Cohesity DataPlatform & DataProtect Version 6.0.1 Security Target

Version 1.0
15 June 2019

Prepared for:

COHESITY

300 Park Ave
Suite 800
San Jose, CA 95110

Prepared By:

leidos

Accredited Testing and Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, MD 21046

TABLE OF CONTENTS

LIST OF TABLES

1. Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE identification, ST and TOE conformance claims, ST conventions, glossary and list of abbreviations.

The ST contains the following additional sections:

- TOE Description (Section 2)—provides an overview of the TOE and describes the physical and logical boundaries of the TOE
- Security Problem Definition (Section 3)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 4)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 5)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 6)—describes the security functions of the TOE and how they satisfy the SFRs
- Rationale (Section 7)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, TOE and CC Identification

ST Title – Cohesity DataPlatform and DataProtect Security Target

ST Version – Version 1.0

ST Date – 15 June 2019

TOE Identification – Cohesity 6.0.1

TOE Developer – Cohesity

Evaluation Sponsor – Cohesity

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1 Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1 Revision 5, April 2017.
    - Part 3 Conformant

This ST and the TOE it describes are conformant to the following package:

- EAL2 Augmented (ALC_FLR.1).

1.3 Conventions

The following conventions are used in this document:

- Security Functional Requirements—Part 1 of the CC defines the approved set of operations that may be applied to functional requirements: iteration; assignment; selection; and refinement.

  - Iteration—allows a component to be used more than once with varying operations. In this ST, iteration is identified with a number in parentheses following the base component identifier. For example, iterations of FCS_COP.1 are identified in a manner similar to FCS_COP.1(1) (for the component) and FCS_COP.1.1(1) (for the elements).
  - Assignment—allows the specification of an identified parameter. Assignments are indicated using bold text and are enclosed by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection—allows the specification of one or more elements from a list. Selections are indicated using bold italics and are enclosed by brackets (e.g., [***selection***]).
  - Refinement—allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). When the refined text is part of an assignment (and would therefore not be visually distinct from the typical assignment text), the SFR is preceded with 'Refinement:' to indicate the deviation from the SFR's original definition.
  - Extended Requirements are allowed to create requirements should the Common Criteria not offer suitable requirements to meet the ST needs. To ensure these requirements are explicitly identified, the ending "_EXT" is appended to the newly created short name and the component.
- Other sections of the ST—other sections of the ST use bolding and/or different fonts (such as `Courier`) to highlight text of special interest, such as captions, commands, or filenames specific to the TOE.
- The requirements are copied verbatim, except for some changes to required identifiers to match the iteration convention of this document from CC Part 2 and for changes of British spelling to American where applicable.

1.4 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this ST:

ACL	Access Control List
AD	Active Directory
AWS	Amazon Web Services
CC	Common Criteria
CSP	Cloud Service Provider
DEK	Data Encryption Key
DOM	Disk on Module
EAL	Evaluation Assurance Level
HDD	Hard Disk Drive
IO	Input/Output
KEK	Key Encryption Key
KMIP	Key Management Interoperability Protocol
NAS	Network-Attached Storage
NFS	Network File System
OS	Operating System
POSIX	Portable Operating System Interface
RAM	Random Access Memory
REK	Root Encryption Key
QoS	Quality of Service
SAR	Security Assurance Requirement
SATA	Serial AT Attachment
SCVMM	System Center Virtual Machine Manager
SFP	Security Function Policy
SFR	Security Functional Requirement

SMB	Server Message Block
SQL	Structured Query Language
SSD	Solid-State Drive
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Function
VM	Virtual Machine

2. TOE Description

2.1 Overview

The TOE is Cohesity DataPlatform and DataProtect (or collectively simply as "Cohesity"), a software suite that is used to hyperconverge secondary storage workloads (i.e., enterprise data backups) into a single managed backup solution, which may be distributed across multiple distributed appliances. The intent of this product is to simplify the infrastructure and resources used to administer data backup and recovery functions across an enterprise. The TOE natively supports backups for various virtual machines, databases, and network-attached storage (NAS) devices. The TOE also interfaces natively with various cloud service providers for long-term archival and retention of backup data. Backup data stored by the TOE is protected against unauthorized modification and disclosure using symmetric encryption. The TOE provides a role-based access control policy for accessing stored data and administrative functionality.

Cohesity is designed to eliminate secondary storage silos by converging all secondary storage and associated data services on one unified solution – including backups, cloud gateway, files, objects, test/dev copies, and data analytics. Cohesity is a software-defined solution that spans from the edge, to the datacenter, and the cloud. With Cohesity, enterprises can:

- Simplify data protection infrastructure by converging legacy backup silos
- Consolidate file and object services
- Build a multicloud data fabric with native cloud integration for archival, tiering and replication
- Accelerate test/dev with copy data management
- Gain visibility into their dark data with in-place analytics
- Reduce total cost of ownership for secondary storage by 50% or more

This capability is securely managed through user interfaces that provide granular control over authentication, authorization, and communications protocols.

The TSF includes the following security functions:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Self-Protection
- Resource Utilization
- Trusted Communications

2.2 Architecture

The TOE comprises two main components:

Cohesity DataPlatform

Cohesity DataPlatform is a software-defined, web-scale platform used consolidate secondary data. Key features of DataPlatform include:

- Distributed architecture: Always-on platform that scales linearly by simply adding nodes, eliminating data migrations and forklift upgrades.
- File and object services: NFS, SMB and S3 interfaces to consolidate files and objects.
- Global space efficiency: Global variable-length dedupe, compression, and erasure coding.
- Cloud integration: Native cloud integration with AWS, Google Cloud, and Microsoft Azure.
- Copy data management: Zero-cost clones to quickly provision test/dev copies.

Cohesity DataProtect

Cohesity DataProtect is a converged backup and recovery solution that runs on DataPlatform and provides:

- Simple data protection: Replace multiple data protection silos (target storage, media servers, master servers, cloud gateways) with a single converged solution.
- Sub-5 minute RPOs and instantaneous RTOs: Fast RPOs and RTOs by keeping each backup as a zero cost, instantly available snapshot.
- Tight integration with VMware, physical Windows and Linux, MS SQL and Oracle databases.
- Primary storage integrations: Integration with Pure Storage and NAS filers for automated data protection.

Administration of the TOE is performed using SSH and Web GUI components that are part of DataProtect. When operating in a cluster (multiple distributed nodes), a single node is designated as the 'primary' node, and this node is what is used to administer the entire cluster.

The TOE is a software application that that can be scaled across physically distributed nodes. It can run on any of the following physical/virtual hardware components:

- Hyperconverged nodes (x86 hardware appliances sold by Cohesity)
- Cloud (supported on Azure, AWS, and Google clouds)
- Virtual (supported on VMware)
- Third-Party dedicated hardware (currently supported on HPE DL380 and Cisco UCS platforms)
- General-Purpose (supported on commercial off-the-shelf hardware running CentOS 7.x on x86 architecture)

The following figure shows an example deployment of the TOE in its operational environment, with TOE components highlighted in green:

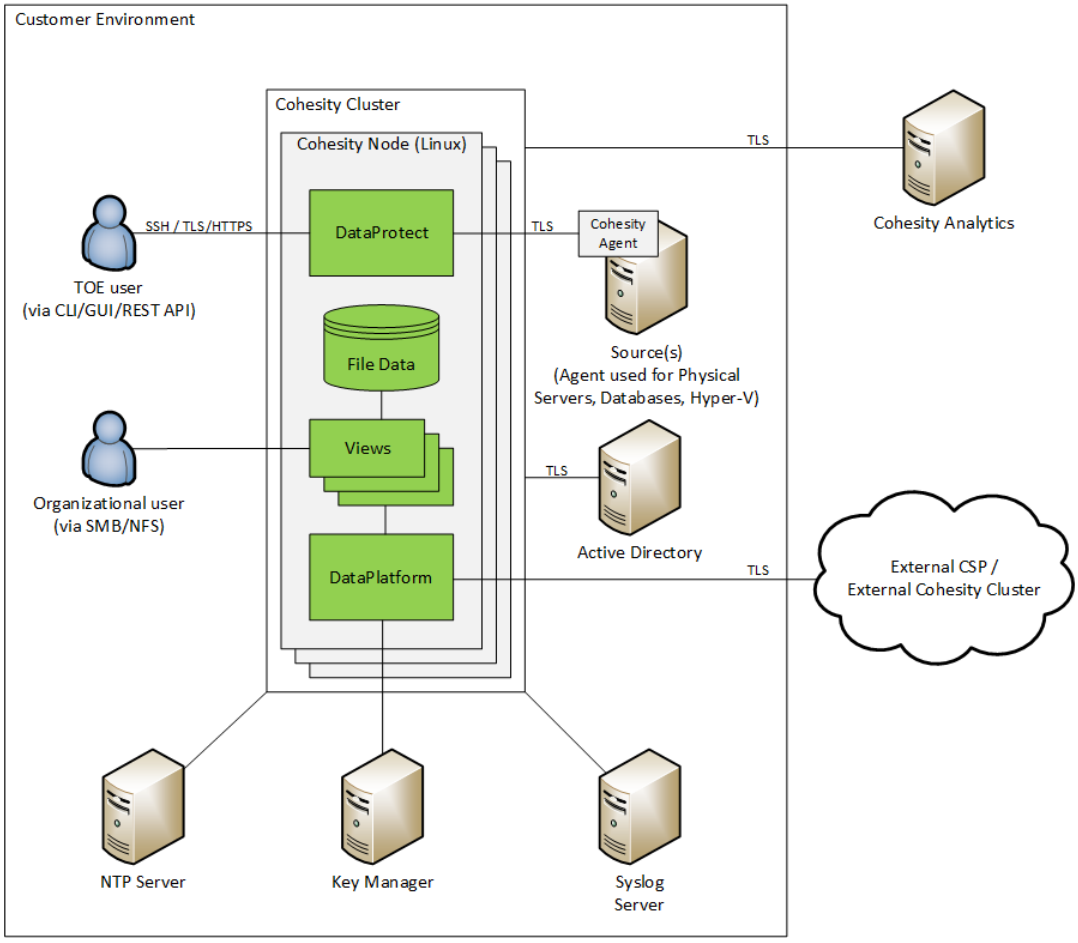

Figure 1: Example TOE Deployment

The components in the figure above are responsible for the following functions:

- TOE components:

    - DataPlatform: Provides the interface to ingest backup data and ensure its availability through replication, indexing, and network accessibility
        - Data stored on DataPlatform as a View can be accessed externally via SMB/NFS
        - Other file data is only accessible through the TOE's management interfaces
    - DataProtect: Manages confidentiality of backup data as well as backup, recovery, and archival operations
- Environmental components:
    - Cohesity Node: the physical or virtual device on which the TOE software is installed
    - Active Directory: user authentication
    - Syslog Server: audit data storage
    - NTP Server: time services
    - KMIP Key Manager: data encryption key (DEK) protection
    - Sources: originators of data backed up by the TOE
    - Cohesity Agent: software installed on some Sources to provide an interface for the TOE to acquire the data stored on that Source
    - Cloud Service Provider: offsite cold storage/replication for backup data
    - External Cohesity Cluster: second deployment of Cohesity for offsite cold storage/replication of backup data
    - Cohesity Analytics: Cohesity-run service for remote telemetry and support automation

2.3 Physical Boundaries

2.3.1 Physical TOE Components

The TOE boundary includes the DataPlatform and DataProtect software. This software is installed on a node, which can be any of the physical and/or virtual components listed in section 2.2 above. As the TOE can be deployed as a distributed application, multiple instances of the software may be installed on multiple nodes. A combination of nodes is referred to as a cluster. A node may contain multiple individual storage disks. In the evaluated configuration, both DataPlatform and DataProtect is present on each node.

If an instance of the TOE is deployed on a first-party hyperconverged node or supported third-party hardware, the only system requirement is that a supported hardware model is used. The dedicated hardware used for the TOE is a 2U4N system, which contains four separate nodes within one single 2U chassis, or block.

If an instance of the TOE is deployed on a cloud platform, the only system requirement is that Microsoft Azure, AWS, or Google Cloud is the cloud service provider (CSP) that is used.

If an instance of the TOE is deployed on a general-purpose computer, that computer must be running CentOS 7.x and have a 64-bit x86 processor architecture. A representative system configuration is provided below—this configuration is identical to the C2105 hyperconverged node sold by Cohesity:

- CPU: Intel Xeon E5-2603 v3
- SATADOM: 8GB
- Memory: 4x16GB
- SSD (for the TOE and its configuration/audit data): 800MB
- HDD (for data backups): 3x2TB
- Network connectivity: 2x1GbE; 2x10GbE

While the actual hardware on which the TOE runs is not part of the TOE, the backup data stored on this hardware is considered to be TSF data and therefore protection of it falls within the scope of the TOE.

2.3.2 Operational Environment Components

The TOE requires the following components in its operational environment to support the enforcement of its security functions:

- Physical/logical storage capable of having backup data ingested by the TSF (also known as Sources) – any of the following are supported:
    - Virtual servers: VMware, Hyper-V, AHV, RHEV

  - Physical servers: Windows, Linux, AIX
  - Applications: Microsoft Exchange
  - Databases: Microsoft SQL Server, Oracle
  - Storage integrations: Pure FlashArray
  - Network Attached Storage: NetApp cluster, Isilon cluster, Pure FlashBlade, generic NFS, SMB
- Cohesity Agent: installed on the following Sources to provide an interface to transfer data from that Source to the TOE:
  - Virtual servers: Hyper-V (or SCVMM server containing multiple Hyper-V VMs), VMware, AHV, RHEV
  - Physical Servers: all
  - Databases: all
- Web browser with HTML5 support (for administration)
- SSH client (for administration)
- KMIP-compliant Key Manager (for management and secure storage of DEKs)

The TOE may or may not make use of the following environmental components, depending on how it is configured:

- VMware vSphere (5.5 or higher) or Microsoft Hyper-V server (2012 or 2016): required if Cohesity Virtual Edition is used
- Active Directory: optional for authentication and authorization
- DNS Server: optional for use of name services
- NTP Server: optional for use of network time
- Syslog Server: optional for remote storage of audit data
- Cloud Service Providers (Microsoft Azure, AWS, Google Cloud, Oracle, or any S3 compatible private cloud or on premise object storage): optional for use of cloud backup
- External Cohesity Cluster: optional for replication of stored data
- Cohesity Analytics: optional service run by Cohesity for remote telemetry and support automation

2.3.3 Excluded from the TOE

The following capabilities are part of the Cohesity product but are excluded from the evaluated configuration because they prevent the TSF from being fully implemented:

- DataPlatform standalone configuration: DataPlatform does not require DataProtect to be present to function, but DataProtect is required for the evaluated configuration in order to provide the security functionality claimed in this ST.
- Internal key manager: Cohesity provides its own key management capability for storage of DEKs, but access to the KEKs used to encrypt the DEKs is controlled through logical access and not a cryptographically-protected REK. Therefore, it is necessary to use a third-party key manager to ensure an appropriate degree of security for the protected data.
- OS root user: Cohesity provides a CLI application, web-based GUI, and REST API to perform security-relevant management functions. By default, Cohesity is installed on a general-purpose operating system by a root-level user account. The root user can perform certain debug-related activities against Cohesity that are not available through any of the TSF-relevant management interfaces. However, these activities are entirely debug-related in nature and will not be invoked when the TOE is operating properly in its evaluated configuration.

2.4 Logical Boundaries

This section summarizes the security functions provided by the TOE.

2.4.1 Security Audit

The TOE generates audits of user activity and security-relevant events that occur on the cluster, such as job failures or disk storage alerts. Audit data is distributed amongst the various nodes in the cluster to ensure that it is replicated. This stored data cannot be modified or deleted by any user or administrator. In the evaluated configuration, the various nodes are configured to send their audit data to a remote syslog server.

2.4.2 Cryptographic Support

The TOE supports TLS (independently and as part of HTTPS) and SSH to perform trusted communications. The TOE also uses symmetric cryptography to encrypt backup data at rest. Long-term storage of symmetric keys used to encrypt data at rest is the responsibility of the Operational Environment. Certificate data and short-term keys, such as keys established to enable TLS communications, are zeroized when no longer in use.

The cryptographic functions used to secure data at rest and in transit are NIST-approved algorithm implementations.

2.4.3 User Data Protection

The TOE provides mechanisms for acquiring data from the operational environment for backup purposes. Data can be acquired from various sources such as physical servers, virtual servers, databases, storage arrays, and NAS. While the data is stored internally to the TOE, it may also be configured to be viewable as a SMB or NFS storage device. Access controls, both within the TOE's management interfaces and on any SMB/NFS shared data, are used to define the data that can be accessed by TOE and organizational users. Data at rest is protected using AES-256 encryption to prevent unauthorized access.

The basic functionality for the TOE"s data protection function is to back up data from environmental sources, store it within the TOE, and use it to perform restore operations as needed. Data can be set as immutable so that an accurate reversion of working data (such as in the case of a ransomware attack) can be restored to the affected environmental systems. Policies define the data that is acquired as well as the frequency of the backup operations, as well as whether full or incremental backups are performed. Data stored on the TOE may also be sent to a remote Cloud Service Provider or a remote Cohesity Cluster (i.e. a second deployment of Cohesity) for replication or cold storage (archival) purposes.

The TOE includes an Analytics Workbench application that provides a MapReduce framework for analysis and reporting on data stored within the TOE. This can be used to search for significant data, such as specific text strings/patterns, strings that may be indicative of cleartext passwords, or uncompressed video. Filters can be applied to search parameters so that, for example, data stored in a certain location or that is of a certain age can be excluded from the search. Additional custom searches can be defined by users.

2.4.4 Identification & Authentication

The TOE requires user authentication prior to accessing any of its security functionality. This is done using either username/password (for web GUI and SSH), public key authentication (for SSH), or token (for REST API). Username/password data for the web GUI can be defined on the TOE or the TOE can connect to an environmental Active Directory server to perform authentication; the SSH interface uses either AD credentials or locally-defined credentials, depending on the functionality that the SSH interface is being used to perform.

The TOE includes self-signed certificates for its server functionality that are implicitly trusted by the TSF. These can be replaced with user-supplied certificates that are subject to validation, including revocation checking. The TSF also performs certificate validation on server certificates presented to it as part of establishing outbound trusted channels with remote servers such as Active Directory.

2.4.5 Security Management

The TSF provides three management interfaces: a web GUI (also known as Cohesity Dashboard), a CLI, and a REST API. The set of management functions available for use to interact with the TSF depends on the interface used to access the TOE.

The TOE has five defined management roles by default. These roles grant differing degrees of access to the management functionality of the TOE. Additional roles can be defined as needed. Individual users may be restricted in the set of objects that they can perform their assigned management privileges against.

2.4.6 Protection of the TSF

The TOE is deployed as a distributed system, which allows for redundant data storage. Redundancy is achieved either through the use of replication factors (i.e. duplicate copies of data stored on different disks/nodes) or erasure coding.

The TOE performs a series of self-tests when a node is powered on. This includes validation of the cryptographic functionality, which is performed by the Cohesity OpenSSL FIPS Object Module (CMVP certificate #2676). It also

includes various boot checks of a node, including correct operation of OS/service boot, storage disks, and network availability. If a node experiences a failure, it will enter a degraded mode of operation and attempt to reboot. The degraded status will be reported to administrators in the management interface.

2.4.7 Resource Utilization

The TOE provides methods for administrators to configure replication of data across multiple nodes or Cohesity clusters.

The TSF also includes a function called 'intelligent data placement' which automatically places data on appropriate nodes based on QoS and IO profiles. This ensures that access to data backup and recovery functions is maintained in the event of the failure/unavailability of individual nodes/disks or in a traffic-constrained environment.

2.4.8 Trusted Path/Channels

The TOE uses its FIPS-validated cryptographic module to provide secure communications between itself and remote IT entities/administrators. Specifically, the following interfaces use the following trusted channels/paths:

- TOE to AD trusted channel – LDAP over TLS
- TOE to remote CSP trusted channel – TLS/HTTPS
- TOE to Secondary Cohesity Cluster trusted channel – TLS/HTTPS
- TOE to Source trusted channel – TLS/HTTPS
- TOE to Cohesity Analytics – TLS
- Remote Source to TOE trusted channel – TLS/HTTPS
- Remote CLI to TOE trusted path – SSH
- Remote GUI to TOE trusted path – TLS/HTTPS
- Remote REST API to TOE trusted path – TLS/HTTPS

2.5 Capabilities Provided by the Operational Environment

The TOE relies on the operational environment for the following components and capabilities:

- Switches and DNS for load balancing and for intra-cluster communication
- Active Directory for user/administrator authentication
- CSP(s) for offsite backup of stored data
- Syslog for centralized offsite audit storage
- NTP for managing system time across nodes
- KMIP key manager for protection of DEKs

2.6 TOE Documentation

This section identifies the guidance documentation included in the TOE.

- Cohesity User Guide, Version 6.0.1
- Cohesity Security Features

3. Security Problem Definition

This section defines the security problem to be addressed by the TOE, in terms of threats to be countered by the TOE or its operational environment, and assumptions about the intended operational environment of the TOE.

3.1 Assumptions

This section contains assumptions regarding the operational environment and the intended usage of the TOE.

A.COMPONENTS_RUNNING	It is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack or failure of one or more of the TOE components.
A.LIMITED	It is assumed that the hardware components that comprise the TOE are used only for the functionality provided by the TSF and that the TOE does not include any other general-purpose computing capabilities that present additional external interfaces to the TSF.
A.NETWORK	It is assumed that the nodes on which the TOE is deployed are connected to one another over a local network that is not subject to unauthorized surveillance.
A.PHYSICAL	It is assumed that the TOE is deployed in a location that is physically secured in its operational environment and not subject to any attacks on the physical interfaces of the TOE or the TOE hardware itself.
A.REGULAR_UPDATES	It is assumed that TOE software/firmware updates are applied on a regular schedule and/or within a reasonable period of time after they have been made available by the vendor.
A.SYSTEM_TIME	The TOE's operational environment is assumed to provide reliable system time for all nodes.
A.TRUSTED_ADMIN	It is assumed that any administrators of the TOE are trusted to be technically competent, non-malicious, and to follow operational and preparatory guidance as directed for the functions that they are authorized to perform.

3.2 Threats

This section identifies and describes the threats to be countered by the TOE and its operational environment.

T.DATA_DISCLOSURE	A malicious user or process may access backup data without authorization.
T.DATA_ERASURE	A malicious user or process may destroy backup data without authorization.
T.DATA_MODIFICATION	A malicious user or process may corrupt or otherwise modify backup data without authorization.
T.TSF_FAILURE	The TSF may be compromised by a malicious user or process or corrupted through general degradation such that it operates in an unknown state.
T.UNAUTH_ACCESS	A malicious individual may access restricted TSF functionality without authorization, either through obtaining valid credentials that they previously lacked or by escalating their privileges in excess of what they were assigned.
T.UNDETECTED_ACTIONS	A malicious or careless user may alter the behavior of the TSF to cause it to operate in an unknown state.
T.UNTRUSTED_COMMS	A malicious user or process may access security-relevant TSF data in transit through the use of unencrypted or poorly encrypted communications channels.

4. Security Objectives

This section identifies the security objectives for the TOE and its operational environment. The security objectives identify the responsibilities of the TOE and its environment in addressing the security problem defined in Section 3.

4.1 Security Objectives for the TOE

The following are the TOE security objectives:

O.AUDIT	The TSF generates audit records of security-relevant events.
O.AVAILABILITY	The TSF provides a physical and logical organization of backup data that ensures retention of the data in the event of a TOE component failure or outage.
O.I_AND_A	All TOE users are identified and authenticated by the TSF before access to security-relevant functionality is granted.
O.PROTECTED_BACKUPS	Backup data at rest is protected by the TSF from unauthorized disclosure, modification, and erasure.
O.PROTECTED_COMMS	The TSF protects communications between itself and external entities using trusted channel protocols and appropriate cryptographic functions.
O.SECURITY_MANAGEMENT	The TSF restricts the ability to perform security management functions on the TOE to authorized users having appropriate roles.
O.SELF_TESTS	The TSF will provide a mechanism to detect failure of its own functionality and provide notification of this to administrators.

4.2 Security Objectives for the Operational Environment

The following are the security objectives for the operational environment of the TOE.

OE.COMPONENTS_RUNNING	The TOE's operational environment will include measures to detect and respond to the unavailability of any TOE components.
OE.LIMITED	TOE administrators will not use the TOE hardware for operations other than its intended purpose.
OE.NETWORK	The TOE's operational environment will include a local area network that is used to connect distributed nodes in a manner that is not subject to unauthorized surveillance.
OE.PHYSICAL	The TOE's operational environment will be designed in such a manner that the TOE components will reside in a location that prevents untrusted physical access.
OE.REGULAR_UPDATES	TOE administrators will ensure that TSF updates are applied as needed.
OE.SYSTEM_TIME	The TOE's operational environment will provide reliable system time for all nodes.
OE.TRUSTED_ADMIN	Individuals chosen to manage the TOE will be vetted to ensure that they are not careless, not malicious, and will not disregard any security-relevant guidance for the setup and operation of the TOE.

5. IT Security Requirements

5.1 Extended Components Definition

5.1.1 Cryptographic Support (FCS)

5.1.1.1 HTTPS Protocol (FCS_HTTPS_EXT)

This family defines behavior for implementation of the HTTPS protocol. This family contains one component, FCS_HTTPS_EXT.1.

FCS_HTTPS_EXT.1, HTTPS Protocol, requires the TSF to implement HTTPS in accordance with a defined standard, and to exhibit certain behavior in the event of certificate validation failure.

Management: **FCS_HTTPS_EXT.1**

There are no management activities foreseen.

Audit: **FCS_HTTPS_EXT.1**

There are no auditable events foreseen.

FCS_HTTPS_EXT.1 – HTTPS Protocol

Hierarchical To: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FCS_TLS_EXT.1 TLS Client Protocol

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement the HTTPS protocol using TLS.

5.1.1.2 Random Bit Generation (FCS_RBG_EXT)

This family defines behavior for random number/bit generation. This family contains one component, FCS_RBG_EXT.1.

FCS_RBG_EXT.1, Random Bit Generation, requires the TSF to perform random bit generation in accordance with selected standards and seeded by an entropy source.

Management: **FCS_RBG_EXT.1**

There are no management activities foreseen.

Audit: **FCS_RBG_EXT.1**

There are no auditable events foreseen.

FCS_RBG_EXT.1 – Random Bit Generation

Hierarchical To: No other components

Dependencies: No dependencies

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [***selection: Hash_DRBG, HMAC_DRBG, CTR_DRBG (AES)***].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***selection: [assignment: number of software-based sources] software-based noise source, [assignment: number of hardware-based sources] hardware-based noise sources***] with a minimum of [**assignment: number of bits**] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

5.1.1.3 SSH Protocol (FCS_SSH_EXT)

This family defines behavior for implementation of the SSH protocol. This family contains one component, FCS_SSH_EXT.1.

FCS_SSH_EXT.1, SSH Protocol, requires the TSF to implement SSH in accordance with selected standards and to define the connection parameters it uses in the establishment of SSH communications.

Management: **FCS_SSH_EXT.1**

The following actions could be considered for the management functions in FMT:

a) Configuration of SSH connection parameters.

Audit: **FCS_SSH_EXT.1**

There are no auditable events foreseen.

FCS_SSH_EXT.1 – SSH Protocol

Hierarchical To: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FCS_RBG_EXT.1 Random Bit Generation

FPT_STM.1 Reliable Time Stamps

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that compiles with RFCs 4251, 4252, 4253, 4254, [***selection: 5647, 5656, 6187, 6668, no other RFCs***].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [***selection: public key-based, password-based***].

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**assignment: number of bytes**] bytes in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses only the following encryption algorithms: [***selection: aes128-cbc, aes256-cbc, aes128-ctr, aes-192-ctr, aes256-ctr, AED_AES_128_GCM, AEAD_AES_256_GCM***].

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses only [***selection: ssh-rsa, ecdsa-sha2-nistp256***] and [***selection: rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, no other public key algorithms***] as its public key algorithm(s).

FCS_SSH_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses only [***hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512***] and [***selection: hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com, AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other MAC algorithms***] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSH_EXT.1.7 The TSF shall ensure that [***selection: diffie-hellman-group14-sha1, ecdh-sha2-nistp256***] and [***selection: ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, diffie-hellman-group14-sha256***] are the only allowed key exchange methods used for the SSH protocol.

5.1.1.4 TLS Protocol (FCS_TLS_EXT)

This family defines behavior for implementation of the TLS protocol. This family contains one component, FCS_TLS_EXT.1.

FCS_TLS_EXT.1, TLS Protocol, requires the TSF to implement TLS in accordance with selected standards and to define the connection parameters it uses in the establishment of TLS communications.

Management: **FCS_TLS_EXT.1**

The following actions could be considered for the management functions in FMT:

a) Configuration of TLS connection parameters.

Audit: **FCS_TLS_EXT.1**

There are no auditable events foreseen.

FCS_TLS_EXT.1 – TLS Protocol

Hierarchical To: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FCS_RBG_EXT.1 Random Bit Generation

FPT_STM.1 Reliable Time Stamps

FCS_TLS_EXT.1.1 The TSF shall implement TLS version [***selection: 1.0, 1.1, 1.2***] and reject all other TLS and SSL versions.

FCS_TLS_EXT.1.2 The TSF shall support the following TLS ciphersuites [**assignment: ciphersuites**].

5.1.2 User Data Protection (FDP)

5.1.2.1 Immutability of Stored Data (FDP_IMM_EXT)

This family defines behavior for ensuring that user data stored on the TOE cannot be modified. This family contains one component, FDP_IMM_EXT.1.

FDP_IMM_EXT.1, Immutability of Stored Data, requires the TSF to specify the mechanism by which stored user data is immutable.

Management: **FDP_IMM_EXT.1**

There are no management activities foreseen.

Audit: **FDP_IMM_EXT.1**

There are no auditable events foreseen.

FDP_IMM_EXT.1 – Immutability of Stored Data

Hierarchical To: No other components

Dependencies: No dependencies

FDP_IMM_EXT.1.1 The TSF shall prevent the modification of stored user data by [**assignment: method**].

5.1.2.2 Stored Data Analysis and Reporting (FDP_SAR_EXT)

This family defines behavior for review of user data stored on the TOE and associated metadata. This family contains one component, FDP_SAR_EXT.1.

FDP_SAR_EXT.1, Stored Data Analysis and Reporting, requires the TSF to provide methods to review stored user data and/or analytical metadata or other metrics pertaining to the stored user data.

Management: **FDP_SAR_EXT.1**

The following actions could be considered for the management functions in FMT:

a) Maintenance (deletion, modification, addition) of the group of users with ability to perform analysis and reporting activities.

Audit: **FDP_SAR_EXT.1**

The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

a) Detailed: Execution of analysis and reporting activities.

FDP_SAR_EXT.1 – Stored Data Analysis and Reporting

Hierarchical To: No other components

Dependencies: FDP_ITC.1 Import of User Data without Security Attributes

FDP_SAR_EXT.1.1 The TSF shall provide [**assignment: authorized users**] with the capability to read [**assignment: list of information**] from [**assignment: user data**].

FDP_SAR_EXT.1.2 The TSF shall provide the ability to apply [**assignment: methods of selection and/or ordering**] of [**assignment: user data**] based on [**assignment: search criteria**].

5.1.2.3 Stored Data Confidentiality (FDP_SDC_EXT)

This family defines behavior for ensuring that stored user data is cryptographically protected from unauthorized disclosure. This family contains one component, FDP_SDC_EXT.1

FDP_SDC_EXT.1, Stored Data Confidentiality, requires the TSF to specify the cryptographic method used to protect stored user data from unauthorized disclosure.

Management: **FDP_SDC_EXT.1**

The following actions could be considered for the management functions in FMT:

a) Configuration of the users that are permitted to view stored data.

b) Configuration of key management and/or cryptographic parameters of the implementation used to encrypt and decrypt stored data.

Audit: **FDP_SDC_EXT.1**

There are no auditable events foreseen.

FDP_SDC_EXT.1 – Stored Data Confidentiality

Hierarchical To: No other components

Dependencies: FCS_COP.1 Cryptographic Operation

FDP_ITC.1 Import of User Data without Security Attributes

FDP_SDC_EXT.1.1 The TSF shall ensure the confidentiality of stored user data by [**assignment: cryptographic method**].

5.1.3 Protection of the TSF (FPT)

5.1.3.1 TSF Self-Test (FPT_TST_EXT)

This family defines behavior for self-testing the TSF for correct operation of selected functions. This family contains one component, FPT_TST_EXT.1.

FPT_TST_EXT.1, TSF Self-Test, requires the TOE to self-test some aspect of its own functionality at situations specified in the component.

Management: **FPT_TST_EXT.1**

The following actions could be considered for the management functions in FMT:

a) Manual execution of self-test

b) Interval used for periodic self-test

Audit: **FPT_TST_EXT.1**

The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

a) Minimal: Successful and failed completion of self-test (if the nature of the self-test is such that the auditing function is available following the failure of a self-test).

FPT_TST_EXT.1 – TSF Self-Test	
Hierarchical To:	No other components
Dependencies:	No dependencies
FPT_TST_EXT.1.1	The TSF shall run a suite of the following self-tests [***selection: during initial start-up (on power on), periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self-tests should occur]***] to demonstrate the correct operation of the TSF: [**assignment: list of self-tests run by the TSF**].

5.2 TOE Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE. SFRs were drawn from Part 2 of the Common Criteria v3.1 Revision 4, and from the extended components defined in Section 5.1 above.

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User Identity Association
	FAU_STG.1: Protected Audit Trail Storage
FCS: Cryptographic Support	FCS_CKM.1: Cryptographic Key Generation
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COP.1(1): Cryptographic Operation (Symmetric Encryption)
	FCS_COP.1(2): Cryptographic Operation (Digital Signature)
	FCS_COP.1(3): Cryptographic Operation (Hashing)
	FCS_COP.1(4): Cryptographic Operation (Message Authentication)
	FCS_HTTPS_EXT.1: HTTPS Protocol
	FCS_RBG_EXT.1: Random Bit Generation
	FCS_SSH_EXT.1: SSH Protocol
	FCS_TLS_EXT.1: TLS Protocol
FDP: User Data Protection	FDP_ACC.1: Subset Access Control
	FDP_ACF.1: Access Control Functions
	FDP_ETC.1(1): Export of User Data without Security Attributes (Cloud Storage)
	FDP_ETC.1(2): Export of User Data without Security Attributes (Recovery)
	FDP_IMM_EXT.1: Immutability of Stored Data
	FDP_ITC.1: Import of User Data without Security Attributes
	FDP_SAR_EXT.1: Stored Data Analysis and Reporting
	FDP_SDC_EXT.1: Stored Data Confidentiality
FIA: Identification and Authentication	FIA_UAU.2: User Authentication before Any Action
	FIA_UID.2: User Identification before Any Action

Requirement Class	Requirement Component
FMT: Security Management	FMT_MOF.1: Management of Security Functions Behaviour
	FMT_MTD.1: Management of TSF Data
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	FPT_FLS.1: Failure with Preservation of Secure State
	FPT_TST_EXT.1: TSF Self-Test
FRU: Resource Utilization	FRU_FLT.1: Degraded Fault Tolerance
FTP: Trusted Path/Channels	FTP_ITC.1: Inter-TSF Trusted Channel
	FTP_TRP.1: Trusted Path

Table 1: TOE Security Functional Components

5.2.1 Security Audit (FAU)

FAU_GEN.1 – Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [***not specified***] level of audit; and
c) [**additional cluster audit events in Table 2; SMB/NFS file audit events for the following operations: mount, create, delete, rename, set attributes]**.

Action	Description
Accept	A user accepted the license agreement.
Activate	A user activated a Protection Job.
Cancel	A user canceled an entity such as running Protection Job or Recover task.
Clone	A user cloned an entity such as a Snapshot, VM, View, or SQL Server.
Close	A user closed an SMB file open.
Create	A user created an entity such as a Protection Job.
Deactivate	A user deactivated a Protection Job.
Delete	A user deleted an entity such as a Protection Job, Protection Policy, or View.
Disjoin	A user disjoined the Cluster from an AD domain.
Download	A user downloaded a VMX file or a file from a VM Snapshot.
Join	A user joined the Cluster to an AD domain.
Login	A user logged in to the Cohesity cluster.
Logout	A user logged out of the Cohesity cluster.
Mark	A user marked an entity for removal such as a disk.
Modify	A user modified an entity such as a User, Protection Job, or Remote Cluster.
Pause	A user paused an entity such as a running Protection Job.
Recover	A user recovered an entity such as a VM, file, or SQL Database.
Refresh	A user refreshed the Object hierarchy in a Source.
Register	A user registered an entity such as an External Target (Vault).
Rename	A user renamed an entity such as a Storage Domain.
Resume	A user performed a resume action on a Protection Job.
Run Now	A user performed a Run Now action on a Protection Job.
Unregister	A user unregistered an entity such as a Source.
Upgrade	A user upgraded the Cohesity cluster.

Table 2: Additional Audit Events

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**category of the event**].

FAU_GEN.2 – User Identity Association	
FAU_GEN.2.1	For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_STG.1 – Protected Audit Trail Storage	
FAU_STG.1.1	The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.
FAU_STG.1.2	The TSF shall be able to [***prevent***] unauthorized modifications to the stored audit records in the audit trail.

5.2.2 Cryptographic Support (FCS)

FCS_CKM.1 – Cryptographic Key Generation	
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**elliptic curve cryptography using "NIST curve" P-384**] and specified cryptographic key sizes [**384 bits**] that meet the following: [**FIPS 186-4**].

FCS_CKM.4 – Cryptographic Key Destruction	
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**key zeroization**] that meets the following: [**FIPS 140-2**].

FCS_COP.1(1) – Cryptographic Operation (Symmetric Encryption)	
FCS_COP.1.1(1)	The TSF shall perform [**symmetric encryption/decryption**] in accordance with a specified cryptographic algorithm [**AES in CBC, GCM, CTR modes**] and cryptographic key sizes [**128 bits, 192 bits, 256 bits**] that meet the following: [**NIST SP 800-38**].
Application Note:	*CBC-256 is used for encryption of stored data. CTR-128, 192, and 256 are used for SSH. GCM-256 is used for TLS.*

FCS_COP.1(2) – Cryptographic Operation (Digital Signature)	
FCS_COP.1.1(2)	The TSF shall perform [**digital signature generation and verification**] in accordance with a specified cryptographic algorithm [**RSA, ECDSA**] and cryptographic key sizes [**2048 bits (RSA), 256 bit (ECDSA)**] that meet the following: [**FIPS 186-2 (RSA), FIPS 186-4 (ECDSA)**].

FCS_COP.1(3) – Cryptographic Operation (Hashing)	
FCS_COP.1.1(3)	The TSF shall perform [**cryptographic hashing**] in accordance with a specified cryptographic algorithm [**SHA-1, SHA-2**] and cryptographic key sizes [**160 bits, 256 bits, 384 bits, 512 bits**] that meet the following: [**FIPS 180-4**].

FCS_COP.1(4) – Cryptographic Operation (Message Authentication)	
FCS_COP.1.1(4)	The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**HMAC**] and cryptographic key sizes [**equal to block sizes of 160 bits, 256 bits, 384 bits, 512 bits**] that meet the following: [**FIPS 198-1**].

FCS_HTTPS_EXT.1 – HTTPS Protocol	
FCS_HTTPS_EXT.1.1	The TSF shall implement the HTTPS protocol that complies with RFC 2818.
FCS_HTTPS_EXT.1.2	The TSF shall implement the HTTPS protocol using TLS.

FCS_RBG_EXT.1 – Random Bit Generation	
FCS_RBG_EXT.1.1	The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [***CTR_DRBG (AES)***].
FCS_RBG_EXT.1.2	The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[1] software-based noise source***] with a minimum of [**256 bits**] of entropy at least equal

to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

FCS_SSH_EXT.1 – SSH Protocol

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that compiles with RFCs 4251, 4252, 4253, 4254, [***5647, 5656, 6668***].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [***public key-based, password-based***].

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**65535**] bytes in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses only the following encryption algorithms: [***aes128-ctr, aes192-ctr, aes256-ctr***].

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses only [***ssh-rsa, ecdsa-sha2-nistp256***] and [***rsa-sha2-256, rsa-sha2-512***] as its public key algorithm(s).

FCS_SSH_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses only [***hmac-sha1, hmac-sha2-256, hmac-sha2-512***] and [***hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com***] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSH_EXT.1.7 The TSF shall ensure that [***diffie-hellman-group14-sha1, ecdh-sha2-nistp256***] and [***ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, diffie-hellman-group14-sha256***] are the only allowed key exchange methods used for the SSH protocol.

FCS_TLS_EXT.1 – TLS Protocol

FCS_TLS_EXT.1.1 The TSF shall implement TLS version [***1.2***] and reject all other TLS and SSL versions.

FCS_TLS_EXT.1.2 The TSF shall support the following TLS ciphersuites [**TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384**].

5.2.3 User Data Protection (FDP)

FDP_ACC.1 – Subset Access Control

FDP_ACC.1.1 The TSF shall enforce the [**data access control policy**] on [

- **Subjects: TOE users, organizational users**
- **Objects: file data**
- **Operations: protect, recover, archive, clone, snapshot, access (read/write/execute), analytics, DataLock**].

FDP_ACF.1 – Access Control Functions

FDP_ACF.1.1 The TSF shall enforce the [**data access control policy**] to objects based on the following: [

- **TOE users: whitelist access to specific Sources, Objects, and Views;**
- **Organizational users (via SMB): Active Directory username/group membership, IP address;**
- **Organizational users (via NFS): UID/GID from NFS client, IP address**].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

- **TOE users: allowed actions are defined by assigned user role, allowed objects are defined through enabling the 'Restrict access to specific Objects' user attribute and specifying the allowed Objects and Views;**
- **TOE users: allowed actions against allowed Sources, Objects, or Views can be scheduled to automatically be performed on a periodic basis;**
- **Organizational users (both SMB and NFS): IP address of access attempt is compared to whitelist to determine if access is authorized;**

- **Organizational users (via SMB): Windows ACLs on files and directories compared to user's Active Directory identity to determine if read/write/execute access is permitted;**
- **Organizational users (via NFS): POSIX bits on files and directories compared to user's UID/GID to determine if read/write/execute access is permitted**].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**a TOE user that does not have the 'Restrict access to specific Objects' user attribute enabled may perform all authorized actions against all Objects**].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**Views with the DataLock property set cannot be modified or deleted regardless of privilege**].

FDP_ETC.1(1) – Export of User Data without Security Attributes (Cloud Storage)

FDP_ETC.1.1(1) The TSF shall enforce the [**data access control policy**] when exporting user data, controlled under the SFP(s), **to a remote CSP or Cohesity Cluster** outside of the TOE.

FDP_ETC.1.2(1) The TSF shall export the user data without the user data's associated security attributes.

FDP_ETC.1(2) – Export of User Data without Security Attributes (Recovery)

FDP_ETC.1.1(2) The TSF shall enforce the [**data access control policy**] when exporting user data, controlled under the SFP(s), **to the Source from which the data originated or other recovery location** outside of the TOE.

FDP_ETC.1.2(2) The TSF shall export the user data without the user data's associated security attributes.

FDP_IMM_EXT.1 – Immutability of Stored Data

FDP_IMM_EXT.1.1 The TSF shall prevent the modification of stored user data by [**use of read-only Snapshots; DataLock attribute**].

FDP_ITC.1 – Import of User Data without Security Attributes

FDP_ITC.1.1 The TSF shall enforce the [**data access control policy**] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [**no additional rules**].

FDP_SAR_EXT.1 – Stored Data Analysis and Reporting

FDP_SAR_EXT.1.1 The TSF shall provide [**TOE users with the Reporting privilege**] with the capability to read [**text patterns, strings representative of potential passwords, uncompressed video, administrator-defined MapReduce data**] from [**Snapshots or files stored directly on a View**].

FDP_SAR_EXT.1.2 The TSF shall provide the ability to apply [**filtering based on job name, Storage Domain, server, view, file type, file type, path, last modified date/time, snapshot range**] of [**search data**] based on [**user-specified inputs**].

FDP_SDC_EXT.1 – Stored Data Confidentiality

FDP_SDC_EXT.1.1 The TSF shall ensure the confidentiality of stored user data by [**AES encryption in accordance with FCS_COP.1(1); unique encryption keys per Storage Domain**].

5.2.4 Identification and Authentication (FIA)

FIA_UAU.2 – User authentication before Any Action

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UID.2 – User identification before Any Action

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.2.5 Security Management (FMT)

FMT_MOF.1 – Management of Security Functions Behaviour

FMT_MOF.1.1 **Refinement:** The TSF shall restrict the ability to [***disable, enable, modify the behavior of, execute***] the functions [**audit, backup, restore, archival, DataLock, storage redundancy, clone, analytics**] to [**roles defined in FMT_SMR.2 based on the following rules:**

- **Admin role may manage all functions behavior except for DataLock**
- **Operator may execute the backup and restore functions**
- **Self Service Data Protection may modify the behavior of the backup function, execute the clone function, and modify the behavior of/execute the restore function**
- **Data Security may enable and disable the DataLock function**
- **Viewer may not manage the behavior of any functions**
- **Members of administrator-defined roles may manage the behavior of the functions granted to their assigned role by the Admin that created or last modified the role**].

Application Note: *In order to perform a function, a user must both belong to a role that is authorized to perform the function, and they must be attempting to perform the function against an object that is within the set of restricted objects associated with their username.*

FMT_MTD.1 – Management of TSF Data

FMT_MTD.1.1 The TSF shall restrict the ability to [***query, modify, delete, [create]***] the [**cluster configuration, external target, policy, storage domain, backup, View, Source, snapshot, user, role data**] to **roles defined in FMT_SMR.2 based on the following rules:**

- **All roles may query all data**
- **Admin role may perform any function on any data**
- **Operator may create backup data and snapshot data**
- **Self Service Data Protection may create backup data**
- **Data Security has no additional access to TSF data**
- **Viewer has no additional access to TSF data**
- **Members of administrator-defined roles may perform certain actions against certain types of TSF data granted to their assigned role by the Admin that created or last modified the role**].

Application Note: *In order to interact with TSF data, a user must both belong to a role that is authorized to perform the desired operation against the data, and they must be attempting to perform this operation against data belonging to an object that is within the set of restricted objects associated with their username.*

FMT_SMF.1 – Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [

- **Access Management**
- **Clone Management**
- **Cluster Management**
- **Data Protection**
- **Recovery Management**
- **Storage Management**
- **Analytics Management**
- **Source Access Control**

].

FMT_SMR.2 – Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles: [**Admin, Operator, Viewer, Self Service Data Protection, Data Security, administrator-defined roles**].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions [**one role per user, no role for organizational user**] are satisfied.

5.2.6 Protection of the TSF (FPT)

FPT_FLS.1 – Failure with Preservation of Secure State

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [**node failure, HDD failure, SSD failure, power supply failure, self-test failure**].

FPT_TST_EXT.1 – TSF Self-Test

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [**services and dependencies self-test, file system integrity self-test, node availability self-test, cryptographic self-tests**].

5.2.7 Resource Utilization (FRU)

FRU_FLT.1 – Degraded Fault Tolerance

FRU_FLT.1.1 The TSF shall ensure the operation of [**data availability**] when the following failures occur: [**disk or node failure**].

5.2.8 Trusted Path/Channels (FTP)

FTP_ITC.1 – Inter-TSF Trusted Channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [***the TSF, another trusted IT product***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**backup, recovery, replication, and archival of backup data; remote audit data storage; Active Directory user authentication; Cohesity Analytics**].

FTP_TRP.1 – Trusted Path

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [***remote***] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [***modification***, ***disclosure***].

FTP_TRP.1.2 The TSF shall permit [***remote users***] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [***initial user authentication***, ***[all subsequent user interactions]***].

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 with ALC_FLR.1 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security Architecture Description
	ADV_FSP.2: Security-Enforcing Functional Specification
	ADV_TDS.1: Basic Design
AGD: Guidance documents	AGD_OPE.1: Operational User Guidance
	AGD_PRE.1: Preparative Procedures

Requirement Class	Requirement Component
ALC: Life-cycle support	ALC_CMC.2: Use of a CM System
	ALC_CMS.2: Parts of the TOE CM Coverage
	ALC_DEL.1: Delivery Procedures
	ALC_FLR.1: Basic Flaw Remediation
ASE: Security Target evaluation	ASE_CCL.1: Conformance Claims
	ASE_ECD.1: Extended Components Definition
	ASE_INT.1: ST Introduction
	ASE_OBJ.2: Security Objectives
	ASE_REQ.2: Derived Security Requirements
	ASE_SPD.1: Security Problem Definition
	ASE_TSS.1: TOE Summary Specification
ATE: Tests	ATE_COV.1: Evidence of Coverage
	ATE_FUN.1: Functional Testing
	ATE_IND.2: Independent Testing – Sample
AVA: Vulnerability assessment	AVA_VAN.2: Vulnerability Analysis

Table 3: TOE Security Assurance Components

5.3.1 Development (ADV)

ADV_ARC.1 – Security Architecture Description

ADV_ARC.1.1D	The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.
ADV_ARC.1.2D	The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.
ADV_ARC.1.3D	The developer shall provide a security architecture description of the TSF.
ADV_ARC.1.1C	The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.
ADV_ARC.1.2C	The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.
ADV_ARC.1.3C	The security architecture description shall describe how the TSF initialization process is secure.
ADV_ARC.1.4C	The security architecture description shall demonstrate that the TSF protects itself from tampering.
ADV_ARC.1.5C	The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
ADV_ARC.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2 – Security-Enforcing Functional Specification

ADV_FSP.2.1D	The developer shall provide a functional specification.
ADV_FSP.2.2D	The developer shall provide a tracing from the functional specification to the SFRs.
ADV_FSP.2.1C	The functional specification shall completely represent the TSF.
ADV_FSP.2.2C	The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C	The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.2.4C	For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.
ADV_FSP.2.5C	For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.
ADV_FSP.2.6C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

ADV_TDS.1 – Basic Design

ADV_TDS.1.1D	The developer shall provide the design of the TOE.
ADV_TDS.1.2D	The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.1.1C	The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.1.2C	The design shall identify all subsystems of the TSF.
ADV_TDS.1.3C	The design shall describe the behavior of each SFR-supporting or SFR non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.
ADV_TDS.1.4C	The design shall summarise the SFR-enforcing behavior of the SFR-enforcing subsystems.
ADV_TDS.1.5C	The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.
ADV_TDS.1.6C	The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.
ADV_TDS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.1.2E	The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance Documents (AGD)

AGD_OPE.1 – Operational User Guidance

AGD_OPE.1.1D	The developer shall provide operational user guidance.
AGD_OPE.1.1C	The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2C	The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.
AGD_OPE.1.3C	The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
AGD_OPE.1.4C	The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_OPE.1.5C	The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C	The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7C	The operational user guidance shall be clear and reasonable.
AGD_OPE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 – Preparative Procedures

AGD_PRE.1.1D	The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1C	The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2C	The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2E	The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle Support (ALC)

ALC_CMC.2 – Use of a CM System

ALC_CMC.2.1D	The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.2.2D	The developer shall provide the CM documentation.
ALC_CMC.2.3D	The developer shall use a CM system.
ALC_CMC.2.1C	The TOE shall be labelled with its unique reference.
ALC_CMC.2.2C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.2.3C	The CM system shall uniquely identify all configuration items.
ALC_CMC.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.2 – Parts of the TOE CM Coverage

ALC_CMS.2.1D	The developer shall provide a configuration list for the TOE.
ALC_CMS.2.1C	The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.
ALC_CMS.2.2C	The configuration list shall uniquely identify the configuration items.
ALC_CMS.2.3C	For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 – Delivery Procedures

ALC_DEL.1.1D	The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2D	The developer shall use the delivery procedures.
ALC_DEL.1.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.1 – Basic Flaw Remediation	
ALC_FLR.1.1D	The developer shall document and provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.1.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.1.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.1.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.1.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR. 1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Security Target Evaluation (ASE)

ASE_CCL.1 – Conformance Claims	
ASE_CCL.1.1D	The developer shall provide a conformance claim.
ASE_CCL.1.2D	The developer shall provide a conformance claim rationale.
ASE_CCL.1.1C	The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.
ASE_CCL.1.2C	The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.
ASE_CCL.1.3C	The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
ASE_CCL.1.4C	The CC conformance claim shall be consistent with the extended components definition.
ASE_CCL.1.5C	The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.
ASE_CCL.1.9C	The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.
ASE_CCL.1.10C	The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.
ASE_CCL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1 – Extended Components Definition	
ASE_ECD.1.1D	The developer shall provide a statement of security requirements.
ASE_ECD.1.2D	The developer shall provide an extended components definition.
ASE_ECD.1.1C	The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C	The extended components definition shall define an extended component for each extended security requirement.
ASE_ECD.1.3C	The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.
ASE_ECD.1.4C	The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.
ASE_ECD.1.5C	The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.
ASE_ECD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_ECD.1.2E	The evaluator shall confirm that no extended component can be clearly expressed using existing components.

ASE_INT.1 – ST Introduction

ASE_INT.1.1D	The developer shall provide an ST introduction.
ASE_INT.1.1C	The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
ASE_INT.1.2C	The ST reference shall uniquely identify the ST.
ASE_INT.1.3C	The TOE reference shall identify the TOE.
ASE_INT.1.4C	The TOE overview shall summarise the usage and major security features of the TOE.
ASE_INT.1.5C	The TOE overview shall identify the TOE type.
ASE_INT.1.6C	The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C	The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.8C	The TOE description shall describe the logical scope of the TOE.
ASE_INT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_INT.1.2E	The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_OBJ.2 – Security Objectives

ASE_OBJ.2.1D	The developer shall provide a statement of security objectives.
ASE_OBJ.2.2D	The developer shall provide a security objectives rationale.
ASE_OBJ.2.1C	The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.
ASE_OBJ.2.2C	The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.
ASE_OBJ.2.3C	The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.
ASE_OBJ.2.4C	The security objectives rationale shall demonstrate that the security objectives counter all threats.
ASE_OBJ.2.5C	The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.
ASE_OBJ.2.6C	The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.
ASE_OBJ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_REQ.2 – Derived Security Requirements

ASE_REQ.2.1D	The developer shall provide a statement of security requirements.

ASE_REQ.2.2D	The developer shall provide a security requirements rationale.
ASE_REQ.2.1C	The statement of security requirements shall describe the SFRs and the SARs.
ASE_REQ.2.2C	All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.
ASE_REQ.2.3C	The statement of security requirements shall identify all operations on the security requirements.
ASE_REQ.2.4C	All operations shall be performed correctly.
ASE_REQ.2.5C	Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
ASE_REQ.2.6C	The security requirements rationale shall trace each SFR back to the security objectives for the TOE.
ASE_REQ.2.7C	The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.
ASE_REQ.2.8C	The security requirements rationale shall explain why the SARs were chosen.
ASE_REQ.2.9C	The statement of security requirements shall be internally consistent.
ASE_REQ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 – Security Problem Definition

ASE_SPD.1.1D	The developer shall provide a security problem definition.
ASE_SPD.1.1C	The security problem definition shall describe the threats.
ASE_SPD.1.2C	All threats shall be described in terms of a threat agent, an asset, and an adverse action.
ASE_SPD.1.3C	The security problem definition shall describe the OSPs.
ASE_SPD.1.4C	The security problem definition shall describe the assumptions about the operational environment of the TOE.
ASE_SPD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 – TOE Summary Specification

ASE_TSS.1.1D	The developer shall provide a TOE summary specification.
ASE_TSS.1.1C	The TOE summary specification shall describe how the TOE meets each SFR.
ASE_TSS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_TSS.1.2E	The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.3.5 Tests (ATE)

ATE_COV.1 – Evidence of Coverage

ATE_COV.1.1D	The developer shall provide evidence of the test coverage.
ATE_COV.1.1C	The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 – Functional Testing

ATE_FUN.1.1D	The developer shall test the TSF and document the results.
ATE_FUN.1.2D	The developer shall provide test documentation.
ATE_FUN.1.1C	The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C	The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.3C	The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.4C	The actual test results shall be consistent with the expected test results.
ATE_FUN.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 – Independent Testing – Sample

ATE_IND.2.1D	The developer shall provide the TOE for testing.
ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.
ATE_IND.2.3E	The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.6 Vulnerability Assessment (AVA)

AVA_VAN.2 – Vulnerability Analysis

AVA_VAN.2.1D	The developer shall provide the TOE for testing.
AVA_VAN.2.1C	The TOE shall be suitable for testing.
AVA_VAN.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.2.2E	The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.
AVA_VAN.2.3E	The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.
AVA_VAN.2.4E	The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This section describes the following security functions implemented by the TOE to satisfy the SFRs claimed in Section 5.2:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- Resource Utilization
- Trusted Path/Channels

6.1 Security Audit

The TOE generates audit events for security-relevant activity. Audit events are distinguished as filer logs and cluster logs. Filer logs refer to all file operations on data stored in SMB/NFS Views of data stored on the TOE that is externally accessible. Filer logs are generated for the following operations: Log On, Log Off, Mount, Open, Close, Create, Delete, Rename, and Set Attributes. Each filer log contains fields for timestamp, protocol, client IP address, username/UID, domain name, share name, entity name(s)/ID(s), and entity attributes.

The table below lists the cluster log events that are generated by the TSF, in addition to logs for startup and shutdown of the TOE as well as any change in audit configuration.

Action	Description
Accept	A user accepted the license agreement.
Activate	A user activated a Protection Job.
Cancel	A user canceled an entity such as running Protection Job or Recover task.
Clone	A user cloned an entity such as a Snapshot, VM, View, or SQL Server.
Close	A user closed an SMB file open.
Create	A user created an entity such as a Protection Job.
Deactivate	A user deactivated a Protection Job.
Delete	A user deleted an entity such as a Protection Job, Protection Policy, or View.
Disjoin	A user disjoined the Cluster from an AD domain.
Download	A user downloaded a VMX file or a file from a VM Snapshot.
Join	A user joined the Cluster to an AD domain.
Login	A user logged in to the Cohesity cluster.
Logout	A user logged out of the Cohesity cluster.
Mark	A user marked an entity for removal such as a disk.
Modify	A user modified an entity such as a User, Protection Job, or Remote Cluster.
Pause	A user paused an entity such as a running Protection Job.
Recover	A user recovered an entity such as a VM, file, or SQL Database.
Refresh	A user refreshed the Object hierarchy in a Source.
Register	A user registered an entity such as an External Target (Vault).
Rename	A user renamed an entity such as a Storage Domain.
Resume	A user performed a resume action on a Protection Job.
Run Now	A user performed a Run Now action on a Protection Job.
Unregister	A user unregistered an entity such as a Source.
Upgrade	A user upgraded the Cohesity cluster.

Table 4: Auditable Events

Each of these audit logs include the following information, as appropriate:

- Date/time
- User
- Entity Name
- Category (high-level identifier for the type of object that the action is performed against, e.g. Active Directory, Disk, Cluster Partition, Group, Protection Job, User)
- Action
- Details

When configuring logs, the TOE-internal retention period for this data may be specified. The default retention period for filer audit logs is 90 days and for cluster audit logs is 180 days. There is no method by which administrators can modify or manually delete audit records, regardless of authorization. In the evaluated configuration, the TOE will be configured to transmit audit data to a remote syslog server. When operating in a Cohesity cluster, each instance of the TOE will independently transmit its own audit events to the same syslog server; there is no intermediate step where the audit data is first aggregated and then sent from a central point.

The Security Audit security function satisfies the following security functional requirements:

- FAU_GEN.1 – audit records are generated for security relevant events and include the date and time of the event, type of event, subject identity, and outcome of the event.
- FAU_GEN.2 – the TOE associates each auditable event resulting from actions of identified users with the identity of the user that caused the event.
- FAU_STG.1 – the TOE protects stored audit records from unauthorized modification and deletion.

6.2 Cryptographic Support

The TOE includes the FIPS-Validated "Cohesity FIPS Object Module for OpenSSL" (CMVP certificate #2676). This module provides the following NIST-validated cryptographic algorithm services to the TOE for use in protection of data at rest and in transit:

Requirement	Function	Algorithm	Certificate
FCS_CKM.1	Asymmetric key generation	FIPS 186-4 ECDSA key pair generation (P-384)	ECDSA #1442
FCS_COP.1(1)	Symmetric encryption and decryption	NIST SP 800-38 AES CBC, GCM, CTR (128/192/256-bit)	AES #5426
FCS_COP.1(2)	Digital signature generation and verification	FIPS 186-2 RSA (2048-bit)	RSA #2906
		FIPS 186-4 ECDSA (256-bit)	ECDSA #1442
FCS_COP.1(3)	Hashing	FIPS 180-4 (SHA-1, SHA-2 256/384/512)	SHS #4353
FCS_COP.1(4)	Message authentication	FIPS 198-1 (SHA-1, SHA-2 256/384/512)	HMAC #3591
FCS_RBG_EXT.1	Random bit generation	NIST SP 800-90A (AES_CTR DRBG)	DRBG #2117

Table 5: NIST-Validated Cryptographic Algorithms

All key and keying material data is zeroized when no longer in use. Unpredictable random data used to seed the DRBG is gathered through the normal functioning of the CentOS platform on which the TOE is run. The TSF collects this data from the /dev/random device on the OS.

The TSF generates keys used for establishment of trusted communications. The TOE relies on the environmental Key Manager to generate and store the 256-bit AES-CBC keys used by the TOE as DEKs for the protection of stored data at rest and for encryption of data stored on an environmental Cloud Service Provider (CSP).

The TSF implements TLS, as both a client and server, for remote administration as well as for external client connections to AD and CSPs. Remote administration and CSP connectivity also use HTTPS. For all uses of TLS, the TSF enforces the use of TLS 1.2 and uses only the TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ciphersuite. By default, the TOE is loaded with self-signed certificates that are implicitly trusted by the TSF. All other certificates presented to the TOE, including user-supplied server certificates that are intended to overwrite the default certificates on the TOE,

are checked for validity before use. In the event the TOE receives an invalid certificate as part of establishing a TLS or HTTPS channel, the connection attempt will be rejected.

The TSF also implements SSH as a server for remote administration via the CLI. The TOE's SSH server accepts both password and public key authentication methods. Packets larger than 65535 bytes will be rejected by the SSH server. Rekeying of the SSH channel occurs when 4GB of data has been transmitted. The SSH server supports the following connection parameters:

- Symmetric encryption: aes128-ctr, aes192-ctr, aes256-ctr
- Public key authentication: ssh-rsa, ecdsa-sha2-nistp256, rsa-sha2-256, rsa-sha2-512
- MAC: hmac-sha1, hmac-sha2-256, hmac-sha2-512, hmac-sha1-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com
- Key exchange: diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, diffie-hellman-group14-sha256

The Cryptographic Support security function satisfies the following security functional requirements:

- FCS_CKM.1 – the TOE generates asymmetric keys for use in encryption.
- FCS_CKM.4 – the TOE destroys stored key material when no longer needed.
- FCS_COP.1(1) – the TOE provides a symmetric cryptographic algorithm implementation to establish cryptographic communications and encrypt data at rest.
- FCS_COP.1(2) – the TOE provides a public key cryptographic algorithm implementation to establish cryptographic communications and verify digital signatures.
- FCS_COP.1(3) – the TOE provides a hash algorithm to establish cryptographic communications and verify digital signatures.
- FCS_COP.1(4) – the TOE provides a message authentication algorithm to establish cryptographic communications.
- FCS_HTTPS_EXT.1 – the TOE provides an HTTPS interface to cryptographically protect user and TSF data in transit.
- FCS_RBG_EXT.1 – the TOE uses a strong random number generator to ensure that generated keys provide their intended strength.
- FCS_SSH_EXT.1 – the TOE provides an SSH interface to cryptographically protect user and TSF data in transit.
- FCS_TLS_EXT.1 – the TOE provides a TLS interface to cryptographically protect user and TSF data in transit.

6.3 User Data Protection

The primary security function of the TOE is to ingest copies of data from entities in the operational environment, referred to as Sources, and store the data as backups within the TOE boundary. The following types of Sources are supported:

- Virtual Servers:
  - VMware
  - Hyper-V
  - AHV
  - Red Hat
- Physical Servers:
  - Windows: Server 2008, 2012, 2016
  - Linux: Ubuntu, CentOS, RHEL, Suse, Debian
  - IBM AIX
- Databases:
  - MS SQL Server
  - Oracle
- Storage Arrays:

    - Pure Storage FlashArray
- NAS:
    - NetApp
    - Isilon
    - Pure FlashBlade
    - Generic NAS

Depending on the type of Source, it may be necessary to install a Cohesity Agent on the target system to create an interface used to acquire the data from it. Specifically, this is necessary for all physical servers, all databases, and all Hyper-V VMs. If multiple Hyper-V VM Sources are running on an SCVMM server, the Cohesity Agent can be installed on the SCVMM server itself.

The TOE contains one or more partitions. Partitions are physically and logically isolated from one another such that a user can only access one partition at a time. A partition contains one or more Storage Domains where the backup data resides. A Storage Domain may contain one or more Views (or datastores), which are logical storage locations with SMB and/or NFS mount paths. The logical representation of Sources in Storage Domains are known as Objects, which may or may not be directly accessible via Views.

All Storage Domains are encrypted using unique AES-256 keys. These keys are generated and maintained by the environmental Key Manager. Encryption ensures that a user cannot view data outside of a Storage Domain that they are authorized to interact with.

The TSF enforces a data access control policy on the ingested file data so that only authorized subjects can interact with it. The TOE defines two distinct types of subjects:

- TOE users, which have the ability to access data through the TSF management interfaces.
- Organizational users, which do not have access to the management interface and access data entirely through its exposure to the operational environment via SMB and/or NFS.

The TSF provides the following actions that can be performed against the backup data:

- Protect: the act of backing up data from a Source.
- Recover: the act of restoring backed-up data to the operational environment, whether it is back to the same Source that it originated from or to a new destination. This can also refer to the act of re-acquiring data from an archive.
- Archive: the act of relegating “cold” data to an off-site storage location, such as a Cloud Service Provider (CSP) or secondary Cohesity cluster. This same interface may also be used for replication or tiering.
- Clone: create a duplicate of backed-up data.
- Snapshot: create a copy of a View.
- Access: browse the contents of a View via SMB/NFS share.
- Analytics: generate reports on the contents of a Snapshot or View.
- DataLock: designate a View or individual files as read-only, regardless of their native permissions as extracted from the original Source.

Of these operations, “Access” is performed by organizational users, and all other operations are performed by TOE users.

The Protect, Recover, Archive, Clone, and Snapshot operations can be performed manually by TOE users, or they can be performed on a pre-configured schedule by defining a Protection Policy and associating it with a Source. This Protection Policy will define how the associated Source has its data protected, replicated, and archived. A Protection Policy can be reused and applied to multiple different Sources.

The data access control policy is enforced on TOE users based on two separate factors. The first is the user’s assigned role, which defines the operations they are allowed to perform (“operation level”). Refer to section 6.5 for a discussion on the various roles and their associated privileges. The second factor is the set of objects that the user is able to perform their authorized operations against (“object level”). Each user account contains a flag called ‘Restrict access to specific

Objects'. If enabled, the user must then be assigned specific Objects and Views that they are authorized to interact with. If disabled, the user can interact with all Objects stored by the TOE.

The data access control policy is enforced on organizational users based on two separate factors, each different from those that apply to TOE users. The first factor is the IP address that they are using to attempt to access the TOE data, either as an SMB or NFS share. The TSF applies IP address whitelisting and does not permit any access to a non-whitelisted IP address. The second factor differs based on the type of network share. For SMB shares, the TSF will use the organizational user's authenticated identity (on their own host system) and derive Active Directory group information from it. This user and group information is then checked against Windows ACLs for individual files and directors to determine if the requested access is permitted. For NFS shares, the user will use an NFS client in the operational environment to be authenticated, and the NFS client will send UID/GID data to the TOE. This data is then used to determine if requested actions (read/write/execute) are permitted using POSIX bits (i.e. Linux-style file permissions).

A TOE user with sufficient privileges can apply the DataLock property to a View. This property places the View in an indefinite read-only mode for all users that attempt to access it. This lasts until the same user removes the DataLock from that View.

Data is imported from the operational environment through the use of a Protection Job, which will back up data from a chosen Source and place it inside of a Storage Domain. The Protection Job also defines the retention period for the data as well as whether it is a full or incremental backup. Backups may also be scheduled such that incremental backups are performed on a schedule but a full backup is also performed but on a longer period. Multiple backups that share unmodified data will automatically be de-duplicated by the TSF.

The data access control policy is enforced not just on acquisition of data, but also on recovery and archival of the data. Sufficient privileges are necessary for a TOE user to configure the CSP(s) and/or remote Cohesity Cluster(s) will archive data to, or for a TOE user to restore data back to a Source (either the Source that the data was originally backed up from or to a different Source of the same type). The TOE supports AWS, Microsoft Azure, and Google Cloud CSPs for archival.

Immutability of stored data is enforced using two methods: the aforementioned DataLock property, and with Snapshots. Snapshots represent the contents of a View at a particular point in time, and are always read-only (they differ from a View with DataLock set in that an authorized administrator can delete a Snapshot but not a View with DataLock set). By capturing frequent Snapshots of a View, the data on a Source can be reverted with minimal disruption, in the event that the Source suffered a malware attack or other catastrophic alteration.

The TOE includes an Analytics Workbench application that provides a MapReduce framework for analysis and reporting on data stored within the TOE. This can be used to search for significant data. By default, the TOE includes three Analytics Workbench apps:

- Pattern Finder – searches for multiple patterns, such as \d{3}-\d{2}-\d{4}, at once across a wide variety of file formats such as Microsoft Office files, plain text files, PDFs, and Zip files.
- Password Detector – searches for files with alphanumeric strings that may be indicative of cleartext passwords or hashes.
- Video Compressor – searches for uncompressed video in formats such as .mpeg, .avi, .mp3, .flv, .wmv, and .mov and compresses them to preserve storage space.

Additional apps can be developed using Java to search and analyze data based on organizational needs. When running an app, it will be targeted against one or more Snapshots or Views. Various filter conditions can be applied to narrow the search based on the type of Object being searched. Filter conditions for searching for files on a Snapshot include:

- Job name – only processes files in Snapshots that were created by a particular Protection Job.
- Storage Domain – only processes files found in Snapshots inside of a particular Storage Domain.
- Server – only processes files found in Snapshots of the specified Server(s).
- File Type – only processes files with the specified file extension(s).
- Path – only processes files found in a particular file path (or recursive subdirectories of that path). Note that wildcards are not supported for this.

- Last modified date/time – only processes files that have been modified within the specified time period.
- Snapshot range – selection to process files in only the most recent Snapshot, in all Snapshots, or in a range of Snapshots taken between a given start/end date range.

Filter conditions for searching on a View include:

- Storage Domain – only processes files found in Views inside of a particular Storage Domain.
- View – only processes files found in Snapshots inside of a specific View.
- File Type – only processes files with the specified file extension(s).
- Path – only processes files found in a particular file path (or recursive subdirectories of that path). Note that wildcards are not supported for this.
- Last modified date/time – only processes files that have been modified within the specified time period.

This functionality can only be performed by authorized users. Refer to 'Analytics Management' in section 6.5 for the role authorizations required to interact with Analytics Workbench.

The User Data Protection security function satisfies the following security functional requirements:

- FDP_ACC.1 – the TOE defines a policy that governs how backup data is backed up and restored to/from the TOE's operational environment and the extent to which these activities can be analyzed by authorized users.
- FDP_ACF.1 – the TOE implements rules that enforce the policy defined by FDP_ACC.1.
- FDP_ETC.1(1) – the TOE provides mechanisms to archive backup data to cloud storage repositories in the TOE's operational environment.
- FDP_ETC.1(2) – the TOE provides mechanisms to restore backup data to various sources in the TOE's operational environment.
- FDP_IMM_EXT.1 – the TOE provides immutable storage for data at rest.
- FDP_ITC.1 – the TOE provides mechanisms to back up data from various sources in the TOE's operational environment.
- FDP_SAR_EXT.1 – the TOE provides authorized users with the ability to perform analysis activities on its data backup and restore operations.
- FDP_SDC_EXT.1 – the TOE provides a method to encrypt stored backup data.

6.4 Identification and Authentication

The TSF requires all users to authenticate to the TOE prior to allowing and security-relevant activity on its management interfaces. The TOE can support the definition of local user accounts, with credential information stored on the TOE, as well as AD authentication. AD authentication for a particular domain is permitted for a Cohesity cluster when that cluster has been joined to that domain. Users defined for management functions are the same regardless of interface is used to manage the TOE; there is no separate set of credentials for CLI access versus web GUI access or REST API access.

The TSF also provides external file access of stored data via SMB and NFS shares. Access to SMB is restricted using Windows-style ACLs. Users attempting to access SMB data are authenticated via AD. Authentication to NFS depends on the client used to access the share; it is not enforced by the TSF. However, the user's UID/GID data is passed to the TOE via the NFS client, so the TSF is able to require external authentication prior to granting any access to files stored by it.

The Identification and Authentication function satisfies the following security functional requirements:

- FIA_UAU.2 – the TOE requires all administrators to be authenticated before access to the TSF can be granted, regardless of the interface used to initiate this access.

- FIA_UID.2 – the TOE requires all administrators to be identified before access to the TSF can be granted, regardless of the interface used to initiate this access.

6.5 Security Management

The TOE provides administrative access to TSF functions and data via a Web GUI (Cohesity Dashboard), CLI, and REST API. To access the CLI, a user must access the TOE's host platform via SSH as the 'cohesity' Linux user account. This user account can then be used to launch the CLI application. The TOE's management functions and the interfaces that can be used to perform them are listed in the following table:

Operation	Function/Data	GUI	CLI	REST API
Modify the behavior of	Audit function	X	X	X
Enable/Disable/Execute	Backup function	X		X
Enable/Disable/Execute	Restore function	X		X
Enable/Disable/Execute	Archival function	X		X
Enable/Disable/Execute	DataLock function	X		X
Enable/Disable/Modify the behavior of	Storage redundancy function	X	X	X
Execute	Clone function	X		X
Execute	Analytics function	X		
Modify the behavior of	Cluster management function	X	X	X
Query, Modify	Cluster configuration data	X		
Query, Create, Modify, Delete	External target data	X		
Query, Create, Modify, Delete	Policy data	X		X
Query, Create, Modify, Delete	Storage domain data	X		X
Query, Create, Modify, Delete	Backup data	X		X
Query, Create, Modify, Delete	View data	X		X
Query, Create, Modify, Delete	Source data	X		X
Query, Create, Modify, Delete	Snapshot data	X		X
Query, Create, Modify, Delete	User data	X		X
Query, Create, Modify, Delete	Role data	X		X

Table 6: Management Actions

Collectively, these management actions are organized into logical groups called workflows. Workflows are defined in additional detail later in this section.

By default the TOE includes five administrative roles. These roles and a summary of their associated privileges are as follows:

- Admin: Users assigned the Admin role have full access to all actions and workflows within the GUI and CLI except for changing DataLock Views and setting their expiration dates. Note that by default the TSF provides a default 'System Admin' user with the same privileges as the Admin role. This default user cannot be deleted.
- Operator: Users assigned the Operator role can run existing Protection Jobs and create Recover Tasks.
- Self Service Data Protection: Users assigned this role can manage Clones, and Protection Jobs, and Policies, and they can also create Recover Tasks.
- Data Security: Users assigned the Data Security role can create DataLock Views and set DataLock expiration dates.
- Viewer: Users assigned the Viewer role have read-only access for all workflows within the GUI.

All roles have the Viewer role permissions as a minimum baseline. This includes both the default roles as well as all custom roles, described below.

If a user has their access restricted to specific objects, then the administrative functions they can perform are limited in scope to the restricted objects.

In addition to these pre-defined roles, the TSF provides the ability for a sufficiently-privileged user (by default, users with the Admin role) to define new roles that can be assigned to users. Roles have their authority defined by granting them access to workflows, which are categories of individual privileges that are all related to a certain function. The table below lists the workflows provided by the TOE, the privileges that are associated with them, and a description of each privilege.

Each user of the TOE's management interfaces is assigned exactly one role. An organizational user who only interacts with the TSF through SMB and NFS (and not the management interfaces) does not need to have an account or role assignment on the TOE.

Workflow	Privilege	Description
Access Management	View Users	Allows viewing users, groups, and other access control information.
	Manage Users	Allows modifying users, groups and other access control information. If the user's access is restricted to specific Objects, then this privilege is not granted, even if the role grants it.
Clone Management	View Clone Tasks	Allows the user to view Clone tasks.
	Manage Clone Tasks	Allows the user to create, modify, and delete Clone tasks.
Cluster Management	View Cluster Details	Allows the user to view Cluster settings. The user cannot make changes.
	Manage Cluster	Allows the user to modify Cluster setup and to add and remove Nodes.
	Cluster Support	Allows the user to perform support related operations such as collecting data and browsing logs.
	Upgrade Cluster	Allows the user to upgrade and patch the Cluster.
	Manage External Targets and Remote Clusters	Allows the user to register and modify External Targets for archival and remote Clusters for replication.
Data Protection	View Protection Policies and Jobs	Allows the user to view Protection Jobs, Protection Policies, Protection Sources, and Protection Job runs. The user cannot make any changes.
	Manage Protection Policies and Jobs	Allows the user to create, modify, and delete Protection Jobs and Policies as well as the ability to delete Job runs.
	Protection Job Operator	Allows the user to run, cancel, or pause a Protection Job.
	Manage Sources	Allows the user to register or delete a Protection Source and modify its information. If the user's access is restricted to specific Objects, then this privilege is not granted, even if the role grants it.
Recovery Management	View Recover Tasks	Allows the user to view Recover and Clone Tasks. The user cannot make any changes.
	Manage Recover Tasks	Allows the user to create Recover Tasks.
	Download File	Allows the user to download files. The user also needs the 'Manage Recover Tasks' privilege to download files.
	Recover Remote External Targets	Allows the user to search External Targets and recover using data archived from a remote Cluster.
Storage Management	Read Cohesity Views	Allows the user to view Cohesity Views.
	Manage Cohesity Views	Allows the user to create, modify and delete Cohesity Views.
Analytics Management	View Analytics Workbench	Allows the user to view Analytics Workbench related entities.
	Manage Analytics Workbench	Allows the user to modify Analytics Workbench related entities.
	Reporting	Allows the user to generate and view reports.
Source Access Control	Data Security	Allows the user to lock a View and set its lock expiration date.

Table 7: Management Functions

The Security Management function satisfies the following security functional requirements:

- FMT_MOF.1 – the TOE provides mechanisms to manage the behavior of its security functionality and restricts these mechanisms to authorized users.
- FMT_MTD.1 – the TOE provides mechanisms to manage security-relevant configuration data and restricts these mechanisms to authorized users.
- FMT_SMF.1 – the TOE provides security-relevant management functionality across multiple different interfaces.
- FMT_SMR.2 – the TOE defines administrative roles that grant varying degrees of permission to interact with the TSF functions and data.

6.6 Protection of the TSF

The TSF maintains data security in the event of a node or component failure. All nodes in a cluster enforce the same access control policy, so any data replicated to other nodes will remain secured. When a node failure occurs, the node will enter a degraded state where it does not provide any security-relevant functionality, and it will attempt to resolve this through a reboot. The following failures will trigger this condition:

- Node failure (NIC, SATA DOM, RAM)
- HDD failure
- SSD failure
- Power supply failure
- Self-test failure

The TOE performs a number of self-tests at power-on to ensure the proper operation of its security functionality. A self-test is performed at service boot to ensure that all TOE services and dependencies are running. Additional tests are performed to ensure that the file system is intact and that any other nodes in the cluster are reachable. The TSF also runs the following suite of cryptographic self-tests at startup:

- Cryptographic module software integrity test (HMAC-SHA1)
- HMAC known answer test
- AES known answer test
- AES-GCM known answer test
- RSA known answer test
- DRBG known answer test
- ECDSA pair-wise consistency test

The Protection of the TSF security function satisfies the following security functional requirements:

- FPT_FLS.1 – the TOE responds to self-test and component failures by ensuring that backup data is retained and insecure operations cannot be performed.
- FPT_TST_EXT.1 – the TOE performs self-testing of its own functionality to ensure that it is operating in a known state at all times.

6.7 Resource Utilization

The TSF maintains availability of stored data through hardware redundancy as well as data redundancy. Hardware redundancy is achieved by ensuring that the TOE will continue operating if the following failure conditions are experienced:

- Node failure (NIC, SATA DOM, RAM): up to one node per Cohesity cluster
- HDD failure: one or more disks from the same node
- SSD failure: one or more disks from the same node
- Power supply failure: one power supply per block

Data redundancy refers to the notion that when the TOE is deployed in a cluster, all data stored on it has a second copy created on a different node. During operation of the cluster, at any point where one copy of this data become unavailable, the remaining available copy will subsequently make a new copy of itself and store that on a different node. In this manner, availability of data is maintained unless a cluster experiences simultaneous failures of multiple nodes.

In addition to this method of data redundancy, the TSF can also employ remote replication. Remote replication is associated with a Protection Policy, so it can be run against an entire cluster, entire node, or specific Snapshots/Views. When remote replication is used, data obtained by the TOE is periodically transmitted to a remote Cohesity cluster in the operational environment. When a Protection Job is used to capture Snapshots of Source data on a periodic interval, remote replication for that data can be configured to occur over any time interval at least as long as the Protection Job interval (e.g., if a Protection Job is configured to run hourly, remote replication may occur on a schedule that is hourly or longer, but it cannot be under an hour).

The Resource Utilization security function satisfies the following security functional requirements:

- FRU_FLT.1 – the TOE provides high availability for backup data so that component failure or unavailability does not compromise the availability of backup data.

6.8 Trusted Path/Channels

The TOE uses its FIPS-validated cryptographic module to provide secure communications between itself and remote IT entities/administrators. Specifically, the following interfaces use the following trusted channels/paths:

- TOE to AD trusted channel – LDAP over TLS
- TOE to remote CSP trusted channel – TLS/HTTPS
- TOE to Secondary Cohesity Cluster trusted channel – TLS/HTTPS
- TOE to Source trusted channel – TLS/HTTPS
- TOE to Cohesity Analytics – TLS
- Remote Source to TOE trusted channel – TLS/HTTPS
- Remote CLI to TOE trusted path – SSH
- Remote GUI to TOE trusted path – TLS/HTTPS
- Remote REST API to TOE trusted path – TLS/HTTPS

For each trusted path and channel, the cryptography and protocol behavior used to establish the communications are described in section 6.2 above. All trusted channels are TSF-initiated except for remote Sources, which can push data to the TOE. All trusted paths are remotely initiated.

Trusted paths are used for remote management. Trusted channels are used for all operations that require transfer of data to or from the TOE (except for remote access via a mounted drive over SMB or NFS), as well as remote audit data storage and user authentication via AD.

The Trusted Path/Channels security function satisfies the following security functional requirements:

- FTP_ITC.1 – the TOE provides secure interfaces to send and receive security-relevant data to/from its operational environment.
- FTP_TRP.1 – the TOE provides secure interfaces for remote administrators to perform management functions.

7. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives
- Security Functional Requirements
- Security Assurance Requirements
- Requirement Dependencies
- TOE Summary Specification.

7.1 Security Objectives Rationale

This section shows that all secure usage assumptions and threats are completely covered by security objectives for the TOE or operational environment. In addition, each objective counters or addresses at least one assumption or threat.

	T.DATA_DISCLOSURE	T.DATA_ERASURE	T.DATA_MODIFICATION	T.TSF_FAILURE	T.UNAUTH_ACCESS	T.UNDETECTED_ACTIONS	T.UNTRUSTED_COMMS	A.COMPONENTS_RUNNING	A.LIMITED	A.PHYSICAL	A.REGULAR_UPDATES	A.SYSTEM_TIME	A.TRUSTED_ADMIN
O.AUDIT						X							
O.AVAILABILITY		X											
O.I_AND_A					X								
O.PROTECTED_BACKUPS	X	X	X										
O.PROTECTED_COMMS					X		X						
O.SECURITY_MANAGEMENT					X								
O.SELF_TESTS				X									
OE.COMPONENTS_RUNNING				X				X					
OE.LIMITED				X	X				X				
OE.NETWORK	X												
OE.PHYSICAL	X	X	X							X			
OE.REGULAR_UPDATES				X							X		
OE.SYSTEM_TIME						X	X					X	
OE.TRUSTED_ADMIN						X							X

Table 8: Security Problem Definition to Security Objective Correspondence

T.DATA_DISCLOSURE

A malicious user or process may access backup data without authorization.

This threat is countered by the following security objectives:

- O.PROTECTED_BACKUPS – addresses this threat by enforcing an access control policy and cryptographic measures that restrict access to backup data to authorized users, preventing its disclosure.

- OE.NETWORK – addresses this threat by ensuring that there is no ability for an unauthorized entity to eavesdrop on communications between distributed nodes because the nodes are connected via a protected local area network.
- OE.PHYSICAL – addresses this threat by ensuring that unauthorized individuals do not have physical access to the TOE storage devices, preventing extraction or duplication of the data.

T.DATA_ERASURE

A malicious user or process may destroy backup data without authorization.

This threat is countered by the following security objectives:

- O.AVAILABILITY – addresses this threat by ensuring that data is placed in redundant storage locations such that the failure or unavailability of a single node does not result in permanent data loss.
- O.PROTECTED_BACKUPS – addresses this threat by enforcing an access control policy and cryptographic measures that restrict access to backup data to authorized users, preventing its deletion.
- OE.PHYSICAL – addresses this threat by ensuring that unauthorized individuals do not have physical access to the TOE storage devices, preventing destruction of the physical media on which the data is stored.

T.DATA_MODIFICATION

A malicious user or process may corrupt or otherwise modify backup data without authorization.

This threat is countered by the following security objectives:

- O.PROTECTED_BACKUPS – addresses this threat by enforcing an access control policy and cryptographic measures that restrict access to backup data to authorized users, preventing its arbitrary modification.
- OE.PHYSICAL – addresses this threat by ensuring that unauthorized individuals do not have physical access to the TOE storage devices, preventing modification or substitution of the stored data through local interfaces.

T.TSF_FAILURE

The TSF may be compromised by a malicious user or process or corrupted through general degradation such that it operates in an unknown state.

This threat is countered by the following security objectives:

- O.SELF_TESTS – addresses this threat by detecting and responding to conditions that are representative of a TSF failure.
- OE.COMPONENTS_RUNNING – addresses this threat by providing environmental methods to detect TSF failures that cannot be reported or mitigated by the TOE, such as simultaneous loss of power to several nodes.
- OE.LIMITED – addresses this threat by minimizing the software interactions on the TOE hardware and reducing the likelihood that the TOE hardware or software is negatively affected by third-party applications.
- OE.REGULAR_UPDATES – addresses this threat by ensuring that flaw remedies published by the vendor are applied in a timely fashion so that any known sources of potential failure are mitigated quickly.

T.UNAUTH_ACCESS

A malicious individual may access restricted TSF functionality without authorization, either through obtaining valid credentials that they previously lacked or by escalating their privileges in excess of what they were assigned.

This threat is countered by the following security objectives:

- O.I_AND_A – addresses this threat by requiring administrators to prove their identity prior to being granted the authorization to access restricted TSF functionality.
- O.PROTECTED_COMMS – addresses this threat by cryptographically securing administrative activities while in transit to prevent session hijacking.

- O.SECURITY_MANAGEMENT – addresses this threat by providing logical access enforcement mechanisms to restrict the authorization granted to different users.
- OE.LIMITED – addresses this threat by ensuring that the TOE does not run arbitrary third-party software that introduces any external interfaces that could be used to access restricted functionality, whether maliciously or unintentionally.

T.UNDETECTED_ACTIONS

A malicious or careless user may alter the behavior of the TSF to cause it to operate in an unknown state.

This threat is countered by the following security objectives:

- O.AUDIT – addresses this threat by ensuring that the TSF generates an audit record of security-relevant behavior that cannot be disabled or erased without attribution.
- OE.SYSTEM_TIME – addresses this threat by providing accurate time data for audit records so that forensic analysis of the TOE's audit records can be performed.
- OE.TRUSTED_ADMIN – addresses this threat by reducing the likelihood that any given administrator of the TOE is malicious or careless.

T.UNTRUSTED_COMMS

A malicious user or process may access security-relevant TSF data in transit through the use of unencrypted or poorly encrypted communications channels.

This threat is countered by the following security objectives:

- O.PROTECTED_COMMS – addresses this threat by providing trusted communications channels that use valid cryptographic algorithm implementations to secure data in transit between the TOE and its operational environment.
- OE.SYSTEM_TIME – addresses this threat by using time data to ensure that trusted channels are established and maintained in accordance with the standards that they conform to.

A.COMPONENTS_RUNNING

It is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack or failure of one or more of the TOE components.

This assumption is satisfied by the following security objective:

- OE.COMPONENTS_RUNNING – this objective satisfies the assumption by ensuring that the TOE's operational environment includes methods of checking the availability of TOE components.

A.LIMITED

It is assumed that the hardware components that comprise the TOE are used only for the functionality provided by the TSF and that the TOE does not include any other general-purpose computing capabilities that present additional external interfaces to the TSF.

This assumption is satisfied by the following security objective:

- OE.LIMITED – this objective satisfies the assumption by ensuring that administrators do not introduce general-purpose computing applications to the operating systems on the TOE devices.

A.PHYSICAL

It is assumed that the TOE is deployed in a location that is physically secured in its operational environment and not subject to any attacks on the physical interfaces of the TOE or the TOE hardware itself.

This assumption is satisfied by the following security objective:

- OE.PHYSICAL – this objective satisfies the assumption by ensuring that all components of the TOE are located in physically protected areas.

A.REGULAR_UPDATES

It is assumed that TOE software/firmware updates are applied on a regular schedule and/or within a reasonable period of time after they have been made available by the vendor.

This assumption is satisfied by the following security objective:

- OE.REGULAR_UPDATES – this objective satisfies the assumption by ensuring that administrators regularly apply TOE updates when they are made available by the vendor.

A.SYSTEM_TIME

The TOE's operational environment is assumed to provide reliable system time for all nodes.

This assumption is satisfied by the following security objective:

- OE.SYSTEM_TIME – this objective satisfies the assumption by ensuring that the distributed TOE components have a method to maintain synchronized time with each other and with other components in the operational environment.

A.TRUSTED_ADMIN

It is assumed that any administrators of the TOE are trusted to be technically competent, non-malicious, and to follow operational and preparatory guidance as directed for the functions that they are authorized to perform.

This assumption is satisfied by the following security objective:

- OE.TRUSTED_ADMIN – this objective satisfies the assumption by ensuring that a process is followed to ensure that administrators are non-malicious and technically capable of executing their administrative responsibilities.

7.2 Security Functional Requirements Rationale

All security functional requirements identified in this Security Target are fully addressed in this section and each is mapped to the objective it is intended to satisfy. Table 9 summarizes the correspondence of functional requirements to TOE security objectives.

	O.AUDIT	O.AVAILABILITY	O.I_AND_A	O.PROTECTED_BACKUPS	O.PROTECTED_COMMS	O.SECURITY_MANAGEMENT	O.SELF_TESTS
FAU_GEN.1	X						
FAU_GEN.2	X						
FAU_STG.1	X						
FCS_CKM.1					X		
FCS_CKM.4					X		
FCS_COP.1(1)				X	X		
FCS_COP.1(2)					X		
FCS_COP.1(3)					X		
FCS_COP.1(4)					X		
FCS_HTTPS_EXT.1					X		

	O.AUDIT	O.AVAILABILITY	O.I_AND_A	O.PROTECTED_BACKUPS	O.PROTECTED_COMMS	O.SECURITY_MANAGEMENT	O.SELF_TESTS
FCS_RBG_EXT.1				X	X		
FCS_SSH_EXT.1			X		X		
FCS_TLS_EXT.1					X		
FDP_ACC.1		X		X			
FDP_ACF.1		X		X			
FDP_ETC.1(1)		X		X			
FDP_ETC.1(2)		X		X			
FDP_IMM_EXT.1		X		X			
FDP_ITC.1		X		X			
FDP_SAR_EXT.1						X	
FDP_SDC_EXT.1				X			
FIA_UAU.2			X				
FIA_UID.2			X				
FMT_MOF.1						X	
FMT_MTD.1						X	
FMT_SMF.1						X	
FMT_SMR.2						X	
FPT_FLS.1		X					X
FPT_TST_EXT.1							X
FRU_FLT.1		X					X
FTP_ITC.1					X		
FTP_TRP.1					X		

Table 9: Objectives to Requirement Correspondence

O.AUDIT

The TSF generates audit records of security-relevant events.

The following security functional requirements contribute to satisfying this security objective:

- FAU_GEN.1 – the ST includes FAU_GEN.1 to specify the security-relevant events that result in the generation of audit records and the security-relevant data that is included in those records.
- FAU_GEN.2 – the ST includes FAU_GEN.2 to associate security-relevant events with the users that caused those events to occur.
- FAU_STG.1 – the ST includes FAU_STG.1 to protect stored audit records from tampering.

O.AVAILABILITY

The TSF provides a physical and logical organization of backup data that ensures retention of the data in the event of a TOE component failure or outage.

The following security functional requirements contribute to satisfying this security objective:

- FDP_ACC.1 – the ST includes FDP_ACC.1 to define a policy for the handling of backup data, to include how it is protected from unauthorized access and what node(s) it should be replicated to as protection against component failure.
- FDP_ACF.1 – the ST includes FDP_ACF.1 to list the policy rules that make up the policy defined by FDP_ACC.1.
- FDP_ETC.1(1) – the ST includes FDP_ETC.1(1) to specify how data availability can be maintained through archival to cloud storage in the TOE's operational environment.
- FDP_ETC.1(2) – the ST includes FDP_ETC.1(2) to specify how backup data is restored back to its points of origin in the TOE's operational environment.
- FDP_IMM_EXT.1 – the ST includes FDP_IMM_EXT.1 to prevent erasure of stored backup data.
- FDP_ITC.1 – the ST includes FDP_ITC.1 to specify how data is preserved when it is backed up from its points of origin in the TOE's operational environment.
- FPT_FLS.1 – the ST includes FPT_FLS.1 to ensure that the TSF preserves a secure state (i.e., preservation of backup data) in the event of node failures or outages.
- FRU_FLT.1 – the ST includes FRU_FLT.1 to ensure that the retention of backup data is maintained in the event of node failures or outages.

O.I_AND_A

All TOE users are identified and authenticated by the TSF before access to security-relevant functionality is granted.

The following security functional requirements contribute to satisfying this security objective:

- FCS_SSH_EXT.1 – the ST includes FCS_SSH_EXT.1 to ensure that user authentication occurs as part of establishing SSH trusted communications with the TOE.
- FIA_UAU.2 – the ST includes FIA_UAU.2 to ensure that only authenticated users can interact with the TSF.
- FIA_UID.2 – the ST includes FIA_UID.2 to ensure that only identified users can interact with the TSF.

O.PROTECTED_BACKUPS

Backup data at rest is protected by the TSF from unauthorized disclosure, modification, and erasure.

The following security functional requirements contribute to satisfying this security objective:

- FCS_COP.1(1) – the ST includes FCS_COP.1(1) to ensure that stored backup data is encrypted and decrypted using proper methods.
- FCS_RBG_EXT.1 – the ST includes FCS_RBG_EXT.1 to ensure that data encryption keys are generated with sufficient randomness to provide the expected strength.
- FDP_ACC.1 – the ST includes FDP_ACC.1 to define a policy to define how backup data is imported, exported, and interacted with while it resides within the TOE boundary.
- FDP_ACF.1 – the ST includes FDP_ACF.1 to list the policy rules that make up the policy defined by FDP_ACC.1.
- FDP_ETC.1(1) – the ST includes FDP_ETC.1(1) to specify when, where, and how backup data is transmitted to cloud storage repositories in the TOE's operational environment.
- FDP_ETC.1(2) – the ST includes FDP_ETC.1(1) to specify when, where, and how backup data is restored back to its points of origin in the TOE's operational environment.
- FDP_IMM_EXT.1 – the ST includes FDP_IMM_EXT.1 to prevent modification of stored backup data.
- FDP_ITC.1 – the ST includes FDP_ITC.1 to specify how when, where, and how backup data is moved from its points of origin in the TOE's operational environment to various storage locations in the TOE.

- FDP_SDC_EXT.1 – the ST includes FDP_SDC_EXT.1 to specify how the confidentiality of stored backup data is maintained while it resides inside the TOE boundary.

O.PROTECTED_COMMS

The TSF protects communications between itself and external entities using trusted channel protocols and appropriate cryptographic functions.

The following security functional requirements contribute to satisfying this security objective:

- FCS_CKM.1 – the ST includes FCS_CKM.1 to ensure that cryptographic keys used in protected communications are generated appropriately.
- FCS_CKM.4 – the ST includes FCS_CKM.1 to ensure that cryptographic keys used in protected communications are destroyed securely when no longer used.
- FCS_COP.1(1) – the ST includes FCS_COP.1(1) to ensure that proper symmetric encryption is used in protected communications.
- FCS_COP.1(2) – the ST includes FCS_COP.1(2) to ensure that proper asymmetric encryption is used in protected communications.
- FCS_COP.1(3) – the ST includes FCS_COP.1(3) to ensure that proper cryptographic hashing is used in protected communications.
- FCS_COP.1(4) – the ST includes FCS_COP.1(2) to ensure that proper message authentication is used in protected communications.
- FCS_HTTPS_EXT.1 – the ST includes FCS_HTTPS_EXT.1 to ensure that HTTPS can be used for protected communications.
- FCS_RBG_EXT.1 – the ST includes FCS_RBG_EXT.1 to ensure that keys and key material used to establish trusted communications are generated with sufficient randomness to ensure the intended security strength is provided.
- FCS_SSH_EXT.1 – the ST includes FCS_SSH_EXT.1 to ensure that SSH can be used for protected communications.
- FCS_TLS_EXT.1 – the ST includes FCS_TLS_EXT.1 to ensure that TLS can be used for protected communications.
- FTP_ITC.1 – the ST includes FTP_ITC.1 to specify how protected communications are used to secure TSF data in transit between the TOE and its operational environment.
- FTP_TRP.1 – the ST includes FTP_TRP.1 to specify how protected communications are used to secure administrator interactions with the TOE.

O.SECURITY_MANAGEMENT

The TSF restricts the ability to perform security management functions on the TOE to authorized users having appropriate roles.

The following security functional requirements contribute to satisfying this security objective:

- FDP_SAR_EXT.1 – the ST includes FDP_SAR_EXT.1 to define the extent to which different users are authorized to interact with stored backup data and any associated metadata.
- FMT_MOF.1 – the ST includes FMT_MOF.1 to define the extent to which different users can manage the TOE's functional behavior.
- FMT_MTD.1 – the ST includes FMT_MTD.1 to define the extent to which different users can manage the TOE's stored security-relevant data.
- FMT_SMF.1 – the ST includes FMT_SMF.1 to define the management functions that can be performed on the TOE.

- FMT_SMR.2 – the ST includes FMT_SMR.2 to define the roles that can be associated with users of the TOE.

O.SELF_TESTS

The TSF will provide a mechanism to detect failure of its own functionality and provide notification of this to administrators.

The following security functional requirements contribute to satisfying this security objective:

- FPT_FLS.1 – the ST includes FPT_FLS.1 to ensure that the TSF maintains a secure state in the event of self-test failures.
- FPT_TST_EXT.1 – the ST includes FPT_TST_EXT.1 to ensure that the TSF performs self-testing of its own functionality.
- FRU_FLT.1 – the ST includes FRU_FLT.1 to ensure that backup data remains available in the event of a failure or outage of a distributed node.

7.3 Security Assurance Requirements Rationale

EAL 2 augmented with ALC_FLR.1 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security. ALC_FLR.1 was selected to exceed EAL2 assurance objectives in order to ensure that identified flaws are addressed. The TOE is intended for use in an environment with good physical access security where it is assumed that attackers will have Basic attack potential. The target assurance level of EAL 2 augmented with ALC_FLR.1 is appropriate for such an environment.

7.4 Requirement Dependency Rationale

The following table identifies the SFRs claimed in the ST, their dependencies as defined in CC Part 2, and how the dependency is satisfied in the ST. It can be seen that all dependencies have been satisfied by inclusion in the ST of the appropriate dependent SFRs or through expectations placed on the TOE's operational environment.

Requirement	Dependencies	How Satisfied
FAU_GEN.1	FPT_STM.1	OE.SYSTEM_TIME
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1 FIA_UID.2
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	[FCS_CKM.2, FCS_COP.1] FCS_CKM.4	FCS_COP.1[4] FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1, FDP_ITC.2, FCS_CKM.1]	FCS_CKM.1
FCS_COP.1(1)	[FDP_ITC.1, FDP_ITC.2, FCS_CKM.1] FCS_CKM.4	FCS_CKM.1 FCS_CKM.4
FCS_COP.1(2)	[FDP_ITC.1, FDP_ITC.2, FCS_CKM.1] FCS_CKM.4	N/A[1] FCS_CKM.4
FCS_COP.1(3)	[FDP_ITC.1, FDP_ITC.2, FCS_CKM.1] FCS_CKM.4	N/A[1] FCS_CKM.4
FCS_COP.1(4)	[FDP_ITC.1, FDP_ITC.2, FCS_CKM.1] FCS_CKM.4	N/A[1] FCS_CKM.4
FCS_HTTPS_EXT.1	FCS_COP.1 FCS_TLS_EXT.1	FCS_COP.1[4] FCS_TLS_EXT.1
FCS_RBG_EXT.1	None	None
FCS_SSH_EXT.1	FCS_COP.1 FCS_RBG_EXT.1 FPT_STM.1	FCS_COP.1[4] FCS_RBG_EXT.1 OE.SYSTEM_TIME
FCS_TLS_EXT.1	FCS_COP.1 FCS_RBG_EXT.1 FPT_STM.1	FCS_COP.1[4] FCS_RBG_EXT.1 OE.SYSTEM_TIME
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1

Requirement	Dependencies	How Satisfied
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1 N/A[2]
FDP_ETC.1(1)	[FDP_ACC.1, FDP_IFC.1]	FDP_ACC.1
FDP_ETC.1(2)	[FDP_ACC.1, FDP_IFC.1]	FDP_ACC.1
FDP_IMM_EXT.1	None	None
FDP_ITC.1	[FDP_ACC.1, FDP_IFC.1] FMT_MSA.3	FDP_ACC.1 N/A[3]
FDP_SAR_EXT.1	FDP_ITC.1	FDP_ITC.1
FDP_SDC_EXT.1	FCS_COP.1 FDP_ITC.1	FCS_COP.1[4] FDP_ITC.1
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UID.2	None	None
FMT_MOF.1	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_SMF.1	None	None
FMT_SMR.2	FIA_UID.1	FIA_UID.2
FPT_FLS.1	None	None
FPT_TST_EXT.1	None	None
FRU_FLT.1	FPT_FLS.1	FPT_FLS.1
FTP_ITC.1	None	None
FTP_TRP.1	None	None

Table 10: Requirement Dependencies

1: This cryptographic operation does not use any TSF-generated keys or imported user data.

2: The TOE's access control policy for backup data is not dependent on any default values for user security attributes.

3: The ability of the TOE to import (back up) data does not require any default security attributes to be associated with the data in order to inforce the access control policy for it.

4: This SFR is iterated in this ST. One or more iterations of this SFR is used to satisfy the dependency, not necessarily all iterations.

7.5 TOE Summary Specification Rationale

Section 6, the TOE Summary Specification, describes how the security functions of the TOE meet the claimed SFRs. The following table provides a mapping of the SFRs to the security function descriptions to support the TOE Summary Specification.

	Security Audit	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Protection of the TSF	Resource Utilization	Trusted Path/Channels
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_STG.1	X							
FCS_CKM.1		X						

	Security Audit	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Protection of the TSF	Resource Utilization	Trusted Path/Channels
FCS_CKM.4		X						
FCS_COP.1(1)		X						
FCS_COP.1(2)		X						
FCS_COP.1(3)		X						
FCS_COP.1(4)		X						
FCS_HTTPS_EXT.1		X						
FCS_RBG_EXT.1		X						
FCS_SSH_EXT.1		X						
FCS_TLS_EXT.1		X						
FDP_ACC.1			X					
FDP_ACF.1			X					
FDP_ETC.1(1)			X					
FDP_ETC.1(2)			X					
FDP_IMM_EXT.1			X					
FDP_ITC.1			X					
FDP_SAR_EXT.1			X					
FDP_SDC_EXT.1			X					
FIA_X509_EXT.1				X				
FIA_X509_EXT.2				X				
FIA_X509_EXT.3				X				
FIA_UAU.2				X				
FIA_UID.2				X				
FMT_MOF.1					X			
FMT_MTD.1					X			
FMT_SMF.1					X			
FMT_SMR.2					X			
FPT_FLS.1						X		
FPT_TST_EXT.1						X		
FRU_FLT.1							X	
FTP_ITC.1								X
FTP_TRP.1								X

Table 11: Security Functions vs. Requirements Mapping