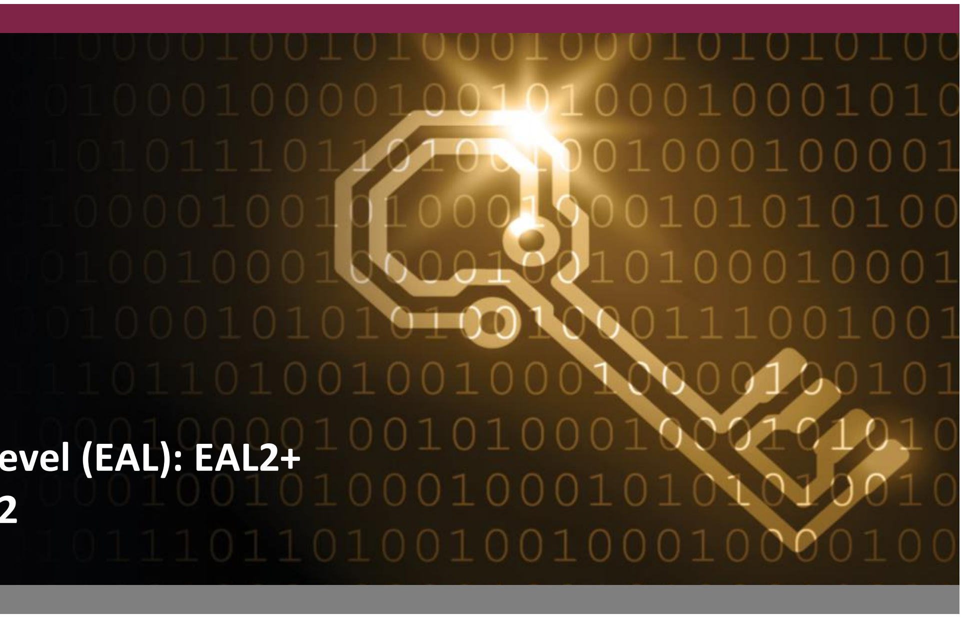
Qumulo, Inc.

Qumulo Core

v5

Security Target



Evaluation Assurance Level (EAL): EAL2+
Document Version: 0.22

Prepared for:

Qumulo, Inc.
1501 4th Avenue
Suite 1600
Seattle, WA 98101
United States of America

Phone: +1 855 577 7544
www.qumulo.com

Prepared by:

Corsec Security, Inc.
12600 Fair Lakes Drive
Suite 210
Fairfax, VA 22003
United States of America

Phone: +1 703 267 6050
www.corsec.com

Classification: Public

Table of Contents

List of Figures

List of Tables

1. Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the ST organization. The Target of Evaluation (TOE) is Qumulo Core v5.1.1 developed by Qumulo, Inc. (Qumulo) and will hereafter be referred to as the TOE throughout this document. The TOE is a file data platform that runs in a redundant configuration for highly scalable file storage.

1.1 Purpose

This ST is divided into nine sections, as follows:

- Introduction (Section 1) – Provides a brief summary of the ST contents and describes the organization of other sections within this document. It also provides an overview of the TOE security functionality and describes the physical and logical scope for the TOE as well as the ST and TOE references.
- Conformance Claims (Section 2) – Provides the identification of any Common Criteria (CC), Protection Profile (PP), and Evaluation Assurance Level (EAL) package claims. It also identifies whether the ST contains extended security requirements.
- Security Problem (Section 3) – Describes the threats, organizational security policies, and assumptions that pertain to the TOE and its environment.
- Security Objectives (Section 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Extended Components (Section 5) – Identifies new components (extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs)) that are not included in CC Part 2 or CC Part 3.
- Security Requirements (Section 6) – Presents the SFRs and SARs to which the TOE adheres.
- TOE Summary Specification (Section 7) – Describes the security functions provided by the TOE that satisfy the SFRs and objectives.
- Rationale (Section 8) – Presents the rationale for the security objectives, requirements, and SFR dependencies as to their consistency, completeness, and suitability.
- Acronyms (Section 9) – Defines the acronyms used within this ST.

1.2 Security Target and TOE References

Table 1 below shows the ST and TOE references.

Table 1 – ST and TOE References

ST Title	*Qumulo, Inc. Qumulo Core v5 Security Target*
ST Version	Version 0.22
ST Author	Corsec Security, Inc.
ST Publication Date	2023-03-06
TOE Reference	Qumulo Core v5.1.1

1.3 Product Overview

The product is Qumulo Core v5.1.1. Qumulo Core is a file data platform that is highly scalable to store files of all sizes. Qumulo Core is run with multiple nodes to provide data redundancy and system resiliency along with file backup and recovery. It uses real-time analytics to provide instant visibility into data and users. Qumulo software assigns aggregation of real-time metadata to all data as they are ingested, giving users real-time insight into their system without performance degradation or long file system scans. The hybrid cloud file storage provides a single namespace, exposed across protocols, that provides a vast number of users centralized access to files whether the data is on-prem, multi-site, or in the cloud. This allows users to simplify storage management by symmetrically scaling capacity and performance, removing data silos to eliminate a tangle of multi-volume mounts, and scaling to billions of files across the data center and cloud.

Qumulo Core provides the ability to take snapshots of the current state of the file system or directory at a given point in time. It provides the ability to restore single files and whole directories with snapshots. Qumulo Core performs continuous replication across storage clusters in order to provide recovery in the event that 1-2 drives fail or 1 node fails. It also performs audit logging and can integrate its auditing features with standard monitoring systems, such as syslog.

The Product Overview provides a high-level description of the product that is the subject of the evaluation. The following section, TOE Overview, will provide the introduction to the parts of the overall product offering that are specifically being evaluated.

1.4 TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. This section provides a context for the TOE evaluation by identifying the TOE type, describing the TOE, and defining the specific evaluated configuration.

The TOE is a software-only TOE that provides functionality of a Data Protection product. It is comprised of four instances of the Qumulo Core software communicating in a clustered setup that each run on an instance of Ubuntu Linux. Each instance of the TOE and operating system (OS) runs on separate node hardware.

The TOE has the ability to generate audits for events pertaining to role management, user management, and file management. To review the audit events, the TOE sends logs to a syslog server. All audits contain the identity of the TOE user that performed the operation that caused an audit.

The TOE enforces an Access Control Security Functional Policy (SFP) to control the user's access to objects such as files, directories, shares, and nodes.

The TOE blocks authentication requests on the connection for 1 second every time incorrect credentials are entered. The TOE associates usernames, roles, and passwords to a TOE user. Passwords must be secure and meet the requirements stated in section 7.1.3. The TOE requires TOE users to be authenticated and identified before allowing any actions besides using the API calls listed in section 7.1.3 or `qq version` CLI command to check the version of the TOE. Passwords being entered into the Web UI are obscured by using bullets to hide the characters. The TOE also employs two methods of authentication: local authentication and directory-based authentication.

The TOE limits the ability to manage security attributes, authentication, user accounts, roles and audit logging to the Administrators Role. The Data-Administrators Role can perform file and share management as well as query users and groups, and the Observers Role can query the security attributes and other management settings. Only the Administrators Role can specify alternative initial values for all security attributes. The Data-Administrators Role can specify alternative initial values for file, share, and directory security attributes. The TOE has three predefined roles: Administrators Role, Data-Administrators Role, and Observers Role. Custom roles can be created to customize privileges for TOE users.

The TOE preserves a secure state and ensures full functionality is preserved when 1-2 drives fail or when 1 node fails. The TOE utilizes the operating system's time to provide reliable timestamps and allows TOE users to terminate their own session.

An HTTP over TLS connection is used for communication when using the Web UI, QQ CLI, and REST API. SMBv3 and FTPS are also used for communicating with the SMB and FTP interfaces, respectively. All of these connections are setup to protect the transmitted data from modification or disclosure. When TOE users communicate with the TOE, they must initiate the secure path to the TOE.

1.4.1 TOE Components

The TOE is the Qumulo Core software working in a redundant 4-instance setup. The following paragraphs provide a brief description of the Qumulo Core software.

The TOE is made of several different layers. The first layer is data access, which contains SMB, NFS, and FTP protocols. The SMB, NFS, and FTP protocols exist as independent and scalable resources on each node of a Qumulo cluster. TOE users see a single namespace which can expand in capacity and performance. This namespace can be accessed from any workstation or other computing device running Windows, Linux, or Mac OS. The authentication layer supports Active Directory connections. Qumulo data services integrate with these global identity systems as managed by customers, enabling access to be controlled across TOE users and data. Each data access protocol uses a common authentication layer to interact with data stored in the filesystem, enabling users to move between applications, operating systems, and environments. The TOE uses both stateful data access protocols, such as SMB, and stateless data access protocols, such as NFS.

The TOE contains the following feature management interfaces in the management and programmability layer: A Web UI, a REST API, and the QQ CLI. The Web UI is a graphical user interface (GUI) used to manage and configure the TOE. It can be accessed through a web browser using the IP address of the overall cluster or using the IP address of a node. The QQ CLI is the command line interface and offers most management functionality. It can be downloaded from the Web UI as a .zip file. The REST API is an application programming interface (API) used to configure all aspects of a system, such as user accounts, snapshot policies, data replication, and data management), gather information about the TOE, and read or write data. REST API can be accessed using an API client such as Postman. The TOE also contains the SMB, NFS, and FTP interfaces for file management within the designated shares.

SMB shares are used to share, read, and write files to a remote host over a local area network (LAN). The share displays files and directories the TOE user has permissions to access. If a TOE user does not have read permissions for a directory, then it is hidden from the TOE user's view. Files can be shared within the TOE using

NFS as well. NFS exports are used to export directories from an NFS server's local hard disk to an NFS client. The NFS client mounts the directory so it can be accessed like any directory. The directory can be mounted on multiple clients, allowing all of the clients to share files with each other using this directory. The TOE allows access to the NFS share based on the client's IP address, which is mapped to a local account in the TOE. FTP is a third method to share files. An FTP server offers access to a directory and its subdirectories. TOE users can connect to the FTP server with an FTP client to share files between hosts.

1.5 TOE Environment

The TOE relies on the operating environment to contain the Linux OS, node hardware, servers, networking equipment, and workstations.

The TOE runs on Ubuntu 18.04 and works best with servers that meet the following minimum requirements:

- Systems with all NVMe drives:
    - Storage: NVMe drives must support hot plug. At least 6 drives should be used.
    - Processer: One core per drive, fewer fast cores are better than more slow cores.
    - Memory: Minimum RAM should be 0.38GB per TB of drive space.
    - Network: Dual-100G or greater network interfaces.
    - Power: Redundant power supply units.
- Systems with SSDs and HDDs:
    - Storage: HDD-to-SSD ratio 3:1 or 4:1 for performance, or up to 6:1 for archiving. Drives must support hot plug. No shingled magnetic recording (SMR) drives. Minimum SSD space should be about 2.5% of HDD space.
    - Processer: One core per 2 HDDs, fewer fast cores are better than more slow cores.
    - Memory: Minimum RAM should be 0.38GB per TB of HDD space.
    - Network: Dual-100G or greater network interfaces for performance or dual-25G for archiving.
    - Power: Redundant power supply units.

For data communication between nodes, a 10G or greater switch is also required.

The TOE requires the following servers in its operating environment:

- A syslog server for uploading the generated audit logs to.
- An Active Directory server for directory-based authentication.
- An NTP server connected to the Linux OS to ensure the time is synchronized with the network.

A workstation is also required for accessing the file shares and managing the TOE through the TOE interfaces. The QQ CLI requires the installation of the local QQ CLI client, which can be downloaded from within the Web UI of the TOE. The REST API requires the installation of a 3rd-party application or customized program used to exercise the REST commands. The Web UI is compatible with Google Chrome 93.0.4577.63 64-bit and above.

Access to the Qumulo Core Knowledge Base is also provided to TOE users that is an online collection of articles on how to set up and manage the TOE. It is located at https://care.qumulo.com/hc/en-us/categories/115000637447-KNOWLEDGE-BASE.

All of the above resources are outside the boundary of the TOE and therefore a part of the TOE environment.

1.6 TOE Description

This section primarily addresses the physical and logical components of the TOE that are included in the evaluation.

1.6.1 Physical Scope

The TOE is a file data platform which runs on Ubuntu 18.04. The TOE runs in a 4-instance cluster on the nodes in environment. The software of the TOE in the evaluated configuration is described in section 1.4.1. The components required in the operational environment are listed in section 1.5. Figure 1 illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE.

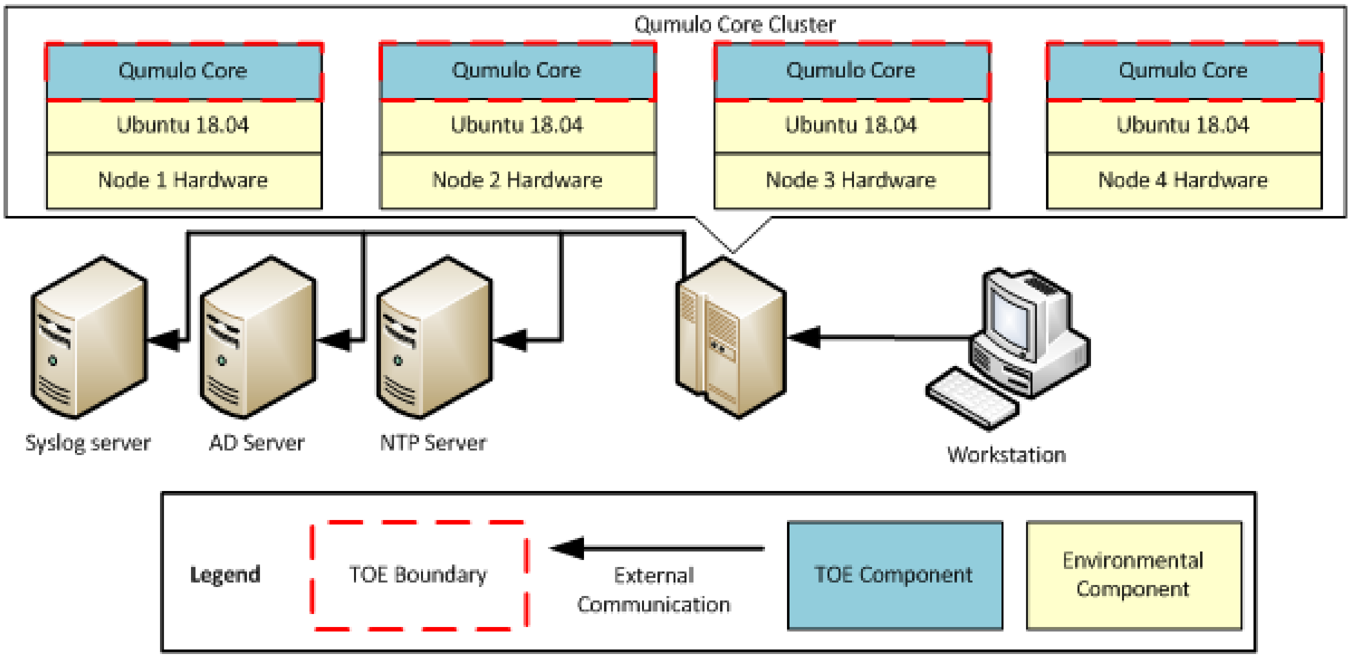


Figure 1 – Physical TOE Boundary

Note: The AD server, NTP server, and syslog server are components of the operational environment.

1.6.1.1 TOE Software

The TOE is a software-only TOE and is comprised of Qumulo Core v5.1.1, which is packaged in the .qimg file format. TOE users can download the image from the Qumulo Care website (https://care.qumulo.com/hc/en-us/community/topics/115000440768-PRODUCT-RELEASES). If node hardware is ordered for a new installation of the TOE, the image will be installed on the node hardware for delivery to the TOE users.

1.6.1.2 Guidance Documentation

Table 2 lists the PDF[1] formatted guides that are required reading and part of the TOE.

[1] PDF – Portable Document Format

Table 2 – Guidance Documentation

Document Name	Description
Qumulo, Inc. Qumulo Core v5 Guidance Documentation Supplement v0.11	Contains information regarding specific configuration for the TOE evaluated configuration.

1.6.2 Logical Scope

The logical boundary of the TOE will be broken down into the following security classes which are further described in sections 6 and 7 of this ST. The logical scope also provides the description of the security features of the TOE. The SFRs implemented by the TOE are usefully grouped under the following Security Function Classes:

- Security Audit
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- Resource Utilization
- TOE Access
- Trusted Path/Channel

1.6.2.1 Security Audit

The TOE generates audit records for startup and shutdown of audit functions, successful authentication, unsuccessful authentication, filesystem operations and management operations. The unique file ID in audit logs is generated by the OS when a file is created. The file ID does not depend on the file name. It is able to associate audit records with the TOE user that caused the audited event. The TOE relies on the Linux OS as a time source to provide reliable timestamps.

1.6.2.2 User Data Protection

The TOE enforces the Access Control SFP to provide access control on TOE users who are accessing nodes, shares, files, and directories. TOE users are given roles which determine their access to objects. Files and directories are written to a node based on file name and directory name.

1.6.2.3 Identification and Authentication

When a TOE user enters incorrect credentials, the TOE will block authentication requests on that connection for 1 second before allowing the TOE user to login again. The TOE maintains usernames, roles, and passwords as security attributes belonging to an individual user. All passwords must be secure and meet requirements detailed in section 7.1.3. Passwords are obscured by bullets when entered in the Web UI. The TOE allows the user to use the API calls listed in section 7.1.3 prior to authentication and identification as well as the `qq version` QQ CLI command to determine the version of the TOE. All other actions cannot be performed until the TOE user is properly authenticated and identified. The TOE uses two authentication mechanisms: local authentication and directory-based authentication.

1.6.2.4 Security Management

The TOE restricts all management functionality to the Administrators Role while allowing the Observers Role to query TOE settings. The Data-Administrators Role has access to all file and share management. The TOE enforces restrictive default values for all security attributes. There are three pre-defined roles maintained by the TOE: Administrators Role, Data-Administrators Role, and Observers Role. Additionally, the TOE allows for custom roles to be created.

1.6.2.5 Protection of the TSF

The TOE preserves a secure state when 1-2 drives fail or when 1 node fails. The TOE provides timestamps for audit logs and utilizes the Linux system time to get the accurate time.

1.6.2.6 Resource Utilization

The TOE ensures full functionality when either 1-2 drives fail or 1 node fails.

1.6.2.7 TOE Access

TOE users can manually terminate sessions in Web UI and QQ CLI. In Web UI this can be done by clicking “sign out”. In QQ CLI, entering the “logout” command will log the TOE user out.

1.6.2.8 Trusted Path/Channel

Communication with the REST API, Web UI, and QQ CLI interfaces are done over a secure channel using HTTP over TLS. Connections to the SMB interface is done over SMBv3 and connections to the FTP interface are secured with FTPS. Remote users must initiate the connection to the TOE.

1.6.3 Product Physical/Logical Features and Functionality not included in the TOE

Features and/or Functionality that are not part of the evaluated configuration of the TOE are:

- Using the local QQ CLI on a node.
- Cloud-based deployment.
- Multi-site deployment.
- Direct Kerberos authentication

2. Conformance Claims

This section and Table 3 provide the identification for any CC, PP, and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Section 8.1.

Table 3 – CC and PP Conformance

Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Release 5, April 2017; CC Part 2 conformant; CC Part 3 conformant; PP claim (none); Parts 2 and 3 Interpretations of the CEM as of July 27, 2021 were reviewed, and no interpretations apply to the claims made in this ST.
PP Identification	None
Evaluation Assurance Level	EAL2+ augmented (Augmented with Flaw Remediation (ALC_FLR.2))

3. Security Problem

This section describes the security aspects of the environment in which the TOE will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all:

- Known and presumed threats countered by either the TOE or by the security environment
- Organizational security policies to which the TOE must comply
- Assumptions about the secure usage of the TOE, including physical, personnel, and connectivity aspects

3.1 Threats to Security

This section identifies the threats to the IT assets against which protection is required by the TOE or by the security environment. The threat agents are divided into two categories:

- Attackers who are not TOE users: They have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings or parameters, and no physical access to the TOE.
- TOE users: They have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings or parameters, and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)

Both are assumed to have a low level of motivation. The IT assets requiring protection are the TSF[2] and user data saved on or transitioning through the TOE and the hosts on the protected network. Removal, diminution, and mitigation of the threats are through the objectives identified in Section 4 Security Objectives. Table 4 below lists the applicable threats.

Table 4 – Threats

Name	Description
T.DATA_CORRUPTION	TOE user data and configuration data could become corrupted due to hardware failure or incorrect system operations.
T.IMPROPER_SERVER	A TOE user or attacker could attempt to bypass the access controls provided by the TOE using one of the systems connected to the TOE.
T.MASQUERADE	A TOE user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.NO_AUDIT	A TOE user or attacker may perform security-relevant operations on the TOE without being held accountable for them.
T.TAMPERING	A TOE user or process may be able to bypass the TOE's security mechanisms by tampering with the TOE or TOE environment.
T.UNAUTH	A TOE user may gain access to security data on the TOE, even though the TOE user is not authorized in accordance with the TOE security policy.

[2] TSF – TOE Security Functionality

3.2 Organizational Security Policies

There are no Organizational Security Policies (OSPs) defined for this ST.

3.3 Assumptions

This section describes the security aspects of the intended environment for the evaluated TOE. The operational environment must be managed in accordance with assurance requirement documentation for delivery, operation, and user guidance. Table 5 lists the specific conditions that are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 5 – Assumptions

Name	Description
A.CONNECTIVITY	It is assumed that the TOE environment will be configured in such a way as to allow TOE users to access the information stored on the TOE.
A.INTERNAL_STORAGE_NETWORK	The network that the TOE uses for storage transfer is intended to be an internal private network that is protected from access by entities outside of the organization. External access to storage services are blocked by the TOE environment
A.INTERNAL_USERS	It is assumed that TOE users accessing the storage on the TOE reside on the internal network and are not careless, negligent, or willfully hostile with regard to their access of the TOE.
A.LOCATE	It is assumed that the TOE is located within a controlled access facility and is physically available to authorized TOE users only.
A.NOEVIL	It is assumed that the TOE users who manage the TOE (only within the internal private network) are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.

4. Security Objectives

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition (see Section 3). The set of security objectives for a TOE form a high-level solution to the security problem. This high-level solution is divided into two part-wise solutions: the security objectives for the TOE, and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment.

4.1 Security Objectives for the TOE

The specific security objectives for the TOE are listed in Table 6 below.

Table 6 – Security Objectives for the TOE

Name	Description
O.ADMIN	The TOE must include a set of functions that allow efficient management of its functions and data, ensuring that TOE users with the appropriate privileges and only those TOE users, may exercise such control.
O.AUDIT	The TOE must record events of security relevance. The TOE must record the resulting actions of the security functional policies and associate the identity of the user causing the audited event with the audit log.
O.AUTHENTICATE	The TOE must be able to identify and authenticate users prior to allowing access to TOE administrative functions and data.
O.FAULT_TOLERANCE	The TOE must be resilient against node or disk failures that might affect the security of the information it contains.
O.USER_DATA	The TOE must prevent unauthorized modifications to configuration and user data that it has been entrusted to protect.

4.2 Security Objectives for the Operational Environment

This section describes the environmental objectives.

4.2.1 IT Security Objectives

Table 7 below lists the IT security objectives that are to be satisfied by the environment.

Table 7 – IT Security Objectives

Name	Description
OE.CONNECT	TOE users will configure the TOE environment so that they have the proper network support to be able to access data on the TOE.
OE.INTERNAL_STORAGE_NETWORK	The TOE environment must limit access to the TOE from external entities such that only internal hosts can access the storage functionality provided by the TOE.
OE.SECURE_COMMUNICATION	The TOE environment must provide un-tampered communications between systems connected to the TOE and between TOE components for any connections that the TOE is not already protecting.

4.2.2 Non-IT Security Objectives

Table 8 below lists the non-IT environment security objectives that are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 8 – Non-IT Security Objectives

Name	Description
NOE.INTERNAL_USERS	Sites using the TOE shall ensure that internal TOE users are not careless, negligent, or willfully hostile.
NOE.NOEVIL	Sites using the TOE shall ensure that TOE users are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.
NOE.PHYSICAL	The TOE will be used in a physically secure site that protects it from interference and tampering by untrusted subjects.

5. Extended Components

This section defines the extended SFRs and extended SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

5.1 Extended TOE Security Functional Components

There are no extended SFRs defined for this ST.

5.2 Extended TOE Security Assurance Components

There are no extended SARs defined for this ST.

6. Security Requirements

This section defines the SFRs and SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

6.1 Conventions

There are several font variations used within this ST. Selected presentation choices are discussed here to aid the Security Target reader.

The CC allows for assignment, refinement, selection and iteration operations to be performed on security functional requirements. All of these operations are used within this ST. These operations are performed as described in Part 2 of the CC and are shown as follows:

- Completed assignment statements are identified using [*italicized text within brackets*].
- Completed selection statements are identified using [underlined text within brackets].
- Completed assignments within selection statements are identified using [*underlined and italicized text within brackets*].

6.2 Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class. Table 9 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 9 – TOE Security Functional Requirements

Name	*Description*	*S*	*A*	*R*	*I*
FAU_GEN.1	Audit Data Generation	✓	✓		
FAU_GEN.2	User Identity Association				
FDP_ACC.1	Subset access control		✓		
FDP_ACF.1	Security attribute based access control		✓		
FIA_AFL.1	Authentication failure handling	✓	✓		
FIA_ATD.1	User attribute definition		✓		
FIA_SOS.1	Verification of secrets		✓		
FIA_UAU.1	Timing of authentication		✓		
FIA_UAU.5	Multiple authentication mechanisms		✓		
FIA_UAU.7	Protected authentication feedback		✓		
FIA_UID.1	Timing of identification		✓		
FMT_MOF.1	Management of security functions behaviour	✓	✓		
FMT_MSA.1	Management of security attributes	✓	✓		
FMT_MSA.3	Static attribute initialisation	✓	✓		

Name	*Description*	*S*	*A*	*R*	*I*
FMT_MTD.1	Management of TSF data	✓	✓		
FMT_SMF.1	Specification of management functions		✓		
FMT_SMR.1	Security roles		✓		
FPT_FLS.1	Failure with preservation of secure state		✓		
FPT_STM.1	Reliable time stamps				
FRU_FLT.2	Limited fault tolerance		✓		
FTA_SSL.4	User-initiated termination				
FTP_TRP.1	Trusted path	✓	✓		

Note: S=Selection; A=Assignment; R=Refinement; I=Iteration

6.2.1 Class FAU: Security Audit

FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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. [*successful authentication, unsuccessful authentication, filesystem operations, authentication management, user account management, role management, audit logging management, file management*].

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 (if applicable), 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, [*the following:*

- *User IP*
- *User ID*
- *Logins*
- *Protocol*
- *File System or Management Operation*
- *Error Status*
- *File ID*
- *File Path*
- *Secondary file path*].

FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

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.

6.2.2 Class FDP: User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1

The TSF shall enforce the [A*ccess Control SFP*] on [the following:

- *Subjects – Users*
- *Objects – Nodes, files, directories, shares*
- *Operations – Read, write, create, delete*

].

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.
Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1

The TSF shall enforce the [*Access Control SFP*] to objects based on the following: [

- *Subjects*
  - *Users – Username, password, role, NFS whitelist entry, SMB allow list entry, SMB deny list entry, groups*
- *Objects*
  - *Nodes – Node name*
  - *Files – File name, file ID*
  - *Directories – Directory name*
  - *Shares – share directory*

].

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 are granted access to nodes and share management based on roles*
- *TOE users are granted access to files, directories, and shares for file management based on groups*].

FDP_ACF.1.3

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*TOE users can access nodes, files, directories, and shares if their information is associated an entry on the NFS whitelist or SMB allow list*]

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the [*TOE users are denied access to nodes, files, directories, and shares when attempting to access objects from a subject that has an entry on the SMB deny list*].

6.2.3 Class FIA: Identification and Authentication

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1

The TSF shall detect when [1] unsuccessful authentication attempts occur related to [*authenticating to the Web UI, QQ CLI, or REST API*].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been [met], the TSF shall [*block authentication requests on that connection for 1 second*].

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: [*username, role, password*]

FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_SOS.1.1

The TSF shall provide a mechanism to verify that secrets meet [*the following requirements:*

- *Minimum of 8 characters*
- *Maximum of 128 characters*
- *At least 3 of the following:*
  - *Lowercase letters*
  - *Uppercase letters*
  - *Numbers*
  - *Special characters*

].

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1

The TSF shall allow [*the qq version CLI command to check the version, the qq login CLI command to login, the Web UI to be accessed at https://<Node IP address>/login, the following API calls:*

- *GET /v1/version*

- *GET /v1/cluster/settings*
- *GET https://<Node IP address>:8000/v1/session/login*

] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2

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

FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_UAU.5.1

The TSF shall provide [*the following authentication mechanisms:*

- *Local authentication mechanisms*
- *Directory-based authentication mechanisms*

] to support user authentication.

FIA_UAU.5.2

The TSF shall authenticate any user's claimed identity according to the [*the following rules:*

- *Local authentication – The user navigates to where the TOE is installed and logs in by entering their username and password. The username and password are verified against the username and password stored by the TOE. If the username and password match, the user is granted access to the TOE.*
- *Directory-based authentication – The user navigates to where the TOE is installed and enters their domain account's credentials. The TOE forwards the credentials to the AD server. The AD server evaluates the credentials, and if the username corresponds to a valid domain user and the password matches the stored password, the AD server sends a successful message back to the TOE. The account's AD groups are queried to assign the correct privileges, and the TOE user is allowed access to the TOE.*
- *The TOE determines which method of authentication to use based on the username that is passed to it in the authentication request. If a username is passed with a domain, either domain\username or username@domain, the TOE authenticates the account against the AD server. Otherwise, it authenticates against the local database of users.*].

FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UAU.7.1

The TSF shall provide only [*bullets*] to the user while the authentication is in progress.

Application note: FIA_UAU.7 only applies to Web UI. The REST API and QQ CLI rely on clients installed on the workstation to obscure their passwords.

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_UID.1.1

The TSF shall allow [*the qq version CLI command to check the version, the qq login CLI command to login, the Web UI to be accessed at https://<Node IP address>/login, the following API calls:*

- *GET /v1/version*
- *GET /v1/cluster/settings*
- *GET https://<Node IP address>:8000/v1/session/login*

] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2

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

6.2.4 Class FMT: Security Management

FMT_MOF.1 Management of security functions behavior

Hierarchical to: No other components.

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MOF.1.1

The TSF shall restrict the ability to [determine the behavior of, modify the behavior of] the functions [*listed in the Security Functions column of Table 10*] to [*roles listed in the Roles column of Table 10*].

Table 10 – Management of Security Functions Behavior

Roles	Security Functions	Permissions
Administrators Role	Authentication management	Determine the behavior of, modify the behavior of
Observers Role		Determine the behavior of
Administrators Role	User account management	Determine the behavior of, modify the behavior of
Observers Role, Data-Administrators Role		Determine the behavior of
Administrators Role	Role management	Determine the behavior of, modify the behavior of
Observers Role		Determine the behavior of
Administrators Role	Audit logging management	Determine the behavior of, modify the behavior of
Observers Role		Determine the behavior of
Administrators Role, Data-Administrators Role	File management	Determine the behavior of, modify the behavior of
Observers Role		Determine the behavior of

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control]
FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

FMT_MSA.1.1

The TSF shall enforce the [*Access Control SFP*] to restrict the ability to [*permissions in the Permissions column of Table 11*] the security attributes [*security attributes in the Security Attributes column of Table 11*] to [*roles listed in the Roles column of Table 11*].

Table 11 – Management of Security Attributes

Security Attributes	Permissions	Roles
Username	Change_default, query, modify	Administrators Role
	Query	Data-Administrators Role, Observers Role
Password	Change_default, modify	Administrators Role
Role	Change_default, query, modify, delete	Administrators Role
	Query	Observers Role
Node name	Change_default, query, modify	Administrators Role
	Query	Observers Role
Node IP address	Change_default, query, modify	Administrators Role
	Query	Observers Role
File name	Change_default, query, modify, delete	Administrators Role, Data-Administrators Role
	Query	Observers Role
Directory name	Change_default, query, modify, delete	Administrators Role, Data-Administrators Role
	Query	Observers Role
SMB allow list entry	Change_default, query, modify, delete	Administrators Role, Data-Administrators Role
	Query	Observers Role
SMB deny list entry	Change_default, query, modify, delete	Administrators Role, Data-Administrators Role
	Query	Observers Role
NFS IP whitelist entry	Change_default, query, modify, delete	Administrators Role, Data-Administrators Role
	Query	Observers Role

FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1

The TSF shall enforce the [*Access Control SFP*] to provide [*restrictive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the [*Administrators Role, Data-Administrators Role*] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MTD.1.1

The TSF shall restrict the ability to [*permissions listed in the Permissions column of Table 12*] the [*TSF data listed in the TSF Data column of Table 12*] to [*roles listed in the Roles column of Table 12*].

Table 12 – Management of TSF data

TSF Data	Permissions	Roles
Manage audit logging	Change_default, query, modify	Administrators Role
	Query	Observers Role
Manage files	Change_default, query, modify, delete	Administrators Role, Data-Administrators Role
	Query	Observers Role
Making shares	Change_default, query, modify, delete	Administrators Role, Data-Administrators Role
	Query	Observers Role
Manage user accounts	Change_default, query, modify, delete	Administrators Role
	Query	Observers Role, Data-Administrators Role

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No Dependencies

FMT_SMF.1.1

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

- *Manage audit logging*
- *Manage files*
- *Making shares*
- *Manage user accounts*

].

FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1

The TSF shall maintain the roles [*Administrators Role, Data-Administrators Role, Observers Role, and any custom role*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

6.2.5 Class FPT: Protection of the TSF

FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1

The TSF shall preserve a secure state when the following types of failures occur: [*1-2 drive failure, 1 node failure*].

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps.

6.2.6 Class FRU: Resource Utilization

FRU_FLT.2 Limited fault tolerance

Hierarchical to: FRU_FLT.1 Degraded fault tolerance

Dependencies: FPT_FLS.1 Failure with preservation of secure state

FRU_FLT.2.1

The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: [*1-2 drive failure, 1 node failure*].

6.2.7 Class FTA: TOE Access

FTA_SSL.4 User-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies

FTA_SSL.4.1

The TSF shall allow user-initiated termination of the user's own interactive session.

6.2.8 Class FTP: Trusted Path/Channels

FTP_TRP.1 Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies

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].

6.3 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3 and are EAL2+ augmented with ALC_FLR.2. Table 13 summarizes these requirements.

Table 13 – Assurance Requirements

Assurance Requirements	
Class 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
Class 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.2 Flaw reporting procedures
Class ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic design
Class AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Class ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – Sample
Class AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

7. TOE Summary Specification

This section presents information to detail how the TOE meets the functional requirements described in previous sections of this ST.

7.1 TOE Security Functionality

Each of the security requirements and the associated descriptions correspond to a security functionality. Hence, each security functionality is described by how it specifically satisfies each of its related requirements. This serves to both describe the security functionality and rationalize that the security functionality satisfies the necessary requirements. Table 14 lists the security functionality and their associated SFRs.

Table 14 – Mapping of TOE Security Functionality to Security Functional Requirements

TOE Security Functionality	SFR ID	Description
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
Identification and Authentication	FIA_AFL.1	Authentication failure handling
	FIA_ATD.1	User attribute definition
	FIA_SOS.1	Verification of secrets
	FIA_UAU.1	Timing of authentication
	FIA_UAU.5	Multiple authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.1	Timing of identification
Security Management	FMT_MOF.1	Management of security functions behaviour
	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialisation
	FMT_MTD.1	Management of TSF data
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of TOE Security Functionality	FPT_FLS.1	Failure with preservation of secure state
	FPT_STM.1	Reliable time stamps
Resource Utilization	FRU_FLT.2	Limited fault tolerance
TOE Access	FTA_SSL.4	User-initiated termination
Trusted Path/Channels	FTP_TRP.1	Trusted path

7.1.1 Security Audit

The TOE generates audit records for start-up and shutdown of audit functions, successful authentication, unsuccessful authentication, filesystem operations, authentication management, user account management, role management, audit logging management, and file management. The file ID included in an audit log is a unique 64-bit identifier generated by the OS when the file is created. The TOE is able to associate each auditable event with the identity of the user that caused the event. the TOE attaches timestamps to audit logs and relies on an NTP server to get the accurate time. The TOE does not provide a way to read or view audit logs, but a syslog server can be used to read and view them.

The TOE audit records contain information listed in Table 15.

Table 15 – Audit Record Contents

Field	Content
User IP	The IP address of the TOE user in IPv4/IPv6 format
User ID	The TOE user that performed the action
Logins	Any successful or unsuccessful login attempt by the TOE user
Protocol	The protocol that initiated the operation.
File System or Management Operation	Filesystem or management operation performed.
Error status	“ok” if the operation succeeded or a Qumulo specific error status code if the operation failed.
File ID	Operation-specific ID.
File Path	Primary operation-specific name.
Secondary file path	Secondary operation-specific name.

TOE Security Functional Requirements Satisfied: FAU_GEN.1, FAU_GEN.2

7.1.2 User Data Protection

In order to control access, the TSF enforces the Access Control SFP on the following:

- Subjects – Users
- Objects – Nodes, files, directories, shares
- Operations – Read, write, create, delete

Access to objects is controlled via the following security attributes:

- Subjects
  - Users – Username, password, role, NFS whitelist entry, SMB allow list entry, SMB deny list entry, groups
- Objects
  - Nodes – node name
  - Files – file name, file ID
  - Directories – directory name

  - Shares – share directory

TOE users are granted access to nodes and share management based on roles. TOE users are granted access to files, directories, and shares for file management based on groups.

TOE users are explicitly allowed access to nodes, files, directories, and shares if their information is associated an entry on the NFS whitelist or SMB allow list. TOE users are explicitly denied access to nodes, files, directories, and shares when attempting to access objects from a subject that has an entry on the SMB deny list.

TOE Security Functional Requirements Satisfied: FDP_ACC.1, FDP_ACF.1.

7.1.3 Identification and Authentication

The TOE detects when a TOE user attempts and fails to log in. Every time incorrect credentials are entered, the TOE blocks authentication requests on the connection to the Web UI, QQ CLI and REST API for 1 second before allowing the TOE user to attempt to login again. The TOE maintains usernames, roles, and passwords as security attributes belonging to individual TOE users. Additionally, it has a password complexity requirement. All passwords must meet the following complexity requirements:

- Minimum of 8 characters
- Maximum of 128 characters
- At least 3 of the following:
  - Lowercase letters
  - Uppercase letters
  - Numbers
  - Special characters

The TOE allows TOE users to use the following API calls before they are authenticated and identified:

- GET /v1/version
- GET /v1/cluster/settings
- GET https://<Node IP address>:8000/v1/session/login

The TOE also allows the `qq version` and `qq login` QQ CLI commands before authentication and identification. The Web UI login screen can also be accessed at https://<Node IP address>/login before authentication and identification. All other operations require users to be authenticated and identified. The TOE utilizes multiple authentication mechanisms:

- Local authentication – The TOE user navigates to where the TOE is installed and logs in by entering their username and password. The username and password are verified against the username and password stored by the TOE. If the username and password match, the user is granted access to the TOE. The TOE uses this method of authentication if the TOE user uses a username stored in the local database to authenticate.
- Directory-based authentication – The user navigates to where the TOE is installed and enters their domain account's credentials. The TOE forwards the credentials to the AD server. The AD server evaluates the credentials, and if the username corresponds to a valid domain user and the password

matches the stored password, the AD server sends a successful message back to the TOE. The account's AD groups are queried to assign the correct privileges, and the TOE user is allowed access to the TOE. The TOE uses this method of authentication if the TOE user enters the domain username instead of a local username.

- The TOE determines which method of authentication to use based on the username that is passed to it in the authentication request. If a username is passed with a domain, either domain\username or username@domain, the TOE authenticates the account against the AD server. Otherwise, it authenticates against the local database of users.

The TOE obscures passwords entered into the Web UI in the form of bullets.

TOE Security Functional Requirements Satisfied: FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.1, FIA_UAU.5, FIA_UAU.7, FIA_UID.1.

7.1.4 Security Management

The TOE restricts the ability to determine and modify the behavior of the security functions, permissions and roles listed in Table 10. The TOE provides TOE users assigned the Administrators Role with the ability to manage the following security functions:

- Authentication management – TOE users can set up AD servers to use directory-based authentication.
- User account management – New user accounts can be created and can be assigned roles. Passwords can be changed.
- Role management – The TOE can assign pre-defined roles to TOE users as well as create custom roles.
- Audit logging management – TOE users can set up a syslog server to view audit messages.
- File management – TOE users can create and delete files and directories.

The Access Control SFP enforce restrictive default values for security attributes listed in Table 11. TOE users assigned the roles listed in Table 11 have the ability to specify alternative initial values to override the default values when an object is created. TOE users assigned the Observers Role can read the values of the security attributes.

The TOE restricts the following management functions to TOE users assigned the roles in Table 12:

- Manage audit logging – TOE users can set up a syslog server to view audit logs generated by the TOE.
- Manage files – Files and directories can be created, renamed, or deleted.
- Making shares – NFS, SMB and FTP file shares can be created to share files and existing shares can be deleted.
- Manage user accounts – New TOE users can be created, and existing TOE users can be assigned roles. Passwords can be changed.

The TOE maintains and associates TOE users to the following roles:

- Administrators Role – TOE users with this role have access to all management functionality, including the Web UI.

- Data-Administrators Role – TOE users with this role have access to all file and share management functionality. This role does not have access to the Web UI.
- Observers Role – This is a read-only role. TOE users with this role can access the Web UI and read-only REST APIs but cannot perform any management functions.
- Custom role – Custom roles can be created to customize privileges for TOE users.

TOE Security Functional Requirements Satisfied: FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, FMT_SMR.1.

7.1.5 Protection of the TSF

The TOE preserves a secure state when 1-2 drives fail or 1 node fails. There are different recovery scenarios depending on the type of failure:

- 1 or 2 drive failures – Data is protected and balanced. The failed drive can be replaced at any time. The cluster will undergo complete reprotection and rebalance.
- 1 node offline – Data is protected.

Upon a 1 to 2 drive failure, the TOE begins a brief process called recovery where in-flight transactions are rolled forward or back. After recovery, TOE users may access any data that is on the TOE. The TOE also immediately begins rebuilding the impacted data in reserved space. This process is called reprotection. Once the failed drive has been replaced with a new drive, the system takes the data in the reserved space and re-balances across the cluster.

TOE storage is divided into individual protected stores (pstores). Data within a pstore is further divided into block stores (bstores). Each bstore in the pstore is located on a different disk. No more than two bstores can be located on the same node. This ensures that data is still available to TOE users if a node fails.

Full security functionality is available even if a node or drive failure occurs.

The TOE provides timestamps that are attached to audit logs. The TOE queries an NTP server in the TOE environment in order to get the accurate time.

TOE Security Functional Requirements Satisfied: FPT_FLS.1, FPT_STM.1.

7.1.6 Resource Utilization

The TOE ensures the operation of all of its capabilities when 1-2 drives fail or 1 node fails. There are different recovery scenarios depending on the type of failure:

- 1 or 2 drive failures – Data is protected and balanced. The failed drive can be replaced at any time. The cluster will undergo complete reprotection and rebalance.
- 1 node offline – Data is protected.

The TOE can detect these failures and go into the secure state provided by FPT_FLS.1.

TOE Security Functional Requirements Satisfied: FRU_FLT.2.

7.1.7 TOE Access

The TOE allows for users to manually terminate Web UI and QQ CLI sessions. In the Web UI this can be accomplished by clicking the “sign out” button and through QQ CLI by entering the “exit” command.

TOE Security Functional Requirements Satisfied: FTA_SSL.4.

7.1.8 Trusted Path/Channels

The TOE uses a secure channel over HTTP with TLS between itself and remote TOE users that use the Web UI, REST API, and QQ CLI interfaces. For other connections to the TOE, the SMB interface uses SMBv3 and the FTP interface uses FTPS. These connections protect the communicated data from modification or disclosure. Only remote users can initiate communication via the trusted path that is used for the initial user authentication.

TOE Security Functional Requirements Satisfied: FTP_TRP.1.

8. Rationale

8.1 Conformance Claims Rationale

This Security Target conforms to Part 2 and Part 3 of the *Common Criteria for Information Technology Security Evaluation*, Version 3.1 Release 5.

8.2 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, and assumption that compose the Security Target. Sections 8.2.1, 8.2.2, and 8.2.3 demonstrate the mappings between the threats, policies, and assumptions to the security objectives are complete. The following discussion provides detailed evidence of coverage for each threat, policy, and assumption.

8.2.1 Security Objectives Rationale Relating to Threats

Table 16 below provides a mapping of the objectives to the threats they counter.

Table 16 – Threats: Objectives Mapping

Threats	Objectives	Rationale
T.DATA_CORRUPTION TOE user data and configuration data could become corrupted due to hardware failure or incorrect system operations.	O.ADMIN The TOE must include a set of functions that allow efficient management of its functions and data, ensuring that TOE users with the appropriate privileges and only those TOE users, may exercise such control.	O.ADMIN mitigates this threat by allowing TOE users to manage the TOE.
	O.FAULT_TOLERANCE The TOE must be resilient against node or disk failures that might affect the security of the information it contains.	O.FAULT_TOLERANCE mitigates this threat by ensuring that the TOE is capable of maintaining a secure state and offering its full set of security functionalities in the event of a node or disk failure.
	O.USER_DATA The TOE must prevent unauthorized modifications to configuration and user data that it has been entrusted to protect.	O.USER_DATA mitigates this threat by providing mechanisms to protect the configuration and user data that has been entrusted to the TOE against unauthorized modifications
T.IMPROPER_SERVER A TOE user or attacker could attempt to bypass the access controls provided by the TOE using one of the systems connected to the TOE.	O.ADMIN The TOE must include a set of functions that allow efficient management of its functions and data, ensuring that TOE users with the appropriate privileges and only those TOE users, may exercise such control.	O.ADMIN mitigates this threat by allowing TOE users to properly configure the mechanisms of the TOE designed to control the access policy.

Threats	Objectives	Rationale
	O.AUTHENTICATE The TOE must be able to identify and authenticate users prior to allowing access to TOE administrative functions and data.	O.AUTHENTICATE mitigates this threat by ensuring that TOE users are authenticated before allowing access to TOE management functionality. This objective also ensures that strong passwords are used for TOE users' credentials.
	OE.SECURE_COMMUNICATION The TOE environment must provide un-tampered communications between systems connected to the TOE and between TOE components for any connections that the TOE is not already protecting.	OE.SECURE_COMMUNICATION mitigates this threat by ensuring that all unprotected communications involving the TOE and nodes are protected by properly configuring the network to ensure communications are untampered for data sent to or from the TOE and between nodes.
	O.USER_DATA The TOE must prevent unauthorized modifications to configuration and user data that it has been entrusted to protect.	O.USER_DATA mitigates this threat by providing adequate mechanisms to give only authorized TOE users access to configuration data.
T.MASQUERADE A TOE user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	O.AUTHENTICATE The TOE must be able to identify and authenticate users prior to allowing access to TOE administrative functions and data.	O.AUTHENTICATE mitigates this threat by ensuring that TOE users are authenticated and identified before they are allowed access to TOE administrative functions and data.
T.NO_AUDIT A TOE user or attacker may perform security-relevant operations on the TOE without being held accountable for them.	O.AUDIT The TOE must record events of security relevance. The TOE must record the resulting actions of the security functional policies and associate the identity of the user causing the audited event with the audit log.	O.AUDIT mitigates this threat by ensuring that an audit trail of management events on the TOE is preserved. Accurate timestamps are also provided for all audit records, allowing order of events to be preserved.
T.TAMPERING A TOE user or process may be able to bypass the TOE's security mechanisms by tampering with the TOE or TOE environment.	O.ADMIN The TOE must include a set of functions that allow efficient management of its functions and data, ensuring that TOE users with the appropriate privileges and only those TOE users, may exercise such control.	O.ADMIN mitigates this threat by ensuring that TOE users are given the least amount of privileges necessary to prevent malicious modification of TOE settings or security mechanisms.
	O.AUDIT The TOE must record events of security relevance. The TOE must record the resulting actions of the security functional policies and associate the identity of the user causing the audited event with the audit log.	O.AUDIT mitigates this threat by ensuring that all actions taken by a TOE user attempting to tamper with the TOE or TOE environment are logged.
	O.AUTHENTICATE The TOE must be able to identify and authenticate users prior to allowing access to TOE administrative functions and data.	O.AUTHENTICATE mitigates this threat by ensuring that TOE users are authenticated and identified before they are allowed access to TOE administrative functions and data.
T.UNAUTH A TOE user may gain access to security data on the TOE, even though the TOE user is not authorized in accordance with the TOE security policy.	O.AUTHENTICATE The TOE must be able to identify and authenticate users prior to allowing access to TOE administrative functions and data.	O.AUTHENTICATE mitigates this threat by ensuring that TOE users are authenticated and identified before they are allowed access to TOE administrative functions and data.

Every threat is mapped to one or more objectives in the table above. This complete mapping demonstrates that the defined security objectives counter all defined threats.

8.2.2 Security Objectives Rationale Relating to Policies

There are no OSPs defined for this ST.

8.2.3 Security Objectives Rationale Relating to Assumptions

Table 17 below gives a mapping of assumptions and the environmental objectives that uphold them.

Table 17 – Assumptions: Objectives Mapping

Assumptions	Objectives	Rationale
A.CONNECTIVITY It is assumed that the TOE environment will be configured in such a way as to allow TOE users to access the information stored on the TOE.	OE.CONNECT TOE users will configure the TOE environment so that they have the proper network support to be able to access data on the TOE.	OE.CONNECT upholds this assumption by ensuring that the TOE environment is configured appropriately to allow TOE users to access information stored on the TOE.
	OE.SECURE_COMMUNICATION The TOE environment must provide un-tampered communications between systems connected to the TOE and between TOE components for any connections that the TOE is not already protecting.	OE.SECURE_COMMUNICATION upholds this assumption by ensuring that the communication between the TOE nodes, NSF share, and external servers are protected by properly configuring the network.
A.INTERNAL_STORAGE_NETWORK The network that the TOE uses for storage transfer is intended to be an intranet that is protected from access by entities outside of the organization. External access to storage services are blocked by the TOE environment.	OE.INTERNAL_STORAGE_NETWORK The TOE environment must limit access to the TOE from external entities such that only internal hosts can access the storage functionality provided by the TOE.	OE.INTERNAL_STORAGE_NETWORK upholds this assumption by ensuring that only internal hosts can access the storage provided by the TOE.
A.INTERNAL_USERS It is assumed that TOE users accessing the storage on the TOE reside on the internal network and are not careless, negligent, or willfully hostile with regard to their access of the TOE.	NOE.INTERNAL_USERS Sites using the TOE shall ensure that internal TOE users are not careless, negligent, or willfully hostile.	NOE.INTERNAL_USERS upholds this assumption by ensuring that the internal users accessing TOE storage are not careless, negligent, or willfully hostile.
A.LOCATE It is assumed that the TOE is located within a controlled access facility and is physically available to authorized TOE users only.	NOE.PHYSICAL The TOE will be used in a physically secure site that protects it from interference and tampering by un-trusted subjects.	NOE.PHYSICAL upholds this assumption by ensuring that physical security is provided for the TOE.
A.NOEVIL It is assumed that the TOE users who manage the TOE (only within the internal private network) are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.	NOE.NOEVIL Sites using the TOE shall ensure that TOE users are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.	NOE.NOEVIL upholds this assumption by ensuring that TOE users are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.

Every assumption is mapped to one or more objectives in the table above. This complete mapping demonstrates that the defined security objectives uphold all defined assumptions.

8.3 Rationale for Extended Security Functional Requirements

There are no extended SFRs defined for this ST.

8.4 Rationale for Extended TOE Security Assurance Requirements

There are no extended SARs defined for this ST.

8.5 Security Requirements Rationale

The following discussion provides detailed evidence of coverage for each security objective.

8.5.1 Rationale for Security Functional Requirements of the TOE Objectives

Table 18 below shows a mapping of the objectives and the SFRs that support them.

Table 18 – Objectives: SFRs Mapping

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN The TOE must include a set of functions that allow efficient management of its functions and data, ensuring that TOE users with the appropriate privileges and only those TOE users, may exercise such control.	FIA_ATD.1 User attribute definition	This requirement meets O.ADMIN by ensuring that TOE user attributes are maintained by the TOE.
	FMT_MOF.1 Management of security functions behavior	This requirement meets O.ADMIN by specifying the security functions of the TOE and which roles can perform them.
	FMT_MSA.1 Management of security attributes	This requirement meets O.ADMIN by specifying the security attributes of the TOE that can be modified and which roles can modify them.
	FMT_MSA.3 Static attribute initialisation	This requirement meets O.ADMIN by specifying that restrictive values are used by access controls enforced by the TOE and specifying the roles that can set alternate values.
	FMT_MTD.1 Management of TSF data	This requirement meets O.ADMIN by specifying what roles can operate on TSF data contained in the TOE configuration.

Objective	Requirements Addressing the Objective	Rationale
	FMT_SMF.1 Specification of management functions	This requirement meets O.ADMIN by specifying each of the management functions that are used to securely manage the TOE. These functions are provided by the TOE management interfaces.
	FMT_SMR.1 Security roles	This requirement meets O.ADMIN by specifying the roles defined to govern management of the TOE.
	FTA_SSL.4 User-initiated termination	This requirement meets O.ADMIN by providing TOE users with the option to log out of an active session with the management interfaces.
O.AUDIT The TOE must record events of security relevance. The TOE must record the resulting actions of the security functional policies and associate the identity of the user causing the audited event with the audit log.	FAU_GEN.1 Audit Data Generation	This requirement meets O.AUDIT by requiring the TOE to produce audit records for the system security events.
	FAU_GEN.2 User Identity Association	This requirement meets O.AUDIT by requiring the TOE to associate the identity of TOE users with the audit event they caused.
	FPT_STM.1 Reliable time stamps	This requirement meets O.AUDIT by providing reliable timestamps for audit logs.
O.AUTHENTICATE The TOE must be able to identify and authenticate users prior to allowing access to TOE administrative functions and data.	FIA_AFL.1 Authentication failure handling	This requirement meets O.AUTHENTICATE by locking out TOE users if they fail to login a certain number of times.
	FIA_SOS.1 Verification of secrets	This requirement meets O.AUTHENTICATE by requiring TOE users to defined secure passwords.
	FIA_UAU.1 Timing of authentication	This requirement meets O.AUTHENTICATE by requiring TOE users to authenticate their claimed identities before the TOE will perform any action on their behalf via the management interfaces.
	FIA_UAU.5 Multiple authentication mechanisms	This requirement meets O.AUTHENTICATE by having multiple forms of authentication.
	FIA_UAU.7 Protected authentication feedback	This requirement meets O.AUTHENTICATE by preventing passwords from being read while typing them into the login prompts for the TOE management interfaces.
	FIA_UID.1 Timing of identification	This requirement meets O.AUTHENTICATE by requiring TOE users to identify themselves before the TOE perform any actions on their behalf.
O.FAULT_TOLERANCE The TOE must be resilient against node or disk failures that might affect the security of the information it contains.	FPT_FLS.1 Failure with preservation of secure state	This requirement meets O.FAULT_TOLERANCE by ensuring that the TOE maintains a secure state in the event of a disk or host failure.
	FRU_FLT.2 Limited fault tolerance	This requirement meets O.FAULT_TOLERANCE by ensuring that the TOE does not lose any functionality in the event of a disk or host failure.

Objective	Requirements Addressing the Objective	Rationale
O.USER_DATA The TOE must prevent unauthorized modifications to configuration and user data that it has been entrusted to protect.	FDP_ACC.1 Subset access control	This requirement meets O.USER_DATA by enforcing an access control policy that ensures that only authorized devices gain access to TOE user and configuration data within the TOE.
	FDP_ACF.1 Security attribute based access control	This requirement meets O.USER_DATA by providing access control functionality to manage access to user and configuration data within the TOE.
	FTP_TRP.1 Trusted path	This requirement meets O.USER_DATA by providing a secure path between the TOE and the Web UI, QQ CLI, FTP, SMB, and REST API and protecting data from disclosure and modification.

8.5.2 Security Assurance Requirements Rationale

EAL2 was chosen to provide a low to moderate level of assurance that is consistent with good commercial practices. As such, minimal additional tasks are placed upon the vendor assuming the vendor follows reasonable software engineering practices and can provide support to the evaluation for design and testing efforts. The chosen assurance level is appropriate with the threats defined for the environment. While the System may monitor a hostile environment, it is expected to be in a non-hostile position and embedded in or protected by other products designed to address threats that correspond with the intended environment. At EAL2, the system will have incurred a search for obvious flaws to support its introduction into the non-hostile environment.

The augmentation of ALC_FLR.2 was chosen to give greater assurance of the developer’s on-going flaw remediation processes.

8.5.3 Dependency Rationale

The SFRs in this ST satisfy all of the required dependencies listed in the Common Criteria, applicable PPs, and SFRs explicitly stated in this ST. Table 19 lists each requirement to which the TOE claims conformance and indicates whether the dependent requirements are included. As the table indicates, all dependencies have been met.

Table 19 – Functional Requirements Dependencies

SFR ID	Dependencies	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	✓	FPT_STM.1 is satisfied by the TOE environment.
FAU_GEN.2	FAU_GEN.1	✓	
	FIA_UID.1	✓	
FDP_ACC.1	FDP_ACF.1	✓	
FDP_ACF.1	FDP_ACC.1	✓	
	FMT_MSA.3	✓	
FIA_AFL.1	FIA_UAU.1	✓	

SFR ID	Dependencies	Dependency Met	Rationale
FIA_ATD.1	No dependencies		
FIA_SOS.1	No dependencies		
FIA_UAU.1	FIA_UID.1	✓	
FIA_UAU.5	No dependencies		
FIA_UAU.7	FIA_UAU.1	✓	
FIA_UID.1	No dependencies		
FMT_MOF.1	FMT_SMR.1	✓	
	FMT_SMF.1	✓	
FMT_MSA.1	FDP_ACC.1	✓	
	FMT_SMR.1	✓	
	FMT_SMF.1	✓	
FMT_MSA.3	FMT_SMR.1	✓	
	FMT_MSA.1	✓	
FMT_MTD.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_SMF.1	No dependencies		
FMT_SMR.1	FIA_UID.1	✓	
FPT_FLS.1	No dependencies		
FPT_STM.1	No dependencies		
FRU_FLT.2	FPT_FLS.1	✓	
FTA_SSL.4	No dependencies		
FTP_TRP.1	No dependencies		

9. Acronyms and Terms

Table 20 defines the acronyms and terms used throughout this document.

Table 20 – Acronyms and Terms

Acronym	Definition
AD	Active Directory
API	Applicable Programming Interface
bstore	Block Store
CC	Common Criteria
CLI	Command Line Interface
CM	Configuration Management
EAL	Evaluation Assurance Level
FTP	File Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
NFS	Network File System
OS	Operating System
OSP	Organizational Security Policy
PP	Protection Profile
pstore	Protected Store
REST	Representational State Transfer
SAR	Security Assurance Requirement
SFP	Security Functional Policy
SFR	Security Functional Requirement
SMB	Server Message Block
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSP	TOE Security Policy
UI	User Interface

Prepared by:

Corsec Security, Inc.

12600 Fair Lakes Drive, Suite 210
Fairfax, VA 22003
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
http://www.corsec.com