EMC Corporation
ViPR Controller v2.1.0.3

Security Target

Evaluation Assurance Level (EAL): EAL 2+
Document Version: 0.10

Prepared for:

EMC²
where information lives®

EMC Corporation
176 South Street
Hopkinton, MA 01748
United States of America

Phone: +1 508 435 1000
http://www.emc.com

Prepared by:

Corsec®

Corsec Security, Inc.
13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

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

Table of Contents

Table of Figures

List of Tables

1 Introduction

This section identifies and describes the Security Target (ST), the Target of Evaluation (TOE), and the ST organization. The Target of Evaluation (TOE) is the EMC ViPR Controller v2.1.0.3, and will hereafter be referred to as ViPR or the TOE throughout this document. The TOE is a software-only storage solution that abstracts, pools, and automates management of a data center's underlying physical storage infrastructure.

1.1 Document Organization

This ST is divided into nine sections, as follows:

- Introduction (Section 1) – Provides a brief summary of the organization and content of this ST. It also presents an overview of the TOE security functionality, describes the physical and logical scope of the TOE, and provides the ST and TOE references.
- Conformance Claims (Section 2) –Identifies any Common Criteria (CC), Protection Profile, 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 met by the TOE.
- TOE Security Specification (Section 7) – Describes the security functions provided by the TOE that satisfy the security functional requirements 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	EMC Corporation ViPR Controller v2.1.0.3 Security Target
ST Version	Version 0.10
ST Author	Corsec Security, Inc.
ST Publication Date	11/6/2015
TOE Reference	EMC ViPR Controller v2.1.0.3 build #602

1.3 Product Overview

The Product Overview provides a high-level description of the product that is the subject of the evaluation. Section 1.4, TOE Overview, will provide the introduction to the parts of the overall product offering that are specifically being evaluated.

ViPR Controller is a software-defined storage solution that abstracts, pools, and automates a data center's physical storage infrastructure. Deployed in the form of a virtual appliance (vApp), this platform provides administrators with a single control plane to administer heterogeneous storage systems. The abstraction of the control path allows several physical storage pools to be combined into a virtual pool. Separating the control and data paths ensures that applications can access storage while administration occurs.

For load-balancing, each instance of ViPR Controller is deployed as a collection of three or five clustered "nodes" bundled into the vApp. Each node is identical in terms of feature sets, as each implements ViPR's core storage provisioning capabilities. ViPR's load-balancing logic is distributed across all nodes, determining which node in the cluster will respond to a given service request and then directing the request to the designated node.

Management of multiple data centers in different geographic locations is also possible using ViPR. These geographically-dispersed data centers can replicate data between them to protect against data center failures. ViPR enables a storage administrator to:

- discover physical storage, storage area networks (SANs), and hosts;
- define policies to control access and replication across the data center;
- automate storage tasks and provisioning;
- define basic storage characteristics such as compression and high availability.

ViPR simplifies and automates repetitive storage provisioning tasks for multi-vendor block and file storage environments. For a full list of supported array and storage protocol support, see the EMC ViPR Support Matrix. ViPR administrators can add, provision, manage, and share storage from a single software control point. Enterprises and service providers can operate multi-tenant environments and provide simple, self-service access to block, file, and object storage resources. ViPR monitors and reports on the health of the physical storage infrastructure as well as usage, available capacity, and performance. ViPR can meter storage usage, provide chargeback to tenants, and integrate with existing billing systems. ViPR provides users with a self-service portal in which they allocate storage out of a virtual storage pool.

ViPR offers three management interfaces: a REST[1] API[2], web-based user interface (UI), and a command line interface (CLI).

- The REST API is available to multiple storage applications, and most data center management tasks are provided through this interface. A separate set of REST APIs is used to integrate with the back-end storage arrays.
- The UI is provided for web-based management of the data center. The UI presents workspaces to users depending on their role's permissions.
- The CLI provides access to the same functionality as the UI, but is installed on the management workstation with access to ViPR.

1.4 TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. The TOE Overview 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 storage management virtual appliance product that is capable of managing external storage resources of different types and supporting different storage protocols. ViPR Controller allows SAN administrators a single point of management for their entire storage network.

The TOE provides access to provision and backup external storage in the form of a virtual data center (VDC). Within a VDC there are one or more tenants, each tenant representing a specific user or user group. A tenant allows the storage to be managed and provisioned on a per-tenant basis. In a multi-tenant environment, each

[1] REST – Representational State Transfer
[2] API – Application Programming Interface

tenant is configured to use specific resources within the VDC, and users are mapped to particular tenants. This allows the storage allocated to a given group within an organization to be isolated from other groups. Projects are tenant resources and are created within a tenant. All file and block resources provisioned using ViPR must be associated with a project.

TOE users are separated into VDC administrators, tenant administrators, project administrators, and end users. A VDC administrator can add physical storage to the TOE and configure the storage into virtual arrays and virtual pools. Tenant administrators have access to all virtual arrays and pools within their tenant and can manage user's access within their tenant. A project administrator has access to all resources within the project and can assign access control lists (ACLs) to give end users access to a project. End users can create and manage storage in all projects to which they are assigned.

The TOE is deployed as a vApp that contains clusters of three or five nodes. TSF data is distributed across all nodes on the TOE, ensuring enough replication of data to continue all functionality during a node failure. If one node fails, the remaining nodes can take over and maintain a secure state. All TOE functionality is present in each individual node, but multiple nodes are required in the TOE environment to support failover and data consistency. In the three-node cluster, one node can fail and all functionality is maintained. In the five-node cluster, two nodes can fail and all functionality is maintained.

The TOE provides a virtual IP address to access the UI and REST API on each node. In addition, a CLI client can be downloaded to the management workstation. This client provides command line commands for administration of the TOE. The client translates these commands into REST APIs and sends it to the REST API on the controller.

Figure 1 shows an example of how the TOE can be deployed in a SAN environment. Figure 1 also includes the following previously undefined acronyms:

- AD – Active Directory
- LDAP – Lightweight Directory Access Protocol
- OS – Operating System
- SMI-S – Storage Management Initiative – Specification

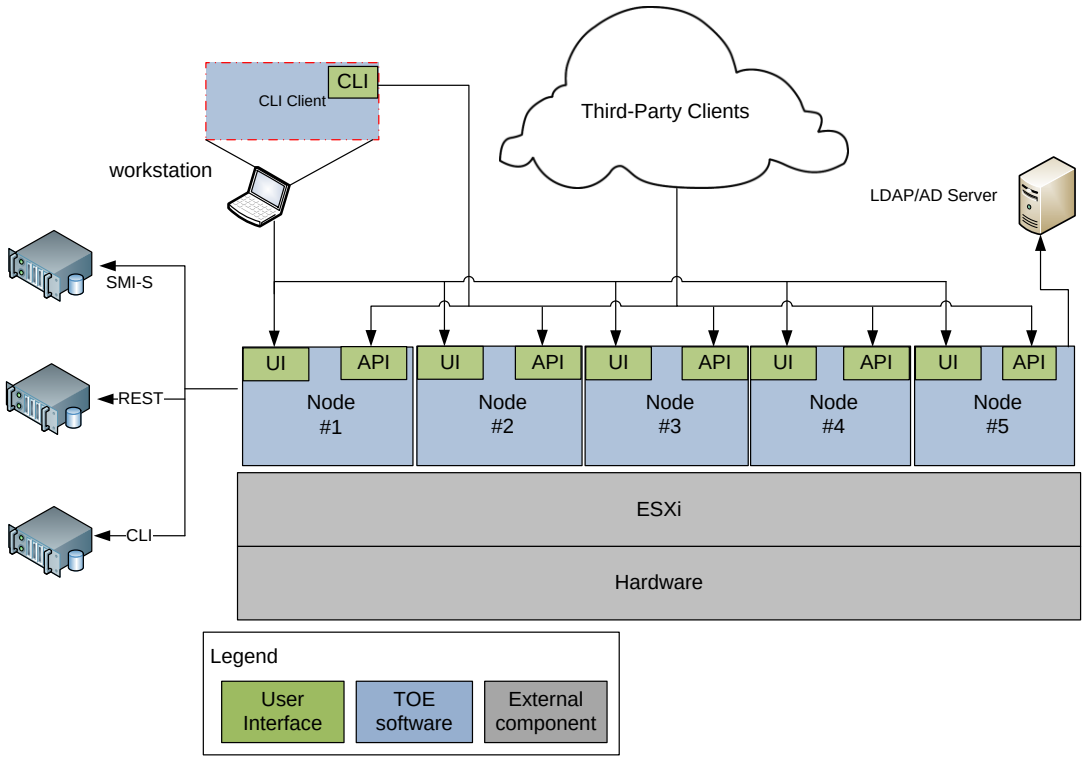

Figure 1 Deployment Configuration of the TOE

1.4.1 Brief Description of the Components of the TOE

The TOE is a software-only TOE. It is deployed in the form of a virtual appliance that includes:

- Node cluster – a collection of three or five nodes containing the core TOE functionality.
- Management interfaces – The ViPR Controller presents a UI and CLI for administration of the TOE. A REST API is presented that integrate with management and reporting applications. This API allows third-party cloud services access to the TOE.
- CLI client – downloaded onto management workstation, allows access to REST API.
- SUSE Linux Enterprise Server 11 Service Pack 3 (SP3) – the guest operating system providing all OS-level functionality.

1.4.2 TOE Environment

The TOE runs on VMware ESXi and a general purpose server, and is deployed with three or five nodes per deployment. Each node runs in its own virtual machine, which virtualizes the underlying physical CPU, memory, and networking infrastructure. An LDAP or Active Directory server must be used for authentication. Physical storage arrays that conform to the EMC support matrix are also required.

Table 2 specifies the minimum system requirements for the proper operation of the TOE.

Table 2 Minimum System Requirements for TOE Operation

Category	Requirement
Authentication servers	LDAP
Physical storage	One or more supported storage provider

Category	Requirement
Virtual machine	• 16 GB[3] RAM[4] per node • 600 GB disk space per node • 4 virtual CPUs[5] per node
Hypervisor	VMware • vSphere v5.5 Enterprise with a VMware ESXi server VMFS 5
Virtualization management	vCenter Server 5 Standard
Hardware	Each virtual CPU requires one physical core of Intel Xeon families with 2.4 GHz or higher speed. For additional requirements, see the VMware Compatibility Guide.
Management workstation	General purpose computer with Windows and one of the following browsers: • Google Chrome v 40 • Internet Explorer v 11 with compatibility mode turned off • Mozilla Firefox v 34 with cookies and pop-ups enabled, add-ons disabled, and JavaScript enabled

1.5 TOE Description

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

1.5.1 Physical Scope

Figure 2 illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE.

The TOE is a software-defined storage platform which runs on VMware ESXi and a general purpose server compliant to the minimum software and hardware requirements as listed in Table 2. The TOE is installed with access to the corporate SAN as depicted in Figure 2 below. There are two configurations for the TOE: a three-node configuration and a five node configuration. Each node includes the same software, features, and functionality. The essential components for the proper operation of the TOE in the evaluated configuration are:

- ViPR Controller with a three-node cluster and a five-node cluster
- HP ProLiant SL 4540 server with VMware ESXi v5.1
- Management workstation with one of the browsers listed in Table 2
- LDAP authentication server
- Physical storage meeting requirements in Table 2

[3] GB – Gigabyte
[4] RAM – Random Access Memory
[5] CPU – Central Processing Unit

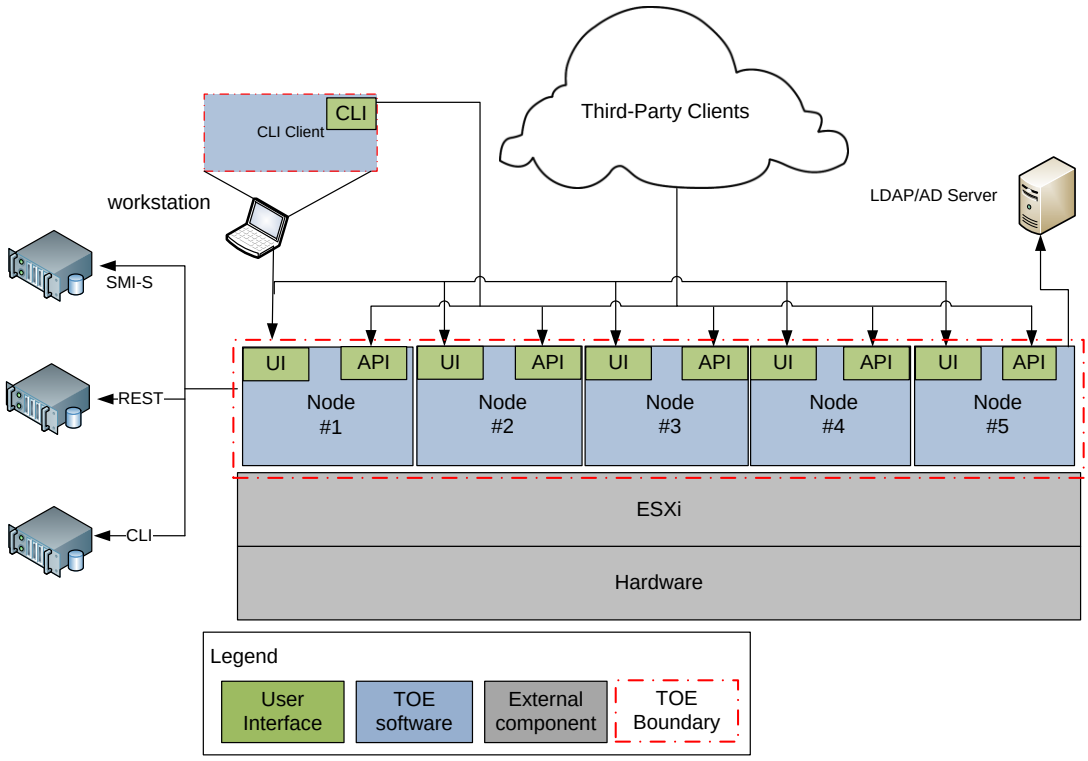

Figure 2 Physical TOE Boundary

1.5.1.1 Guidance Documentation

EMC provides TOE documentation through an online community site, EMC ViPR 2.1 Product Documentation Index Version 12 at https://community.emc.com/docs/DOC-38306 . The following pages are part of the administration guide for the TOE:

- ViPR 2.1 – Understanding ViPR Users, Roles, and ACLs
- ViPR 2.1 – Installing the ViPR CLI
- ViPR 2.1 – Access the ViPR REST API
- ViPR 2.1 – Install EMC ViPR Controller

The following additional guides are also part of the TOE:

- EMC ViPR Release Notes, Release number 2.1.0, 302-001-392, 02, October, 2014
- EMC ViPR v2.1 Security Configuration Guide, 302-001-391, 02, October 2014
- EMC ViPR v2.1 Command Line Reference, 302-001-390, September, 2014

1.5.2 Logical Scope

The logical boundary of the TOE is broken down into the following functional 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 security functional requirements implemented by the TOE are usefully grouped under the functional classes described below.

1.5.2.1 Security Audit

The TOE generates audit logs for startup and shutdown of the Controller, API services, authentication services, nginx service, and general configuration. The System Auditor has access to read, filter, and order the audit data.

1.5.2.2 User Data Protection

End users of the TOE are subject to the Storage Access Control SFP when accessing external storage resources for provisioning and backup. Users and virtual arrays must be assigned to services, projects, and tenants. The user can perform actions their role is authorized for and within the projects and tenants they are assigned.

1.5.2.3 Identification and Authentication

After five unsuccessful authentication attempts, the TOE blocks access from the IP address where the attempts originated. Users must successfully identify and authenticate to the TOE prior to TSF access. When authenticating, the password that the user types in is obscured with dots ("●").

1.5.2.4 Security Management

The TOE maintains the roles Tenant Administrator, Project Administrator, Tenant Approver, System Administrator, Security Administrator, System Monitor, and System Auditor. Each of these roles has unique permissions to perform functions within the TOE. Security attributes within the TOE are initially given permissive default values. These values must be changed by an authorized administrator.

1.5.2.5 Protection of the TOE Security Functionality (TSF)

The TOE provides a reliable timestamps for the Security Audit TSF. The TOE also maintains a secure state when a ViPR Controller fails. TSF data is kept consistent across the TOE nodes through quorum acknowledgements.

1.5.2.6 Resource Utilization

The TOE is deployed in configurations with three or five Controllers. When one Controller fails, all TSF capabilities are maintained by the remaining Controllers.

1.5.2.7 TOE Access

Interactive sessions with the TOE are terminated after a non-configurable time period of eight hours for dynamic interfaces and two hours ten minutes for non-dynamic interfaces. The user must re-authenticate in order to regain access to the TOE.

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

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

- ConnectEMC
- ViPR Data Services

2 Conformance Claims

Table 3 identifies all CC, Protection Profile (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, Revision 4, September 2012; CC Part 2 extended; CC Part 3 conformant; Parts 2 and 3 Interpretations of the CEM as of 2014/09/17 were reviewed, and no interpretations apply to the claims made in this ST.
PP Identification	None
Evaluation Assurance Level	EAL 2+ 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 with 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[6] assets against which protection is required by the TOE or by the security environment. The threat agents are divided into three 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.)
- Node or interface failures: These components within the TOE provide access to or include the TOE's security mechanisms. Loss of these components may compromise these mechanisms and provide access to an attacker who is not a TOE user.

Both are assumed to have a low level of motivation. The IT assets requiring protection are the TSF 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.CRITICAL_FAILURE	The TOE or the storage nodes managed by the TOE may experience a failure that prevents users and administrators from being able to access TOE functionality or data managed by the TOE.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.TAMPERING	A user or process may be able to bypass the TOE's security mechanisms and access user data by tampering with the TOE or TOE environment.
T.UNAUTH	A user may gain access to data or functionality on the TOE, even though the user is not authorized in accordance with the TOE security policy.

3.2 Organizational Security Policies

There are no organizational security policies defined for this Security Target.

[6] IT – Information Technology

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.NETCON	The TOE environment provides the network connectivity required to allow the TOE to perform its intended functions.
A.INSTALL	The TOE is installed on the appropriate, dedicated hardware and hypervisor.
A.LOCATE	The TOE is located within a controlled access facility.
A.PROTECT	The TOE software will be protected from unauthorized modification.
A.MANAGE	There are one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NOEVIL	The users who manage the TOE are non-hostile, 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.ACCESS	The TOE must enforce an access control policy in order to prevent unauthorized users from gaining access to user data stored on the TOE. The TOE must also monitor data for integrity errors to ensure correct data is available to users.
O.FAIL_SECURE	The TOE must provide mechanisms to allow for secure failure.
O.TIMESTAMP	The TOE must provide reliable time stamps.
O.AUDIT	The TOE must record events of security relevance at the "not specified level" of audit. The TOE must record the resulting actions of the security functional policies and provide the authorized administrators with the ability to review and sort the audit trail.
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.AUTHENTICATE	The TOE must be able to identify and authenticate users prior to allowing access to TOE administrative functions and data.
O.PROTECT	The TOE must ensure the integrity of audit and system data by protecting itself from unauthorized modifications and access to its functions and data.

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.ACCESSIBILITY	The TOE is positioned on the network such that authorized users are able to access the TOE's functionality while unauthorized users are blocked from accessing the TOE.
OE.PROTECT	The TOE environment must protect itself and the TOE from external interference or tampering.
OE.PLATFORM	The TOE hardware and hypervisor must support all required TOE functions.

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
OE.MANAGE	Sites deploying the TOE will provide competent, non-hostile TOE administrators who are appropriately trained and follow all administrator guidance. TOE administrators will ensure the system is used securely.
OE.PHYSICAL	The physical environment must be suitable for supporting a computing device in a secure setting.

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

This section specifies the extended SFRs for the TOE. The extended SFRs are organized by class. Table 9 identifies all extended SFRs implemented by the TOE

Table 9 Extended TOE Security Functional Requirements

Name	Description
EXT_FPT_RTC.1	Replicated TSF data consistency

5.1.1 Class FPT: Protection of the TSF

EXT_FPT_RTC.1 has been created to require consistency of replicated TSF data. This SFR is modeled after FPT _TRC.1, Internal TSF consistency. TSF data stored with the TOE may be replicated to multiple components within the TOE. When read and writing this data, the TOE must ensure that the data is consistent prior to the read or write action.

The replication takes place internal to the TOE. The TOE relies on the underlying hypervisor to protect this communication, as demonstrated by OE.PROTECT and OE.PLAFORM. Therefore, the dependency upon FPT_ITT.1 is not required or appropriate for this extended SFR.

5.1.1.1 Replicated TSF data consistency (EXT_FPT_RTC)

Family Behavior

This family defines the requirements for Replicated TSF data consistency functionality.

Component Leveling

Figure 3 EXT_FPT_RTC Replicated TSF data consistency Family Decomposition

EXT_FPT_RTC.1 Replicated TSF data ensures that the TOE can store TSF data across TOE components, to ensure availability in case of a component failure. The TSF must ensure that the data is consistent when stored on multiple components.

Management: EXT_FPT_RTC.1

No management activities foreseen.

EXT_FPT_RTC.1 Replicated TSF data consistency
Hierarchical to: No other components
EXT_FPT_RTC.1.1
The TSF shall ensure that TSF data is consistent when replicated between components of the TOE.
Dependencies: No dependencies

5.2 Extended TOE Security Assurance Components

There are no extended TOE Security Assurance components.

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].
- Refinements are identified using **bold text**. Any text removed is stricken (Example: ~~TSF Data~~) and should be considered as a refinement.
- Extended Functional and Assurance Requirements are identified using "EXT_" at the beginning of the short name.
- Iterations are identified by appending a letter in parentheses following the component title. For example, FAU_GEN.1(a) Audit Data Generation would be the first iteration and FAU_GEN.1(b) Audit Data Generation would be the second iteration.

6.2 Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class. Table 10 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 10 TOE Security Functional Requirements

Name	Description	S	A	R	I
FAU_GEN.1	Audit Data Generation	✓	✓		
FAU_SAR.1	Audit review		✓		
FAU_SAR.2	Restricted audit review				
FAU_SAR.3	Selectable audit review		✓		
FDP_ACC.1	Subset access control		✓		
FDP_ACF.1	Security attribute based access control		✓		
FIA_AFL.1	Authentication failure handling	✓	✓		
FIA_UAU.2	User authentication before any action				
FIA_UAU.7	Protected authentication feedback		✓		
FIA_UID.2	User identification before any action				
FMT_MSA.1	Management of security attributes	✓	✓		
FMT_MSA.3	Static attribute initialisation	✓	✓		
FMT_SMF.1	Specification of management functions		✓		
FMT_SMR.1	Security roles		✓		

Name	Description	S	A	R	I
FPT_FLS.1	Failure with preservation of secure state		✓		
EXT_FPT_RTC.1	Replicated TSF data consistency				
FPT_STM.1	Reliable time stamps				
FRU_FLT.2	Limited fault tolerance		✓		
FTA_SSL.3	TSF-initiated termination		✓		

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) [*API services, authentication services, Controller services, nginx service, and general configuration*].

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, [*service that generated the audit event*].

FAU_SAR.1 Audit review
Hierarchical to: No other components.
Dependencies: FAU_GEN.1 Audit data generation
FAU_SAR.1.1

The TSF shall provide [*System Auditor, System Monitor, and System Administrator*] with the capability to read [*specified audit information*] from the audit records.

FAU_SAR.1.2

The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

FAU_SAR.2 Restricted audit review
Hierarchical to: No other components.
Dependencies: FAU_SAR.1 Audit review
FAU_SAR.2.1

The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

FAU_SAR.3 Selectable audit review
Hierarchical to: No other components.
Dependencies: FAU_SAR.1 Audit review
FAU_SAR.3.1

The TSF shall provide the ability to apply [*filtering and ordering*] of audit data based on [*service, time, level, or event message*].

6.2.3 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 [*Storage Access Control SFP*] on [*the following:*
Subjects: end users and administrators;
Objects: service catalog, virtual array, virtual pool; and
Operations: create order].

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 [*Storage Access Control SFP*] to objects based on the following: [
Subjects: end users and administrators
Attributes: role, username, tenant membership, project assignment, project ACL
Objects: service catalog
Attributes: tenant assignment, service category
Virtual array and virtual pool
Attributes: tenant assignment]

FDP_ACF.1.2

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

1. *If the username is mapped to the tenant (i.e. the user is a member of a tenant) but not a project owner or assigned a project ACL, then the user can create orders using any service in the tenant service catalog with a USE permission.*
2. *If a user is the owner of a project or is assigned a project ACL within a tenant, that user's project ACL determines access to virtual arrays and virtual pools. The user can perform services within the tenant's service catalog. The user's access will be limited to those services with a USE permission and permitted in the project ACL.*
3. *Users assigned to a tenant can access virtual arrays and virtual pools that are associated with their tenant. Users can perform services on virtual arrays and virtual pools that have no tenant association or that are associated with the tenant to which the user is assigned*].

FDP_ACF.1.3

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*if the user's role is Tenant Administrator the user is able to access all services within the tenant's service catalog associated with the user's tenant membership; if the user's role is System Administrator or Security Administrator the user is able to access all services within all service catalogs*].

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [*none*].

6.2.4 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 [[*5*]] unsuccessful authentication attempts occur related to [*UI or REST API interface*].
FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been [met], the TSF shall [*block the IP address of the unsuccessful authentication attempts*].

FIA_UAU.2 User authentication before any action
Hierarchical to: FIA_UAU.1 Timing of authentication
Dependencies: FIA_UID.1 Timing of identification
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_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 [*dots or no feedback*] to the user while the authentication is in progress.

FIA_UID.2 User identification before any action
Hierarchical to: FIA_UID.1 Timing of identification
Dependencies: No dependencies
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.

6.2.5 Class FMT: Security Management

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 [*Storage Access Control SFP*] to restrict the ability to [change_default, modify, delete] the security attributes [*listed in Attribute column of Table 11*] to [*the authorized roles listed in the Role column of Table 11*].

Table 11 Management of Attributes

Attribute	Role
VDC role	Security Administrator
virtual array name, virtual pool name	System Administrator
tenant, project	Tenant Administrator and Security Administrator
Tenant role	Tenant Administrator and Security Administrator

FMT_MSA.3 Static attribute initialisation
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 [*Storage Access Control SFP*] to provide [permissive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the [*the authorized roles listed in the Role column of Table 11*] to specify alternative initial values to override the default values when an object or information is created.

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: [*assigning roles and ACLs, creating tenants and projects, adding physical storage, configuring virtual storage, management of external authentication servers, viewing audit logs*].

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 [*Tenant Administrator, Project Administrator, Tenant Approver, System Administrator, Security Administrator, System Monitor, and System Auditor*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

6.2.6 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: [*single ViPR Controller 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.

EXT_FPT_RTC.1 Replicated TSF data consistency
Hierarchical to: No other components.
Dependencies: No dependencies.
FPT_RTC.1.1
The TSF shall ensure that TSF data is consistent when replicated between components of the TOE.

6.2.7 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: [*failure of a ViPR Controller node*].

6.2.8 Class FTA: TOE Access

FTA_SSL.3 TSF-initiated termination
Hierarchical to: No other components.
Dependencies: No dependencies.
FTA_SSL.3.1

The TSF shall terminate an interactive session after a [*two hour ten minute period of inactivity on non-dynamic interfaces and an eight hour period on dynamic interfaces*].

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 EAL 2 augmented with ALC_FLR.2. Table 12 summarizes the requirements.

Table 12 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 Security 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 function. Hence, each security function 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 13 lists the security functions and their associated SFRs.

Table 13 Mapping of TOE Security Functionality to Security Functional Requirements

TOE Security Function	SFR ID	Description
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_SAR.1	Audit review
	FAU_SAR.2	Restricted audit review
	FAU_SAR.3	Selectable audit review
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_UAU.2	User authentication before any action
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification before any action
Security Management	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialisation
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of TOE Security Functions	FPT_FLS.1	Failure with preservation of secure state
	EXT_FPT_RTC.1	Replicated TSF data consistency
	FPT_STM.1	Reliable time stamps
Resource Utilization	FRU_FLT.2	Limited fault tolerance
TOE Access	FTA_SSL.3	TSF-initiated termination

7.1.1 Security Audit

The TOE generates service log records and audit logs for event that occur within the TOE. Each service within the TOE generates service log records that are compiled and stored in the /opt/storageos/logs folder, with each service having a file named after the service. The System Monitor and System Administrator have access to these service logs. Service log events include:

- API services – includes administrative and third-party consumer actions taken over the API; such as assigning ACLs, adding authentication providers, adding storage, and creating and configuring virtual arrays.
- Authentication services – includes addition and deletion of external authentication servers and user authentication events
- Controller services – includes creating and expanding attached block and file storage, provisioning of virtual arrays and pools
- Network access coordination service – includes node outages, IP address synchronization locks, and naming and configuration changes
- General configuration service – includes all changes made to TSF functions at the system level
- Ngnix service – includes host connections to TOE, loss of node interfaces, and blocking IP addresses for multiple authentication failures

The service log records contain a level field. The levels can be: error, warn, info, or debug. The System Monitor or System Administrator can also order the logs by time, level, and service. Only the System Monitor and System Administrator roles can review the per-service log files. The per-service log files are available through all interfaces.

An audit log is generated from key events in the service log files and stored in the TOE's database. These audit logs are available for review from the management interfaces by only the System Auditor role. The shutdown and startup of the system is audited. The audit function cannot be shut down separately; therefore, the audit of system startup and shutdown events account for the startup and shutdown of the audit function.

All TOE audit records contain the information listed in Table 14 below.

Table 14 Audit Record Contents

Field	Content
Timestamp	date and time that event occurred
Result	description of the event including success or failure
User	association with user that caused the event, when applicable
Service Type	all events are categorized by the type of service that generates the message. Possible options are: • Project • Tenant • VDC • System Audit
Description	Additional details on the event.

Only the System Auditor role can view, filter, and order all audit logs from the UI. The audit logs can be sorted and filtered by date and time.

TOE Security Functional Requirements Satisfied: FAU_GEN.1, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3

7.1.2 User Data Protection

The Storage Access Control SFP manages end user's and administrator's access to the service catalog. The service catalog is a listing of pre-defined services the TOE can perform on managed, external storage resources. There are three sets of permissions with the TOE: virtual array and virtual pools ACL, service catalog ACL, and project ACL. A System Administrator can create an ACL for a virtual array or virtual pool. This ACL gives a Tenant permission to USE the virtual arrays or pools. This then allows users within that Tenant to order services be performed on these resources. Virtual arrays and pools are public by default.

Similarly, a Tenant Administrator can assign a USE ACL to the service catalog or to individual services within the service catalog. All users within that Tenant can access the service catalog and perform the services with the USE permissions. These services vary by storage type and include:

- Creating storage (block or file)
- Expanding storage
- Mounting a block volume
- Creating an NFS datastore

By default, the service catalog within a Tenant is public to users within that Tenant.

A Tenant Administrator or Project Administrator can create projects within a Tenant, but access to the project is prohibited by default. The creator of a project is assigned as the owner of the project using the OWN ACL. An owner can:

- Perform create, read, update, and delete operations on resources within the project
- Set the ACLs and properties on the project
- Delete the project

The owner of the project can give the available tenant services two types of ACLs: ALL or BACKUP. The ALL ACL allows users assigned to the project to use all available services in the tenant service catalog. This includes create, read, update, delete operations on the resources in the project. The BACKUP ACL provides read-only access to volumes, file systems, and buckets and full access to services to create, delete, and export snapshots for the resources within the project. Users can be assigned both ALL and BACKUP. The Tenant Administrator maps end users to Tenants and configures ACLs for projects within their Tenant.

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

7.1.3 Identification and Authentication

All users must be successfully identified and authenticated to the TOE prior to accessing any TSF functionality. Users attempting to access the TOE will have their IP address blocked after five unsuccessful authentication attempts for 60 minutes.

The root, sysmonitor, proxyuser, and svcuser user accounts are default accounts and are authenticated using a local database. All other users' added to the TOE from a configured LDAP or AD server. The users must enter a valid username, domain, and password in order to access the TOE. The password is obscured when entered, using dots on the GUI and no feedback on the CLI. The TOE sends the username and password to the authentication server to verify the user's credentials prior to allowing access to the TOE. Once the TOE receives verification of the validity of the user's credentials, the user is mapped to a TOE role. The user then has access to the TOE functions according to that role.

TOE Security Functional Requirements Satisfied: FIA_AFL.1, FIA_UAU.2, FIA_UAU.7, FIA_UID.2.

7.1.4 Security Management

Users within the TOE are only presented with functionality for which their role has permissions. The following management roles are available within the TOE:

- Tenant Administrator
- Project Administrator
- Tenant Approver
- System Administrator
- Security Administrator
- System Monitor
- System Auditor

A user must be assigned to one or more of these roles for access to the functions specified by that role. Users acting as a VDC administrator will be assigned to the System Administrator, Security Administrator, System Monitor, or System Auditor role all have permissions to manage resources across the VDC. The security attributes managed by these roles are listed in Table 11. If no role is assigned, the user is called an end user and has access only to User mode, the service catalog, orders that the user creates, and resources that the user is assigned. The Tenant Administrator creates ACLs to control end users access. Each role's permissions are described in Table 15 below.

Table 15 Management Access

Role	Permissions
Tenant Administrator	• Assign tenant roles, • add host and clusters of hosts, • configure service catalog for assigned Tenant, • create additional Tenants • access to all projects in the Tenant, • configure approval notifications • All permissions of Project Administrator
Project Administrator	• creates projects • creates ACLs for users on projects
Tenant Approver	• approves orders for tenant
System Administrator	• Add physical storage • Manage resources • Create virtual arrays and pools • Assigns tenants to virtual arrays and pools • Configure object storage • Views system health • View service log files
Security Administrator	• Adds external authentication servers • Sets configuration parameters for storage resources with the virtual data center • Assigns users to administrator roles
System Monitor	• View bulk events and statistics for data center • Read-only access to objects in data center • View service log files
System Auditor	• View data center audit files

Storage arrays are not assigned to a tenant or project when created. Access to these arrays is available to all tenants unless assigned to a tenant. The TOE additionally supports a role of end-user. The end-user role does not have permissions for any management functionality and can only perform service catalog actions covered by the Storage Access Control SFP.

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

7.1.5 Protection of the TSF

The TOE is deployed in a 2+1 or 3+2 combination of Controllers. Each individual ViPR Controller is capable of performing all claimed functions. If one Controller fails, the remaining Controllers can continue to perform all functions and maintain a secure state.

Each node of the TOE contains a database that stores TSF data for the TOE. This TSF data is distributed across the nodes, creating enough replicas of the data to ensure service is not disrupted due to a node failure. Prior to a read or write operation on this TSF data the nodes ensure that a quorum acknowledgement of the operation is received. This quorum acknowledgement requires any node affected by the change to acknowledge receipt of the change before the operation is complete, ensuring data consistency. Writes are blocked until a quorum acknowledgment is received. Reads are compared from the different nodes, if there is a difference the database returns the most recent version of the data to any out-of-sync nodes.

The TOE provides a reliable time stamp. This time stamp is received from the underlying hardware at startup and maintained by the TOE's OS.
TOE Security Functional Requirements Satisfied: FPT_FLS.1, EXT_FPT_RTC.1, and FPT_STM.1.

7.1.6 Resource Utilization

As described in 7.1.5, the secure state and operation of all TOE capabilities is maintained when a ViPR Controller fails. All TOE capabilities are taken over by the remaining ViPR Controllers.

TOE Security Functional Requirements Satisfied: FRU_FLT.2.

7.1.7 TOE Access

After two hours ten minutes of an inactive session on a non-dynamic interface and eight hours on a dynamic interface, such as the Dashboard where status is updated in the background, the TOE will terminate the user's access. When the session is terminated the user must re-authenticate to regain access. This timeout exists on all TOE interfaces. Only the proxyuser account is exempt from the inactive session timeout.

TOE Security Functional Requirements Satisfied: FTA_SSL.3.

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 4.

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.CRITICAL_FAILURE The TOE or the storage nodes managed by the TOE may experience a failure that prevents users and administrators from being able to access TOE functionality or data managed by the TOE.	O.FAIL_SECURE The TOE must provide mechanisms to allow for secure failure.	O.FAIL_SECURE counters this threat by ensuring that the TOE provides mechanisms to allow for secure failure of the TOE or a managed storage node.
T.MASQUERADE A 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.	By ensuring that The TOE is able to identify and authenticate users prior to allowing access to TOE administrative functions and data, O.AUTHENTICATE satifies this threat.
T.TAMPERING A user or process may be able to bypass the TOE's security mechanisms by tampering with the TOE or TOE environment.	O.TIMESTAMP The TOE must provide reliable time stamps.	The objective O.TIMESTAMP ensures that an accurate timestamp is provided for audit records. This ensures that an accurate forensic trail can be followed if an administrator suspects tampering.
	O.AUDIT The TOE must record events of security relevance at the "not specified level" of audit. The TOE must record the resulting actions of the security functional policies and provide the authorized administrators with the ability to review and sort the audit trail.	The objective O.AUDIT ensures that security relevant events that may indicate attempts to tamper with the TOE are recorded.

Threats	Objectives	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.	O.ADMIN supports the mitigation of this threat by ensuring that only authorized users may configure the TOE security mechanisms.
	O.PROTECT The TOE must ensure the integrity of audit and system data by protecting itself from unauthorized modifications and access to its functions and data.	O.PROTECT mitigates this threat by providing mechanisms to protect the TOE data from unauthorized modification.
	OE.PROTECT The TOE environment must protect itself and the TOE from external interference or tampering.	OE.PROTECT ensures that the TOE is protected from external interference or tampering.
T.UNAUTH A user may gain access to data or functionality on the TOE, even though the user is not authorized in accordance with the TOE security policy.	O.ACCESS The TOE must enforce an access control policy in order to prevent unauthorized users from gaining access to user data stored on the TOE. The TOE must also monitor data for integrity errors to ensure correct data is available to users.	The objective O.ACCESS ensures that users have the proper permissions prior to accessing data and functions on the TOE.
	O.AUDIT The TOE must record events of security relevance at the "not specified level" of audit. The TOE must record the resulting actions of the security functional policies and provide the authorized administrators with the ability to review and sort the audit trail.	The objective O.AUDIT ensures that unauthorized attempts to access the TOE are recorded.
	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.	The objective O.ADMIN ensures that access to TOE security data is limited to those users with access to the management functions of the TOE.

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.	The objective O.AUTHENTICATE ensures that users are identified and authenticated prior to gaining access to TOE security 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 organizational security policies 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.NETCON The TOE environment provides the network connectivity required to allow the TOE to perform its intended functions.	OE.ACCESSIBILITY The TOE is positioned on the network such that authorized users are able to access the TOE's functionality while unauthorized users are blocked from accessing the TOE.	OE.ACCESSIBILITY satisfies the assumption that the TOE environment will provide the appropriate connectivity to allow the TOE to perform its function.
A.INSTALL The TOE is installed on the appropriate, dedicated hardware and hypervisor.	OE.PLATFORM The TOE hardware and hypervisor must support all required TOE functions.	OE.PLATFORM ensures that the TOE hardware and OS supports the TOE functions.
	OE.MANAGE Sites deploying the TOE will provide competent, non-hostile TOE administrators who are appropriately trained and follow all administrator guidance. TOE administrators will ensure the system is used securely.	Those responsible for the TOE will provide competent individuals to perform management of the security of the environment, and restrict these functions and facilities from unauthorized use. OE.MANAGE satisfies this assumption.
A.LOCATE The TOE is located within a controlled access facility.	OE.PHYSICAL The physical environment must be suitable for supporting a computing device in a secure setting.	Physical security is provided within the TOE environment to provide appropriate protection to the network resources. OE.PHYSICAL satisfies this assumption.

Assumptions	Objectives	Rationale
A.PROTECT The TOE software will be protected from unauthorized modification.	OE.PROTECT The TOE environment must protect itself and the TOE from external interference or tampering.	The TOE environment provides protection from external interference or tampering. OE.PROTECT satisfies this assumption.
A.MANAGE There are one or more competent individuals assigned to manage the TOE and the security of the information it contains.	OE.MANAGE Sites deploying the TOE will provide competent, non-hostile TOE administrators who are appropriately trained and follow all administrator guidance. TOE administrators will ensure the system is used securely.	OE.MANAGE satisfies the assumption that competent individuals are assigned to manage the TOE and the TSF.
A.NOEVIL The users who manage the TOE are non-hostile, appropriately trained, and follow all guidance.	OE.MANAGE Sites deploying the TOE will provide competent, non-hostile TOE administrators who are appropriately trained and follow all administrator guidance. TOE administrators will ensure the system is used securely.	OE.MANAGE satisfies the assumption that the users who manage the TOE are non-hostile, 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

A family of Replicated TSF data consistency requirements was created to specifically address the consistency of TSF data when replicated among TOE components. The purpose of this family of requirements is to ensure the TSF verifies the consistency of the data prior to performing a read or write operation on the data. These requirements have no dependencies since the stated requirements and associated IT Environment Objectives embody all the necessary security functions. These requirements exhibit functionality that can be easily documented in the ADV assurance evidence and thus do not require any additional Assurance Documentation.

8.4 Rationale for Extended TOE Security Assurance Requirements

No extended assurance requirements are claimed for this Security Target.

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.ACCESS The TOE must enforce an access control policy in order to prevent unauthorized users from gaining access to user data stored on the TOE. The TOE must also monitor data for integrity errors to ensure correct data is available to users.	FDP_ACC.1 Subset access control	The requirement meets the objective by ensuring that access control is applied to all users before granting access to data stored on the TOE.
	FDP_ACF.1 Security attribute based access control	The requirement meets the objective by ensuring that the TOE enforces access control based on the implemented policy.
	FMT_MSA.1 Management of security attributes	The requirement meets the objective by ensuring that only authorized administrators have the capability to modify the permissions for the access control policy.
	FMT_MSA.3 Static attribute initialisation	The requirement meets the objective by ensuring that appropriate default values are granted for end user accounts and storage and that only authorized administrators can modify the initial default permissions.
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.	FAU_SAR.2 Restricted audit review	The requirement meets this objective by ensuring that the function of reading the audit logs is restricted to those users with permission to read the audit logs.
	FMT_MSA.1 Management of security attributes	The requirement meets the objective by ensuring that security attributes can only be changed by authorized users.
	FMT_SMF.1 Specification of management functions	The requirement meets the objective by ensuring that the TOE includes administrative functions to facilitate the management of the TSF.
	FMT_SMR.1 Security roles	The requirement meets the objective by ensuring that the TOE associates users with roles to provide access to TSF management functions and data.

Objective	Requirements Addressing the Objective	Rationale
O.AUDIT The TOE must record events of security relevance at the "not specified level" of audit. The TOE must record the resulting actions of the security functional policies and provide the authorized administrators with the ability to review and sort the audit trail.	FAU_GEN.1 Audit Data Generation	The requirement meets this objective by ensuring that the TOE maintains a record of defined security related events, including relevant details about the event.
	FAU_SAR.1 Audit review	The requirement meets the objective by ensure that the TOE provides the ability to review logs.
	FAU_SAR.2 Restricted audit review	The requirement meets this objective by ensuring that only those users with permission to read the audit, can read the audit logs.
	FAU_SAR.3 Selectable audit review	The requirement meets this objective by ensuring that the TOE provides a mechanism to filter and order the 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	In order to ensure that users are properly authenticated prior to access, the TOE enforces a lockout after five unsuccessful authentication attempts. The requirement for authentication failure handling meets the objective by mitigating the risk of a brute force attack on a username and password.
	FIA_UAU.2 User authentication before any action	The requirement meets the objective by ensuring that users are authenticated before access to TOE administrative functions is allowed.
	FIA_UAU.7 Protected authentication feedback	The requirement meets the objective by obscuring a user's password while it is being typed into the login prompt for the user interfaces provided by the TOE. This prevents an adversary from reading the password as it is being entered by a user and logging in with their credentials.
	FIA_UID.2 User identification before any action	The requirement meets the objective by ensuring that the users are identified before access to TOE administrative functions is allowed.

Objective	Requirements Addressing the Objective	Rationale
O.FAIL_SECURE The TOE must provide mechanisms to allow for secure failure.	EXT_FPT_RTC.1 Replicated TSF data consistency	The requirement meets the objective by ensuring that the TOE validates TSF data consistency before allowing reads and writes.
	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1 supports this objective by ensuring that the TOE preserves a secure state when a node or virtual machine fails.
	FRU_FLT.2 Limited fault tolerance	The requirement meets the objective by ensuring that all TOE functions are still available when a node or virtual machine fails.
O.PROTECT The TOE must ensure the integrity of audit and system data by protecting itself from unauthorized modifications and access to its functions and data.	FIA_UAU.2 User authentication before any action	The requirement meets the objective by ensuring that the TOE protects itself from unauthorized modification. The TOE does this by ensuring that only authenticated users are allowed access to TOE functions.
	FIA_UID.2 User identification before any action	The requirement meets the objective by ensuring that the TOE protects itself from unauthorized modification. The TOE does this by ensuring that only identified users are allowed access to TOE functions.
	FPT_FLS.1 Failure with preservation of secure state	The requirement meets the objective by ensuring the TOE maintains a secure state when a node or virtual machine fails.
	FTA_SSL.3 TSF-initiated termination	The requirement meets the objective by ensuring that the TSF terminates inactive sessions and require re-authentication.
O.TIMESTAMP The TOE must provide reliable time stamps.	FPT_STM.1 Reliable time stamps	The requirement meets the objective by requiring the TOE to provide a reliable timestamp.

8.5.2 Security Assurance Requirements Rationale

EAL 2 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. At EAL 2, 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	✓	
FAU_SAR.1	FAU_GEN.1	✓	
FAU_SAR.2	FAU_SAR.1	✓	
FAU_SAR.3	FAU_SAR.1	✓	
FDP_ACC.1	FDP_ACF.1	✓	
FDP_ACF.1	FDP_ACC.1	✓	
	FMT_MSA.3	✓	
FIA_AFL.1	FIA_UAU.1	✓	Although FIA_UAU.1 is not included, FIA_UAU.2, which is hierarchical to FIA_UAU.1 is included. This satisfies this dependency.
FIA_UAU.2	FIA_UID.1	✓	Although FIA_UID.1 is not included, FIA_UID.2, which is hierarchical to FIA_UID.1 is included. This satisfies this dependency.
FIA_UAU.7	FIA_UAU.1	✓	Although FIA_UAU.1 is not included, FIA_UAU.2, which is hierarchical to FIA_UAU.1 is included. This satisfies this dependency.
FIA_UID.2	No dependencies	N/A	
FMT_MSA.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
	FDP_ACC.1	✓	
FMT_MSA.3	FMT_MSA.1	✓	

SFR ID	Dependencies	Dependency Met	Rationale
	FMT_SMR.1	✓	
FMT_SMF.1	No dependencies	N/A	
FMT_SMR.1	FIA_UID.1	✓	Although FIA_UID.1 is not included, FIA_UID.2, which is hierarchical to FIA_UID.1 is included. This satisfies this dependency.
FPT_FLS.1	No dependencies	N/A	
EXT_FPT_RTC.1	No dependencies	N/A	
FPT_STM.1	No dependencies	N/A	
FRU_FLT.2	FPT_FLS.1	✓	
FTA_SSL.3	No dependencies	N/A	

9 Acronyms

Table 20 defines the acronyms used throughout this document.

Table 20 Acronyms

Acronym	Definition
ACL	Access Control List
AD	Active Directory
API	Application Programming Interface
CC	Common Criteria
CLI	Command Line Interface
CM	Configuration Management
CPU	Central Processing Unit
EAL	Evaluation Assurance Level
GB	Gigabyte
IP	Internet Protocol
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
MD5	Message Digest 5
OS	Operating System
PP	Protection Profile
RAM	Random Access Memory
REST	Representational State Transfer
SAN	Storage Areas Network
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SMI-S	Storage Management Initiative - Specification
SP	Service Pack
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSP	TOE Security Policy
UI	User Interface
vApp	Virtual Appliance
VDC	Virtual Data Center

Acronym	Definition
XML	eXtensible Markup Language

Prepared by:
Corsec Security, Inc.

13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

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