboot the device.**
4. **Access the Boot Manager during startup and select the USB device as the boot source.**
5. **Allow the USB installation process to start and follow the prompts to complete the installation.**
6. **Reboot the device for the fresh Junos installation to take effect.**

The file show /var/log/messages command displays the system log.

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.5.4 FPT_RPL.1 REPLAY DETECTION

5.1.5.4.1 FPT_RPL.1 TSS [TD0881]

The evaluator shall examine the TSS to determine that it describes how replay is detected for MKPDUs and MPDUs and how replayed MKPDUs and MPDUs are handled by the TSF.

Evaluator Findings:
The evaluator reviewed the TSS row FPT_RPL.1 and verified that the TSS describes how replay is detected for MKPDUs and MPDUs and how replayed MKPDUs and MPDUs are handled by the TSF. The relevant information is found in the following section: TOE Summary Specification row FPT_RPL.1 Upon investigation, the evaluator found that the TSS states that: To protect against replay (within the Control Plane) each participant in the protocol chooses a random 96-bit member identifier (MI) when MKA begins, and this MI is used, together with a 32-bit message number (MN) initialized to 1 and incremented with each MKPDU transmitted. To prevent replay attacks, the TOE verifies that each incoming MKPDU contains a unique and incremented MN. Any MKPDU featuring a duplicate or outdated MN is discarded and an error message is logged immediately by the TSF. The Data Plane replay functionality ensures that a man-in-the middle cannot replay a snooped packet or reuse packet numbers. As bounded receive delay functionality is not supported, it is necessary to configure replay protection in the evaluated configuration using replay-protect. The replay-window-size specifies the number of packets which can be replayed. If set to zero this means no replays are permitted (and should not be used when out of ordering is expected).

Verdict:

PASS.

5.1.5.4.2 FPT_RPL.1 AGD

There are no guidance EAs for this component.

5.1.5.5 FPT_SKP_EXT.1 PROTECTION OF TSF DATA (FOR READING OF ALL SYMMETRIC KEYS) (CPP_ND_V3.0E)

5.1.5.5.1 FPT_SKP_EXT.1 TSS

The evaluator shall examine the TSS to determine that it details how any preshared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, by any enabled role, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

Evaluator Findings:

The evaluator examined the TSS row **FPT_SKP_EXT.1** and determined that it details how any pre- shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, by any enabled role, as outlined in the application note. If these values are not stored in plaintext, the TSS describes how they are protected/obscured.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FPT_SKP_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE does not provide a CLI interface to permit the viewing of keys. Cryptographic keys are protected through the enforcement of kernel-level file access rights, limiting access to the contents of cryptographic key containers to processes with cryptographic rights or shell users with root permission. Table 18 (ST) includes additional information on key storage and protection of non-plaintext keys.

Verdict:

PASS.

5.1.5.5.2 FPT_SKP_EXT.1 AGD

None.

5.1.5.6 FPT_STM_EXT.1 RELIABLE TIME STAMPS (CPP_ND_V3.0E)

5.1.5.6.1 FPT_STM_EXT.1 TSS

The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

Evaluator Findings:

The evaluator examined the TSS row **FPT_STM_EXT.1** and ensured that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FPT_STM_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The clock function of the TOE provides a source of date and time information for the appliance, used in audit timestamps, which is maintained using the hardware Time Stamp Counter as the clock source. Time on the TOE can be configured manually by the security administrator along with support for synchronization to NTP servers. Timestamps are also used for security relevant items such as rekey thresholds, inactivity timeouts and account lockouts.

If “obtain time from the underlying virtualization system” is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay.

Evaluator Findings:

The ST does not select obtain time from the underlying virtualization system hence this part of the objective is not applicable.

Verdict:

PASS.

5.1.5.6.2 FPT_STM_EXT.1 AGD

The evaluator shall examine the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

Evaluator Findings:

The evaluator examined the guidance documentation sections **Configuring Network Time Protocol** and **Customize Time in FIPS Mode** and ensured that it instructs the administrator how to set the time. The TOE supports the use of an NTP server and the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and configuration of the NTP client on the TOE to support this communication.

The relevant information is found in the following section(s): **Configuring Network Time Protocol** and **Customize Time in FIPS Mode**

Upon investigation, the evaluator found that the claimed AGD section **Configuring Network Time Protocol** states that:

The device can be configured to sync with a Network Time Protocol (NTP) server. It supports time updates using NTP version 4. The device authenticates updates using an administrator-configured SHA-1 or SHA-256 symmetric key. The device rejects broadcast and multicast time updates by default without requiring any configuration. It does not place a limit on the number of NTP time sources that can be configured.

To configure the device in client mode, include the server statement and other statements at the [edit system ntp] hierarchy level:

[edit system ntp]

security-administrator@hostname:fips# server address key key-number version 4 prefer

security-administrator@hostname:fips# authentication-key key-number type sha1|sha256 value <key value>

security-administrator@hostname:fips# trusted-key trusted <key-number>

The above commands specify the address of the system acting as the time server. One specifies an address, not a hostname. The same command can be used to configure multiple NTP servers.

It also configures the authentication key sent in all messages sent to the time server, using the key option. The key corresponds to the key number specified in the authentication-key statement. By default, the device sends NTP version 4 packets to the time server. If more than one time server is configured, one server can be marked as preferred by including the prefer option.

The following example shows how to configure the device to operate in client mode:

[edit system ntp]

security-administrator@hostname:fips# authentication-key 12 type sha256 value

"\$9\$TQFn/9t0OlcYwY4oGU9At"

security-administrator@hostname:fips# server 10.1.1.1 key 12 prefer

```
security-administrator@hostname:fips# trusted-key 12
```

By default, NTP operates in an entirely unauthenticated manner. If a malicious attempt to influence the accuracy of a router or switch's clock succeeds, it could have negative effects on system logging, make troubleshooting and intrusion detection more difficult, and impede other management functions.

For common criteria compliance, use trusted authentication using SHA1 or SHA256 as the message digest algorithm(s) to make sure that the NTP peer is trusted.

For IP version 4 (IPv4), one can specify the source address the TOE will use to access the network when it is sending NTP requests to the server.

To configure the specific source address that the request will always use, include the source-address statement at the [edit system ntp] hierarchy level. The source-address is a valid IP address configured on one of the router or switch interfaces.

```
[edit system ntp]
```

```
security-administrator@hostname:fips# set source-address source-address
```

Also, the AGD section **Customize Time in FIPS Mode** shows how to set time manually on the TOE:

To customize time, disable NTP and then set the date and time.

1. Disable NTP from the configuration mode.

```
[edit]
```

```
security-administrator@hostname:fips# deactivate groups global system ntp
```

2. Set the date and time from the operational mode. Date and time format is YYYYMMDDHHMM.ss.

```
security-administrator@hostname:fips> set date 201803202034.00
```

If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay.

Evaluator Findings:
The ST does not select obtain time from the underlying virtualization system since it is not a vND. Hence this part of the objective is not applicable.

Verdict:

PASS.

5.1.5.7 FPT_TST_EXT.1 TSF TESTING (CPP_ND_V3.0E)

5.1.5.7.1 FPT_TST_EXT.1 TSS [TD0836]

The evaluator shall examine the TSS to ensure that it details each of the self-tests that are identified by the SFR; 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).

Evaluator Findings:
The evaluator examined the TSS row FPT_TST_EXT.1 and ensured that it details each of the self-tests that are identified by the SFR; this description includes an outline of what the tests are actually doing The relevant information is found in the following section(s): TOE Summary Specification row FPT_TST_EXT.1 Upon investigation, the evaluator found that the TSS states that: The TOE runs the following set of self-tests during power on to check the correct operation of the Junos OS firmware: • Power on test – determines the boot-device responds, and performs a memory size check to confirm the amount of available memory. • File integrity test – verifies the integrity of all mounted signed packages, ensuring that system files have not been tampered with. To test the integrity of the firmware, the SHA256 fingerprints of executables and other immutable files are recalculated and verified using the ECDSA P-256 digital signature against the signed fingerprints contained in the manifest file. • Crypto integrity test – checks integrity of major CSPs, such as SSH hostkeys, passwords, and various keys. The test can be run on demand. This test calculates the SHA-256 hash of files in the TOE directory and compares each of them against a stored value in a file with the same file name and the changed extension. If any of the calculated hashes does not match, the test fails. This behaviour is the same across all platforms. • Simulated veriexec error – verifies that the veriexec file integrity enforcement framework is enabled and operates as expected using /opt/sbin/kats/cannot-exec.real, which triggers an intentional failure and verifies that it gets detected. • Crypto self-tests (Kernel, libmd, OpenSSL and MACsec libraries) for SSH and MACsec – verifies correct output from known answer tests for all cryptographic algorithms. • Noise Source Health Test - Noise source health tests - verify the correct operation of the noise source. Tests include a repetitive count test and an adaptive proportion test. Self-tests demonstrating the correct operation of the TSF on the TOE are performed during initial start-up (power-on) to verify the integrity of the TOE firmware and software. Self-tests are also performed prior to providing any cryptographic service to verify the correct operation of cryptographic implementations necessary to fulfill the TSF.

The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

Evaluator Findings:

The evaluator ensured that the TSS row **FPT_TST_EXT.1** makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FPT_TST_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

These NIST-approved self-tests cover all cryptographic primitives and security-critical TSF components and are executed before any security function is used, they provide a complete and reliable demonstration that the TSF is operating correctly. If one of the self-tests fail, the device panics and reboots continuously. The TOE's administrator can recover the device by performing a fresh installation of Junos using a bootable USB drive created from a USB-compatible Junos image.

If more than one failure response is listed in FPT_TST_EXT.1.2, the evaluator shall examine the TSS to ensure it clarifies which response is associated with which type of failure.

Evaluator Findings:

There is only one failure response is listed in FPT_TST_EXT.1.2, hence this activity is not applicable.

For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these self-tests are run. The evaluator shall also examine the TSS to ensure it describes how the TOE reacts if one or more TOE components fail self-testing (e.g. halting and displaying an error message; failover behaviour).

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:

PASS.

5.1.5.7.2 FPT_TST_EXT.1 AGD

The evaluator shall also ensure that the guidance documentation describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

Evaluator Findings:

The evaluator also ensured that the guidance documentation section '**FIPS Self-Tests Overview**' describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors correspond to those described in the TSS.

The relevant information is found in the following section: **FIPS Self-Tests Overview**

Upon investigation, the evaluator found that the AGD claimed section states that:

The cryptographic module enforces security rules to ensure that the Juniper Networks Junos operating system (Junos OS) in FIPS mode meets the security requirements of FIPS 140-3 Level 1. To validate the output of cryptographic algorithms approved for FIPS and test the integrity of some system modules, the device performs the following series of known answer test (KAT) self-tests:

- **kernel_kats**—KAT for kernel cryptographic routines
- **macsec_kats**—KAT for MACsec cryptographic implementation

- md_kats—KAT for libmd and libc
- openssl_kats—KAT for OpenSSL cryptographic implementation

NOTE: For MACsec qualifications, the LC KATs has to perform on AES-GCM-128 and AES-GCM-256.

The device automatically performs the KAT self-tests at startup. The device also performs the conditional self-tests automatically to verify digitally signed software packages, generated random numbers, RSA and ECDSA key pairs, and manually entered keys.

When the device successfully completes the KATs, it updates the log (syslog) file to display the executed tests. During bootup entropy health test is performed. The system software integrity check is done during bootup. Each software module is verified before being loaded. The messages pertaining to the entropy health check and software integrity checks can be viewed on the console during system bootup.

If one of the self-tests fail, the device panics and reboot continuously. The TOE's administrator can recover the device using USB install as follows:

1. Download the Junos USB installation image and create a bootable USB drive.
2. Insert the bootable USB drive into the Routing Engine USB port.
3. Reboot the device.
4. Access the Boot Manager during startup and select the USB device as the boot source.
5. Allow the USB installation process to start and follow the prompts to complete the installation.
6. Reboot the device for the fresh Junos installation to take effect.

The file `show /var/log/messages` command displays the system log.

Based on these findings, this assurance activity is considered satisfied.

For distributed TOEs the evaluator shall ensure that the guidance documentation describes how to determine from an error message returned which TOE component has failed the self-test.

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:

PASS.

5.1.5.8 FPT_TUD_EXT.1 TRUSTED UPDATE (CPP_ND_V3.0E)

5.1.5.8.1 FPT_TUD_EXT.1 TSS

The evaluator shall verify that the TSS describe how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS shall describe how and when the inactive version becomes active. The evaluator shall verify this description.

Evaluator Findings:

The evaluator verified that the TSS row **FPT_TUD_EXT.1** describes how to query the currently active version. A trusted update cannot be installed on the TOE with a delayed activation. The evaluator verified this description.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FPT_TUD_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

Security Administrators are able to query the current version of the TOE firmware using the CLI command "show version". The TOE does not support delayed activation or partial updates for the TOE.

The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware and software). The evaluator shall verify that the description includes a digital signature verification of the software of the software before installation and that installation fails if the verification fails.

The evaluator shall verify that the TSS describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification.

Evaluator Findings:

The evaluator verified that the TSS row **FPT_TUD_EXT.1** describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware and software). The evaluator verified that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. Also the TSS describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FPT_TUD_EXT.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE does not support delayed activation or partial updates for the TOE. If a new version of the TOE firmware is available an update of the TOE firmware can be initiated. Updates are downloaded and applied manually.

The installable firmware package containing the TOE firmware has a digital signature that is checked when the Security Administrator attempts to install the package. The TOE uses Juniper "Veriexec" to verify the package integrity. The hash signature is verified using public key during package installation.

If the image fails the signature check, then the image is deleted from the device and no upgrade occurs, with the appropriate error visible on the CLI. Successful update is only initiated when the signature verification of the TOE passes. Logs are generated in both cases. The TOE does not support automatic updates.

If the options 'support automatic checking for updates' or 'support automatic updates' are chosen from the selection in FPT_TUD_EXT.1.2, the evaluator shall verify that the TSS explains what actions are involved in automatic checking or automatic updating by the TOE, respectively.

Evaluator Findings:

The ST does not select 'support automatic checking for updates' or 'support automatic updates' for FPT_TUD_EXT.1.2 hence this part of the objective is not applicable.

For distributed TOEs, the evaluator shall examine the TSS to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. Alternatively, this description can be provided in the guidance documentation. In that case the evaluator should examine the guidance documentation instead.

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

Verdict:

PASS.

5.1.5.8.2 FPT_TUD_EXT.1 AGD

The evaluator shall verify that the guidance documentation describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the guidance documentation needs to describe how to query the loaded but inactive version.

Evaluator Findings:

The evaluator verified that the guidance documentation section **Install Junos OS Software Packages** describes how to query the currently active version. The evaluator also confirmed that the TOE does not support delayed activation of updates and hence the active version is the only version that is queried.

The relevant information is found in the following section: **Install Junos OS Software Packages**

Upon investigation, the evaluator found that the AGD states that:

Once the system completes rebooting, log in and use the show version command to verify the successful installation of the new software version.

It also states that:

Junos OS does not support delayed activation while performing software updates.

The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

Evaluator Findings:

The evaluator verified that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification). The description includes the procedures for successful and unsuccessful verification. The description corresponds to the description in the TSS.

The relevant information is found in the following section: **Install Junos OS Software Packages**

Upon investigation, the evaluator found that the AGD activity states that:

NOTE: Junos OS is delivered in signed packages with manifests which validate the immutability of the contents. There is also a signature of the manifest that can be validated with public keys that are packaged. During install of the software packages, Junos OS validates the signature and the hashes of the contents present in the manifest. If the image fails the signature check, then the image is deleted from the device and no upgrade occurs, with the appropriate error visible on the CLI. Successful update is only initiated when the signature verification of the TOE passes. Logs are generated in both cases. Software update performed in FIPS mode is similar to that in non-FIPS mode for the functionalities available on the device.

For distributed TOEs the evaluator shall verify that the guidance documentation describes how the versions of individual TOE components are determined for FPT_TUD_EXT.1, how all TOE components are updated, and the error conditions that may arise from checking or applying the update (e.g. failure of signature verification, or exceeding available storage space) along with appropriate recovery actions. The guidance documentation only has to describe the procedures relevant for the Security Administrator; it does not need to give information about the internal communication that takes place when applying updates.

Evaluator Findings:

The TOE is not a distributed TOE hence this assurance activity is not applicable.

If this information was not provided in the TSS: For distributed TOEs, the evaluator shall examine the Guidance Documentation to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component.

Evaluator Findings:
The TOE is not a distributed TOE hence this assurance activity is not applicable.

If this information was not provided in the TSS: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the certificates are contained on the device. The evaluator shall also ensure that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

Evaluator Findings:
The ST does not select a certificate-based mechanism for software update digital signature verification hence this assurance activity is not applicable.

Verdict:

PASS.

5.1.6 TOE ACCESS (FTA)

5.1.6.1 FTA_SSL_EXT.1 TSF-INITIATED SESSION LOCKING (CPP_ND_V3.0E)

5.1.6.1.1 FTA_SSL_EXT.1 TSS

The evaluator shall examine the TSS to determine that it details whether local administrative session locking or termination is supported and the related inactivity time period settings.

Evaluator Findings:
The evaluator examined the TSS row FTA_SSL_EXT.1 and determined that it details the administrative local session termination and the related inactivity time period. The relevant information is found in the following section(s): TOE Summary Specification row FTA_SSL_EXT.1 Upon investigation, the evaluator found that the TSS states that: User sessions, both remote and local, with the TOE can also be terminated due to inactivity (time threshold). Security Administrators can configure this inactivity timer on user sessions. This setting is controlled by the idle-timeout parameter, which ranges from 0 to 4,294,967,295 minutes. Administrators can configure this inactivity timer via the CLI. Once the set inactivity period is crossed the user would be logged out from the TOE and the session would be terminated. For each user session, the TOE maintains a count of clock cycles (provided by the system clock) since the last user activity. The count is reset each time the user performs any activity within the session. When the counter reaches the number of clock cycles equating to the configured period of inactivity, the user session is terminated.

Verdict:

PASS.

5.1.6.1.2 FTA_SSL_EXT.1 AGD

The evaluator shall confirm that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

Evaluator Findings:

The evaluator confirmed that the guidance documentation section '**Configure Session Termination**' states local administrative session termination is supported and instructions for configuring the inactivity time period.

The relevant information is found in the following section: **Configure Session Termination**

Upon investigation, the evaluator found that the AGD section **Configure Session Termination** states that:

Terminate the session after the security administrator specifies inactive timeout period. The inactive timeout period applies to both local console sessions as well as the remote SSH sessions.

It also provides the configuration step below to set the idle timeout for the security-admin role:

[edit]

security-administrator@host:fps# set system login class security-admin idle-timeout 2

Specify the idle timeout period in minutes. The configurable idle timeout period range is 0 through 4294967295 minutes.

Verdict:

PASS.

5.1.6.2 FTA_SSL.3 TSF-INITIATED TERMINATION (CPP_ND_V3.0E)

5.1.6.2.1 FTA_SSL.3 TSS

The evaluator shall examine the TSS to determine that it details the administrative remote session termination and the related inactivity time period.

Evaluator Findings:

The evaluator examined the TSS row **FTA_SSL.3** and determined that it details how the remote administrative session (and if applicable the local administrative session) are terminated.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FTA_SSL.3**

Upon investigation, the evaluator found that the TSS states that:

User sessions, both remote and local, with the TOE can also be terminated due to inactivity (time threshold).

Security Administrators can configure this inactivity timer on user sessions. This setting is controlled by the idle-timeout parameter, which ranges from 0 to 4,294,967,295 minutes. Administrators can configure this inactivity timer via the CLI. Once the set inactivity period is crossed the user would be logged out from the TOE and the session would be terminated. For each user session, the TOE maintains a count of clock cycles (provided by the system clock) since the last user activity. The count is reset each time the user performs any activity within the session. When the counter reaches the number of clock cycles equating to the configured period of inactivity, the user session is terminated.

Verdict:

PASS.

5.1.6.2.2 FTA_SSL.3 AGD

The evaluator shall confirm that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination.

Evaluator Findings:

The evaluator confirmed that the guidance documentation section '**Configure Session Termination**' includes instructions for configuring the inactivity time period for remote administrative session termination.

The relevant information is found in the following section: **Configure Session Termination**

Upon investigation, the evaluator found that the AGD section **Configure Session Termination** states that: **Terminate the session after the security administrator specifies inactive timeout period. The inactive timeout period applies to both local console sessions as well as the remote SSH sessions.**

It also provides the configuration step below to set the idle timeout for the security-admin role:

[edit]

security-administrator@host:fps# set system login class security-admin idle-timeout 2

Specify the idle timeout period in minutes. The configurable idle timeout period range is 0 through 4294967295 minutes.

Verdict:

PASS.

5.1.6.3 FTA_SSL.4 USER-INITIATED TERMINATION (CPP_ND_V3.0E)

5.1.6.3.1 FTA_SSL.4 TSS

The evaluator shall examine the TSS to determine that it details how the remote administrative session (and if applicable the local administrative session) are terminated.

Evaluator Findings:

The evaluator examined the TSS row **FTA_SSL.4** and determined that it details how the remote administrative session (and if applicable the local administrative session) are terminated.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FTA_SSL.4**

Upon investigation, the evaluator found that the TSS states that:

User sessions, both remote and local, with the TOE can be terminated by users. The administrative user can logout from the TOE using CLI by typing "exit" in operational mode to terminate the session.

Verdict:

PASS.

5.1.6.3.2 FTA_SSL.4 AGD

The evaluator shall confirm that the guidance documentation states how to terminate a remote interactive session (and if applicable the local administrative session).

Evaluator Findings:

The evaluator confirmed that the guidance documentation section, '**Sample Output for User Initiated Termination**' states how to terminate a remote and local interactive session.

The relevant information is found in the following section(s): **Configure Security Admin Access** and **Sample Output for User Initiated Termination**

The evaluator examined the AGD section **Sample Output for User Initiated Termination** which states that:

Local console session and the remote SSH session users can use this method to terminate the session.

This is a LOGIN message for MX 240.

Authorized users only !!!

FreeBSD/amd64 (MX240) (ttyu0)

login: admin

Password:

Last login: Fri Feb 20 09:11:31 from 10.1.2.146

--- JUNOS 24.4R2-S2.2 Kernel 64-bit JNPR-15.0-20251006.0b8e28c_buil

admin@MX240:fips>

admin@MX240:fips>

admin@MX240:fips> exit

acumensec@TestVM3:~\$

acumensec@TestVM3:~\$

acumensec@TestVM3:~\$ ssh admin@10.1.2.80

This is a LOGIN message for MX 240.

Authorized users only !!!

Password:

Last login: Fri Feb 20 09:11:21 2026 from 10.1.2.146

--- JUNOS 24.4R2-S2.2 Kernel 64-bit JNPR-15.0-20251006.0b8e28c_buil

admin@MX240:fips>

admin@MX240:fips>

admin@MX240:fips> exit

Connection to 10.1.2.80 closed.

acumensec@TestVM3:~\$

acumensec@TestVM3:~\$

acumensec@TestVM3:~\$

Verdict:

PASS.

5.1.6.4 FTA_TAB.1 DEFAULT TOE ACCESS BANNERS (CPP_ND_V3.0E)

5.1.6.4.1 FTA_TAB.1 TSS

The evaluator shall check the TSS to ensure that it details each administrative method of access (local and/or remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS).

Evaluator Findings:

The evaluator checked the TSS row **FTA_TAB.1** and ensured that it details each administrative method of access (local and remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS).

The relevant information is found in the following section(s): **TOE Summary Specification** row **FTA_TAB.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE can be accessed for administrative purposes either locally via console (CLI) or remotely using SSH (CLI).

The evaluator shall check the TSS to ensure that all administrative methods of access available to the Security Administrator are listed and that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access. The advisory notice and the consent warning message might be different for different administrative methods of access and might be configured during initial configuration (e.g. via configuration file).

Evaluator Findings:

The evaluator checked the TSS row **FTA_TAB.1** and ensured that all administrative methods of access available to the Security Administrator are listed and that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access. The advisory notice and the consent warning message might be different for different administrative methods of access and might be configured during initial configuration (e.g. via configuration file).

The relevant information is found in the following section(s): **TOE Summary Specification** section row **FTA_TAB.1**

Upon investigation, the evaluator found that the TSS states that:

For both local and remote methods of access, the TOE provides an access banner prior to login, with the authentication prompt. The banner can be configured by the security administrator for both local and remote login methods. The banner can provide warnings against unauthorized access to the secure TOE as well as any other information that the Security Administrator wishes to communicate.

Verdict:

PASS.

5.1.6.4.2 FTA_TAB.1 AGD

The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

Evaluator Findings:

The evaluator examined the guidance documentation section '**Configure System Login Message and Announcement**' and ensured that it describes how to configure the banner message.

The relevant information is found in the following section: **Configure System Login Message and Announcement**

Upon investigation, the evaluator found that the claimed AGD section states that:

A system login message appears before the user logs in and a system login announcement appears after the user logs in. By default, no login message or announcement is displayed on the device.

To configure a system login message for local console and SSH management interfaces, use the following command:

[edit]

security-administrator@host:fips# set system login message login-message-banner-text

To configure system announcement, use the following command:

[edit]

security-administrator@host:fips# set system login announcement system-announcement-text

Commit the configuration:

[edit]

security-administrator@host:fips# commit

NOTE:

- If the message text contains any spaces, enclose it in quotation marks.
- You can format the message using the following special characters:
 - \n—New line
 - \t—Horizontal tab
 - \'—Single quotation mark
 - \"—Double quotation mark
 - \\—Backslash

Based on these findings, this assurance activity is considered satisfied.

Verdict:

PASS.

5.1.7 TRUSTED PATH/CHANNELS (FTP)

5.1.7.1 FTP_ITC.1 INTER-TSF TRUSTED CHANNEL (CPP_ND_V3.0E)

5.1.7.1.1 FTP_ITC.1 TSS

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each secure communication mechanism is identified in terms of the allowed protocols for that IT entity, whether the TOE acts as a server or a client, and the method of assured identification of the non-TSF endpoint.

Evaluator Findings:

The evaluator examined the TSS row **FTP_ITC.1** and determined that, for all communications with authorized IT entities identified in the requirement, each secure communication mechanism is identified in terms of the allowed protocols for that IT entity, whether the TOE acts as a server, and the method of assured identification of the non-TSF endpoint.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FTP_ITC.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports SSHv2 via public key authentication for trusted channel implementation to syslog server. The TOE provides trusted channels using SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server (syslog server). The TOE acts as an SSH server when connecting with the configured syslog server.

Export of audit information to a secure, remote server is achieved by setting up an event trace monitor that sends event log messages by using NETCONF over SSH to the remote system event logging server. The SSHv2 protocol ensures that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.

The evaluator shall also confirm that all secure communication mechanisms are described in sufficient detail to allow the evaluator to match them to the cryptographic protocol Security Functional Requirements listed in the ST.

Evaluator Findings:

The evaluator also confirmed that all secure communication mechanisms are described in sufficient detail to allow the evaluator to match them to the cryptographic protocol Security Functional Requirements listed in the ST.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FTP_ITC.1**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports SSHv2 via public key authentication for trusted channel implementation to syslog server. The TOE provides trusted channels using SSHv2 protocol which ensures the confidentiality and integrity of communication with the remote audit server (syslog server). The TOE acts as an SSH server when connecting with the configured syslog server.

The SSHv2 protocol ensures that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.

The quoted TSS text maps the trusted channel to the FCS_SSH_EXT.1 and FCS_SSHS_EXT.1 SFRs in the ST.

Verdict:

PASS.

5.1.7.1.2 FTP_ITC.1 AGD

The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

Evaluator Findings:

The evaluator confirmed that the AGD sections '**Sample Syslog Server Configuration on a Linux System**' contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

The relevant information is found in the following section: **Sample Syslog Server Configuration on a Linux System**

Upon investigation, the evaluator found that the AGD section **Sample Syslog Server Configuration on a Linux System** states that:

Use the configuration details and establish a session between the target of evaluation (TOE) and the audit server. Examine the traffic that passes between the audit server and the TOE during several activities, and the generated audit data to be transferred to the audit server.

To configure event logging to a remote server when the SSH connection to the TOE is initiated from the remote system log server.

1. Generate an RSA public key on the remote syslog server. The TOE's administrator can also use ECDSA keys on the syslog server.

```
$ ssh-keygen -b 2048 -t rsa -C 'syslog-monitor key pair' -f ~/.ssh/syslog-monitor
```

The TOE's administrator will be prompted to enter the desired pass phrase. The storage locations for the syslog-monitor key pair is displayed.

2. On the TOE, create a class named monitor that has permission to trace events.

```
[edit system login]
```

```
security-administrator@host:fips# set class monitor permissions trace
```

3. Create a user named syslog-mon with the class monitor, and with public key-based or password-based authentication.

For public key-based authentication: [

```
edit system login]
```

```
security-administrator@host:fips# set user syslog-mon class monitor authentication ssh-rsa "public-key"
```

For password-based authentication:

```
[edit system login]
```

```
security-administrator@host:fips# set user syslog-mon class monitor authentication (plaintext-password | encrypted-password)
```

4. Set up NETCONF with SSH.

```
[edit system services]
```

```
security-administrator@host:fips# set netconf ssh
```

5. Configure syslog to log all the messages at /var/log/file-name.

```
[edit system]
```

```
security-administrator@host:fips# set syslog file file-name any any  
commit
```

6. To configure syslog in year format:

```
[edit system]
```

```
security-administrator@host:fips# syslog time-format year  
commit
```

7. On the remote system log server, start up the SSH agent ssh-agent. The start up is required to simplify the handling of the syslog-monitor key.

```
$ eval `ssh-agent -s`
```

8. On the remote syslog server, add the syslog-monitor key pair to the ssh-agent.

```
$ ssh-add ~/.ssh/syslog-monitor
```

The TOE's administrator will be prompted to enter the desired passphrase. Enter the same passphrase used in Step 1.

9. After logging in to the external_syslog_server session, establish a tunnel to the device and start NETCONF.

```
security-administrator@host:fips# $ssh syslog-mon@NDcPP_TOE -s netconf > test.out
```

10. If the session becomes disconnected, first verify and restore physical and network connectivity with the syslog server. Then, repeat the command in Step 9 to re-establish the NETCONF connection.

11. After NETCONF is established, configure a system log events message stream. This RPC will cause the NETCONF service to start transmitting messages over the SSH connection that is established.

```
<rpc><get-syslog-events><stream>messages</stream></get-syslog-events></rpc>
```

12. Monitor the event log generated for admin actions on TOE are received on syslog server. Examine the traffic that passes between the audit server and the TOE, observing that these data are not viewed during this transfer, and that they are successfully received by the audit server. Match the logs between local event logging and remote event logged in syslog server and record the particular software (name, version) used on the audit server during testing.

Verdict:

PASS.

5.1.7.2 FTP_ITC.1/MACSEC INTER-TSF TRUSTED CHANNEL (MACSEC COMMUNICATIONS)

5.1.7.2.1 FTP_ITC.1/MACSEC

This SFR is addressed through evaluation of FCS_MACSEC_EXT.1 through FCS_MACSEC_EXT.4.

5.1.7.3 FTP_TRP.1/ADMIN TRUSTED PATH (CPP_ND_V3.0E)

5.1.7.3.1 FTP_TRP.1/ADMIN TSS

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

Evaluator Findings:

The evaluator examined the TSS row **FTP_TRP.1/Admin** and determined that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator also examined the TSS section row and confirmed that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

The relevant information is found in the following section(s): **TOE Summary Specification** row **FTP_TRP.1/Admin**

Upon investigation, the evaluator found that the TSS states that:

The TOE supports remote administration using SSH. A trusted path is provided by the TOE using SSHv2 protocol between itself and remote administrators so that the contents of administrative sessions are protected against unauthorized disclosure or modification. Remote administrators can initiate communication to the TOE's CLI through the SSH tunnel created by the SSH session. Assured identification is guaranteed by using password or public key-based authentication mechanisms. The SSHv2 protocol ensures

that the data transmitted over a SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS cryptographic module.

Verdict:

PASS.

5.1.7.3.2 FTP_TRP.1/ADMIN AGD

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

Evaluator Findings:

The evaluator confirmed that the guidance documentation sections '**Configure a Network Device Collaborative Protection Profile Authorized Administrator**' and '**Configure SSH on the Evaluated Configuration**' contains instructions for establishing the remote administrative sessions for each supported method.

The relevant information is found in the following section(s): **Configure a Network Device Collaborative Protection Profile Authorized Administrator** and **Configure SSH on the Evaluated Configuration**

Upon investigation, the evaluator found that the AGD section 'Configure a Network Device Collaborative Protection Profile Authorized Administrator' states that:

An account for root is always present in a configuration and is not intended for use in normal operation. In the evaluated configuration, the root account is restricted to the initial installation and configuration of the evaluated device.

An NDcPPv3.0e authorized administrator must have all permissions, including the ability to change the device configuration.

To configure an authorized administrator:

1. Create a login class named security-admin with all permissions.

[edit]

```
security-administrator@host:fips# set system login class security-admin permissions all
```

2. Configure the hashed algorithm S HA256 or SHA512 for plain-text passwords. SHA512 is the default hashing algorithm.

[edit]

```
security-administrator@host:fips# set system login password format sha512
```

3. Commit the changes.

[edit]

```
security-administrator@host:fips# commit
```

4. Define your NDcPPv3.0e user authorized administrator using the command below. The same command is used to change the password.

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication encrypted-password
```

or

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication plain-text-password
```

To configure public key authentication:

[edit]

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh ecdsa <ecdsa-public-key>
```

```
security-administrator@host:fips# set system login user NDcPPv3-user class security-admin authentication ssh rsa <rsa-public-key>
```

5. Set the log-key-changes configuration statement to log when SSH authentication keys are added or removed.

[edit]

```
security-administrator@host:fips# set system services ssh log-key-changes
```

NOTE: When the log-key-changes configuration statement is enabled and committed (with the commit command in configuration mode), Junos OS logs the changes to the set of authorized SSH keys for each user (including the keys that were added or removed). Junos OS logs the differences since the last time the log-key-changes configuration statement was enabled. If the log-key-changes configuration statement was never enabled, then Junos OS logs all the authorized SSH keys.

6. Commit the changes.

[edit]

```
security-administrator@host:fips# commit
```

Further, the evaluator found that the AGD section 'Configure SSH on the Evaluated Configuration' states that:
To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521
```

```
security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa
```

NOTE: Although the last hostkey algorithm option mentioned above only mentions RSA, it covers all the claimed RSA algorithms, that is SSH-RSA, RSA-SHA2-256, and RSA-SHA2-512

2. Specify the SSH key-exchange for Elliptic Curve Diffie-Hellman keys for the system services.

[edit]

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256
```

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384
```

```
security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521
```

3. Specify all the permissible message authentication code algorithms for SSHv2

[edit]

```
security-administrator@host:fips# set system services ssh macs hmac-sha2-256
```

```
security-administrator@host:fips# set system services ssh macs hmac-sha2-512
```

4. Specify the ciphers allowed for protocol version 2.

[edit]

```
security-administrator@host:fips# set system services ssh ciphers aes128-cbc
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-cbc
```

```
security-administrator@host:fips# set system services ssh ciphers aes128-ctr
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-ctr
```

```
security-administrator@host:fips# set system services ssh ciphers aes128-gcm@openssh.com
```

```
security-administrator@host:fips# set system services ssh ciphers aes256-gcm@openssh.com
```

5. Commit the changes:

[edit]

```
security-administrator@host:fips# commit
```

Verdict:

PASS.

6 SECURITY ASSURANCE REQUIREMENTS

6.1 ASE: SECURITY TARGET EVALUATION

6.1.1 GENERAL EVALUATION ACTIVITIES FOR TOE SUMMARY SPECIFICATION (ASE_TSS.1) FOR ALL TOES (CPP_ND_V3.0E)

6.1.1.1 EVALUATION ACTIVITY

When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator shall ensure the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).

Evaluator Findings:
The evaluator performed the work units as presented in the CEM. Furthermore, the evaluator examined the TTS and ensured the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).

Verdict:

PASS.

6.2 ADV: DEVELOPMENT

6.2.1 BASIC FUNCTIONAL SPECIFICATION (ADV_FSP.1) (CPP_ND_V3.0E)

6.2.1.1 EVALUATION ACTIVITY

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Explicitly labeling TSFI as security relevant or non-security relevant is not necessary. A TSFI is implicitly security relevant if it is used to satisfy an evaluation activity, or if it is identified in the ST or guidance documentation as adhering to the security policies (as presented in the SFRs). The intent is that these interfaces will be adequately tested and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied. According to the description above 'security relevant' corresponds to the combination of 'SFRenforcing' and 'SFR-supporting' as defined in CC Part 3, paragraph 224 and 225.

The set of TSFI that are provided as evaluation evidence are contained in the Security Target and the guidance documentation.

Evaluator Findings:
TOE Design information that can be made public is available in the guidance documentation and in the ST. Any sensitive or proprietary information required by this protection profile is not to be made public.
It is not necessary to provide a complete specification of the TSFIs. For NDcPP, additional “functional specification” documentation is not necessary because this requirement is satisfied by multiple other documents (AGD, TSS, and Testing). All associated activities are covered in the Test Report, ST, and AGD documents.
NDcPP3.0e, section 7.2.1 states that: “For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation.”
All of the above information is applicable to the ADV Evaluation Activities (5.2.1.1, 5.2.1.2, and 5.2.1.3) in NDcPP3.0e-SD.

The evaluator examined the ST (Security Target) and the AGD (interface documentation) to verify that it describes the purpose and method of use for each TSFI that is identified as being security relevant. The evaluator examined the entire AGD. The evaluator verified the AGD describes the purpose and method of use for each security relevant TSFI by verifying the AGD satisfies all the AGD Evaluation Activities.

During testing, the evaluator used the product and its interfaces extensively and did not find any areas that were deficient.

Verdict:

PASS.

6.2.1.2 EVALUATION ACTIVITY

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

Evaluator Findings:

The evaluator checked the interface documentation (AGD) and ensured it identifies and describes the parameters for each TSFI that is identified as being security relevant. This is covered in the previous evaluation activity above.

Verdict:

PASS.

6.2.1.3 EVALUATION ACTIVITY

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator shall use the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces.

It should be noted that there may be some SFRs that do not have a TSFI that is explicitly “mapped” to invoke the desired functionality. For example, generating a random bit string or destroying a cryptographic key that is no longer needed are capabilities that may be specified in SFRs, but are not invoked by an interface.

The required EAs define the design and interface information required to meet ADV_FSP.1. If the evaluator is unable to perform some EA, then the evaluator shall conclude that an adequate functional specification has not been provided.

Evaluator Findings:

The evaluator examined the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator used the provided documentation to first identify, and then examine a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces.

This is covered in the previous evaluation activity above.

Verdict:

PASS.

6.3 AGD: GUIDANCE DOCUMENTATION

6.3.1 OPERATIONAL USER GUIDANCE (AGD_OPE.1) (CPP_ND_V3.0E)

6.3.1.1 EVALUATION ACTIVITY

The evaluator shall ensure the Operational guidance documentation is distributed to Security Administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that Security Administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

Evaluator Findings:

The evaluator checked the requirements above are met by the AGD. The AGD is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration. Upon investigation, the evaluator found that the CC guidance will be published with the CC certificate on www.niap-ccevs.org.

Verdict:

PASS.

6.3.1.2 EVALUATION ACTIVITY

The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Evaluator Findings:

The evaluator ensured that the AGD is provided for every Operational Environment that the product supports as claimed in the Security Target. The section titled '**Supported Platforms**' of the AGD was used to determine the verdict of this assurance activity. The AGD specifies that the platforms supported are:

For the features described in this document, the following platforms are supported to qualify NDcPPv2.2e:

Model	RE	RE CPU	MACsec Linecard	MIC-3D-10GESGP-E within MPC3E-3D-NG
MX240	RE-S-X6-128G RE-S-X6-64G	Intel HASWELL-EP Xeon E5-2608L V3	MPC10-10C and MPC10E-15C, with JNPR Penta Silicon MACsec ASIC	BMC82756 MACsec ASIC with CAVP num AES4550
MX480				
MX960				
EX9204	EX9200-RE2	Intel HASWELL-EP Xeon E5-2608L V3	EX9200-15C, with JNPR Penta Silicon MACsec ASIC	NA
EX9208				
EX9214				

Verdict:

PASS.

6.3.1.3 EVALUATION ACTIVITY

The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.

Evaluator Findings:

The evaluator ensured that the AGD section **Supported Cryptographic Algorithms** contains instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It provides a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.

The AGD states that:

WARNING: For FIPS 140-3 compliance, use only FIPS-approved cryptographic algorithms in Junos OS in FIPS mode. The Cryptographic Algorithms referred to in Chapters "Configure SSH and Console Connection" and "Configure MACsec" have been tested for Common Criteria Evaluation of the device. Use of other cryptographic implementations was not evaluated nor tested.

Verdict:

PASS.

6.3.1.4 EVALUATION ACTIVITY

The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

Evaluator Findings:

The entire AGD was used to determine the verdict of this work unit. Each confirmation command indicates tested options. Additionally, the section titled '**Security Functions Provided by the TOE**' specifies features that are not assessed and tested by the EAs. The evaluator ensured the AGD makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

Verdict:

PASS.

6.3.1.5 EVALUATION ACTIVITY

In addition, the evaluator shall ensure that the following requirements are also met:

- The guidance documentation shall contain instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.
- The evaluator shall verify that this process includes 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).
- The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

Evaluator Findings:

The evaluator verified the AGD contains instructions for configuring any cryptographic implementations in section '**Configuring SSH on the Evaluated Configuration**'.

The claimed AGD section states that:

To configure SSH on the TOE:

1. Specify the permissible SSH host-key algorithms for the system services.

[edit]

security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp256

security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp384

security-administrator@host:fips# set system services ssh hostkey-algorithm-list ecdsa-sha2-nistp521

security-administrator@host:fips# set system services ssh hostkey-algorithm-list rsa

NOTE: Although the last hostkey algorithm option mentioned above only mentions RSA, it covers all the claimed RSA algorithms, that is SSH-RSA, RSA-SHA2-256, and RSA-SHA2-512.

2. Specify the SSH key-exchange for Elliptic Curve Diffie-Hellman keys for the system services.

[edit]

security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp256

security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp384

security-administrator@host:fips# set system services ssh key-exchange ecdh-sha2-nistp521

3. Specify all the permissible message authentication code algorithms for SSHv2

[edit]

security-administrator@host:fips# set system services ssh macs hmac-sha2-256

security-administrator@host:fips# set system services ssh macs hmac-sha2-512

4. Specify the ciphers allowed for protocol version 2.

[edit]

security-administrator@host:fips# set system services ssh ciphers aes128-cbc

security-administrator@host:fips# set system services ssh ciphers aes256-cbc

security-administrator@host:fips# set system services ssh ciphers aes128-ctr

security-administrator@host:fips# set system services ssh ciphers aes256-ctr

security-administrator@host:fips# set system services ssh ciphers aes128-gcm@openssh.com

security-administrator@host:fips# set system services ssh ciphers aes256-gcm@openssh.com

5. Commit the changes:

[edit]

security-administrator@host:fips# commit

Supported SSH host-key algorithm:

ecdsa-sha2-nistp256

ecdsa-sha2-nistp384

ecdsa-sha2-nistp521

rsa

Supported SSH key-exchange algorithm:

ecdh-sha2-nistp256 The EC Diffie-Hellman on nistp256 with SHA2-256

ecdh-sha2-nistp384 The EC Diffie-Hellman on nistp384 with SHA2-384

ecdh-sha2-nistp521 The EC Diffie-Hellman on nistp521 with SHA2-512

Supported MACs algorithm:

hmac-sha2-256 Hash-based MAC using Secure Hash Algorithm (SHA2)

hmac-sha2-512 Hash-based MAC using Secure Hash Algorithm (SHA2)

Supported SSH ciphers algorithm:

aes128-cbc

128-bit AES with Cipher Block Chaining

aes128-ctr

128-bit AES with Counter Mode

aes256-cbc	256-bit AES with Cipher Block Chaining
aes256-ctr	256-bit AES with Counter Mode
aes128-gcm@openssh.com	128-bit AES with Galois/Counter Mode
aes256-gcm@openssh.com	256-bit AES with Galois/Counter Mode

The evaluator also examined the section 'Configuring MACsec with keychain using Layer 3 Traffic' of the AGD and found that:

3. Associate the preshared keychain name with the connectivity association.

[edit]

security-administrator@host:fips# set security macsec connectivity-association CA1 pre-shared-key-chain macsec-kc1

security-administrator@host:fips# set security macsec connectivity-association CA1 offset 50

security-administrator@host:fips# set security macsec connectivity-association CA1 cipher-suite gcm-aes-256

Moreover, the AGD section **Supported Cryptographic Algorithms** provides a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE as follows:

WARNING: For FIPS 140-3 compliance, use only FIPS-approved cryptographic algorithms in Junos OS in FIPS mode. The Cryptographic Algorithms referred to in Chapters "Configure SSH and Console Connection" and "Configure MACsec" have been tested for Common Criteria Evaluation of the device. Use of other cryptographic implementations was not evaluated nor tested.

The evaluator also verified sections 'Download Software Packages from Juniper Networks' of the AGD and found that the AGD describes the instructions for obtaining the update itself.

The AGD states that:

The TOE's administrator can download the Junos OS software package from the Juniper Networks website. Before the TOE's administrator begin to download the software, ensure that the TOE's administrator have a Juniper Networks Web account and a valid support contract. To obtain an account, complete the registration form at the Juniper Networks website: <https://userregistration.juniper.net/>.

To download software packages from Juniper Networks:

1. Using a Web browser, follow the links to the download URL on the Juniper Networks webpage. <https://support.juniper.net/support/downloads/>
2. Log in to the Juniper Networks authentication system using the username (generally your e-mail address) and password supplied by Juniper Networks representatives.
3. Download the software.

Additionally, the section titled 'Product Functionality Not Included in the Scope of the Evaluation' specifies features that are not assessed and tested by the EAs. The claimed AGD section states that:

The following product functionalities are not included in the CC evaluation:

- Use of telnet, since it violates the Trusted Path requirement set
- Use of FTP, since it violates the Trusted Path requirement set
- Use of SNMP, since it violates the Trusted Path requirement set
- Use of SSL, including management via J-Web, JUNOScript and JUNOScope, since it violates the Trusted Path requirement set
- Use of CLI account super-user and Linux root account.

Verdict:

PASS.

6.3.2 PREPARATIVE PROCEDURES (AGD_PRE.1) (CPP_ND_V3.0E)

6.3.2.1 EVALUATION ACTIVITY

The evaluator shall examine the Preparative procedures to ensure they include a description of how the Security Administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).

The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

Evaluator Findings:

The evaluator examined the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality. The evaluator reviewed the sections titled **TOE Environmental Components** of the AGD. The evaluator found that this section states the following:

Components	Description
Local Management Console	Any management workstation (computer) that is directly (serially) connected to the TOE's console port may be used by the TOE administrator for local administration of the TOE.
Remote Management Workstation	Any management workstation (computer) with a SSHv2 client installed that may be used by the TOE administrator for remote administration of the TOE through SSH protected channel.
Syslog Server	A syslog server, used for remote storage of audit records that have been generated by and transmitted from the TOE securely using SSH tunnel (SSHv2).
NTP Server	A NTPv4 server, used to provide reliable timestamps to the TOE.
MACsec Peer	A MACsec Peer, used for establishing MACsec communication for securing Layer 2 data using MACsec functionality.

Verdict:

PASS.

6.3.2.2 EVALUATION ACTIVITY

The evaluator shall examine the preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Evaluator Findings:

The evaluator checked the requirements above are met by the preparative procedures. The entire AGD was used to determine the verdict of this work unit. Upon investigation, the evaluator found that the AGD describes each of the devices in the operating environment.

The section titled **Supported Platforms** of AGD identifies the following supported platform:

Model	RE	RE CPU	MACsec Linecard	MIC-3D-10GESGP-E within MPC3E-3D-NG
MX240	RE-S-X6-128G RE-S-X6-64G	Intel HASWELL-EP Xeon E5-2608L V3	MPC10-10C and MPC10E-15C, with JNPR Penta Silicon MACsec ASIC	BMC82756 MACsec ASIC with CAVP num AES4550
MX480				
MX960				
EX9204	EX9200-RE2	Intel HASWELL-EP Xeon E5-2608L V3	EX9200-15C, with JNPR Penta Silicon MACsec ASIC	NA
EX9208				
EX9214				

Verdict:

PASS.

6.3.2.3 EVALUATION ACTIVITY

The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

Evaluator Findings:

The evaluator checked the requirements are met by the preparative procedures. The entire AGD was used to determine the verdict of this work unit. Upon investigation, the evaluator found that AGD describes all of the instructions necessary to install and configure the TOE to work in the target operating environment, including:

- Configuring Administrative Accounts and Passwords
- Configuring SSH
- Configuring the Remote Syslog Server
- Configuring Audit Log Options
- Configuring MACsec

Verdict:

PASS.

6.3.2.4 EVALUATION ACTIVITY

The evaluator shall examine the preparative procedures to ensure they include instructions on how to manage the TSF as a product and as a component of the larger Operational Environment in a manner that allows to preserve integrity of the TSF.

The intent of this requirement is to ensure there exists adequate preparative procedures (guidance in most cases) to put the TSF in a secure state (i.e., evaluated configuration). AGD_PRE.1 lists general requirements, the specific assurance activities implementing it are performed as part of FMT_SMF.1, FMT_MTD.1 and FMT_MOF.1 series of SFRs.

Evaluator Findings:

The evaluator ensured the preparative procedures include instructions to manage the TSF as a product and as a component of the larger operational environment in a manner that preserves the integrity of the TSF. The entire AGD was used to determine the verdict of this work unit. The same commands, configurations, and interfaces used to install the TOE are also used for ongoing management, so this is satisfied by AGD_PRE.1 Test #3.

Verdict:

PASS.

6.3.2.5 EVALUATION ACTIVITY

In addition, the evaluator shall ensure that the following requirements are also met. The preparative procedures must:

- include instructions to provide a protected administrative capability; and
- identify TOE passwords that have default values associated with them and mandate that they shall be changed.

Evaluator Findings:

The evaluator ensured the preparative procedures include instructions to provide a protected administrative capability and changing default passwords. The sections titled '**Configure a Network Device Collaborative Protection Profile Authorized Administrator**' and '**Configure SSH and Console Connection**' were used to determine the verdict of this work unit.

In the section '**Enable FIPS Mode**', the evaluator confirmed that the documentation identifies that the TOE initially allows login to the root account with no password and mandates that the administrator must configure a root password in compliance with the FIPS password requirements before further configuration. The procedures explicitly require the administrator to set and commit a new password, ensuring that default credentials are changed in accordance with security best practices.

The evaluator also observed that after zeroization, the device enters Amnesiac mode and again allows login using the root account with no password, after which the administrator is required to configure a new root password.

Verdict:

PASS.

6.4 AVA: VULNERABILITY ASSESSMENT

6.4.1 VULNERABILITY SURVEY (AVA_VAN.1) (CPP_ND_V3.0E)

6.4.1.1 EVALUATION ACTIVITY (DOCUMENTATION)

In addition to the activities specified by the CEM in accordance with Table 2, the evaluator shall perform the following activities.

The evaluator shall examine the documentation outlined below provided by the developer to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.

The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify compute-capable hardware components, at a minimum that must include the processor, and where applicable, discrete crypto ASICs, TPMs, etc. used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic implementations, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis.

If the TOE is a distributed TOE then the developer shall provide:

- a. documentation describing the allocation of requirements between distributed TOE components as in [NDcPP, 3.4]
- b. a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, Table 2]
- c. additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.4.1.2 and 3.5.1.2.

Evaluator Findings:

The TOE is not distributed; hence, this requirement is not applicable.

Verdict:

PASS.

6.4.1.2 EVALUATION ACTIVITY

The evaluator shall formulate hypotheses in accordance with process defined in Appendix A. The evaluator shall document the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

Evaluator Findings:

The evaluator documented their analysis and testing of potential vulnerabilities with respect to this requirement. Public searches were performed against all keywords found within the Security Target and AGD that may be applicable to specific TOE components. This included protocols, TOE software version, and TOE hardware to ensure sufficient coverage under AVA. The evaluator searched the Internet for potential vulnerabilities in the TOE using the web sites listed below. The sources of the publicly available information are provided below:

- o <https://nvd.nist.gov/view/vuln/search>
- o <https://cve.mitre.org/cve>
- o <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- o <https://kb.juniper.net>

The evaluator performed the public domain vulnerability searches using the following key words. The searches were performed on **Sep 26, 2025, Nov 24, 2025, April 30, 2026 and June 18, 2026.**:

- o Juniper MX240 (cpe:2.3:h:juniper:mx240:-:*:*:*:*:**)
- o Juniper MX480 (cpe:2.3:h:juniper:mx480:-:*:*:*:*:**)
- o Juniper MX960 (cpe:2.3:h:juniper:mx960:-:*:*:*:*:**)
- o Juniper EX9204 (cpe:2.3:h:juniper:ex9204:-:*:*:*:*:**)
- o Juniper EX9208 (cpe:2.3:h:juniper:ex9208:-:*:*:*:*:**)
- o Juniper EX9214 (cpe:2.3:h:juniper:ex9214:-:*:*:*:*:**)
- o JUNOS OS 24.4R2
- o Intel Xeon E5-2608L V3 (cpe:2.3:h:intel:xeon_e5-2608l_v3:-:*:*:*:*:**)
- o Juniper Penta Silicon
- o Broadcom BCM82756
- o OpenSSL 1.1.1zb
- o OpenSSH 9.7p1 (cpe:2.3:a:openbsd:openssh:9.7:p1:-:*:*:*:**)
- o Junos libMD
- o Junos libquicksec 7.0

The evaluation lab examined each result provided from NVD and other databases to determine if the current TOE version or component within the environment was vulnerable. Based upon the analysis, any issues found that were generated were patched in the TOE version and prior versions, mitigating the risk factor.

Verdict:

PASS.

7 DETAILED TEST CASES (TEST ACTIVITIES)

7.1 TEST ACTIVITIES

7.1.1 SECURITY AUDIT (FAU)

7.1.1.1 FAU_GEN.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different identity and authentication (I&A) mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. 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. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.
Test Steps	• The evaluator triggers each auditable event on the TOE. • The evaluator verifies that each audit record is generated and contains the required information. • The evaluator verifies all the audit record matches the audit records in all the test cases.
Pass/Fail with Explanation	Pass. Covered by audit records in each test case.

7.1.1.2 FAU_GEN.1/MACSEC TEST #1 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall complete the evaluation activity for FAU_GEN.1 as described in the NDcPP for the auditable events defined in the PP-Module in addition to the applicable auditable events that are defined in the NDcPP. The evaluator shall also ensure that the administrative actions defined for this PP-Module are appropriately audited.
Test Steps	• Trigger each auditable event on the TOE. Verify that each audit record is generated and contains the required information.
Pass/Fail with Explanation	Pass. The audit records associated with each test case are recorded with each test case. A comparison of required audit records to the presented audit records was additionally performed. This analysis shows that each required audit record is generated by the TOE.

7.1.1.3 FAU_GEN.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of auditable events to TOE components in the Security Target. For all events involving more than one TOE component when an audit event is triggered, the evaluator has to check that the event has been audited on both sides (e.g. failure of building up a secure communication channel between the two components). This is not limited to error cases but includes also events about successful actions like successful build up/tear down of a secure communication channel between TOE components. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.
Pass/Fail with Explanation	NA. The TOE is not distributed; hence, this activity is not applicable.

7.1.1.4 FAU_GEN.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.
Pass/Fail with Explanation	Pass. This activity is accomplished in conjunction with the testing of FAU_GEN.1.1.

7.1.1.5 FAU_GEN.2 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For distributed TOEs the evaluator shall verify that where auditable events are instigated by another component, the component that records the event associates the event with the identity of the instigator. The evaluator shall perform at least one test on one component where another component instigates an auditable event. The evaluator shall verify that the event is recorded by the component as expected and the event is associated with the instigating component. It is assumed that an event instigated by another component can at least be generated for building up a secure channel between two TOE components. If for some reason (could be e.g. TSS or Guidance Documentation) the evaluator would come to the conclusion that the overall TOE does not generate any events instigated by other components, then this requirement shall be omitted.
Pass/Fail with Explanation	NA. The TOE is not distributed; hence, this activity is not applicable.

7.1.1.6 FAU_STG_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.
Test Steps	• Record the audit server name and version information. • Configure a syslog server to communicate with the TOE by generating a public key on the remote syslog server. • On the TOE, create a class named monitor that has permission to trace events. • On the TOE, create a user named syslog-mon with the class monitor and with public key authentication using the key generated earlier. • On the TOE, configure NETCONF over SSH. • Establish an SSH session from the remote syslog server to the TOE using NETCONF and fetch the syslog events. • Verify that the logs generated on the TOE and the ones seen on the remote syslog server are the same. • Verify that the traffic between the TOE and Syslog is not sent in plaintext using packet capture.
Pass/Fail with Explanation	Pass. The TOE is capable of transferring audit data to an external audit server automatically without administrative intervention and it is protected by SSH when it is exported. This meets the testing requirement.

7.1.1.7 FAU_STG_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 2: For distributed TOEs, Test 1 defined above shall be applicable to all TOE components that forward audit data to an external audit server.
Pass/Fail with Explanation	NA. The TOE is not distributed; hence, this activity is not applicable.

7.1.1.8 FAU_STG_EXT.1 TEST #3 (A) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 3: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall then make note of whether the TSS claims persistent or non-persistent logging and perform one of the following actions: i. If persistent logging is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are still maintained within the local audit storage.
Test Steps	• Perform an activity to generate audit data. • Verify that the above activity generates an audit log and note the same. • Restart the TOE and verify that the highlighted log from the above steps is still available in the TOE's log store.
Pass/Fail with Explanation	Pass. The TOE retains log events in its local audit storage even after it is rebooted. This meets the testing requirement.

7.1.1.9 FAU_STG_EXT.1 TEST #3 (B) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 3: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall then make note of whether the TSS claims persistent or non-persistent logging and perform one of the following actions: ii. If non-persistent logging is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are no longer present within the local audit storage.
Pass/Fail with Explanation	NA. Persistent logging is selected in the ST, hence this test is not applicable.

7.1.1.10 FAU_STG_EXT.1 TEST #4 (A) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement:

	Test 4: The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.5. Depending on the configuration this means that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that: The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.5).
Pass/Fail with Explanation	NA. 'Overwrite previous audit records' is selected in the ST, hence this test is not applicable.

7.1.1.11 FAU_STG_EXT.1 TEST #4 (B) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 4: The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.5. Depending on the configuration this means that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that: The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.5)
Test Steps	- Configure a local logfile on the TOE along with the overwriting parameters. - Perform log-generating actions. - Verify that the logs are locally stored and when the audit data is filled to the max, the existing audit data is overwritten.
Pass/Fail with Explanation	Pass. The TOE overwrites the oldest audit records when the local audit space is filled. This meets the testing requirement.

7.1.1.12 FAU_STG_EXT.1 TEST #4 (C) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 4: The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in

	FAU_STG_EXT.1.5. Depending on the configuration this means that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that: iii. The TOE behaves as specified (for the option ' other action ' in FAU_STG_EXT.1.5).
Pass/Fail with Explanation	NA. 'Other action' is not selected in the ST, hence this test is not applicable.

7.1.1.13 FAU_STG_EXT.1 TEST #5 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 5: For distributed TOEs, for the local storage according to FAU_STG_EXT.1.4, Test 1 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2, Test 2 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.
Pass/Fail with Explanation	NA. The TOE is not distributed; hence, this activity is not applicable.

7.1.1.14 FAU_STG_EXT.1 TEST #6 (CPP_ND_V3.0E) [TD0886]

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 6 [Conditional]: In case manual export or ability to view locally is selected in FAU_STG_EXT.1.6, during interruption the evaluator shall perform a TSF-mediated action and verify the event is recorded in the audit trail. Note: The intent of the test is to ensure that the local audit TSF (as specified by FAU_STG_EXT.1.3) operates independently from the ability to transmit the generated audit data to an external audit server (as specified in FAU_STG_EXT.1.1). There are no specific requirements on the interruption of the connection between the TOE and the external audit server (as for FTP_ITC.1). TD0886 has been applied.
Test Steps	- Establish an SSH session from the remote syslog server to the TOE using NETCONF and start fetching syslog events. - Verify that the SSH session was established using TOE logs. - Create a read-only user so that logs are generated for the same.

	• Verify that the logs generated on the TOE for the above activity and those seen on the remote syslog server are the same. • Interrupt the connection between the remote syslog server and the TOE by shutting down the syslog server interface. • Attempt a configuration change on the device. • Import the local TOE logfile to a remote entity. • Verify the logs after the interruption are visible on the TOE locally and in the file imported on to the remote device.
Pass/Fail with Explanation	Pass. Logs generated using a TSF-mediated action are recorded in the audit trail during an interruption of the connection with the remote syslog server. This meets the testing requirement.

7.1.2 CRYPTOGRAPHIC SUPPORT (FCS)

7.1.2.1 FCS_CKM.1 RSA (CPP_ND_V3.0E) [TD0921]

Item	Data
Test 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. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). Key Generation for FIPS PUB 186-4 or FIPS PUB 186-5 RSA Schemes The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d. This test must be repeated for each supported RSA modulo and generation method. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: a) Random Provable Primes (p and q shall be provable primes) b) Random Probable primes (p and q shall be probable primes) c) Provable Primes with Conditions (p_1, p_2, q_1, q_2, p and q shall all be provable primes) d) Provable/probable primes with Conditions (p_1, p_2, q_1, and q_2 shall be provable primes and p and q shall be probable primes) e) Probable primes with Conditions (p_1, p_2, q_1, q_2, p and q shall all be probable primes) The Random Provable primes, and all the Primes with Conditions can be tested in the same manner because each of these begin with a starting random number and calculate the p and q values from this value. The test instructs the TSF to generate intermediate values and the p, q, n, and d values. The evaluator then validates the correctness of the values generated by the TSF. To test the key generation method for the Random Provable primes method or 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. If the TSF provides the input to the key generation function, such input must be recorded and verified. For each RSA key length (modulo) claimed, the evaluator shall generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing Key Pair values generated by the TSF with those generated using the same set of input values using a known good implementation. The Random Probable primes must be tested in a different way because this test generates different random numbers, not related to each other, until the number satisfies the "probably prime" requirements. This validation method requires two tests for Random Probable primes. These include the Known Answer Test and the Miller-Rabin probabilistic primality test. To test the key generation method for the Random Probable primes, the known answer test must be used. The evaluator shall compare the results of a known good implementation with the TSF results for a set (of a corresponding size depending on modulus) of supplied values containing private prime factor p, private prime factor q and confirm all public keys, e, match. Then the evaluator shall use the TSF to generate prime p, q pairs for each modulus size and perform the Miller-Rabin tests. TD0921 has been applied.
Pass/Fail with Explanation	Algorithm: RSA KeyGen (FIPS186-5) Key Generation Mode: probable Key size / Modulus: 2048, 3072, 4096 Fixed Public Exponent: 010001 Private Key Format: standard Public Exponent Mode: fixed CAVP #: A6978 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.2 FCS_CKM.1 ECC (CPP_ND_V3.0E) [TD0921]

Item	Data
Test 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. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). Key Generation for Elliptic Curve Cryptography (ECC) <i>Key Generation for ECDSA Schemes</i> Key pairs for the ECDSA consist of pairs (d, Q), where the private key, d, is an integer, and the public key, Q, is an elliptic curve point. The evaluator shall verify the implementation of ECC Key Generation by the TOE using the following components tests:

	• ECC Key Pair Generation • ECC Public Key Validation (PKV) <i>ECC Key Pair Generation Test</i> There are four methods by which these pairs may be generated: • FIPS 186-4 Section B.4.1 Key Pair Generation Using Extra Random Bits Using this method, 64 more bits are requested from the RBG than are needed for d so that bias produced by the mod function is negligible. • FIPS 186-4 Section B.4.2 Key Pair Generation by Testing Candidates Using this method, a random number is obtained and tested to determine that it will produce a value of d in the correct range. If d is out-of-range, another random number is obtained (i.e., the process is iterated until an acceptable value of d is obtained). • FIPS 186-5 Section A.2.1 ECDSA Key Pair Generation using Extra Random Bits Using this method, more bits are requested from the DRBG than are needed for d so that the bias produced by the mod function is negligible. • FIPS 186-5 Section A.2.2 ECDSA Key Pair Generation by Rejection Sampling Using this method, a random number is obtained and tested to determine that it will produce a value of d in the correct range. If d is out of range, an ERROR is returned. The evaluator shall test the ECC Key Pair Generation by having the TSF produce 10 key pairs (d, Q) for each implemented key generation method using each supported curve (i.e., P-256, P-384, and P-521). The steps are the same for each key generation method and curve. The private key, d, shall be generated using the output of an approved DRBG converted to an integer via modular reduction or the discard method. The known private key is then used by the TSF to compute the public key, Q'. The evaluator then validates the correctness by comparing the value Q' computed by the TSF to the public key, Q generated by a known good implementation. <i>ECC Public Key Verification (PKV) Test</i> The evaluator shall generate 12 key pairs (d, Q) for each selected curve, with 6 valid public keys, Q, using a known good implementation and 6 modified, Q', and determine whether the TSF can accurately detect these modifications. Q' should be otherwise valid but include at least one of the following errors: a) point X or Y not on the curve, b) point X or Y is too large for the field for the given curve. The evaluator encouraged to make sure that the modification does not inadvertently result in another valid public key (e.g., modifying Y and accidentally hitting a different point on the curve). TD0921 has been applied.
Pass/Fail with Explanation	Algorithm: ECDSA KeyGen (FIPS186-5) Curves: P-256, P-384, P-521 Secret Generation Mode: testing candidates Algorithm: ECDSA KeyVer (FIPS186-5) Curves: P-256, P-384, P-521 CAVP #: A6978 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.3 FCS_CKM.1 FFC - FIPS PUB 186-4 (CPP_ND_V3.0E)

Item	Data
Test 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. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). 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: • 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: • 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: • $\text{len}(q)$ bit output of RBG where $1 \leq x \leq q-1$ • $\text{len}(q) + 64$ bit output of RBG, followed by a mod $q-1$ operation and a $+1$ operation, where $1 \leq x \leq q-1$. The security strength of the RBG must be at least that of the security offered by the FFC parameter set. To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set. For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm • $g \neq 0,1$ • q divides $p-1$ • $g^q \bmod p = 1$ • $g^x \bmod p = y$ for each FFC parameter set and key pair.

Pass/Fail with Explanation	N/A. FFC schemes using cryptographic key sizes of 2048-bit or greater are not claimed in this evaluation, hence this test is not applicable.
-----------------------------------	--

7.1.2.4 FCS_CKM.1 FFC - "SAFE-PRIME" GROUPS (CPP_ND_V3.0E)

Item	Data
Test 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. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported). FFC Schemes using "safe-prime" groups Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.
Pass/Fail with Explanation	N/A. FFC Schemes using 'safe-prime' groups are not claimed in this evaluation; hence this test is not applicable.

7.1.2.5 FCS_CKM.2 RSA (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	RSA-based key establishment The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses RSAES-PKCS1-v1_5.
Pass/Fail with Explanation	N/A. RSA-based key establishment schemes are not claimed in this evaluation; hence this test is not applicable.

7.1.2.6 FCS_CKM.2 SP800-56A - ECC (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Key Establishment Schemes The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below. ECC and FIPS 186-type FFC 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 for ECC and FIPS186- type. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented

correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag.

Function Test

The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme- key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields.

If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values.

If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm.

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACtag, and any inputs used in the KDF, such as the other info and TOE id fields.

The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACtag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator shall 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.
Pass/Fail with Explanation	Algorithm: KAS-ECC-SSC Sp800-56Ar3 CAVP #: A6978 Domain Parameter Generation Methods: P-256, P-384, P-521 Scheme: ephemeralUnified: KAS Role: initiator, responder Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.7 FCS_CKM.2 SP800-56A - FFC (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Key Establishment Schemes The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below. ECC and FIPS 186-type FFC 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 for ECC and FIPS186- type. 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. <i>Function Test</i> The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme- key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested. The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields.

	If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret. The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values. If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm. <i>Validity Test</i> 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 shall 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.
Pass/Fail with Explanation	NA. The ST does not claim FFC Schemes using "FIPS 186-Type" parameter-size sets; hence this test is not applicable.

7.1.2.8 FCS_CKM.2 FFC SAFE-PRIME (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.

Pass/Fail with Explanation	N/A. FFC Schemes using “safe-prime” groups are not claimed in this evaluation; hence this test is not applicable.
-----------------------------------	---

7.1.2.9 FCS_CKM.4 (CPP_ND_V3.0E)

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

7.1.2.10 FCS_COP.1/DATAENCRYPTION AES-CBC (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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. 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. 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. 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 keys 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.

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.

AES-CBC Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i -block message where $1 < i \leq 10$. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.

The evaluator shall also test the decrypt functionality for each mode by decrypting an i -block message where $1 < i \leq 10$. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

AES-CBC Monte Carlo Tests

The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

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

	The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation. The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AESCBC-Decrypt.
Pass/Fail with Explanation	Algorithm: AES-CBC Direction: Decrypt, Encrypt Key Length: 128, 256 CAVP #: A6978 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.11 FCS_COP.1/DATAENCRYPTION AES-GCM (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 a) 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. a) 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. b) Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested. The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known. The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

Pass/Fail with Explanation	Algorithm: AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256 IV Generation: External IV Generation Mode: 8.2.2 Tag Length: 128 IV Length: 96 Payload Length: 128, 1024 AAD Length: 64 CAVP #: A6978, A4708 Algorithm: AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256 Tag Length: 64, 96, 104, 112, 120, 128 IV Length: 96 Payload Length: 120, 128 AAD Length: 120, 128 CAVP #: AES 4550 Pass. Based on these findings, this assurance activity is considered satisfied.
-----------------------------------	--

7.1.2.12 FCS_COP.1/DATAENCRYPTION AES-CTR (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	AES-CTR Known Answer Tests The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Since the Counter Mode does not specify the counter that is used, it is not possible to implement an automated test for this mode. The generation and management of the counter is tested if the TSF is validated against the requirements of the Functional Package for Secure Shell referenced in Section 2.2 of the cPP. If CBC and/or GCM are selected in FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AES-GCM Known Answer Test, or the following test shall be performed (all of these tests demonstrate the correctness of the AES algorithm): There are four Known Answer Tests (KATs) described below to test a basic AES encryption operation (AES-ECB mode). For all KATs, the plaintext, ~~IV~~, and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator 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. KAT-1 To test the encrypt functionality, the evaluator shall supply a set of 5 plaintext values for each selected keysize and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros. KAT-2 To test the encrypt functionality, the evaluator shall supply a set of 5 key values for each selected keysize and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value. KAT-3 To test the encrypt functionality, the evaluator shall supply a set of key values for each selected keysize as described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values. A set of 128 128-bit keys, a set of 192 192-bit keys, and/or a set of 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]. KAT-4 To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the ciphertext values that result from encryption of the given plaintext using each selected keysize with a key value of all zeros (e.g. 256 ciphertext values will be generated if 128 bits and 256 bits are selected and 384 ciphertext values will be generated if all keysizes are selected). Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128]. AES-CTR Multi-Block Message Test The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10 (test shall be performed using AES-ECB mode). For each i the evaluator shall choose a key and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key using a known good implementation. The evaluator shall perform this test using each selected keysize. AES-CTR Monte-Carlo Test The evaluator shall test the encrypt functionality using 100 plaintext/key pairs. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows: # Input: PT, Key for i = 1 to 1000: CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i] The ciphertext computed in the 1000th iteration 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 perform this test using each selected keysize. There is no need to test the decryption engine.
--	--

Pass/Fail with Explanation	Algorithm: AES-CTR Direction: Decrypt, Encrypt Key Length: 128, 256 Payload Length: 8-128 Increment 8 Incremental Counter Counter Tests Performed CAVP #: A6978 Pass. Based on these findings, this assurance activity is considered satisfied.
----------------------------	---

7.1.2.13 FCS_COP.1/HASH (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The TSF hashing functions can be implemented in one of two modes. The first mode is the byteoriented 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 bitoriented 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 bitoriented vs. the byteoriented testmacs. The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. Short Messages Test - Bit-oriented Mode The evaluator shall devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluator shall compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Short Messages Test - Byte-oriented Mode The evaluator shall devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF. Selected Long Messages Test - Bit-oriented Mode The evaluator shall devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 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.

	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 (e.g. 512 bits for SHA-256). The length of the ith message is $m + 8 \cdot 99 \cdot i$, 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. Pseudorandomly Generated Messages Test This test is for byteoriented implementations only. The evaluator shall 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 evaluator shall then ensure that the correct result is produced when the messages are provided to the TSF.
Pass/Fail with Explanation	Algorithm: SHA-1, SHA2-256, SHA2-384, SHA2-512 Message Length: 0-65536 Increment 8 CAVP #: A6978 Algorithm: SHA2-256, SHA2-512 Message Length: 0-65536 Increment 8 CAVP #: A6975 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.14 FCS_COP.1/KEYEDHASH (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.
Pass/Fail with Explanation	Algorithm: HMAC-SHA2-256, HMAC-SHA2-512 MAC: 256, 512 Key Length: 256, 512 CAVP #: A6978 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.15 FCS_COP.1/CMAC TEST #1 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: CMAC Generation Test To test the generation capability of AES-CMAC, the evaluator shall provide to the TSF, for each key

	length-message length-CMAC length tuple (in bytes), a set of eight arbitrary key-plaintext tuples that will result in the generation of a known MAC value when encrypted. The evaluator shall then verify that the correct MAC was generated in each case.
Pass/Fail with Explanation	Algorithm: AES-CMAC Direction: Generation, Verification Key Length: 128, 256 MAC Length: 8, 64, 120 Message Length: 0, 32768, 524288 CAVP #: A6976 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.16 FCS_COP.1/CMAC TEST #2 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: CMAC Verification Test To test the verification capability of AES-CMAC, the evaluator shall provide to the TSF, for each key length-message length-CMAC length tuple (in bytes), a set of 20 arbitrary key-MAC tuples that will result in the generation of known messages when verified. The evaluator shall then verify that the correct message was generated in each case. The following information should be used by the evaluator to determine the key length-message length-CMAC length tuples that should be tested: ▪ Key length: Values will include the following: ○ 16 ○ 32 ▪ Message length: Values will include the following: ○ 0 (optional) ○ Largest value supported by the implementation (no greater than 65536) ○ Two values divisible by 16 ○ Two values not divisible by 16 ▪ CMAC length: ○ Smallest value supported by the implementation (no less than 1) ○ 16 ▪ Any supported CMAC length between the minimum and maximum values
Pass/Fail with Explanation	Algorithm: AES-CMAC Direction: Generation, Verification Key Length: 128, 256 MAC Length: 8, 64, 120

	Message Length: 0, 32768, 524288 CAVP #: A6976 Pass. Based on these findings, this assurance activity is considered satisfied.
--	---

7.1.2.17 FCS_COP.1/MACSEC TEST #3 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform testing for AES-GCM as required by the NDcPP in FCS_COP.1/DataEncryption. In addition to the tests specified in the NDcPP for other iterations of FCS_COP.1, the evaluator shall perform the following tests: Test 3: KW-AE Test: To test the authenticated encryption capability of AES key wrap (KW), the evaluator shall provide five sets of 100 messages and keys to the TOE for each key length supported by the TSF. Each set of messages and keys shall correspond to one of five plaintext message lengths (detailed below). The evaluator shall have the TSF encrypt the messages with the associated key. The evaluator shall verify that the correct ciphertext was generated in each case.
Pass/Fail with Explanation	Algorithm: AES-KW Direction: Decrypt, Encrypt Cipher: Cipher Key Length: 128, 256 Payload Length: 128, 192, 256, 320, 4096 CAVP #: A6976 Algorithm: AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256 IV Generation: External IV Generation Mode: 8.2.2 Tag Length: 128 IV Length: 96 Payload Length: 128, 1024 AAD Length: 64 CAVP #: A4708 Algorithm: AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256 Tag Length: 64, 96, 104, 112, 120, 128 IV Length: 96

	Payload Length: 120, 128 AAD Length: 120, 128 CAVP #: AES 4550 Pass. Based on these findings, this assurance activity is considered satisfied.
--	---

7.1.2.18 FCS_COP.1/MACSEC TEST #4 (MOD_MACSEC_V1.0) [TD0728]

Item	Data
Test Assurance Activity	The evaluator shall perform testing for AES-GCM as required by the NDcPP in FCS_COP.1/DataEncryption. In addition to the tests specified in the NDcPP for other iterations of FCS_COP.1, the evaluator shall perform the following tests: Test 4: KW-AD Test: To test the authenticated decryption capability of AES KW, the evaluator shall provide five sets of 100 ciphertext values and keys to the TOE for each key length supported by the TSF. Each set of ciphertexts and keys shall correspond to one of five plaintext message lengths (detailed below). For each set of 100 ciphertext values, 20 shall not be authentic (i.e., fail authentication). The evaluator shall have the TSF decrypt the ciphertext messages with the associated key. The evaluator shall then verify the correct plaintext was generated or the failure to authenticate was correctly detected. The messages in each set for both tests shall be the following lengths: • two that are non-zero multiples of 128 bits (two semiblock lengths) • two that are odd multiples of the semiblock length (64 bits) • the largest supported plaintext length less than or equal to 4096 bits TD0728 has been applied.
Pass/Fail with Explanation	Algorithm: AES-KW Direction: Decrypt, Encrypt Cipher: Cipher Key Length: 128, 256 Payload Length: 128, 192, 256, 320, 4096 CAVP #: A6976 Algorithm: AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256 IV Generation: External IV Generation Mode: 8.2.2 Tag Length: 128 IV Length: 96 Payload Length: 128, 1024

	AAD Length: 64 CAVP #: A4708 Algorithm: AES-GCM Direction: Decrypt, Encrypt Key Length: 128, 256 Tag Length: 64, 96, 104, 112, 120, 128 IV Length: 96 Payload Length: 120, 128 AAD Length: 120, 128 CAVP #: AES 4550 Pass. Based on these findings, this assurance activity is considered satisfied.
--	---

7.1.2.19 FCS_COP.1/SIGGEN ECDSA (CPP_ND_V3.0E) [TD0921]

Item	Data
Test Assurance Activity	ECDSA Signature Algorithm Tests The evaluator shall verify the implementation of ECDSA Signature generation by the TOE using the Signature Generation Test. This test validates the TSF generation of the (r, s) pair that represent the digital signature. The digital signature shall be verified using the same domain parameters and hash function that were used during signature generation. An approved hash function or an XOF shall be used for this purpose. Signature Generation Test The purpose of this test is to verify the ability of the TSF to produce correct signatures. To test signature generation, the evaluator supplies 10 pseudorandom messages to the TSF and a key pair, (d, Q), generated by a known good implementation. Exercising each applicable curve (i.e., P-256, P 384, or P-521) and hash algorithm or extendable-output function combination, the TSF generates signatures for each supplied message and returns the corresponding signatures. Using a known-good implementation, the evaluator validates the signatures by using the associated public key, Q, to verify the signature. Signature Verification Test The purpose of this test is to verify the ability of the TSF to accept valid signatures and reject invalid signatures. For each curve/hash algorithm or extendable-output function combination supported by the TSF, the evaluator shall use a known good implementation to generate a key pair, (d, Q), and use known good implementation with the private key, d, to sign 15 pseudorandom messages of 1024

	bits. The evaluator shall alter some of the messages or signatures so that signature verification should fail. To test signature verification, the evaluator supplies the public key, Q, and 15 pseudorandom messages, including altered messages, to the TSF for verification of signatures. The evaluator shall then verify that the TSF validates correct signatures on the original messages and flags or rejects the altered messages. TD0921 has been applied.
Pass/Fail with Explanation	Algorithm: ECDSA SigGen (FIPS186-5), ECDSA SigVer (FIPS186-5) Curve: P-256, P-384, P-521 Hash Algorithm: SHA2-256, SHA2-384, SHA2-512 CAVP #: A6978 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.20 FCS_COP.1/SIGGEN RSA (CPP_ND_V3.0E) [TD0921]

Item	Data
Test Assurance Activity	RSA Signature Algorithm Tests Signature Generation Test The evaluator shall verify the implementation of RSA Signature generation by the TOE using the Signature Generation Test. This test verifies the ability of the TSF to produce correct signatures. There are 2 different RSA Signature algorithms that can be implemented. These include: - RSASSA-PKCS1-v1.5 - RSASSA-PSS To test signature generation, the evaluator generates or obtains 10 messages for each modulus size/hash or extendable-output function combination supported by the TOE. Using a key generated by a known good implementation, the TSF generates and returns the corresponding signatures to a known good implementation that validates the signatures by using the associated public key to verify the signature. The evaluator shall verify the correctness of the TOE's signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures. Signature Verification Test The evaluator shall verify the implementation of RSA Signature verification by the TOE using the Signature Verification Test. This test verifies the ability of the TSF to recognize valid and invalid signatures. There are 2 different RSA Signature algorithms that can be implemented. These include:

	a. RSASSA-PKCS1-v1.5 b. RSASSA-PSS For each modulus size/hash or extendable-output function combination supported by the TOE, the evaluator shall use a known good implementation to generate a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits. Some of the public keys, e, messages, IR format, or signatures must be altered so that signature verification should fail. The modifications must cover distinct “key modified”, “message modified”, “signature modified”, “IR moved”, and “trailer moved” tests. The modulus, hash or extendable-output algorithm, public key e values, messages, and signatures are forwarded to the TSF. The TSF then attempts to verify the signatures and returns the results. The evaluator then compares the received results with the stored results from a known good implementation. The evaluator verifies that the TSF validates correct signatures on the original messages and flags or rejects the altered messages. TD0921 has been applied.
Pass/Fail with Explanation	Algorithm: RSA SigGen (FIPS186-5) Signature Type: PKCS 1.5 Key size / Modulus: 2048, 3072, and 4096 Hash Algorithm: SHA2-256, SHA2-512 Algorithm: RSA SigVer (FIPS186-5) Signature Type: PKCS 1.5 Key size / Modulus: 2048, 3072, and 4096 Hash Algorithm: SHA2-256, SHA2-512 Fixed Public Exponent: 010001 Public Exponent Mode: fixed CAVP #: A6978 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.21 FCS_MACSEC_EXT.1 TEST #5 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 5: The evaluator shall successfully establish a MACsec channel between the TOE and a MACsec-capable peer in the operational environment and verify that the TSF logs the communications. The evaluator shall capture the traffic between the TOE and the operational environment to determine the SCI that the TOE uses to identify the peer. The evaluator shall then configure a test system to capture traffic between the peer and the TOE to modify the SCI that is used to identify the peer. The evaluator then verifies that the TOE does not reply to this traffic and logs that the traffic was discarded.
Test Steps	Configure the TOE to establish a MACsec channel with a peer.

	• Configure the peer to establish a MACsec channel with the TOE. • Activate the MACsec channel and initiate a ping from the peer to the TOE. • Verify that the MACsec connection is successfully established and that the TOE logs the communication. • Verify the successful connection via PCAP and note the SCI the TOE uses to identify the peer. • Re-attempt MACsec communication between the peer and TOE, using the Acumen MACsec tool to modify the SCI used to identify the peer. • Verify via PCAP that the TOE does not reply to this traffic. • Verify that the TOE logs this traffic being discarded.
Pass/Fail with Explanation	Pass. The TOE successfully established a MACsec connection with a peer having a valid SCI and logged the same. The TOE did not reply to traffic wherein the SCI used to identify the peer was modified and logs the discarded traffic.

7.1.2.22 FCS_MACSEC_EXT.1 TEST #6 (MOD_MACSEC_V1.0) [TD0884]

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 6: The evaluator shall send Ethernet traffic to the TOE's MAC address that iterates through the full range of supported EtherType values (refer to List of Documented EtherTypes) and observes that traffic for all EtherType values is discarded by the TOE except for the traffic which has an EtherType value described in the SFR. Note that there are a large number of EtherType values so the evaluator is encouraged to execute a script that automatically iterates through each value. TD0884 has been applied.
Test Steps	• Activate the MACsec channel between the TOE and peer and verify successful establishment using TOE logs. • Use the Acumen MACsec tool to send Ethernet traffic to the TOE's MAC address, iterating through the full range of EtherType values. • Verify via PCAP that the TOE discards traffic for all EtherTypes other than 88-8E, 88-E5, and 88-08.
Pass/Fail with Explanation	Pass. The TOE discards traffic for all EtherType values other than 88-8E, 88-E5, and 88-08.

7.1.2.23 FCS_MACSEC_EXT.2 TEST #7 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 7: The evaluator shall transmit MACsec traffic to the TOE from a MACsec-capable peer in the operational environment. The evaluator shall verify via packet captures, audit logs, or both that the frame bytes after the MACsec Tag values in the received traffic is not obviously predictable.
Test Steps	• Activate the MACsec channel on the TOE. • Transmit MACsec traffic to the TOE from a MACsec-capable peer.

	• Verify the MACsec connection is established using logs. • Verify via packet capture that the frame bytes after the MACsec Tag values in the received traffic is not obviously predictable.
Pass/Fail with Explanation	Pass. The frame bytes after the MACsec Tag values in the received traffic were not obviously predictable.

7.1.2.24 FCS_MACSEC_EXT.2 TEST #8 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 8: The evaluator shall transmit valid MACsec traffic to the TOE from a MACsec-capable peer in the operational environment that is routed through a test system set up as a man-in-the-middle. The evaluator shall use the test system to intercept this traffic to modify one bit in a packet payload before retransmitting to the TOE. The evaluator shall verify that the traffic is discarded due to an integrity failure.
Test Steps	• Activate the MACsec channel on the TOE. • Verify the MACsec connection is established using logs. • Transmit MACsec traffic to the TOE from a MACsec-capable peer, use Acumen MACsec tool on an MitM server to intercept traffic between the TOE and a MACsec peer, modify the packet payload, and retransmit it to the TOE. • Verify that the TOE discards the modified traffic due to an integrity failure.
Pass/Fail with Explanation	Pass. The TOE detects and discards the MACsec traffic with modified packet payloads due to integrity failure.

7.1.2.25 FCS_MACSEC_EXT.3 TEST #1 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	Testing of the TOE's MACsec capabilities and verification of the deterministic random bit generator is sufficient to demonstrate that this SFR has been satisfied.
Pass/Fail with Explanation	Pass. This testing is sufficiently covered under FCS_RBG_EXT.1.

7.1.2.26 FCS_MACSEC_EXT.4 TEST #9 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 9: For each supported method of peer authentication in FCS_MACSEC_EXT.4.1, the evaluator shall follow the operational guidance to configure the supported method (if applicable). The evaluator shall set up a packet sniffer between the TOE and a MACsec-capable peer in the operational environment. The evaluator shall then initiate a connection between the TOE and the peer such that authentication occurs and a secure connection is established. The evaluator shall

	wait one minute and then disconnect the TOE from the peer and stop the sniffer. The evaluator shall use the packet captures to verify that the SC was established via the selected mechanism and that the non-VLAN EtherType of the first data frame sent between the TOE and the peer is 88-E5.
Test Steps	• Configure the environment to establish MACsec communication between the TOE and a pre-shared-key (PSK) authenticated peer via the bridge. • Ping the TOE from the peer. • Wait one minute and shut down the bridge to disconnect the TOE from the peer. • Verify via PCAP that the SC was established via PSKs and that the non-VLAN EtherType of the first data frame sent between the TOE and the peer is 88-E5.
Pass/Fail with Explanation	Pass. The TOE successfully establishes the MACsec SC via PSKs. The non-VLAN EtherType of the first data frame sent between the TOE and the peer was 88-E5.

7.1.2.27 FCS_MACSEC_EXT.4 TEST #10 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 10: The evaluator shall capture traffic between the TOE and a MACsec-capable peer in the operational environment. The evaluator shall then cause the TOE to distribute a SAK to that peer, capture the MKPDUs from that operation, and verify the key is wrapped in the captured MKPDUs.
Test Steps	• Configure the environment to establish MACsec communication between the TOE and a peer, assigning the TOE a lower key-server priority number to prompt it to distribute a SAK to the peer. • Initiate MACsec traffic from the peer to the TOE. • Verify via PCAP that the key is wrapped in the captured MKPDUs.
Pass/Fail with Explanation	Pass. The TOE ensures that the key is wrapped in the captured MKPDUs.

7.1.2.28 FCS_MKA_EXT.1.3 TEST #11 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 11: The evaluator shall transmit MKA traffic (MKPDUs) to the TOE from an MKA-capable peer in the operational environment. The evaluator shall verify via packet captures, audit logs, or both that the last 16 octets of the MKPDUs in the received traffic do not appear to be predictable.
Test Steps	• Configure the TOE for MACsec. • Configure the PEER for MACsec. • Activate MACsec on the PEER to initiate MKA traffic to the TOE from the PEER. • Verify via PCAP that the last 16 octets of the MKPDUs in the received traffic do not appear to be predictable.
Pass/Fail with Explanation	Pass. The TOE ensured that the last 16 octets of the MKPDU are not predictable.

7.1.2.29 FCS_MKA_EXT.1.3 TEST #12 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 12: The evaluator shall transmit valid MKA traffic to the TOE from an MKA-capable peer in the operational environment that is routed through a test system set up as a man-in-the-middle. The evaluator shall use the test system to intercept this traffic to modify one bit in a packet payload before retransmitting to the TOE. The evaluator shall verify that the traffic is discarded due to an integrity failure.
Test Steps	• Activate the MACsec channel between the TOE and peer and verify successful establishment using TOE logs. • Verify live peer list before sending modified byte. • Modify a packet payload in transit by using the Acumen-MACsec tool. • Verify that the traffic from the peer is modified via packet capture. • Verify via TOE logs the connection between the TOE and Peer is terminated. • Verify the Peer is removed from the TOE Live Peer list.
Pass/Fail with Explanation	Pass. The TOE detects and discards the MKA traffic with modified packet payloads due to the integrity failure.

7.1.2.30 FCS_MKA_EXT.1.4 TEST #13A (MOD_MACSEC_V1.0) [TD0882]

Item	Data
Test Assurance Activity	The tests below require the TOE to be deployed in an environment with two MACsec-capable peers, identified as devices B and C, that the TOE can communicate with. Prior to performing these tests, the evaluator shall follow the steps in the guidance documentation to configure the TOE as the key server and principal actor (peer). The evaluator shall then perform the following tests using a traffic sniffer to capture this traffic: Test 13a: The evaluator shall configure the TOE to establish a MKA session with a new peer. The evaluator shall verify that the TOE sends a fresh SAK to the peer and sends other MKPDUs required for a new session. The evaluator shall verify from packet captures that MKPDUs are sent at least once every two seconds or every half-second, in accordance with the SFR selection. TD0882 has been applied.
Test Steps	• Configure the TOE and 2 peers for MACsec connections and enable MACsec on the TOE and Peer 1. • Verify via TOE logs and packet capture that the MACsec connection is established between the TOE and Peer 1. • Enable MACsec on Peer 2 and verify that the MACsec connection is established between the TOE and Peer 2. • Verify that both Peers are listed on the TOE. • Verify using packet capture that the TOE sends a fresh SAK to Peer 2 and sends other MKPDUs required for a new session. • Verify using packet capture that the MKPDUs are sent at least once every two seconds.

Pass/Fail with Explanation	Pass. The TOE sends a fresh SAK to the new peer and sends other MKPDUs required for a new session. The TOE also ensures that the MKPDUs are sent every two seconds during the MKA session.
-----------------------------------	--

7.1.2.31 FCS_MKA_EXT.1.4 TEST #13B (MOD_MACSEC_V1.0) [TD0882]

Item	Data
Test Assurance Activity	The tests below require the TOE to be deployed in an environment with two MACsec-capable peers, identified as devices B and C, that the TOE can communicate with. Prior to performing these tests, the evaluator shall follow the steps in the guidance documentation to configure the TOE as the key server and principal actor (peer). The evaluator shall then perform the following tests using a traffic sniffer to capture this traffic: Test 13b: (Conditional - If "EAPTLS with DevIDs" is selected in FCS_MACSEC_EXT.4.1) The evaluator shall use EAP-TLS to derive a CAK and configure the TOE's peer to send "0" in the MKA parameter field for MACsec Capability (Table 11-6 in 802.1X-2020). The evaluator shall observe that the peer is deleted from the connection after MKA Life Time has passed. TD0882 has been applied.
Pass/Fail with Explanation	NA. EAPTLS with DevIDs is not selected in the ST, hence this test is not applicable.

7.1.2.32 FCS_MKA_EXT.1.4 TEST #14A (MOD_MACSEC_V1.0) [TD0882]

Item	Data
Test Assurance Activity	The tests below require the TOE to be deployed in an environment with two MACsec-capable peers, identified as devices B and C, that the TOE can communicate with. Prior to performing these tests, the evaluator shall follow the steps in the guidance documentation to configure the TOE as the key server and principal actor (peer). The evaluator shall then perform the following tests using a traffic sniffer to capture this traffic: Test 14a: (Conditional - if any "group CAK" selection is made in FCS_MKA_EXT.1.5) The evaluator shall configure the TOE to send a fresh SAK with two peers as active participants. The evaluator shall verify that the TOE sends a fresh SAK to the peers and sends other MKPDUs required for a new session. The evaluator shall verify from packet captures that MKPDUs are sent at least once every two seconds or every half second in accordance with the SFR Selection. TD0882 has been applied.
Pass/Fail with Explanation	NA. Group CAK is not selected in the ST, hence this test is not applicable.

7.1.2.33 FCS_MKA_EXT.1.4 TEST #14B (MOD_MACSEC_V1.0) [TD0882]

Item	Data
Test Assurance Activity	The tests below require the TOE to be deployed in an environment with two MACsec-capable peers, identified as devices B and C, that the TOE can communicate with. Prior to performing these tests, the evaluator shall follow the steps in the guidance documentation to configure the TOE as the key server and principal actor (peer). The evaluator shall then perform the following tests using a traffic sniffer to capture this traffic: Test 14b: (Conditional - if any "group CAK" selection is made in FCS_MKA_EXT.1.5) Disconnect one of the peers. Arbitrarily introduce an artificial delay in sending a fresh SAK following the change in the Live Peer List. For this delayed fresh SAK, use a man-in-the-middle device to observe that the MKA Life Time of 6.0 seconds is enforced by the TSF. TD0882 has been applied.
Pass/Fail with Explanation	NA. Group CAK is not selected in the ST, hence this test is not applicable.

7.1.2.34 FCS_MKA_EXT.1.7 TEST #15 (MOD_MACSEC_V1.0) [TD0882]

Item	Data
Test Assurance Activity	The tests below require the TOE to be deployed in an environment with two MACsec-capable peers, identified as devices B and C, that the TOE can communicate with. Prior to performing these tests, the evaluator shall follow the steps in the guidance documentation to configure the TOE as the key server and principal actor (peer). The evaluator shall then perform the following tests using a traffic sniffer to capture this traffic: Test 15: (Conditional - if any "group CAK" selection is made in FCS_MKA_EXT.1.5) The evaluator shall perform the following steps: 1. Load one PSK onto the TOE and device B and a second PSK onto the TOE and device C. This defines two pairwise CAs. 2. Generate a group CAK for the group of three devices using ieee8021XKayCreateNewGroup. 3. Observe via packet capture that the TOE distributes the group CAK to the two peers, protected by AES key wrap using their respective PSKs. 4. Verify that B can form an SA with C and connect securely. 5. Disable the KaY functionality of device C using ieee8021XPaePortKayMkaEnable. 6. Generate a group CAK for the TOE and B using ieee8021XKayCreateNewGroup and observe they can connect. 7. The evaluator shall have B attempt to connect to C and observe this fails. 8. Re-enable the KaY functionality of device C. 9. Invoke ieee8021XKayCreateNewGroup again. 10. Verify that both the TOE can connect to C and that B can connect to C. TD0882 has been applied.
Pass/Fail with Explanation	NA. Group CAK is not selected in the ST, hence this test is not applicable.

7.1.2.35 FCS_MKA_EXT.1.7 TEST #16 (MOD_MACSEC_V1.0) [TD0889]

Item	Data
Test Assurance Activity	The tests below require the TOE to be deployed in an environment with two MACsec-capable peers, identified as devices B and C, that the TOE can communicate with. Prior to performing these tests, the evaluator shall follow the steps in the guidance documentation to configure the TOE as the key server and principal actor (peer). The evaluator shall then perform the following tests: Test 16: The evaluator shall start an MKA session between the TOE and an environmental MACsec peer and then perform the following steps: 1. Send an MKPDU to the TOE's individual MAC address from a peer. Verify the frame is dropped and logged. 2. Send an MKPDU to the TOE that is less than 32 octets long. Verify the frame is dropped and logged. 3. Send an MKPDU to the TOE whose length in octets is not a multiple of four. Verify the frame is dropped and logged. 4. Send an MKPDU to the TOE that is one byte short. Verify the frame is dropped and logged. 5. Send an MKPDU to the TOE with unknown Agility Parameter. Verify the frame is dropped and logged. TD0889 has been applied.
Test Steps	<u>MKPDU to the TOE's individual MAC address from a peer</u> • Start an MKA session between the TOE and peer by activating the MACsec channel on the TOE. • Send an MKPDU to the TOE's individual MAC address from a peer using the acumen-macsec tool. • Verify the frame is dropped using packet capture. • Verify that the connection between the TOE and Peer is terminated via TOE logs. • Verify the frame is dropped using TOE's output. <u>MKPDU that is less than 32 octets long</u> • Start an MKA session between the TOE and peer by activating the MACsec channel on the TOE. • Send an MKPDU to the TOE that is less than 32 octets long. • Verify the frame is dropped using packet capture. • Verify that the connection between the TOE and Peer is terminated via TOE logs. • Verify the frame is dropped using TOE's output. <u>MKPDU whose length in octets is not a multiple of four</u> • Start an MKA session between the TOE and peer by activating the MACsec channel on the TOE. • Send an MKPDU to the TOE whose length in octets is not a multiple of 4. • Verify the frame is dropped using packet capture. • Verify that the connection between the TOE and Peer is terminated via TOE logs. • Verify the frame is dropped using TOE's output.

	<u>MKPDU that is one byte short</u> • Start an MKA session between the TOE and peer by activating the MACsec channel on the TOE. • Send an MKPDU to the TOE that is one byte short. • Verify the frame is dropped using packet capture. • Verify the frame is dropped using TOE's output. • Verify that the connection between the TOE and Peer is terminated via TOE logs. <u>MKPDU with an unknown Agility Parameter.</u> • Start an MKA session between the TOE and peer by activating the MACsec channel on the TOE. • Send an MKPDU to the TOE with an unknown Agility Parameter. • Verify the frame is dropped using packet capture. • Verify the modified agility is logged into TOE using TOE's Output. • Verify that the connection between the TOE and Peer is terminated via TOE logs.
Pass/Fail with Explanation	Pass. The TOE drops and logs the following MKPDUs from the peer: • MPKDU received on the TOE's individual MAC address. • MPKDU that is less than 32 octets long. • MPKDU whose length in octets is not a multiple of four. • MPKDU that is one byte short. • MPKDU with an unknown Agility Parameter.

7.1.2.36 FCS_NTP_EXT.1.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The version of NTP selected in element 1.1 and specified in the ST shall be verified by observing establishment of a connection to an external NTP server known to be using the specified version(s) of NTP. This may be combined with tests of other aspects of FCS_NTP_EXT.1 as described below.
Test Steps	<u>NTP Version 4:</u> • Check current timestamp on the TOE. • Verify current timestamp on the NTP server. • Configure the TOE to sync time from the NTP server. • Verify the TOE logs for configuration of the new NTP server. • Verify the message digest algorithm keys configured on the NTP server. • Verify that the TOE syncs time with the NTP server. • Verify the updated timestamp on the TOE and ensure it is synced with the NTP server. • Verify the TOE logs for successful time synchronization with NTP server. • Verify the NTP version and successful synchronization with a packet capture.
Pass/Fail with Explanation	Pass. The TOE is able to sync time with an external NTP server using NTPv4. This meets the testing requirement.

7.1.2.37 FCS_NTP_EXT.1.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The cryptographic algorithms selected in element 1.2 and specified in the ST will have been specified in an FCS_COP SFR and tested in the accompanying Evaluation Activity for that SFR. Likewise, the cryptographic protocol selected in in element 1.2 and specified in the ST will have been specified in an FCS SFR and tested in the accompanying Evaluation Activity for that SFR. [conditional] If the message digest algorithm is claimed in element 1.2, the evaluator shall change the message digest algorithm used by the NTP server in such a way that the new value does not match the configuration on the TOE and confirms that the TOE does not synchronize to this time source. The evaluator shall use a packet sniffer to capture the network traffic between the TOE and the NTP server. The evaluator shall use the captured network traffic, to verify the NTP version, to observe time change of the TOE and uses the TOE's audit log to determine that the TOE accepted the NTP server's timestamp update. The captured traffic is also used to verify that the appropriate message digest algorithm was used to authenticate the time source and/or the appropriate protocol was used to ensure integrity of the timestamp that was transmitted in the NTP packets.
Test Steps	Valid SHA-1 • Check current timestamp on the TOE. • Verify current timestamp on the NTP server. • Verify that a supported SHA-1 message digest algorithm key is configured on the NTP server. • Configure the TOE to sync time from the NTP server with a supported SHA-1 message digest algorithm key. • Verify TOE is configured with a valid message digest algorithm key. • Verify the TOE logs for configuration of the NTP server authentication key. • Verify the TOE logs for configuration of the NTP server. • Verify that the TOE is able to sync time with the NTP server. • Verify the current timestamp on the TOE and ensure it is synced with the NTP server. • Verify the TOE logs for successful time synchronization with NTP server. • Verify successful time synchronization with the NTP server using packet capture. Invalid SHA-1 • Check current timestamp on the TOE. • Verify current timestamp on the NTP server. • Verify that an invalid message digest algorithm key is configured on the NTP server. • Verify TOE is configured with a valid message digest algorithm key. • Verify that the TOE is not able to sync time with the NTP server. • Verify the current timestamp on the TOE and ensure it is not synced with the NTP server. • Verify the TOE logs for unsuccessful time synchronization with NTP server. • Verify unsuccessful time synchronization with the NTP server using packet capture.

	Valid SHA-256 • Check current timestamp on the TOE. • Verify current timestamp on the NTP server. • Verify that a supported SHA-256 message digest algorithm key is configured on the NTP server. • Configure the TOE to sync time from the NTP server with a supported SHA-256 message digest algorithm key. • Verify TOE is configured with a valid message digest algorithm key. • Verify the TOE logs for configuration of the NTP server authentication key. • Verify the TOE logs for configuration of the NTP server. • Verify that the TOE is able to sync time with the NTP server. • Verify the current timestamp on the TOE and ensure it is synced with the NTP server. • Verify the TOE logs for successful time synchronization with NTP server. • Verify successful time synchronization with the NTP server using packet capture. Invalid SHA-256 • Check current timestamp on the TOE. • Verify current timestamp on the NTP server. • Verify that an invalid message digest algorithm key is configured on the NTP server. • Verify TOE is configured with a valid message digest algorithm key. • Verify that the TOE is not able to sync time with the NTP server. • Verify the current timestamp on the TOE and ensure it is not synced with the NTP server. • Verify the TOE logs for unsuccessful time synchronization with NTP server. • Verify unsuccessful time synchronization with the NTP server using packet capture.
Pass/Fail with Explanation	Pass. When the NTP server was configured with SHA-1 or SHA-256 authentication matching the TOE's configuration, the TOE successfully synchronized time using NTPv4. When the NTP server was configured with a modified message digest algorithm, the TOE did not synchronize time for NTPv4. This meets the testing requirements.

7.1.2.38 FCS_NTP_EXT.1.3 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure NTP server(s) to support periodic time updates to broadcast and multicast addresses. The evaluator shall confirm the TOE is configured to not accept broadcast and multicast NTP packets that would result in the timestamp being updated. The evaluator shall check that the time stamp is not updated after receipt of the broadcast and multicast packets.
Test Steps	<u>Broadcast:</u> • Check current timestamp on the TOE. • Verify current timestamp on the NTP server. • Configure NTP server to send broadcast packets.

	• Configure the TOE to sync time from the NTP server. • Verify the TOE logs for configuration of the new NTP server. • Verify that the TOE is unable to sync time with the NTP server. • Verify the current timestamp on the TOE and ensure it is not synced with the NTP server. • Verify the TOE logs for unsuccessful time synchronization with NTP server. • Verify unsuccessful time synchronization with the NTP server using packet capture when broadcast packets are received. <u>Multicast:</u> • Check current timestamp on the TOE. • Verify current timestamp on the NTP server. • Configure NTP server to send multicast packets. • Configure the TOE to sync time from the NTP server. • Verify the TOE logs for configuration of the new NTP server. • Verify that the TOE is unable to sync time with the NTP server. • Verify the current timestamp on the TOE and ensure it is not synced with the NTP server. • Verify the TOE logs for unsuccessful time synchronization with NTP server. • Verify unsuccessful time synchronization with the NTP server using packet capture when multicast packets are received.
Pass/Fail with Explanation	Pass. The TOE time stamp is not updated after receipt of the broadcast and multicast NTP packets from an NTP server. This meets the testing requirement.

7.1.2.39 FCS_NTP_EXT.1.4 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall confirm the TOE supports configuration of at least three (3) NTP time sources. The evaluator shall configure at least three NTP servers to support periodic time updates to the TOE. The evaluator shall confirm the TOE is configured to accept NTP packets that would result in the timestamp being updated from each of the NTP servers. The evaluator shall check that the time stamp is updated after receipt of the NTP packets. The purpose of this test to verify that the TOE can be configured to synchronize with multiple NTP servers. It is up to the evaluator to determine that the multi- source update of the time information is appropriate and consistent with the behaviour prescribed by the RFC 1305 for NTPv3 and RFC 5905 for NTPv4.
Test Steps	• Check current timestamp on the TOE. • Verify current timestamp on the NTP server. • Configure the TOE to sync time from 3 NTP server. • Verify the TOE logs for configuration of all the NTP servers. • Verify that the TOE synchronizes time with one of the configured NTP servers. • Verify the current timestamp on the TOE and ensure it is synced with the NTP server.

	• Verify the TOE logs for successful time synchronization with the NTP servers. • Delete the NTP server with which the TOE last synchronized. • Verify the TOE logs for deletion of the NTP server. • Verify that the TOE synchronizes time with one of the remaining NTP servers. • Verify the current timestamp on the TOE and ensure it is synced with the NTP server. • Remove the second NTP server with which the TOE synchronized. • Verify the TOE logs for deletion of the NTP server. • Verify that the TOE synchronizes time with the third NTP server. • Verify the current timestamp on the TOE and ensure it is synced with the NTP server. • Verify successful time synchronization with all the NTP servers using packet capture.
Pass/Fail with Explanation	Pass. The TOE can be configured to synchronize time with at least three NTP servers using NTPv4. This meets the testing requirement.

7.1.2.40 FCS_NTP_EXT.1.4 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: (The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers). The evaluator shall confirm that the TOE would not synchronize to other, not explicitly configured time sources by sending an otherwise valid but unsolicited NTP Server response indicating different time from the TOE's current system time. This rogue time source needs to be configured in a way (e.g. degrade or disable valid and configured NTP servers) that could plausibly result in unsolicited updates becoming a preferred time source if they are not discarded by the TOE. The TOE is not mandated to respond in a detectable way or audit the occurrence of such unsolicited updates. It is up to the evaluator to craft and transmit unsolicited updates in a way that would be consistent with the behaviour of a correctly-functioning NTP server.
Test Steps	• Check current timestamp on the TOE. • Verify accurate time on the NTP server. • Configure NTP server on the TOE. • Verify the TOE logs for configuration of the NTP server. • Verify that the TOE syncs time with the NTP server. • Verify the updated timestamp on the TOE and ensure it is synced with the NTP server. • Verify the TOE logs for successful time synchronization with the NTP server. • Verify successful time synchronization with the NTP server using packet capture. • Delete the NTP server configured on the TOE. • Verify the TOE logs for deletion of the NTP server. • Configure a different NTP server on the TOE.

	• Verify the TOE logs for configuration of the new NTP server. • Simultaneously replay packets from the first NTP server that were captured during the initial sync. • Verify with packet capture that replay packets are sent to the TOE, but it does not synchronize as it is no longer configured. • Verify that the TOE syncs time with the new NTP server. • Verify the updated timestamp on the TOE and ensure it is synced with the new NTP server. • Verify successful time synchronization with the new NTP server using packet capture.
Pass/Fail with Explanation	Pass. The TOE only accepts NTP time updates from configured NTP Servers and disregards updates from unconfigured ones. This meets the testing requirements.

7.1.2.41 FCS_RBG_EXT.1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. 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 shall verify 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 shall verify that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. Entropy input: the length of the entropy input value must equal the seed length. Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length. Personalization string: The length of the personalization string must be $\leq$ seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use

	personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied. Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.
Pass/Fail with Explanation	Algorithm: HMAC DRBG Mode: SHA2-512 Entropy Input: 256 Nonce: 128 Personalization String Length: 0-256 Increment 8 Additional Input: 8-256 Increment 8 Returned Bits: 1024 CAVP #: A7045 Pass. Based on these findings, this assurance activity is considered satisfied.

7.1.2.42 FCS_SSH_EXT.1.1 TEST #1 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	There are no test evaluation activities for this component. This SFR is evaluated by activities for other SFRs.
Pass/Fail with Explanation	NA. There are no test evaluation activities for this component. This SFR is evaluated by activities for other SFRs.

7.1.2.43 FCS_SSH_EXT.1.2 TEST #1A (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	Test 1: [conditional] If the TOE is acting as SSH Server: The evaluator shall use a suitable SSH Client to connect to the TOE, enable debug messages in the SSH Client, and examine the debug messages to determine that only the configured authentication methods for the TOE were offered by the server. If the Client supports only one authentication method, the evaluator can test this failure of connection by configuring the Server with an authentication method not supported by the Client. In order to facilitate this test, it is acceptable for the evaluator to configure an authentication method that is outside of the selections in the SFR.
Test Steps	<u>Public-Key Based:</u> SSH-RSA (2048-bit) - Generate a key pair on the client using the SSH-RSA algorithm with a 2048-bit key length and note its fingerprint. - Copy the generated public key to configure it on the TOE. - Configure the TOE user with the generated public key for SSH-RSA public key-based authentication.

- Attempt an SSH connection to the TOE and ensure it is successful.
- Verify via the SSH client logs that only the claimed authentication methods are being offered by the TOE.
- Verify the SSH client logs for the successful connection using public key authentication.
- Verify the TOE logs for a successful SSH connection.
- Verify the successful SSH connection using packet capture.

RSA-SHA2-256 (3072-bit)

- Generate a key pair on the client using the RSA-SHA2-256 algorithm with a 3072-bit key length and note its fingerprint.
- Copy the generated public key to configure it on the TOE.
- Configure the TOE user with the generated public key for RSA-SHA2-256 public key-based authentication.
- Attempt an SSH connection to the TOE and ensure it is successful.
- Verify the SSH client logs for the successful connection using public key authentication.
- Verify the TOE logs for a successful SSH connection.
- Verify the successful SSH connection using packet capture.

RSA-SHA2-512 (4096-bit)

- Generate a key pair on the client using the RSA-SHA2-512 algorithm with a 4096-bit key length and note its fingerprint.
- Copy the generated public key to configure it on the TOE.
- Configure the TOE user with the generated public key for RSA-SHA2-512 public key-based authentication.
- Attempt an SSH connection to the TOE and ensure it is successful.
- Verify the SSH client logs for the successful connection using public key authentication.
- Verify the TOE logs for a successful SSH connection.
- Verify the successful SSH connection using packet capture.

ECDSA-SHA2-NISTP256

- Generate a key pair on the client using the ECDSA-SHA2-NISTP256 algorithm and note its fingerprint.
- Copy the generated public key to configure it on the TOE.
- Configure the TOE user with the generated public key for ECDSA-SHA2-NISTP256 public key-based authentication.
- Attempt an SSH connection to the TOE and ensure it is successful.
- Verify the SSH client logs for the successful connection using public key authentication.
- Verify the TOE logs for a successful SSH connection.
- Verify the successful SSH connection using packet capture.

ECDSA-SHA2-NISTP384

- Generate a key pair on the client using the ECDSA-SHA2-NISTP384 algorithm and note its fingerprint.
- Copy the generated public key to configure it on the TOE.
- Configure the TOE user with the generated public key for ECDSA-SHA2-NISTP384 public key-based authentication.
- Attempt an SSH connection to the TOE and ensure it is successful.
- Verify the SSH client logs for the successful connection using public key authentication.
- Verify the TOE logs for a successful SSH connection.
- Verify the successful SSH connection using packet capture.

	ECDSA-SHA2-NISTP521 • Generate a key pair on the client using the ECDSA-SHA2-NISTP521 algorithm and note its fingerprint. • Copy the generated public key to configure it on the TOE. • Configure the TOE user with the generated public key for ECDSA-SHA2-NISTP521 public key-based authentication. • Attempt an SSH connection to the TOE and ensure it is successful. • Verify the SSH client logs for the successful connection using public key authentication. • Verify the TOE logs for a successful SSH connection. • Verify the successful SSH connection using packet capture. <u>Password Based:</u> • Configure a TOE user for password-based authentication. • Attempt to log into the TOE using password-based authentication with correct credentials and ensure the connection being successful. • Verify the SSH client logs for the successful connection using password-based authentication. • Verify the TOE logs for successful SSH connection. • Verify the successful SSH connection using packet capture. <u>Keyboard interactive:</u> • Ensure the TOE user is configured with a password for keyboard-interactive authentication. • Attempt to log into the TOE via keyboard-interactive authentication and ensure the connection being successful. • Verify the SSH client logs for the successful connection using keyboard-interactive authentication. • Verify the TOE logs for successful SSH connection. • Verify the successful SSH connection using packet capture.
Pass/Fail with Explanation	Pass. The remote client is able to establish a successful SSH connection with the TOE using each one of the claimed authentication methods: public key algorithms (SSH-RSA, RSA-SHA2-256, RSA-SHA2-512, ECDSA-SHA2-NISTP256, ECDSA-SHA2-NISTP384, ECDSA-SHA2-NISTP521), password-based authentication and keyboard-interactive authentication. This meets the testing requirements.

7.1.2.44 FCS_SSH_EXT.1.2 TEST #1B (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	Test 1: [conditional] If the TOE is acting as SSH Server: [conditional] If the SSH server supports X509 based Client authentication options: - The evaluator shall initiate an SSH session from a client where the username is associated with the X509 certificate. The evaluator shall verify the session is successfully established. - Next the evaluator shall use the same X509 certificate as above but include a username not associated with the certificate. The evaluator shall verify that the session does not establish. - Finally, the evaluator shall use the correct username (from step a above) but use a different X509 certificate which is not associated with the username. The evaluator shall verify that the session does not establish.

	If the Client supports only one authentication method, the evaluator can test this failure of connection by configuring the Server with an authentication method not supported by the Client. In order to facilitate this test, it is acceptable for the evaluator to configure an authentication method that is outside of the selections in the SFR.
Pass/Fail with Explanation	NA. The TOE does not support X509 based client authentication for SSH server, hence this test is not applicable.

7.1.2.45 FCS_SSH_EXT.1.2 TEST #2 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	Test 2: [conditional] If the TOE is acting as SSH Client, the evaluator shall test for a successful configuration setting of each authentication method as follows: - The evaluator shall initiate a SSH session using the authentication method configured and verify that the session is successfully established. - Next, the evaluator shall use bad authentication data (e.g. incorrectly generated certificate or incorrect password) and ensure that the connection is rejected. Steps a-b shall be repeated for each independently configurable authentication method supported by the server. If the Client supports only one authentication method, the evaluator can test this failure of connection by configuring the Server with an authentication method not supported by the Client. In order to facilitate this test, it is acceptable for the evaluator to configure an authentication method that is outside of the selections in the SFR.
Pass/Fail with Explanation	NA. The TOE does not operate as an SSH Client, hence this test is not applicable.

7.1.2.46 FCS_SSH_EXT.1.2 TEST #3 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	Test 3: [conditional] If the TOE is acting as SSH Client, the evaluator shall verify that the connection fails upon configuration mismatch as follows: - The evaluator shall configure the Client with an authentication method not supported by the Server. - The evaluator shall verify that the connection fails. If the Client supports only one authentication method, the evaluator can test this failure of connection by configuring the Server with an authentication method not supported by the Client. In order to facilitate this test, it is acceptable for the evaluator to configure an authentication method that is outside of the selections in the SFR.

Pass/Fail with Explanation	NA. The TOE does not operate as an SSH Client, hence this test is not applicable.
-----------------------------------	---

7.1.2.47 FCS_SSH_EXT.1.3 TEST #1 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	Test 1: The evaluator shall demonstrate that the TOE accepts the maximum allowed packet size.
Test Steps	<u>Note:</u> Two-phased testing is needed due to padding rules for the SSH protocol: with s=262140 (slightly less than the maximum allowed size of 256k bytes), the total is 262144 (mod 16 = 0) and the TOE accepts it; with s=262144 (256k bytes i.e. maximum allowed size), the total is 262148 (mod 16 !=0) and the TOE flags invalid padding. <u>Testing with 262140 bytes</u> • Using acumen-sshs tool start an SSH session with the TOE and send a packet with the maximum allowed size of 256K bytes. • Verify with the logs that the TOE accepts the connection. • Verify the successful connection with the packet capture. <u>Testing with 262144 bytes</u> • Using acumen-sshs tool start an SSH session with the TOE and send a packet with the maximum allowed size of 256K bytes. • Verify with the logs that the TOE accepts the connection. • Verify the successful connection with the packet capture.
Pass/Fail with Explanation	Pass. TOE accepts the maximum allowed packet size of 256k bytes. This meets the testing requirement.

7.1.2.48 FCS_SSH_EXT.1.3 TEST #2 (PKG_SSH_V1.0) [TD0732]

Item	Data
Test Assurance Activity	Test 2: This test is performed to verify that the TOE drops packets that are larger than size specified in the component. - The evaluator shall establish a successful SSH connection with the peer. - Next the evaluator shall craft a packet that is slightly larger than the maximum size specified in this component and send it through the established SSH connection to the TOE. The packet should not be greater than the maximum packet size + 16 bytes. If the packet is larger, the evaluator shall justify the need to send a larger packet. - The evaluator shall verify that the packet was dropped by the TOE. The method of verification will vary by the TOE. Examples include reviewing the TOE audit log for a dropped packet audit or observing the TOE terminates the connection. TD0732 has been applied.
Test Steps	• Using the acumen-sshs tool start an SSH session with the TOE and send packets slightly larger (+1 byte) than the maximum packet size of 256k bytes.

	• Verify that the TOE closes the connection when it receives the packet with bad packet length using TOE logs. • Verify that the TOE disconnects the SSH session using packet capture.
Pass/Fail with Explanation	Pass. The TOE drops the connection when the packet size is larger than 256k bytes. This meets the testing requirement.

7.1.2.49 FCS_SSH_EXT.1.4 TEST #1 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests. If the TOE can be both a client and a server, these tests must be performed for both roles. Test 1: The evaluator must ensure that only claimed algorithms and cryptographic primitives are used to establish an SSH connection. To verify this, the evaluator shall establish an SSH connection with a remote endpoint. The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g. using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers only the algorithms defined in the ST for the TOE for SSH connections. The evaluator shall perform one successful negotiation of an SSH connection and verify that the negotiated algorithms were included in the advertised set. If the evaluator detects that not all algorithms defined in the ST for SSH are advertised by the TOE or the TOE advertises additional algorithms not defined in the ST for SSH, the test shall be regarded as failed. The data collected from the connection above shall be used for verification of the advertised hashing and shared secret establishment algorithms in FCS_SSH_EXT.1.5 and FCS_SSH_EXT.1.6 respectively.
Test Steps	• Configure the TOE to support AES128-CTR, AES256-CTR, AES128-CBC, AES256-CBC, aes128-gcm@openssh.com, aes256-gcm@openssh.com. • Attempt to establish a SSH connection with the TOE and ensure it is successful. • Verify the successful connection attempt with audit log evidence. • Verify that all ciphers claimed in the ST are supported by the TOE using the Packet capture evidence.
Pass/Fail with Explanation	Pass. The TOE supports successful SSH negotiations when using the claimed encryption algorithms i.e. AES128 CTR, AES256 CTR, AES128 CBC, AES256 CBC, aes128-gcm@openssh.com and aes256-gcm@openssh.com. This meets the testing requirements.

7.1.2.50 FCS_SSH_EXT.1.4 TEST #2 (PKG_SSH_V1.0)

Item	Data
------	------

Test Assurance Activity	The evaluator shall perform the following tests. If the TOE can be both a client and a server, these tests must be performed for both roles. Test 2: For the connection established in Test 1, the evaluator shall terminate the connection and observe that the TOE terminates the connection.
Test Steps	<u>Note</u>: Only SSH server functionality is claimed for the TOE. • Terminate the connection established in FCS_SSH_EXT.1.4 TEST #1. • Verify that the TOE terminates the SSH connection using TOE logs. • Verify that the TOE terminates the SSH connection using packet capture.
Pass/Fail with Explanation	Pass. The TOE terminates the connection in response to a client-initiated termination. This meets the testing requirement.

7.1.2.51 FCS_SSH_EXT.1.4 TEST #3 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests. If the TOE can be both a client and a server, these tests must be performed for both roles. Test 3: The evaluator shall configure the remote endpoint to only allow a mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the attempt fails.
Test Steps	<u>Note</u>: Only SSH server functionality is claimed for the TOE. • Verify the ciphers supported by the TOE. • Attempt a connection to the TOE using an unsupported cipher (aes192-ctr) and observe that it fails. • Verify that the SSH session was not established via TOE logs. • Verify that the SSH session was not established via packet capture.
Pass/Fail with Explanation	Pass. The TOE does not establish the connection when an unsupported cipher is offered. This meets the testing requirement.

7.1.2.52 FCS_SSH_EXT.1.5 TEST #1 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	Test 1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.
Test Steps	• Verify that only the hashing algorithms claimed in the ST are supported by the TOE using the Packet capture evidence from FCS_SSH_EXT.1.4 Test #1.

Pass/Fail with Explanation	Pass. The TOE offers only the MAC algorithms hmac-sha2-256 and hmac-sha2-512 as claimed in the ST. This meets the testing requirements.

7.1.2.53 FCS_SSH_EXT.1.5 TEST #2 (PKG_SSH_V1.0) [TD1023]

Item	Data
Test Assurance Activity	Test 2: [conditional] If aes*-gcm@openssh.com is not being negotiated, the evaluator shall configure an SSH peer to allow only a hashing algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.
Test Steps	• Verify the hashing algorithms supported by the TOE. • Attempt a connection to the TOE using an unsupported hashing (HMAC-MD5) algorithm and observe that it fails to connect. • Verify that the SSH session was not established via TOE logs. • Verify that the SSH session was not established via packet capture.
Pass/Fail with Explanation	Pass. The TOE does not establish the connection when an unsupported hashing algorithm is offered. This meets the testing requirements.

7.1.2.54 FCS_SSH_EXT.1.6 TEST #1 (PKG_SSH_V1.0) [TD0967]

Item	Data
Test Assurance Activity	Test 1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised. Note that it is permissible for the TOE to advertise the "kex-strict-c-v00@openssh.com" or "kex-strict-s-v00@openssh.com" value alongside the other key exchange mechanisms. These values are associated with client and server, respectively. TD0967 has been applied.
Test Steps	• Verify that only the claimed key exchange methods in the ST are supported by the TOE using the Packet capture evidence from FCS_SSH_EXT.1.4 Test #1.
Pass/Fail with Explanation	Pass. The TOE offers only the key exchange methods ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521 as claimed in the ST. This meets the testing requirements.

7.1.2.55 FCS_SSH_EXT.1.6 TEST #2 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	Test 2: The evaluator shall configure an SSH peer to allow only a key exchange method that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.

Test Steps	• Verify the key exchange methods supported by the TOE. • Attempt a connection to the TOE using an unsupported key exchange method (dh-group14-sha1) and observe that it fails to connect. • Verify that the SSH session was not established via TOE logs. • Verify that the SSH session was not established via packet capture.
Pass/Fail with Explanation	Pass. The TOE does not permit connections when an unsupported key exchange method is offered. This meets the testing requirement.

7.1.2.56 FCS_SSH_EXT.1.6 TEST #3 (PKG_SSH_V1.0) [TD0967]

Item	Data
Test Assurance Activity	Test 3: The evaluator shall configure the SSH peer to omit the kex-strict value from the kex_algorithms field. The evaluator shall verify that the TOE is able to successfully establish a connection with the SSH peer. TD0967 has been applied.
Test Steps	• Use the acumen-ssh tool to attempt an SSH connection with the TOE while omitting the kex-strict value from the kex_algorithms field and ensure that it is successful. • Verify the successful connection with TOE audit log. • Examine the captured network traffic (PCAP) to confirm that the SSH peer omits the kex-strict value in the kex_algorithms field during key exchange.
Pass/Fail with Explanation	Pass. The TOE successfully establishes the SSH connection when the peer omits the kex-strict value from the kex_algorithms field. This meets the testing requirement.

7.1.2.57 FCS_SSH_EXT.1.6 TEST #4 (PKG_SSH_V1.0) [TD0967]

Item	Data
Test Assurance Activity	Test 4: The evaluator shall configure the SSH peer to include the kex-strict value in the kex_algorithms field. The evaluator shall verify that the TOE is able to successfully establish a connection with the SSH peer. TD0967 has been applied.
Test Steps	• Attempt an SSH connection with the TOE including the kex-strict value in the kex_algorithms field and ensure that it is successful. • Verify the successful connection with TOE audit log. • Examine the captured network traffic (PCAP) to confirm that the SSH peer advertises the kex-strict value in the kex_algorithms field during key exchange.
Pass/Fail with Explanation	Pass. The TOE successfully establishes the SSH connection when the peer includes the kex-strict value in the kex_algorithms field. This meets the testing requirement.

7.1.2.58 FCS_SSH_EXT.1.7 TEST #1 (PKG_SSH_V1.0)

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

7.1.2.59 FCS_SSH_EXT.1.8 TEST #1 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	The test harness needs to be configured so that its connection rekey or termination limits are greater than the limits supported by the TOE -- it is expected that the test harness should not be initiating the connection rekey or termination. Test 1: Establish an SSH connection. Wait until the identified connection rekey limit is met. Observed that a connection rekey or termination is initiated. This may require traffic to periodically be sent, or connection keep alive to be set, to ensure that the connection is not closed due to an idle timeout.
Test Steps	• Verify the rekeying time limit on the SSH client. • Configure the time limit for rekeying on the TOE to 10 minutes. • Verify with the logs that the rekey is configured. • Establish an SSH session with the TOE. • Note the login time via TOE logs. • Wait for 10 minutes and verify that the rekey occurs, initiated by the TOE as per the configured time-based rekey threshold.
Pass/Fail with Explanation	Pass. The TOE initiates a rekey after the specified time limit. This meets the testing requirements.

7.1.2.60 FCS_SSH_EXT.1.8 TEST #2 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	The test harness needs to be configured so that its connection rekey or termination limits are greater than the limits supported by the TOE -- it is expected that the test harness should not be initiating the connection rekey or termination. Test 2: Establish an SSH connection. Transmit data from the TOE until the identified connection rekey or termination limit is met. Observe that a connection rekey or termination is initiated.
Test Steps	• Verify the volume limit for rekeying on the SSH client. • Configure the volume limit for rekeying on the TOE to 100MB. • Verify with TOE logs that the rekey is configured. • Verify a 110 MB file is available for transfer on the TOE.

	• Initiate a connection from the VM to the TOE and transfer data from the TOE. • Wait for the 110 MB file to transfer and verify that the rekey is initiated by the TOE. • Verify the session using TOE logs.
Pass/Fail with Explanation	Pass. The TOE initiates a rekey after sending the specified amount of data traffic. This meets the testing requirement.

7.1.2.61 FCS_SSH_EXT.1.8 TEST #3 (PKG_SSH_V1.0)

Item	Data
Test Assurance Activity	The test harness needs to be configured so that its connection rekey or termination limits are greater than the limits supported by the TOE -- it is expected that the test harness should not be initiating the connection rekey or termination. Test 3: Establish an SSH connection. Send data to the TOE until the identified connection rekey limit or termination is met. Observe that a connection rekey or termination is initiated.
Test Steps	• Verify the volume limit for rekeying on the SSH client. • Configure the volume limit for rekeying on the TOE to 100MB. • Verify with TOE logs that the rekey is configured. • Verify a 110 MB file is available for transfer on the SSH client. • Send data to the TOE. • Wait for the 100 MB file to transfer and verify that the rekey is initiated by the TOE. • Verify the session using TOE logs.
Pass/Fail with Explanation	Pass. The TOE initiates a rekey when it receives a specified amount of data traffic. This meets the testing requirement.

7.1.2.62 FCS_SSHS_EXT.1 TEST #1 (PKG_SSH_V1.0) [TD0682]

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall use a suitable SSH Client to connect to the TOE and examine the list of server host key algorithms in the SSH_MSG_KEXINIT packet sent from the server to the client to determine that only the configured server authentication methods for the TOE were offered by the server. TD0682 has been applied.
Test Steps	• Verify the Host key algorithms supported by the TOE. • Attempt to establish an SSH connection with the TOE and ensure it is successful. • Verify via SSH client logs that the host key algorithms offered by the TOE match with the ones claimed in the ST.

	• Verify the successful connection attempt with TOE logs. • Verify that all host key algorithms claimed in the ST are supported by the TOE using the Packet capture evidence.
Pass/Fail with Explanation	Pass. The TOE only offers the host key algorithms claimed in the ST. This meets the testing requirements.

7.1.2.63 FCS_SSHS_EXT.1 TEST #2 (PKG_SSH_V1.0) [TD0682]

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall test for a successful configuration setting of each server authentication method as follows. The evaluator shall initiate a SSH session using the authentication method configured and verify that the session is successfully established. Repeat this process for each independently configurable server authentication method supported by the server. TD0682 has been applied.
Test Steps	• Verify the Host key algorithms supported by the TOE. • Attempt an SSH connection to the TOE using the ssh-rsa host key algorithm and ensure it is successful. • Verify the successful SSH connection using TOE logs. • Verify the successful SSH connection using the ssh-rsa host key algorithm via packet capture. • Attempt an SSH connection to the TOE using the rsa-sha2-256 host key algorithm and ensure it is successful. • Verify the successful SSH connection using TOE logs. • Verify the successful SSH connection using the rsa-sha2-256 host key algorithm via packet capture. • Attempt an SSH connection to the TOE using the rsa-sha2-512 host key algorithm and ensure it is successful. • Verify the successful SSH connection using TOE logs. • Verify the successful SSH connection using the rsa-sha2-512 host key algorithm via packet capture. • Attempt an SSH connection to the TOE using the ecdsa-sha2-nistp256 host key algorithm and ensure it is successful. • Verify the successful SSH connection using TOE logs. • Verify the successful SSH connection using the ecdsa-sha2-nistp256 host key algorithm via packet capture. • Attempt an SSH connection to the TOE using the ecdsa-sha2-nistp384 host key algorithm and ensure it is successful. • Verify the successful SSH connection using TOE logs.

	• Verify the successful SSH connection using the ecdsa-sha2-nistp384 host key algorithm via packet capture. • Attempt an SSH connection to the TOE using the ecdsa-sha2-nistp521 host key algorithm and ensure it is successful. • Verify the successful SSH connection using TOE logs. • Verify the successful SSH connection using the ecdsa-sha2-nistp521 host key algorithm via packet capture.
Pass/Fail with Explanation	Pass. The TOE allows the client to connect using all the supported host key algorithms ssh-rsa, rsa-sha2-256 and rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521. This meets the testing requirement.

7.1.2.64 FCS_SSHS_EXT.1 TEST #3 (PKG_SSH_V1.0) [TD0682]

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 3: The evaluator shall configure the peer to only allow an authentication mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the TOE sends a disconnect message. TD0682 has been applied.
Test Steps	• Verify the Host key algorithms supported by the TOE. • Attempt an SSH connection to the TOE using unsupported host key algorithm ssh-dss and ensure it is rejected. • Verify through logs that the SSH session was not established. • Verify through packet capture that the SSH session was not established.
Pass/Fail with Explanation	Pass. TOE rejects the connection if the session is attempted using an unsupported host key algorithm. This meets the testing requirement.

7.1.3 IDENTIFICATION AND AUTHENTICATION (FIA)

7.1.3.1 FIA_AFL.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application): Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE (and, if the time period selection in

	FIA_AFL.1.2 is included in the ST, then the evaluator shall also use the operational guidance to configure the time period after which access is reenabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.
Test Steps	• Configure the unsuccessful authentication attempts threshold and lockout period. • Start an SSH session with the TOE and attempt to login with the wrong password and lock the user. • Verify the user is locked out for configured time with logs. • Attempt to open another connection and attempt to login with valid password before the lockout period expires. • Verify with logs the attempt failed due to locked out account.
Pass/Fail with Explanation	Pass. The TOE does not allow authentication once the unsuccessful authentication attempt limit has been reached. This meets the testing requirement.

7.1.3.2 FIA_AFL.1 TEST #2A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application): Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows. If the administrator action selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote administrator's access results in successful access (when using valid credentials for that administrator).
Test Steps	• Configure the unsuccessful authentication attempts threshold and lockout period. • Start an SSH session with the TOE and attempt to login with wrong passwords and lockout the user. • Verify with logs that the user has been locked out. • Confirm that an attempt to establish an SSH session with correct credentials prior to the lockout period expiry or administrative action failed. • Verify that the user gets unlocked after the admin action. • Confirm that an attempt to establish an SSH session with correct credentials succeeds now.
Pass/Fail with Explanation	Pass. Once the account is locked, the TOE does not allow authentication before the lockout period elapses unless the administrator has unlocked the account. This meets the testing requirement.

7.1.3.3 FIA_AFL.1 TEST #2B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application): Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows. If the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall wait for just less than the time period configured in Test 1 and show that an authorisation attempt using valid credentials does not result in successful access. The evaluator shall then wait until just after the time period configured in Test 1 and show that an authorisation attempt using valid credentials results in successful access
Test Steps	• Configure the unsuccessful authentication attempts threshold and lockout period. • Start an SSH session with the TOE and attempt to login with wrong passwords and lockout the user. • Verify with logs that the user has been locked out. • Verify that the lockout time matches the configured lockout period. • Confirm that an attempt to establish an SSH session with correct credentials just before the end of the lockout period fails. • Verify that the user gets unlocked after the end of the lockout period. • Confirm that an attempt to establish an SSH session with correct credentials just after the end of the lockout period succeeds.
Pass/Fail with Explanation	Pass. The TOE does not allow authentication until the configured lock-out time period has expired. Once the configured time period has elapsed, the TOE allows the administrator to login. This meets the testing requirement.

7.1.3.4 FIA_PMG_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests. Test 1: The evaluator shall compose passwords that 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, and a minimum length listed in the requirement are supported and justify the subset of those characters chosen for testing
Test Steps	• Set the minimum password requirements. ○ Minimum 15-character length. ○ Minimum 4 upper case. ○ Minimum 4 lower case. ○ Minimum 3 digits. ○ Minimum 3 special characters. • Create below users having 15-character password with combination of uppercase, lowercase, numbers and characters. ○ username: good001 & password: AB1CD7+!a@bc1d, ○ username: good002 & password: FG2HI8-#f\$gh2i. ○ username: good003 & password: KL3MN9/%k^lm3n:

	○ username: good004 & password: PQ4RS0;&p*qr4s< ○ username: good005 & password: UV5WX1"(u'vw5x> ○ username: good006 & password: ZT6B2D)z=_b6cd ○ username: good007 & password: E1Jj?[Oo8]\tY4y ○ username: good008 & password: Aa3b`{5cD}9~ETe • Show that all the users were created. • Verify that all the users were created using TOE logs.
Pass/Fail with Explanation	Pass. The TOE allows configuring passwords meeting the requirements. This meets the testing requirement.

7.1.3.5 FIA_PMG_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests. Test 2: The evaluator shall compose passwords that do not meet the requirements in some way. For each password, the evaluator shall verify that the TOE does not support the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that the TOE enforces the allowed characters and the minimum length listed in the requirement and justify the subset of those characters chosen for testing.
Test Steps	• Set the minimum password requirements. ○ Minimum 15-character length. ○ Minimum 4 upper case. ○ Minimum 4 lower case. ○ Minimum 3 digits. ○ Minimum 3 special characters. • Create below having passwords not meeting the configured requirements. ○ Username: bad001 and Password: ef1gh7+!a@bc1d, ○ Username: bad002 and Password: FG2HI8-#A\$BC2D. ○ Username: bad003 and Password: KLaMNb/%k^Imcn: ○ Username: bad004 and Password: PQ4RS012p3qr4s5 ○ Username: bad005 and Password: UV5WX1=(u)vw5x
Pass/Fail with Explanation	Pass. The TOE does not allow to create a user with a weak password that does not meet the password combination requirements. This meets the test requirement.

7.1.3.6 FIA_PSK_EXT.1 TEST #17 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be performed with a single test case. Test 17: (conditional, the TOE supports PSKs of multiple lengths) The evaluator shall use the minimum length, the maximum length, a length inside the allowable range, and invalid lengths

	beyond the supported range (both higher and lower). The minimum, maximum, and included length tests should be successful, and the invalid lengths must be rejected by the TOE.
Test Steps	For AES-128, the CAK should be 32 hexadecimal characters. <u>Minimum length:</u> • Configure the TOE for MACsec with a PSK that is 1-character long. • Verify via TOE logs the PSK was created. • Verify the successful MACsec connection via packet capture and TOE logs. <u>Maximum Length:</u> • Configure the TOE for MACsec with a PSK that is 32-character long. • Verify via TOE logs the PSK was created. • Verify the successful MACsec connection via packet capture and TOE logs. <u>Length in Allowable Range:</u> • Configure the TOE for MACsec with a PSK that is 16-character long. • Verify via TOE logs the PSK was created. • Verify the successful MACsec connection via packet capture and TOE logs. <u>Length Beyond Supported Range (Lower):</u> • Attempt to configure the TOE for MACsec with a PSK that is 0-character long i.e. blank PSK and verify that the TOE rejects it. <u>Length Beyond Supported Range (Higher):</u> • Attempt to configure the TOE for MACsec with a PSK that is 33-character long i.e. Length more than supported PSK range and verify that the TOE rejects it. For AES-256, the CAK should be 64 hexadecimal characters. <u>Minimum length:</u> • Configure the TOE for MACsec with a PSK that is 1-character long. • Verify via TOE logs the PSK was created. • Verify the successful MACsec connection via packet capture and TOE logs. <u>Maximum Length:</u> • Configure the TOE for MACsec with a PSK that is 64-character long. • Verify via TOE logs the PSK was created. • Verify the successful MACsec connection via packet capture and TOE logs. <u>Length in Allowable Range:</u> • Configure the TOE for MACsec with a PSK that is 48-character long. • Verify via TOE logs the PSK was created. • Verify the successful MACsec connection via packet capture and TOE logs. <u>Length Beyond Supported Range (Lower):</u> • Attempt to configure the TOE for MACsec with a PSK that is 0-character long i.e. blank PSK and verify that the TOE rejects it. <u>Length Beyond Supported Range (Higher):</u> • Attempt to configure the TOE for MACsec with a PSK that is 65-character long i.e. Length more than supported PSK range and verify that the TOE rejects it.

Pass/Fail with Explanation	Pass. The TOE accepts the configuration of PSKs having supported lengths while rejecting the configuration of PSKs with unsupported lengths.
-----------------------------------	--

7.1.3.7 FIA_PSK_EXT.1 TEST #18 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be performed with a single test case. Test 18: (conditional, the TOE does not generate bit-based PSKs) The evaluator shall obtain a bit-based PSK of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.
Pass/Fail with Explanation	Pass. Testing for supported PSK lengths is covered in FIA_PSK_EXT.1 TEST #17.

7.1.3.8 FIA_PSK_EXT.1 TEST #19 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be performed with a single test case. Test 19: (conditional, the TOE can generate bit-based PSKs) The evaluator shall generate a bit-based PSK of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.
Pass/Fail with Explanation	NA. AS per ST, The TOE cannot generate bit-based PSK.

7.1.3.9 FIA_UIA_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For all combinations of supported credentials and login methods, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

Test Steps	<u>Local (password-based)</u> • Attempt to log into the TOE via console with bad credentials; this should fail with appropriate TOE logs. • Attempt to log into the TOE via console with good credentials; this should succeed with appropriate TOE logs. <u>Remote (password-based)</u> • Log into the TOE via SSH with bad credentials; this should fail. • Log into the TOE via SSH with good credentials; this should succeed. • Verify audit logs reflect both attempts. <u>Remote (Public key-based)</u> • Log into the TOE via SSH with bad credentials; this should fail. • Log into the TOE via SSH with good credentials; this should succeed. • Verify audit logs reflect both attempts.
Pass/Fail with Explanation	Pass. • The TOE denies local (password-based) user authentication via serial console with incorrect credentials but successfully authenticates with correct credentials. • The TOE denies remote (password-based) user authentication via SSH with incorrect credentials but successfully authenticates with correct credentials. • The TOE denies remote (public key-based) user authentication via SSH with incorrect credentials but successfully authenticates with correct credentials.

7.1.3.10 FIA_UIA_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.
Test Steps	<u>Remote (password-based)</u> • Show that the Ping is allowed prior to authentication. • Show that commands are not available prior to login. • Verify authentication logs reflect failure. • Verify that only the login banner was displayed and observe that TOE allows the establishment of an SSH session with a remote management station prior to the identification and authentication process. • Login into the TOE. • Show that the previously enabled commands are now available. • Verify authentication logs reflect success. <u>Remote (public key-based)</u> • Show that the Ping is allowed prior to authentication. • Show that commands are not available prior to login. • Verify authentication logs reflect failure.

	• Verify that only the login banner was displayed and observe that TOE allows the establishment of SSH session with remote management station prior to the identification and authentication process. • Login into the TOE. • Show that the previously enabled commands are now available. • Verify authentication logs reflect success.
Pass/Fail with Explanation	Pass. • The TOE only displays the warning banner and replies to the ping requests before authenticating the remote (password-based) SSH user. • The TOE only displays the warning banner and replies to the ping requests before authenticating the remote (public key-based) SSH user.

7.1.3.11 FIA_UIA_EXT.1 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.
Test Steps	***NOTE: The "ICMP echo reply" claim in the ST is only applicable for remote access. • Show that commands are not available prior to login. • Verify authentication logs reflect failure. • Verify that only the login banner was displayed. • Login into the TOE. • Show that the previously enabled commands are now available. • Verify authentication logs reflect success.
Pass/Fail with Explanation	Pass. The TOE only displays the warning banner before authenticating the local console user using password. This meets the testing requirement.

7.1.3.12 FIA_UIA_EXT.1 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.
Pass/Fail with Explanation	NA. The TOE is not distributed, hence this test is not applicable.

7.1.3.13 FIA_UAU.7 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each method of local login allowed: Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.
Test Steps	• Connect to the TOE via console with incorrect authentication credentials and verify that the most obscured feedback is provided. • Verify the failed login attempt using TOE logs. • Connect to the TOE via console with correct authentication credentials and verify that the most obscured feedback is provided. • Verify the successful login attempt using TOE logs.
Pass/Fail with Explanation	Pass. • TOE gives obscured feedback after entering the incorrect credentials during local console login using password. • TOE gives obscured feedback after entering the correct credentials during local console login using password.

7.1.4 SECURITY MANAGEMENT (FMT)

7.1.4.1 FMT_MOF.1/ FUNCTIONS TEST #1A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	If 'transmission of audit data to external IT entity' is selected from the second selection together with 'modify the behaviour of' in the first selection The evaluator shall perform the following tests: Test 1: The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to modify the security related parameters can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.
Test Steps	• Log into the TOE as a lower privileged user. • Make sure that "tester" is a lower privileged user. • Attempt to modify the parameters involved with the syslog server and verify the command is rejected.

Pass/Fail with Explanation	Pass. TOE prevents the non-security administrator from modifying security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity. This meets the testing requirement.
-----------------------------------	--

7.1.4.2 FMT_MOF.1/ FUNCTIONS TEST #2A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	If 'transmission of audit data to external IT entity' is selected from the second selection together with 'modify the behaviour of' in the first selection The evaluator shall perform the following tests: Test 2: The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity with prior authentication as Security Administrator. The effects of the modifications should be confirmed. The evaluator does not have to test all possible values of the security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity but at least one allowed value per parameter.
Pass/Fail with Explanation	Pass. This testing is covered by the requirements in FAU_STG_EXT.1 Test #1.

7.1.4.3 FMT_MOF.1/ FUNCTIONS TEST #1B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	If 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection The evaluator shall perform the following tests: Test 1: The evaluator shall try to modify all security related parameters for configuration of the handling of audit data without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator. The term 'handling of audit data' refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.4, FAU_STG_EXT.1.5 and FAU_STG_EXT.2.
Test Steps	- Log into the TOE as a lower privileged user. - Make sure that "tester" is a lower privileged user.

	Attempt to modify the parameters for configuration of the handling of audit data and verify the command is rejected.
Pass/Fail with Explanation	Pass. TOE prevents non-security administrators from modifying the security-related parameters for configuration of the handling of audit data. This meets the testing requirement.

7.1.4.4 FMT_MOF.1/ FUNCTIONS TEST #2B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	If 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection The evaluator shall perform the following tests: Test 2: The evaluator shall try to modify all security related parameters for configuration of the handling of audit data with prior authentication as Security Administrator. The effects of the modifications should be confirmed. The term 'handling of audit data' refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.4, FAU_STG_EXT.1.5 and FAU_STG_EXT.2. The evaluator does not necessarily have to test all possible values of the security related parameters for configuration of the handling of audit data but at least one allowed value per parameter.
Test Steps	- Login to the TOE as an admin. - Make sure that "admin" is a privileged user. - Attempt to modify the parameters for configuration of the handling of audit data. - Verify using the TOE logs that the parameters involved with handling of audit data were modified.
Pass/Fail with Explanation	Pass. - TOE allows a security administrator to modify security related parameters for audit data handling configurations. - TOE log verifies that it allows a security administrator to modify security related parameters for audit data handling configurations. This meets the testing requirements.

7.1.4.5 FMT_MOF.1/ FUNCTIONS TEST #1C (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	If 'audit functionality when Local Audit Storage Space is full' is selected from the second selection together with 'modify the behaviour of' in the first selection The evaluator shall perform the following tests: Test 1: The evaluator shall try to modify the behaviour when Local Audit Storage Space is full without prior authentication as Security Administrator (by authentication as a user with no

	administrator privileges or without user authentication at all). This attempt should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.
Pass/Fail with Explanation	NA. 'Audit functionality when Local Audit Storage Space is full' is not selected in the ST, hence this test is not applicable.

7.1.4.6 FMT_MOF.1/ FUNCTIONS TEST #2C (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	If 'audit functionality when Local Audit Storage Space is full' is selected from the second selection together with 'modify the behaviour of' in the first selection The evaluator shall perform the following tests: Test 2: The evaluator shall try to modify the behaviour when Local Audit Storage Space is full with prior authentication as Security Administrator. This attempt should be successful. The effect of the change shall be verified. The evaluator does not necessarily have to test all possible values for the behaviour when Local Audit Storage Space is full but at least one change between allowed values for the behaviour.
Pass/Fail with Explanation	NA. 'Audit functionality when Local Audit Storage Space is full' is not selected in the ST, hence this test is not applicable.

7.1.4.7 FMT_MOF.1/ FUNCTIONS TEST #1D (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	If in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection The evaluator shall perform the following tests: Test 1: The evaluator shall try to determine the behaviour of all options chosen from the second selection without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions without administrator authentication shall fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

Pass/Fail with Explanation	NA. 'Determine the behaviour of' is not selected in the ST, hence this test is not applicable.
-----------------------------------	--

7.1.4.8 FMT_MOF.1/ FUNCTIONS TEST #2D (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	If in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection The evaluator shall perform the following tests: Test 2: The evaluator shall try to determine the behaviour of all options chosen from the second selection with prior authentication as Security Administrator. This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions with Security Administrator authentication shall be successful.
Pass/Fail with Explanation	NA. 'Determine the behaviour of' is not selected in the ST, hence this test is not applicable.

7.1.4.9 FMT_MOF.1/MANUALUPDATE TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall try to perform the update using a legitimate update image without prior authentication as Security Administrator (either by authentication as a user with no administrator privileges or without user authentication at all – depending on the configuration of the TOE). The attempt to update the TOE shall fail.
Test Steps	• Create a read only user to attempt to perform an update using a legitimate update image. • Verify that the user "tester" fails to update the TOE, as it is not authorized to do so.
Pass/Fail with Explanation	Pass. The TOE does not allow a non-security administrator to update it with a legitimate image. This meets the testing requirements.

7.1.4.10 FMT_MOF.1/MANUALUPDATE TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall try to perform the update with prior authentication as Security Administrator using a legitimate update image. This attempt should be successful. This test case should be covered by the tests for FPT_TUD_EXT.1 already
Pass/Fail with Explanation	Pass. This testing is covered by the requirements in FPT_TUD_EXT.1 Test #1.

7.1.4.11 FMT_MOF.1/SERVICES TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) without prior authentication as Security Administrator (either by authenticating as a user with no administrator privileges, if possible, or without prior authentication at all). The attempt to enable/disable this service/these services should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to enable/disable this service/these services can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.
Test Steps	• Start an SSH session onto the TOE with a lower privilege user. • Make sure the user is a non-privileged user. • Attempt to enable and disable the services and verify that the user is unable to do so.
Pass/Fail with Explanation	Pass. TOE does not allow a non-security administrator to enable or disable any services. This meets the testing requirement.

7.1.4.12 FMT_MOF.1/SERVICES TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) with prior authentication as Security Administrator. The attempt to enable/disable this service/these services should be successful.
Test Steps	• Start an SSH session onto the TOE with the admin user. • Attempt to enable and disable the services and observe it succeeds. • Verify using the TOE logs that security administrator was able to enable or disable services.
Pass/Fail with Explanation	Pass. • TOE allows the security administrator to enable or disable services. • TOE logs verify it allows the security administrator to enable or disable services. This meets the testing requirement.

7.1.4.13 FMT_MTD.1/COREDATA TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	No separate testing for FMT_MTD.1/CoreData is required unless one of the management functions has not already been exercised under any other SFR.

Pass/Fail with Explanation	Pass. All management functions have been exercised under other SFRs.
-----------------------------------	--

7.1.4.14 FMT_MTD.1/CRYPTOKEYS TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall try to perform at least one of the related actions (modify, delete, generate/import) without prior authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all). Attempts to perform related actions without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to manage cryptographic keys can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.
Test Steps	- Start an SSH session onto the TOE with a non-administrator user. - Attempt to perform an operation on the cryptographic keys i.e. import SSH key on the TOE. This will fail.
Pass/Fail with Explanation	Pass. TOE does not allow non-administrative users to perform an operation on cryptographic keys. This meets the testing requirement.

7.1.4.15 FMT_MTD.1/CRYPTOKEYS TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall try to perform at least one of the related actions with prior authentication as Security Administrator. This attempt should be successful.
Test Steps	- Start an SSH session onto the TOE with the security administrator user. - Attempt to perform an operation on cryptographic keys i.e. import SSH key on the TOE and verify it is successful. - Verify using the TOE logs that cryptographic keys were imported.
Pass/Fail with Explanation	Pass. - TOE allows security administrator to perform an operation on cryptographic keys. - TOE logs verify that it allows the security administrator to modify cryptographic keys. This meets the testing requirement.

7.1.4.16 FMT_SMF.1 TEST #1 (CPP_ND_V3.0E) [TD0880]

Item	Data
Test Assurance Activity	The evaluator shall test management functions as part of testing the SFRs identified in Section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR. TD0880 has been applied
Test Steps	• <i>Ability to administer the TOE remotely;</i> • <i>Ability to configure the access banner;</i> • <i>Ability to configure the remote session inactivity time before session termination;</i> • <i>Ability to update the TOE, and to verify the updates using <u>digital signature</u> capability prior to installing those updates;</i> • [○ <u>Ability to start and stop services;</u> ○ <u>Ability to configure local audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full, changes to local audit storage size);</u> ○ <u>Ability to modify the behaviour of the transmission of audit data to an external IT entity;</u> ○ <u>Ability to manage the cryptographic keys;</u> ○ <u>Ability to manage the cryptographic functionality;</u> ○ <u>Ability to configure thresholds for SSH rekeying;</u> ○ <u>Ability to re-enable an Administrator account;</u> ○ <u>Ability to configure the local session inactivity time before session termination or locking;</u> ○ <u>Ability to configure the authentication failure parameters for FIA_AFL.1;</u> ○ <u>Ability to set the time which is used for time-stamps;</u> ○ <u>Ability to configure NTP;</u> ○ <u>Ability to administer the TOE locally;</u> ○ <u>Ability to manage the trusted public keys database;</u>
Pass/Fail with Explanation	Pass. All management functions identified have been tested throughout the evaluation. Thus, this requirement has been met.

7.1.4.17 FMT_SMF.1/MACSEC TEST #20 (MOD_MACSEC_V1.0) [TD0803]

Item	Data
Test Assurance Activity	The evaluator shall set up an environment where the TOE can connect to two other MACsec devices, identified as devices B and C, with the ability of PSKs to be distributed between them. The evaluator shall configure the devices so that the TOE will be elected key server and principal actor, i.e., has highest key server priority. The evaluator shall follow the relevant operational guidance to perform the tests listed below. Note that if the TOE claims multiple management interfaces, the tests should be performed for each interface that supports the functions. Test 20: The evaluator shall connect to the PAE of the TOE and install a PSK. The evaluator shall then specify a CKN and that the PSK is to be used as a CAK. • Repeat this test for both 128-bit and 256-bit key sizes. • Repeat this test for a CKN of valid length (1-32 octets), and observe success. • (conditional, the length of the CKN is not configurable) Repeat this test again for CKN of invalid lengths zero and 33, and observe failure.

	(conditional, the length of the CKN is configurable) The evaluator shall observe that the admin guidance instructs the admin to configure the TOE to only allow CKN lengths of 1-32 bytes. TD0803 has been applied.
Test Steps	For AES-128, the CKN should be (2-64) hexadecimal characters i.e. 1-32 octets in length. <u>Minimum length:</u> - Configure the TOE for MACsec with a CKN that is 1-Octet (2 hexadecimal characters) in length. - Verify via TOE logs the CKN was created. - Verify the successful MACsec connection via packet capture and TOE logs. <u>Maximum Length:</u> - Configure the TOE for MACsec with a CKN that is 32-Octet (64 hexadecimal characters) in length. - Verify via TOE logs the CKN was created. - Verify the successful MACsec connection via packet capture and TOE logs. <u>Length in Allowable Range:</u> - Configure the TOE for MACsec with a CKN that is 16-Octet (32 hexadecimal characters) in length. - Verify via TOE logs the CKN was created. - Verify the successful MACsec connection via packet capture and TOE logs. <u>Length Beyond Supported Range (Lower):</u> - Attempt to configure the TOE for MACsec with a CKN that is less than 1-Octet (1 hexadecimal character) in length and verify that the TOE rejects it. <u>Length Beyond Supported Range (Higher):</u> - Attempt to configure the TOE for MACsec with a CKN that is 33-Octet (66 hexadecimal character) in length and verify that the TOE rejects it. For AES-256, the CKN should be (2-64) hexadecimal characters i.e. 1-32 octets in length. <u>Minimum length:</u> - Configure the TOE for MACsec with a CKN that is 1-Octet (2 hexadecimal characters) in length. - Verify via TOE logs the CKN was created. - Verify the successful MACsec connection via packet capture and TOE logs. <u>Maximum Length:</u> - Configure the TOE for MACsec with a CKN that is 32-Octet (64 hexadecimal characters) in length. - Verify via TOE logs the CKN was created. - Verify the successful MACsec connection via packet capture and TOE logs. <u>Length in Allowable Range:</u> - Configure the TOE for MACsec with a CKN that is 16-Octet (32 hexadecimal characters) in length. - Verify via TOE logs the CKN was created. - Verify the successful MACsec connection via packet capture and TOE logs. <u>Length Beyond Supported Range (Lower):</u>

	- Attempt to configure the TOE for MACsec with a CKN that is less than 1-Octet (1 hexadecimal character) in length and verify that the TOE rejects it. <u>Length Beyond Supported Range (Higher):</u> - Attempt to configure the TOE for MACsec with a CKN that is 33-Octet (66 hexadecimal character) in length and verify that the TOE rejects it.
Pass/Fail with Explanation	Pass. The TOE accepts and enforces valid CKN and CAK configurations and rejects invalid CKN lengths. This meets the testing requirement.

7.1.4.18 FMT_SMF.1/MACSEC TEST #21 (MOD_MACSEC_V1.0) [TD0889]

Item	Data
Test Assurance Activity	The evaluator shall set up an environment where the TOE can connect to two other MACsec devices, identified as devices B and C, with the ability of PSKs to be distributed between them. The evaluator shall configure the devices so that the TOE will be elected key server and principal actor, i.e., has highest key server priority. The evaluator shall follow the relevant operational guidance to perform the tests listed below. Note that if the TOE claims multiple management interfaces, the tests should be performed for each interface that supports the functions. Test 21: (conditional: "Cause key server to generate a new group CAK..." is selected) The evaluator shall test the ability of the TOE to enable and disable MKA participants using the management function specified in the ST. The evaluator shall install PSKs in devices B and C, and take any necessary additional steps to create corresponding MKA participants. The evaluator shall disable the MKA participant on device C, then observe that the TOE can communicate with B but cannot communicate with device C. The evaluator shall re-enable the MKA participant of device C and observe that the TOE is now able to communicate with devices B and C. TD0889 has been applied.
Pass/Fail with Explanation	NA. "Cause key server to generate a new group CAK..." selection is not made in ST.

7.1.4.19 FMT_SMF.1/MACSEC TEST #22 (MOD_MACSEC_V1.0) [TD0840]

Item	Data
Test Assurance Activity	The evaluator shall set up an environment where the TOE can connect to two other MACsec devices, identified as devices B and C, with the ability of PSKs to be distributed between them. The evaluator shall configure the devices so that the TOE will be elected key server and principal actor, i.e., has highest key server priority.

	The evaluator shall follow the relevant operational guidance to perform the tests listed below. Note that if the TOE claims multiple management interfaces, the tests should be performed for each interface that supports the functions. Test 22: For TOEs using only PSKs, the TOE should be the key server in both tests and only one peer (B) needs to be tested. The tests are: - Test 22.1 (Conditional: The TOE supports MKA keychains with multiple CKN/CAKs): Switch to unexpired CKN: TOE and Peer B have CKN1(10 minutes) and CKN2. CKN2 can either be configured with a longer overlapping lifetime (20 minutes) or be configured with a lifetime starting period of more than 10 minutes after the CKN1 start. The TOE and Peer B start using CKN1 and after 10 minutes, verify that the TOE expires SAK1. This can be verified by either 1) seeing the TOE immediately distribute a new SAK to the peer if the lifetime of CKN2 overlaps CKN1, or 2) by terminating the connection with CKN1 and distributing a new SAK once the lifetime period of CKN2 begins. - Test 22.2: Reject CA with expired CKN: TOE has CKN1 (10 minutes). Peer B has CKN1 (20 minutes). TOE and Peer B start using CKN1 and after 10 minutes, verify that the TOE rejects (or ignores) peer's request to use (or distribute) a SAK using CKN1. TD0840 has been applied.
Test Steps	Switch to unexpired CKN (Peer has same configurations). - Configure a start time for CKN1 on the TOE. - Configure the start time for CKN2 to 10 minutes after the CKN1 start time. - Enable MACsec on the TOE. - Verify via TOE logs the connection was established. - Verify that after 10 minutes a rekey occurs, the TOE expires CKN1 and distributes a new SAK via TOE logs. - Verify that rekey occurs via packet capture. Reject CA with expired CKN - Configure the start time for CKN1 on the TOE and the start time for CKN2 to 10 minutes after CKN1. - Configure the start time for CKN1 on the Peer and the start time for CKN2 to 20 minutes after CKN1. - Enable MACsec on the TOE. - Verify via TOE logs the connection was established. - Verify via TOE logs that CKN1 expires after 10 minutes. - Verify via TOE logs that the connection between the TOE and Peer is no longer active. - Verify via packet capture the TOE switches to CKN2 and does not respond to the Peer's request to use CKN1.
Pass/Fail with Explanation	Pass. TOE switches to unexpired CKN when the shorter lifetime CKN expires. TOE also rejects expired CKN used by the MACsec peer when the CKN is configured with a shorter lifetime on the TOE.

7.1.4.20 FMT_SMF.1/MACSEC TEST #23 (MOD_MACSEC_V1.0)

Item	Data
------	------

Test Assurance Activity	Test 23: (conditional, "Cause key server to generate a new group CAK..." is selected) The evaluator shall connect to the PAE of the TOE, set the management function specified in the ST (e.g., set ieee8021XKeyCreateNewGroup to true), and observe that the TOE distributes a new group CAK.
Pass/Fail with Explanation	NA. "Cause key server to generate a new group CAK..." selection is not made in ST.

7.1.4.21 FMT_SMR.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH, if the TSF shall be validated against the Functional Package for Secure Shell referenced in Section 2.2 of the cPP; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.
Pass/Fail with Explanation	Pass. There are two interfaces where these can be tested (over the local console and remote SSH). They are covered across FIA_UIA_EXT.1 Test #2, FIA_UIA_EXT.1 Test #3, FTA_SSL_EXT.1 Test #1, FTA_SSL.3 Test #1, FTA_SSL.4 Test #1 and FTA_SSL.4 Test #2. Moreover, the TSF has been validated against the Functional Package for Secure Shell. This meets the testing requirements.

7.1.5 PROTECTION OF THE TSF (FPT)

7.1.5.1 FPT_APW_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	No test assurance activities have been identified.
Pass/Fail with Explanation	No test assurance activities have been identified.

7.1.5.2 FPT_CAK_EXT.1 TEST #1 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	There are no test EAs for this component.
Pass/Fail with Explanation	There are no test EAs for this component.

7.1.5.3 FPT_FLS.1 TEST #24 (MOD_MACSEC_V1.0) [TD0816]

Item	Data
Test Assurance Activity	The following test may require the vendor to provide access to a test platform that provides the evaluator with the ability to modify the TOE internals in a manner that is not provided to end customers: Test 24: For each failure mode specified in the ST that can be deliberately induced, the evaluator shall ensure that the TOE attains a secure state (e.g., shutdown) after initiating each failure mode type. For TOEs with redundant failover capability, the evaluator shall determine that the failed components attain a secure state and the behavior of the TOE is consistent with the operational guidance. For each component, the evaluator shall repeat each type of self-test that can be deliberately induced to fail. TD0816 has been applied.
Test Steps	• Deliberately induce AES Known Answer Test failure and verify evidence using TOE output. • Deliberately induce GCM Known Answer Test failure and verify evidence using TOE output. • Deliberately induce RSA Signature Known Answer Test failure and verify evidence using TOE output. • Deliberately induce ECDSA Signature Known Answer Test failure and verify evidence using TOE output. • Deliberately induce ECDH Known Answer Test failure and verify evidence using TOE output. • Deliberately induce DRBG Known Answer Test failure and verify evidence using TOE output. • Deliberately induce SHA Known Answer Test failure and verify evidence using TOE output. • Deliberately induce HMAC-SHA Known Answer Test failure and verify evidence using TOE output. • Deliberately induce KAS-ECC Known Answer Test failure and verify evidence using TOE output. • Deliberately induce Software integrity Test failure and verify evidence using TOE output.
Pass/Fail with Explanation	Pass. The TOE attains a secure state for each failure mode that can be deliberately induced.

7.1.5.4 FPT_RPL.1 TEST #25 (MOD_MACSEC_V1.0) [TD0746, TD0881]

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Before performing each test, the evaluator shall successfully establish a MACsec channel between the TOE and a MACsec-capable peer in the operational environment sending enough traffic to see it working and verify the MN and PN values increase for each direction. Test 25: The evaluator shall set up a MACsec connection with an entity in the operational environment. The evaluator shall then capture traffic sent from this remote entity to the TOE. The evaluator shall retransmit copies of this traffic to the TOE in order to impersonate the remote entity where the PN values in the SecTag of these packets are less than the lowest acceptable PN

	for the SA. The evaluator shall observe that the TSF does not take action in response to receiving these packets and that the audit log indicates that the replayed traffic was discarded. TD0746 and TD0881 has been applied.
Test Steps	• Enable MACsec on the TOE. • Verify the MACsec connection is established via TOE logs. • Capture and record MACsec from the peer to the TOE and retransmit the same traffic to the TOE using Acumen-MACsec tool. • Verify that the replay traffic being dropped is logged on the TOE. • Verify via packet capture that the TOE only responds to legitimate ping traffic.
Pass/Fail with Explanation	Pass. The TOE does not take action in response to receiving retransmitted packets and the audit log indicated that the replayed traffic was discarded.

7.1.5.5 FPT_RPL.1 TEST #26 (MOD_MACSEC_V1.0) [TD0881]

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Before performing each test, the evaluator shall successfully establish a MACsec channel between the TOE and a MACsec-capable peer in the operational environment sending enough traffic to see it working and verify the MN and PN values increase for each direction. Test 26: The evaluator shall capture frames during an MKA session and record the lowest MN from the Basic Parameter set observed in a particular time range. The evaluator shall then send a frame with a lower MN, and then verify that this frame is dropped. The evaluator shall verify that the device logged this event. TD0881 has been applied.
Test Steps	• Enable MACsec on the TOE. • Verify the MACsec connection is established via TOE logs. • Using Acumen MACsec tool capture and record MACsec from the peer to the TOE and send a frame with an MN lower than the lowest observed MN. • Verify that the frame is dropped and the event logged via TOE logs. • Verify via packet capture that the TOE does not respond.
Pass/Fail with Explanation	Pass. The TOE rejected frame with an MN value lower than the lowest observed MN and logged the event.

7.1.5.6 FPT_SKP_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	No test assurance activities have been identified.
Pass/Fail with Explanation	No test assurance activities have been identified.

7.1.5.7 FPT_STM_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: If the TOE supports direct setting of the time by the Security Administrator then the evaluator shall use the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly. If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.
Test Steps	<u>Note</u>: The TOE is standalone and does not consist of multiple components. • Confirm the current time on the TOE. • Set a new time on the TOE with the help of the guidance document. • Verify that the new time is set via the TOE's logs.
Pass/Fail with Explanation	Pass. The TOE allows the administrative user to manually set the time on the TOE. This meets the testing requirement.

7.1.5.8 FPT_STM_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE and set up a communication path with the NTP server. The evaluator shall observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation. If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.
Pass/Fail with Explanation	Pass. Test has been exercised in FCS_NTP_EXT.1.1 Test #1.

7.1.5.9 FPT_STM_EXT.1 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Test 3 [conditional]: If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance. If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.
Pass/Fail with Explanation	NA. The TOE does not get its time from an underlying virtual system since it is not a vND. Hence, this test is not applicable.

7.1.5.10 FPT_TST_EXT.1 TEST #1 (CPP_ND_V3.0E) [TD0836]

Item	Data
Test Assurance Activity	It is expected that at least the following tests are performed: - Verification of the integrity of the firmware and executable software of the TOE - Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs. Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to: [FIPS 140-2], Section 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE. [FIPS 140-2], Section 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate. The evaluator shall verify that the self-tests described above are carried out according to the SFR and in agreement with the descriptions in the TSS. For distributed TOEs the evaluator shall perform testing of self-tests on all TOE components according to the description in the TSS about which self-test are performed by which component. TD0836 has been applied.
Test Steps	- Reboot the TOE. - Observe boot processes for integrity testing and cryptographic self-tests. - Verify via logs that the boot processes for integrity testing and cryptographic self-tests were successful.
Pass/Fail with Explanation	Pass. The TOE performs all integrity tests, cryptographic self-tests and noise source health tests. This meets the testing requirements.

7.1.5.11 FPT_TUD_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall perform the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps ('activation' could be performed e.g. by a distinct activation step or by rebooting the device). In that case, the evaluator shall verify after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator shall perform the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again. The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates). For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.
Test Steps	• Verify the current version of the TOE. • Attempt to install a legitimate update. • Reboot the TOE and verify that the new version of the TOE is successfully installed and that the logs reflect the same. • Verify the latest version of the TOE.
Pass/Fail with Explanation	Pass. The TOE was successfully upgraded to the new version when a legitimate update was used. This meets the testing requirements.

7.1.5.12 FPT_TUD_EXT.1 TEST #2 (A) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator shall first confirm that no updates are pending and then perform the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator shall obtain or produce illegitimate updates as defined below and attempt to install them on the TOE.

	The evaluator shall verify that the TOE rejects all of the illegitimate updates. The evaluator shall perform this test using all of the following forms of illegitimate updates: i. A modified version (e.g. using a hex editor) of a legitimately signed update The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify that both the current version and most recently installed version, reflect the same version information as prior to the update attempt. The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates). For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.
Test Steps	• Verify the current version of the TOE. • Modify the legitimately signed update file. • Attempt to install a modified version of a legitimate update. • Verify through logs that the TOE rejects the update. • Verify that the version of the TOE remains unchanged after the update attempt.
Pass/Fail with Explanation	Pass. The TOE software was able to detect when an image was corrupted and rejected the image. This meets the testing requirements.

7.1.5.13 FPT_TUD_EXT.1 TEST #2 (B) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator shall first confirm that no updates are pending and then perform the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator shall obtain or produce illegitimate updates as defined below and attempt to install them on the TOE. The evaluator shall verify that the TOE rejects all of the illegitimate updates. The evaluator shall perform this test using all of the following forms of illegitimate updates: ii. An image that has not been signed The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the

	evaluator shall verify that both the current version and most recently installed version, reflect the same version information as prior to the update attempt. The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates). For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.
Test Steps	• Verify the current version of the TOE. • Remove the signature from the image. • Attempt to install the update without a signature. • Verify through logs that the TOE rejects the update. • Verify that the version of the TOE remains unchanged after the update attempt.
Pass/Fail with Explanation	Pass. The TOE software was able to detect when an image was not signed and rejected the image. This meets the testing requirements.

7.1.5.14 FPT_TUD_EXT.1 TEST #2 (C) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator shall first confirm that no updates are pending and then perform the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator shall obtain or produce illegitimate updates as defined below and attempt to install them on the TOE. The evaluator shall verify that the TOE rejects all of the illegitimate updates. The evaluator shall perform this test using all of the following forms of illegitimate updates: iii. An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature) The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify that both the current version and most recently installed version, reflect the same version information as prior to the update attempt. The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates). For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.
Test Steps	• Verify the current version of the TOE. • Modify the signature of the update.

	• Attempt to install an update with a corrupt/modified signature. • Verify through logs that the TOE rejects the update. • Verify that the version of the TOE remains unchanged after the update attempt.
Pass/Fail with Explanation	Pass. The TOE software was able to detect when an image had an invalid signature and rejected the image. This meets the testing requirements.

7.1.6 TOE ACCESS (FTA)

7.1.6.1 FTA_SSL_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following test: Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a local interactive session with the TOE. The evaluator shall then observe that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator shall then ensure that reauthentication is needed when trying to unlock the session.
Test Steps	• Configure an inactivity timeout period of 1 minute. • Verify the logs on the TOE for the updated session timeout. • Connect to the TOE from the local console connection and allow the session to timeout. • Verify using TOE logs that the session was terminated due to inactivity timeout. • Configure an inactivity timeout period of 2 minutes. • Verify the logs on the TOE for the updated session timeout. • Connect to the TOE from the local console connection and allow the session to timeout. • Verify using TOE logs that the session was terminated due to inactivity timeout.
Pass/Fail with Explanation	Pass. • TOE allows the configuration of the session inactivity time-period. • TOE terminates the local console session when no activity is observed for the configured inactivity time-period. • TOE log verifies the user's local console session is terminated. This meets the testing requirement.

7.1.6.2 FTA_SSL.3 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For each method of remote administration, the evaluator shall perform the following test: Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a remote interactive session with the TOE. The evaluator shall then observe that the session is terminated after the configured time period.

Test Steps	• Configure the TOE with an inactivity timeout period of 1 minute. • Verify that a log was created for configuring the timeout period. • Log into the TOE via remote SSH connection and allow the session to time out. • Verify the logs for session timeout and subsequent termination. • Configure the TOE with an inactivity timeout period of two minutes. • Verify that a log was created for configuring the timeout period. • Log into the TOE via remote SSH connection and allow the session to time out. • Verify the logs for session timeout and subsequent termination.
Pass/Fail with Explanation	Pass. • TOE terminates a remote SSH session, when no activity is observed for the configured inactivity time period. • TOE logs verify the user SSH session is terminated due to inactivity timeout. This meets the testing requirement.

7.1.6.3 FTA_SSL.4 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1 [conditional]: If the TOE supports local administration, the evaluator shall initiate an interactive local session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.
Test Steps	• Log onto the TOE through a local console administrative interface. • Log out of the TOE by following the guidance documentation to exit the session. • Verify the logs reflect that a session has been created and terminated.
Pass/Fail with Explanation	Pass. • The TOE allows the user to terminate the local console administrative sessions. • TOE log verifies that the interactive local console session can be exited by following the guidance documentation. This meets the testing requirement.

7.1.6.4 FTA_SSL.4 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: For each method of remote administration, the evaluator shall initiate an interactive remote session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.
Test Steps	• Log onto the TOE remotely via SSH. • Log out of the TOE by following the guidance documentation to exit the session. • Verify the logs reflect that a session has been created and terminated.
Pass/Fail with Explanation	Pass.

	• TOE administrator can be exited from an interactive remote SSH session following the guidance documentation. • TOE logs verify that an interactive remote SSH session can be exited by following the guidance documentation. This meets the testing requirements.
--	---

7.1.6.5 FTA_TAB.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall also perform the following test: Test 1: The evaluator shall follow the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.
Test Steps	• Configure an access banner for the TOE. • Verify that the audit records reflect the configuration step. • Log into the TOE via SSH and verify that the access banner is displayed. • Log into the TOE via console and verify that the access banner is displayed.
Pass/Fail with Explanation	Pass. The TOE allows configuration of an access banner and displays the same before the authentication step when a user accesses the TOE through the local console or remote SSH. This meets the testing requirement.

7.1.7 TRUSTED PATH/CHANNELS (FTP)

7.1.7.1 FTP_ITC.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The evaluator shall perform the following tests: Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful. Further assurance activities are associated with the specific protocols.

	For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target. The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.
Pass/Fail with Explanation	Pass. This testing is covered by the requirements in FAU_STG_EXT.1 Test #1 for the SSH channel with the syslog server.

7.1.7.2 FTP_ITC.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The evaluator shall perform the following tests: Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target. The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.
Pass/Fail with Explanation	NA. This test is not applicable as TOE acts as a server and cannot initiate the SSH communication channel between itself and the authorized IT entity.

7.1.7.3 FTP_ITC.1 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be

	sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The evaluator shall perform the following tests: Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target. The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.
Pass/Fail with Explanation	Pass. This testing is covered by the requirements in FAU_STG_EXT.1 Test #1 for the SSH channel with the syslog server.

7.1.7.4 FTP_ITC.1 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report. The evaluator shall perform the following tests: Test 4: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route to the external IT entities. The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, interrupt the connection of that IT entity for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration shorter than the application layer timeout but of sufficient length to interrupt the network link layer. The evaluator shall ensure that, when the connectivity is restored, communications are appropriately protected and no TSF data is sent in plaintext.

	In the case where the TOE is able to detect TOE external interruption (such as a cable being physically removed or a virtual connection being disabled), another network device shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall be external to the TOE (i.e., by manipulating the test environment and not by TOE configuration change). Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target. The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.
Pass/Fail with Explanation	NA. This test is not applicable as TOE acts as a server and not as a client in the secure SSH communication channel between itself and the authorized IT entity.

7.1.7.5 FTP_ITC.1/MACSEC TEST #1 (MOD_MACSEC_V1.0)

Item	Data
Test Assurance Activity	This SFR is addressed through evaluation of FCS_MACSEC_EXT.1 through FCS_MACSEC_EXT.4.
Pass/Fail with Explanation	Pass. This SFR is addressed through evaluation of FCS_MACSEC_EXT.1 through FCS_MACSEC_EXT.4. This meets the testing requirements.

7.1.7.6 FTP_TRP.1/ADMIN TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.
Test Steps	• Attempt to establish a remote administration session with the TOE via SSH. • Verify that Wireshark shows a successful communication and that it's not in plain text. • Verify via logs that the TOE shows a successful connection.

Pass/Fail with Explanation	Pass. Remote administrators can access the TOE via an SSH trusted path. This meets the testing requirement.
-----------------------------------	---

7.1.7.7 FTP_TRP.1/ADMIN TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.
Pass/Fail with Explanation	Pass. The communication channel data is not sent in plaintext. This is covered by FTP_TRP.1/Admin_Test#1 and FCS_SSH_EXT.1 testing wherein no data was sent in plaintext.

8 CAVP CERTIFICATE TABLE

This section provides a table that lists all SFRs for which a CAVP certificate is claimed, the CAVP algorithm list name and the CAVP Certificate number in accordance with the capability mapping of the TOE.

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of [2048, 3072 and 4096 bits] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;	Junos 24.4R1 OpenSSL	RSA KeyGen (FIPS186-5) Modulo: 2048, 3072, 4096	A6978
	ECC schemes using 'NIST curves' [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.;	Junos 24.4R1 OpenSSL	ECDSA KeyGen (FIPS186-5) Curves: P-256, P-384, P-521 ECDSA KeyVer (FIPS186-5) Curves: P-256, P-384, P-521	A6978
FCS_CKM.2	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Junos 24.4R1 OpenSSL	KAS-ECC-SSC Sp800-56Ar3 Domain Parameter Generation Methods: P-256, P-384, P-512	A6978
FCS_COP.1/ DataEncryption	AES used in [CBC, CTR, GCM] mode and cryptographic key sizes [128 bits, 256 bits] [CBC as specified in ISO	Junos 24.4R1 OpenSSL	AES-CBC Key Length: 128, 256	A6978

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
	10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772].		AES-CTR Key Length: 128, 256 AES-GCM Key Length: 128, 256	
		Juniper Penta Silicon	AES-GCM Key Length: 128, 256	A4708
		AES ECB 128bit & 256bit Encryption/Decryption Engine (BCM82756)		AES 4550
FCS_COP.1/ SigGen	For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,	Junos 24.4R1 OpenSSL	RSA SigGen (FIPS186-5) Modulo: 2048, 3072, 4096 RSA SigVer (FIPS186-5) Modulo: 2048, 3072, 4096	A6978
	For ECDSA schemes implementing [selection: P-256, P-384, P-521] curves that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" curves"; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3,	Junos 24.4R1 OpenSSL	ECDSA SigGen (FIPS186-5) Curves: P-256, P-384, P-521 ECDSA SigVer (FIPS186-5) Curves: P-256, P-384, P-521	A6978

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
	"IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.			
FCS_COP.1/Hash	[SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 256, 384, 512] bits that meet the following: ISO/IEC 10118-3:2004.	Junos 24.4R1 OpenSSL	SHA-1, SHA2-256, SHA2-384, SHA2-512	A6978
		Junos 24.4R1 LibMD	SHA2-256, SHA2-512	A6975
FCS_COP.1/KeyedHash	[HMAC-SHA- 256, HMAC-SHA-512] and cryptographic key sizes [256 and 512 bits] and message digest sizes [256, 512] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".	Junos 24.4R1 OpenSSL	HMAC-SHA2- 256, HMAC-SHA2-512	A6978
FCS_RBG_EXT.1	HMAC_DRBG [SHA-512] in accordance with ISO/IEC 18031:2011	MX and EX Junos 24.4R1 Kernel	HMAC DRBG Mode: SHA2-512	A7045
FCS_COP.1.1/CMAC	cryptographic algorithm [AES-CMAC] and cryptographic key sizes [128, 256] bits and message digest size of 128 bits that meets the following: [NIST SP 800-38B]	Junos 24.4R1 Macsec	AES-CMAC Key Length: 128, 256	A6976
FCS_COP.1.1/MACSEC	cryptographic algorithm [AES used in AES Key Wrap, GCM] and cryptographic key sizes [128, 256] bits that meets the following: [AES as specified in ISO 18033-3, AES Key Wrap as specified in NIST SP 800-38F, GCM as specified in ISO 19772].	Junos 24.4R1 Macsec	AES-KW Key Length: 128, 256	A6976
		AES ECB 128bit & 256bit Encryption/Decryption Engine (BCM82756)	AES-GCM Key Length: 128, 256	AES 4550
		Juniper Penta Silicon		A4708

9 CONCLUSION

The testing shows that all test cases required for conformance have passed testing.