Nutanix, Inc.

Virtual Computing Platform v3.5.1

Security Target

Evaluation Assurance Level (EAL): EAL2+
Document Version: 0.11

Prepared for:

Nutanix, Inc.
1740 Technology Drive, Suite 400
San Jose, CA 95110
United States of America

Phone: +1 855 688 2649
http://www.nutanix.com

Prepared by:

Corsec Security, Inc.
13135 Lee Jackson Memorial Hwy., Suite 220
Fairfax, VA 22033
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 the Security Target (ST), Target of Evaluation (TOE), and the ST organization. The Target of Evaluation (TOE) is the Virtual Computing Platform v3.5.1, and will hereafter be referred to as the TOE throughout this document. The TOE is a set of four nodes that compose a storage and virtualization system housed within one or more physical chassis running the Nutanix Operating System (NOS). Multiple chassis can be combined into a single cluster to provide a unified storage and virtualization pool. Virtualization support is provided to run Virtual Machines (VMs) on the system, while storage is provided to meet the storage needs of those VMs. This allows the TOE to be a unified solution for virtual server management while eliminating administration overhead by removing the need for a separate storage network.

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, 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	Nutanix, Inc. Virtual Computing Platform v3.5.1 Security Target
ST Version	Version 0.11
ST Author	Corsec Security, Inc.
ST Publication Date	2014-08-27
TOE Reference	Nutanix Virtual Computing Platform v3.5.1

1.3 Product Overview

The Product Overview provides the introduction to the overall product offering.

The Virtual Computing Platform is a virtualization platform composed of a networked cluster of nodes that can host VMs offering services to users (typically as virtual servers). These virtual servers can be used for any application the users of the server require, such as web, email, or others. The Virtual Computing Platform also offers storage for those VMs to use when offering services. The unification of storage and virtualization on a single platform eliminates the need for a separate storage network. This cuts overhead and administrative costs for operating a virtualization platform. Additionally, the Virtual Computing Platform scales linearly to meet increased virtual server processing or storage needs by allowing additional nodes to be added to the cluster individually. This compounds the overhead-reducing effects inherent in a virtualization platform, which reduces hardware needs dramatically as compared to a traditional server infrastructure.

Nodes are hardware boards housed within one or more chassis running the NOS software and provide all of the functionality for the cluster except data storage. Data storage is provided by the disk hardware housed within the chassis. Each node provides storage and virtualization services to users, with multiple nodes being used for redundancy. Each chassis can hold one to four nodes, depending on the model. A node is a blade with a complete instantiation of server hardware (processor, memory, storage, and network) that supports the virtualization and storage needs of the system's users.

The Virtual Computing Platform makes use of the existing physical network infrastructure to connect each node together using standard network protocols, rather than a private physical network. The Virtual Computing Platform provides it's own private subnet for internal cluster communications. This subnet is typically configured to be inaccessible by entities other than cluster nodes.

The foundational unit for the Nutanix Virtual Computing Platform is a grouping of four nodes within one to four chassis, each of which contains processors, memory, and local storage (Solid State Drives (SSDs) and hard disks) and runs a standard hypervisor. Each node hosts a Nutanix Controller VM (CVM) also known as Service VM that enables the pooling of local storage from all nodes in the cluster.

Each chassis model can contain differing numbers of nodes. For example, the 1050 and 30x0 series can hold up to four nodes within a single chassis, the 60x0 series can hold only two nodes per chassis, and the 7110 series holds only a single node per chassis. The number of nodes in the chassis determines the second number of the model. For example, an NX-3050 series chassis with four nodes would be referred to as the NX-3450. The models differ only in the hardware specifications for each node, such as processing power, amount of memory, and number of network connections supported. In all other respects, the nodes operate identically.

When a guest VM running on a node submits a write request through the hypervisor, that request is sent to the CVM on the node. To provide a rapid response to the guest VM, this data is first stored on the SSD-PCIe[1] device, within a subset of storage called the HOT Cache. This cache is rapidly distributed across the 10 GbE[2] network to other SSD-PCIe devices in the cluster. HOT Cache data is periodically transferred to persistent storage within the cluster. Data is written locally for performance and replicated on multiple nodes for high availability.

When the guest VM sends a read request through the hypervisor, the CVM first checks local storage for a copy of the data. If the host does not contain a local copy, then the CVM will read across the network from another host in the cluster that does contain a copy. As remote data is accessed, it is migrated to storage devices on the current host, so that future read requests are local.

[1] PCIe – Peripheral Component Interconnect Express
[2] GbE – Gigabit Ethernet

The Nutanix Distributed Filesystem (NDFS) is at the core of the Nutanix Platform. NDFS manages all metadata and data, and enables core features. NDFS is the underpinning architectural element that connects the storage, compute resources, CVM, and the hypervisor. It also provides full Information Lifecycle Management (ILM), including localizing data to the optimal node. The software portion of the solution is referred to as NOS.

1.3.1 Smart metadata

Metadata is distributed among all nodes in the cluster in order to eliminate any single point of failure and to allow scalability that increases linearly with cluster growth. The metadata is partitioned using a consistent hashing scheme to minimize the redistribution of keys during cluster-sizing modifications.

The system enforces strong consistency through a distributed consensus algorithm. Quorum-based leadership election eliminates potential "split brain" scenarios, which ensures strict consistency of configuration data.

1.3.2 Data availability

NDFS was designed from the ground up to be extremely fault-resilient. It ensures data availability in the event of a node, controller, or disk failure. NDFS creates redundant copies of the data and keeps the replicas (copies) on separate nodes. Writes to system data are logged in the fastest disk tier that includes PCIe SSDs. These can be configured to replicate to another controller before the write is committed. If a node or disk failure occurs, NDFS automatically rebuilds data copies to ensure a copy of the data remains available as much as possible.

The platform is designed to recover automatically from loss of physical components. By leveraging the distributed nature of the cluster, the platform proactively scrubs data to resolve disk or data errors. If a controller VM fails, all disk I/O[3] requests are automatically forwarded to another controller VM until the local controller becomes available again. This technology is completely transparent to the hypervisor, and guest VMs continue to run normally. In the case of a node failure, an HA[4] event is automatically triggered and VMs fail over to other hosts within the cluster. Nutanix localizes disk I/O operations by migrating data to the virtual machine's local CVM. Simultaneously, data is re-replicated to maintain a consistent number of replicas.

NDFS provides built-in converged backup and disaster recovery. The converged-backup capabilities leverage array-side snapshots[5] and clones[6], which are performed using sub-node-level change-tracking at the VM and file level. The snapshots and clones are instantaneous, and thin provisioning maintains very low overhead. These capabilities also support hypervisor array offload[7] capabilities.

Snapshots can be configured on a standard schedule and can be replicated to remote sites using array-side replication[8]. This replication is configurable at the VM level, and only the sub-node-level changes are shipped to the remote replication site.

1.3.3 Data efficiency

A core design principle of the Nutanix platform is data localization. It keeps data proximate to the VM and allows write I/O operations to be localized on that same node. If a VM migrates to another host in an event such as Distributed Resource Scheduling (DRS), the data automatically follows the VM so it maintains the highest performance. After a certain number of read requests made by a VM to a controller that resides on

[3] I/O – Input/Output
[4] HA –High Availability
[5] Snapshots refer to point-in-time backups of files that exist as they were at the time the snapshot was taken.
[6] Clones are copies of VMs.
[7] Array offload refers to distributing computational workloads to the storage array rather than having the client perform these tasks.
[8] Array-side replication refers to the storage system's ability to copy snapshots.

another node, Nutanix ILM transparently moves the remote data to the local controller. The read I/O is served locally, instead of traversing the network.

Nutanix incorporates heat-optimized tiering (HOT), which leverages multiple tiers of storage and optimally places data on the tier that provides the best performance. The architecture was built to support local disks attached to the controller VM (PCIe SSD, HDD) as well as remote (NAS[9]) and cloud-based source targets. The tiering logic is fully extensible, allowing new tiers to be dynamically added and extended. The Nutanix system continuously monitors data-access patterns to determine whether access is random, sequential, or a mixed workload. Random I/O workloads are maintained in an SSD tier to minimize seek times. Sequential workloads are automatically placed into HDD to improve endurance.

The most frequently accessed data (hot data) resides on the highest performance tier (PCIe SSD). That tier is not just a cache – it is a truly persistent tier for both read and write operations. The next hottest data is placed on the SSD tier, which serves as spillover for the highest-performance tier (PCIe SSD), as well as Quality of Service (QoS)-controlled data. Cold data sits on hard disk drives, the highest-capacity, most economical tier.

NDFS array-side compression capabilities work in combination with Nutanix ILM. For sequential workloads, data is compressed during the write operation using in-line compression. For batch workloads, post-process compression adds significant value as data is compressed once it becomes idle and ILM has moved it down to the highest capacity tier (HDD). All compression configurations are carried out at a container level, and operate at a granular VM and file level. Decompression is done at the sub-node level to ensure precise granularity. The operations are monitored by the ILM process, which proactively moves frequently accessed, decompressed data up to a higher performance data tier.

1.3.4 VM Support

The product provides the capabilities to run VMs in the operating environment via a VM controller installed as part of the hypervisor. The guest VMs host the virtualized services that make sure of the storage provided by and managed by the CVM. VMs can be imported to the product from any supported VM controller such as Vmware ESXi, KVM[10], and Hyper-V. Administrative users can backup VM data along with user data through replication functionality available on the product. The product supports a number of guest VMs, including Windows Server 2008 Service Pack (SP) 2 and Centos 6.4.

1.3.5 Management Interfaces

The product offers both a Command Line Interface (CLI) called the Nutanix CLI (nCLI) and a web Graphical User Interface (GUI) called the Web Console that can be used to maintain and configure the product. The CLI is run over Secure Hyptertext Transfer Protocol (HTTPS) and is downloaded from the Web Console. Once retrieved, the CLI is run as a standalone application on the administrative user's workstation. The Web Console also communicates via HTTPS.

1.4 TOE Overview

The TOE Overview summarizes the major security features of the TOE. The TOE is hardware and software that provides the security functionality defined below. The TOE consists of all the software and node hardware running on the Nutanix Virtual Computing Platform in a four-node configuration. This means the chassis and hard disks are considered within the TOE environment.

The TOE enforces a Virtual Disk Access Security Functionality Policy (SFP) on virtual servers that the TOE hosts. This SFP controls virtual server access to the storage that the TOE provides. In order to

[9] NAS – Network Attached Storage
[10] KVM – Kernel-based Virtual Machine

determine if a virtual server can access a virtual disk, the TOE first checks an NFS whitelist, and then checks to determine if the virtual server has been configured to access the NFS share.

The TOE enforces a Virtual Disk Locking SFP on clients attempting to write to or execute files stored on virtual disks. This SFP allows a read or execute operation if the process requesting the operation has obtained a virtual disk lock. If a virtual disk lock does not currently exist for the virtual disk, the TOE allows the process to obtain a virtual disk lock. Otherwise, the operation request is denied.

The TOE is designed to be especially resilient against the failure of its hardware components. In the event of a node or disk failure, the TOE continues offering all of its functionality to TOE users.

The TOE generates audit records for all configuration changes made via the management interfaces (the Web Console and the nCLI). Within these audit records, the TOE includes basic information about the event in a human-readable format. The TOE provides reliable time stamps that are used to preserve the order of events for the audit records.

The TOE includes a set of management interfaces that administrators can use to view the audit logs, configure failover functionality, manage TOE settings, manage user accounts, and configure the storage provided by the TOE. The management interfaces can also be used to configure the Virtual Disk Access SFP and Virtual Disk Locking SFPs. Storage options include access type (pass-through or virtual disk format), tiering options (SSD or HDD), and maximum capacity allocated. There is only one administrative role defined for the TOE. Administrative users can log out of their management sessions at any time.

The TOE requires administrative users to perform identification and authentication before accessing any TOE functionality besides the nCLI `help` command. During authentication via the Web Console, only obscured feedback is provided to the administrative user. The TOE maintains passwords for local user accounts and their associated usernames. Passwords must be at least eight characters long.

1.4.1 TOE Environment

The TOE environment can optionally contain additional Nutanix nodes or multiple instances of the TOE to provide increased redundancy and scalability.

The TOE is designed to run and store multiple guest VMs that offer virtualized services to end users. These VMs are considered to be environmental components running on the TOE and other instances of the TOE. At least one guest VM must be running in order to make use of the storage functionality provided by the TOE.

The TOE requires users to access storage and services through appropriate clients on a general purpose computer. Administrative users should access the Web Console through a modern graphical browser running Java v5.0 or higher. Administrative users should access the nCLI via the provided application code[11] that can be downloaded via the Web Console. The network infrastructure that provides connectivity between users and the TOE is also part of the environment.

It is assumed that only trusted users or software have access to the TOE hardware components. In addition, the TOE hardware components are intended to be deployed in a physically secure cabinet, room, or data center with the appropriate level of physical access control and physical protection (e.g. badge access, fire control, locks, alarms, etc.).

The TOE must have access to an NTP server that can provide reliable time stamps to the TOE.

[11] It should be noted that the application code (running on the administrative user workstation) for the nCLI provides similar functionality to a web browser in the sense that it is only used to display return data from and pass commands to the nCLI (running on the TOE hardware). Therefore the nCLI application code is considered to be a required component of the IT Environment and not part of the TOE.

Figure 1 shows the details of the deployment configuration of the TOE.

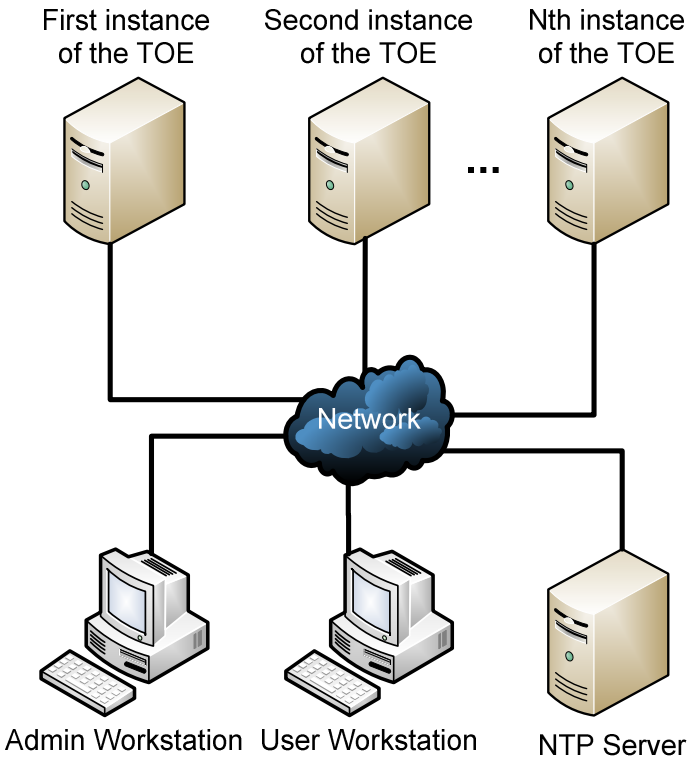

Figure 1 Deployment Configuration of the TOE

1.5 TOE Description

The TOE Description provides a context for the TOE evaluation by defining the specific evaluated configuration. This section primarily addresses the TOE environment and the physical and logical components of the TOE that are included in the evaluation.

1.5.1 Physical Scope

The physical scope of the TOE includes the hypervisor, the node hardware, and NOS. The hypervisor provides the basic interface to the system hardware and provides a virtualized space for NOS to run within a CVM. NOS provides all of the non-virtualization functionality for the TOE. The TOE includes four nodes running in a single cluster on one or more of the chassis listed below.

The evaluated configuration of the TOE was tested on the NX-3450 hardware platform running NOS v3.5.1. NOS was not tested on, but is capable of running on other platforms and is derived from a single image, with different functionality enabled or disabled to support the hardware platform. The hypervisor used for testing was VMware ESXi v5.1 U1. The supported chassis include:

- NX-1050 series,
- NX-3050 series,

- NX-3051 series,
- NX-3060 series,
- NX-3061 series,
- NX-6020 series,
- NX-6050 series,
- NX-6060 series,
- NX-6070 series,
- NX-6080 series,
- NX-7110 series.

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

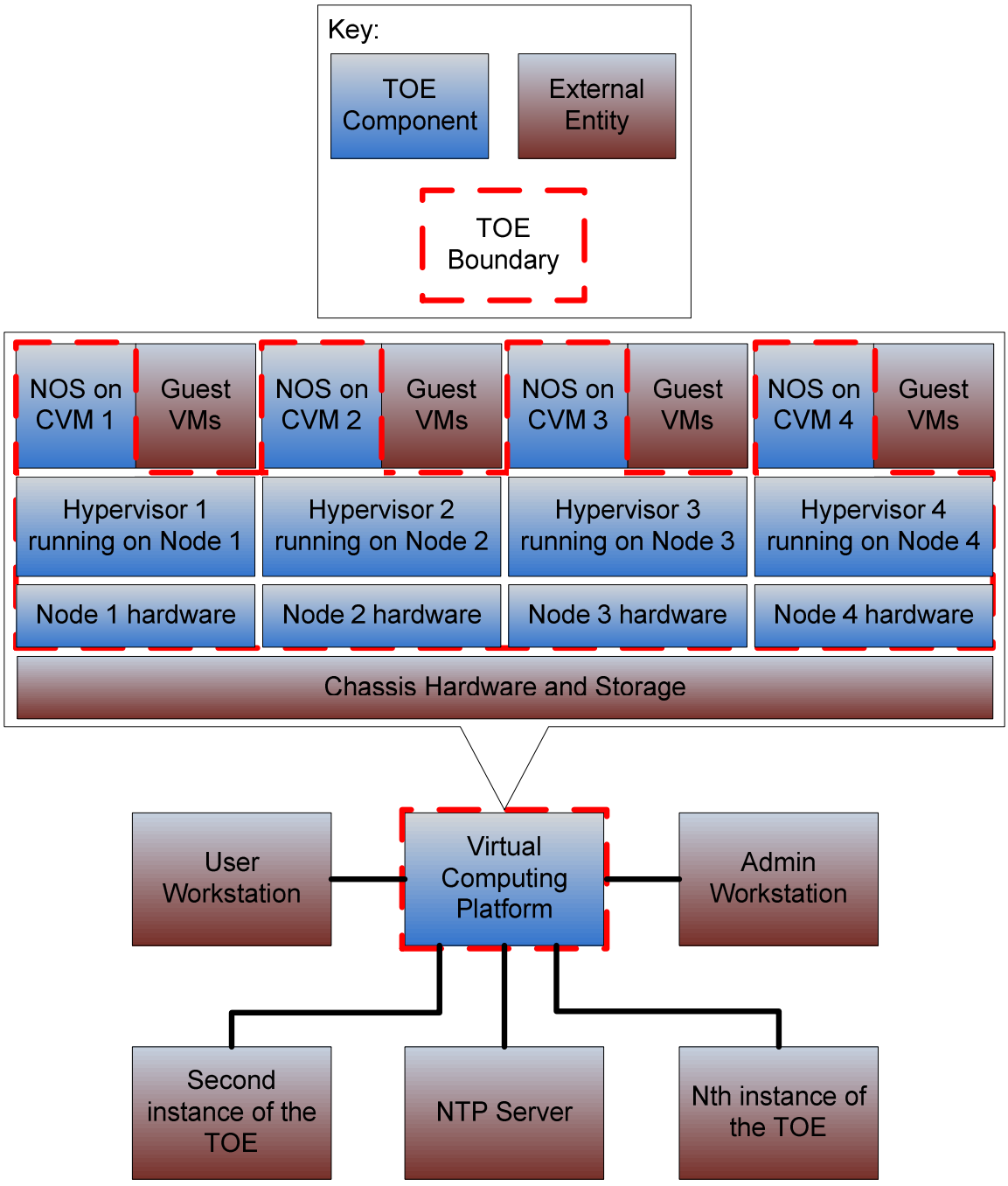

Figure 2 Physical TOE Boundary

The TOE boundary includes all of the Nutanix-developed parts of a four-node deployment of the Virtual Computing Platform. Any third-party source code or software that Nutanix has modified for the Virtual Computing Platform is also considered to be TOE software. The TOE boundary does not include any of the environmental components shown above in Figure 2:

- Guest VMs running on the system,
- Administrator workstation,

- User workstation,
- nCLI appliance code (running on the administrator workstation)
- TOE instances 2 through n.

The TOE also does not include the chassis hardware or disks.

It should be noted that at least one guest VM must be running as part of the TOE environment in order for the storage provided by the TOE to be used.

1.5.1.1 Guidance Documentation

Table 2 guides are required reading and part of the TOE.

Table 2 Guidance Documentation

Document Name	Description
Nutanix Platform Administration Guide NOS 3.5 *15-Apr-2014*	Contains information on how to maintain and configure the TOE.
Nutanix Web Console Guide NOS 3.5 15-April-2014	Contains information on how to use the Web Console management interface.
Nutanix Command Reference Guide NOS 3.5 *13-Mar-2014*	Lists the commands available from the nCLI along with a description of how to use each command.
Nutanix Setup Guide NOS 3.5 *15-Apr-2014*	Contains information for the initial setup of the TOE.
Nutanix REST API Reference NOS 3.5 03-Jan-2014	Contains information for the REST API used to access TOE functionality.
Nutanix Physical Installation Guide NX-1000, NX-3050, NX-6000, and NX-7000 Series 764-0015-0004 Rev A 15-Apr-2014	Contains information related to setting up the TOE hardware.
Nutanix Upgrade Guide NOS 3.5.1 27-Mar-2014	Contains information on upgrading the TOE to the evaluated version of NOS.
Nutanix Release Notes NOS 3.5.1 18-Feb-2014	Contain information on updates since the previous release of the Virtual Computing Platform.
Nutanix, Inc. Virtual Computing Platform v3.5.1 Guidance Supplement v0.6	Contains information regarding specific configuration for the TOE evaluated configuration.

1.5.2 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 security functional requirements 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[12],
- Resource Utilization, and
- TOE Access.

1.5.2.1 Security Audit

The TOE records the actions of administrative users made through the nCLI or Web Console. Logs can only be reviewed through the Web Console. Administrative users are the only ones with access to the Web Console.

1.5.2.2 User Data Protection

The TOE enforces access controls on storage allocated to virtual machines. This storage is provided via NFSv4 shares. Access to this storage is controlled via an NFS whitelist that lists the IP address of every host that is allowed to access the storage. The TOE also provides information controls so that only one client can modify virtual disk data at a time.

1.5.2.3 Identification and Authentication

The TOE requires users to identify and authenticate themselves to the TOE before granting permission to access any of the TOE's functionality. Administrative users are required to define strong passwords for themselves. The TOE stores each local user account's username, password, and role. While logging in, the TOE obscures passwords for administrative users.

1.5.2.4 Security Management

The TOE provides the Web Console and the nCLI that administrative users can use to manage the TOE. Administrative users can manage attributes related to the virtual storage and Virtual Disk Locking policies via these interfaces. The Virtual Disk Access policy allows any process to connect via NFSv4 and uses a white-list to restrict access to the NFS shares. The Virtual Disk Locking policy allows any storage access requests to be made by default, unless a virtual disk is already locked. Administrative users can also manage user accounts, storage, and view statistics for virtual machines on the TOE. Administrative users can assume the User Administrator role, Cluster Administrator role, or can be assigned both sets of privileges at once for the Web Console and nCLI.

1.5.2.5 Protection of the TSF

The TOE has the capability to provide time stamps for audit records in order to preserve the proper order of events. The TOE also maintains its full capabilities when a physical disk or node fails.

1.5.2.6 Resource Utilization

The TOE makes use of redundant nodes to prevent a single point of failure. The TOE remains fully operational with all data intact even if an entire physical disk or node fails.

1.5.2.7 TOE Access

The TOE provides the capability for administrative users to log out from the Web Console and nCLI.

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

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

- Guest VMs are not included within the TOE boundary and none of the functionality they provide has been tested as part of this evaluation.

[12] TSF – TOE Security Functionality

- The management interfaces for the hypervisor are not included within the TOE boundary and should be considered part of the IT environment.
- The Intelligent Platform Management Interface (IPMI) and SSH access to the CVM are excluded from the evaluation.
- The cryptography used in the HTTPS connections for the management via GUI and CLI has not been tested as part of this evaluation.
- The data efficiency claims in section 1.3.3 have not been explicitly evaluated as part of this evaluation.
- Physical disks.

2 Conformance Claims

This section and Table 3 provide the identification for any 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 conformant; CC Part 3 conformant; Parts 2 and 3 Interpretations of the CEM[13] as of 2013-06-21 were reviewed, and no interpretations apply to the claims made in this ST.
PP Identification	None
Evaluation Assurance Level	EAL2+ conformant augmented with ALC_FLR.2

[13] CEM – Common Evaluation Methodology

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[14] 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.)
- Natural threats: There are threats to the TOE Security Functionality (TSF) that are a natural byproduct of the systems that compose the TOE, such as electromagnetic interference on a line during transmission of user data.

All are assumed to have a low level of motivation. The IT assets requiring protection are the TSF[15] 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	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 by using one of the systems connected to the TOE.
T.NO_AUDIT	An TOE user or attacker may perform security-relevant operations on the TOE without being held accountable for them.

3.2 Organizational Security Policies

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

[14] IT – Information Technology
[15] TSF – TOE Security Functionality

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.NOEVIL	It is assumed that the administrators who manage the TOE are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.
A.LOCATE	It is assumed that the TOE is located within a controlled access facility and is physically available to authorized administrators only.
A.CONNECTIVITY	It is assumed that the IT Environment will be configured in such a way as to allow TOE users to access the information stored on the TOE.
A.TIME	It is assumed that the IT Environment will provide the time for the TOE from a reliable source.
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.

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 provide a method for administrative users to manage the TOE.
O.USER_DATA	The TOE must prevent unauthorized modifications to configuration and user data that it has been entrusted to protect.
O.AUDIT	The TOE must record events of security relevance at the "not specified" level of audit. The TOE must provide authorized administrators with the ability to review the audit trail in order to identify when misconfigurations have occurred.
O.FAULT_TOLERANCE	The TOE must be resilient against node or disk failures that might affect the security of the information it contains.
O.AUTHENTICATE	The TOE must authenticate administrative users before granting them access to TOE functionality that can affect the enforcement of security functionality provided by the TOE. Administrative users must use secure credentials to access TOE functionality.

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.PROPER_NAME_ASSIGNMENT	Each VM within the TOE Environment must provide accurate unique server identifiers for each server (running as a guest VM) that accesses storage on the TOE.
OE.SECURE_COMMUNICATION	The TOE Environment must provide un-tampered communications

Name	Description
	between systems connected to the TOE.
OE.CONNECT	The TOE administrators will configure the IT Environment so that users have proper network support to be able to access data on the TOE.
OE.TIME	The TOE Environment must ensure that the time is provided to the TOE from a reliable source.
OE.INTERNAL_STORAGE_NETWORK	The TOE environment must limit access to the TOE from external entities such that only internal hosts can access the NFS storage functionality provided by the TOE.

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

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].
- 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 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_SAR.1	Audit review		✓		
FDP_ACC.1	Subset access control		✓		
FDP_ACF.1	Security attribute based access control		✓		
FDP_IFC.1	Subset information flow control		✓		
FDP_IFF.1	Simple security attributes		✓		
FIA_ATD.1	User attribute definition		✓		
FIA_SOS.1	Verification of secrets		✓		
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	✓	✓		

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		✓		
FRU_FLT.2	Limited fault tolerance		✓		
FTA_SSL.4	User-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) [*all configuration changes made via the Web Console and nCLI related to management of the Virtual Disk Access SFP, management of user accounts, management of containers, management of virtual disks, and management of virtual machines*].

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, [*no other information*].

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 [*Web Console administrative users*] with the capability to read [*all 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.

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 [*Virtual Disk Access SFP*] on [
Subjects:
- *Virtual Servers*[16]

Objects:
- *NFS share*

].

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 [*Virtual Disk Access SFP*] to objects based on the following: [
Subject (Virtual Server) attributes:
- *Name*
- *Host ID*

Object (NFS share) attributes:
- *Name*
- *Maximum Capacity*
- *NFS whitelist*

].
FDP_ACF.1.2
The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*If the host is on the NFS whitelist, then access is allowed Otherwise, access is denied*].
FDP_ACF.1.3
The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*no other rules*].
FDP_ACF.1.4
The TSF shall explicitly deny access of subjects to objects based on the [*If the maximum capacity is reached, access is denied*].

FDP_IFC.1 Subset information flow control
Hierarchical to: No other components.
Dependencies: FDP_IFF.1 Simple security attributes
FDP_IFC.1.1
The TSF shall enforce the [*Virtual Disk Locking SFP*] on [
Subjects:
- *Clients*[17]

Information:
- *Virtual Disks*

Operations:
- *write*
- *execute*

].

[16] Virtual servers are the guest VMs running on the TOE that access the storage provided by the TOE.
[17] Clients are processes on guest VMs that access storage provided by the TOE.

FDP_IFF.1 Simple security attributes
Hierarchical to: No other components.
Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialization

FDP_IFF.1.1

The TSF shall enforce the [*Virtual Disk Locking SFP*] based on the following types of subject and information security attributes: [

Subject (Processes) attributes:

- *Process ID*
- *Hostname*
- *Host IP*[18] *address*
- *Idle time*

Information attributes:

- *Virtual Disk ID*
- *Virtual disk lock*

].

FDP_IFF.1.2

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [*If the process (identified by process ID, hostname, and host IP address) is designated in the virtual disk lock, access is allowed. Otherwise, access is denied*].

FDP_IFF.1.3

The TSF shall enforce the [*If the virtual disk does not currently have a virtual disk lock issued, the process may obtain a virtual disk lock from a leader node*[19]*. If the process idle time is 10 minutes then the disk lock is released*].

FDP_IFF.1.4

The TSF shall explicitly authorize an information flow based on the following rules: [*no other rules*].

FDP_IFF.1.5

The TSF shall explicitly deny an information flow based on the following rules: [*no other rules*].

[18] IP – Internet Protocol
[19] A leader node is a node in the cluster that is responsible for issuing virtual disk locks.

6.2.4 Class FIA: Identification and Authentication

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, 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 [*at least eight characters in total length*].[20]

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 [*obscured feedback via the Web Console*] to the user while the authentication is in progress.

Application note: FIA_UAU.7 only applies to the Web Console. The nCLI interface does not obscure the password as it is typed in by an administrative user.

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.

[20] It should be noted that this requirement applies only to the Web Console and nCLI interfaces.

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 [*Virtual Disk Access SFP*] to restrict the ability to [change_default, query, modify] the security attributes [*type, tiering options, maximum capacity*] to [*administrative users*].

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 [*Virtual Disk Access 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 [*administrative users*] 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 [query, modify, delete] the [*user accounts, containers, and virtual disks*] to [*administrative users*].

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: [*configuration of the following functions*:

- *Virtual Disk Access SFP attributes*
- *user accounts*
- *storage*
- *virtual machines (view statistics)*
- *management of the system time*

].

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 [*User Administrator, Cluster Administrator*[21]] **for the Web Console and nCLI**.

FMT_SMR.1.2

[21] An administrative user can have one or both of these roles.

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: [

- *failure of a single node in a multi-node[22] configuration,*
- *failure of up to all disks on a single node in a multi-node configuration.*

].

[22] Multi-node refers to configurations with two or more nodes installed on a single system.

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 single node in a multi-node configuration,*
- *failure of up to all disks on a single node in a multi-node configuration*

].

6.2.8 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.

6.3 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3 and are EAL2+. Table 10 Assurance Requirements summarizes the requirements.

Table 10 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 Basic 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 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 11 lists the security functionality and their associated SFRs.

Table 11 Mapping of TOE Security Functionality to Security Functional Requirements

TOE Security Functionality	SFR ID	Description
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_SAR.1	Audit review
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
	FDP_IFC.1	Subset information flow control
	FDP_IFF.1	Simple security attributes
Identification and Authentication	FIA_ATD.1	User attribute definition
	FIA_SOS.1	Verification of secrets
	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_MTD.1	Management of TSF data
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.1	Security roles
Protection of the TSF	FPT_FLS.1	Failure with preservation of secure state
Resource Utilization	FRU_FLT.2	Limited fault tolerance
TOE Access	FTA_SSL.4	User-initiated termination

7.1.1 Security Audit

The TOE records audits for each administrative user action on the nCLI and Web Console. These audit records can only be viewed by administrative users via the Web Console. Audit is started upon startup of the TOE and does not halt until the TOE is shutdown. Although the TOE does not audit the startup and shutdown of the audit function, it does audit the startup and shutdown of the TOE, thereby indicating when the audit function is started and stopped as well.

The TOE audit records contain the information listed in Table 12.

Table 12 Audit Record Contents

Field	Content
timestamp	The date and time that the event occurred.
Userid	The subject (user) who performed the action.
Action performed	A description of the action, including the outcome (success or failure) and the event type.

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

7.1.2 User Data Protection

Storage for the cluster is provisioned as units called containers which are created from one or more tiers of disk storage (storage pools). The TOE can provide access to containers via NFS shares, which provide access to storage to hosts on the network.

The TOE implements a Virtual Disk Access SFP that controls what storage virtual servers can access on the TOE. This SFP controls access based on an NFS whitelist stored on the TOE. Additionally, each NFS share is allocated a certain amount of storage space that, once reached, results in users not being able to access additional storage.

The TOE enforces a Virtual Disk Locking SFP. The Virtual Disk Locking SFP allocates access to Virtual Disks via a mechanism called *virtual disk locking*. Virtual disk locking occurs when a process on a virtual server requests access to storage represented by a virtual disk from the leader. If the virtual disk is currently being accessed by a different process, then the TOE denies access to the requesting process until the current client goes inactive for ten minutes. If the virtual disk is not currently locked, then the leader issues a lock specifying the process ID, hostname, and host IP address of the client. The lock allows exclusive access to the virtual disk until the client goes idle (stops sending requests) for ten minutes. The lock is automatically extended if the client becomes active again.

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

7.1.3 Identification and Authentication

The TOE stores attributes for administrative users locally. These attributes include the username and (obscured for the Web Console) password for each locally-stored administrative user. The TOE requires Web Console and nCLI administrative users to use secure passwords. Secure passwords for these interfaces are defined as being at least eight characters in length.

Administrative users must identify and authenticate themselves to the TOE before being granted access to any of the management functionality provided via the Web Console and nCLI. Passwords are obfuscated when being typed into a login prompt on the Web Console.

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

7.1.4 Security Management

All management of the TOE functionality takes place through the Web Console and nCLI. The Web Console offers various pages for user management, guest VM management (for viewing statistics), and storage management. Likewise, the nCLI offers a set of commands for managing these features. The NFS whitelist can be managed to permit access to NFS shares on the storage system (this access is restricted by default). The roles available from the management interfaces are the same for the nCLI and Web Console; they are User Administrator, and Cluster Administrator. Administrative users can assume both roles simultaneously.

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

7.1.5 Protection of the TSF

In the event of a node or disk failure, the TOE maintains a secure state by continuing to offer all of its functionality in the event of:

- failure of a single node in a multi-node system,
- failure of up to all disks on a node in a multi-node system.

This is possible because the TOE stores metadata for each virtual disk on three different nodes, and data for each virtual disk on two different nodes for full-node redundancy. Additionally, the TOE stripes data across RAID 1 arrays that are setup on the file system, preventing data loss from the failure of a single disk.

TOE Security Functional Requirements Satisfied: FPT_FLS.1.

7.1.6 Resource Utilization

The TOE duplicates virtual disk data across multiple nodes to provide redundancy in the event of:

- failure of a single node in a multi-node system,
- failure of up to all disks on a node in a multi-node system.

This allows the TOE to remain fully operational in the event that one of these components fails.

TOE Security Functional Requirements Satisfied: FRU_FLT.2.

7.1.7 TOE Access

The TOE provides the ability for administrative users to terminate their sessions via the management interfaces. This can be accomplished through the Web Console by clicking the “logout” button,and through the nCLI by typing the `exit` command.

TOE Security Functional Requirements Satisfied: FTA_SSL.4.

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. There are no extended SFRs or SARs contained within this ST. There are no protection profile claims for this ST.

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

Table 13 Threats: Objectives Mapping

Threats	Objectives	Rationale
T.DATA_CORRUPTION User data and configuration data could become corrupted due to hardware failure or incorrect system operations.	O.ADMIN The TOE must provide a method for administrative users to manage the TOE.	O.ADMIN counters this threat by allowing administrative users to properly configure the mechanisms of the TOE that prevent data corruption.
	O.USER_DATA The TOE must prevent unauthorized modifications to configuration and user data that it has been entrusted to protect.	O.USER_DATA counters this threat by providing mechanisms to protect the configuration and user data that has been entrusted to the TOE against unauthorized modifications as a result of race conditions.
	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 counters this threat by ensuring that the TOE is capable of maintaining a secure state and offering its full set of functionality in the event of a node or disk failure.
T.IMPROPER_SERVER A TOE user or attacker could attempt to bypass the access controls provided by the TOE by using one of the systems connected to the TOE.	O.ADMIN The TOE must provide a method for administrative users to manage the TOE.	O.ADMIN counters this threat by allowing administrative users to properly configure the mechanisms of the TOE designed to control the Access and Information Flow Control Policies.
	OE.PROPER_NAME_ASSIGNMENT Each VM within the TOE	OE.PROPER_NAME_ASSIGNMENT counters this threat by ensuring that the unique server

Threats	Objectives	Rationale
	Environment must provide accurate unique server identifiers for each server (running as a guest VM) that accesses storage on the TOE.	identifiers provided to the TOE are accurate. This allows the mechanisms provided by O.PROTECT to properly protect data.
	O.USER_DATA The TOE must prevent unauthorized modifications to configuration and user data that it has been entrusted to protect.	O.USER_DATA counters this threat by providing adequate mechanisms to give only authorized servers access to configuration data.
	OE.SECURE_COMMUNICATIO N The TOE Environment must provide un-tampered communications between systems connected to the TOE.	OE.SECURE COMMUNICATIONS counters this threat by ensuring that all communications with the TOE are un-tampered for administration of the TOE, internal TOE communications, and data sent to or from the TOE.
	O.AUTHENTICATE The TOE must authenticate administrative users before granting them access to TOE functionality that can affect the enforcement of security functionality provided by the TOE. Administrative users must use secure credentials to access TOE functionality.	O.AUTHENTICATE counters this threat by ensuring that administrative users are authenticated before allowing access to TOE management functionality. This objective also ensures that strong credentials are used for administrative user login.
T.NO_AUDIT An 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 at the "not specified" level of audit. The TOE must provide authorized administrators with the ability to review the audit trail in order to identify when misconfigurations have occurred.	O.AUDIT counters this threat by ensuring that an audit trail of management events and alerts on the TOE is preserved, and that accurate timestamps are provided for all audit records, allowing the order of events to be preserved.

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

Table 14 Assumptions: Objectives Mapping

Assumptions	Objectives	Rationale
A.NOEVIL It is assumed that the administrators who manage the TOE are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.	OE.NOEVIL Sites using the TOE shall ensure that TOE administrators are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.	OE.NOEVIL upholds this assumption by ensuring that administrative users are not careless, negligent, or willfully hostile, are appropriately trained, and follow all guidance.
A.LOCATE It is assumed that the TOE is located within a controlled access facility and is physically available to authorized administrators only.	OE.PHYSICAL The TOE will be used in a physically secure site that protects it from interference and tampering by un-trusted subjects.	OE.PHYSICAL upholds this assumption by ensuring that physical security is provided for the TOE.
A.CONNECTIVITY It is assumed that the IT Environment will be configured in such a way as to allow TOE users to access the information stored on the TOE.	OE.CONNECT The TOE administrators will configure the IT Environment so that users have proper network support to be able to access data on the TOE.	OE.CONNECT upholds this assumption by ensuring that the IT Environment is configured appropriately to allow users to access information stored on the TOE.
A.TIME It is assumed that the IT Environment will provide the time for the TOE from a reliable source.	OE.TIME The TOE Environment must ensure that the time is provided to the TOE from a reliable source.	OE.TIME upholds this assumption by ensuring that the time will be provided to the TOE from a reliable source.
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.	OE.INTERNAL_STORAGE_NETWORK The TOE environment must limit access to the TOE from external entities such that only internal hosts can access the NFS storage functionality provided by the TOE.	OE.INTERNAL_STORAGE_NETWORK upholds this assumption by ensuring that only internal hosts can access the NFS 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.	OE.INTERNAL_USERS Sites using the TOE shall ensure that internal users are not careless, negligent, or willfully hostile.	OE.INTERNAL_USERS upholds this assumption by ensuring that the internal users accessing TOE storage are not careless, negligent, or willfully hostile.

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

Table 15 Objectives: SFRs Mapping

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN The TOE must provide a method for administrative users to manage the TOE.	FIA_ATD.1 User attribute definition	This requirement supports O.ADMIN by ensuring that administrative user attributes are maintained by the TOE.
	FMT_MSA.1 Management of security attributes	This requirement supports O.ADMIN by specifying the security attributes of the TOE that can be modified and which administrative roles can modify them.
	FMT_MSA.3 Static attribute initialisation	This requirement supports O.ADMIN by specifying that restrictive values are used by the access controls enforced by the TOE and specifying the administrative roles that can set alternate values.
	FMT_MTD.1 Management of TSF data	This requirement supports O.ADMIN by specifying what roles can operate on TSF data contained in the TOE configuration.
	FMT_SMF.1 Specification of Management Functions	This requirement supports O.ADMIN by specifying each of the management functions that

Objective	Requirements Addressing the Objective	Rationale
		are used to securely manage the TOE. These functions are provided by the TOE management interfaces.
	FMT_SMR.1 Security roles	This requirement supports O.ADMIN by specifying the administrative roles defined to govern management of the TOE.
	FTA_SSL.4 User-initiated termination	This requirement supports O.ADMIN by providing administrative 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 at the "not specified" level of audit. The TOE must provide authorized administrators with the ability to review the audit trail in order to identify when misconfigurations have occurred.	FAU_GEN.1 Audit Data Generation	This requirement supports O.AUDIT by requiring the TOE to produce audit records for the system security events.
	FAU_SAR.1 Audit review	This requirement supports O.AUDIT by requiring the TOE to make the recorded audit records available for review.
O.AUTHENTICATE The TOE must authenticate administrative users before granting them access to TOE functionality that can affect the enforcement of security functionality provided by the TOE. Administrative users must use secure credentials to access TOE functionality.	FIA_SOS.1 Verification of secrets	This requirement supports O.AUTHENTICATE by requiring administrative users to define secure passwords.
	FIA_UAU.2 User authentication before any action	This requirement supports O.AUTHENTICATE by requiring TOE administrators to authenticate their claimed identities before the TOE will perform any actions on their behalf via the management interfaces.
	FIA_UAU.7 Protected authentication feedback	This requirement supports O.AUTHENTICATE by preventing passwords from being read while typing them into the login prompts for the TOE management interfaces.
	FIA_UID.2 User identification before any action	This requirement supports O.AUTHENTICATE by requiring administrators to identify themselves before the TOE will perform any actions on their behalf.

Objective	Requirements Addressing the Objective	Rationale
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 supports O.FAULT_TOLERANCE by ensuring that the TOE maintains a secure state in the event of a disk or node failure.
	FRU_FLT.2 Limited fault tolerance	This requirement supports O.FAULT_TOLERANCE by ensuring that the TOE does not lose any functionality in the event of a disk or node failure.
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 supports O.USER_DATA by enforcing an access control policy that ensures that only authorized devices gain access to user and configuration data within the TOE.
	FDP_ACF.1 Security attribute based access control	This requirement supports O.USER_DATA by providing access control functionality to manage access to user and configuration data within the TOE.
	FDP_IFC.1 Subset information flow control	This requirement supports O.USER_DATA by enforcing an information flow control policy that ensures that access to user data is granted in a controlled manner to prevent data anomalies.
	FDP_IFF.1 Simple security attributes	This requirement supports O.USER_DATA by providing information flow control functionality to manage data flows to user data within the TOE.

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. The TOE 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 and applicable PPs. Table 16 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 16 Functional Requirements Dependencies

SFR ID	Dependencies	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	✓	Although FPT_STM.1 is not claimed, the TOE acquires the time from a trusted NTP server in the TOE environment.
FAU_SAR.1	FAU_GEN.1	✓	
FDP_ACC.1	FDP_ACF.1	✓	
FDP_ACF.1	FDP_ACC.1	✓	
	FMT_MSA.3	✓	
FDP_IFC.1	FDP_IFF.1	✓	
FDP_IFF.1	FDP_IFC.1	✓	
	FMT_MSA.3	✓	There is no management available for the information flow control policy beyond the automatic assignment, release, and renewal of virtual disk locks. Therefore, FMT_MSA.3 does not need to be met for this requirement.
FIA_ATD.1	None	Not applicable	
FIA_SOS.1	None	Not applicable	
FIA_UAU.2	FIA_UID.1	✓	Although FIA_UID.1 is not claimed, FIA_UID.2, which is hierarchical to FIA_UID.1, is.
FIA_UAU.7	FIA_UAU.1	✓	Although FIA_UAU.1 is not claimed, FIA_UAU.2, which is hierarchical to FIA_UAU.1, is.
FIA_UID.2	None	Not applicable	
FMT_MSA.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	

SFR ID	Dependencies	Dependency Met	Rationale
	FDP_ACC.1	✓	
FMT_MSA.3	FMT_MSA.1	✓	
	FMT_SMR.1	✓	
FMT_MTD.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_SMF.1	None	Not applicable	
FMT_SMR.1	FIA_UID.1	✓	Although FIA_UID.1 is not claimed, FIA_UID.2, which is hierarchical to FIA_UID.1, is.
FPT_FLS.1	None	Not applicable	
FRU_FLT.2	FPT_FLS.1	✓	
FTA_SSL.4	None	Not applicable	

9 Acronyms

This section and Table 17 define the acronyms used throughout this document.

Table 17 Acronyms

Acronym	Definition
CC	Common Criteria
CEM	Common Evaluation Methodology
CLI	Command Line Interface
CM	Configuration Management
CVM	Controller Virtual Machine
DAS	Direct Attached Storage
DRS	Distributed Resource Scheduling
EAL	Evaluation Assurance Level
GbE	Gigabit Ethernet
GFS	Google File System
GUI	Graphical User Interface
HA	High Availability
HDD	Hard Disk Drive
HOT	Heat Optimized Tiering
HTTPS	Secure Hypertext Transfer Protocol
ID	Identifier
ILM	Information Lifecycle Management
I/O	Input/Output
IP	Internet Protocol
IPMI	Intelligent Platform Management Interface
IT	Information Technology
KVM	Kernel-based Virtual Machine
NAS	Network Attached Storage
nCLI	Nutanix CLI
NDFS	Nutanix Distributed File System
NFS	Network File System
NOS	Nutanix Operating System
PCIe	Peripheral Component Interconnect Express
PP	Protection Profile

Acronym	Definition
QoS	Quality of Service
RF	Replication Factor
SAR	Security Assurance Requirement
SFP	Security Functionality Policy
SFR	Security Functional Requirement
SP	Service Pack
SSD	Solid State Drive
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
U	Unit
VM	Virtual Machine

Prepared by:
Corsec Security, Inc.

13135 Lee Jackson Memorial Highway
Suite 220
Fairfax, VA 22033
United States of America

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