NetApp Data ONTAP Version 7.2.5.1 Security Target

NetApp, Inc.
8405 Greensboro Drive Suite 1000
McLean, VA 22102
Phone: 703-918-7200
Fax: 703-918-7301

May 14, 2007 – revised January 9, 2009 (version 37)

DOCUMENT INTRODUCTION

Prepared By:

Common Criteria Consulting LLC
15804 Laughlin Lane
Silver Spring, MD 20906

Prepared For:

NetApp, Inc.
8405 Greensboro Dr. Suite 1000
McLean, VA 22102

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Data ONTAP Version 7.2.5.1. 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 IT Security Functions provided by the TOE which meet the set of requirements.

REVISION HISTORY

Version	Description
10	December 18, 2006 – Initial draft
11	January 2, 2007 – Upgrades for Data ONTAP Version 7.2.1
12	January 19, 2007 – Corrections for mixed qtree behaviour
13	January 21, 2007 - Minor edits including Figure 2 changes
14	January 23, 2007 – Editorial changes for internal review
15	January 25, 2007 – Version submitted for evaluation
16	March 8, 2007 – Address OR 01
17	April 9, 2007 – Corrections for admin roles, version 7.2.2
18	April 27, 2007 – Additional correction regarding admin access
19	April 28, 2007 – Additional changes regarding admin access
20	May 14, 2007 – Address issues from ST evaluation
23	October 4, 2007 – Figure 2 and Table 12 Updated; 7.2.4 added to eval.
24	October 22, 2007 – Add FAS2020, FAS2050, and FAS6080 to Table 1
25	November 6, 2007 – Minor edits for version, date, page numbers
26	January 16, 2008 – edits for OR from CSE

28	February 22, 2008 – Restore SEP & REM SFR; added V6080; updated Figure 1 and Table 1
29	February 28, 2009 – Fixes in 5.1.2.5 and Table 16.
30	March 3, 2008 – Update TOC
31	April 23, 2008 – Added audit functionality, Added Version 7.2.5, Removed Table 1 and renumbered Tables, Updated Company Name to NetApp.
32	May 01, 2008 – Corrected: Updated TOC and Index lists, minor formatting corrections, add SFR column to Table 9.
33	May 29, 2008 – Edits for OR 6.
34	July 21, 2008 – Modified: Figure 1, Modified version to 7.2.5.1
35	August 4, 2008 – Minor edits for Roles
36	December 12, 2008 – SFR Corrections
37	January 09, 2009 - DAC table modifications

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYM AND ABBREVIATION LIST

ACE	Access Control Entry
ACL	Access Control List
ADMIN	Administration
ANSI	American National Standards Institute
ATA	Advanced Technology Attachment
CC	Common Criteria
CIFS	Common Internet File System
CLI	Command Line Interface
DAC	Discretionary Access Control
DC	Domain Controller (when used in context of resolving client information)
DC	Delete Child (when used in context of ACEs)
EAL2	Evaluation Assurance Level 2
FTP	File Transfer Protocol
GID	Group ID
ID	IDentifier
IP	Internet Protocol
IT	Information Technology
I&A	Identification and Authentication
LDAP	Lightweight Directory Access Protocol
NAS	Network-Attached Storage
NDMP	Network Data Management Protocol
NIS	Network Information Service
NFS	Network File System
NT	New Technology
NTFS	NT File System
PP	Protection Profile
SAN	Storage Area Network
SD	Security Descriptor
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirement
SID	Security ID
SOF	Strength of Function
ST	Security Target
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
UAC	User Access Control
UDP	User Datagram Protocol
UID	User ID
UNIX	UNiversal Interactive eXecutive
WAFL	Write Anywhere File Layout

CHAPTER 1

1. Security Target Introduction

This Security Target (ST) describes the objectives, requirements and rationale for Data ONTAP Version 7.2.5.1 Target of Evaluation (TOE). The language used in this Security Target is consistent with the *Common Criteria for Information Technology Security Evaluation, Version 2.3*, the ISO/IEC JTC 1/SC27, *Guide for the Production of PPs and STs, Version 0.9* and all international interpretations through January 2, 2007. As such, the spelling of terms is presented using the internationally accepted English.

1.1 ST and TOE Identification

This section provides information needed to identify and control this ST and its TOE.

ST TITLE	NetApp Data ONTAP Version 7.2.5.1 Security Target
ST Version	Version 37
Publication Date	May 14, 2007 revised January 9, 2009
Vendor	NetApp, Inc.
ST Author	Common Criteria Consulting LLC for NetApp, Inc.
TOE Identification	Data ONTAP Version 7.2.5.1 Target of Evaluation
CC Identification	Common Criteria for Information Technology Security Evaluation, Version 2.3
Common Criteria Conformance	The ST is compliant with the Common Criteria (CC) Version 2.3 functional requirements (extended) and assurance requirements for EAL2 augmented by ALC_FLR.3 Systematic flaw remediation.
Protection Profile Conformance	The TOE does not claim conformance to any Protection Profile.
Keywords	Operating System, access control, discretionary access control (DAC).

1.2 TOE Overview

Data ONTAP Version 7.2.5.1 Target of Evaluation (TOE) is a microkernel operating system that supports multi-protocol services and advanced data management capabilities for consolidating and protecting data for enterprise applications and users. NetApp's storage appliances are based on the Data ONTAP Version 7.2.5.1 TOE microkernel operating system.

1.2.1 Security Target Organization

Chapter 1 of this ST provides introductory and identifying information for the TOE.

Chapter 2 describes the TOE and provides some guidance on its use.

Chapter 3 provides a security environment description in terms of assumptions, threats and organizational security policies.

Chapter 4 identifies the security objectives of the TOE and of the Information Technology (IT) environment.

Chapter 5 provides the TOE Security Functional Requirements, as well as requirements on the IT Environment.

Chapter 6 is the TOE Summary Specification, a description of the Security Functions and assurance requirements provided by Data ONTAP Version 7.2.5.1 TOE.

Chapter 7 identifies claims of conformance to a registered Protection Profile (PP).

Chapter 8 provides a rationale for the security objectives, Security Functional Requirements, TOE summary specification and PP claims.

CHAPTER 2

2. TOE Description

This section provides the context for the TOE evaluation by identifying the product and describing the evaluated configuration.

2.1 TOE Overview

Data ONTAP Version 7.2.5.1 TOE is a proprietary microkernel operating system developed by NetApp. The microkernel is included in the distribution of several of NetApp's storage solution products including Filer, V-Series Virtual Filer (Virtual Filer), and NearStore. The Data ONTAP Version 7.2.5.1 TOE provides data management functions that include providing secure data storage and multi-protocol access.

2.2 TOE Functionality Included in the Logical Boundary

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 including cross-protocol support.

Management

The Management functionality included in the TOE's logical boundary supports functionality that 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.

2.3 Product Overview

The TOE is a software product that is distributed with the following NetApp storage solution products:

Filer NetApp's Filer systems offer seamless access to a full range of enterprise data for users on a variety of platforms. Filer systems support NFS and CIFS for file access, as well as FCP and iSCSI for block-storage access.

Virtual Filer The Virtual Filer product family (V-Series) provides unified NAS and SAN access to data stored in Fibre Channel SAN storage arrays enabling data center storage deployment.

NearStore NearStore is a disk-based nearline storage solution and offers additional functionality including simplified backup, accelerated recovery and robust remote disaster recovery.

For a complete list of NetApp Storage Controllers on which each version of the TOE operates, refer to the release notes for the specific version of Data ONTAP®. Both single controller and High Availability controller pairs, where the platform allows, are supported by the TOE.

2.4 TOE Component Systems

Data ONTAP Version 7.2.5.1 TOE (hereafter referred to as Data ONTAP) is divided into three components: System Administration, WAFL and Operating System Kernel. The three modules are described below. Their relationship to the IT Environment supplied components is depicted in Figure 1.

WAFL

The TOE's WAFL module is responsible for implementing the TOE's DAC 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).

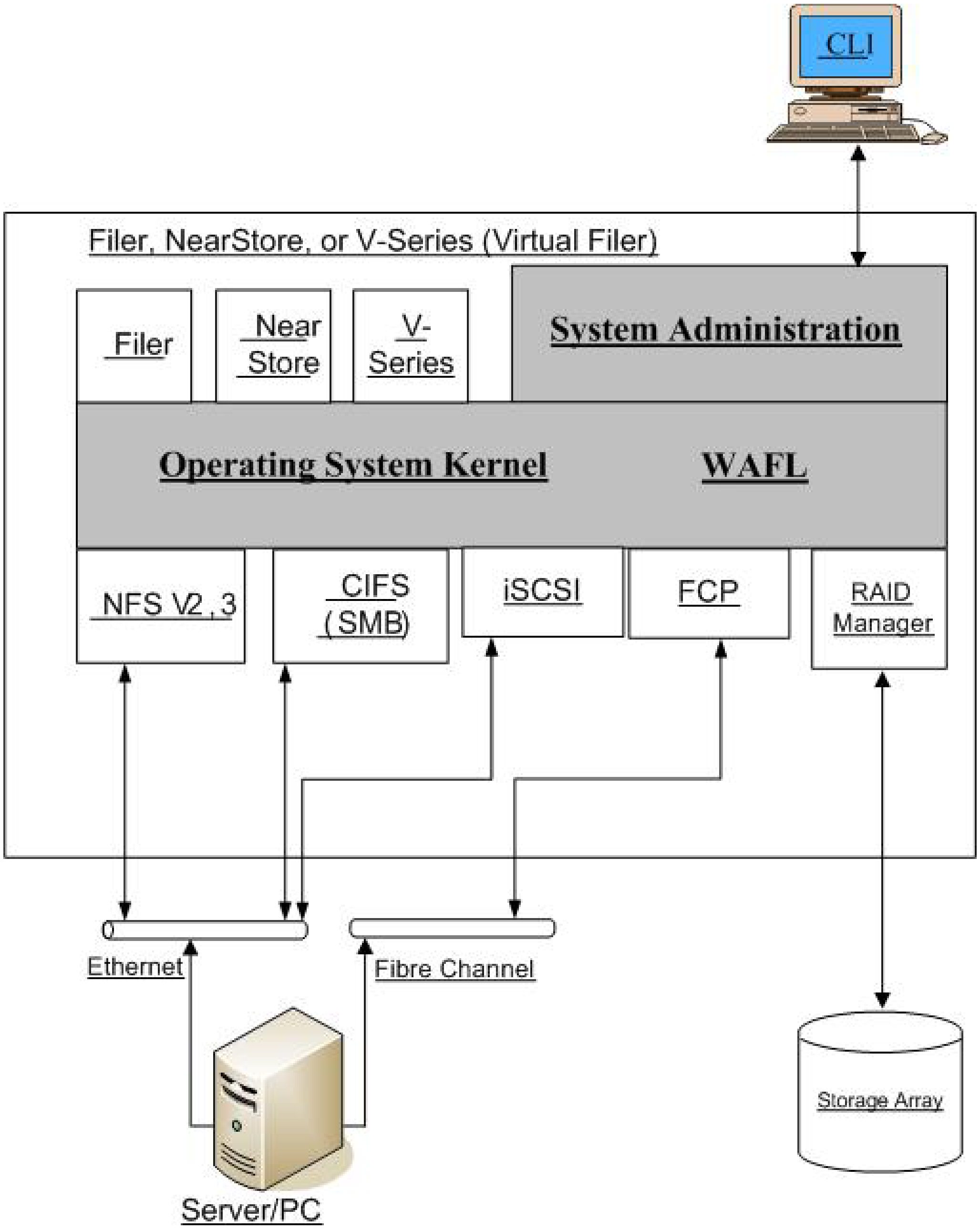
System Administration

The System Administration module includes providing an operator 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 the NetApp Administrator role, and this functionality is only available via the local Command Line Interface (CLI) on the TOE. System Administration functions performed at the CLI are audited by default.

Operating System Kernel

The Kernel provides the communications between the components of the Operating System.

Figure 1 - TOE Components

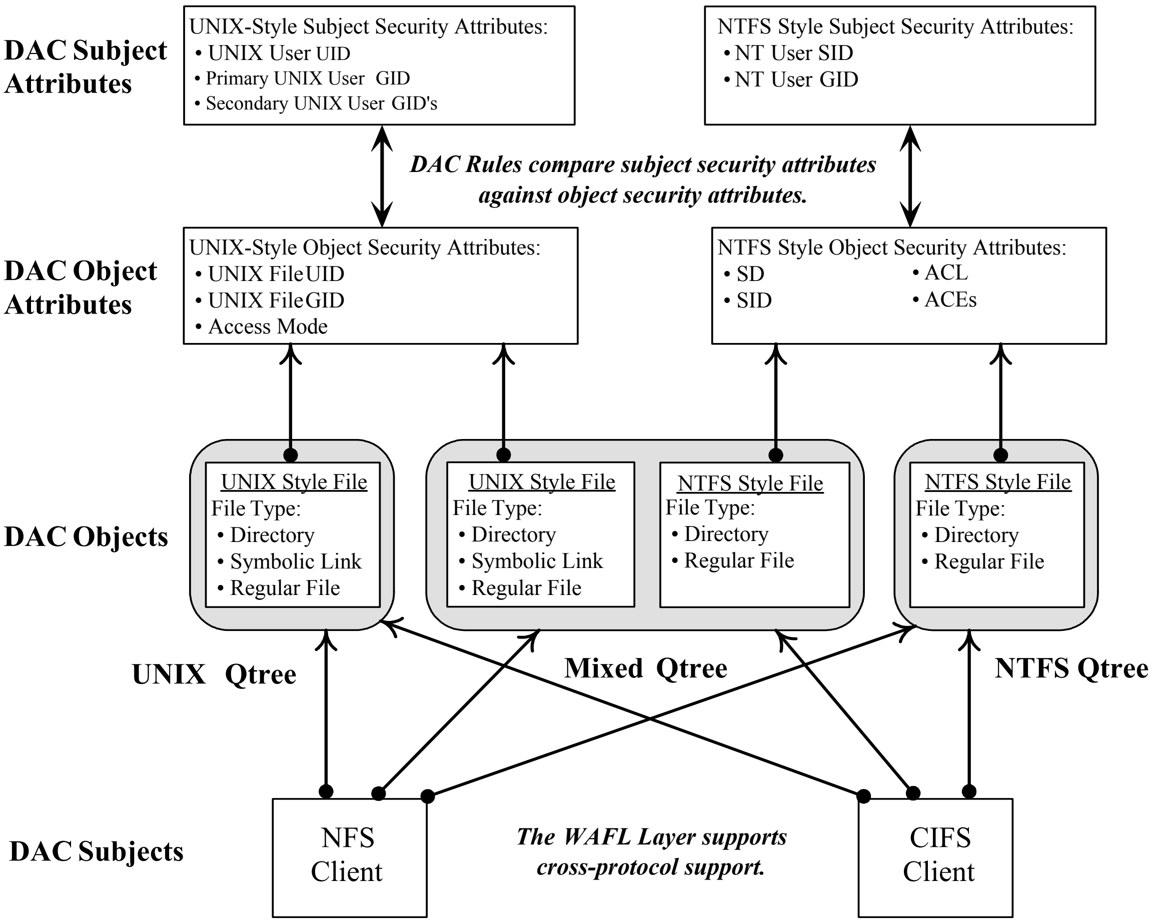


The TOE boundary is illustrated by the shaded boxes in the figure above. The TOE does not include any hardware or firmware.

2.4.1 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 originate from end users on remote systems that access the TOE via NFS or CIFS. Figure 2 below depicts the WAFL functionality.

Figure 2 - WAFL Functionality Detail



2.4.1.1 Files

The TSF User Data that is covered by the DAC SFP are files. Each file maintained by the TOE has a file style associated with it. 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 attributes. Additionally, a file may be both a UNIX-Style file and an NTFS-Style file.

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.

In addition to the file type, the TOE maintains three different storage types: UNIX qtrees, NTFS qtrees and 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 style 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.

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. When creating a file in a Mixed qtree, the TOE will create security attributes that correspond to the protocol used by the client creating the file. The TOE will create NTFS-Style attributes for a file created by a CIFS client and create UNIX-Style attributes for a file created by an NFS client.

2.4.1.2 Clients

The TOE supports two client protocols: NFS Clients and CIFS Clients. Both clients access the TOE via remote system client software that interfaces to the IT Environment's NFS or CIFS server implementation. The TOE interfaces to the IT Environment's NFS and CIFS servers.

To determine if file access is allowed, the TOE compares a client's security attributes with the file's security attributes. 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, 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, primary UNIX User GID and any secondary UNIX User GIDs. 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. However, the TOE also supports cross-protocol access: NFS Clients can be mapped to NTFS-Style security attributes and CIFS Clients can be mapped to UNIX-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 logged onto the remote system and joined the Windows domain (which the TOE is a member of). 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.

Cross-protocol access requires additional TSF data (usernames) to resolve the appropriate client security attributes. UNIX User UIDs and Windows User UIDs (Windows User SIDs) are not directly mapped by the TOE. Instead, UIDs are mapped to the username associated with the UID, the username is then mapped to the other protocol's username, and then this new username is used to find the new protocol's UID.

2.5 TOE and TOE Relied Upon IT Environment

2.5.1 TOE Relied Upon IT Environment Hardware

The IT Environment Hardware includes the appliance hardware of the Filer, V-Series Virtual Filer, and NearStore.

2.5.2 TOE Relied Upon IT Environment Software

CIFS Client software, CIFS Server software, NFS Client software, and NFS Server software

The TOE provides secure access to files under control of the TOE. The TOE provides a protective layer between NFS Clients and CIFS Clients and NFS Servers and CIFS Servers ensuring that only authorized users (clients) can access TOE protected files. The CIFS Client, NFS Client, CIFS Server and NFS Server (NFS V2 and NFS V3) software is supplied by the IT Environment (Figure 1).

Filer, V-Series Virtual Filer, and NearStore Products

The TOE is a proprietary microkernel operating system. The microkernel is included in the distribution of several of NetApp's storage solution products including Filer, V-Series Virtual Filer, and NearStore. The product functionality provided by the Filer, V-Series Virtual File and NearStore products is supplied by the IT Environment.

RAID Manager

The RAID Manager supports multiple disk drives which provide fault tolerance and performance. The RAID Manager is supplied by the IT Environment.

TCP/IP Protocol

The UDP/TCP/IP protocol stack is supplied by the IT Environment.

2.6 Physical Boundary

The TOE's physical boundary includes the WAFL component and the System Administration components described in section 2.4. Figure 1 depicts the TOE's physical boundary (shaded portions) in relationship to the IT Environment supplied components.

2.6.1 TOE Data

The following sections describe the TOE data included in the TOE's physical boundary.

2.6.1.1 User Data

The User Data included in the TOE's physical boundary includes the files protected by the DAC SFP.

2.6.1.2 TSF Data

The following section identifies the TSF Data included in the TOE's physical boundary.

2.6.1.2.1 Files

/etc/usermap.cfg

A TOE resident appliance file that contains Windows Username and UNIX Username mappings used for cross protocol access for both CIFS Clients and UNIX Clients.

wafl.default_unix_user

A TOE resident appliance setting that contains a default UNIX Username used to resolve a CIFS Client's Windows Username for cross protocol access. This specifies the UNIX user account to use when an authenticated Windows user does not match an entry in the usermap.cfg file. If this option is set to the null string, Windows users which are not matched in the usermap.cfg file will not be allowed to gain access. The default value for this option is 'pcuser'. This default user name is used to determine the user permissions.

wafl.default_nt_user
A TOE resident appliance setting that contains a default Windows Username used to resolve an NFS Client's UNIX Username for cross protocol access. This specifies the Windows user account to use when a UNIX user accesses a file with Windows security (has an ACL), and that UNIX user would not otherwise be mapped. If this option is set to the null string, such accesses will be denied. The default value for this option is the null string. This default user name is used to determine the user permissions.

/etc/log/auditlog
CLI Audit log. The auditlog file keeps a record of all administrator operations performed on the storage system and the administrator who performed it, as well as any operations that failed due to insufficient capabilities.

/etc/log/cifsaudit.alf
Internal TOE Audit log file for recording enabled security events.

/etc/log/adtlog.evt
Default External (exportable) Audit log file name.

2.6.1.3 Security Attributes

Figure 2 above depicts the security attributes that are used by the TOE's Discretionary Access Control (DAC) SFP and included in the TOE's physical boundary. The security attributes include attributes for clients (NFS Clients and CIFS Clients) and attributes for files managed by the TOE (UNIX-Style file and NTSF-Style files). The following section describes the security attributes included in the TOE's physical boundary.

2.6.1.3.1 Client Security Attributes

Windows User GID
The Windows group ID. Each user in a Windows system is assigned to a group and that group is assigned a unique GID.

Windows User SID
The Windows user ID number. Each user in a Windows system is assigned a unique Windows User UID.

Primary UNIX User GID
The UNIX user GID number. Each user in an UNIX system is assigned to a group and that group is assigned a unique GID.

Secondary UNIX User GID
The UNIX user GID number. Each user in an UNIX system may be assigned to a secondary group and if so, the Secondary UNIX User GID reflects the additional group ID.

UNIX User UID
The UNIX user ID number. Each user in a UNIX system is assigned a unique UNIX User UID.

2.6.1.3.2 File Security Attributes

Access Control Entry (ACE)

A data structure associated with NTFS-Style files. Each ACE explicitly allows or denies access to a user or group for a specific NTSF-Style supported operation.

Access Control List (ACL)

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 identifies 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 (SD)

A data structure associated with NTFS-Style files. A SD contains a SID and an ACL.

Security ID (SID)

The CIFS User SID of the file’s owner.

UNIX File GID

A UNIX File GID identifies the groups associated with the UNIX-Style file.

UNIX File UID

The UNIX User UID of the file’s owner.

2.7 TSF Functional Summary

The TOE’s security functions are described below and described in detail in Chapter 6.

DAC

The DAC security function claimed by the TOE enforces access rules to user data (files) maintained by the TOE based on client type, client security attributes, file type, file security attributes, and operation. DAC is implemented by the TOE’s WAFL component.

Administrative

The Administrative security function claimed by the TOE includes supporting operator functions including enforcing identification and authentication, user roles and providing the necessary user interface commands that enable a NetApp Administrator to support the TOE's security functionality.

Self Protection

The TOE provides for self protection and non-bypassability of functions within the TOE's scope of control (TSC). The TOE controls actions carried out by a user by controlling a user session and the actions carried out during a user session. By maintaining and controlling a user session a user has with the TOE, the TOE ensures that no security functions within the TSC are bypassed and that there is a separate domain for the TOE that prevents the TOE from being interfered with or tampered with for those users that are within the TSC.

Audit

The TOE generates audit events for each administrator login and configuration change.

2.8 Rationale for Non-Bypassability and Separation for the TOE

The responsibility for non-bypassability and non-interference is split between the TOE and the IT Environment. The WAFL and System Administration TOE components are software only components and therefore, the non-bypassability and non-interference claims are dependent upon hardware mechanisms.

Non-bypassability

> The TOE ensures that the security policy is applied and succeeds before further processing is permitted whenever a security relevant interface is invoked: permissions are verified before any application may begin execution, and management actions are limited to the permissions of the authenticated users. Non-security relevant interfaces do not interact with the security functionality of the TOE. The OS ensures that the security relevant interfaces are invoked: all incoming network packets are delivered to the TOE for inspection and attempts to invoke applications are validated by the TOE before the applications begin execution.

Non-interference

> The TOE is implemented with well defined interfaces that can be categorized as security relevant or non-security relevant. The TOE is implemented such that non-security relevant interfaces have no means of impacting the security functionality of the TOE. Security relevant management interfaces maintain appropriate access permissions to TOE data according to the authenticated user utilizing the management interface. Unauthenticated users may not perform any actions via the TOE management interfaces. The hardware provides virtual memory and user/kernel separation which the TOE utilizes to ensure that other (non-TOE) processes may not interfere with the TOE; all interactions are limited to the defined TOE interfaces.

2.9 TOE Evaluated Configuration

2.9.1 TOE Evaluated Configuration Systems

The evaluated configuration will include one or more instances of a product in one of the following NetApp families: Filer, V-Series Virtual Filer, and NearStore.

2.9.2 TOE Evaluated Configuration Options

The following sections describe the evaluated configuration options.

2.9.2.1 Access Protocol Options

The IT Environment supports multiple protocol servers. The evaluated configuration supports NFS and CIFS clients only. The evaluated configuration also supports FCP and iSCSI access protocols. The following servers are disabled: telnet, tftp, ftp, ndmp and http.

2.9.2.2 Miscellaneous

A) The wafl.root_only_chown option for the evaluated configuration is disabled. When Enabled, only a root user has permission to change the owner of a file. When Disabled, the wafl.root_only_chown option enables the owner of a file to change ownership of a file.

B) All authorized NetApp Administrators have the NetApp Administrator role.

C) NetApp Administrator access is via the local console only.

D) The security.admin.authentication parameter is set to "internal."

E) Primary and secondary UNIX primary GIDs are evaluated; multiple client UNIX User GIDs are included in the evaluated configuration.

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

G) The evaluation configuration disables CIFS and NFS access to the /etc directory.

2.10 Functionality Excluded From the Evaluation

The following functionality is excluded from the evaluation:

A) Shared level ACLs

B) Bypass traverse checking option

C) Password aging

D) Windows Group Policy Objects

E) Native File Blocking (File Screening)

F) NFSv4

G) Administrator account lockout

H) Administrator password reuse limits

I) Kerberos

CHAPTER 3

3. Security Environment

This chapter identifies the following:

A) Significant assumptions about the TOE's operational environment.

B) IT related threats to the organization countered by the TOE.

C) Environmental threats requiring controls to provide sufficient protection.

D) Organizational security policies for the TOE as appropriate.

This document uses the following naming conventions to identify the assumptions and threats: Assumptions are identified by an A. and followed by the assumption name (e.g. A.PEER). Threats are identified by a T. and followed by the threat name (e.g. T.ADMIN).

3.1 Assumptions

The specific conditions listed in the following subsections are assumed to exist in the TOE's IT Environment. This includes information about the connectivity, personnel, and physical aspects of the environment.

3.1.1 Connectivity Assumptions

The TOE is intended for use in areas that have physical control and monitoring. It is assumed that the following connectivity conditions will exist.

Table 1 - Connectivity Assumptions

Assumption	Assumption Description (Connectivity)
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.

3.1.2 Personnel Assumptions

The TOE is intended to be managed by competent non-hostile individuals. It is assumed that the following personnel conditions will exist.

Table 2 - Personnel Assumptions

Assumption	Assumption Description (Personnel)
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.

3.1.3 Physical Assumptions

The TOE is intended for use in areas that have physical control and monitoring. It is assumed that the following physical conditions will exist.

Table 3 - Physical Assumptions

Assumption	Assumption Description (Physical)
A.PROTECT	The processing resources of the TOE critical to the security policy enforcement will be protected from unauthorized physical modification by potentially hostile outsiders.

3.2 Threats

Table 5 identifies the threats against the TOE and the TOE's operational environment.

Table 4 - Threats

Threat	Threat Description
T.CONFIG_CORRUPT	Configuration data or other trusted data may be tampered with by unauthorized users due to failure of the system to protect this data.
T.UNAUTH_ACCESS	An unauthorized user may attempt to access TOE data or Security Functions by bypassing a security mechanism.

3.3 Organizational Security Policies

The following table identifies the Organizational Security Policies (OSPs) with which the TOE must comply.

OSP	OSP Description
P.ADMIN_ACCESS	Administrative functionality shall be restricted to authorized administrators.
P.AUDIT	All administrator authentication attempts, whether successful or unsuccessful, as well as configuration changes must be audited.
P.USER_ACCESS	Authorized users shall only be granted access to user data for which they have been authorized.

CHAPTER 4

4. Security Objectives

The chapter identifies the security objectives for the TOE, the IT Environment and the non-IT Environment.

This document uses the following naming conventions to identify the security objectives: Security Objectives for the TOE are identified by an O. and followed by the security objective name (e.g. O.ACCESS). Security Objectives for the IT Environment are identified by an O.E. and followed by the security objective name (e.g. O.E.ACCESS). Security Objectives for the non-IT Environment are identified by an O.N. and followed by the security objective name (e.g. O.N.ACCESS).

4.1 Security Objectives for the TOE

Table 5 lists the security objectives for the TOE and their descriptions. These objectives describe the security functionality that is to be achieved by the TOE.

Table 5 - Security Objectives for the TOE

Security Objective (TOE)	TOE Security Objective 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 configuration changes.
O.DAC_ACC	The TOE will control access to user data based on the identity of users and groups of users.
O.ENFORCE	The TOE is designed and implemented in a manner that ensures the security policies can't be bypassed or interfered with via mechanisms within the TSC.
O.I&A	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.

4.2 Security Objectives for the Environment

The following sections describe the objectives for the TOE's environment. These objectives describe properties of the operational environment of the TOE necessary in order for the TOE to be able to provide its security functionality.

4.2.1 Security Objectives for the IT Environment

Table 6 identifies the security objectives for the TOE's IT Environment.

Table 6 - Security Objectives for the IT Environment

Security Objective (IT Environment)	IT Environment Security Objective Description
O.E.ACCESS	The IT Environment will ensure that users gain only authorized access to the data the IT Environment manages.
O.E.ADMIN_ROLES	The IT Environment will provide administrative roles to isolate administrative actions.
O.E.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 TSC.
O.E.I&A	The IT Environment must require authorized CIFS and NFS Clients to successfully I&A before allowing access to the TOE.
O.E.SUBJECTDATA	The IT Environment will provide the TOE with the appropriate subject security attributes.
O.E.TIME	The IT Environment will provide a reliable timestamp for use by the TOE.

4.2.2 Security Objectives for the Non-IT Environment

Table 7 identifies the security objectives for the TOE's Non-IT Environment. These objectives describe properties of the non-IT operational environment of the TOE necessary in order for the TOE to be able to provide its security functionality.

Table 7 - Security Objectives for the Non-IT Environment

Security Objective (Non-IT Environment)	Non-IT Environment Security Objective Description
O.N.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.
O.N.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.
O.N.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE and the IT Environment critical to security policy are protected from any physical attack that might compromise the IT security objectives.
O.N.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.

CHAPTER 5

5. Security Requirements

This section identifies the Security Functional Requirements for the TOE and for the IT Environment. The Security Functional Requirements included in this section are derived verbatim from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 2.3* and all international interpretations with the exception of the items listed below.

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC.

Assignments: *indicated in italics*.

Selections: <u>indicated in underlined text.</u>

Assignments within selections: <u>*indicated in italics and underlined text*</u>.

Refinements: **indicated with bold text.**

Multiple Security Functional Requirement instances (iterations) are identified by the Security Functional Requirement component identification followed by the instance number in parenthesis (e.g. FAU_SAR.1(1)) and the Security Functional Requirement element name followed by the instance number in parenthesis (e.g. FAU_SAR.1.1(1)). This document continues the iteration numbering for Security Functional Requirements that apply to both the TOE and the IT Environment.

5.1 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear in Table 8 are described in more detail in the following subsections.

Table 8 - Security Functional Requirements of the TOE

Security Functional Requirement (TOE)	Security Functional Requirement Name
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FDP_ACC.1	Subset Access Control
FDP_ACF.1	Security Attribute Based Access Control
FIA_UAU.2	User Authentication Before any Action
FIA_UID.2	User Identification Before any Action
FMT_MSA.1	Management of Security Attributes
FMT_MSA.3	Static Attribute Initialisation
FMT_MTD.1	Management of TSF Data

Security Functional Requirement (TOE)	Security Functional Requirement Name
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security Roles
FPT_RVM_SW_EXP.1	Non-Bypassability of the TSP for Software TOEs
FPT_SEP_SW_EXP.1	TSF Domain Separation for Software TOEs

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_GEN.1 Audit Data Generation

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

a) Start-up and shutdown of the audit functions;

b) All auditable events for the not specified level of audit; and

c) *The events specified in the following table.*

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *the additional information specified in the following table.*

Table 9 - Audit Generation Details

SFR Addressed	Auditable Events	Additional Event Information
FIA_UAU.2(1), FIA_UID.2(1)	Successful local logon	User identity
FIA_UAU.2(1), FIA_UID.2(1)	Unsuccessful local logon	User identity supplied
FMT_SMF.1(1)	User created	Userid created, userid of the administrator performing the action
FMT_SMF.1(1)	User deleted	Userid deleted, userid of the administrator performing the action
FMT_SMF.1(1)	Group created	Group created, userid of the administrator performing the

SFR Addressed	Auditable Events	Additional Event Information
		action
FMT_SMF.1(1)	Group deleted	Group deleted, userid of the administrator performing the action
FMT_SMF.1(1)	Group member added	Userid and group associated, userid of the administrator performing the action
FMT_SMF.1(1)	Group member deleted	Userid and group disassociated, userid of the administrator performing the action

5.1.1.2 FAU_GEN.2 User Identity Association

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.2 User Data Protection (FDP)

5.1.2.1 FDP_ACC.1(1) Subset Access Control

FDP_ACC.1.1(1) The TSF shall enforce the *Discretionary Access Control (DAC) SFP* on *the subjects, objects, and operations among subjects and objects listed below.*

Table 10 - *FDP_ACC.1.1(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*	
NFS Client	*UNIX-Style file*	*Directory, Symbolic link, Regular file*	*UNIX Qtree or Mixed Qtree*	*Create, read, write, execute, delete, change permissions, change ownership*
	NTFS-Style file	*Directory, Regular file*	*NTFS Qtree or Mixed Qtree*	*Create, read, write, execute, delete, change permissions*
CIFS Client	*NTFS-Style file*	*Directory, Regular file*	*NTFS Qtree or Mixed Qtree*	*Create, read, write, execute, delete, change permissions, change ownership*
	UNIX-Style file	*Directory, Regular file*	*UNIX Qtree or Mixed Qtree*	*Create, read, write, execute, delete, change permissions*

5.1.2.2 FDP_ACF.1(1) Security Attribute Based Access Control

FDP_ACF.1.1(1) The TSF shall enforce the *DAC SFP* to objects based on *the following:*

Table 11 - *FDP_ACF.1.1(1) Detail*

Operation	*Subject*	*Object (File)*	*Subject*		*Object (file) Security Attribute*	*Other Objects and Security Attribute used for DAC SFP*
			Security Attribute	*Other TSF Data*		
Create	*NFS Client*	*UNIX-Style file*	*Windows User SID, Windows User GID, UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs*	*UNIX Username, Windows Username*	*N/A*	*Qtree type, Parent directory’s SID and ACEs, UNIX Parent Directory UID, UNIX Parent Directory GID and access mode*
	CIFS Client		*Windows User SID, UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs*	*Windows Username, UNIX Username*	*N/A*	*UNIX Parent Directory UID, UNIX Parent Directory GID and access mode*

Operation	Subject	Object (File)	Subject		Object (file) Security Attribute	Other Objects and Security Attribute used for DAC SFP
			Security Attribute	Other TSF Data		
	NFS Client	NTFS-Style File	UNIX User UID, Windows User SID, Windows User GID	UNIX Username, Windows Username	N/A	Parent directory’s SID and ACEs
	CIFS Client		Windows User SID, Windows User GID, UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs	UNIX Username, Windows Username	N/A	Qtree type, Parent directory’s SID and ACEs, UNIX Parent Directory UID, UNIX Parent Directory GID and access mode
Read, Write, Execute	NFS Client	UNIX- Style file	UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs	None	UNIX file UID, UNIX file GID, access mode	None

Operation	Subject	Object (File)	Subject		Object (file) Security Attribute	Other Objects and Security Attribute used for DAC SFP
			Security Attribute	Other TSF Data		
	CIFS Client		Windows User SID, UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs	Windows Username, UNIX Username	UNIX file UID, UNIX file GID, access mode	None
	NFS Client	NTFS-Style File	UNIX User UID, Windows User SID, Windows User GID	UNIX Username, Windows Username	SID and ACEs	None
	CIFS Client		Windows User SID, Windows User GID	Windows Username	SID and ACEs	None
Delete	NFS Client or CIFS Client	UNIX- Style file	Windows User SID, Windows User GID, UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs	UNIX Username, Windows Username	None	Qtree type, Parent directory's SID and ACEs, UNIX Parent Directory UID, UNIX Parent Directory GID and access mode

Operation	*Subject*	*Object (File)*	*Subject*		*Object (file) Security Attribute*	*Other Objects and Security Attribute used for DAC SFP*
			Security Attribute	*Other TSF Data*		
	NFS Client or CIFS Client	*NTFS-Style File*	*Windows User SID, Windows User GID, UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs*	*UNIX Username, Windows Username*	*SID and ACEs*	*Qtree type, Parent directory's SID and ACEs, UNIX Parent Directory UID, UNIX Parent Directory GID and access mode*
Change Permission	*NFS Client or CIFS Client*	*UNIX- Style file*	*Windows User SID, Windows User GID, UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs*	*UNIX Username, Windows Username*	*None*	*Qtree type, Parent directory's SID and ACEs, UNIX Parent Directory UID, UNIX Parent Directory GID and access mode*

Operation	*Subject*	*Object (File)*	*Subject*		*Object (file) Security Attribute*	*Other Objects and Security Attribute used for DAC SFP*
			Security Attribute	*Other TSF Data*		
	NFS Client or CIFS Client	*NTFS- Style file*	*Windows User SID, Windows User GID, UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs*	*UNIX Username, Windows Username*	*SID and ACEs*	*Qtree type, Parent directory's SID and ACEs, UNIX Parent Directory UID, UNIX Parent Directory GID and access mode*
Change Owner	*NFS Client*	*UNIX- Style file*	*UNIX User UID*	*None*	*None*	*None*
	CIFS Client	*NTFS-Style file*	*Windows User SID, Windows User GID*	*None*	*SID and ACEs*	*None*

FDP_ACF.1.2(1) 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 is true:*

Table 12 - *FDP_ACF.1.2(1) Detail*

Subject	*Operation*	*Object (File)*			*DAC Rule*
		Qtree Style	*File Style*	*Parent Directory Style*	
NFS Client or CIFS Client	*Create*	*UNIX or Mixed*	*N/A*	*UNIX*	*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).*
		NTFS or Mixed	*N/A*	*NTFS*	*4. 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).* *5. 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).*

Subject	*Operation*	*Object (File)*			*DAC Rule*
		Qtree Style	*File Style*	*Parent Directory Style*	
	Read, Write, Execute	*UNIX or Mixed*	*UNIX*	*N/A*	*6. The subject is the owner of the file and the owner has been granted access for the specific operation (UNIX-Style security attributes).* *7. 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).* *8. 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)*
		NTFS or Mixed	*NTFS*	*N/A*	*9. 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).* *10. 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	*UNIX or Mixed*	*UNIX*	*UNIX*	*11. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory).*

Subject	Operation	Object (File)			DAC Rule
		Qtree Style	File Style	Parent Directory Style	
		NTFS or Mixed	NTFS	NTFS	12. Rule 9 or 10 above is true for Delete operation (subject has Delete NTFS-Style permission for object). 13. Rule 12 above fails and Rule 14 or 15 below are true (subject has Delete Child NTFS-Style permission for parent directory) 14. 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). 15. 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).
		Mixed	NTFS	UNIX	16. Rule 9 or 10 above is true for Delete operation (subject has Delete NTFS-Style permission for object). 17. Rule 16 above fails and Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory).
		Mixed	UNIX	NTFS	18. Rule 14 or 15 above is true (subject has Delete Child NTFS-Style permission for the parent directory).
	Change Permission	UNIX or Mixed	UNIX	UNIX	19. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory) and rule 6 (UNIX-Style permission for object).

Subject	*Operation*	*Object (File)*			*DAC Rule*
		Qtree Style	*File Style*	*Parent Directory Style*	
		NTFS or Mixed	*NTFS*	*NTFS*	*20. Rule 4 or 5 above is true (subject has Write and Execute NTFS-Style permission for parent directory) and rule 9 or 10 above is true for Change Permission operation (NTFS-Style permission for object).*
		Mixed	*UNIX*	*NTFS*	*21. Rule 4 or 5 above is true (subject has Write and Execute NTFS-Style permission for parent directory) and rule 6 (UNIX-Style permission for object).*
		Mixed	*NTFS*	*UNIX*	*22. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory) and rule 9 or 10 above is true for Change Permission operation (NTFS-Style permission for object).*
CIFS Client	*Change Ownership*	*NTFS or Mixed*	*NTFS*	*N/A*	*23. Rule 9 or 10 above is true for Change Ownership operation (subject has Change Owner NTFS-Style permission for object).*
NFS Client	*Change Ownership*	*UNIX or Mixed*	*UNIX*	*N/A*	*24. If the UNIX UID is root, the operation is allowed.*

FDP_ACF.1.3(1) The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *access is granted if one of the following conditions is true:*

Table 13 - *FDP_ACF.1.3(1) Detail*

Subject	*Operation*	*Object*	*DAC Rule*
NFS Client	*Change Owner*	*UNIX-Style file*	*The subject is root*

FDP_ACF.1.4(1) The TSF shall explicitly deny access of subjects to objects based on the *following additional rules: access is denied if one of the following conditions is true.*

Table 14 - *FDP_ACF.1.4(1) Detail*

Subject	*Operation*	*Object*	*DAC Rule*
NFS Client	*Change Owner*	*NTFS-Style file*	*Request denied* *(If subject does not have an Administrative Role.)*
CIFS Client	*Change Owner*	*UNIX-Style file*	*Request denied* *(If subject does not have an Administrative Role in the CIFS Domain)*

5.1.2.3 FIA_UAU.2(1) User Authentication Before any Action

FIA_UAU.2.1(1) The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note: This iteration addresses NetApp administrator login via the local console. CIFS and NFS Clients are addressed by the IT Environment.

5.1.2.4 FIA_UID.2(1) User Identification Before any Action

FIA_UID.2.1(1) The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Application note: This iteration addresses NetApp administrator login via the local console. CIFS and NFS Clients are addressed by the IT Environment.

5.1.3 Security Management (FMT)

5.1.3.1 FMT_MSA.1(1) Management of Security Attributes

FMT_MSA.1.1(1) The TSF shall enforce the *Discretionary Access Control (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 *a NetApp Administrator*.

5.1.3.2 FMT_MSA.1(2) Management of Security Attributes

FMT_MSA.1.1(2) The TSF shall enforce the *Discretionary Access Control (DAC) SFP* to restrict the ability to modify, delete, *add* the security attributes *Secondary TOE User GIDs maintained locally by the TOE* to *a NetApp Administrator*.

5.1.3.3 FMT_MSA.3(1) Static Attribute Initialisation

FMT_MSA.3.1(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(1) The TSF shall allow the *none* to specify alternative initial values to override the default values when an object or information is created.

5.1.3.4 FMT_MTD.1(1) Management of TSF Data

FMT_MTD.1.1(1) The TSF shall restrict the ability to modify, delete, *add* the *TOE Username* to *a NetApp Administrator*.

5.1.3.5 FMT_MTD.1(2) Management of TSF Data

FMT_MTD.1.1(2) The TSF shall restrict the ability to modify, delete, *add* the *TOE User Password* to *a NetApp Administrator*.

5.1.3.6 FMT_MTD.1(3) Management of TSF Data

FMT_MTD.1.1(3) The TSF shall restrict the ability to modify, delete, *add* the *Windows Username and UNIX Username mapping stored in /etc/usermap.cfg* to *a NetApp Administrator*.

5.1.3.7 FMT_MTD.1(4) Management of TSF Data

FMT_MTD.1.1(4) The TSF shall restrict the ability to modify, delete, *add* the *UNIX Username stored in the wafl.default_unix_user value* to *a NetApp Administrator*.

5.1.3.8 FMT_MTD.1(5) Management of TSF Data

FMT_MTD.1.1(5) The TSF shall restrict the ability to modify, delete, *add* the *Windows Username stored in the wafl.default_nt_user value* to *a NetApp Administrator*.

5.1.3.9 FMT_SMF.1(1) Specification of Management Functions

FMT_SMF.1.1(1) The TSF shall be capable of performing the following security management functions:

1) *Provides a CLI management interface accessible via the TOE's hardware serial port that provides an interface to enable a NetApp Administrator to manage TSF Data and configure TSF Functions.*

2) *Provides I&A functions that require a NetApp Administrator to identify and authenticate themselves to the TOE before allowing any modifications of TSF Data.*

3) *Provides a CLI function that enables a NetApp Administrator to specify DAC subject security attribute resolution.*

5.1.3.10 FMT_SMR.1(1) Security Roles

FMT_SMR.1.1(1) The TSF shall maintain the roles *NetApp Administrator and non-administrator*.

FMT_SMR.1.2(1) The TSF shall be able to associate users with roles.

Application note: NetApp Administrators access the TOE via the local TOE and have access to the CLI. Non-administrators are users that do not have administrative access on the TOE or on the remote systems.

5.1.4 Protection of the TSF (FPT)

5.1.4.1 FPT_RVM_SW_EXP.1 Non-Bypassability of the TSP for Software TOEs

Rationale for explicitly stated SFR: Software TOEs are unable to fully satisfy FPT_RVM by themselves. This explicitly stated SFR states the portion of FPT_RVM that can be addressed by the TOE. See FPT_RVM_HW_EXP (levied on the IT Environment) for the remaining functionality.

FPT_RVM_SW_EXP.1.1 The TSF, when invoked shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed

5.1.4.2 FPT_SEP_SW_EXP.1 TSF Domain Separation for Software TOEs

Rationale for explicitly stated SFR: Software TOEs are unable to fully satisfy FPT_SEP by themselves. This explicitly stated SFR states the portion of FPT_SEP that can be addressed by the TOE. See FPT_SEP_HW_EXP (levied on the IT Environment) for the remaining functionality.

FPT_SEP_SW_EXP.1.1 The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects in the TSC.

FPT_SEP_SW_EXP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.2 Security Functional Requirements for the IT Environment

This section describes the Security Functional Requirements for the IT Environment. The Security Functional Requirements identified in the following table and are described in more detail in the following subsections.

Table 15 - Security Functional Requirements of the IT Environment

Security Functional Requirement (IT Environment)	Security Functional Requirement Name
FDP_ACC.1	Subset Access Control
FDP_ACF.1	Security Attribute Based Access Control
FIA_ATD.1	User Attribute Definition
FIA_UAU.2	User Authentication Before any Action
FIA_UID.2	User Identification Before any Action
FMT_MSA.1	Management of Security Attributes
FMT_MSA.3	Static Attribute Initialisation
FMT_MTD.1	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security Roles
FPT_RVM_HW_EXP.1	Non-Bypassability of the TSP
FPT_SEP_HW_EXP.1	Domain Separation environment
FPT_STM.1	Reliable Time Stamps

5.2.1 User Data Protection (FDP)

5.2.1.1 FDP_ACC.1(2) Subset Access Control

FDP_ACC.1.1(2) The **IT Environment** shall enforce the *IT Environment UAC SFP* on *Server Administrators (subjects), files containing the UNIX username/UNIX User UID/Primary UNIX User GID/Secondary UNIX User GID/ Windows User SID/Windows User GID/Windows Username (objects), read and write access (operations)*.

5.2.1.2 FDP_ACF.1(2) Security attribute based access control

FDP_ACF.1.1(2) The **IT Environment** shall enforce the *IT Environment UAC SFP* to objects based on the following: *only Server Administrators can have read and write access to files containing the UNIX username/UNIX User UID/Primary UNIX User GID/Secondary UNIX User GID/ Windows User SID/Windows User GID/Windows Username*.

FDP_ACF.1.2(2) The **IT Environment** shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *only Server Administrators can have read and write access to files containing the UNIX username/UNIX User UID/Primary UNIX User GID/Secondary UNIX User GID/ Windows User SID/Windows User GID/Windows Username*.

FDP_ACF.1.3(2) The **IT Environment** shall explicitly authorise access of subjects to objects based on the following additional rules: *no additional rules*.

FDP_ACF.1.4(2) The **IT Environment** shall explicitly deny access of subjects to objects based on the *no additional rules*.

5.2.2 Identification and Authentication (FIA)

5.2.2.1 FIA_ATD.1(1) User Attribute Definition

FIA_ATD.1.1(3) The **IT Environment** shall maintain the following list of security attributes belonging to individual users: *UNIX User UID and Primary UNIX User GID.*

5.2.2.2 FIA_ATD.1(2) User Attribute Definition

FIA_ATD.1.1(4) The **IT Environment** shall maintain the following list of security attributes belonging to individual users: *Secondary UNIX User GIDs.*

5.2.2.3 FIA_ATD.1(3) User Attribute Definition

FIA_ATD.1.1(5) The **IT Environment** shall maintain the following list of security attributes belonging to individual users: *Windows User SID and Windows User GID.*

5.2.2.4 FIA_UAU.2(2) User Authentication Before any Action

FIA_UAU.2.1(2) The **IT Environment** shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note: This iteration addresses CIFS and NFS Client login. NetApp Administrator login via the local console is addressed by the TOE.

5.2.2.5 FIA_UID.2(2) User Identification Before any Action

FIA_UID.2.1(2) The **IT Environment** shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Application note: This iteration addresses CIFS and NFS Client login. NetApp Administrator login via the local console is addressed by the TOE.

5.2.3 Security Management (FMT)

5.2.3.1 FMT_MSA.1(3) Management of Security Attributes

FMT_MSA.1.1(3) The **IT Environment** shall enforce the *IT Environment UAC SFP* to restrict the ability to modify, delete, *add* the security attributes *UNIX User UID received in the NFS Request* to *a Server Administrator.*

5.2.3.2 FMT_MSA.1(4) Management of Security Attributes

FMT_MSA.1.1(4) The **IT Environment** shall enforce the *IT Environment UAC SFP* to restrict the ability to modify, delete, *add* the security attributes *UNIX User UID and Primary UNIX User GID* to *a Server Administrator*.

5.2.3.3 FMT_MSA.1(5) Management of Security Attributes

FMT_MSA.1.1(5) The **IT Environment** shall enforce the *IT Environment UAC SFP* to restrict the ability to modify, delete, *add* the security attributes *Secondary UNIX User GIDs* to *a Server Administrator*.

5.2.3.4 FMT_MSA.1(6) Management of Security Attributes

FMT_MSA.1.1(6) The **IT Environment** shall enforce the *IT Environment UAC SFP* to restrict the ability to modify, delete, *add* the security attributes *Windows User SID and Windows User GID maintained by the domain controller* to *a Server Administrator.*

5.2.3.5 FMT_MSA.3(2) Static attribute initialisation

FMT_MSA.3.1(2) The **IT Environment** shall enforce the *IT Environment UAC SFP* to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(2) The **IT Environment** shall allow the *Server administrator* to specify alternative initial values to override the default values when an object or information is created.

Application note: FMT_MSA.3(2) applies to the security attributes defined for the IT Environment by FMT_MSA.1.

5.2.3.6 FMT_MTD.1(6) Management of TSF Data

FMT_MTD.1.1(6) The **IT Environment** shall restrict the ability to modify, delete, *add* the *UNIX Username maintained by the IT Environment* to *a Server Administrator.*

5.2.3.7 FMT_MTD.1(7) Management of TSF Data

FMT_MTD.1.1(7) The **IT Environment** shall restrict the ability to modify, delete, *add* the *Windows Username maintained by a CIFS Client* to *a Server Administrator.*

5.2.3.8 FMT_MTD.1(8) Management of TSF Data

FMT_MTD.1.1(8) The **IT Environment** shall restrict the ability to modify, delete, *add* the *Windows Username maintained by the IT Environment's Domain Controller* to *a Server Administrator.*

5.2.3.9 FMT_SMF.1(2) Specification of Management Functions

FMT_SMF.1.1(2) The **IT Environment** shall be capable of performing the following security management functions:

1) *Enable a Server Administrator to manage UNIX User UIDs and UNIX User GIDs.*
2) *Enable a Server Administrator to manage Windows User SIDs and Windows User GIDs.*
3) *Enable a Server Administrator to manage UNIX Usernames.*
4) *Enable a Server Administrator to manage Windows Usernames.*

5.2.3.10 FMT_SMR.1(2) Security Roles

FMT_SMR.1.1(2) The **IT Environment** shall maintain the roles *Server Administrator*.

FMT_SMR.1.2(2) The **IT Environment** shall be able to associate users with roles.

5.2.4 Protection of the TSF (FPT)

5.2.4.1 FPT_RVM_HW_EXP.1 Non-Bypassability of the TSP for Hardware

Rationale for explicitly stated SFR: Software TOEs are unable to fully satisfy FPT_RVM by themselves. This explicitly stated SFR states the portion of FPT_RVM supplied by the hardware in support of the overall FPT_RVM functionality. See FPT_RVM_SW_EXP (levied on the TOE) for the remaining functionality.

FPT_RVM_HW_EXP.1.1 The security functions of the storage appliance hardware shall ensure that security policy enforcement functions are invoked and succeed before each function within the scope of control of the operating system and hardware is allowed to proceed.

5.2.4.2 FPT_SEP_HW_EXP.1 TSF Domain Separation for Hardware

Rationale for explicitly stated SFR: Software TOEs are unable to fully satisfy FPT_SEP by themselves. This explicitly stated SFR states the portion of FPT_SEP supplied by the hardware in support of the overall FPT_SEP functionality. See FPT_SEP_SW_EXP (levied on the TOE) for the remaining functionality.

FPT_SEP_HW_EXP.1.1 The security functions of the storage appliance hardware shall support a security domain for the TOE's execution that protects the TOE from interference and tampering by untrusted subjects outside the TSC.

FPT_SEP_HW_EXP.1.2 The security functions of the storage appliance hardware shall enforce separation between the security domains of the TOE and subjects outside the TSC.

5.2.4.3 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 The **IT Environment** shall be able to provide reliable time-stamps for its own use.

Application Note: In ONTAP, the current time-of-day is a global variable that represents the number of microseconds since January 1, 1970. To provide an accurate time-of-day, The 'timed' service can be enabled to synchronize with an NTP or rdate server. The timed service will determine how far off the system's time-of-day is from a server and tell the kernel to slowly adjust the time to match. In addition, after timed has synchronized the time, it's possible to calculate the amount of drift caused by the 1 msec timer.

5.3 TOE Security Assurance Requirements

The TOE meets the assurance requirements for EAL2 and is augmented by ALC_FLR.3 Systematic flaw remediation. These requirements are summarised in the following table.

Table 16 - Assurance Requirements

Assurance Class	Component ID	Component Description
Configuration Management	ACM_CAP.2	Configuration items
Delivery and Operation	ADO_DEL.1	Delivery Procedures
	ADO_IGS.1	Installation, Generation, and Start-up Procedures
Development	ADV_FSP.1	Informal functional specification
	ADV_HLD.1	Descriptive High-Level Design
	ADV_RCR.1	Informal Correspondence Demonstration
Guidance Documents	AGD_ADM.1	Administrator Guidance
	AGD_USR.1	User Guidance
Life Cycle Support	ALC_FLR.3	Systematic flaw remediation
Tests	ATE_COV.1	Evidence of Coverage
	ATE_FUN.1	Functional Testing
	ATE_IND.2	Independent testing - sample

Assurance Class	Component ID	Component Description
Vulnerability Assessment	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer Vulnerability Analysis

5.4 TOE Strength of Function Claim

Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, August 1999, defines “Strength of Function (SOF)” in terms of the minimum efforts assumed necessary to defeat the expected security behavior of a TOE security function.

The only probabilistic or permutational mechanism in the TOE is the Identification and Authentication (I&A) security function, which uses a probabilistic or permutational mechanism when comparing passwords to authenticate TOE local users accessing the TOE via a serial connection.

The TOE minimum strength of function claim is SOF-basic. SOF-basic is defined in CC Part 1 section 2.3 as: "A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential." The rationale for the chosen level is based on the low attack potential of the threat agents identified in this ST and the strength of the minimum password length. The SOF-basic strength level is sufficient to meet the objectives of the TOE, given the security environment described in this ST.

CHAPTER 6

6. TOE Summary Specification

6.1 TOE Security Functions

This section identifies and describes the Security Functions implemented by the TOE.

6.1.1 Administrative (ADMIN) Security Function Summary

The CLI Administrative interface provides the necessary operator functions to allow a NetApp Administrator to manage and support the TSF.

The TOE maintains the following roles for users: NetApp Administrators and non-administrators. NetApp Administrators access the TOE via the local console and have access to the CLI. Non-administrators are users that do not have administrative access on the TOE.

The TOE enforces local human users (NetApp Administrators) to identify and authenticate themselves to the TOE before allowing any modifications to TOE managed TSF Data.

The TOE ensures that only NetApp Administrators can delete, modify, or add to the local files that contain identification and authentication data and files that contain security attributes necessary for enforcement of the DAC SFP.

The TOE ensures that all functions are invoked and succeed before the next function is invoked.

6.1.2 Discretionary Access Control (DAC) Security Function Summary

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 (TSF user data). The TOE maintains files with either NTFS-Style security attributes or UNIX-Style security attributes or both.

The access modes covered by the DAC SFP are: create, read, write, execute, change permission and change owner.

6.1.3 Self Protection Security Function

The TOE provides for self protection and non-bypassability of functions within the TOE's scope of control (TSC).

6.1.4 Audit

The TOE generates audit event records for security relevant events.

6.2 Discretionary Access Control (DAC) Security Function

The DAC SFP protects user data (FDP_ACC.1(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 Requirement that meets the Security Function.

6.2.1.1 DAC SFP Object Security Attributes

The TSF 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(1)). NTFS-Style files do not have symbolic links; therefore, the file type will be either directory or regular file (FDP_ACC.1(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. NTFS-Style files stored in mixed qtrees always take on NTFS-Style security attributes when updated by an NFS Client.

6.2.1.1.1 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 identifies 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(1)).

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 and any Secondary UNIX User GIDs 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(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.

6.2.1.1.2 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 it called a Security Descriptor (SD). This SD contains, the file owner's Security ID (SID) and an Access Control List (ACL). Each ACL consists of one or more Access Control Entries (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.

6.2.1.2 DAC SFP Access Requests

Access requests define what operation a subject (Server Administrators and non-administrators) 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(1)). The following sections define the operations.

6.2.1.2.1 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 17 - 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.

DAC SFP Operation	UNIX-Style File Types		
	Directory	Symbolic Link	Normal 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.
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.

6.2.1.2.2 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 18 - NTFS-Style File Access Modes

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

DAC SFP Operation	NTFS-Style File Types	
	Directory	Normal File
		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.

6.2.1.3 DAC Operations and Rules

In general the TOE supports access to all objects from subjects (Server Administrators and non-administrators). However, the following exceptions apply:

Client
The DAC SFP supports client cross-protocol support for create, read, write, execute, delete and change permission operations. The DAC SFP does not support cross-protocol support of the change owner command. Only NFS Clients can change the owner of UNIX-Style files. Only CIFS Clients can change the owner of NTFS-Style files.

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, 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.

6.2.1.4 DAC SFP Subject Security Attributes

The subjects that apply to the DAC SFP are Server Administrators and non-administrators; they access the TOE as NFS Clients and CIFS Clients (FDP_ACC.1(1)). To determine if access is permitted for an object, the TOE requires the security attributes associated with the client. These security attributes are provided by 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. However, the TOE also supports multi-protocol access: NFS Clients can be mapped to NTFS-Style security attributes and CIFS Clients can be mapped to UNIX-Style security attributes. The following sections describe the TOE's subject security attribute resolution used to enforce the DAC SFP.

6.2.1.4.1 Derivation of UNIX-Style Client Subject Security Attributes

If the TOE determines that UNIX-Style security attributes should be used to determine access for an object, the TOE requires a client's (subject's) UNIX User UID, primary UNIX User GID and any secondary UNIX User GIDs (FDP_ACF.1(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. The TOE then uses the UNIX username to obtain the primary and any secondary UNIX User GIDs (FDP_ACF.1(1)).

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 system and joined the Windows Domain (IT Environment). To get UNIX-Style security attributes for the CIFS Client, the TOE searches the /etc/usermap.cfg file to find a UNIX username for the Windows username. If a match is found, the UNIX username is used. The TOE may also use the UNIX User username specified in the wafl.default_unix_user.

For the remainder of this document, when the DAC SFP rules state that the TOE uses UNIX-Style security attributes for the subject, the TOE performs the steps described above to obtain the security attributes.

6.2.1.4.2 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.

If the access request is initiated by an NFS Client, the TOE received the client's UNIX User UID in the NFS request. To obtain the NTFS-Style subject security attributes for the NFS Client, the TOE uses the UNIX username to find the Windows username in the /etc/usermap.cfg file. If a match is not found, the TOE uses the Windows username specified in the wafl.default_nt_user file. Given the Windows username, the TOE finds the Windows User SID and Windows User GID by querying the IT Environment's Domain Controller.

For the remainder of this document, when the DAC SFP rules state that the TOE obtains NTFS-Style security attributes for the subject, the TOE performs the steps described above.

6.2.1.5 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 and is considered for cross-protocol access requests. The six 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(1)). In an NTFS qtree, the new file inherits the NTFS-Style security attributes from the parent directory (FMT_MSA.3(1)). In a mixed qtree, the security attribute style for the created file is determined by the access protocol of the Client.

If the parent directory is UNIX-Style, the TOE uses 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(1)). In a UNIX qtree, the new file inherits the UNIX-Style security attributes from the parent directory (FMT_MSA.3(1)). In a mixed qtree, the security attribute style for the created file is determined by the access protocol of the Client.

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 UNIX-style security attributes, the TOE uses 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(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 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(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.

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

The TOE, using 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(1)). Otherwise, access is denied.

UNIX-Style File stored in an NTFS-Style Parent Directory

The TOE, using NTFS-Style security attributes for both subject and object, 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(1)). Otherwise, access is denied.

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

The TOE, using 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(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(1)). Otherwise, access is denied.

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

The TOE, using 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(1)). If not, the TOE, using 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(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.

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

The TOE, using 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(1)). Otherwise, access is denied.

UNIX-Style File stored in an NTFS-Style Parent Directory

The TOE, using UNIX-Style security attributes for both subject and object, determines if the client has write access for the file. If so, the TOE, using NTFS-Style security attributes for both subject and object, 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(1)). Otherwise, access is denied.

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

The TOE, using 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(1)). Otherwise, access is denied.

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

The TOE, using 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, using 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 change permission access is permitted (FDP_ACF.1(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. The TOE does not support cross-protocol support of the Change Owner access request. Only NFS Clients can change ownership of UNIX-Style files; only CIFS Clients can change ownership of NTFS-Style files.

NFS Clients

If an NFS Client requests a Change Owner request (chown) for an NTFS-Style file, the request is denied (FDP_ACF.1(1)). If an NFS Client sends a Change Owner request (chown) for an NFS-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). If the client is root, access is allowed (FDP_ACF.1(1)) and the TOE changes the object's owner to the owner specified in the chown request. If the object had an ACL (mixed qtree), the TOE removes the ACL. If the owner is not root, 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(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(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.

6.3 Administrative Security Function

The Administrative Security Function 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 TSF Functions. The TSF Data includes both authentication data (used to authenticate NetApp administrators), security attribute data (used for DAC SFP enforcement) and other TSF data (used for DAC SFP subject security attribute resolution).

6.3.1 CLI

The CLI Administrative interface provides the necessary operator functions to allow a NetApp Administrator to manage and support the TSF (FMT_SMF.1).

6.3.2 Roles

The TOE maintains the following roles for users: NetApp Administrators, and non-administrators (FMT_SMR.1(1)).

A NetApp Administrator is any local human user who is assigned the permission to accesses the TOE via the serial port. This permission (login-console) is assigned to the "root" account by default. The non "Root" administrators receive the permission by being part of the "administrators" group. 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.

6.3.3 I&A

The Administrative Security Function's I&A functionality enforces local human users (NetApp Administrator role) to identify and authenticate themselves to the TOE before allowing any modifications to TOE managed TSF Data (FIA_UID.2(1), FIA_UAU.2(1)). NetApp Administrator's authentication data is maintained by the TOE in a local file. The file contains the username, password, username and the full name, password aging, and other similar characteristics for each NetApp administrator.

6.3.4 TSF Data Management

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:

1) TOE Username Management.
2) UNIX Username to Windows Username Mapping.
3) Default usernames for cross protocol access requests.
4) Deny unauthorized administrative login attempts via Data ONTAP.
5) Implement a "Sleep Mode" function call to Data ONTAP to deny access and initiate a time out period for further login attempts, for brute force password guessing.

<u>TOE Username Management</u>

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

<u>UNIX Usernames to Windows Username Mapping</u>

The TOE maintains a local /etc/usermap.cfg file that contains a mapping of UNIX Usernames to Windows Usernames. The file is used to support the TOE's DAC Security Functionality for client cross protocol access requests. The TOE uses this file for NFS Clients requesting access to a file that requires NTFS-Style security attributes. Likewise, the TOE uses this file to resolve CIFS Clients requesting access to a file that requires UNIX-Style security attributes Only NetApp Administrators may modify the /etc/usermap.cfg file (FMT_MTD.1(3)).

Default Users

The TOE maintains two files that contain TOE maintained TSF data that is used to resolve a client's username for cross protocol access. Wafl.default_unix_user contains the default UNIX username for CIFS Client. Wafl.default_nt_user contains the default Windows username for NFS Client. Only NetApp Administrators may modify the two files (FMT_MTD.1(4) and FMT_MTD.1(5)).

6.4 Self Protection Security Function

The TOE provides for self protection and non-bypassability of functions within the TOE's scope of control (TSC). The TOE controls actions carried out by a user by controlling a user session and the actions carried out during a user session. By maintaining and controlling a user session a user has with the TOE, the TOE ensures that no security functions within the TSC are bypassed and that there is a separate domain for the TOE that prevents the TOE from being interfered with or tampered with for those users that are within the TSC.

6.5 Audit Security Function

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. If the audit function is started or stopped, an audit event record is generated. All audit event records include a timestamp (obtained from the IT Environment).

The audit events related to administrator logons are:

1. Successful logon – the event includes the userid of the administrator.
2. Unknown user – the event includes the supplied userid. This event could indicate an invalid userid or an invalid password for a valid userid.

The audit event records related to configuration changes of the locally defined users and groups are:

1. User created – the event includes the userid created and the userid of the administrator performing the action
2. User deleted - the event includes the userid deleted and the userid of the administrator performing the action.
3. Group created – the event includes the group created and the userid of the administrator performing the action
4. Group deleted - the event includes the group deleted and the userid of the administrator performing the action.
5. Group member added – the event includes the userid and group of the association being created, and the userid of the administrator performing the action.
6. Group member removed – the event includes the userid and group of the association being removed, and the userid of the administrator performing the action.

6.6 Security Function Strength of Function Claim

The I&A functionality of the Administrative Security Function uses a probabilistic or permutational mechanism when comparing passwords for authentication. This mechanism is SOF-basic.

CHAPTER 7

7. Protection Profile Claims

This chapter provides detailed information in reference to the Protection Profile conformance identification that appears in Chapter 1, Section 1.4 Protection Profile Conformance.

7.1 Protection Profile Reference

This Security Target does not claim conformance to any registered Protection Profile.

7.2 Protection Profile Refinements

This Security Target does not claim conformance to any registered Protection Profile.

7.3 Protection Profile Additions

This Security Target does not claim conformance to any registered Protection Profile.

7.4 Protection Profile Rationale

This Security Target does not claim conformance to any registered Protection Profile.

CHAPTER 8

8. Rationale

Section 8.1 provides the rationale of objectives to threats and assumptions.

Section 8.2 provides the rationale of Security Functional Requirements to objectives.

Section 8.3 provides the rationale of the Security Functions to Security Functional Requirements.

Section 8.4 provides the rationale Security Functional Requirements proving hierarchy and dependencies.

Section 8.5 provides PP rationale.

Section 8.6 provides Assurance Measures Rationale for TOE Assurance Requirements

8.1 Security Objectives Rationale

This section demonstrates that the identified security objectives are covering all aspects of the security needs. This includes showing that each threat and assumption is addressed by a security objective. Table 19 demonstrates the correspondence between the security objectives identified in Chapter 4 to the assumptions and threats identified in Chapter 3. Table 20 provides the rationale proving that each threat and assumption is addressed.

Table 19 - Threats and Assumptions to Security Objectives Mapping

Threat/Assumption	TOE, IT Environment and Non-IT Environment Objectives															
	O.ADMIN_ROLES	O.AUDIT	O.DAC_ACC	O.ENFORCE	O.I&A	O.MANAGE	O.E.ACCESS	O.E.ADMIN_ROLES	O.E.ENFORCE	O.E.I&A	O.E.SUBJECTDATA	O.E.TIME	O.N.CREDEN	O.N.INSTALL	O.N.PHYSICAL	O.N.TRAINED
T.CONFIG_CORRUPT	X				X	X	X	X								
T.UNAUTH_ACCESS	X			X	X	X	X	X	X							
A.COOP													X			
A.MANAGE														X		X
A.NO_EVIL_ADM														X		
A.PEER											X					
A.PROTECT															X	
P.ADMIN_ACCESS	X				X	X										
P.AUDIT		X										X				
P.USER_ACCESS			X							X	X					

Table 20 - Threats and Assumptions to Security Objectives Rationale

Threat/Assumption	Security Objective Rationale (TOE, IT Environment and Non-IT Environment)
T.CONFIG_CORRUPT	O.MANAGE – The TOE will have defined methods and permissions for modification of configuration data. O.I&A - The TOE will monitor attempts to access configuration data or other trusted data that could result in system failure resulting in unauthorized access to trusted data. Users are required to identify and authenticate themselves to the TOE before attempting to modify TSF data or administrative functions. O.ADMIN_ROLES – The TOE will monitor attempts to access configuration data or other trusted data that could result in system failure resulting in unauthorized access to trusted data. Authorized roles are required for users to perform administrative procedures, thus isolating the amount of damage a user can perform. O.E.ACCESS – The IT Environment will monitor attempts to access configuration data or other trusted data that could result in system failure resulting in unauthorized access to trusted data. O.E.ADMIN_ROLES – The IT Environment will monitor attempts to access configuration data or other trusted data that could result in system failure resulting in unauthorized access to trusted data. Authorized roles are required for users to perform administrative procedures, thus isolating the amount of damage a user can perform.
T.UNAUTH_ACCESS	O.MANAGE – The TOE will have defined methods and permissions for modification of configuration data. O.ENFORCE – this objective addresses this threat by ensuring the security policy enforcement of the TOE is invoked and not interfered with inside the TOE. O.I&A - this objective builds on the O.MANAGE objective by requiring users to identify and authenticate themselves to the TOE before attempting to modify TSF data or security attributes via a serial connection. O.ADMIN_ROLES – this objective supports the O.MANAGE by requiring authorized roles for users to perform administrative procedures therefore, isolating the amount of damage a user can perform. O.E.ACCESS – this objective builds on O.MANAGE objective by providing an IT Environment enforced UAC SFP that defines and enforces restrictive access and modification rules for security attributes and TSF Data managed by the IT Environment and used by the TOE to enforce the DAC SFP. O.E.ENFORCE – this objective builds on O.ENFORCE by ensuring the security policy enforcement of the TOE is invoked and not interfered with outside the TOE. O.E.ADMIN_ROLES – this objective supports the O.E.ACCESS objective by requiring authorized roles for users to perform administrative procedures thus, isolating the amount of damage a user can perform.
A.COOP	O.N.CREDEN - this objective provides for the physical protection of the TOE's access credentials.

Threat/Assumption	Security Objective Rationale (TOE, IT Environment and Non-IT Environment)
A.MANAGE	O.N.INSTALL - this objective ensures that the TOE will be managed appropriately. O.N.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.
A.NO_EVIL_ADM	O.N.INSTALL - this objective ensures that the TOE will be managed appropriately.
A.PEER	O.E.SUBJECTDATA – The security attributes provided by the IT Environment will be meaningful because the representations between the TOE and IT Environment systems are consistent.
A.PROTECT	O.N.PHYSICAL – this objective provides for the physical protection of the TOE.
P.ADMIN_ACCESS	O.ADMIN_ROLES – this objective addresses this policy by requiring a distinct role for administrators, thereby enabling administrative functions to be associated with just that role. O.I&A – this objective addresses this policy by requiring a mechanism for the TOE to I&A specific users, which is a prerequisite for associating a role with a user. O.MANAGE – this objective addresses this policy by requiring administrative functions that can be restricted to administrative roles.
P.AUDIT	O.AUDIT – this objective addresses this policy by requiring the TOE to audit events related to administrator login and configuration changes. O.E.TIME – this objective supports the policy by requiring the IT Environment to supply a reliable timestamp for use in the audit event records.
P.USER_ACCESS	O.DAC_ACC – this objective addresses this policy by defining a DAC SFP that defines the access control rules for user data managed by the TOE based on subjects, objects, subject security attributes, object security attributes and operations. O.E.I&A – this objective addresses this policy by requiring a mechanism for the IT Environment to I&A users on the client systems, which is a prerequisite for associating security attributes with a user. O.E.SUBJECTDATA – this objective supports the O.DAC_ACC objective by requiring the IT Environment to provide the security attributes and TSF data managed by the IT Environment and used by the TOE to enforce the DAC SFP to the TOE.

8.2 Security Functional Requirements Rationale

8.2.1 Rationale for Security Functional Requirements of the TOE Objectives

This section provides the rationale for the Security Functional Requirements demonstrating that the Security Functional Requirements are suitable to address the security objectives. Table 21 identifies for each Security Functional Requirement identified in Section 5.1, and the TOE security objective(s) identified in Section 4.1 that address it. The following table provides the rationale proving that each security objective is addressed by a Security Functional Requirement.

Table 21 - TOE SFRs to TOE Security Objectives Mapping

Security Functional Requirement (TOE)	TOE Objective					
	O.ADMIN_ROLES	O.AUDIT	O.DAC_ACC	O.ENFORCE	O.I&A	O.MANAGE
FAU_GEN.1		X				
FAU_GEN.2		X				
FDP_ACC.1(1)			X			
FDP_ACF.1(1)			X			
FIA_UAU.2(1)					X	
FIA_UID.2(1)					X	
FMT_MSA.1(1)						X
FMT_MSA.1(2)						X
FMT_MSA.3(1)			X			
FMT_MTD.1(1)						X
FMT_MTD.1(2)						X
FMT_MTD.1(3)						X
FMT_MTD.1(4)						X
FMT_MTD.1(5)						X
FMT_SMF.1						X
FMT_SMR.1(1)	X					
FPT_RVM_SW_EXP.1				X		
FPT_SEP_SW_EXP.1				X		

Table 22 - TOE SFRs to TOE Security Objectives Rationale

Objective (TOE)	TOE Security Objectives Rationale	
	Note	Rationale
O.ADMIN_ROLES		FMT_SMR.1(1) – Defines the user roles implemented by the DAC SFP requiring authorized roles for NetApp Administrators to perform administrative procedures.
O.AUDIT		FAU_GEN.1 – Requires the TOE to generate audit event records for administrator logons and configuration changes, and defines the information saved in these records. FAU_GEN.2 – Requires the TOE to associate a specific user with the audit event records.
O.DAC_ACC	DAC Subjects	FDP_ACC.1(1) – Identifies the subjects covered by the DAC SFP. FDP_ACF.1(1) – Identifies the subject security attributes used to enforce the DAC SFP. .
	DAC Objects	FDP_ACC.1(1) – Identifies the objects covered by the DAC SFP. FDP_ACF.1(1) – Identifies the object security attributes used to enforce the DAC SFP. FMT_MSA.3(1) – Ensures restrictive default values are defined for the TOE's object security attributes used to enforce the DAC SFP.
	DAC Operations	FDP_ACC.1(1) – Identifies the operations (access requests) of subjects on objects covered by the DAC SFP.
	DAC Rules	FDP_ACF.1(1) – Defines the DAC rules enforced by the TOE that define access rules for TOE managed user data.
O.ENFORCE		FPT_RVM_SW_EXP.1 - The TOE ensures that all functions within the TSC are invoked and succeed before the next function may proceed. Without this assurance, there would not be assurance the TSF could not be bypassed within the TOE. FPT_SEP_SW_EXP.1 – The TOE tracks user sessions individually and enforces the TSP appropriately for each session. User sessions can not interfere with one another within the TOE. Without this assurance, there would not be assurance that the TOE could not be interfered with.
O.I&A		FIA_UID.2(1) – Ensures that users must identify themselves before any TSF mediated access to the TOE functions or TSF data is allowed. FIA_UAU.2(1) – Ensures that users must authenticate themselves before any TSF mediated access to the TOE functions or TSF data is allowed.

Objective (TOE)	TOE Security Objectives Rationale	
	Note	Rationale
O.MANAGE		FMT_SMF.1 – Defines the TSF management functions provided by the TOE that ensures the TOE's SFPs can be enforced. FMT_MSA.1(1) Only authorized NetApp Administrators responsible for the management of TOE security may modify, delete or add the TOE security attributes maintained locally by the TOE and used to enforce the DAC SFP. FMT_MSA.1(2) Only authorized NetApp Administrators responsible for the management of TOE security may modify, delete or add the maintained locally by the TOE and used to enforce the DAC SFP. FMT_MTD.1(1) – Defines the restrictions enforced by the DAC SFP to modify TOE usernames managed by the TOE and used to enforce the DAC SFP and I&A. FMT_MTD.1(2) – Defines the restrictions enforced by the DAC SFP to modify the user account passwords managed by the TOE and used to enforce the I&A. FMT_MTD.1(3) – Defines the restrictions enforced by the DAC SFP to modify UNIX username and Windows username mappings (/etc/usermap.cfg) managed by the TOE and used to enforce the DAC SFP FMT_MTD.1(4) – Defines the restrictions enforced by the DAC SFP to modify the default UNIX username (wafl.default_unix_user) managed by the TOE and used to enforce the DAC SFP. FMT_MTD.1(5) – Defines the restrictions enforced by the DAC SFP to modify the default Windows username (wafl.default_nt_user) managed by the TOE and used to enforce the DAC SFP

8.2.2 Rationale for Security Functional Requirements of the IT Environment

This section provides the rationale for the IT Environment's Security Functional Requirements demonstrating that the IT Environment's Security Functional Requirements are suitable to address the IT Environment's security objectives. The following table identifies for each IT Environment Security Functional Requirement identified in Section 5.2, the IT Environment's security objective(s) identified in Section 4.2 that address it. Table 23 provides the rationale proving that each IT Environment security objective is addressed by an IT Environment Security Functional Requirement.

Table 23 - IT Environment Security Functional Requirements to IT Environment Objectives Mapping

Security Functional Requirements (IT Environment)	IT Environment Objective					
	O.E.ACCESS	O.E.ADMIN_ROLES	O.E.ENFORCE	O.E.I&A	O.E.SUBJECTDATA	O.E.TIME
FIA_ATD.1(1)					X	
FIA_ATD.1(2)					X	
FIA_ATD.1(3)					X	
FIA_UAU.2(2)				X		
FIA_UID.2(2)				X		
FDP_ACC.1(2)	X					
FDP_ACF.1(2)	X					
FMT_MSA.1(3)	X					
FMT_MSA.1(4)	X					
FMT_MSA.1(5)	X					
FMT_MSA.1(6)	X					
FMT_MSA.3(2)	X					
FMT_MTD.1(6)	X					
FMT_MTD.1(7)	X					
FMT_MTD.1(8)	X					
FMT_SMF.1(2)		X				
FMT_SMR.1(2)		X				
FPT_RVM_HW_EXP.1			X			
FPT_SEP_HW_EXP.1			X			
FPT_STM.1						X

Table 24 - IT Environment Security Functional Requirements to IT Environment Objectives Rationale

Objective (IT Environment)	IT Environment Security Objectives Rationale
O.E.ACCESS	FDP_ACC.1(2) and FDP_ACF.1(2) – Defines the IT Environment UAC SFP to access TSF data managed by the IT Environment FMT_MSA.1(3) – Defines the restrictions enforced by the IT Environment's UAC SFP to modify UNIX User UIDs received in an NFS request and used by the TOE to enforce the DAC SFP. FMT_MSA.1(4) – Defines the restrictions enforced by the IT Environment's UAC SFP to modify UNIX User UIDs and Primary UNIX User GIDs maintained by the IT Environment and used by the TOE to enforce the DAC SFP. FMT_MSA.1(5) – Defines the restrictions enforced by the IT Environment's UAC SFP to modify Secondary UNIX User GIDs maintained by the IT Environment and used by the TOE to enforce the DAC SFP. FMT_MSA.1(6) – Defines the restrictions enforced by the IT Environment's UAC SFP to modify Windows User SIDs and Windows User GIDs maintained by the IT Environment's Domain Controller and used by the TOE to enforce the DAC SFP. FMT_MSA.3(2) – Defines the restrictions that the Server Administrator is only able to set default values or modify the default values for the files containing the TSF data. FMT_MTD.1(6) – Defines the restrictions enforced by the IT Environment's UAC SFP to modify UNIX usernames managed by the IT Environment and used to enforce the DAC SFP. FMT_MTD.1(7) – Defines the restrictions enforced by the IT Environment's UAC SFP to modify Windows usernames managed by the IT Environment's CIFS Server and used to enforce the DAC SFP. FMT_MTD.1(8) – Defines the restrictions enforced by the IT Environment's UAC SFP to modify Windows usernames managed by the IT Environment's Domain Controller and used to enforce the DAC SFP.
O.E.ADMIN_ROLES	FMT_SMF.1(2) – Defines the management functions available to the roles defined by FMT_SMT.1(2). FMT_SMR.1(2) – Defines the user roles implemented by the IT Environment's UAC SFP requiring authorized roles for Server Administrators to perform administrative procedures.

Objective (IT Environment)	IT Environment Security Objectives Rationale
O.E.ENFORCE	FPT_RVM_HW_EXP.1 – Ensures that the TOE cannot be bypassed, therefore allowing the TOE to perform its policy enforcement. FPT_SEP_HW_EXP.1 – Ensures that the TOE has its own domain of execution to prevent interference from outside the TOE, therefore allowing the TOE to perform its policy enforcement securely.
O.E.I&A	FIA_UID.2(2) – Ensures that users must identify themselves to the IT Environment before allowing any TSF mediated access to the TOE functions or TSF data. FIA_UAU.2(2) - Ensures that users must authenticate themselves to the IT Environment before allowing any TSF mediated access to the TOE functions or TSF data.
O.E.SUBJECTDATA	FIA_ATD.1(1) - Identifies the subject security attributes (UNIX User UID and Primary UNIX User GID) maintained by the IT environment and used by the TOE to enforce the DAC SFP. FIA_ATD.1(2) - Identifies the subject security attributes (Secondary UNIX User GIDs) maintained by the IT environment and used by the TOE to enforce the DAC SFP. FIA_ATD.1(3) - Identifies the subject security attributes (Windows User SID and Windows User GID) maintained by the IT environment and used by the TOE to enforce the DAC SFP.
O.E.TIME	FPT_STM.1 – Requires the IT Environment to supply a reliable time stamp for use by the TOE in audit records.

8.3 TOE Summary Specification Rationale

This section demonstrates that the TOE’s Security Functions identified in Section 6.1 completely and accurately meet the TOE’s Security Functional Requirements identified in Section 5.1. The following table demonstrates the correspondence between the Security Functions and the TOE Security Functional Requirements. The subsequent sections describe the rationale proving that the Security Functions provide the functionality of the Security Functional Requirements.

Table 25 - Security Functional Requirements to Security Functions Mapping

Security Functional Requirements (TOE)	Security Functions			
	ADMIN	Audit	DAC	Self Protection
FAU_GEN.1		X		
FAU_GEN.2		X		
FDP_ACC.1(1)			X	
FDP_ACF.1(1)			X	

Security Functional Requirements (TOE)	Security Functions			
	ADMIN	Audit	DAC	Self Protection
FIA_UAU.2(1)	X			
FIA_UID.2(1)	X			
FMT_MSA.1(1)	X			
FMT_MSA.1(2)	X			
FMT_MSA.3(1)			X	
FMT_MTD.1(1)	X			
FMT_MTD.1(2)	X			
FMT_MTD.1(3)	X			
FMT_MTD.1(4)	X			
FMT_MTD.1(5)	X			
FMT_SMF.1(1)	X			
FMT_SMR.1(1)	X			
FPT_RVM_SW_EXP.1				X
FPT_SEP_SW_EXP.1				X

Table 26 - Security Functional Requirements to Security Functions Rationale

Security Functional Requirements (TOE)	Security Function Rationale
FAU_GEN.1	Audit – The TOE generates audit event records when the audit function is started or stopped, for administrator logons, and for configuration changes to the locally defined users and groups.
FAU_GEN.2	Audit – The audit event records generated by the TOE include the userid associated with the event.
FDP_ACC.1(1)	DAC - The TOE restricts access to files based on the Client type and operation requested.
FDP_ACF.1(1)	DAC – The TOE defines a defined set of rules for access of files from Clients based on client security attributes, file (object) security attributes and the operation requested.
FIA_UAU.2(1)	ADMIN – The TOE requires NetApp Administrators to authenticate themselves before allowing any access to any TSF mediated actions.
FIA_UID.2(1)	ADMIN - The TOE requires NetApp Administrators to identify themselves before allowing any access to

Security Functional Requirements (TOE)	Security Function Rationale
	any TSF mediated actions.
FMT_MSA.1(1)	ADMIN – The TOE provides support to create and modify UNIX User UIDs and Primary UNIX User GIDs used to support the DAC SFP and restricts creation and modification of these security attributes to NetApp Administrators.
FMT_MSA.1(2)	ADMIN – The TOE provides support to create and modify Secondary UNIX User GIDS used to support the DAC SFP and restricts creation and modification of Secondary UNIX User GIDs to NetApp Administrators.
FMT_MSA.3(1)	DAC – The TOE does provides for strict default values for security attributes used to enforce the DAC SFP and does not support a mechanism to modify the values defined as default values.
FMT_MTD.1(1)	ADMIN – The TOE provides administrative support that enables NetApp Administrators to create and modify TOE usernames used to support the DAC SFP. The TOE restricts access to TOE usernames managed by the TOE to users with a NetApp Administrator role.
FMT_MTD.1(2)	ADMIN - The TOE provides administrative support that enables administrators to create and modify TOE user passwords used to support the DAC SFP. The TOE restricts access to TOE passwords managed by the TOE to users with a NetApp Administrator role.
FMT_MTD.1(3)	ADMIN - The TOE provides administrative support that enables administrators to create and modify Windows Username and UNIX Username mappings used to support the DAC SFP. The TOE restricts access to Windows Username/UNIX Usernames managed by the TOE to users with a NetApp Administrator role.
FMT_MTD.1(4)	ADMIN - The TOE provides administration functionality that enables a user to modify the UNIX Username stored in wafl.default_unix_user value. Access to wafl.default_unix_user value is limited by the TOE to NetApp Administrators.
FMT_MTD.1(5)	ADMIN – The TOE provides administration functionality that enables a user to modify the Windows Username stored in wafl.default_nt_user value. Access to wafl.default_nt_user value is limited by the TOE to NetApp Administrators.
FMT_SMF.1(1)	ADMIN – The TOE provides a CLI management interface that enables NetApp Administrators to modify TSF Data used to enforce the TOE's SFPs.

Security Functional Requirements (TOE)	Security Function Rationale
	Access to the CLI requires NetApp Administrators to identify and authenticate themselves to the TOE.
FMT_SMR.1(1)	ADMIN – The TOE supports the following user roles: NetApp Administrator and non-administrator. The TOE uses these roles to restrict access to TSF Data and invocation of functions that affect the TSF security behaviour.
FPT_RVM_SW_EXP.1	Self Protection - Security functions of the TSF may not be bypassed by activities within the TSC. Interfaces to the TSF ensure that security policies are enforced. TOE interfaces that do not invoke the TSF can not be used to bypass the TSF.
FPT_SEP_SW_EXP.1	Self Protection - Untrusted subjects within the TSC have strictly limited functionality that prevents interference or tampering with the TSF.

8.4 CC Component Hierarchies and Dependencies

This section of the ST demonstrates that the ST identified TOE Security Functional Requirement Components include the appropriate dependencies.

Components are hierarchical to and dependent upon any necessary rationale. N/A means the Security Functional Requirements Component has no dependencies and therefore, no dependency rationale is required. The term “Satisfied” means that the Security Functional Requirements dependency was included in the ST.

8.4.1 TOE Security Functional Component Hierarchies and Dependencies

This section of the ST demonstrates that the identified TOE and IT Security Functional Requirements include the appropriate hierarchical SFRs and dependent SFRs. The following tables list the SFRs for the TOE and IT Environment, the SFRs that are hierarchical to and dependent upon, and any necessary rationale.

N/A in the Rationale column means the Security Functional Requirement has no dependencies and therefore, no dependency rationale is required. Satisfied in the Rationale column means the Security Functional Requirements dependency was included in the ST.

Table 27 - TOE Security Functional Requirements Dependency Rationale

Security Functional Requirement (TOE)	Hierarchical To	Dependency	Rationale
FAU_GEN.1	No Components	FPT_STM.1	FPT_STM.1 is satisfied in the IT Environment

Security Functional Requirement (TOE)	Hierarchical To	Dependency	Rationale
FAU_GEN.2	No Components	FAU_GEN.1, FIA_UID.1	Satisfied FIA_UID.2 is hierarchical to FIA_UID.1. FIA_UID.2 is included in the TOE; therefore, this dependency is satisfied.
FDP_ACC.1	No Components	FDP_ACF.1	Satisfied
FDP_ACF.1	No Components	FDP_ACC.1, FMT_MSA.3	Satisfied Satisfied
FIA_UAU.2	FIA_UAU.1	FIA_UID.1	FIA_UID.2 is hierarchical to FIA_UID.1. FIA_UID.2 is included in the TOE; therefore, this dependency is satisfied.
FIA_UID.2	FIA_UID.1	No dependencies	N/A
FMT_MSA.1	No Components	[FDP_ACC.1 or FDP_IFC.1], FMT_SMF.1, FMT_SMR.1	Satisfied N/A Satisfied Satisfied
FMT_MSA.3	No Components	FMT_MSA.1, FMT_SMR.1	Satisfied Satisfied
FMT_MTD.1	No components.	FMT_SMF.1, FMT_SMR.1	Satisfied Satisfied
FMT_SMF.1	No Components	No dependencies	N/A
FMT_SMR.1	No Components	FIA_UID.1	FIA_UID.2 is hierarchical to FIA_UID.1. FIA_UID.2 is included in the TOE; therefore, this dependency is satisfied.
FPT_RVM_SW_EXP.1	No Components	No dependencies	N/A
FPT_SEP_SW_EXP.1	No Components	No dependencies	N/A

Table 28 - IT Environment SFRs Dependency Rationale

Security Functional Requirement (IT Environment)	Hierarchical To	Dependency	Rationale
FIA_ATD.1	No components.	No dependencies	N/A
FIA_UAU.2	FIA_UAU.1	FIA_UID.1	FIA_UID.2 is hierarchical to FIA_UID.1. FIA_UID.2 is included in the IT Environment; therefore, this dependency is satisfied.
FIA_UID.2	FIA_UID.1	No dependencies	N/A
FDP_ACC.1	No components	FDP_ACF.1	Satisfied
FDP_ACF.1	No components	FDP_ACC.1, FMT_MSA.3	Satisfied Satisfied
FMT_MSA.1	No components.	[FDP_ACC.1 or FDP_IFC.1], FMT_SMF.1, FMT_SMR.1	Satisfied N/A Satisfied Satisfied
FMT_MSA.3	No components.	FMT_MSA.1, FMT_SMR.1	Satisfied Satisfied
FMT_MTD.1	No components.	FMT_SMF.1, FMT_SMR.1	Satisfied Satisfied
FMT_SMF.1	No Components	No dependencies	N/A
FMT_SMR.1	No components.	FIA_UID.1	FIA_UID.2 is hierarchical to FIA_UID.1. FIA_UID.2 is included in the IT Environment; therefore, this dependency is satisfied.
FPT_RVM_HW_EXP.1	No Components	No dependencies	N/A
FPT_SEP_HW_EXP.1	No Components	No dependencies	N/A

Security Functional Requirement (IT Environment)	Hierarchical To	Dependency	Rationale
FPT_STM.1	No Components	No dependencies	N/A

8.5 PP Claims Rationale

The rationale for the Protection Profile conformance claims is defined in Chapter 7, Section 7.4 Protection Profile Rationale.

8.6 Assurance Measures Rationale for TOE Assurance Requirements

The TOE stresses assurance through vendor actions that are within the bounds of current best commercial practice. The TOE provides, primarily via review of vendor-supplied evidence, independent confirmation that these actions have been competently performed.

The general level of assurance for the TOE is:

1. Consistent with current best commercial practice for IT development and provides a product that is competitive against non-evaluated products with respect to functionality, performance, cost, and time-to-market.
2. The TOE assurance also meets current constraints on widespread acceptance, by expressing its claims against EAL2 from part 3 of the Common Criteria.
3. The TOE assurance is augmented by ALC_FLR.3 Systematic flaw remediation to reflect the vendor mechanisms in place to collect, address and track security flaws in the TOE.

The following table provides a reference between each TOE assurance requirement and the related vendor documentation that satisfies each requirement.

Table 29 - Assurance Measures

Assurance Class	Component ID	Documentation Satisfying Component
Configuration Management	ACM_CAP.2	SCM process for OnTAP Ontap7251filelist.txt
Delivery and Operation	ADO_DEL.1	Delivery and Operations
	ADO_IGS.1	Installation, Generation and Start Up Procedures
Development	ADV_FSP.1	Data ONTAP 7.2.5.1 Functional Specification
	ADV_HLD.1	Data ONTAP 7.2.5.1 High Level Design
	ADV_RCR.1	Data ONTAP 7.2.5.1 Correspondence Mapping

Assurance Class	Component ID	Documentation Satisfying Component
Guidance Documents	AGD_ADM.1	Administrator and User guidance for Data ONTAP Common Criteria deployments for Data ONTAP 7.2.5.1 Net App Man Pages
	AGD_USR.1	Administrator and User guidance for Data ONTAP Common Criteria deployments for Data ONTAP 7.2.5.1 Net App Man Pages
Lice cycle support	ALC_FLR.3	Data ONTAP Security Flaw Remediation Documentation for Common Criteria Evaluation
Tests	ATE_COV.1	Data ONTAP 7.2.5.1 Test Documentation for Common Criteria EAL2 Evaluation
	ATE_FUN.1	Data ONTAP 7.2.5.1 Test Documentation for Common Criteria EAL2 Evaluation
	ATE_IND.2	Data ONTAP 7.2.5.1 Test Documentation for Common Criteria EAL2 Evaluation
Vulnerability Assessment	AVA_SOF.1	Strength of Function Analysis for Common Criteria EAL2, Data ONTAP 7.2.5.1
	AVA_VLA.1	Developer Vulnerability Analysis Data ONTAP 7.2.5.1 Common Criteria