

Assurance Activities Report

for

NetApp Storage Encryption (NSE) running ONTAP 9.7P13

Version 1.0

31 August 2021

Prepared by:



Leidos Inc.

<https://www.leidos.com/CC-FIPS140>

Common Criteria Testing Laboratory

6841 Benjamin Franklin Drive

Columbia, MD 21046

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:

NetApp, Inc.
1395 Crossman Avenue
Sunnyvale, CA 94089

The TOE Evaluation was Sponsored by:

NetApp, Inc.
1395 Crossman Avenue
Sunnyvale, CA 94089

Evaluation Personnel:

Anthony Apted
Kevin Steiner
Punit Patel

Common Criteria Version:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.

Common Evaluation Methodology Version:

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

Protection Profile:

- *collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition*, Version 2.0+Errata, 1 February 2019

Revision History

Version	Date	Description
0.1	9 October 2020	Initial draft
0.2	10 December 2020	Update based on modified ST
1.0 draft	15 June 2021	Final draft for check-out
1.0	31 August 2021	Final

Contents

1	Introduction	1
1.1	Technical Decisions	1
1.2	References	1
1.3	Glossary.....	2
2	Security Functional Requirement Evaluation Activities.....	4
2.1	Cryptographic Support (FCS).....	4
2.1.1	Authorization Factor Acquisition (FCS_AFA_EXT.1).....	4
2.1.2	Timing of Authorization Factor Acquisition (FCS_AFA_EXT.2).....	5
2.1.3	Cryptographic Key Generation (Symmetric Keys) (FCS_CKM.1(b))	6
2.1.4	Cryptographic Key Destruction (Power Management) (FCS_CKM.4(a)).....	7
2.1.5	Cryptographic Key Destruction (Software TOE, 3 rd Party Storage) (FCS_CKM.4(d)).....	7
2.1.6	Cryptographic Key and Key Material Destruction (Destruction Timing) (FCS_CKM_EXT.4(a)) 12	
2.1.7	Cryptographic Key and Key Material Destruction (Power Management) (FCS_CKM_EXT.4(b))	12
2.1.8	Cryptographic Operation (Signature Verification) (FCS_COP.1(a)).....	13
2.1.9	Cryptographic Operation (Hash Algorithm) (FCS_COP.1(b))	15
2.1.10	Cryptographic Operation (Keyed Hash Algorithm) (FCS_COP.1(c))	16
2.1.11	Cryptographic Operation (Key Wrapping) (FCS_COP.1(d)).....	16
2.1.12	Cryptographic Key Derivation (FCS_KDF_EXT.1).....	17
2.1.13	Key Chaining (Initiator) (FCS_KYC_EXT.1)	18
2.1.14	Cryptographic Password Construct and Conditioning (FCS_PCC_EXT.1).....	19
2.1.15	Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1).....	20
2.1.16	Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) (FCS_SNI_EXT.1).....	21
2.1.17	Validation (FCS_VAL_EXT.1).....	22
2.2	Security Management (FMT)	24
2.2.1	Management of Functions Behavior (FMT_MOF.1)	24
2.2.2	Specification of Management Functions (FMT_SMF.1).....	25
2.2.3	Security Roles (FMT_SMR.1).....	27
2.3	Protection of the TSF (FPT)	28
2.3.1	Protection of Key and Key Material (FPT_KYP_EXT.1).....	28

2.3.2	Power Saving States (FPT_PWR_EXT.1)	29
2.3.3	Timing of Power Saving States (FPT_PWR_EXT.2)	30
2.3.4	TSF Testing (FPT_TST_EXT.1)	30
2.3.5	Trusted Update (FPT_TUD_EXT.1)	32
3	Security Assurance Requirements	34
3.1	ASE: Security Targeted Evaluation	34
3.1.1	Conformance Claims (ASE_CCL.1)	34
3.2	Development (ADV)	35
3.2.1	Basic Functional Specification (ADV_FSP.1)	35
3.3	Guidance Documents (AGD)	37
3.3.1	Operational User Guidance (AGD_OPE.1)	37
3.3.2	Preparative Procedures (AGD_PRE.1)	38
3.4	Life-Cycle Support (ALC)	39
3.4.1	Labeling of the TOE (ALC_CMC.1)	39
3.4.2	TOE CM Coverage (ALC_CMS.1)	39
3.5	Tests (ATE)	39
3.6	Vulnerability Assessment (AVA)	41
3.6.1	Vulnerability Survey (AVA_VAN.1)	41

1 Introduction

This document presents results from performing assurance activities associated with the NetApp Storage Encryption (NSE) running ONTAP 9.7P13 evaluation. This report contains sections documenting the performance of evaluation activities associated with each of the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) as specified in the following Protection Profile:

- collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition, Version 2.0+Errata, 1 February 2019.

Note that, in accordance with NIAP Policy Letter #5, all cryptography in the TOE for which NIST provides validation testing of FIPS-approved and NIST-recommended cryptographic algorithms and their individual components must be NIST validated. The CCTL will verify that the claimed NIST validation complies with the NIAP-approved PP requirements the TOE claims to satisfy. The CCTL verification of the NIST validation will constitute performance of the associated assurance activity. As such, Test activities associated with functional requirements within the scope of Policy Letter #5 are performed by verification of the relevant CAVP certification and not through performance of any testing as specified in the supporting documents.

1.1 Technical Decisions

The following subsection lists the Technical Decisions that have been issued by NIAP against the claimed Protection Profile, along with rationale as to their applicability or otherwise to this evaluation.

- TD0458: FIT Technical Decision for FPT_KYP_EXT.1 evaluation activities

This TD has been applied to this evaluation.

1.2 References

[CPP_FDE_AA_V2.0E]	collaborative Protection Profile for Full Drive Encryption – Authorization Acquisition, Version 2.0+Errata, 1 February 2019
[CPP_FDE_AA_SD_V2.0E]	Supporting Document – Mandatory Technical Document – Full Drive Encryption: Authorization Acquisition, Version 2.0+Errata 20190201, 1 February 2019
[ST]	NetApp Storage Encryption (NSE) running ONTAP 9.7P13 Security Target, Version 1.0, 14 June 2021
[KMD]	NetApp ONTAP 9.7P13 with NetApp Storage Encryption and Onboard Key Manager Common Criteria Full Drive Encryption – Authorization Acquisition Key Management Description, Version 0.9, 14 June 2021
[NEPG]	NetApp® Encryption Power Guide, June 2021
[CMPR]	Commands: Manual Page Reference, November 2019
[SAR]	System Administration Reference, April 2020
[AARPG]	Administrator Authentication and RBAC Power Guide, April 2020
[UEG]	Upgrade Express Guide, January 2020
[URDG]	Upgrade and Revert/Downgrade Guide, April 2020.

1.3 Glossary

The following terms appear throughout [ST], [KMD], and the various guidance documents, and are used in the descriptions of evaluation activities performed by the evaluation team.

Term	Definition
Authentication Key (AK)	Used to authenticate ONTAP to a Self-Encrypting Drive (SED). In the context of the TOE, the AK is the Border Encryption Value (BEV).
AK1	One of two AKs generated by the TOE. AK1 can be used as the AK for the SED's data port, the SED's FIPS port, or both.
AK2	One of two AKs generated by the TOE. As with AK1, it can be used as the AK for the SED's data port, the SED's FIPS port, or both.
Border Encryption Value (BEV)	Defined in [CPP_FDE_AA_V2.0E] as a value passed from the Authorization Acquisition (AA) component to the Encryption Engine (EE) component to link the key chains of the two components.
Cluster Key Encryption Key (CKEK)	Used to protect each of the SVM-KEKs.
Cluster Passphrase (CP)	A 64-256 byte customer-generated ASCII string used as the authorization factor.
Cluster Pass-phrase Key Encryption Key (CP-KEK)	A 32-byte KEK that is derived from the Cluster Passphrase using PBKDFv2. The CP-KEK is used to protect the CKEK.
Data Encryption Key (DEK)	Defined in [CPP_FDE_AA_V2.0E] as a key used to encrypt data-at-rest.
Intermediate Key	Defined in [CPP_FDE_AA_V2.0E] as a key used in a point between the initial user authorization and the DEK.
Key Chaining	Defined in [CPP_FDE_AA_V2.0E] as the method of using multiple layers of encryption keys to protect data. A top layer key encrypts a lower layer key, which encrypts the data; this method can have any number of layers.
Key Encryption Key (KEK)	Defined in [CPP_FDE_AA_V2.0E] as a key used to encrypt other keys, such as DEKs.
Key Material	Defined in [CPP_FDE_AA_V2.0E]—key material is commonly known as critical security parameter (CSP) data, and includes authorization data, nonces, and metadata.
Key Sanitization	Defined in [CPP_FDE_AA_V2.0E] as a method of sanitizing encrypted data by securely overwriting or destroying the key that was encrypting the data.
Key Wrapping/ Unwrapping	Defined by NIST as a method of encrypting and decrypting keys and (possibly) associated data using a symmetric key; both confidentiality and integrity protection are provided.

Term	Definition
Non-Volatile Memory	Defined in [CPP_FDE_AA_V2.0E] as a type of computer memory that will retain information without power.
Onboard Key Manager (OKM)	TOE component providing local key-management without requiring an external key management server.
Protected Data	Defined in [CPP_FDE_AA_V2.0E]—refers to all data on the storage device with the exception of a small portion required for the TOE to function correctly. It is all space on the disk a user could write data to and includes the operating system, applications, and user data. Protected data does not include the Master Boot Record or Pre-authentication area of the drive – areas of the drive that are necessarily unencrypted.
Replicated Database (RDB)	TOE component used to persist configuration data.
Submask	Defined in [CPP_FDE_AA_V2.0E]—a bit string that can be generated and stored in a number of ways.
Storage Virtual Machine (SVM)	TOE component that serves data to clients and hosts. Like a virtual machine running on a hypervisor, an SVM is a logical entity that abstracts physical resources. Data accessed through the SVM is not bound to a location in storage. Network access to the SVM is not bound to a physical port.
Volume Data Encryption Key (VDEK)	A symmetric key used to encrypt/decrypt the volume data.
wAK	Wrapped (i.e. encrypted) version of AK. The wAK is formed by wrapping an AK with the CKEK.
wAK1	Wrapped version of AK1, formed by wrapping AK1 with the CKEK.
wAK2	Wrapped version of AK2, formed by wrapping AK2 with the CKEK.
wCKEK	The wrapped version of the CKEK.

2 Security Functional Requirement Evaluation Activities

This section describes the evaluation activities associated with the SFRs defined in the ST and the results of those activities as performed by the evaluation team. The evaluation activities are drawn from [CPP_FDE_AA_SD_V2.0E]. NIAP Technical Decisions have been applied and are identified as appropriate.

2.1 Cryptographic Support (FCS)

2.1.1 Authorization Factor Acquisition (FCS_AFA_EXT.1)

2.1.1.1 TSS Activities

The evaluator shall first examine the TSS to ensure that the authorization factors specified in the ST are described. For password-based factors the examination of the TSS section is performed as part of FCS_PCC_EXT.1 Evaluation Activities. Additionally in this case, the evaluator shall verify that the operational guidance discusses the characteristics of external authorization factors (e.g., how the authorization factor must be generated; format(s) or standards that the authorization factor must meet) that are able to be used by the TOE.

Section 6.2.1.1 of [ST] (“FCS_AFA_EXT.1 Authorization Factor Acquisition”) specifies in FCS_AFA_EXT.1.1 the TSF accepts a submask derived from a password authorization factor. Refer to the TSS Activities for FCS_PCC_EXT.1.

If other authorization factors are specified, then for each factor, the TSS specifies how the factors are input into the TOE.

The password-based factor is the only authorization factor accepted by the TOE.

2.1.1.2 Guidance Activities

The evaluator shall verify that the AGD guidance includes instructions for all of the authorization factors. The AGD will discuss the characteristics of external authorization factors (e.g., how the authorization factor is generated; format(s) or standards that the authorization factor must meet, configuration of the TPM device used) that are able to be used by the TOE.

Section “Configuring NetApp hardware-based encryption > Configuring onboard key management > Enabling onboard key management in ONTAP 9.6 and later” of [NEPG] provides instructions for configuring the Cluster Passphrase authorization factor, using the `security key-manager onboard enable` command. It specifies that in the evaluated configuration (`cc-mode-enabled=yes`), the passphrase is a string of between 64 and 256 ASCII characters.

2.1.1.3 KMD Activities

The evaluator shall examine the Key Management Description to confirm that the initial authorization factors (submasks) directly contribute to the unwrapping of the BEV.

Section 2.1 of [KMD] (“Border Encryption Values”) describes how the initial authorization factor directly contributes to unwrapping the BEV. The Cluster Passphrase (CP) entered by the user is concatenated with the Cluster Salt (CS) and this bit string is passed through the PBKDF2 function to produce the Cluster Passphrase Key Encryption Key (CP-KEK). The CP-KEK is then used to unwrap the Cluster Key Encryption Key (CKEK) and the CKEK is used to unwrap the Authentication Key (AK), which in TOE terms is the BEV.

The evaluator shall verify the KMD describes how a submask is produced from the authorization factor (including any associated standards to which this process might conform), and verification is performed to ensure the length of the submask meets the required size (as specified in this requirement).

Section 2.1.1 of [KMD] (“Cluster Passphrase”) describes the method by which the password is encoded and fed to the SHA algorithm. The password is a 64 to 256 byte ASCII string that is concatenated with a salt generated by the TOE’s DRBG and fed to the PBKDF2 function that meets NIST SP 800-132 to derive the Cluster Passphrase Key Encryption Key (CP-KEK). Section 2.4 of [KMD] (“Critical Security Parameters Summary”) describes the parameters of the PBKDF2 function, consistent with the selections in the ST—using the HMAC-SHA-512 algorithm with 1024 iterations and an output size of 256 bits.

2.1.1.4 Test Activities

The password authorization factor is tested in FCS_PCC_EXT.1.

The evaluator shall also perform the following tests:

Test 1 (conditional): If there is more than one authorization factor, ensure that failure to supply a required authorization factor does not result in access to the decrypted plaintext data.

The TOE supports a single authorization factor, the Cluster Passphrase, which is tested in FCS_PCC_EXT.1. Therefore, this test is not applicable to the TOE.

2.1.2 Timing of Authorization Factor Acquisition (FCS_AFA_EXT.2)

2.1.2.1 TSS Activities

The evaluator shall examine the TSS for a description of authorization factors and which of the factors are used to gain access to user data after the TOE entered a Compliant power saving state. The TSS is inspected to ensure it describes that each authorization factor satisfies the requirements of FCS_AFA_EXT.1.1.

Section 7.1.1 of [ST] (“FCS_AFA_EXT.1: Authorization Factor Acquisition”) states the Cluster Passphrase serves as the password authorization factor—it is the only authorization factor specified in the ST. It is required to be entered at system boot in order for the user to gain access to user data.

Section 7.1.2 of [ST] (“FCS_AFA_EXT.2: Timing of Authorization Factor Acquisition”) states the TOE provides the Compliant power saving states of G2(S5) (soft off) and G3 (mechanical off). After the TOE has entered either state, the Cluster Passphrase must be entered in order for the user to gain access to user data.

Section 7.1.14 of [ST] (“FCS_PCC_EXT.1: Cryptographic Password Construct and Conditioning”) describes how the Cluster Passphrase satisfies the requirements of FCS_AFA_EXT.1.1.

2.1.2.2 Guidance Activities

The evaluator shall examine the guidance documentation for a description of authorization factors used to access plaintext data when resuming from a Compliant power saving state.

Section “Configuring NetApp hardware-based encryption > Configuring onboard key management > Enabling onboard key management in ONTAP 9.6 and later” of [NEPG] states in the evaluated configuration (`cc-mode-enabled=yes`), the passphrase must be entered whenever the TOE is rebooted (i.e., resumes from the G2(S5) (soft off) or the G3 (mechanical off) Compliant power saving state to the powered state).

2.1.2.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.2.4 Test Activities

The evaluator shall also perform the following test:

- Enter the TOE into a Compliant power saving state
- Force the TOE to resume from a Compliant power saving state
- Release an invalid authorization factor and verify that access to decrypted plaintext data is denied
- Release a valid authorization factor and verify that access to decrypted plaintext data is granted.

The evaluator rebooted the TOE and entered the incorrect cluster passphrase then attempted to mount the volume and confirmed the mount failed and no plaintext data was accessible. The evaluator then rebooted the TOE and input the correct cluster passphrase and successfully mounted the volume and accessed the decrypted data.

2.1.3 Cryptographic Key Generation (Symmetric Keys) (FCS_CKM.1(b))

2.1.3.1 TSS Activities

The evaluator shall review the TSS to determine that a symmetric key is supported by the product, that the TSS includes a description of the protection provided by the product for this key. The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE.

Section 7.1.3 of [ST] (“FCS_CKM.1(b): Cryptographic Key Generation (Symmetric Keys)”) states the TOE generates the Cluster Key Encryption Key (CKEK), which is a 256-bit AES key used to wrap the Authentication Keys (AKs). Section 7.1.3 further states the TOE generates two 32-byte authentication keys (collectively known as AKs) when the Onboard Key Manager component is configured. They are designated AK1 and AK2. Customers are free to assign AK1 (or AK2) to either the data or FIPS port of the controller’s NSE drive. Best practices dictate that one AK be used for the drive’s data port and the other for the drive’s FIPS port.

Neither AK1 nor AK2 is persistently stored. The wrapped versions of the AKs (wAK1 and wAK2) are persistently stored in the OKM DB.

2.1.3.2 Guidance Activities

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

All AES keys the TOE generates are 256 bit keys. As such, there is no configuration required to configure the TOE to use selected key sizes, and no guidance instructions are necessary.

2.1.3.3 KMD Activities

If the TOE uses a symmetric key as part of the key chain, the KMD should detail how the symmetric key is used as part of the key chain.

Section 2 of [KMD] (“Keys and Key Hierarchy”) describes the Cluster Key Encryption Key (CKEK) and the Authentication Keys (AK1 and AK2) and how they are used as part of the key chain.

2.1.3.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.4 Cryptographic Key Destruction (Power Management) (FCS_CKM.4(a))

2.1.4.1 TSS Activity

The evaluator shall verify the TSS provides a high level description of how keys stored in volatile memory are destroyed. The evaluator to verify that TSS outlines:

- if and when the TSF or the Operational Environment is used to destroy keys from volatile memory;
- if and how memory locations for (temporary) keys are tracked;
- details of the interface used for key erasure when relying on the OE for memory clearing.

Section 7.1.4 of [ST] (“FCS_CKM.4(a)(AA): Cryptographic Key Destruction (Power Management)”) states the TOE provides the Compliant power saving states of G2(S5) (soft off) and G3 (mechanical off). All keys in volatile memory are destroyed when entering the G2(S5) or G3 Compliant power saving state. In both states, power is removed from memory and all values drain to a zero state.

2.1.4.2 Guidance Activities

The evaluator shall check the guidance documentation if the TOE depends on the Operational Environment for memory clearing and how that is achieved.

The TOE does not depend on the Operational Environment for memory clearing.

2.1.4.3 KMD Activities

The evaluator shall check to ensure the KMD lists each type of key, its origin, possible memory locations in volatile memory.

Section 2.4 of [KMD] (“Critical Security Parameters Summary”) provides information for each of the keys and critical security parameters required by the TOE, including for each key and parameter its type, origin, and possible memory locations in volatile memory.

2.1.4.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.5 Cryptographic Key Destruction (Software TOE, 3rd Party Storage) (FCS_CKM.4(d))

2.1.5.1 TSS + KMD Activities (KMD may be used if necessary details describe proprietary information)

The evaluator examines the TSS to ensure it describes how the keys are managed in volatile memory. This description includes details of how each identified key is introduced into volatile memory (e.g. by derivation from user input, or by unwrapping a wrapped key stored in non-volatile memory) and how they are overwritten.

Section 2.4 of [KMD] (“Critical Security Parameters Summary”), Table 4 (“Critical Security Parameters”) provides the following information regarding management in volatile memory for each of the keys and critical security parameters required by the TOE:

- Cluster Passphrase—is temporarily stored as a variable or a CPU register value by functions involved in the calculation of the CP-KEK. It is supplied by the cluster administrator when the `security key-manager onboard enable` and `security key-manager onboard update-passphrase` commands are executed.
- Cluster Salt—is temporarily stored as a variable or a CPU register value by functions involved in the calculation of the CP-KEK, or functions that need to store the Cluster Salt in non-volatile memory. It is generated by the TOE’s cryptomodule.
- CP-Salt—is temporarily stored as a variable or a CPU register value by functions involved in the calculation of the CP-KEK, or functions that need to store the CP-Salt in non-volatile memory. It is generated by the TOE’s cryptomodule.
- CP-Hash—is temporarily stored as a variable or a CPU register value by functions that need to validate the cluster passphrase, or functions that need to store the CP-Hash in non-volatile memory. It is calculated by the TOE’s cryptomodule as the SHA-256 hash of the CP-Salt concatenated with the Cluster Passphrase.
- Cluster Passphrase Key Encryption Key (CP-KEK)—is temporarily stored as a variable or a CPU register value by functions that need to wrap or unwrap the CKEK. It is additionally stored in the key table of the TOE’s cryptomodule. It is derived from the Cluster Passphrase and Cluster Salt using PBKDFv2.
- Cluster Key Encryption Key (CKEK)—is temporarily stored as a variable or a CPU register value by functions that need to wrap or unwrap the Authentication Keys. It is additionally stored in the key table of the TOE’s cryptomodule. It is generated by the TOE’s cryptomodule.
- Wrapped CKEK (wCKEK)—is temporarily stored as a variable or a CPU register value by functions that need to wrap the CKEK, unwrap the wCKEK, or store the wCKEK in non-volatile memory. It is created by cryptographically wrapping the CKEK using the KWP algorithm with the CP-KEK as the wrapping key.
- Authentication Key (AK)—is temporarily stored as a variable or a CPU register value by functions that need to wrap the AK, unwrap the wAK, or use the AK to authenticate to a self-encrypting drive. It is additionally stored in the key table of the TOE’s cryptomodule. It is generated by the TOE’s cryptomodule.
- Wrapped AK (wAK)—is temporarily stored as a variable or a CPU register value by functions that need to wrap the AK, unwrap the wAK, or store the wAK in non-volatile memory. It is created by cryptographically wrapping the AK using the KWP algorithm with the CKEK as the wrapping key.

Section 4.6 of [KMD] (“Key Lifetime and Destruction”) states keys stored in volatile memory are destroyed in the following manner:

- Memory location overwritten with DRBG generated random data.
- Memory location overwritten with zeroes.
- Memory location read and compared with zero.

All keys and CSPs in volatile memory are also destroyed by removal of power to memory when the TOE enters the G2(S5) or G3 Compliant power saving state.

The evaluator shall check to ensure the TSS lists each type of key that is stored in non-volatile memory, and identifies how the TOE interacts with the underlying platform to manage keys (e.g., store, retrieve, destroy). The description includes details on the method of how the TOE interacts with the platform, including an identification and description of the interfaces it uses to manage keys (e.g., file system APIs, platform key store APIs).

The evaluator examines the interface description for each different media type to ensure that the interface supports the selection(s) and description in the TSS.

Section 2 of [KMD] (“Keys and Key Hierarchy”) states the ONTAP Onboard Key Manager component of the TOE uses the OKM database (file location: `/cfcard/kmip/km_onboard.wkeydb`) and ONTAP’s replicated database (RDB) (directory location: `/mroot/etc/cluster_config/rdb/Management`) when storing data in non-volatile memory.

Section 2.4 of [KMD], Table 4 identifies the following values that are stored in non-volatile memory:

- Cluster Salt—stored in an RDB table and the OKM database
- CP-Salt—stored in an RDB table and the OKM database
- CP-Hash—stored in an RDB table
- Wrapped CKEK (wCKEK)—stored in an RDB table and the OKM database
- Wrapped AK (wAK)—stored in an RDB table and the OKM database.

None of these values represent keys, key material, or CSPs that require protection. The Cluster Salt is used as the salt value for the PBKDF2 function used to derive the CP-KEK from the Cluster Passphrase. The CP-Salt is used with the Cluster Passphrase to generate the CP-Hash. The CP-Hash is the SHA-256 hash of the CP-Salt concatenated with the Cluster Passphrase. Knowledge of the Cluster Salt, CP-Salt, or CP-Hash does not provide any information that could be used to derive the Cluster Passphrase or any of the keys in the key hierarchy. The wCKEK and wAK, by their nature, are protected and knowledge of them does not provide any information that could be used to derive the Cluster Passphrase or any keys in the key hierarchy.

Section 4.6 of [KMD] states the TOE uses `SecureFileDeleter::remove()` to clear keys stored in the OKM database and `cluster_kdb::remove_imp()` to remove entries from the RDB.

Section 1.3 of [KMD] (“Authorization Acquisition Overview”) describes how the TOE abstracts physical storage devices and can be used with HDD, SSD, hybrid, and array LUN technologies, and with any RAID type. Section 2 of [KMD] describes the keys and key material stored in non-volatile memory and identifies how the TOE interacts with the underlying platform to manage the keys.

The evaluator shall check that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement.

The ST does not identify any configuration or circumstances that do not strictly conform to the key destruction requirement.

If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the TSS to ensure it describes how that pattern is obtained and used. The evaluator shall verify that the pattern does not contain any CSPs.

The ST does not make use of the open assignment.

2.1.5.2 Guidance Activities

There are a variety of concerns that may 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 relevant Required Supplementary Information. 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-leveling and garbage collection. This may create additional copies of the key that are logically inaccessible but persist physically. In this case, it is assumed the drive supports the TRIM command and implements garbage collection to destroy these persistent copies when not actively engaged in other tasks.

Drive vendors implement garbage collection in a variety of different ways, as such there is a variable amount of time until data is truly removed from these solutions. There is a risk that data may persist for a longer amount of time if it is contained in a block with other data not ready for erasure. It is assumed the operating system and file system of the OE support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion.

It is assumed that if a RAID array is being used, only set-ups that support TRIM are utilized. It is assumed if the drive is connected via PCI-Express, the operating system supports TRIM over that channel. It is assumed the drive is healthy and contains minimal corrupted data and will be end of life before a significant amount of damage to drive health occurs, it is assumed there is a risk small amounts of potentially recoverable data may remain in damaged areas of the drive.

Finally, it is assumed the keys are not stored using a method that would be inaccessible to TRIM, such as being contained in a file less than 982 bytes which would be completely contained in the master file table.

As noted above, none of the values stored in non-volatile memory represent keys, key material, or CSPs that require protection. Furthermore, the ST does not identify any configuration or circumstances that do not strictly conform to the key destruction requirement.

2.1.5.3 Test Activities

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

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

7. Break the key value from Step #1 into 3 similar sized pieces and perform a search using each piece.

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

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

The evaluator confirmed the keys were generated and recorded their values. The evaluator then deleted the keys and confirmed via memory dumps that no traces of the full or partial keys remained in memory.

The following tests apply only to selection a), since the TOE in this instance has more visibility into what is happening within the underlying platform (e.g., a logical view of the media). In selection b), the TOE has no visibility into the inner workings and completely relies on the underlying platform, so there is no reason to test the TOE beyond test 1.

For selection a), the following tests are used to determine the TOE is able to request the platform to overwrite the key with a TOE supplied pattern.

Test 2: Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use a tool that provides a logical view of the media (e.g., MBR file system):

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

This test is not applicable because no keys are held in non-volatile memory.

Test 3: Applied to each key held as non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator shall use a tool that provides a logical view of the media:

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

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

This test is not applicable because no keys are held in non-volatile memory.

2.1.6 Cryptographic Key and Key Material Destruction (Destruction Timing) (FCS_CKM_EXT.4(a))

2.1.6.1 TSS Activities

The evaluator shall verify the TSS provides a high level description of what it means for keys and key material to be no longer needed and when then should be expected to be destroyed.

Section 7.1.6 of [ST] (“FCS_CKM_EXT.4(a) Cryptographic Key and Key Material Destruction (Destruction Timing)”) states the TOE deletes all keys and key material when no longer needed. The AKs are not persistently stored, but the wrapped AKs are persistently stored in the OKM database and are destroyed when the OKM is deleted. In addition, Section 7.1.5 of [ST] (“FCS_CKM.4(d): Cryptographic Key Destruction (Software TOE, 3rd Party Storage)”) states the following keys are destroyed when the OKM is deleted: CP-KEK; CKEK; wCKEK; AKs; and wAKs. The OKM is deleted under the following conditions: there is no further need for a key manager; migration to an external configuration.

2.1.6.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.6.3 KMD Activities

The evaluator shall verify the KMD includes a description of the areas where keys and key material reside and when the keys and key material are no longer needed.

Section 2.4 of [KMD] (“Critical Security Parameters Summary”) describes each of the keys, key material, and critical security parameters in the key hierarchy and where they reside, while Section 4.6 (“Key Destruction”) describes the circumstances under which keys and key material are no longer needed and are destroyed.

The evaluator shall verify the KMD includes a key lifecycle, that includes a description where key material reside, how the key material is used, how it is determined that keys and key material are no longer needed, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM.4(a) for the destruction.

Section 2 of [KMD] (“Keys and Key Hierarchy”) provides the key lifecycle, including descriptions of where key material resides and how it is used. Section 4.6 describes how it is determined keys and key material are no longer needed and how they are destroyed. The descriptions are consistent with the requirements in FCS_CKM.4(a).

2.1.6.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.7 Cryptographic Key and Key Material Destruction (Power Management) (FCS_CKM_EXT.4(b))

2.1.7.1 TSS Activities

The evaluator shall verify the TSS provides a description of what keys and key material are destroyed when entering any Compliant power saving state.

Section 7.1.4 of [ST] (“FCS_CKM.4(a): Cryptographic Key Destruction (Power Management)”) states the TOE provides the Compliant power saving states of G2(S5) (soft off) and G3 (mechanical off). Section 7.1.7 of [ST] (“FCS_CKM_EXT.4(b): Cryptographic Key and Key Material Destruction (Power Management)”) states the TOE clears the CP, CP-KEK, CKEK, and AKs when power to the memory is removed, as occurs whenever the G2(S5) or G3 power saving states are entered.

2.1.7.2 Guidance Activities

The evaluator shall validate that guidance documentation contains clear warnings and information on conditions in which the TOE may end up in a non-Compliant power saving state indistinguishable from a Compliant power saving state. In that case it must contain mitigation instructions on what to do in such scenarios.

Section 7.1.4 of [ST] states the TOE provides the Compliant power saving states of G2(S5) (soft off) and G3 (mechanical off). It further states the TOE must be fully rebooted from each Compliant power saving state. As such, the TOE is either operational, or is in a state that requires it to be fully rebooted in order to resume operation, and there are no non-Compliant power saving states indistinguishable from the Compliant power saving states supported by the TOE. As such, there is no need for any warnings or mitigation instructions in the guidance documentation.

2.1.7.3 KMD Activities

The evaluator shall verify the KMD includes a description of the areas where keys and key material reside.

Section 2.4 of [KMD] (“Critical Security Parameters Summary”) describes where keys and key material reside.

The evaluator shall verify the KMD includes a key lifecycle that includes a description where key material resides, how the key material is used, and how the material is destroyed once it is not needed and that the documentation in the KMD follows FCS_CKM.4(d) for the destruction.

Section 2 of [KMD] (“Keys and Key Hierarchy”) provides the key lifecycle, including descriptions of where key material resides and how it is used. Section 4.6 of [KMD] (“Key Lifetime and Destruction”) describes how it is determined keys and key material are no longer needed and how they are destroyed. The descriptions are consistent with the requirements in FCS_CKM.4(d).

2.1.7.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.8 Cryptographic Operation (Signature Verification) (FCS_COP.1(a))

This requirement is used to verify digital signatures attached to updates from the TOE manufacturer before installing those updates on the TOE. Because this component is to be used in the update function, additional Evaluation Activities to those listed below are covered in other evaluation activities sections in this document. The following activities deal only with the implementation for the digital signature algorithm; the evaluator performs the testing appropriate for the algorithm(s) selected in the component.

Hash functions and/or random number generation required by these algorithms must be specified in the ST; therefore the Evaluation Activities associated with those functions are contained in the associated Cryptographic Hashing and Random Bit Generation sections. Additionally, the only function required by the TOE is the verification of digital signatures. If the TOE generates digital signatures to support the

implementation of any functionality required by this cPP, then the applicable evaluation and validation scheme must be consulted to determine the required evaluation activities.

2.1.8.1 TSS Activities

The evaluator shall check the TSS to ensure that it describes the overall flow of the signature verification. This should at least include identification of the format and general location (e.g., "firmware on the hard drive device" rather than "memory location 0x00007A4B") of the data to be used in verifying the digital signature; how the data received from the operational environment are brought on to the device; and any processing that is performed that is not part of the digital signature algorithm (for instance, checking of certificate revocation lists).

Section 7.1.8 of [ST] ("FCS_COP.1(a): Cryptographic Operation (Signature Verification)") states the TOE implements the RSA digital signature algorithm with a key size (modulus) of 3072 bits and SHA-384 signatures to verify authenticity of trusted updates.

Upon receiving an update and the signature file, the TOE uses the embedded public key stored in the firmware on the NetApp appliance. The TOE will verify the signature before installing the update and reject any update with an invalid signature.

2.1.8.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.8.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.8.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 7.1 of [ST] ("Cryptographic Support"), Table 9 ("NetApp Cryptographic Security Module (NCSM) Algorithm Certificates") identifies the CAVP certification verifying RSA signature verification, as follows.

Algorithm	Tested Capabilities	Certificates
RSA as defined in FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1-v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3, for RSA schemes	FIPS 186-4: Signature Generation PSS: Mod 3072: SHA-384 Signature Generation PKCS1.5: Mod 3072: SHA-384 Signature Verification PSS: Mod 3072: SHA-384 Signature Verification PKCS1.5: Mod 3072: SHA-384	C2114: RSA SigGen (186-4) C2114: RSA SigVer (186-4)

2.1.9 Cryptographic Operation (Hash Algorithm) (FCS_COP.1(b))

2.1.9.1 TSS Activities

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

Section 7.1.9 of [ST] (“FCS_COP.1(b): Cryptographic Operation (Hash Algorithm)”) states the TOE uses the SHA-384 hash function as part of the RSA signature verification function for trusted updates and uses the SHA-512 function as part of the PBKDFv2 to derive the Cluster Passphrase Key Encryption Key (CP-KEK). In addition, the TOE uses SHA-256 to generate the CP-Hash value that is used to validate the Cluster Passphrase when it is entered by an administrator.

2.1.9.2 Guidance Activities

The evaluator checks the operational guidance documents to determine that any system configuration necessary to enable required hash size functionality is provided.

No system configuration is necessary to enable required hash functionality. As such, this activity is not applicable.

2.1.9.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.9.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 7.1 of [ST] (“Cryptographic Support”), Table 8 (“CryptoMod version 2.2 Algorithm Certificates”) identifies the CAVP certification verifying the SHA hash algorithm implemented by CryptoMod, as follows.

Algorithm	Tested Capabilities	Certificates
SHS as defined in ISO/IEC 10118-3:2004	SHA-256	C1884: SHA-256
	SHA-512	C1884: SHA-512

Section 7.1 of [ST] (“Cryptographic Support”), Table 9 (“NetApp Cryptographic Security Module (NCSM) Algorithm Certificates”) identifies the CAVP certification verifying the SHA hash algorithm implemented by NCSM, as follows.

Algorithm	Tested Capabilities	Certificates
SHS as defined in ISO/IEC 10118-3:2004	SHA-256	C2114: SHA-256
	SHA-384	C2114: SHA-384
	SHA-512	C2114: SHA-512

2.1.10 Cryptographic Operation (Keyed Hash Algorithm) (FCS_COP.1(c))

2.1.10.1 TSS Activities

If HMAC was selected:

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

Section 7.1.10 of [ST] (“FCS_COP.1(c): Cryptographic Operation (Keyed Hash Algorithm)”) specifies the key lengths, block size, output MAC length, and hash function used by the TOE’s HMAC implementation.

If CMAC was selected:

The evaluator shall examine the TSS to ensure that it specifies the following values used by the CMAC function: key length, block cipher used, block size (of the cipher), and output MAC length used.

The ST does not select CMAC.

2.1.10.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.10.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.10.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 7.1 of [ST] (“Cryptographic Support”), Table 8 (“CryptoMod version 2.2 Algorithm Certificates”) identifies the CAVP certification verifying HMAC keyed hash algorithm, as follows.

Algorithm	Tested Capabilities	Certificates
HMAC that meets: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	HMAC-SHA-512 Key sizes < block size Key size = block size Key sizes > block size	C1884: HMAC-SHA2-512

2.1.11 Cryptographic Operation (Key Wrapping) (FCS_COP.1(d))

2.1.11.1 TSS Activities

The evaluator shall verify the TSS includes a description of the key wrap function(s) and shall verify the key wrap uses an approved key wrap algorithm according to the appropriate specification.

Section 7.1.11 of [ST] (“FCS_COP.1(d): Cryptographic Operation (Key Wrapping)”) states the TOE uses the KWP-AE(P) key wrapping function defined in NIST 800-38F to wrap keys, and the corresponding KWP-AD(C) function to unwrap keys. These are approved algorithms, consistent with the selections made in FCS_COP.1(d).

2.1.11.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.11.3 KMD Activities

The evaluator shall review the KMD to ensure that all keys are wrapped using the approved method and a description of when the key wrapping occurs.

Section 2.4 of [KMD] (“Critical Security Parameters Summary”) identifies two keys in the key hierarchy that are wrapped: Cluster Key Encryption Key (CKEK); and Authentication Key (AK).

The CKEK is wrapped using KWP-AE as defined in NIST SP 800-38F, using the Cluster Passphrase Key Encryption Key (CP-KEK) as the encryption key.

The AKs (the TOE supports two Authentication Keys, designated as AK1 and AK2) are wrapped using KWP-AE as defined in NIST SP 800-38F, using the CKEK as the encryption key.

Section 2.1.6 of [KMD] (“Cluster Key Encryption Key (CKEK)”) states the CKEK, AK1, and AK2 are wrapped (to produce wCKEK, wAK1, and wAK2 respectively) in the `crypto_create_onboard_key_hierarchy_v2` function call that is used to create the OKM base hierarchy keys.

2.1.11.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.12 Cryptographic Key Derivation (FCS_KDF_EXT.1)

2.1.12.1 TSS Activities

The evaluator shall verify the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108 and SP 800-132.

Section 7.1.12 of [ST] (“FCS_KDF_EXT.1: Cryptographic Key Derivation”) states the TOE implements PBKDFv2 with HMAC-SHA-512, 1024 iterations, and a salt value of 512 bits to transform the user password into a derived key as specified in SP 800-132.

2.1.12.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.12.3 KMD Activities

The evaluator shall examine the vendor’s KMD to ensure that all keys used are derived using an approved method and a description of how and when the keys are derived.

Section 2.3 of [KMD] (“Key Hierarchy”) describes the source of each of the keys and critical security parameters in the key hierarchy. It identifies the following as being derived from other material:

- Cluster Passphrase Key Encryption Key (CP-KEK)—derived from the Cluster Passphrase and Cluster Salt using PBKDFv2 as specified in NIST SP 800-132, using HMAC-SHA-512 as the pseudo-random function, with a 512 bit random salt and 1,024 iterations
- CP-Hash—derived from the CP-Salt and the CP using SHA-256.

The CP-Salt, Cluster-Salt, CKEK, and AKs are generated by the TOE using its approved CTR_DRBG function, while the Cluster Passphrase is entered by the cluster administrator at boot time. All key generation and derivation operations occur when the OKM base hierarchy keys are created.

2.1.12.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.13 Key Chaining (Initiator) (FCS_KYC_EXT.1)

2.1.13.1 TSS Activities

The evaluator shall verify the TSS contains a high-level description of the BEV sizes – that it supports BEV outputs of no fewer 128 bits for products that support only AES-128, and no fewer than 256 bits for products that support AES-256.

Section 7.1.13 of [ST] (“FCS_KYC_EXT.1: Key Chaining (initiator)”) states the TOE uses the PBKDFv2 key derivation function to derive the CP-KEK from the Cluster Passphrase. As stated in Section 7.1.5 of [ST] (“FCS_CKM.4(d): Cryptographic Key Destruction (Software TOE, 3rd Party Storage”), Table 10 (“Critical Security Parameters”), the CP-KEK is a 256 bit value used as an AES key. The CP-KEK is used to protect the CKEK, which is also a 256 bit AES key, generated by the TOE’s DRBG. The CKEK, in turn, is used to protect the Authentication Keys, which constitute the BEV. The Authentication Keys are also 256 bit AES keys, generated by the TOE’s DRBG. Therefore, the TOE is able to support 256 bit BEV outputs, consistent with its use of AES-256.

2.1.13.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.13.3 KMD Activities

The evaluator shall examine the KMD describes a high level description of the key hierarchy for all authorizations methods selected in FCS_AFA_EXT.1 that are used to protect the BEV. The evaluator shall examine the KMD to ensure it describes the key chain in detail. The description of the key chain shall be reviewed to ensure it maintains a chain of keys using key wrap or key derivation methods that meet FCS_COP.1(d) and FCS_KDF_EXT.1.

Section 2 of [KMD] (“Keys and Key Hierarchy”) describes the various keys within the key hierarchy and how they are used to protect the BEV. It provides a detailed description of the key chain.

The TOE supports a single authorization factor (FCS_AFA_EXT.1), the Cluster Passphrase (CP), which is a 64-256 byte, user-defined ASCII string. The TOE uses its approved CTR_DRBG function to generate a 256 bit random number called the Cluster Salt (CS). The CS is concatenated with the CP to form a bit string of between 512 and 2560 bits (depending on the length of CP). The TOE uses the PBKDFv2 function to derive the Cluster Passphrase Key Encryption Key (CP-KEK) from the concatenation of CS and CP (FCS_KDF_EXT.1, FCS_PCC_EXT.1).

The TOE uses its approved CTR_DRBG function to generate a 256 bit random number called the Cluster Key Encryption Key (CKEK). The TOE uses the KWP-AE(P) function to wrap the CKEK using the CP-KEK (FCS_COP.1(d)).

The TOE uses its approved CTR_DRBG function to generate two 256 bit random numbers called the Authentication Keys (AK1, AK2). One AK is used for the self-encrypting drive’s data port and the other is

used for its FIPS port, and they constitute the BEV the TOE passes to the encryption engine. The TOE uses the KWP-AE(P) function to wrap the AKs using the CKEK (FCS_COP.1(d)).

The evaluator shall examine the KMD to ensure that it describes how the key chain process functions, such that it does not expose any material that might compromise any key in the chain. (e.g. using a key directly as a compare value against a TPM) This description must include a diagram illustrating the key hierarchy implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or the initial authorization value and the effective strength of the BEV is maintained throughout the key chain.

Section 2.3 of [KMD] (“Key Hierarchy”) describes how the key chain process functions. The key chain process does not expose any material that might compromise a key in the chain. When keys are generated, they either are stored in volatile (non-persistent) storage in the key table of the TOE’s cryptomodule, or they are wrapped before being stored in non-volatile (persistent) storage.

Section 2.3 includes Figure 6 (“Key Hierarchy”), which illustrates the key hierarchy implemented by the TOE. Section 2.4 (“Critical Security Parameters Summary”) includes a table that details where all keys and keying material are stored and how each key and critical security parameter originates or is derived.

The evaluator’s examination of the key hierarchy determined that at no point can the chain be broken without a cryptographic exhaust or possession of the initial authorization value. The effective strength of the BEV of 256 bits is maintained throughout the key hierarchy.

The evaluator shall verify the KMD includes a description of the strength of keys throughout the key chain.

The table in Section 2.4 describes the strength of every key and piece of keying material used throughout the key chain.

2.1.13.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.14 Cryptographic Password Construct and Conditioning (FCS_PCC_EXT.1)

2.1.14.1 TSS Activities

The evaluator shall ensure the TSS describes the manner in which the TOE enforces the construction of passwords, including the length, and requirements on characters (number and type). The evaluator also verifies that the TSS provides a description of how the password is conditioned and the evaluator ensures it satisfies the requirement.

Section 7.1.14 of [ST] (“FCS_PCC_EXT.1: Cryptographic Password Construct and Conditioning”) states the TOE accepts passwords up to 256 characters. The character set can consist of all upper case characters, lower case characters, numbers, and all printable ASCII characters. The password is conditioned using PBKDFv2 that meets SP 800-132. The cryptographic algorithm implements HMAC-SHA-512, a salt value of 512 bits from the DRBG, and 1024 iterations to produce a cryptographic key size of 256-bits. This description satisfies the statement of FCS_PCC_EXT.1.1 in Section 6.2.1.14 of [ST] (“FCS_PCC_EXT.1 Cryptographic Password Construct and Conditioning”).

2.1.14.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.14.3 KMD Activities

The evaluator shall examine the KMD to ensure that the formation of the BEV and intermediary keys is described and that the key sizes match that selected by the ST author.

Section 2.1 of [KMD] (“Border Encryption Values”) describes the formation of the Border Encryption Value (BEV) and intermediary keys. In TOE terms, the Authentication Key (AK) is the BEV. The AK is 256 bits, consistent with the key size selected by the ST author.

The evaluator shall check that the KMD describes the method by which the password/passphrase is first encoded and then fed to the SHA algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator shall verify that these are supported by the selections in this component as well as the selections concerning the hash function itself. The evaluator shall verify that the KMD contains a description of how the output of the hash function is used to form the submask that will be input into the function and is the same length as the BEV as specified above.

Section 2.1.1 of [KMD] (“Cluster Passphrase”) describes the method by which the password is encoded and fed to the SHA algorithm. The password is a 64 to 256 byte ASCII string that is concatenated with a salt generated by the TOE’s DRBG and fed to the PBKDFv2 function that meets NIST SP 800-132 to derive the Cluster Passphrase Key Encryption Key (CP-KEK). Section 2.1.5 of [KMD] (“Cluster Passphrase Key Encryption Key (CP-KEK)”) describes the parameters of the PBKDFv2 function, consistent with the selections in the ST—using the HMAC-SHA-512 algorithm with 1024 iterations and an output size of 256 bits.

2.1.14.4 Test Activities

The evaluator shall also perform the following tests:

Test 1: Ensure that the TOE supports passwords/passphrases of a minimum length of 64 characters.

The evaluator attempted to configure a cluster passphrase on the TOE of 64 characters and verified it was accepted.

Test 2: If the TOE supports a password/passphrase length up to a maximum number of characters, n (which would be greater than 64), then ensure that the TOE will not accept more than n characters.

The TOE only accepts passphrases up to 256 characters. The evaluator attempted to configure a cluster passphrase on the TOE that was 257 characters and confirmed this attempt was denied.

Test 3: Ensure that the TOE supports passwords consisting of all characters assigned and supported by the ST author.

The evaluator attempted to configure the TOE with a passphrase consisting of at least one of each printable ASCII character and confirmed that it was accepted.

2.1.15 Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

2.1.15.1 TSS Activities

For any RBG services provided by a third party, the evaluator shall ensure the TSS includes a statement about the expected amount of entropy received from such a source, and a full description of the processing of the output of the third-party source. The evaluator shall verify that this statement is consistent with the selection made in FCS_RBG_EXT.1.2 for the seeding of the DRBG. If the ST specifies

more than one DRBG, the evaluator shall examine the TSS to verify that it identifies the usage of each DRBG mechanism.

Section 7.1.15 of [ST] (“FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)”) states random bits are produced by a DRBG implemented within the TOE’s own cryptographic module (NetApp CryptoMod). As such, there is no reliance on third party RBG services.

Section 7.1.15 of [ST] indicates the TOE implements a single DRBG mechanism, CTR_DRBG using AES-256, which is seeded using random bits read from the `/dev/random` pseudo-device implemented by the TOE.

2.1.15.2 Guidance Activities

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected DRBG mechanism(s), if necessary, and provides information regarding how to instantiate/call the DRBG for RBG services needed in this cPP.

The TOE provides a single DRBG mechanism implemented within its own cryptographic module and used by default. As such, no configuration is required and no guidance documentation is necessary.

2.1.15.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.15.4 Test Activities

Performed in accordance with NIAP Policy Letter #5.

Section 7.1 of [ST] (“Cryptographic Support”), Table 8 (“CryptoMod version 2.2 Algorithm Certificates”) identifies the CAVP certification verifying random bit generation implemented by CryptoMod, as follows.

Algorithm	Tested Capabilities	Certificates
CTR_DRBG (AES)	Counter: Modes: AES-256	C1884: Counter DRBG

2.1.16 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation) (FCS_SNI_EXT.1)

2.1.16.1 TSS Activities

The evaluator shall ensure the TSS describes how salts are generated. The evaluator shall confirm that the salt is generating using an RBG described in FCS_RBG_EXT.1 or by the Operational Environment. If external function is used for this purpose, the TSS should include the specific API that is called with inputs.

Section 7.1.16 of [ST] (“FCS_SNI_EXT.1 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)”) describes how salts are generated. It states the TOE generates salts using its own DRBG as specified in FCS_RBG_EXT.1.

The evaluator shall ensure the TSS describes how nonces are created uniquely and how IVs and tweaks are handled (based on the AES mode). The evaluator shall confirm that the nonces are unique and the IVs and tweaks meet the stated requirements.

Section 6.2.1.16 of [ST] (“FCS_SNI_EXT.1 Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)”) specifies the TOE does not use nonces, IVs, or tweaks. This is consistent with the rest of the SFRs and the description in the TSS.

The TOE does not implement any of the AES modes listed in FCS_SNI_EXT.1.3 (i.e., CBC, CCM, XTS, or GCM), so this part of the requirement is vacuously satisfied.

2.1.16.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.1.16.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.1.16.4 Test Activities

There are no test evaluation activities for this SFR.

2.1.17 Validation (FCS_VAL_EXT.1)

2.1.17.1 TSS Activities

The evaluator shall examine the TSS to determine which authorization factors support validation.

Section 7.1.17 of [ST] (“FCS_VAL_EXT.1 Validation”) states the Cluster Passphrase is a 64-256 byte customer generated ASCII string that is used as an authorization factor. The TOE uses the KWP-AE(P) key wrapping function defined in NIST 800-38F to wrap keys in the key hierarchy (specifically, the CKEK and the AKs), and so validation of the Cluster Passphrase is performed inherently. During booting of the TOE or following recovery from a failure in the boot media, if an incorrect Cluster Passphrase is entered, the TOE will boot but the WAFL component will not be able to mount any user data volumes. As a consequence, the TOE will be unable to service any user data. This condition can only be cleared by rebooting the TOE and entering the correct passphrase.

The TOE also uses the CP-Hash to validate the Cluster Passphrase is entered correctly when the administrator is required to enter the Cluster Passphrase as part of a command, such as when the administrator wishes to modify the Cluster Passphrase.

The evaluator shall examine the TSS to review a high-level description if multiple submasks are used within the TOE, how the submasks are validated (e.g., each submask validated before combining, once combined validation takes place).

The Cluster Passphrase is the only submask used by the TOE.

2.1.17.2 Guidance Activities

(conditional) If the validation functionality is configurable, the evaluator shall examine the operational guidance to ensure it describes how to configure the TOE to ensure the limits regarding validation attempts can be established.

The validation functionality is not configurable in the evaluated configuration (i.e., `cc-mode-enabled=yes`). The administrator has one attempt to enter the correct Cluster Passphrase at boot or after recovering from a failure in the boot media, and five attempts to enter the correct current Cluster Passphrase when attempting to modify the Cluster Passphrase.

(conditional) If the validation functionality is specified by the ST author, the evaluator shall examine the operational guidance to ensure that it states the values that the TOE uses for limits regarding validation attempts.

Section “Configuring NetApp hardware-based encryption > Configuring onboard key management > Enabling onboard key management in ONTAP 9.6 and later” of [NEPG] states if the administrator fails to enter the correct Cluster Passphrase at boot, encrypted volumes will not be mounted. To correct this, the administrator needs to reboot the node and enter the correct Cluster Passphrase. Once the TOE is booted, it allows up to 5 consecutive attempts to correctly enter the Cluster Passphrase in a 24 hour period or any command that requires the Cluster Passphrase as a parameter (e.g., to change the Cluster Passphrase). Note, however, this is to validate the administrator is authorized to perform the command and does not entail validation of the BEV (which was done during the boot process).

2.1.17.3 KMD Activities

The evaluator shall examine the KMD to verify that it described the method the TOE employs to limit the number of consecutively failed authorization attempts.

Section 2 of [KMD] (“Keys and Key Hierarchy”) describes the key chain from the authorization factor (the Cluster Passphrase) down to the BEV (the AK). The TOE uses the KWP-AE(P) key wrapping function defined in NIST 800-38F to wrap keys in the key chain (specifically, the CKEK and the AKs), and so validation of the Cluster Passphrase is performed inherently. During booting of the TOE or following recovery from a failure in the boot media, if an incorrect Cluster Passphrase is entered, the TOE will boot but the WAFL component will not be able to mount any user data volumes. As a consequence, the TOE will be unable to service any user data. This condition can only be cleared by rebooting the TOE and entering the correct passphrase.

Section 2.1.1 of [KMD] (“Cluster Passphrase”) describes the TOE’s methods for limiting the number of consecutive failed authentication attempts. The TOE allows five such attempts when a user attempts to change the Cluster Passphrase, or perform some other command that requires the Cluster Passphrase as a parameter. When five consecutive failed attempts occur, one or more nodes within the cluster must be successfully rebooted, or a period of 24 hours needs to elapse before the user is allowed to change the Cluster Passphrase.

The evaluator shall examine the vendor’s KMD to ensure it describes how validation is performed. The description of the validation process in the KMD provides detailed information how the TOE validates the submasks. The KMD describes how the process works, such that it does not expose any material that might compromise the submask(s).

As noted above, validation of the authorization factor is performed inherently during unwrapping of the keys in the key chain to the BEV.

In addition, as described in Section 2.1.4 of [KMD] (“CP-Hash”), for validation of correct entry of the Cluster Passphrase as a command parameter, the TOE uses the CP-Hash. The CP-Hash is formed as follows: the TOE uses its cryptomodule to generate a random 64-byte value, the CP-Salt; the CP-Salt is concatenated with the Cluster Passphrase and a SHA-256 hash is calculated over this string; the result is the CP-Hash, which is stored persistently in the RDB. Any time the Cluster Passphrase is entered after the TOE has booted (e.g., when the administrator attempts to change the Cluster Passphrase), the TOE validates the Cluster Passphrase by concatenating the entered Cluster Passphrase with the persistently stored CP-Salt, calculating the SHA-256 hash of the concatenated string, and comparing the result with the stored CP-

Hash. If the values match, the correct Cluster Passphrase has been entered and the TOE can proceed. Otherwise, an incorrect Cluster Passphrase has been entered and the requested operation will fail.

2.1.17.4 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator shall determine the limit on the average rate of the number of consecutive failed authorization attempts. The evaluator will test the TOE by entering that number of incorrect authorization factors in consecutive attempts to access the protected data. If the limit mechanism includes any “lockout” period, the time period tested should include at least one such period. Then the evaluator will verify that the TOE behaves as described in the TSS.

The evaluator confirmed that the encrypted volume was online then rebooted the TOE and input the incorrect cluster passphrase then confirmed the volume was offline. The evaluator then rebooted the TOE again and input the correct cluster passphrase and confirmed that the volume was online.

Test 2: For each validated authorization factor, ensure that when the user provides an incorrect authorization factor, the TOE prevents the BEV from being forwarded outside the TOE (e.g., to the EE).

The evaluator logged into the TOE and confirmed the BEV could be accessed. The evaluator then rebooted the TOE and provided the incorrect cluster passphrase. The evaluator then confirmed that the BEV was not accessible to be queried or forwarded off the TOE.

2.2 Security Management (FMT)

2.2.1 Management of Functions Behavior (FMT_MOF.1)

2.2.1.1 TSS Activities

If support for Compliant power saving state(s) are claimed in the ST, the evaluator shall ensure the TSS describes how these are managed and shall ensure that TSS describes how only privileged users (administrators) are allowed to manage the states.

Section 7.2.1 of [ST] (“FMT_MOF.1 Management of Functions Behavior”) states the TOE provides the Compliant power saving states of G2(S5) (soft off) and G3 (mechanical off). The TOE transitions to the G2(S5) state through execution of the `system node halt` command. This command is restricted to cluster administrators at the **admin** privilege level. The TOE transitions to the G3 state when the administrator removes device power via mechanical switch.

2.2.1.2 Guidance Activities

The evaluator to check if guidance documentation describes which authorization factors are required to change Compliant power saving state behavior and properties.

The TOE does not support modification of the behavior of Compliant power saving states or their properties. The TOE supports only a single powered state and the Compliant power saving states of G2(S5) (soft off) and G3 (mechanical off). The TOE enters the powered state when power is applied to the TOE and it boots up. The TOE enters the G2(S5) state when an administrator executes the `system node halt` command, and enters the G3 state via removal of power.

2.2.1.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.2.1.4 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator presents a privileged authorization credential to the TSF and validates that changes to Compliant power saving state behavior and properties are allowed.

The evaluator logged in to the TOE as an authorized user and confirmed that the user could change the compliant power state.

Test 2: The evaluator presents a non-privileged authorization credential to the TSF and validates that changes to Compliant power saving state behavior are not allowed.

The evaluator logged in to the TOE as an unauthorized user and confirmed that the user could not change the compliant power state.

2.2.2 Specification of Management Functions (FMT_SMF.1)

2.2.2.1 TSS Activities

If item a) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE sends the request to the EE to change the DEK.

Section 7.2.2 of [ST] (“FMT_SMF.1: Specification of Management Functions”) states the TOE forwards requests to change the DEK to the EE when the authorized user executes the `storage encryption disk sanitize` CLI command.

If item b) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes how the TOE sends the request to the EE to cryptographically erase the DEK.

Section 7.2.2 of [ST] states the TOE cryptographically erases the DEK via the authorized user execution of the `storage encryption disk destroy` CLI command.

If item c) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the methods by which users may change the set of all authorization factor values supported.

Section 7.2.2 of [ST] states the TOE permits users to change the TOE’s authorization factor via execution of the `security key-manager onboard update-passphrase` CLI command.

If item d) is selected in FMT_SMF.1.1: The evaluator shall ensure the TSS describes the process to initiate TOE firmware/software updates.

Section 7.2.2 of [ST] states the TOE initiates TOE firmware/software updates via execution of the `cluster image update` CLI command.

If item e) is selected in FMT_SMF.1.1: If power saving states can be managed, the evaluator shall ensure that the TSS describes how this is performed, including how the TOE supports disabling certain power saving states if more than one are supported. If additional management functions are claimed in the ST, the evaluator shall ensure the TSS describes the additional functions.

The ST selects “no other functions” for item e) in FMT_SMF.1.

2.2.2.2 Guidance Activities

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how the functions for A and B can be initiated by the user.

Section “storage commands” in [CMPR] describes the CLI commands available to the authorized administrator to initiate changing the DEK (storage encryption disk destroy) and cryptographically erasing the DEK (storage encryption disk destroy).

If item c) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how selected authorization factor values are changed.

Section “security commands” in [CMPR] describes the CLI command available to the authorized administrator to change the TOE’s authorization factor (security key-manager onboard update-passphrase).

If item d) is selected in FMT_SMF.1.1: The evaluator shall examine the operational guidance to ensure that it describes how to initiate TOE firmware/software updates.

Section “cluster commands” in [CMPR] describes the CLI command available to the authorized administrator to initiate a TOE firmware/software update (cluster image update).

If item e) is selected in FMT_SMF.1.1: Default Authorization Factors: It may be the case that the TOE arrives with default authorization factors in place. If it does, then the selection in section E must be made so that there is a mechanism to change these authorization factors. The operational guidance shall describe the method by which the user changes these factors when they are taking ownership of the device. The TSS shall describe the default authorization factors that exist.

Disable Key Recovery: The guidance for disabling this capability shall be described in the AGD documentation.

Power Saving: The guidance shall describe the power saving states that are supported by the TSF, how these states are applied, how to configure when these states are applied (if applicable), and how to enable/disable the use of specific power saving states (if applicable).

The ST selects “no other functions” for item e) in FMT_SMF.1.

2.2.2.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.2.2.4 Test Activities

If item a) and/or b) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to forward a command to the EE to change and cryptographically erase the DEK. The actual testing of the cryptographic erase will take place in the EE.

The evaluator recorded the value of the VDEK and then issued a rekey command on the TOE. The evaluator then queried the VDEK and confirmed that it was a new value.

If item c) is selected in FMT_SMF.1.1: The evaluator shall initialize the TOE such that it requires the user to input an authorization factor in order to access encrypted data.

Test 1: The evaluator shall first provision user authorization factors, and then verify all authorization values supported allow the user access to the encrypted data. Then the evaluator shall exercise the management functions to change a user's authorization factor values to a new one. Then he or she will verify that the TOE denies access to the user's encrypted data when he or she uses the old or original authorization factor values to gain access.

The evaluator booted the TOE and input the correct cluster passphrase and confirmed that the volume was online. The evaluator then changed the cluster passphrase, rebooted the TOE and provided the previous (incorrect) cluster passphrase. The evaluator confirmed that the volume was offline and not accessible.

If item d) is selected in FMT_SMF.1.1: The evaluator shall verify that the TOE has the functionality to initiate TOE firmware/software updates.

If item e) is selected in FMT_SMF.1.1: If additional management functions are claimed, the evaluator shall verify that the additional features function as described.

Test 2 (conditional): If the TOE provides default authorization factors, the evaluator shall change these factors in the course of taking ownership of the device as described in the operational guidance. The evaluator shall then confirm that the (old) authorization factors are no longer valid for data access.

Test 3 (conditional): If the TOE provides key recovery capability whose effects are visible at the TOE interface, then the evaluator shall devise a test that ensures that the key recovery capability has been or can be disabled following the guidance provided by the vendor.

Test 4 (conditional): If the TOE provides the ability to configure the power saving states that are entered by certain events, the evaluator shall devise a test that causes the TOE to enter a specific power saving state, configure the TSF so that this activity causes a different state to be entered, repeat the activity, and observe the new state is entered as configured.

Test 5 (conditional): If the TOE provides the ability to disable the use of one or more power saving states, the evaluator shall devise a test that enables all supported power saving states and demonstrates that the TOE can enter into each of these states. The evaluator shall then disable the supported power saving states one by one, repeating the same set of actions that were performed at the start of the test, and observe each time that when a power saving state is configured to no longer be used, none of the behavior causes the disabled state to be entered.

The evaluator performed testing for item d) in conjunction with FPT_TUD_EXT.1 testing. The ST selects "no other functions" for item e) in FMT_SMF.1. Therefore, none of these tests are applicable.

2.2.3 Security Roles (FMT_SMR.1)

2.2.3.1 TSS Activities

There are no TSS evaluation activities for this SFR. Evaluation of this SFR is performed as part of evaluating FMT_MOF.1 and FMT_SMF.1.

2.2.3.2 Guidance Activities

There are no guidance evaluation activities for this SFR. Evaluation of this SFR is performed as part of evaluating FMT_MOF.1 and FMT_SMF.1.

2.2.3.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.2.3.4 Test Activities

There are no test evaluation activities for this SFR. Evaluation of this SFR is performed as part of evaluating FMT_MOF.1 and FMT_SMF.1.

2.3 Protection of the TSF (FPT)

2.3.1 Protection of Key and Key Material (FPT_KYP_EXT.1)

2.3.1.1 TSS Activities

Modified in accordance with TD0458.

The evaluator shall examine the TSS and verify it identifies the methods used to protect keys stored in non-volatile memory.

Table 10 of [ST] (“Critical Security Parameters”) identifies the keys, key material, and critical security parameters used by the TOE and for each describes its storage in volatile and non-volatile memory. The following CSPs are stored in non-volatile memory:

- CP-Salt—a salt value that does not require protection
- CP-Hash—a calculated hash value that does not require protection
- Cluster Salt—a salt value that does not require protection
- wCKEK—wrapped Cluster Key Encryption Key, i.e., the TOE protects the CKEK by wrapping it using NIST 800-38F KWP-AE(P) with CP-KEK used as the wrapping key
- wAK—wrapped Authentication Key, i.e., the TOE protects the AK by wrapping it using NIST 800-38F KWP-AE(P) with CKEK used as the wrapping key.

2.3.1.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.3.1.3 KMD Activities

Modified in accordance with TD0458.

The evaluator shall verify the KMD to ensure it describes the storage location of all keys and the protection of all keys stored in non-volatile memory. The description of the key chain shall be reviewed to ensure the selected method is followed for the storage of wrapped or encrypted keys in non-volatile memory and plaintext keys in non-volatile memory meet one of the criteria for storage.

Section 2.4 of [KMD] (“Critical Security Parameters Summary”) describes the storage location of all keys and how all keys are protected.

The following are stored in volatile memory:

- CP-KEK—stored in the cryptomodule’s key table
- CKEK—stored in the cryptomodule’s key table

- AK—stored in the cryptomodule’s key table.

The following are stored in non-volatile memory:

- CP-Salt—persistently stored in RDB and OKM database
- CP-Hash—persistently stored in RDB
- Cluster-Salt—persistently stored in RDB and OKM database
- wCKEK—persistently stored in RDB and OKM database, wrapped using NIST 800-38F KWP-AE(P) with CP-KEK used as the wrapping key
- wAK—persistently stored in RDB and OKM database, wrapped using NIST 800-38F KWP-AE(P) with CKEK used as the wrapping key.

The selected method (key wrapping as specified in FCS_COP.1(d)) is followed for the storage of wrapped or keys (wCKEK, wAK) in non-volatile memory. The TOE does not store plaintext keys in non-volatile memory.

Note that, in addition to the above, key material may be stored temporarily as a variable or a CPU register value by functions involved in calculations or other operations using that key material.

2.3.1.4 Test Activities

There are no test evaluation activities for this SFR.

2.3.2 Power Saving States (FPT_PWR_EXT.1)

2.3.2.1 TSS Activities

The evaluator shall validate the TSS contains a list of Compliant power saving states.

Section 7.1.4 of [ST] (“FCS_KM.4(a): Cryptographic Key Destruction (Power Management)”) states the TOE provides the Compliant power saving states of G2(S5) (soft off) and G3 (mechanical off).

2.3.2.2 Guidance Activities

The evaluator shall ensure that guidance documentation contains a list of Compliant power saving states. If additional power saving states are supported, then the evaluator shall validate that the guidance documentation states how non-Compliant power states are disabled.

Section 7.1.4 of [ST] states the TOE provides the Compliant power saving states of G2(S5) (soft off) and G3 (mechanical off). It further states the TOE must be fully rebooted from each Compliant power saving state. As such, the TOE is either operational, or is in a state that requires it to be fully rebooted in order to resume operation, and there are no other power saving states, Compliant or otherwise. As such, there is no need for instructions in the guidance documentation on how to disable non-Compliant power saving states. The guidance, including [CMPR] and [SAR], is clear on how to boot the TOE and how to transition the TOE to either of the Compliant power saving states.

2.3.2.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.3.2.4 Test Activities

The evaluator shall confirm that for each listed compliant state all key/key materials are removed from volatile memory by using the test defined in FCS_CKM.4(d).

The evaluator repeated the test defined in FCS_CKM.4(d) by testing for volatile keys after a soft reboot and hard power cycle and confirmed that no traces of the full or partial key remained for each case.

2.3.3 Timing of Power Saving States (FPT_PWR_EXT.2)

2.3.3.1 TSS Activities

The evaluator shall validate that the TSS contains a list of conditions under which the TOE enters a Compliant power saving state.

Section 7.3.2 of [ST] (“FPT_PWR_EXT.1: Power Saving States / FPT_PWR_EXT.2: Timing of Power Saving States”) states the TOE enters the G3 (mechanical off) state when the administrator removes the device’s power via a mechanical switch, and enters the G2 (S5) (soft off) state when an authorized user executes the `system node halt` command.

The TOE must be fully rebooted from each Compliant power saving state.

2.3.3.2 Guidance Activities

The evaluator shall check that the guidance contains a list of conditions under which the TOE enters a Compliant power saving state. Additionally, the evaluator shall verify that the guidance documentation states whether unexpected power-loss events may result in entry to a non-Compliant power saving state and, if that is the case, validate that the documentation contains information on mitigation measures.

Section “system commands > system node commands” of [CMPR] describes the `system node halt` command that an authorized administrator uses to move the TOE to the G2(S5) (soft off) Compliant power saving state. In addition, “Managing nodes > Shutting down a node” subsection of [SAR] provides guidance to the administrator on shutting down a node of the TOE using the `system node halt` command.

The TOE is either operational, or is in a state that requires it to be fully rebooted in order to resume operation, and there are no other power saving states, Compliant or otherwise.

2.3.3.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.3.3.4 Test Activities

The evaluator shall trigger each condition in the list of identified conditions and ensure the TOE ends up in a compliant power saving state by running the test identified in FCS_CKM.4(d).

The evaluator tested the TOE can transition to both compliant power saving states in conjunction with FPT_PWR_EXT.1.

2.3.4 TSF Testing (FPT_TST_EXT.1)

2.3.4.1 TSS Activities

The evaluator shall verify that the TSS describes the known-answer self-tests for cryptographic functions.

Section 7.3.3 of [ST] (“FPT_TST_EXT.1: TSF Testing”) states the known answer tests for cryptographic functions consist of executing each function on data for which the correct answer is already known. The output produced by the tested function is compared with the known answer. If they are not identical, the known answer test fails. The TOE implements the following known-answer tests:

- 256-AES-KWP—encryption/decryption test (for key wrapping)
- DRBG—tested per SP 800-90A, including the Health Testing identified in Section 11.3.
- HMAC SHA-512—keyed-hash message authentication code test
- PBKDF2—Password-Based Key Derivation Function 2 test
- RSA—signature generation/verification test
- SHA-256, SHA-384, SHA-512—hashing test.

The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TOE and the method by which the TOE tests those functions. The evaluator shall verify that the TSS includes each of these functions, the method by which the TOE verifies the correct operation of the function. The evaluator shall verify that the TSF data are appropriate for TSF Testing. For example, more than blocks are tested for AES in CBC mode, output of AES in GCM mode is tested without truncation, or 512-bit key is used for testing HMAC-SHA-512.

Section 7.3.3 of [ST] describes the software integrity test performed by the TOE. During power-on self-testing, the module performs a self-integrity check and compares the results against the build time generated hash digests. During power on, the bootloader validates the whitelist database of secure boot keys with the signature associated with each module that is loaded. After each module is validated and loaded, the boot process continues with the ONTAP initialization. If signature validation fails for any module, the system reboots.

If FCS_RBG_EXT.1 is implemented by the TOE and according to NIST SP 800-90, the evaluator shall verify that the TSS describes health tests that are consistent with section 11.3 of NIST SP 800-90.

Section 7.3.3 of [ST] states the TOE’s DRBG implementation is tested in accordance with SP 800-90A, including the Health Tests identified in Section 11.3.

If any FCS_COP functions are implemented by the TOE, the TSS shall describe the known-answer self-tests for those functions.

Section 7.3.3 of [ST] describes the known-answer self-tests for algorithms specified in the FCS_COP.1 requirements, as follows:

- FCS_COP.1(a): RSA signature generation and verification test
- FCS_COP.1(b): SHA-256, SHA-384, and SHA-512 hashing test
- FCS_COP.1(c): HMAC-SHA-512 keyed-hash message authentication code test
- FCS_COP.1(d): 256-AES-KWP encryption/decryption test (for key wrapping).

The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TSF, the method by which those functions are tested. The TSS will describe, for each of these functions, the method by which correct operation of the function/component is verified. The evaluator shall determine that all of the identified functions/components are adequately tested on start-up.

Section 7.3.3 of [ST] describes the software integrity test performed by the TOE. During power-on self-testing, the module performs a self-integrity check and compares the results against the build time generated hash digests. During power on, the bootloader validates the whitelist database of secure boot keys with the signature associated with each module that is loaded. After each module is validated and loaded, the boot process continues with the ONTAP initialization. If signature validation fails for any module, the system reboots.

2.3.4.2 Guidance Activities

There are no AGD evaluation activities for this SFR.

2.3.4.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.3.4.4 Test Activities

There are no test evaluation activities for this SFR.

2.3.5 Trusted Update (FPT_TUD_EXT.1)

2.3.5.1 TSS Activities

The evaluator shall examine the TSS to ensure that it describes information stating that an authorized source signs TOE updates and will have an associated digital signature. The evaluator shall examine the TSS contains a definition of an authorized source along with a description of how the TOE uses public keys for the update verification mechanism in the Operational Environment. The evaluator ensures the TSS contains details on the protection and maintenance of the TOE update credentials.

Section 7.3.4 of [ST] (“FPT_TUD_EXT.1: Trusted Update”) states NetApp code signing ensures that TOE images installed through non-disruptive image updates or automated non-disruptive image updates are authentically produced by NetApp and have not been tampered with. The NetApp updates are cryptographically signed using a RSA Digital Signature algorithm with a key size of 3072-bits with a SHA-384 signature. The TOE will verify the signature before installing the update and reject any update with an invalid signature. The private key used for code signing is stored in a limited access hardware security module at NetApp. If the TOE’s public keys are tampered with, then an update will fail.

If the Operational Environment performs the signature verification, then the evaluator shall examine the TSS to ensure it describes, for each platform identified in the ST, the interface(s) used by the TOE to invoke this cryptographic functionality.

All cryptographic operations, including signature verification, are performed by the TOE.

2.3.5.2 Guidance Activities

The evaluator ensures that the operational guidance describes how the TOE obtains vendor updates to the TOE; the processing associated with verifying the digital signature of the updates (as defined in FCS_COP.1(a)); and the actions that take place for successful and unsuccessful cases.

Both [UEG] and [URDG] describe how the administrator obtains TOE updates from the vendor.

Section “Configuring NetApp hardware-based encryption > Configuring onboard key management > Enabling onboard key management in ONTAP 9.6 and later” of [NEPG] states system image updates will use the NetApp RSA-3072 code signing certificate together with SHA-384 code signed digests to check the image integrity instead. The upgrade command will verify that the image contents have not been altered or corrupted by checking various digital signatures. The image update process will proceed to the next step if validation succeeds; otherwise, the image update will fail.

2.3.5.3 KMD Activities

There are no KMD evaluation activities for this SFR.

2.3.5.4 Test Activities

The evaluators shall perform the following tests (if the TOE supports multiple signatures, each using a different hash algorithm, then the evaluator performs tests for different combinations of authentic and unauthentic digital signatures and hashes, as well as for digital signature alone):

Test 1: The evaluator performs the version verification activity to determine the current version of the TOE. After the update tests described in the following tests, the evaluator performs this activity again to verify that the version correctly corresponds to that of the update.

The evaluator verified the version of the TOE then initiated a TOE update. The evaluator confirmed that the TOE successfully accepted the update and confirmed that the version number had incremented to the expected number.

Test 2: The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that an update successfully installs on the TOE. The evaluator shall perform a subset of other evaluation activity tests to demonstrate that the update functions as expected.

After updating the TOE the evaluator generated cryptographic keys to and verified that the TOE functioned as expected.

3 Security Assurance Requirements

3.1 ASE: Security Targeted Evaluation

An evaluation activity is defined here for evaluation of Exact Conformance claims against a cPP in a Security Target. Other aspects of ASE remain as defined in the CEM.

3.1.1 Conformance Claims (ASE_CCL.1)

The table below indicates the actions to be taken for particular ASE_CCL.1 elements in order to determine exact conformance with a cPP.

ASE_CCL.1 element	Evaluation Action
ASE_CCL.1.8C	The evaluator shall check that the statements of security problem definition in the PP and ST are identical.
ASE_CCL.1.9C	The evaluator shall check that the statements of security objectives in the PP and ST are identical.
ASE_CCL.1.10C	The evaluator shall check that the statements of security requirements in the ST include all the mandatory SFRs in the cPP, and all of the selection-based SFRs that are entailed by selections made in other SFRs (including any SFR iterations added in the ST). The evaluator shall check that if any other SFRs are present in the ST (apart from iterations of SFRs in the cPP) then these are taken only from the list of optional SFRs specified in the cPP (the cPP will not necessarily include optional SFRs, but may do so). If optional SFRs from the cPP are included in the ST then the evaluator shall check that any selection-based SFRs entailed by the optional SFRs adopted are also included in the ST.

3.1.1.1 ASE_CCL.1.8C

Section 4 of [ST] includes by reference the security problem definition from [CPP_FDE_AA_V2.0E]. As such, the security problem definition in [ST] is identical to the statement of security problem definition specified in [CPP_FDE_AA_V2.0E].

3.1.1.2 ASE_CCL.1.9C

Section 5 of [ST] includes by reference the statement of security objectives from [CPP_FDE_AA_V2.0E]. As such, the statement of security objectives in [ST] is identical to the statement of security objectives specified in [CPP_FDE_AA_V2.0E].

3.1.1.3 ASE_CCL.1.10C

Section 6 of [ST] indicates that security functional requirements included in the ST are drawn from [CPP_FDE_AA_V2.0E].

The evaluator examined the requirements specified in Section 6 of [ST] and confirmed the following:

- All mandatory functional requirements from [CPP_FDE_AA_V2.0E] are included in [ST] through reproduction
- All selection-based requirements from [CPP_FDE_AA_V2.0E] indicated by selections made in the mandatory requirements are included in [ST] through reproduction

- One optional requirement from [CPP_FDE_AA_V2.0E] is included in [ST] through reproduction. This requirement does not entail any further selection-based requirements.
- No requirements not specified in [CPP_FDE_AA_V2.0E] have been specified in [ST].

3.2 Development (ADV)

3.2.1 Basic Functional Specification (ADV_FSP.1)

The EAs for this assurance component focus on understanding the interfaces (e.g., application programming interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Section 2 (*Evaluation Activities for SFRs*), and in EAs for AGD, ATE and AVA SARs in other parts of Section 5.

The EAs presented in this section address the CEM work units ADV_FSP.1- 1, ADV_FSP.1-2, ADV_FSP.1-3, and ADV_FSP.1-5.

The EAs are reworded for clarity and interpret the CEM work units such that they will result in more objective and repeatable actions by the evaluator. The EAs in this SD are intended to ensure the evaluators are consistently performing equivalent actions.

The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any required supplementary information required by the cPP: no additional “functional specification” documentation is necessary to satisfy the EAs. The interfaces that need to be evaluated are also identified by reference to the EAs listed for each SFR, and are expected to be identified in the context of the Security Target, AGD documentation, and any required supplementary information defined in the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the EAs for each SFR also means that the tracing required in ADV_FSP.1.2D (work units ADV_FSP.1-4, ADV_FSP.1-6 and ADV_FSP.1-7) is treated as implicit and no separate mapping information is required for this element.

3.2.1.1 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). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. 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.

The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance.

Through review of [ST] and [CMPR], the evaluation team identified that the CLI is the only security relevant TSFI. In particular, the following CLI commands are specifically identified in [ST] and were exercised during testing:

- system node halt

- system node image update
- security key-manager onboard disable
- security key-manager onboard enable
- storage encryption disk sanitize
- storage encryption disk destroy
- security key-manager onboard update-passphrase
- cluster image update
- cluster image show
- version
- volume encryption rekey.

The evaluation team determined the interface documentation described the purpose and method of use for each TSFI identified as being security relevant, sufficient to enable each of the evaluation activities to be completed satisfactorily. The evaluation team's results from performing the evaluation activities are documented in Section 2 of this AAR.

3.2.1.2 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.

The evaluation team determined the interface documentation identified and described the parameters for each TSFI identified as being security relevant, sufficient to enable each of the evaluation activities to be completed satisfactorily. The evaluation team's results from performing the evaluation activities are documented in Section 2 of this AAR.

3.2.1.3 Evaluation Activity

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2 (*Evaluation Activities for SFRs*), including the EAs associated with testing of the interfaces.

It should be noted that there may be some SFRs that do not have an interface that is explicitly "mapped" to invoke the desired functionality. For example, generating a random bit string, destroying a cryptographic key that is no longer needed, or the TSF failing to a secure state, are capabilities that may be specified in SFRs, but are not invoked by an interface.

However, if the evaluator is unable to perform some other required EA because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a 'fail'.

The evaluation team examined the interface documentation and was able to map interfaces to SFRs, sufficient to enable each of the evaluation activities to be completed satisfactorily. The evaluation team's results from performing the evaluation activities are documented in Section 2 of this AAR.

3.3 Guidance Documents (AGD)

It is not necessary for a TOE to provide separate documentation to meet the individual requirements of AGD_OPE and AGD_PRE. Although the Evaluation Activities in this section are described under the traditionally separate AGD families, the mapping between real TOE documents and AGD_OPE and AGD_PRE requirements may be many-to-many, as long as all requirements are met in documentation that is delivered to administrators and users (as appropriate) as part of the TOE.

3.3.1 Operational User Guidance (AGD_OPE.1)

Specific requirements and checks on the user guidance documentation are identified (where relevant) in the individual Evaluation Activities for each SFR, and for some other SARs (e.g. ALC_CMC.1).

3.3.1.1 Evaluation Activity

The evaluator shall check the requirements below are met by the operational guidance.

Operational guidance documentation shall be 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.

The guidance documentation specific to the evaluation is identified in Section 3 of [ST]. The vendor provides all documents via their product and support web site at <http://docs.netapp.com/ontap-9/index.jsp>. Administrators are able to view the documentation on-line and to download the documentation in .PDF format.

Operational guidance must be provided for every Operational Environment that the TOE supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target. This may be contained all in one document.

The ST identifies the TOE is supported on a range of Fibre Attached Storage (FAS) and All Flash FAS (AFF) network storage controllers, but the same TOE software runs on each controller. As such, the guidance documentation adequately addresses all platforms claimed for the TOE in the ST.

The contents of the operational guidance will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in sections 2, 3, and 4 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

- The operational guidance shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.
- The TOE will likely contain security functionality that does not fall under the scope of evaluation under this cPP. The operational guidance shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

The cryptographic modules included in the TOE do not require any configuration, and the TOE does not incorporate any unevaluated cryptographic module. The guidance documentation identifies and describes commands used to place the TOE into its Common Criteria mode of operation and the need to enable the Onboard Key Manager when using the product in its evaluated configuration.

3.3.2 Preparative Procedures (AGD_PRE.1)

As for the operational guidance, specific requirements and checks on the preparative procedures are identified (where relevant) in the individual Evaluation Activities for each SFR.

3.3.2.1 Evaluation Activity

The evaluator shall check the requirements below are met by the preparative procedures.

The contents of the preparative procedures will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in section 2 above.

Preparative procedures shall be 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.

The guidance documentation specific to the evaluation is identified in Section 3 of [ST]. The vendor provides all documents via their product and support web site at <http://docs.netapp.com/ontap-9/index.jsp>. Administrators are able to view the documentation on-line and to download the documentation in .PDF format.

The contents of the preparative procedures will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in section 2 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

Preparative procedures must include a description of how the 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 itself).

The vendor provides guides for installing each of the storage controllers identified in [ST] as part of the evaluated configuration at <https://docs.netapp.com/platstor/index.jsp>. The guides also provide preparative procedures for configuring ONTAP. Guidance covers how the administrator verifies the operational environment can fulfil its role to support the security functionality.

Preparative procedures must be provided for every Operational Environment that the TOE supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target. This may be contained all in one document.

The ST identifies the TOE is supported on a range of Fibre Attached Storage (FAS) and All Flash FAS (AFF) network storage controllers, but the same TOE software runs on each controller. As such, the guidance documentation adequately addresses all platforms claimed for the TOE in the ST.

The preparative procedures must include

- instructions to successfully install the TSF in each Operational Environment; and
- instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and
- instructions to provide a protected administrator capability.

The vendor provides guides for installing each of the storage controllers identified in [ST] as part of the evaluated configuration at <https://docs.netapp.com/platstor/index.jsp>. The guides also provide preparative procedures for configuring ONTAP.

The guidance provided in [NEPG] and [SAR] includes instructions for managing the security of the TOE as a product and as a component of the operational environment, while [AARPG] provides instructions for providing a protected administrator capability.

3.4 Life-Cycle Support (ALC)

3.4.1 Labeling of the TOE (ALC_CMC.1)

When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.

3.4.2 TOE CM Coverage (ALC_CMS.1)

When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM.

3.5 Tests (ATE)

Testing is performed to confirm the functionality described in the TSS as well as the operational guidance documentation. The focus of the testing is to confirm that the requirements specified in the SFRs are being met.

The evaluator should consult Appendix B FDE Equivalency Considerations when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

The SFR-related Evaluation Activities in the SD identify the specific testing activities necessary to verify compliance with the SFRs. The tests identified in these other Evaluation Activities constitute a sufficient set of tests for the purposes of meeting ATE_IND.1.2E. It is important to note that while the Evaluation Activities identify the testing that is necessary to be performed, the evaluator is responsible for ensuring that the interfaces are adequately tested for the security functionality specified for each SFR.

Evaluation Activity: The evaluator shall examine the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

The evaluator received the TOE from the vendor and confirmed that it conforms with the hardware, configuration and firmware described in the ST.

Evaluation Activity: The evaluator shall examine the TOE to determine that it has been installed properly and is in a known state

The evaluator received the TOE from the vendor and powered it on. The evaluator confirmed it presented no errors and entered a running state. The evaluator performed a version and model verification activity prior to testing.

Evaluation Activity: The evaluator shall prepare a test plan that covers all of the testing actions for ATE_IND.1 in the CEM and in the SFR-related Evaluation Activities. While it is not necessary to have one test case per test listed in an Evaluation Activity, the evaluator must show in the test plan that each applicable testing requirement in the SFR-related Evaluation Activities is covered.

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

The test plan describes the composition and configuration of each platform to be tested, and any setup actions that are necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of any cryptographic engine to be used (e.g. for cryptographic protocols being evaluated).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives, and the expected results.

The test report (which could just be an updated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure, so that a fix was then installed and then a successful re-run of the test was carried out, then the report would show a “fail” result followed by a “pass” result (and the supporting details), and not just the “pass” result.

The evaluator prepared a test plan prior to testing outlining the required test activities to be performed. Throughout testing the test plan was updated with results to eventually become the test report.

The test plan laid out a subset of all instances of the TOE in the evaluation to be tested. Similar instances of the TOE were grouped together based on hardware. Each instance of the TOE did not have any variations with software dependencies, software binaries, libraries used, management interfaces or functional differences so hardware only needed to be considered. When an instance or instances of the TOE were not tested justification was provided in the test report via equivalency rationale.

The test report lists each instance of the TOE tested for each SFR and details it as defined in the ST.

The test plan established and set of procedures to follow with steps and configuration necessary to achieve the expected result.

The test report describes in detail the activities the evaluator performed along with actual results in the form of evidence to accomplish each test. Each test account is accompanied by a test result in the form of ‘pass’ or ‘fail’. The test report also establishes an overall result for the cumulative test activities stated by a ‘pass’ or ‘fail’.

Independent testing took place at Leidos CCTL facilities in Columbia, Maryland, from 17 September through 21 December 2020.

The evaluation team established a test configuration comprising:

- TOE components:
 - ONTAP 9.7P6 installed on following NetApp Storage Encryption appliances:
 - FAS 2650
 - FAS 8300
 - FAS 9000
 - AFF A800
- Test environment components:
 - Kali Linux Server.

Subsequent to the conclusion of functional testing, NetApp issued the following patches: 9.7P7; 9.7P8; 9.7P9; 9.7P10; 9.7P11; 9.7P12; and 9.7P13. Of these patches, 9.7P8, 9.7P9, and 9.7P13 address published vulnerabilities (CVE-2021-26988, CVE-2021-26989, and CVE-2021-26994 respectively). The evaluation team reviewed the list of changes for each patch release and flagged no changes as evaluation relevant, as they were non-evaluation relevant in relation to the security functional requirement claims. In these incremental maintenance releases, the vendor did not introduce any new product functionality that related to any of the claimed security functional requirements and did not affect any of the identified TSFI (in terms of CLI commands called out in the ST as SFR-relevant, or exercised during testing). This process is similar to what would be followed in preparing an Impact Analysis Report as part of an Assurance Maintenance activity. As such, the evaluated version of the TOE (9.7P13) can be considered equivalent to the version on which evaluation testing was performed (9.7P6).

3.6 Vulnerability Assessment (AVA)

3.6.1 Vulnerability Survey (AVA_VAN.1)

While vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator follows a set of well-defined activities, and documents their findings so others can follow their arguments and come to the same conclusions as the evaluator. While this does not guarantee that different evaluation facilities will identify exactly the same type of vulnerabilities or come to exactly the same conclusions, the approach defines the minimum level of analysis and the scope of that analysis, and provides Certification Bodies a measure of assurance that the minimum level of analysis is being performed by the evaluation facilities.

3.6.1.1 Evaluation Activity (Documentation)

The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components apply to all systems claimed in the ST, and should identify at a minimum the processors used by the TOE. Software components include any libraries used by the TOE, such as cryptographic libraries. 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 hypotheses during their analysis.

The evaluator shall examine the documentation outlined below provided by the vendor 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 TOE is software-only, consisting of the ONTAP 9.7P13 software. Table 7 of [ST] (“Third Party Components”) lists the following third-party libraries included in the TOE:

- OpenSSL 1.0.2s-fips
- Intel Intelligent Storage Acceleration Crypto Library v2.22.

In addition to the activities specified by the CEM in accordance with Table 2 above, the evaluator shall perform the following activities.

3.6.1.2 Evaluation Activity

The evaluator formulates hypotheses in accordance with process defined in Appendix A.1. The evaluator documents 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.

The evaluation team performed a search of the following public sources of vulnerability information as selected by the iTC:

- Common Vulnerabilities and Exposures: <http://cve.mitre.org/cve/>
- National Vulnerability Database: <https://nvd.nist.gov/>
- US-CERT: <http://www.kb.cert.org/vuls/html/search.>

The list of sources above was searched with the following terms:

- General (for all):
 - Product name
 - Underlying components (e.g., OS, software libraries (crypto libraries), chipsets)
 - Drive encryption, disk encryption
 - Key destruction/sanitization
- AA:
 - Underlying components (e.g., smart card libraries)
 - Opal management software, SED management software
 - Password caching
- For Software FDE (AA or EE):
 - Key caching.

These search criteria specified by the iTC were applied as follows:

- Product name—the evaluation team searched on the following terms:
 - “netapp”/ “netapp ontap”
 - “netapp fas”
 - “netapp aff”
 - “network storage encryption”
- Underlying components—the evaluation team searched on the following terms:
 - “ontap 9.7p13”
 - “openssl 1.0.2s”
 - “intel isa-l crypto library 2.22”
 - “intel storage acceleration library”

- “x440_phm2800mcto”
- “x440_tpm3v800amd”
- “x4010s172b1t9nte”
- “x417_hcbfe900a10”
- “x417_sltn900a10”
- Search terms specified in [SD]—the evaluation team searched on the following terms:
 - “drive encryption”
 - “disk encryption”
 - “key destruction”
 - “key sanitization”
 - “opal management software”/ “opal”
 - “sed management software”/ “self encrypting drive”
 - “password caching”.

Searches of the public sources of vulnerability information listed above were conducted on 31 August 2021. The evaluation team determined that no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.