Clustered Data ONTAP® 8.2.1

Security Target

EVALUATION ASSURANCE LEVEL: EAL2+

NOVEMBER 7TH 2014 | VERSION 1.0

Prepared for:

NetApp, Inc.
495 East Java Drive

Sunnyvale, CA 94089
United States of America
Phone: +1 408 822 6000
http://www.netapp.com

Prepared by:

Corsec Security, Inc.
13135 Lee Jackson Mem. Highway
Suite 220
Fairfax, VA 22030
United States of America
Phone: +1 703 267 6050
http://www.corsec.com

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Clustered Data ONTAP® 8.2.1. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements, and the Information Technology (IT) Security Functions provided by the TOE which meet the set of requirements.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the ST organization. The TOE is the NetApp Clustered Data ONTAP® 8.2.1 Operating System, and will hereafter be referred to as the TOE or Data ONTAP throughout this document. The TOE includes the kernel operating system that supports multi-protocol services and advanced data management capabilities for consolidating and protecting data for enterprise applications and users as well as the hardware appliances on which it runs. The TOE includes a separate software-only management GUI[1] called the OnCommand System Manager. This GUI is used to manage the TOE security functionality (TSF). The TOE also includes a separate software-only monitoring and diagnostic component called the OnCommand Unified Manager (OCUM). OCUM allows administrators to quickly identify and troubleshoot problems that arise in the monitored storage cluster.

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 TSF 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 conformance 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] GUI – Graphical User Interface

1.2 SECURITY TARGET AND TOE REFERENCES

Table 1 – ST and TOE References

ST Title	NetApp, Inc. Clustered Data ONTAP® 8.2.1 Security Target
ST Version	Version 1.0
ST Author	Corsec Security, Inc.
Publication Date	2014-11-07
TOE Reference	NetApp Clustered Data ONTAP® 8.2.1, including Data ONTAP 8.2.1 Software, FAS or V-Series Appliance (as specified in Section 1.5.1.2), OnCommand™ System Manager 3.1, and OnCommand™ Unified Manager 6.1R1.

1.3 PRODUCT OVERVIEW

The Product Overview provides a high-level description of the product that is the subject of the evaluation.

Clustered Data ONTAP® 8.2.1 is a proprietary operating system developed by NetApp. Clustered Data ONTAP® 8.2.1 provides data management functions that include providing secure data storage and multi-protocol access.

Clustered Data ONTAP® 8.2.1 is distributed with the following NetApp storage solution products:

- **FAS** – NetApp's FAS systems offer seamless access to a full range of enterprise data for users on a variety of platforms. FAS systems support NFS[2] and CIFS[3] for file access, as well as FCP[4] and iSCSI[5] for block-storage access.
- **V-Series** – The V-Series product family provides unified NAS[6] and SAN[7] access to data stored in FC SAN storage arrays enabling data center storage deployment.

V-Series and FAS products use the same hardware controller and run the same Clustered Data ONTAP® 8.2.1 operating systems. The key difference between a V-Series system front-ending a storage array and a FAS system with NetApp disks is that the V-Series controller no longer runs Redundant Array of Independent Disks (RAID) 4 or RAID-DP®[8]. Instead, the V-Series system offloads the RAID protection to the storage array. V-Series storage pools are large RAID 0 stripe sets of iSCSI or FC[9] Logical Unit Numbers (LUNs).

As shown in Figure 1, a typical clustered ONTAP system consists of two or more individual NetApp® storage controllers (including V-Series) with attached disks. The storage clusters are also called nodes. The basic building block is the High Availability (HA) pair. An HA pair consists of two identical controllers; each controller actively provides data services and has redundant cabled paths to the other controller's disk storage. If either controller is down for any planned or unplanned reason, its HA partner can take over its storage and maintain access to the data. When the downed system rejoins the cluster, the partner will give back the storage resources. A single node cluster is a special implementation of a cluster running on a standalone node. In a single node cluster, the HA mode is set to standalone. This configuration does not require a cluster network, and enables you to use the cluster ports to serve data traffic. In this document, the HA pair is usually not shown, for clarity.

Multiple HA pairs are combined together into a cluster to form a shared pool of physical resources available to applications, SAN hosts, and NAS clients. The shared pool appears as a single system image for

[2] NFS – Network File System
[3] CIFS – Common Internet File System
[4] FCP – Fibre Channel Protocol
[5] iSCSI – Internet Small Computer System Interface
[6] NAS – Network-Attached Storage
[7] SAN – Storage Area Network
[8] RAID-DP – A NetApp proprietary Double Parity RAID 6 implementation that prevents data loss when two drives fail
[9] FC – Fibre Channel

management purposes. This means there is a single common point of management, whether through GUI or CLI tools, for the entire cluster. While the members of each HA pair must be the same controller type, the cluster can consist of heterogeneous HA pairs. Over time, as the cluster grows and new controllers are released, it is likely to evolve into a combination of several different node types. All cluster capabilities are supported, regardless of the underlying controllers in the cluster.

A Clustered Data ONTAP® 8.2.1 system can scale from one to 24 nodes, supporting up to 61.4 PiB[10] of raw drive capacity. A cluster hosts virtualized storage systems called Vservers. Vservers provide SAN and NAS data access to hosts and clients.

Figure 1 also shows the underlying network architecture of clustered Data ONTAP. Three networks are shown:

- **Cluster interconnect** – A 10 Gbps[11], private, dedicated, redundant, high-throughput network used for communication between the cluster nodes and for data motion within the cluster. The cluster interconnect infrastructure is provided with every Clustered Data ONTAP® 8.2.1 configuration to support this network.
- **Management network** – All management traffic passes over this network. Management network switches can be included in a clustered Clustered Data ONTAP® 8.2.1 configuration, or customer-provided switches can be used. OnCommand™ System Manager is available for management, and configuration of clustered ONTAP systems. OnCommand™ System Manager provides GUI management, including a number of easy-to-use wizards for common tasks. In addition, a CLI is available.
- **Diagnostic tools** – OCUM is a separate storage monitoring and diagnostic interface designed to give administrators an overview of cluster health from a graphical dashboard. Using OCUM, administrators can assess the overall capacity, availability, and protection health of the managed storage clusters. Using this information, administrators can locate, diagnose, and troubleshoot any issues that arise within the storage cluster. OCUM provides the same capabilities via a callable API that third party applications can use for integration. OCUM also provides a CLI called the maintenance console that allows administrators to monitor, diagnose, and resolve operating system issues, version upgrade issues, user access issues, and network issues related to the OCUM server itself. The maintenance console is even available when the graphical dashboard interface is down.
- **Data networks** – Provide data access services over Ethernet or Fibre Channel to the SAN hosts and NAS clients. These networks are customer provided according to requirements and could also include connections to other clusters acting as volume replication targets for data protection.

[10] PiB - Pebibyte
[11] Gbps – gigabits per second

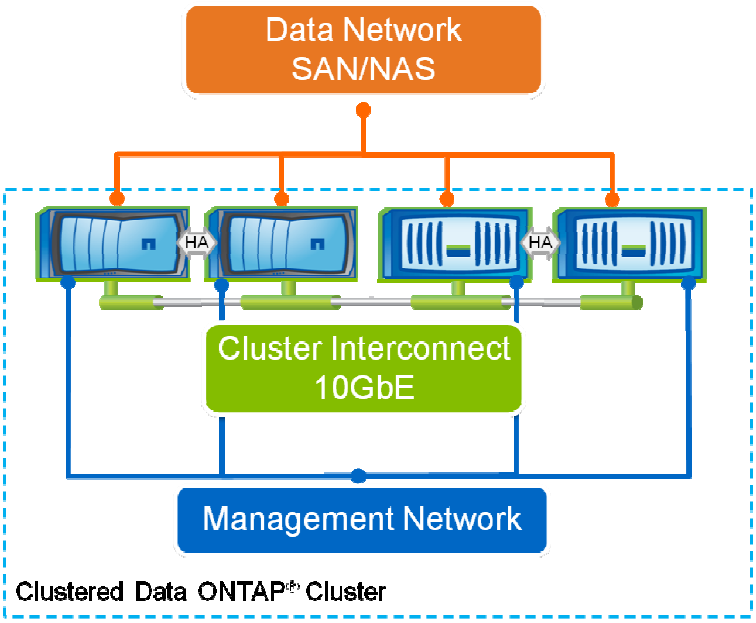

Figure 1 – Clustered Data ONTAP overview

Storage controllers, while they can be of different types, are by default considered equivalently in the cluster configuration in that they are all presented and managed as cluster nodes. Individual disks are managed by defining them into aggregates: groups of disks of a particular type that are protected using NetApp RAID-DP®. Network interface cards and HBAs[12] provide physical ports (Ethernet and Fibre Channel) for connection to the management and data networks. The physical components are visible only to cluster administrators, and not directly to the applications and hosts that are using the cluster. The physical components constitute a pool of resources from which are constructed the logical cluster resources. Applications and hosts access data only through Vservers that contain volumes and logical interfaces.

The primary logical cluster component is the Vserver. Clustered ONTAP supports from one to hundreds of Vservers in a single cluster. Each Vserver is configured for the client and host access protocols it will support – any combination of SAN and NAS. Each Vserver contains at least one volume and at least one logical interface. The administration of each Vserver can also be delegated if desired, so that separate administrators could be responsible for provisioning volumes and other Vserver–specific operations. This is particularly appropriate for multi-tenanted environments or where workload separation is desired.

For more information on NetApp Storage Controllers, see section 1.5.1.2. The products support both single controller and High Availability controller pairs as Storage Controller options on some models.

Clustered Data ONTAP® 8.2.1 supports multiple authentication mechanisms:

- For CIFS sharing, Clustered Data ONTAP® 8.2.1 can authenticate end users with Kerberos[13] or New Technology Local Area Network Manager (NTLM)† against an Active Directory (AD) domain, with NTLM† against an Windows NT-style domain, or locally using NT-style NTLM authentication against a local user database.
- For NFS sharing, the TOE can authenticate end users with Kerberos against both an Active Directory domain and a Network Information Service (NIS) domain, or locally against User Identifiers (UID) and passwords in local UNIX identity stores and /etc/passwd/.
- For administration, the TOE authenticates administrators against a local user repository.

In addition to the above interfaces and functionality, Data ONTAP provides storage policy automation in the form of the Workflow Automation package. Workflow Automation allows storage administrators to predefine

[12] HBA – Host Bus Adapter

[13] Off-box Identification and Authentication to a NIS or AD domain via either NTLM or Kerberos is a functionality provided by the IT Environment. Identification and Authentication of end-users is not a claimed security functionality of the TOE whether local or remote.

common cloud-based storage workflows and enable them automatically, without having to go through the entire setup process each time. These workflows can automate common tasks, such as:

- provisioning, migrating, or decommissioning storage,
- setting up a new virtualization environment,
- setting up storage for an application as part of an orchestration process.

OnCommand Performance Manager (OPM) is the data acquisition and statistical analysis package for Unified Manager 6.1R1 designed for clustered Data ONTAP environments. OPM provides:

- automated detection of performance issues,
- analysis of the cause of the issues,
- alerting to inform administrators of the issues, and;
- recommendations on how to resolve the issues.

OPM monitors the performance of storage-specific resources (i.e. it does not monitor the performance of VMs, hosts, or apps). This package creates a continuous performance baseline that adjusts to workflow changes. Dynamic thresholds are used to adjust to changes in the infrastructure of the storage system. OPM identifies workloads that monopolize resources from other tasks so that allocations can be adjusted.

The following section, TOE Overview, will provide the introduction to the parts of the overall product offering that are specifically being evaluated.

For management of the common storage system functions, a browser based graphical user interface (GUI) called the OnCommand™ System Manager is used. Figure 2 shows a complete two-cluster deployment with the OnCommand™ System Manager.

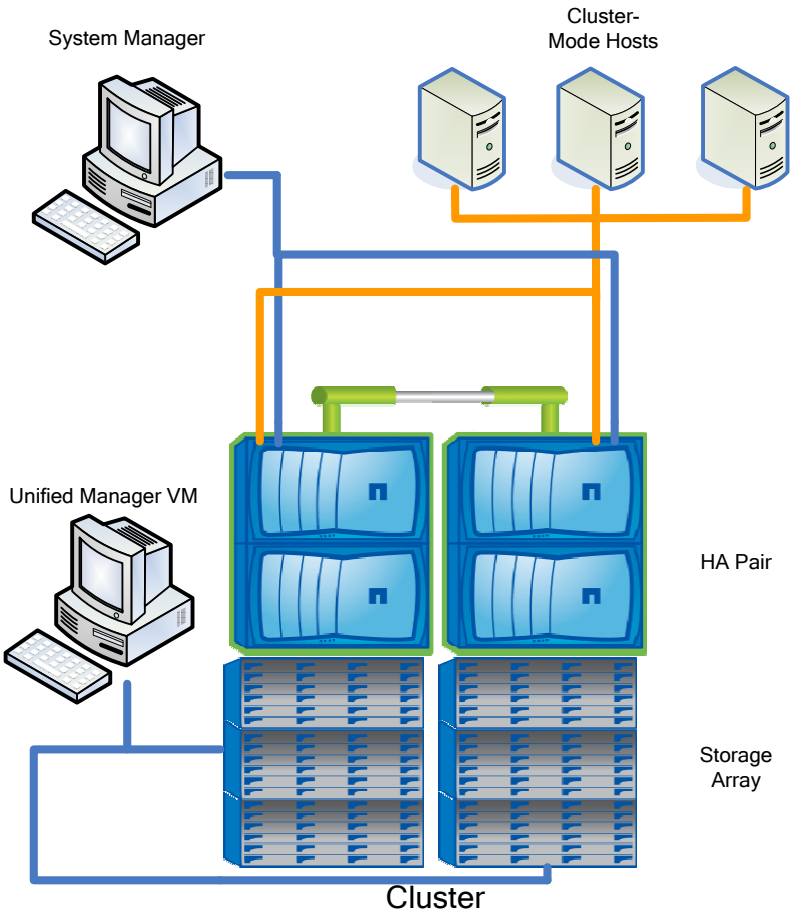

Figure 2 – Data ONTAP Two Cluster Deployment

See section 1.4.3 for the specified test environment configuration for System Manger. OCUM must be installed on a separate server running VMware ESXi v5.1 hypervisor as a Virtual Machine. This server must be present on the same management network.

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 product, and defining the specific evaluated configuration. The TOE is a data storage system. It is a hardware and software TOE. Functionality included in the logical software components of the TOE boundary includes:

- **Secure Multi-Protocol Data Storage Access** – Secure storage is provided by the TOE by implementing strict access control rules to data managed by the TOE. Multi-protocol access support is provided by the TOE by supporting both NFS and CIFS clients and providing transparent access to data.
- **Identification and Authentication** – The TOE supports on-box Identification and Authentication of administrators against a local user repository.
- **Domain Separation** – The TOE can function as a storage server for multiple groups of users within the TOE's control that must remain isolated from one another through the implementation of NetApp's Vserver and Storage Virtual Machine (SVM) technology.

- **Management** – The Management functionality included in the TOE's logical boundary enables users to modify TOE data and TSF security functional behavior.
- **Audit** – The Audit functionality provided by the TOE generates audit records for administrator logins and configuration changes.

1.4.1 Brief Description of the Components of the TOE

The software component of the Clustered Data ONTAP® 8.2.1 TOE is divided into six primary components: Write Anywhere File Layout® (WAFL), System Administration, the Operating System Kernel, Management Host, the OnCommand™ System Manager, and the OnCommand™ Unified Manager. The six components are described below. Their relationship to the IT Environment-supplied components is depicted in Figure 4.

- **WAFL** – The TOE's WAFL component is responsible for implementing the TOE's Discretionary Access Control (DAC) Security Function Policy (SFP). The DAC SFP includes enforcing access rules to user data based on client type, client security attributes, file types, file security attributes and access request (create, read, write, execute, delete, change permission, and change owner).
- **System Administration** – The System Administration component provides an administrator with an interface supporting operator functions including enforcing identification and authentication, user roles, and providing the necessary user interface commands that enable an operator to support the TOE's security functionality. The System Administration function is performed by a user with the *admin* role, and this functionality is available locally and remotely via a Command Line Interface (CLI), or remotely via one of several management interfaces detailed in section 7.1.4. System Administration functions are audited by default.
- **Operating System Kernel** – The Kernel facilitates communication between the components of the Operating System. The Kernel is a small portion of the operating system through which all references to information and all changes to authorizations must pass.
- **Management Host** – Host the management and services applications for the node. One of the functions of the Management Host (M-Host) is the Cluster Admin which is responsible for the CLI interface for the cluster and the Volume Location Data Base (VLDB) which locate the physical location of volumes in a node.
- **OnCommand™ System Manager** – The OnCommand™ System Manager component provides an authorized administrator with a web-based GUI that supports administrator functions including enforcing identification and authentication, user roles, and providing the necessary user interface commands that enable an authorized administrator to support the TOE's security functionality. The OnCommand™ System Manager GUI function is performed by a user with the admin role and provides remote management of the TSF. All security relevant actions within the OnCommand™ System Manager GUI are audited by default. The OnCommand™ System Manager GUI is a separate TOE component that must be installed on a management workstation.
- **OnCommand™ Unified Manager** – The OnCommand™ Unified Manager component provides an authorized administrator with a web-based GUI, a console interface, and an exposed API. Together, these interfaces provide an authorized administrator the ability to view the status for capacity, availability, and protection relationships of the monitored systems in the storage cluster.

1.4.1.1 WAFL Functionality Detail

The TOE's WAFL Component protects User data. The TOE uses the subject, subject's security attributes, the object, the object's security attributes and the requested operation to determine if access is granted. The subjects are end users on remote systems that access the TOE via NFS or CIFS. Figure 3 depicts the WAFL functionality.

The following acronyms not yet defined are used in Figure 3 below:

- ACL – Access Control List
- ACE – Access Control Entry
- GID – Group Identifier
- NTFS – New Technology File System
- SD – Security Descriptor
- SID – Security Identifier

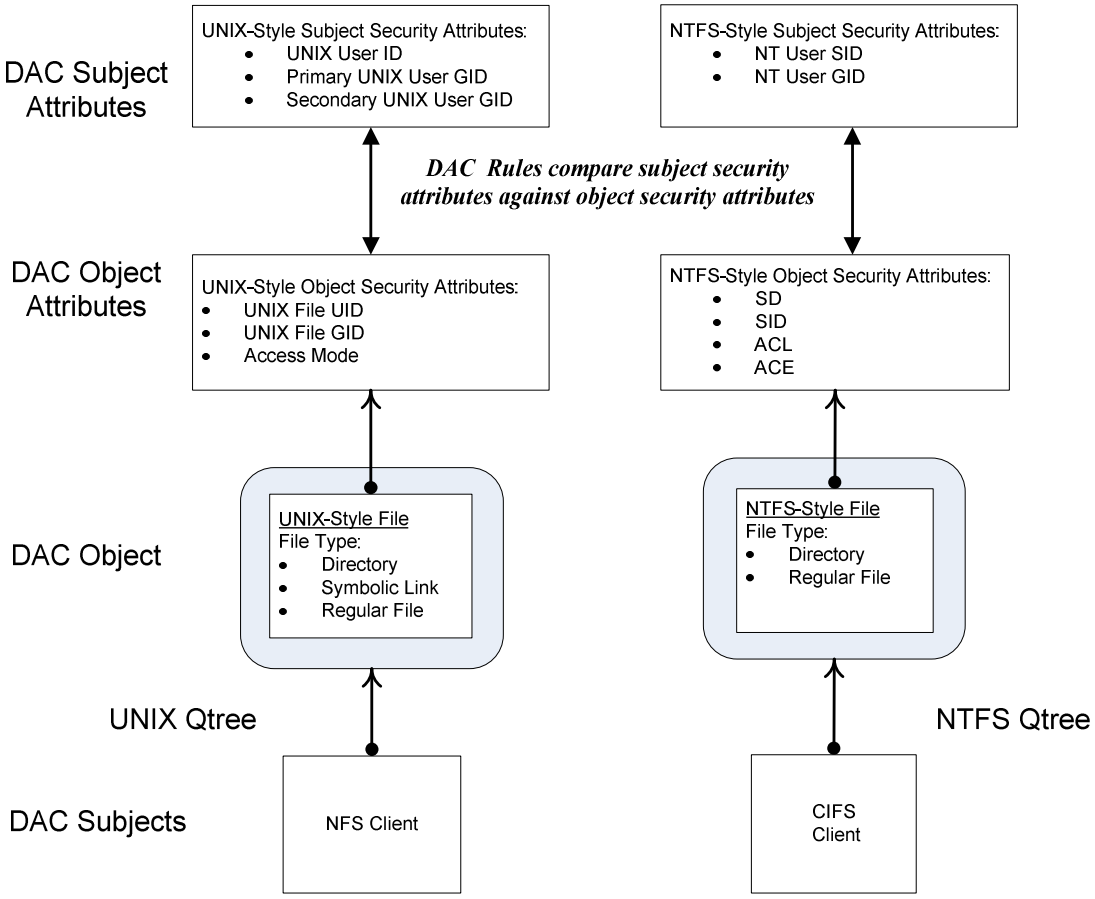

Figure 3 – WAFL Functionality Detail

1.4.1.1.1 User Data

The User Data that is covered by the DAC SFP are the user files on NetApp disks attached to an FAS appliance or SANs attached to a V-Series appliance. Each file maintained by the TOE has a file style associated with it. The TOE maintains three styles of files: NFSv3 UNIX-Style files, NFSv4 UNIX-Style files, and NTFS-Style files. NFSv3 UNIX-Style files have UNIX-Style security attributes, NFSv4 UNIX-Style files have NFSv4 security attributes, and NTFS-Style files have NTFS-Style security attributes.

In addition to a file style, each file has a file type. The file types may be directories, symbolic links, or regular files. UNIX-Style files may be a directory, a symbolic link or a regular file. NTFS-Style files do not have symbolic links; therefore the file type will be either a directory or a regular file.

A Qtree is a disk space partition. In addition to the file type, the TOE maintains three different storage types: UNIX Qtrees, NTFS Qtrees and Mixed Qtrees. UNIX Qtrees store UNIX-Style files with UNIX-Style security attributes. NTFS Qtrees store NTFS-Style files with NTFS Style security attributes. Mixed Qtrees store both styles of files. Files stored in Mixed Qtrees always have the security attributes associated with the client that was last used to change their access permissions or ownership. Mixed Qtrees are not part of the evaluated configuration.

A file's security attributes are determined when the file is created. The TOE will create UNIX-Style security attributes for a file stored in a UNIX Qtree. The TOE will create NTFS-Style security attributes for a file stored in an NTFS Qtree. These security attributes are outlined in Table 2 below.

Table 2 – TSF User Data Security Attribute Descriptions

Security Attribute	Description
Access Control Entry	A data structure associated with NTFS-Style files. Each ACE explicitly allows or denies access to a user or group for a specific NTFS-Style supported operation.
Access Control List	A data structure associated with NTFS-Style files. Each ACL includes one or more ACEs.
Access Mode	A data structure associated with a UNIX-Style Files. An access mode string is the last nine characters of a UNIX-Style File Permission string (drwxrwxrwx). The nine characters represent the access mode for the file in three sets of rwx triplets. The first triplet specifies the permission for the file's owner (UID). The next triplet specifies the permissions for the group associated with the file (UNIX file GID). The last three characters specify the permission for the users who are neither the owner nor members of the file's group (other). The rwx triplet identifies the permission for that set (owner, group, other). The three characters represent read, write or execute privileges. If the character is a dash, the set does not have permissions to perform the specific action.
File Permission String	A data structure associated with a UNIX-Style file. The file permission string is represented in ten characters common to all UNIX files (e.g. drwxrwxrwx). The first character contains one of three characters that identify the file type: d for directory, l for a symbolic link, or a dash (-) indicates the file is a regular file. The following 9 characters represent the access mode for the file in three sets of rwx triplets.
Security Descriptor	A data structure associated with NTFS-Style files. A SD contains a SID and an ACL.
Security Identifier	The CIFS User SID of the file's owner.
Group Identifier	A UNIX File GID identifies the groups associated with the UNIX-Style file.
User Identifier	The UNIX User UID of the file's owner.

1.4.1.1.2 TOE Clients

End-user access to TSF data is possible through the use of at either the NFS or CIFS client protocol. In a typical deployment as depicted in Figure 2 below, end user workstations or the file, web, mail, or application servers of the IT Environment connect to the TOE that hosts the TSF data residing on the storage arrays. The TOE is positioned between these workstations and servers, and the storage arrays, facilitating seamless NFS or CIFS connectivity between them while adding increased performance, efficiency, manageability, scalability, security, redundancy, and fault tolerance.

End-system workstations and the file, web, mail, or application servers authenticate with the TOE according to the operating procedures of the organization and IT Environment. Typical scenarios include the file, web, mail, or application servers prompting end users for credentials as they attempt to access a web page, e-mail system, or stand alone application or the TOE prompting end users for credentials as they attempt to access shared network directories (NFS or CIFS). The TOE facilitates server and end-user authentication of the end users attempting to access the TSF data via NFS or CIFS.

To determine if file access is allowed, the TOE compares a client's security attributes with the file's security attributes, listed in Table 3 below. The type of client security attributes (UNIX-Style or NTFS-Style) required by the TOE depends on the type of security attributes maintained by the file and the operation requested. The file or operation will require UNIX-Style subject security attributes (NFSv3 or NFSv4), NTFS-Style subject security attributes or both. If the file or operation requires UNIX-Style security attributes for a client, the TOE will attempt to obtain the client's UNIX User UID and UNIX User GID. If the file or operation requires NTFS-Style subject security attributes, the TOE will attempt to acquire the client's Windows User SID and a Windows User GID. Because of the native operating systems of the two clients, NFS clients are associated with UNIX-Style security attributes and CIFS Clients are associated with NTFS-Style security attributes.

The resolution of client security attributes is processed differently by the TOE for each type of client because the two protocols are different. NTFS-Style security attributes for a CIFS client are resolved when the CIFS client logs into the remote system and joins the Windows domain (of which the TOE is a member). Therefore, NTFS-Style security attributes for a CIFS client is completed before the TOE receives a CIFS

request. Alternatively, NFS client security attributes are resolved per NFS request. The UNIX User UID is passed in each NFS request and this UID is used to resolve the required client security attributes.

Table 3 – TOE Client Security Attribute Descriptions

Security Attribute	Description
Windows User SID	The Windows user ID number. Each user in a Windows system is assigned a unique Windows User SID.
Windows User GID	The Windows group ID number. Each user in a Windows system is assigned to a group and that group is assigned a unique GID.
UNIX User UID	The UNIX user ID number. Each user in a UNIX system is assigned a unique UNIX User UID.
UNIX User GID	The UNIX group ID number. Each user in an UNIX system is assigned to a group and that group is assigned a unique GID.

1.4.2 TOE Environment Hardware

The IT Environment Hardware includes the OnCommand™ System Manager & OnCommand™ Unified Manager Host System, called the Management Workstation. The Storage array, using FC, SAS, or SATA disk drives is also a required IT environment component. The product functionality provided by the FAS and V-Series products is supplied by the IT Environment.

1.4.3 TOE Environment Software

The following functionality is used by the TOE, however is not evaluated as part of the TOE:

- Browser Software, SNMPv3 Protocol

  The web browser used to access the Security Manager web interface and the SNMPv3 protocol used to communicate between the remote workstation and the TOE are supplied by the IT Environment. The OnCommand™ System Manager V2 does not support SNMPv3; as a result, the community string can be changed to use SNMPv1/v2c.

Before an authorized administrator begins the software setup process, he must ensure that the network and storage environment for the new storage system has been prepared according to the Guidance Documentation. For further information, refer to Section “Prerequisites to initial configuration” in the *Data ONTAP*® 8.2 *Software Setup Guide For Cluster-Mode*. The following sections must be referred to in the previously stated document:

- Requirements for the administration host
- High-availability (HA) requirements
- Requirements for Windows domains
- Requirements for Active Directory authentication
- Time services requirements
- Switch configuration requirements for interface groups
- DHCP requirements for remote access
- Managing feature licenses
- Requirements for creating array LUNs for V-Series systems
- V-Series system licensing requirements

Once the proper configuration has been met, the administrator must gather the appropriate configuration items from the network and storage environment and keep them handy for proper installation of the TOE. If the V-Series is ordered with native disks, the factory has pre-installed Clustered Data ONTAP® 8.2.1 software and licenses for the TOE administrator. If the system was ordered without native disks, the TOE administrator must install the Clustered Data ONTAP® 8.2.1 software and licenses after running the setup program.

System Requirements for OnCommand™ System Manager and OnCommand™ Unified Manager

The OnCommand™ System Manager and OnCommand™ Unified Manager can be hosted on wide variety of operating systems, and browsers. The OnCommand™ System Manager and OnCommand™ Unified Manager Host system must meet the following minimum requirements:

- Pentium x86 processor
- 1 GB RAM
- 1 GB video display RAM
- 1 GB free disk space
  If you are upgrading from an earlier version, you might require additional disk space for the existing log files.
- Wireless or Ethernet connection to the network
- A 32-bit or 64-bit Windows or Linux operating system
- Adobe Flash Player 11.0 or later
- 32-bit or 64-bit Oracle Java Runtime Environment (JRE) 7
  Installing 32-bit or 64-bit JRE depends on the operating system. If you have a 32-bit Windows or Linux operating system, 32-bit JRE must be installed. Similarly, if you have a 64-bit Windows or Linux operating system, 64-bit JRE must be installed.

A Windows Management Workstation must be running:

- Windows XP
- Windows Server 2003
- Windows Server 2008
- Windows Server 2012
- Windows Vista
- Windows 7
- Windows 8

A Linux system must be running one of the following:

- Red Hat Enterprise Linux 5 or 6
- SUSE Linux Enterprise Server 11

A Linux system must have a graphical desktop environment, such as GNOME or KDE, installed.

The web browser for OnCommand™ System Manager must be one of the following:

- Internet Explorer 8.0 and 9.0 (for Windows)
- Internet Explorer 10.0 in compatibility mode (for Windows)
- Mozilla Firefox 15, 16, 17, and 18 (for both Windows and Linux)
- Google Chrome 23 and 24 (for Windows)

The web browser for OnCommand™ Unified Manager must be one of the following:

- Internet Explorer 8.0, 9.0, and 10.0
- Mozilla Firefox 19 and 20
- Google Chrome 25 and 26

Note: You can run either a 32-bit browser or a 64-bit browser on a 64-bit operating system.

See the NetApp Interoperability Matrix Tool for the latest versions: http://now.netapp.com/matrix/ Note: This web site requires a login to view the matrix.

1.5 TOE DESCRIPTION

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

1.5.1 Physical Scope

Figure 4 illustrates the physical scope and the physical boundary of the overall solution, its deployment in a networked environment, and ties together all of the components of the TOE and the constituents of the TOE Environment. The essential physical component for the proper operation of the TOE in the evaluated configuration is the Clustered Data ONTAP® 8.2.1 operating system and the NetApp Appliance Hardware. The Clustered Data ONTAP® 8.2.1 operating system consists of:

- **WAFL** – TOE's WAFL component is responsible for implementing the TOE's DAC SFP.
- **System Administration** – Service supporting management of the node.
- **Operating System Kernel** – Provides messaging between individual components of Data ONTAP
- **Management Host** – An administrator may directly communicate to the Cluster through a connection to the Management Host. When commands are directed to a node on another storage controller, the request is forwarded to the specific node's System Administration service through the Management Network shown in Figure 1.
- **OnCommand™ System Manager** – The OnCommand™ System Manager component provides an authorized administrator with a web-based GUI that supports administrator functions.
- **OnCommand™ Unified Manager** – The OnCommand™ Unified Manager component provides an authorized administrator with a web-based GUI, a console interface, and an exposed API. Together, these interfaces provide an authorized administrator the ability to view the status for capacity, availability, and protection relationships of the monitored systems in the storage cluster.

The TOE components are depicted in Figure 4 below.

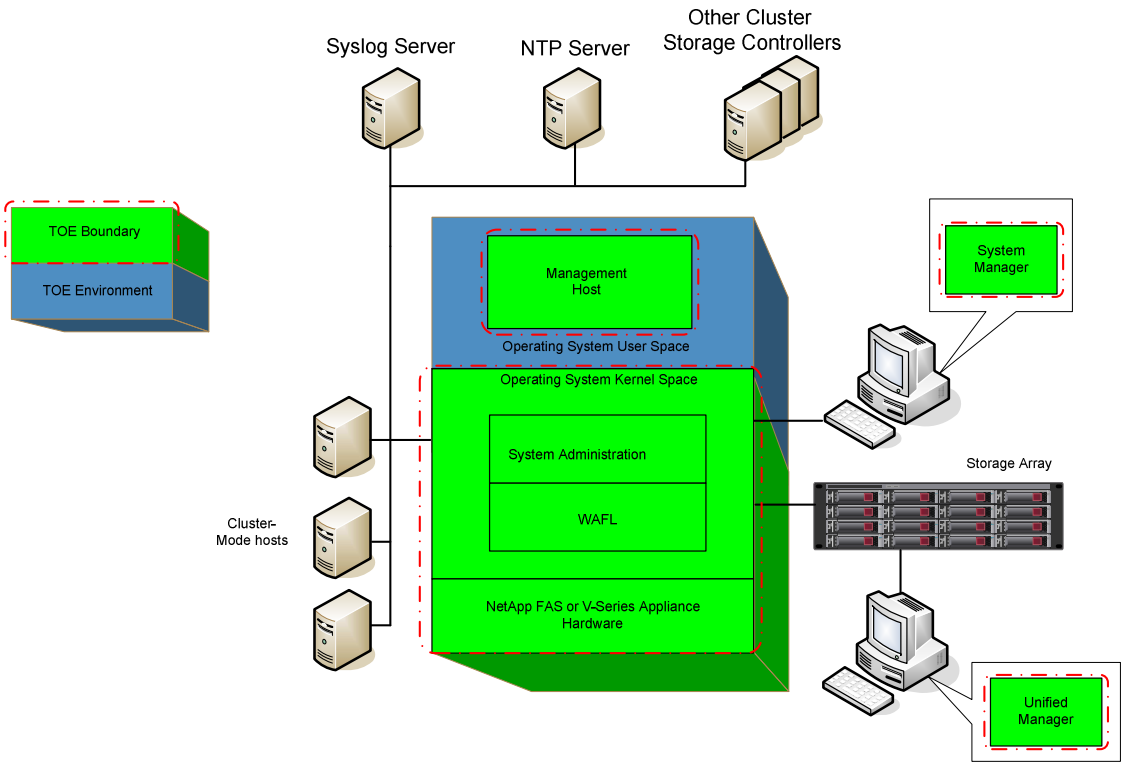

Figure 4 – Physical TOE Boundary

The TOE Environment includes the following components:

- Cluster Mode Hosts – SAN/NFS/SMB hosts talking to C-M volumes through Cluster-Mode (C-M) protocols

- Storage Array – Disk Aggregate managed by Raid Manager

1.5.1.1 TOE Software

The TOE software is a kernel operating system which runs on a subset of NetApp's proprietary 64-bit x86-based storage controller platforms listed in section 1.5.1.2.

The follow commands set the TOE's evaluated configuration:

The `vserver nfs modify –chown-mode` option for the evaluated configuration is disabled. When enabled, only a admin user has permission to change the owner of a file. When disabled, the `vserver nfs modify –chown-mode` option enables the owner of a file to change ownership of a file.

All authorized NetApp Administrators have the TSF admin role.

The `security login modify` command is set to "internal". When set to "internal", administrators are authenticated locally, and LDAP, NIS, etc. authentication is disabled.

The `security login role config modify` command is set to "on". When set to "on", all admin users, including the 'root' and 'administrator' accounts are subject to password rules such as account lock-out.

The `security audit` command parameters are set to "on". When set to "on", the auditing functionality of the TOE is enabled.

The evaluated configuration does not support changing a Qtree's style once the Qtree is configured.

1.5.1.2 TOE Hardware

The Clustered Data ONTAP® 8.2.1 runs on the NetApp's storage appliances; including the 8000 series, the 6200 series, the 6000 series, the 3200 series, the 3100 series, and the 2200 series appliances. The TOE includes the following hardware appliances, each one running one instances of the TOE software components:

- FAS8060
- FAS8040
- FAS8020
- FAS6290 and V-Series 6290
- FAS6280 and V-Series 6280
- FAS6250 and V-Series 6250
- FAS6240 and V-Series 6240
- FAS6220 and V-Series 6220
- FAS6210 and V-Series 6210
- FAS6080
- FAS6040
- FAS3270 and V-Series 3270
- FAS3250 and V-Series 3250
- FAS3240 and V-Series 3240
- FAS3220 and V-Series 3220
- FAS3210 and V-Series 3210
- FAS3170
- FAS3160
- FAS3140

- FAS2240-2 and FAS2240-4
- FAS2220

For a complete list of NetApp Storage Controllers on which the TOE operates, refer to the “New and changed platform and hardware support” section of the release notes for Clustered Data ONTAP® 8.2.1.

1.5.1.3 Guidance Documentation

The following guides are required reading and part of the TOE:

Clustered Data ONTAP® 8.2.1 Guidance Documentation Supplement

Clustered Data ONTAP® 8.2 Commands: Manual Page Reference

Clustered Data ONTAP® 8.2 System Administration Guide For Cluster Administrators

Clustered Data ONTAP® 8.2 System Administration Guide for SVM Administrators

Clustered Data ONTAP® 8.2 File Access and Protocols Management Guide for NFS

Clustered Data ONTAP® 8.2 File Access and Protocols Management Guide for CIFS

Clustered Data ONTAP® 8.2 Software Setup Guide

Clustered Data ONTAP® 8.2 High-Availability Configuration Guide

Clustered Data ONTAP® 8.2 Network Management Guide

V-Series Systems Installation Requirements and Reference Guide

Clustered Data ONTAP® 8.2 Physical Storage Management Guide

Clustered Data ONTAP® 8.2 Logical Storage Management Guide

Clustered Data ONTAP® 8.2 Data Protection Tape Backup and Recovery Guide

Clustered Data ONTAP Security Guidance

Clustered Data ONTAP® 8.2.1 Release Notes For Cluster-Mode

OnCommand® Unified Manager 6.1 Administration Guide

OnCommand® Unified Manager 6.1 Installation and Setup Guide

OnCommand® System Manager 3.1 Managing Clustered Data ONTAP® Using the GUI

OnCommand® System Manager 3.1 Installation and Setup Guide

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:

1.5.2.1 Security Audit

The TOE keeps track of auditable events through the Audit Log, stored in `/etc/log/auditlog`. An audit log is a record of commands executed at the console or a secure shell (SSH). All the commands executed in a source file script are also recorded in the audit log. Administrative Hypertext Transfer Protocol (HTTP) operations, such as those resulting from the use of OnCommand™ System Manager, are logged. All login attempts to access the storage system, with success or failure, are also logged.

In addition, changes made to configuration and registry files are logged. Read-only Application Programming Interfaces (APIs) by default are not logged but an administrator can enable auditing with the `security audit command` parameters.

For configuration changes, the audit log shows the following information:

- What configuration files were accessed
- When the configuration files were accessed
- What has been changed in the configuration files

For commands executed through the console or an SSH shell, the audit log shows the following information:

- What commands were executed
- Who executed the commands
- When the commands were executed

The TOE ensures that the audit trail storage is protected by rotating log files as they reach an administrator-configurable maximum size, and overwriting the oldest log file when the audit trail reaches an administrator-configurable maximum size. In addition, the log files are accessible for viewing by an authorized administrator via NFS, CIFS, or HTTPS.

For more information on the Security Audit functionality of the TOE, see section 7.1.1.

1.5.2.2 User Data Protection

User data protection defines how users connecting to the TOE are allowed to perform operations on objects.

User access to objects controlled by the TOE is governed by the enforcement of the DAC SFP. Access to NTFS-Style files via a CIFS share is authorized locally by file ACEs. Access to NFSv3 UNIX-Style files via an NFSv3 export is authorized locally by file/directory ownership and UNIX-Style security attributes. Access to NFSv4 UNIX-Style files via an NFSv4 export is authorized locally by file ACEs.

The TOE provides authorized administrators with several management interfaces outlined in section 1.5.2.4 to configure end-user network access. The management interfaces provide for the creation of rules that define actions the TOE is to take based on a set of conditions. The conditions and actions affect either the allowed access to user data by end-users (DAC SFP), or the way administrators interact with the TOE.

For more information on the User Data Protection functionality of the TOE, see section 7.1.2.

1.5.2.3 Identification and Authentication

The Identification and Authentication (I&A) functionality of the TOE forces human administrators to identify and authenticate themselves to the TOE before allowing any modifications to TOE managed TSF Data. Authentication credentials are maintained by the TOE in a local registry.

The TOE enforces minimum password strength requirements. The **`security login role config create`** and **`modify`** commands offer parameters to specify the minimum number of alphabetic characters, a mix of alphabetic with numeric, and the number of special characters that a password must contain. Passwords must have a length of at least 8 characters and contain at least one numeric character and at least two alphabetic characters.

The TOE will lock out an administrator account if the user fails to enter the proper credentials after **`–max-failed-login-attempts`** failed login attempts set by the **`security login role config modify`** or **`create`** command.

For more information on the I&A functionality of the TOE, see section 7.1.3.

1.5.2.4 Security Management

The TSF management functionality provides the necessary functions to allow a NetApp administrator to manage and support the TSF. Included in this functionality are the rules enforced by the TOE that define access to TOE-maintained TSF Data and its corresponding security attributes, and TSF Functions.

The security attributes include authentication data (used to authenticate end users), roles, security attribute data (used for DAC SFP enforcement) and other TSF data (used for DAC SFP subject security attribute resolution).

The TOE maintains the following roles for System Manager users:

- *admin*
- *autosupport*
- *backup*
- *readonly*
- none

The TOE also maintains the following roles for vServer administrators:

- vsadmin
- vsadmin-volume
- vasdmin-protocol
- vsadmin-backup
- vsadmin-readonly

And the following roles for OCUM users:

- *operator*,
- *storage administrator*,
- *OnCommand administrator*.

A “NetApp Administrator” is defined to be any human user who is assigned any of the administrative roles (except for none) listed above.

The TSF Functions are managed using the following capabilities (which are defined in detail in Table 22):

- Command Directory Access
- Access
- Query

For more information on the TSF management functionality, see section 7.1.4.

1.5.2.5 Protection of TOE Security Functionality

The TOE protects the TSF via the implementation of domain separation made possible by Secure Multi-Tenancy (SMT) functionality.

For more information on domain separation and Protection of the TSF, see section 7.1.5.

1.5.2.6 TOE Access

The TOE mitigates unauthorized administrator access by automatically terminating administrator sessions after 30 minutes of inactivity at the CLI.

For more information on the TOE Access functionality of the TOE, see section 7.1.6.

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

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

- OnCommand™ System Manager host hardware and operating system
- OnCommand™ Unified Manager host hardware and VMware ESX/ESXi hypervisor
- Remote resolution of authentication data via the nsswitch.conf passwd file (i.e. UNIX LDAP)
- Cross-protocol support (NFS access to NTFS-Style files, CIFS access to UNIX-Style files)
- Shared level ACLs
- Bypass traverse checking option
- Windows Group Policy Objects
- wafl.root_only_chown is disabled in the evaluated version (when disabled, file owners, in addition to the root account, can change ownership of files)
- Native File Blocking (File Screening)
- Mixed Qtrees
- Changing a Qtree’s style once the Qtree has been configured

- Remote CLIs accessible via:
  - Ethernet connections to an RLM[14] or a SP[15] or a BMC[16] installed in the appliance
  - A Telnet session to the appliance
  - A remote shell program, such as RSH [17]
  - FTP
  - Trivial File Transfer Protocol (TFTP)
  - HTTP (including WebDAV support)
  - VMware console as part of OCUM

[14] RLM – Remote Local Area Network (LAN) Management
[15] SP – Service Processor
[16] BMC – Baseboard Management Controller
[17] RSH – Remote Shell

2 CONFORMANCE CLAIMS

This section provides 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 4 – 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; PP claim (none); Parts 2 and 3 Interpretations of the CEM[18] as of 06/14/2013 were reviewed, and no interpretations apply to the claims made in this ST
PP Identification	None
Evaluation Assurance Level	EAL2+ (Augmented with Flaw Remediation (ALC_FLR.3))

[18] 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
- All organizational security policies with which the TOE must comply
- All assumptions about the secure usage of the TOE, including physical, personnel and connectivity aspects

3.1 THREATS TO SECURITY

This section identifies the threats to the IT assets against which protection is required by the TOE or by the security environment. The threat agents are divided into 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.)
- Agents or processes working either on behalf of attackers or autonomously: They may or may not have knowledge of the public or proprietary TOE configuration. These agents and processes can take many forms, such as bots or botnets designed to exploit common vulnerabilities or deny others access to IT products and services.

All three are assumed to have a low level of motivation. The IT assets requiring protection are the TSF and user data resident in 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. The following threats are applicable:

Table 5 – Threats

Name	Description
T.MASQUERADE	A TOE user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.TAMPER	A TOE user or process may be able to bypass the TOE's security mechanisms by tampering with the TOE or TOE environment.
T.UNAUTH	A TOE user may gain access to security data on the TOE, even though the user is not authorized in accordance with the TOE SFP.
T.DATALOSS	Threat agents may attempt to remove or destroy data collected and produced by the TOE.
T.NO_AUDIT	Threat agents may perform security-relevant operations on the TOE without being held accountable for it.
T.IA	Threat agents may attempt to compromise the TOE or network resources controlled by the TOE by attempting actions that it is not authorized to perform on the TOE or network resources.

3.2 ORGANIZATIONAL SECURITY POLICIES

An Organizational Security Policy (OSP) is a set of security rules, procedures, or guidelines imposed by an organization on the operational environment of the TOE. No OSPs are presumed to be imposed upon the TOE or its operational environment by any organization implementing the TOE in the CC evaluated configuration.

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. The specific conditions in Table 6 are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 6 – Assumptions

Name	Description
A.PEER	Any other systems with which the TOE communicates are assumed to be under the same management control and use a consistent representation for specific user and group identifiers.
A.NETWORK	Security Management shall be provided to protect the Confidentiality and Integrity of transactions on the network.
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NO_EVIL_ADM	The system administrative personnel are not hostile and will follow and abide by the instructions provided by the administrator documentation.
A.COOP	Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.
A.PROTECT	The processing resources of the TOE critical to the SFP enforcement will be protected from unauthorized physical modification by potentially hostile outsiders.
A.ADMIN_ACCESS	Administrative functionality shall be restricted to authorized administrators.
A.NTP	The IT Environment will be configured to provide the TOE to retrieve reliable time stamps by implementing the Network Time Protocol (NTP).
A.PHYSICAL	Physical security of the TOE and network, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

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 as follows:

Table 7 – Security Objectives for the TOE

Name	Description
O.ADMIN_ROLES	The TOE will provide administrative roles to isolate administrative actions.
O.AUDIT	The TOE will audit all administrator authentication attempts, whether successful or unsuccessful, as well as TOE user account configuration changes.
O.DAC_ACC	TOE users will be granted access only to user data for which they have been authorized based on their user identity and group membership.
O.ENFORCE	The TOE is designed and implemented in a manner that ensures the SFPs can't be bypassed or interfered with via mechanisms within the TOE's control.
O.IA	The TOE will require users to identify and authenticate themselves.
O.MANAGE	The TSF will provide functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.
O.STRONG_PWD	The TOE must ensure that all passwords will be at least 8 characters in length and will consist of at least one number and at least two alphabetic characters. Password construction will be complex enough to avoid use of passwords that are easily guessed or otherwise left vulnerable, e.g. names, dictionary words, phone numbers, birthdays, etc. should not be used.
O.INACTIVE	The TOE will terminate an inactive management session after a configurable interval of time.
O.TIMESTAMP	The TOE will provide a reliable timestamp for use by the TOE.

4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

4.2.1 IT Security Objectives

The following IT security objectives are to be satisfied by the environment:

Table 8 – IT Security Objectives

Name	Description
OE.ACCESS	The IT Environment will ensure that users gain only authorized access to the data the IT Environment manages.
OE.ADMIN_ROLES	The IT Environment will provide administrative roles to isolate administrative actions.
OE.ENFORCE	The IT Environment will support the TOE by providing mechanisms to ensure the TOE is neither bypassed nor interfered with via mechanisms outside the TOE's control.
OE.IA	The IT Environment must require authorized CIFS and NFS Clients to successfully I&A before allowing access to the TOE.
OE.NETWORK	The network path between the TOEs is a trusted channel. The network path between the CLI client and the TOE is a trusted channel.
OE.NTP	The IT Environment will enable the TOE to provide reliable time stamps by implementing NTP.
OE.SUBJECTDATA	The IT Environment will provide the TOE with the appropriate subject security attributes.

4.2.2 Non-IT Security Objectives

The following non-IT environment security objectives 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 9 – Non-IT Security Objectives

Name	Description
ON.CREDEN	Those responsible for the TOE must ensure that all access credentials, such as passwords, are protected by the users in a manner that maintains IT security objectives.
ON.INSTALL	Those responsible for the TOE and hardware required by the TOE must ensure that the TOE is delivered, installed, configured, managed, and operated in a manner which maintains IT security objectives.
ON.PHYSICAL	Those responsible for the TOE and the network on which it resides must ensure that those parts of the TOE and the IT Environment critical to SFP are protected from any physical attack that might compromise the IT security objectives.

Name	Description
ON.TRAINED	Those responsible for the TOE will be properly trained and provided the necessary information that ensures secure management of the TOE and the IT Environment.

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 10 identifies all extended SFRs implemented by the TOE

Table 10 – Extended TOE Security Functional Requirements

Name	Description
FPT_SEP_EXT.1	TSF domain separation for software TOEs

5.1.1 Class FPT: Extended Protection of the TSF

Families in this class address the requirements for functions to implement domain separation functionality as defined in CC Part 2

5.1.1.1 Family FPT_SEP_EXT: TSF Domain Separation for Software TOEs

Family Behavior

This family defines the requirements for domain separation of TSF data. This section defines the extended components for the FPT_SEP_EXT family.

Component Leveling

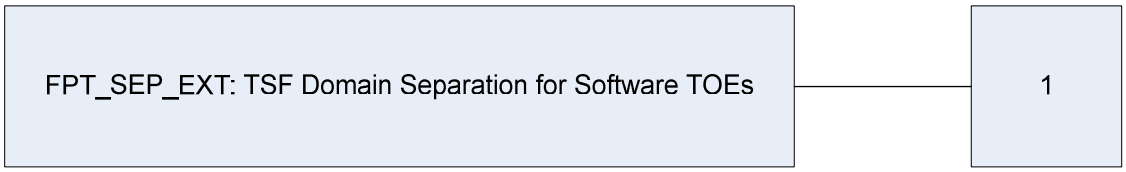

Figure 5 – TSF Domain Separation for Software TOEs family decomposition

The extended FPT_SEP_EXT.1 component is considered to be part of the FPT_SEP_EXT family.

FPT_SEP_EXT.1: TSF Domain Separation for Software TOEs provides the capability of the TOE to maintain a separate security domain to protect it from untrusted objects under the TOE's control. The extended family "FPT_SEP_EXT" was modeled after other Class FPT SFRs.

Management: FPT_SEP_EXT.1

The following actions could be considered for the management functions in FPT_SEP_EXT.1:

- Physical storage system administrators performing maintenance (deletion, modification, addition) of Vserver units, volumes, users, and groups of users, and their assignment to various Vservers within the TOE's control.
- Vserver (security domain) administrators performing maintenance (deletion, modification, addition) of volumes, users, and groups of users within the Vserver unit (virtual storage controller).

Audit: FPT_SEP_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Maintenance (deletion, modification, addition) of Vserver units, users, and groups of users, and their assignment to various security domains within the TOE's control.

FPT_SEP_EXT.1 TSF Domain Separation for Software TOEs

Hierarchical to: No other components

Dependencies: No Dependencies

FPT_SEP_EXT.1.1

The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects in the TOE's control.

FPT_SEP_EXT.1.2

The TSF shall enforce separation between the security domains of subjects in the TOE's control.

5.2 EXTENDED TOE SECURITY ASSURANCE COMPONENTS

There are no extended TOE Security Assurance Components 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 end 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 11 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 11 – TOE Security Functional Requirements

Name	Description	S	A	R	I
FAU_GEN.1	Audit data generation	✓	✓		
FAU_GEN.2	User Identity Association				
FAU_SAR.1	Audit review		✓		
FAU_SAR.2	Restricted audit review				
FAU_STG.1	Protected audit trail storage	✓			
FAU_STG.4	Prevention of audit data loss	✓	✓		
FDP_ACC.1	Subset access control		✓		
FDP_ACF.1	Security attribute based access control		✓		
FIA_AFL.1	Authentication failure handling	✓	✓		
FIA_ATD.1	User attribute definition		✓		
FIA_SOS.1	Verification of secrets		✓		

Name	Description	S	A	R	I
FIA_UAU.2	User authentication before any action			✓	
FIA_UID.2	User identification before any action			✓	
FMT_MOF.1	Management of security function behaviour	✓	✓	✓	
FMT_MSA.1	Management of security attributes	✓	✓		
FMT_MSA.3	Static attribute initialisation	✓	✓		
FMT_MTD.1(a)	Management of TSF data	✓	✓		✓
FMT_MTD.1(b)	Management of TSF data	✓	✓		✓
FMT_SMF.1	Specification of management functions		✓		
FMT_SMR.1	Security roles		✓		
FPT_SEP_EXT.1	TSF domain separation for software TOEs				
FPT_STM.1	Reliable Time Stamps				
FTA_SSL.3	TSF-initiated termination		✓		

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

6.2.1 Class FAU: Security Audit

Application Note: the Security Audit requirements do not apply to the OCUM.

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) *[The events specified in Table 12 below].*

FAU_GEN.1.2

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

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*the additional event information specified in Table 12 below]*.

Table 12 – FAU_GEN.1.2 Audit Generation Details

SFR Addressed	Auditable Events	Additional Event Information
FIA_UAU.2, FIA_UID.2	Successful local logon	User identity, security domain
FIA_UAU.2, FIA_UID.2	Unsuccessful local logon	User identity supplied, security domain
FMT_SMF.1	User created	User ID[19] created, User ID of the administrator performing the action, security domain
FMT_SMF.1	User deleted	User ID deleted, User ID of the administrator performing the action, security domain
FMT_SMF.1	Group created	Group created, User ID of the administrator performing the action, security domain
FMT_SMF.1	Group deleted	Group deleted, User ID of the administrator performing the action, security domain
FMT_SMF.1	Group member added	User ID and group associated, User ID of the administrator performing the action, security domain
FMT_SMF.1	Group member deleted	User ID and group disassociated, user ID of the administrator performing the action, security domain

[19] ID - Identifier

FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

FAU_GEN.2.1

For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

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 [*authorized administrators*] with the capability to read [*all 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_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1

The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2

The TSF shall be able to [prevent] unauthorized modifications to the stored audit records in the audit trail.

FAU_STG.4 Prevention of audit data loss

Hierarchical to: FAU_STG.3 Action in case of possible audit data loss

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.4.1

The TSF shall [overwrite the oldest stored audit records] and [*no other actions*] if the audit trail is full.

6.2.2 Class FDP: User Data Protection

Application Note: the User Data Protection requirements do not apply to the OCUM.

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 [*DAC SFP*] on [*the subjects, objects, and operations among subjects and objects listed in Table 13 below*].

Table 13 – FDP_ACC.1.1 Detail

Subject	*Object (Files on the Storage Appliance)*			*Operation among Subject and Object covered by the DAC SFP*
	File Style	*File Type*	*Qtree Type*	
NFSv3 Client	*NFSv3 UNIX-Style File*	*Directory, Symbolic Link, Regular File*	*UNIX Qtree*	*Create, read, write, execute, delete, change permissions, change ownership*
NFSv4 Client	*NFSv4 Unix-Style File*	*Directory, Symbolic Link, Regular File*	*UNIX Qtree*	*Create, read, write, execute, delete, change permissions, change ownership*
CIFS Client	*NTFS-Style File*	*Directory, Regular File*	*NTFS Qtree*	*Create, read, write, execute, delete, change permissions, change ownership*

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 [*DAC SFP*] to objects based on the following: [*the subjects, objects, operations, and associated security attributes listed in Table 14 below:*]

Table 14 – FDP_ACF.1.1 Detail

Operation	*Subject*	*Object (File)*	*Subject*		*Object (file)*	*Other Objects and Security Attributes used for DAC SFP*
			Security Attribute	*Other TSF Data*	*Security Attribute*	
Create	*NFSv3 Client*	*NFSv3 UNIX-Style file*	*UNIX User UID, UNIX User GID*	*UNIX Username*	*N/A*	*UNIX Parent Directory UID, UNIX Parent Directory GID and access mode*

Operation	Subject	Object (File)	Subject		Object (file)	Other Objects and Security Attributes used for DAC SFP
			Security Attribute	Other TSF Data	Security Attribute	
	NFSv4 Client	NFSv4 UNIX-Style file	UNIX User UID, UNIX User GID	UNIX Username	N/A	UNIX Parent Directory UID, UNIX Parent Directory ACEs
	CIFS Client	NTFS-Style file	Windows User SID, Windows User GID	Windows Username	N/A	Qtree type, Parent directory's SID and ACEs
Read, Write, Execute	NFSv3 Client	NFSv3 UNIX-Style file	UNIX User UID, UNIX User GID	None	UNIX file UID, UNIX file GID, access mode	None
	NFSv4 Client	NFSv4 UNIX-Style file	UNIX User UID, UNIX User GID	None	UNIX User UID, ACEs	None
	CIFS Client	NTFS-Style file	Windows User SID, Windows User GID	Windows Username	SID and ACEs	None
Delete	NFSv3 Client	NFSv3 UNIX-Style file	UNIX User UID, UNIX User GID	UNIX Username	None	UNIX Parent Directory UID, UNIX Parent Directory GID and access mode
	NFSv4 Client	NFSv4 UNIX-Style file	UNIX User UID UNIX User GID	None	UNIX User UID, ACEs	UNIX Parent Directory UID, UNIX Parent Directory ACEs
	CIFS Client	NTFS-Style file	Windows User SID, Windows User GID	Windows Username	SID and ACEs	Parent directory's SID and ACEs
Change Permission	NFSv3 Client	NFSv3 UNIX-Style file	UNIX User UID, UNIX User GID	UNIX Username	None	UNIX Parent Directory UID, UNIX Parent Directory GID and access mode
	NFSv4 Client	NFSv4 UNIX-Style file	UNIX User UID UNIX User GID	None	UNIX User UID, ACEs	UNIX Parent Directory UID, UNIX Parent Directory ACEs
	CIFS Client	NTFS-Style file	Windows User SID, Windows User GID	Windows Username	SID and ACEs	Parent directory's SID and ACEs
Change Owner	NFSv3 Client	NFSv3 UNIX-Style file	UNIX User UID	None	UNIX User UID	None

Operation	Subject	Object (File)	Subject		Object (file)	Other Objects and Security Attributes used for DAC SFP
			Security Attribute	Other TSF Data	Security Attribute	
	NFSv4 Client	NFSv4 UNIX-Style file	UNIX User UID	None	UNIX User UID	None
	CIFS Client	NTFS-Style file	Windows User SID, Windows User GID	None	SID and ACEs	None

FDP_ACF.1.2

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*access is granted if one of the following conditions listed in Table 15 below is true:*]

Table 15 – FDP_ACF.1.2 Detail

Operation	Subject	Object (File)	=DAC Rule
Create	NFSv3 Client	NFSv3 UNIX-Style file	1. The subject is the owner of the parent directory and the owner has been granted Write and Execute access (UNIX-Style security attributes). 2. The subject is not the owner of the parent directory but is a member of the parent directory's group and the group has Write and Execute access (UNIX-Style security attributes). 3. The subject is neither the owner of the parent directory nor a member of the parent directory's group but Write and Execute access has been granted to all subjects (UNIX-Style security attributes).
	NFSv4 Client	NFSv4 UNIX-Style file	4. The subject is the owner of the parent directory and the owner has been granted Write and Execute access (UNIX-Style security attributes). 5. There is no parent directory ACE that denies Write or Execute access to the subject and parent directory ACEs exist that grant Write and Execute permission to the subject (NFSv4-Style security attributes). 6. There is no parent directory ACE that denies Write or Execute access to any group that the subject is a member of and parent directory ACEs exist that grant Write and Execute permission to any group the subject is a member of (NFSv4-Style security attributes).

Operation	*Subject*	*Object (File)*	*=DAC Rule*
	CIFS Client	*NTFS-Style file*	*7. There is no parent directory ACE that denies Write or Execute access to the subject and parent directory ACEs exist that grant Write and Execute permission to the subject (NTFS-Style security attributes).* *8. There is no parent directory ACE that denies Write or Execute access to any group that the subject is a member of and parent directory ACEs exist that grant Write and Execute permission to any group the subject is a member of (NTFS-Style security attributes).*
Read, Write Execute	*NFSv3 Client*	*NFSv3 UNIX-Style file*	*9. The subject is the owner of the file and the owner has been granted access for the specific operation (UNIX-Style security attributes).* *10. The subject is not the owner of the file but is a member of the object's group and the object's group has access for the specific operation (UNIX-Style security attributes).* *11. The subject is neither the owner of the file nor a member of the object's group but the specific access request has been granted to all subjects (UNIX-Style security attributes)*
	NFSv4 Client	*NFSv4 UNIX-Style file*	*12. The subject is the owner of the file and the owner has been granted access for the specific operation (UNIX-Style security attributes).* *13. There is no ACE that denies access to the subject for the specific operation and an ACE exists that grants permission to the subject for the specific operation (NFSv4-Style security attributes).* *14. There is no ACE that denies access for the specific operation to any group that the subject is a member of and an ACE exists that grants permission to any group the subject is a member of for the specific operation (NFSv4-Style security attributes).*
	CIFS Client	*NTFS-Style file*	*15. There is no ACE that denies access to the subject for the specific operation and an ACE exists that grants permission to the subject for the specific operation (NTFS-Style security attributes).* *16. There is no ACE that denies access for the specific operation to any group that the subject is a member of and an ACE exists that grants permission to any group the subject is a member of for the specific operation (NTFS-Style security attributes).*
Delete	*NFSv3 Client*	*NFSv3 UNIX-Style file*	*17. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory).*
	NFSv4 Client	*NFSv4 UNIX-Style file*	*18. Rule 12, 13, or 14 above is true (subject has Delete NFSv4-style permission or is UNIX owner for parent directory)*

Operation	*Subject*	*Object (File)*	*=DAC Rule*
	CIFS Client	*NTFS-Style file*	*19. Rule 15 or 16 above is true for Delete operation (subject has Delete NTFS-Style permission for object).* *20. Rule 12 above fails and Rule 14 or 15 below are true (subject has Delete Child NTFS-Style permission for parent directory)* *21. There is no parent directory ACE that denies Delete Child access to the subject and a parent directory ACE exists that grants Delete Child permission to the subject (NTFS-Style security attribute).* *22. There is no parent directory ACE that denies Delete Child access to any group that the subject is a member of and an object ACE exists that grants Delete Child permission to a group the subject is a member of (NTFS-Style security attribute).*
Change Permission	*NFSv3 Client*	*NFSv3 UNIX-Style file*	*23. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory) and rule 6, 7 or 8 above is true for Write operation (UNIX-Style permission for object).*
	NFSv4 Client	*NFSv4 UNIX-Style file*	*24. Rule 4, 5, or 6 above is true (subject has Write and Execute NFSv4-Style permission for parent directory) and rule 12, 13, or 14 above is true for Change Permission operation (UNIX and NFSv4 Style permission for object)*
	CIFS Client	*NTFS-Style file*	*25. Rule 7 or 8 above is true (subject has Write and Execute NTFS-Style permission for parent directory) and rule 15 or 16 above is true for Change Permission operation (NTFS-Style permission for object).*
Change Ownership	*NFSv3 Client*	*NFSv3 UNIX-Style file*	*26. If the UNIX UID is root, or the owner of the file, the operation is allowed.*
	NFSv4 Client	*NFSv4 UNIX-Style file*	*27. Rule 12, 13, or 14 above is true for Change Ownership operation (subject has Change Owner NFSv4-Style permission or is UNIX-Style owner for object)*
	CIFS Client	*NTFS-Style file*	*28. Rule 15 or 16 above is true for Change Ownership operation (subject has Change Owner NTFS-Style permission for object).*

FDP_ACF.1.3

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*access is granted if the object is a UNIX-style file and the subject is root*].

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the following additional rule: *[access is denied if the subject does not have an Administrative Role*].

6.2.3 Class FIA: Identification and Authentication

Application Note: the Identification and Authentication requirements of FIA_AFL.1, FIA_ATD.1, and FIA_SOS.1 do not apply to the OCUM.

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 an administrator configurable positive integer within [*0 – 4,294,967,295*] unsuccessful authentication attempts occur related to [*login attempts*].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been [surpassed], the TSF shall [*lock the user, except for the root account, out of the system*].

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: [*TOE user name, password, group membership, UNIX User UID and GID; Windows User SID and GID*].

FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_SOS.1.1

The TSF shall provide a mechanism to verify that secrets meet [*the following criteria: at least 8 characters in length and consist of at least one number and at least one alphabetic character*].

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***~~ **administrator** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that 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***~~ ***administrator*** to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.2.4 Class FMT: Security Management

Application Note: the Security Management requirements of FMT_MSA.1, FMT_MSA.3, FMT_MTD.1(b), and FMT_SMF.1 do not apply to the OCUM.

FMT_MOF.1 Management of security functions behavior

Hierarchical to: No other components.

Dependencies: FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

FMT_MOF.1.1

The TSF shall restrict the ability to [~~*determine the behavior of, disable, enable, modify the dminist of*~~ **perform**] the functions [*Default command access in Table 16 below*] to [*the roles listed in Table 16 below*].

Table 16 – Roles maintained by the TOE

System Manager Role	*Level of access…*	*to command directory or directories*
admin	*all*	*All command directories (DEFAULT)*
autosupport	*all*	• *set* • *system node autosupport*
backup	*all*	• *security login password* • *set* • *vserver services ndmp*
	Readonly	• *volume*
readonly	*readonly*	*All command directories (DEFAULT)*
none	*None*	*All command directories (DEFAULT)*
vServer Administrator Role	*Level of access*	
vsadmin	*This role is the super user role for a Vserver and is assigned by default. A Vserver administrator with this role has the following capabilities:* • *Managing own user account local password and key information* • *Managing volumes, quotas, qtrees, Snapshot copies, and files.* • *Managing LUNs* • *Configuring protocols: NFS, CIFS, iSCSI, and FC (FCoE included)* • *Configuring services: DNS, LDAP, and NIS* • *Monitoring jobs* • *Monitoring network connections and network interface* • *Monitoring the health of a Vserver*	
vsadmin-volume	*A Vserver administrator with this role has the following capabilities:*	

	• *Managing own user account local password and key information* • *Managing volumes, quotas, qtrees, Snapshot copies, and files.* • *Managing LUNs* • *Configuring protocols: NFS, CIFS, iSCSI, and FC (FCoE included)* • *Configuring services: DNS, LDAP, and NIS* • *Monitoring network interface* • *Monitoring the health of a Vserver*
vsadmin-protocol	*A Vserver administrator with this role has the following capabilities:* • *Managing own user account local password and key information* • *Configuring protocols: NFS, CIFS, iSCSI, and FC (FCoE included)* • *Configuring services: DNS, LDAP, and NIS* • *Managing LUNs* • *Monitoring network interface* • *Monitoring the health of a Vserver*
vsadmin-readonly	*A Vserver administrator with this role has the following capabilities:* • *Managing own user account local password and key information* • *Monitoring the health of a Vserver* • *Monitoring network interface* • *Viewing volumes and LUNs* • *Viewing services and protocols*
vsadmin-backup	*A Vserver administrator with this role has the following capabilities:* • *Managing NDMP operations* • *Making a restored volume as read-write* • *Viewing volumes and LUNs* ***Note**: A Vserver administrator with vsadmin-backup role cannot manage own user account local password and key information.*
OCUM Role	*Level of access*
operator	*View all data, assign and resolve events*
storage administrator	*All operator access plus:* • *manage storage service objects* • *define alerts* • *manage storage management options* • *manage storage management policies*
OnCommand administrator (includes maintenance user capabilities)	*All storage administrator access plus:* • *manage users* • *manage administrative options* • *manage database access* • *configure network access for OCUM* • *upgrade the UCOM software* • *increase data disk or swap disk size* • *change the time zone* • *send on-demand AutoSupport messages to technical support* • *send periodic AutoSupport messages to technical support* • *generate support bundles to send to technical support*

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 [*DAC SFP*] to restrict the ability to [modify, delete*, add*] the security attributes [*TOE User UID and Primary TOE User GID maintained locally by the TOE*] to [*an authorized administrator*].

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 [*DAC 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 [*no authorized identified roles*] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1(a) 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(a).1

The TSF shall restrict the ability to [query, modify, delete] the [*local user account repository*] to [*authorized administrators with the Admin or OnCommand administrator role*].

FMT_MTD.1(b) 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(b).1

The TSF shall restrict the ability to [modify] the [*state of the TOE*] to [*authorized administrators with the Admin role*].

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No Dependencies

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions: [*management of security functions behavior, management of security attributes, and management of TSF data*].

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 [*System Manager, vServer Administrator, and OCUM roles as identified in Table 16*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

6.2.5 Class FPT: Protection of the TSF

Application Note: the Protection of the TSF requirements do not apply to the OCUM.

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.

FPT_SEP_EXT.1 TSF Domain Separation for Software TOEs

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_SEP_EXT.1.1

The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects in the TOE's control

FPT_SEP_EXT.1.2

The TSF shall enforce separation between the security domains of subjects in the TOE's control.

6.2.6 Class FTA: TOE Access

Application Note: the TOE Access requirements do not apply to the OCUM.

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 [*configurable time interval of user inactivity at the CLI, defaulting to 30 minutes*].

6.3 SECURITY ASSURANCE REQUIREMENTS

This section defines the assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3 and are EAL2 augmented with ALC_FLR.3. Table 17 – Assurance Requirements summarizes the requirements.

Table 17 – 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.3 Systematic Flaw Remediation
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 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 18 lists the security functionality and their associated SFRs.

Table 18 – Mapping of TOE Security Functionality to Security Functional Requirements

TOE Security Functionality	SFR ID	Description
Security Audit	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User Identity Association
	FAU_SAR.1	Audit review
	FAU_SAR.2	Restricted audit review
	FAU_STG.1	Protected audit trail storage
	FAU_STG.4	Prevention of audit data loss
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
Identification and Authentication	FIA_AFL.1	Authentication failure handling
	FIA_ATD.1	User attribute definition
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UID.2	User identification before any action
Security Management	FMT_MOF.1	Management of security function behaviour
	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialisation
	FMT_MTD.1(a)	Management of TSF data
	FMT_MTD.1(b)	Management of TSF data

TOE Security Functionality	SFR ID	Description
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of the TSF	FPT_SEP_EXT.1	TSF domain separation for software TOEs
	FPT_STM.1	Reliable Time Stamps
TOE Access	FTA_SSL.3	TSF-initiated termination

7.1.1 Security Audit

The TOE generates audit event records for events involving administrator logons as well as configuration changes, specifically for locally-defined users and groups. The audit function is normally executing any time the TOE is operational and the `security audit modify` command parameters are set to "on". If the audit function is started or stopped, an audit event record is generated. All audit event records include a reliable timestamp.

The TOE ensures that the audit trail storage is protected by rotating log files as they reach an administrator-configurable maximum size, and overwriting the oldest log file when the audit trail reaches an administrator-configurable maximum size. The TOE can also ensure audit trail storage by periodically starting a new audit file based upon specifying a time schedule. Administrators are protected by the flexibility given by Data ONTAP, since the audit trail may be rotated under multiple conditions, which can fit most organizations' procedural requirements.

The maximum size of the audit-log file is specified by the `vserver audit create` and `modify` commands using their `–rotate-size parameter`. The individual Vserver audit configuration may specify when its consolidated audit log is rotated. Based upon a flexible schedule configuration, the `/etc/log/auditlog` file is copied to `/etc/log/auditlog.0`, `/etc/log/auditlog.0` is copied to `/etc/log/auditlog.1`, and so on. This also occurs if the audit-log file reaches the maximum size specified by `rotate-size` parameter.

The system saves audit-log files for a configurable number of files. This is set using the `–rotate-limit` parameter of the `vserver audit create and modify` commands.

Administrators can access the audit-log files using the NFS or CIFS clients, or using HTTPS. The TOE ensures that the audit trail storage is protected from unauthorized deletion or modification by enforcing role-based permissions to the audit trail as described in Table 19 below:

Table 19 – Audit Trail Storage Access by Role

Role	Permission
admin	create, read, write, execute, delete, change permission, change owner
autosupport	read
backup	read
readonly	read
none	read

To access the log files via NFS, the administrator must mount the root directory <system_name>:/vol/vol0) to a desired mount point on the management workstation (where <system_name> is the short name, Fully Qualified Domain Name (FQDN), or IP address of the storage system). The administrator can then change directories to <mount point>/etc/log/ to view log files in a XML viewing tool (where <mount point> is the desired mount point on the management workstation).

To access the log files via CIFS, the administrator must mount the \\<system_name>\C$ share to a desired drive letter on the management workstation (where <system_name> is the short name, FQDN, or IP address of the storage system). The administrator can then change directories to <drive letter>\etc\log\ to view log files in a XML viewing tool (where <drive letter> is the desired drive letter on the management workstation).

To access the log files via HTTPS, the administrator must ensure that the vserver services web –enabled command is set to true to **allow administrative access**. The administrator can then point the web browser on the management workstation to https://<system_name>/na_admin/logs/ to download log files to the management workstation (where <system_name> is the short name, FQDN, or IP address of the storage system).

Administrators can also configure auditing for specific file access protocols, and forward audit logs to a remote Syslog log host.

The OnCommand™ System Manager generates audit records based on the audit logging level configured in the OnCommand™ System Manager. The OnCommand™ System Manager enables an authorized administrator to refine the logging output by selecting which type of log statements are output. By default, system logging is set to INFO. An authorized administrator can choose one of the following log levels:

- TRACE
- DEBUG
- INFO
- WARN
- ERROR
- FATAL
- OFF

These levels function hierarchically. If the log level is set to OFF indicates no logging of messages. In the evaluated configuration, the audit level cannot be set to OFF. The TRACE level logging includes all logs ranging from DEBUG to FATAL. These audit records include the date and time of the event, the type of event, and the outcome (success or failure) of the event. The OnCommand™ System Manager associates each auditable event (command executed) with the identity of the administrator that initiated the event. The OnCommand™ System Manager stores the log files on the local machine where the OnCommand™ System Manager is installed. The OnCommand™ System Manager only displays the following ONTAP logs through the OnCommand™ System Manager:

- Sys Log
- Audit Log
- SnapMirror Log

All the logs that are displayed via the OnCommand™ System Manager are read only. An authorized administrator cannot modify or delete any logs from the OnCommand™ System Manager interface.

TOE Security Functional Requirements Satisfied: FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_STG.1, FAU_STG.4

7.1.2 User Data Protection

The TSF mediates access of subjects and objects. The subjects covered by the DAC SFP are NFS Clients and CIFS Clients. The objects covered by the DAC SFP are files (user data). The TOE maintains files with either NTFS-Style security attributes or UNIX-Style security attributes. The access modes covered by the DAC SFP are: create, read, write, execute, delete, change permission and change owner.

The DAC SFP is detailed below:

7.1.2.1 Discretionary Access Control Security Function Policy

The DAC SFP protects user data (FDP_ACC.1). The DAC SFP uses the subject type, subject's security attributes, the object, the object's security attributes and the access mode (operation) to determine if access is granted. For some operations, the security attributes of the object's parent directory are also used. The following sections describe the DAC SFP and provide the Security Functional Requirements that meet the Security Function.

7.1.2.1.1 DAC SFP Object Security Attributes

The User Data that is covered by the DAC SFP are files (objects). Each file maintained by the TOE has a file style associated with it. The type of security attributes associated with the file defines a file style. The TOE maintains two styles of files: UNIX-Style files and NTFS-Style files. UNIX-Style files have UNIX-Style security attributes and NTFS-Style files have NTFS-Style security. Each file style is assigned different security attributes that are used by the DAC SFP to determine if access is granted for a subject.

In addition to a file style, each file has a file type. The file types may be directories, symbolic links or regular files. UNIX-Style files may be a directory, a symbolic link or a regular file (FDP_ACC.1). NTFS-Style files do not have symbolic links; therefore, the file type will be either directory or regular file (FDP_ACC.1).

In addition to the file type, the TOE maintains three different storage types: UNIX Qtrees, NTFS Qtrees or mixed Qtrees. A Qtree is a disk space partition. UNIX Qtrees store UNIX-Style files with UNIX-Style security attributes. NTFS Qtrees store NTFS-Style files with NTFS-Style security attributes. Mixed Qtrees store both styles of files. Any file may have either UNIX-Style security attributes or NTFS-Style security attributes associated with them. Mixed Qtrees will not be part of the evaluated configuration. The following sections describe the security attributes associated with the objects.

7.1.2.1.1.1 NFSv3 UNIX-Style File Security Attribute Description

A UNIX-Style file managed by the TOE has eleven security attributes that are used to determine file access. The security attributes include a UNIX File UID, a UNIX file GID and a nine character access mode string. The UNIX File UID is the UID of the file's owner. The UNIX file GID is the GID associated with the file. The access mode is a subset of characters within the file's file permission string. The file permission string is represented in ten characters common to all UNIX files (e.g. drwxrwxrwx). The first character contains one of three characters that identify the file type: d for directory, l for a symbolic link, or a dash (-) indicates the file is a regular file. The following 9 characters represent the access mode for the file in three sets of rwx triplets. The first triplet specifies the permission for the file's owner (UID). The next triplet specifies the permissions for the group associated with the file (UNIX file GID). The last three characters specify the permission for the users who are neither the owner nor members of the file's group (other). The rwx triplet identifies the permission for that set (owner, group, other). The three characters represent read, write or execute privileges. If the character is a dash, the set does not have permissions to perform the specific action (FDP_ACF.1). A directory's permission string may also contain a "sticky bit" represented at the end of the nine character access mode string by a "T" (e.g. drwxrwxrwxT). A sticky bit-enabled directory signifies that files or folders created within this directory can only be deleted by the file owner.

To determine if a client has read, write or execute permission for a UNIX-Style file, the TOE first compares the client's UNIX User UID with the file's UID. If a match occurs (the client is the owner) and the file's access mode specifies permission for the specific access request (rwx), the request is allowed. If the owner does not have permission to perform the request, the request is denied. If the client is not the file's owner, the TOE determines if the client is a member of the file's group by comparing the client's Primary UNIX User GID to the file's GID. If the client is a member of the file's group and the access mode specifies permission for the specific access request, the request is allowed. If the group does not have permission to perform the request, the request is denied. If the client is not the file's owner or a member of the file's group, the TOE then determines if all others (the last triplet) have permission to perform the request. If all others have permission, the request is honored. Otherwise the request is denied (FDP_ACF.1).

For the remainder of this document, when the DAC SFP rules state that the TOE determines if a client, using UNIX-Style security attributes, has access, the above steps are what the TOE performs: the TOE walks through the owner, group and other attributes to determine access.

7.1.2.1.1.2 NFSv4 UNIX-Style File Security Attribute Description

The TOE's NFSv4 UNIX-Style file security attributes are NFSv4 ACLs. Each file has a data structure associated with it containing the file owner's UID and an ACL. Each ACL consists of one or more ACE. Each ACE explicitly allows or denies access to a single user or group. Access is allowed if there is no ACE

that denies access to the user or any group that the user is a member of and if an ACE exists that grants permission to the user or any group the user is a member of.

This determination is made by consulting the ownership, permissions, and ACEs on the file or directory and comparing against the UID and GID of the requesting user. The group memberships (and possibly username to UID number mapping) are obtained from local files or a directory service, while the file permissions and ACLs are stored in the file system.

For the remainder of this document, when the DAC SFP rules state that the TOE determines if a client, using NFSv4 UNIX-Style security attributes, has access, the above steps are what the TOE performs to determine access.

7.1.2.1.1.3 NTFS-Style File Security Attributes Description

The TOE's NTFS-Style file security attributes are standard Windows file security attributes. Each file has a data structure associated with an SD. This SD contains the file owner's SID, group's SID, DACL[20], and SACL[21]. Each ACL consists of one or more ACEs. Each ACE explicitly allows or denies access to a single user or group. Access is allowed if there is no ACE that denies access to the user or any group that the user is a member of and if an ACE exists that grants permission to the user or any group the user is a member of.

For the remainder of this document, when the DAC SFP rules state that the TOE determines if a client, using NTFS-Style security attributes, has access, the above steps are what the TOE performs to determine access.

7.1.2.1.2 DAC SFP Access Requests

Access requests define what operation a subject requests to perform on an object. The TOE's DAC SFP addresses seven access requests: create, read, write, execute, delete, change permissions, and change owner (FDP_ACC.1). The following sections define the operations.

7.1.2.1.2.1 UNIX-Style Access Requests

The following table identifies the operations of subjects on UNIX-Style files (objects) covered by the DAC SFP and explains what each of the file access request means.

Table 20 – UNIX-Style File Access Requests

DAC SFP Operation	UNIX-Style File Types		
	Directory	Symbolic Link	Normal File
Create	Create a directory.	Create a symbolic link.	Create a file.
Read	Get info about the directory or its contents.	Read the file the symbolic link contains the name of.	Read the file.
Write	Add a file in the directory.	Write to the file the symbolic link contains the name of.	Append/write/truncate the file.
Execute	Traverse the directory; change the working directory or access a file or subdirectory in the directory.	Execute the file the symbolic link contains the name of.	Execute the file.
Delete	Delete the directory.	Delete the symbolic link.	Delete the file.

[20] DACL – Discretionary ACL: Used to determine permissions.
[21] SACL – System ACL: Used for auditing purposes.

DAC SFP Operation	UNIX-Style File Types		
	Directory	Symbolic Link	Normal File
Change Permission	Change the permission of the directory.	Change the permission of the symbolic link.	Change the permission of the file.
Change Owner	No effect.	Become the symbolic link's owner.	Become the file's owner.

7.1.2.1.2.2 NTFS-Style File Access Requests

The NTFS-Style file security attributes define more access modes than UNIX does. There are, however, no symbolic links in NTFS-Style files. The following table identifies the operations of subjects on NTFS-Style files (objects) covered by the DAC SFP and explains what each of the basic file access request means.

Table 21 – NTFS-Style File Access Requests

DAC SFP Operation	NTFS-Style File Types	
	Directory	Normal File
Create	Create a directory	Create a file.
Read	Get info about the directory or its contents	Read the file.
Write	Add a file in the directory	Truncate, append, or overwrite the file.
Execute	No effect	If the file has an extension of .exe or .com, attempt to execute it as a native binary. If it has an extension of .bat or .cmd, attempt to execute it as a batch or command file using the command interpreter.
Delete	Delete the directory. Delete privilege must be explicitly granted on the contained files and subdirectories before they can be deleted. A directory may not be deleted unless it is empty.	Delete the file.
Change Permission	Change the permissions on the directory (change the directory's ACL)	Change the file's ACL.
Change Owner	Become the directory's owner	Become the file's owner.

7.1.2.1.3 DAC Operations and Rules

In general the TOE supports access to all objects from all subjects. However, the following exceptions apply:

- **Client** – The DAC SFP supports client protocol-specific support for create, read, write, execute, delete, change permission and change owner operations.

- **File Style** – The file style (UNIX-Style or NTFS-Style) is considered in the TOE's DAC SFP Rules because the type of security attributes maintained by the object aids in determining the type of security attributes required by the client.
- **File Type** – The file type (directory, symbolic link or regular file) is considered when determining if object access is allowed for a subject. The CIFS protocol does not know about symbolic links. Therefore, CIFS Clients will not request an operation for a symbolic link; the only operations for objects with file type of symbolic link applicable to the DAC SFP are NFS Client operations for UNIX-Style files.
- **Additional Data** – As well as client security attributes and object security attributes, certain operations require the TOE to examine the security attributes of other objects to determine if access is allowed, specifically, the object's parent directory. The TOE examines the security attributes of an object's parent directory for create, delete and change permission operations.
- **Operation** – The operations supported by the DAC are: Create, Read, Write, Execute, Delete, Change Permissions, and Change Owner. The execute command is treated differently for the different file styles and file types. Executing an NTFS directory has no effect. Executing a UNIX-Style directory means to traverse the directory, change the working directory, or access a file or subdirectory in the directory.

7.1.2.1.4 DAC SFP Subject Security Attributes

The subjects that apply to the DAC SFP are subjects with or without administrative roles; they access the TOE as NFS Clients and CIFS Clients (FDP_ACC.1). To determine if access is permitted for an object, the TOE requires the security attributes associated with the client. These security attributes may be resolved by the TOE or the IT Environment.

The subject security attributes required by the DAC SFP depend on the type of security attributes maintained by the object; the object will require either UNIX-Style subject security attributes or NTFS-Style subject security attributes to determine if access is permitted. Based on the native systems, NFS clients are typically associated with UNIX-Style security attributes and CIFS Clients are associated with NTFS-Style security attributes. The following sections describe the TOE's subject security attribute resolution used to enforce the DAC SFP.

7.1.2.1.4.1 Derivation of UNIX-Style Client Subject Security Attributes

If the TOE determines that NFSv3 UNIX-Style security attributes should be used to determine access for an object, the TOE requires a client's (subject's) UNIX User UID and UNIX User GID (FDP_ACF.1).

If the TOE determines that the NFSv4 UNIX-Style security attributes should be used to determine access for an object, the TOE requires a client's (subject's) UNIX User UID and GID with permission matching the file's ACL (FDP_ACF.1).

If the access request is initiated by an NFS Client, the TOE received the NFS Client's UNIX User UID in the NFS request (IT Environment). The TOE then searches the IT Environment to get the UNIX User GID and UNIX username (FDP_ACF.1).

7.1.2.1.4.2 Derivation of NTFS-Style Client Subject Security Attributes

If the TOE determines that NTFS-Style security attributes should be used to determine access for an object, the TOE requires two subject security attributes: a Windows User SID and a Windows User GID.

If the access request is initiated by a CIFS Client, the TOE obtained the CIFS Client's username (Windows username) when the client logged onto the remote system and joined the Windows Domain. In addition to this, the IT Environment queried the domain controller to obtain the Windows User SID and the Windows User GID.

7.1.2.1.5 DAC SFP Rules

The DAC SFP rules that apply depend on the subject, the operation, and the object. In addition, the objects file type (directory, symbolic link and regular) is used to determine access and the type of Qtree the file is stored in. The five access modes under the control of the TOE DAC SFP are described below.

CREATE ACCESS REQUEST

To determine if a client has permissions to create a file, the TOE first looks at the parent directory's security attributes.

If the parent directory is NTFS-Style, the TOE uses NTFS-Style security attributes for both subject and object to determine if access is permitted. If the client does not have write and execute privileges to the parent directory, the request is denied. If the client has write and execute privileges for the parent directory, the file is created (FDP_ACF.1). In an NTFS Qtree, the new file inherits the NTFS-Style security attributes from the parent directory (FMT_MSA.3).

If the parent directory is NFSv3 UNIX-Style, the TOE uses NFSv3 UNIX-Style security attributes for both subject and object to determine access. If the client does not have write and execute privileges to the parent directory, the request is denied. If the client has write and execute privileges for the parent directory, the file is created (FDP_ACF.1). In a UNIX Qtree, the new file's NFSv3 UNIX-Style security attributes are determined by the file mode creation mask, also known as the User Mask (umask) of the user-owned process creating the file (FMT_MSA.3).

If the parent directory is NFSv4 UNIX-Style, the TOE uses NFSv4-Style ACL security attributes for the object, and UNIX user UID and GID for the subject. If the client has write and execute privileges for the parent directory, the file is created (FDP_ACF.1). In an NFSv4 UNIX-Style Qtree, the new file inherits the NFSv4 UNIX-Style security attributes from the parent directory (FMT_MSA.3).

READ, WRITE, EXECUTE ACCESS REQUESTS

To determine if a client has permission to read, write or execute a file, the TOE first examines the client type. If a client requests access to a file with NFSv3 UNIX-style security attributes, the TOE uses NFSv3 UNIX-Style security attributes for both subject and object to determine if read, write or execute access request is permitted. If the client has read, write or execute permission for the file, access is permitted (FDP_ACF.1). If the client does not have access, the request is denied.

Otherwise, the TOE uses the file's ACL to determine if read, write or execute permission is allowed. The TOE uses NFSv4 or NTFS-Style security attributes for both subject and object to determine access. The TOE determines if the file's ACEs allow permission for the specific request. If they do, access is granted (FDP_ACF.1). If the ACEs do not grant permission, access is denied.

CLIENT DELETE ACCESS REQUEST

To determine if a client has permission to delete a file, the TOE looks at the styles of the file and parent directory.

NFSv3 UNIX-Style File stored in a UNIX-Style Parent Directory

The TOE, using NFSv3 UNIX-Style security attributes for both subject and object, determines if the client has write and execute access for the file's parent directory. If the client does, the delete access is permitted (FDP_ACF.1). Otherwise, access is denied.

NFSv4 and NTFS-Style File stored in an NTFS-Style Parent Directory

The TOE, using NFSv4 and NTFS-Style security attributes for both subject and object, first determines if the file's ACL grants the client delete access to the file. If so, access is granted (FDP_ACF.1). If the file's ACEs do not grant delete permission for the client, the TOE determines if the parent directory has a DC (Delete Child) ACE that grants access for the subject. If the parent does, delete access is permitted (FDP_ACF.1). Otherwise, access is denied.

CHANGE PERMISSION ACCESS REQUESTS

To determine if a client has permission to change the permissions of a file, the TOE looks at the styles of the file and parent directory.

NFSv3 UNIX-Style File stored in a UNIX-Style Parent Directory

The TOE, using NFSv3 UNIX-Style security attributes for both subject and object, determines if the client has write and execute access for the file's parent directory. If the client does, and the client also has write access for the file, the change permission access is permitted (FDP_ACF.1). Otherwise, access is denied.

NFSv4 and NTFS-Style File stored in an NTFS-Style Parent Directory

The TOE, using NFSv4 and NTFS-Style security attributes for both subject and object, determines if the file's ACL grants the client change permission access to the file. If so, the TOE determines if the parent directory's ACL grants write and execute access for the subject. If so, change permission access is permitted (FDP_ACF.1). Otherwise, access is denied.

CHANGE OWNER ACCESS REQUESTS

The DAC SFP distinguishes between the NFS Client Change Owner (chown) UNIX command and the CIFS Client Change Owner (Change Ownership) command.

NFSv3 Clients

If an NFSv3 Client requests a Change Owner request (chown) for an NTFS-Style file, the request is denied (FDP_ACF.1). If an NFS Client sends a Change Owner request (chown) for an NFSv3 UNIX-Style directory, the request is denied. For other UNIX-Style file types, the TOE determines if the client is root (UNIX User UID is root UID) or the file owner. If the client is root or the file owner, access is allowed (FDP_ACF.1) and the TOE changes the object's owner to the owner specified in the chown request. If the object had an ACL, the TOE removes the ACL.

NFSv4 Clients

If an NFSv4 Client requests a Change Owner request for an NTFS-Style file, the request is denied (FDP_ACF.1). If the file is an NFSv4 UNIX-Style file, the TOE determines if the client has Change Owner ACE privileges for the file. If the client does, access is allowed (FDP_ACF.1). The TOE will replace the existing owner ACE with the new ACE sent in the command. If the NFSv4 Client does not have Change Owner privileges, the request is denied.

CIFS Client

If a CIFS Client requests a Change Owner request for a UNIX-Style file, the request is denied (FDP_ACF.1). If the file is an NTFS-Style file, the TOE determines if the client has Change Owner ACE privileges for the file. If the client does, access is allowed (FDP_ACF.1). The TOE will replace the existing owner ACE with the new ACE sent in the command. If the CIFS Client does not have Change Owner privileges, the request is denied.

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

7.1.3 Identification and Authentication

The TOE's I&A functionality enforces human administrators to identify and authenticate themselves to the TOE before allowing any modifications to TOE managed TSF Data (FIA_UID.2, FIA_UAU.2).

Administrators' authentication credentials are maintained by the TOE in a local registry. The file contains the username, password, full name, password aging, role, and other similar characteristics for each administrator. Authentication credentials are maintained by the TOE in a local registry. Several roles exist for administrator authentication: *admin, autosupport, backup, readonly, and none*.

The TOE enforces minimum password strength requirements. Passwords must have a length of at least 8 characters and contain at least one numeric character and at least one alphabetic character (FIA_SOS.1). The TOE also maintains the following attributes for administrative accounts: *TOE user name, password, group membership*, *UNIX User UID and GID, and Windows User SID and GID* (FIA_ATD.1).

The TOE enforces minimum password strength requirements. The **`security login role config create`** and **`modify`** commands offer parameters to specify the minimum number of alphabetic characters, a mix of alphabetic with numeric, and the number of special characters that a password must contain. The parameters setting password requirements are:

- **`-passwd-minlength –`** This specifies the required minimum length of a password. Possible values range from 3 to 64 characters. The default setting is 8 characters.
- **`-passwd-alphanum –`** This specifies whether a mix of alphabetic and numeric characters is required in the password. If this parameter is enabled, a password must contain at least one letter and one number. This needs to be enabled.
- **`-passwd-min-special-chars –`** This specifies the minimum number of special characters required in a password. Possible values range from 0 to 64 special characters.

The TOE will lock out an administrator account if the user fails to enter the proper credentials after **`–max-failed-login-attempts`** failed login attempts set by the **`security login role config modify`** or **`create`** command. (FIA_AFL.1).

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

7.1.4 Security Management

The Administrative Security Function provides the necessary functions, or capabilities, to allow NetApp cluster and Vserver administrators to manage and support the TSF. Included in this functionality are the rules enforced by the TOE that define access to TOE-maintained TSF Data and TSF Functions. The TSF Functions are categorized into the groups of capabilities listed in Table 22 below:

Table 22 – Security Function Capabilities

Role Configurable Capability	Capabilities
Command Directory Access	Grants the specified role specific command capabilities. This specifies the command or command directory to which the role has access. This includes permission for all variations of the security command.
Access	The possible access level settings are none, read-only, and all. The default setting is all.
Query	This optionally specifies the object that the role is allowed to access. The query object must be applicable to the command or directory name specified by –cmddirname. The query object must be enclosed in double quotation marks (“”), and it must be a valid field name.

Only end users associated with the specific roles as outlined in Table 16 for System Manager may modify the association between users, groups, and any of the above capabilities.

For OCUM, the management functionality available to a user is dependent on the assigned role; the assigned role may be *operator*, *storage administrator*, or *OnCommand administrator*. Each role has access to an increasing set of functionality available via the OCUM web diagnostic interface or API. The maintenance console may only be accessed by the account created during the installation and setup procedures. The default account is known as the ‘maintenance user’ and has the role of *OnCommand administrator*; this is the only user that may access the maintenance console. The maintenance console is used to manage local settings and other maintenance tasks for the virtual appliance. Only users assigned the *OnCommand administrator* role have the capability to manage users for OCUM.

The TOE provides several interfaces for administrators to use to manage the behavior of the TSFs. The various management interfaces available to administrators are outlined below:

CLI

Local CLI available via a serial terminal connected to the console port of the appliance

Remote CLI available via a secure shell program, such as SSH, OpenSSH, PuTTY, etc.

(See section 1.5.3 for a list of other methods of accessing the CLI which are not included in the evaluated configuration of the TOE)

OnCommand™ System Manager GUI

The OnCommand™ System Manager GUI is installed on a separate management workstation. The OnCommand™ System Manager GUI makes API calls to the System Administration TOE component for management of the TOE security functions.

OCUM Web Interface, Maintenance Console, and API

The OCUM is a virtual appliance installed on a separate management workstation on an ESX/ESX(i) hypervisor. The external interfaces of OCUM provide diagnostics based on storage availability, capacity, and protection and allows for operators, storage administrators, and OnCommand administrators to analyze collected data and take corrective or preventative actions if necessary..

7.1.4.1 Management of Security Attributes

The TOE protects TSF data via the implementation of the DAC SFP as described in section 7.1.2.1 above. The security attributes upon which the DAC SFP relies for access control are configurable only by users who are owners of the object or users who are assigned the *admin* role.

The management of security attributes is performed by editing the attributes of individual objects such as the SD of NTFS-style files, the ACL of NFSv4 UNIX-style files, or the nine character access mode string of NFSv3 UNIX-style files, by editing the file's group membership, or by editing a user's membership in a group. For more information on the security attributes of TSF data, see section 7.1.2.1.1 and its subsections above.

7.1.4.2 Management of TSF Data

The TOE's Administration Security Function includes TSF Data Management. The TSF Data Management includes management of both authentication data and security attributes. The following data is managed by the TOE:

- TOE Username Management.
- Deny unauthorized administrative login attempts via Data ONTAP.
- Implement a "Sleep Mode" function call to Data ONTAP to deny access and initiate a time out period for further login attempts, to counter brute force password guessing.

TOE USERNAME MANAGEMENT

The TOE maintains authentication data locally that is used authenticate the NetApp Administrators. This authentication database can only be accessed through the security command.

7.1.4.3 Management of Roles

Within System Manager, the TOE maintains the following roles for users: *admin, autosupport, backup, readonly, none, vsadmin, vsadmin-volume, vsadmin-protocol, vsadmin-backup, and vsadmin-readonly.* The *admin* role has the default capability to administratively access the TOE and modify security attributes. The other administrative roles have varying functionality as defined in Table 16. Within Unified Manager, the TOE maintains the following roles for users: *OnCommand administrator*, *storage administrator*, and *operator*. The *OnCommand administrator* role has the capability to administer users and perform the generic maintenance tasks required for TOE operation. The other roles have varying functionality as defined in the latter half of Table 16.

NetApp Administrators are required to identify and authenticate themselves to the TOE. The authentication data used for I&A, username and password, is maintained locally by the TOE; administration of user authentication data by the IT Environment is not supported. NetApp Administrators are allowed to modify TOE-managed TSF data including authentication data, security attributes and other TSF Data.

Non-administrators are users who access the TOE via a remote system using NFS or CIFS client software (process acting on behalf of a user). Non-administrators have access to TOE managed user data, but do not have authority to modify TOE managed TSF data. Access to TOE managed user data by non-administrators is covered by the TOE's DAC SFP.

TOE Security Functional Requirements Satisfied: FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1(a), FMT_MTD.1(b), FMT_SMF.1, FMT_SMR.1

7.1.5 Protection of the TSF

The TOE protects the TSF via the implementation of SMT (i.e., domain separation) made possible by SVM functionality.

Secure Multi-Tenancy is the use of secure virtual partitions within a shared physical storage environment for the purpose of sharing the physical environment among multiple distinct tenants. SMT allows the consolidation of tenants into shared resources, at the same time providing assurance that tenants cannot access resources not explicitly assigned to them.

Clustered Data ONTAP is an inherently multi-tenant storage operating system and is architect to provide data access through secure virtual storage partitions. A cluster can be a single partition representing the resources of the entire cluster, or can be divided into multiple partitions, each representing specific subset of cluster resources. These secure virtual storage partitions are called Storage Virtual Machines (SVMs).