

**Network Appliance Data ONTAP Versions 7.0.3 and 7.0.4 Security Target**



Network Appliance, Inc.  
8405 Greensboro Drive Suite 1000  
McLean, VA 22102

Phone: 703-918-7200  
Fax: 703-918-7301



COACT, Inc.  
Rivers Ninety Five  
9140 Guilford Road, Suite N  
Columbia, MD 21046-2587

Phone: 301-498-0150  
Fax: 301-498-0855

March 28, 2007

COACT, Inc. assumes no liability for any errors or omissions that may appear in this document.

## DOCUMENT INTRODUCTION

Prepared By:

Prepared For:

COACT, Inc.

Network Appliance, Inc.

9140 Guilford Road, Suite N

8405 Greensboro Dr. Suite 1000

Columbia, Maryland 21046-2587

McLean, VA 22102

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Data ONTAP Versions 7.0.3 and 7.0.4. 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

<u>Rev</u>	<u>Description</u>
1-0	Initial ST

## TABLE OF CONTENTS

<b>LIST OF FIGURES .....</b>	<b>xi</b>
<b>LIST OF TABLES .....</b>	<b>xiii</b>
<b>LIST OF ACRONYMS .....</b>	<b>xv</b>
<b>1. SECURITY TARGET INTRODUCTION.....</b>	<b>1</b>
1.1 ST and TOE Identification.....	1
1.2 TOE Overview .....	1
1.2.1 Security Target Organization.....	1
<b>2. TOE DESCRIPTION .....</b>	<b>3</b>
2.1 TOE Overview .....	3
2.2 TOE Functionality Included in the Logical Boundary .....	3
2.3 Product Overview .....	3
2.4 TOE Component Systems.....	4
2.4.1 WAFL Functionality Detail.....	5
2.4.1.1 Files.....	6
2.4.1.2 Clients .....	7
2.5 TOE and TOE Relied Upon IT Environment.....	7
2.5.1 TOE Relied Upon IT Environment Hardware.....	7
2.5.2 TOE Relied Upon IT Environment Software .....	8
2.6 Physical Boundary .....	8
2.6.1 TOE Data .....	9
2.6.1.1 User Data .....	9
2.6.1.2 TSF Data.....	9
2.6.1.3 Files.....	9
2.6.1.4 Security Attributes .....	9
2.7 Functionality Excluded from the TOE’s Logical Boundary.....	11
2.8 TSF Functional Summary.....	11
2.9 Rationale for Non-Bypassability and Separation for the TOE .....	11
2.10 TOE Evaluated Configuration .....	12
2.10.1 TOE Evaluated Configuration Systems.....	12
2.10.2 TOE Evaluated Configuration Options.....	12
2.10.2.1 Access Protocol Options.....	12
2.10.2.2 Name Service Options .....	12
2.10.2.3 Miscellaneous .....	12
<b>3. SECURITY ENVIRONMENT.....</b>	<b>14</b>
3.1 Assumptions.....	14
3.1.1 Connectivity Assumptions.....	14
3.1.2 Personnel Assumptions.....	14
3.1.3 Physical Assumptions .....	15
3.2 Threats.....	15
3.3 Organizational Security Policies.....	15
<b>4. SECURITY OBJECTIVES .....</b>	<b>16</b>
4.1 Security Objectives for the TOE.....	16

4.2 Security Objectives for the Environment.....	16
4.2.1 Security Objectives for the IT Environment.....	16
4.2.2 Security Objectives for the Non-IT Environment.....	17
<b>5. SECURITY REQUIREMENTS.....</b>	<b>18</b>
5.1 TOE Security Functional Requirements .....	18
5.1.1 User Data Protection (FDP).....	19
5.1.1.1 FDP_ACC.1(1) Subset Access Control.....	19
5.1.1.2 FDP_ACF.1(1) Security Attribute Based Access Control.....	20
5.1.2 Identification and Authentication (FIA) .....	30
5.1.2.1 FIA_ATD.1(1) User Attribute Definition.....	30
5.1.2.2 FIA_ATD.1(2) User Attribute Definition.....	30
5.1.2.3 FIA_UAU.2(1) User Authentication Before any Action.....	30
5.1.2.4 FIA_UID.2(1) User Identification Before any Action.....	31
5.1.2.5 FIA_USB.1 User-Subject Binding.....	31
5.1.3 Security Management (FMT) .....	31
5.1.3.1 FMT_MSA.1(1) Management of Security Attributes .....	31
5.1.3.2 FMT_MSA.1(2) Management of Security Attributes .....	31
5.1.3.3 FMT_MSA.3(1) Static Attribute Initialisation .....	31
5.1.3.4 FMT_MTD.1(1) Management of TSF Data .....	31
5.1.3.5 FMT_MTD.1(2) Management of TSF Data .....	31
5.1.3.6 FMT_MTD.1(3) Management of TSF Data .....	32
5.1.3.7 FMT_MTD.1(4) Management of TSF Data .....	32
5.1.3.8 FMT_MTD.1(5) Management of TSF Data .....	32
5.1.3.9 FMT_SMF.1 Specification of Management Functions .....	32
5.1.3.10 FMT_SMR.1(1) Security Roles.....	32
5.1.4 Protection of the TSF (FPT) .....	32
5.2 Security Functional Requirements for the IT Environment.....	33
5.2.1 User Data Protection (FDP).....	33
5.2.1.1 FDP_ACC.1(2) Subset Access Control .....	33
5.2.1.2 FDP_ACF.1(2) Security attribute based access control .....	33
5.2.2 Identification and Authentication (FIA) .....	34
5.2.2.1 FIA_ATD.1(3) User Attribute Definition.....	34
5.2.2.2 FIA_ATD.1(4) User Attribute Definition.....	34
5.2.2.3 FIA_ATD.1(5) User Attribute Definition.....	34
5.2.2.4 FIA_UAU.2(2) User Authentication Before any Action.....	34
5.2.2.5 FIA_UID.2(2) User Identification Before any Action.....	34
5.2.3 Security Management (FMT) .....	35
5.2.3.1 FMT_MSA.1(3) Management of Security Attributes .....	35
5.2.3.2 FMT_MSA.1(4) Management of Security Attributes .....	35
5.2.3.3 FMT_MSA.1(5) Management of Security Attributes .....	35
5.2.3.4 FMT_MSA.1(6) Management of Security Attributes .....	35
5.2.3.5 FMT_MSA.3(2) Static attribute initialisation.....	35
5.2.3.6 FMT_MTD.1(6) Management of TSF Data .....	35
5.2.3.7 FMT_MTD.1(7) Management of TSF Data .....	36
5.2.3.8 FMT_MTD.1(8) Management of TSF Data .....	36
5.2.3.9 FMT_SMR.1(2) Security Roles.....	36

5.2.4 Protection of the TSF (FPT) .....	36
5.2.4.1 FPT_RVM_HW_EXP.1 Non-Bypassability of the TSP for Os's .....	36
5.3 TOE Security Assurance Requirements.....	36
5.4 TOE Strength of Function Claim.....	37
<b>6. TOE SUMMARY SPECIFICATION.....</b>	<b>38</b>
6.1 TOE Security Functions.....	38
6.1.1 Administrative (ADMIN) Security Function Summary .....	38
6.1.2 Discretionary Access Control (DAC) Security Function Summary .....	38
6.1.3 Self Protection Security Function .....	38
6.2 Discretionary Access Control (DAC) Security Function.....	38
6.2.1.1 DAC SFP Object Security Attributes .....	38
6.2.1.2 DAC SFP Access Requests.....	40
6.2.1.3 DAC Operations and Rules.....	42
6.2.1.4 DAC SFP Subject Security Attributes .....	43
6.2.1.5 DAC SFP Rules .....	44
6.3 Administrative Security Function .....	47
6.3.1 CLI.....	47
6.3.2 Roles .....	47
6.3.3 I&A .....	48
6.3.4 TSF Data Management .....	48
6.4 Self Protection Security Function .....	49
6.5 Security Function Strength of Function Claim .....	49
<b>7. PROTECTION PROFILE CLAIMS.....</b>	<b>51</b>
7.1 Protection Profile Reference .....	51
7.2 Protection Profile Refinements .....	51
7.3 Protection Profile Additions .....	51
7.4 Protection Profile Rationale.....	51
<b>8. RATIONALE .....</b>	<b>53</b>
8.1 Security Objectives Rationale.....	53
8.2 Security Functional Requirements Rationale.....	58
8.2.1 Rationale for Security Functional Requirements of the TOE Objectives.....	58
8.2.2 Rationale for Security Functional Requirements of the IT Environment.....	61
8.3 TOE Summary Specification Rationale.....	63
8.4 CC Component Hierarchies and Dependencies.....	66
8.4.1 TOE Security Functional Component Hierarchies and Dependencies .....	66
8.5 PP Claims Rationale .....	69
8.6 Assurance Measures Rationale for TOE Assurance Requirements.....	69

## LIST OF FIGURES

Figure 1 - TOE Components.....	5
Figure 2 - WAFL Functionality Detail .....	6

## LIST OF TABLES

Table 1 -	Network Appliance Products included in the TOE's Evaluated Configuration 4	
Table 2 -	Connectivity Assumptions .....	14
Table 3 -	Personnel Assumptions .....	14
Table 4 -	Physical Assumptions .....	15
Table 5 -	Threats.....	15
Table 6 -	Security Objectives for the TOE.....	16
Table 7 -	Security Objectives for the IT Environment.....	17
Table 8 -	Security Objectives for the Non-IT Environment.....	17
Table 9 -	Security Functional Requirements of the TOE.....	18
Table 10 -	<i>FDP_ACC.1.1(1) Detail</i> .....	19
Table 11 -	<i>FDP_ACF.1.1(1) Detail</i> .....	20
Table 12 -	<i>FDP_ACF.1.2(1) Detail</i> .....	25
Table 13 -	<i>FDP_ACF.1.3(1) Detail</i> .....	29
Table 14 -	<i>FDP_ACF.1.4(1) Detail</i> .....	30
Table 15 -	Security Functional Requirements of the IT Environment .....	33
Table 16 -	Assurance Requirements.....	36
Table 17 -	UNIX-Style File Access Requests .....	40
Table 18 -	NTFS-Style File Access Modes.....	41
Table 19 -	Threats and Assumptions to Security Objectives Mapping .....	53
Table 20 -	Threats and Assumptions to Security Objectives Rationale .....	54
Table 21 -	TOE SFRs to TOE Security Objectives Mapping .....	58
Table 22 -	TOE SFRs to TOE Security Objectives Rationale .....	59
Table 23 -	IT Environment Security Functional Requirements to IT Environment Objectives Mapping .....	61
Table 24 -	IT Environment Security Functional Requirements to IT Environment Objectives Rationale .....	62
Table 25 -	Security Functional Requirements to Security Functions Mapping .....	64
Table 26 -	Security Functional Requirements to Security Functions Rationale .....	64
Table 27 -	TOE Security Functional Requirements Dependency Rationale.....	67
Table 28 -	IT Environment SFRs Dependency Rationale.....	68
Table 29 -	Assurance Measures.....	69

## ACRYONYM 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
IPSec.....	Internet Protocol Security
IT.....	Information Technology
I&A.....	Identification and Authentication
LDAP.....	Lightweight Directory Access Protocol
NAS.....	Network-Attached Storage
NDMP.....	Network Data Management Protocol
NetApp.....	Network Appliance
NIAP.....	National Information Assurance Partnership
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
SSH.....	Secure Shell
SSL.....	Secure Socket Layer
ST.....	Security Target
TCP.....	Transmission Control Protocol
TFTP.....	Trivial File Transfer Protocol
TOE.....	Target of Evaluation
TSC.....	TSF Scope of Control
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 Versions 7.0.3 and 7.0.4 Target of Evaluation (TOE). The language used in this Security Target is consistent with the *Common Criteria for Information Technology Security Evaluation, Version 2.2*, the ISO/IEC JTC 1/SC27, *Guide for the Production of PPs and STs, Version 0.9* and all international interpretations and selected National Information Assurance Partnership (NIAP) through January 9, 2006. 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.

<b>ST TITLE</b>	Network Appliance Data ONTAP Versions 7.0.3 and 7.0.4 Security Target
<b>ST Version</b>	Version 1-0
<b>Publication Date</b>	March 28, 2007
<b>Vendor</b>	Network Appliance, Inc.
<b>ST Author</b>	COACT, Inc.
<b>TOE Identification</b>	Data ONTAP Versions 7.0.3 and 7.0.4 Target of Evaluation
<b>CC Identification</b>	Common Criteria for Information Technology Security Evaluation, Version 2.2, January 2004
<b>Common Criteria Conformance</b>	The ST is compliant with the Common Criteria (CC) Version 2.2 assurance requirements conformant for EAL2.
<b>Protection Profile Conformance</b>	The TOE does not claim conformance to any Protection Profile.
<b>Keywords</b>	Operating System, access control, discretionary access control (DAC).

#### 1.2 TOE Overview

Data ONTAP Versions 7.0.3 and 7.0.4 Targets 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. Network Appliance's storage appliances are based on the Data ONTAP Versions 7.0.3 and 7.0.4 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 Versions 7.0.3 and 7.0.4 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 Versions 7.0.3 and 7.0.4 TOE is a proprietary microkernel operating system developed by Network Appliance. The microkernel is included in the distribution of several of Network Appliance's storage solution products including Filer, V-Series Virtual Filer (Virtual Filer), and NearStore. The Data ONTAP Versions 7.0.3 and 7.0.4 TOE provides data management functions that include providing secure data storage and multi-protocol access.

The security functionality provided by Data ONTAP Version 7.0.3 and Data ONTAP version 7.0.4 is identical. Data ONTAP Version 7.0.4 incorporates minor bug fixes (unrelated to any security functionality) and support for additional disk drives.

#### 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.

#### 2.3 Product Overview

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

Filer	NetApps 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 centered 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.

Included in each family are products. Table 1 identifies the Network Appliance families and the individual products included in the TOE's evaluated configuration.

**Table 1 - Network Appliance Products included in the TOE's Evaluated Configuration**

Family	Series	Products
Filer	FAS200 Series	FAS250, FAS270, FAS270C, FAS270HA,
	FAS900 Series	FAS920, FAS920C, FAS940, FAS940C, FAS940HA, FAS960, FAS960C, FAS960HA, FAS980, FAS980C, FAS980HA
	FAS3000 Series	FAS3020, FAS3020C, FAS3020HA, FAS3050, FAS3050HA
NearStore	n/a	NearStore R200
V-Series	n/a	GF270HA, GF920HA, GF940, GF940HA, GF960, GF960HA, GF980, GF980HA

## 2.4 TOE Component Systems

Data ONTAP Versions 7.0.3 and 7.0.4 TOE (hereafter referred to as Data ONTAP) is divided into two components: System Administration and WAFL. The two modules are described below. Their relationship to the IT Environment supplied components is depicted in Figure 1.

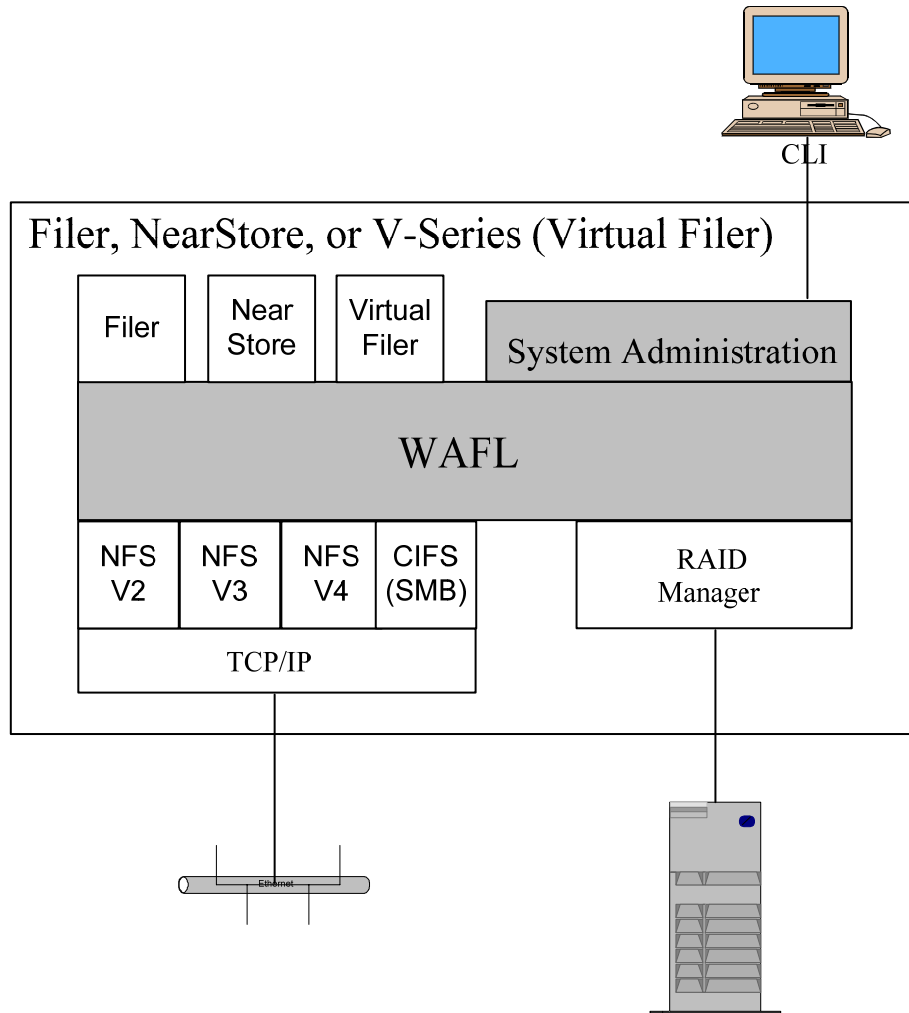
### WAFL

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.

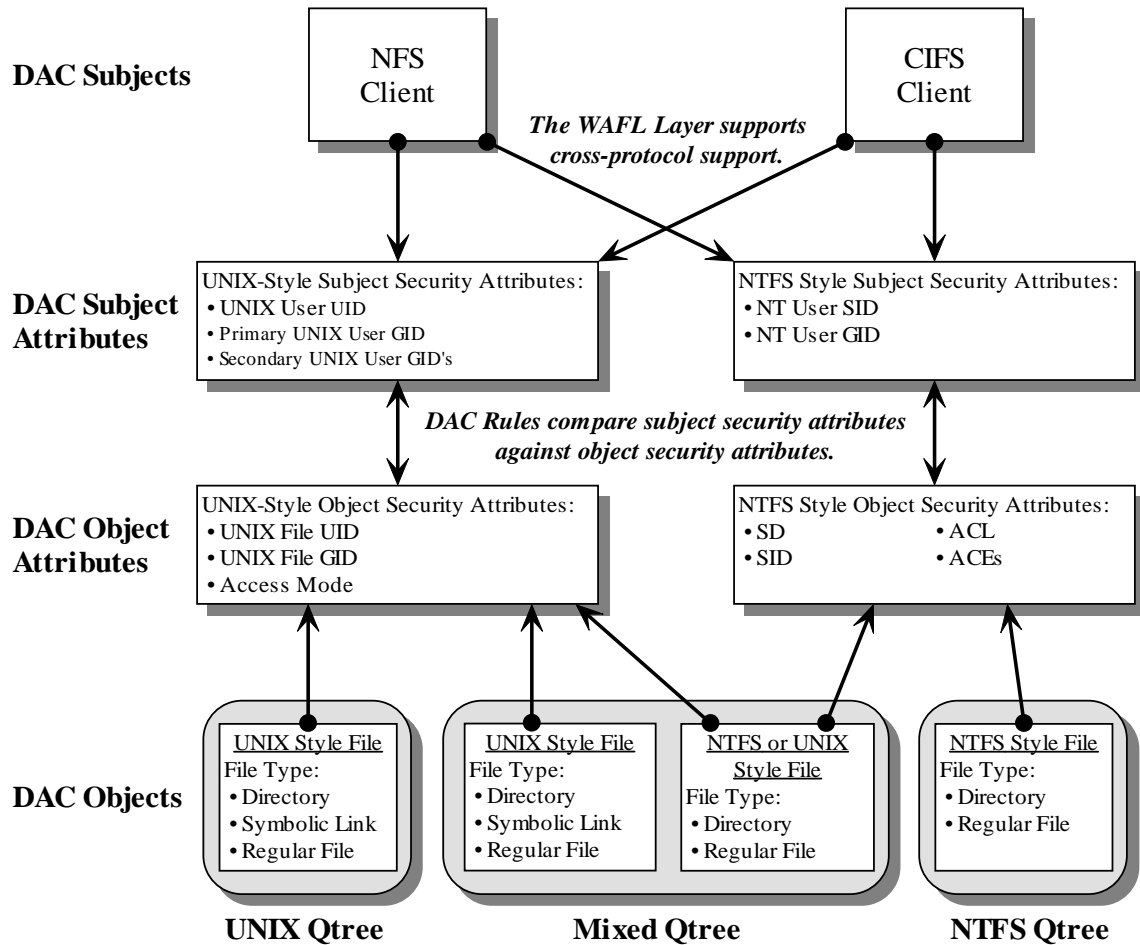
Figure 1 - TOE Components



### 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. 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 or Mixed qtrees. A qtree is a disk space partition. UNIX qtrees store UNIX-Style files with UNIX-Style security attributes. NTFS qtrees and Mixed qtrees store both style of files. Files stored in NTFS and 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 an NTFS or Mixed qtree based on the ACL. However, the TOE will not create an ACL based on UNIX-Style security attributes. Therefore, files in NTFS and mixed qtrees always have UNIX-Style security attributes but may not have an ACL (NTFS-Style security attributes).

#### **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 NT User SID and an NT 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 NT 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 NT User UIDs (NT 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, NFS V3, and NFS V4) 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 Network Appliance'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.

LDAP Server

The TOE's evaluated configuration provides an option to maintain UNIX username/UNIX User UID/UNIX User GID mappings either via a local TOE managed file (/etc/passwd) or via an IT Environment supplied NIS Server or IT Environment supplied LDAP Server. The LDAP Server is supplied by the IT Environment.

NIS Server

The TOE's evaluated configuration provides an option to maintain UNIX username/UNIX User UID/UNIX User GID mapping either via a local TOE managed file (/etc/passwd) or via the IT Environment (NIS or LDAP). The NIS Server is supplied by the IT Environment.

RAID Manager

The RAID Manager supports multiple disk drive support that supports 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.

### 2.6.1.3 Files

/etc/groups	A TOE resident appliance file that contains entries that specify a group name, the group's GID (Secondary UNIX User GID), and the UNIX usernames of users who belong to the group.
/etc/passwd file	A TOE resident appliance file used to authenticate administrators (local human users) connecting via the IT Environment supported serial port and also used, optionally to resolve NFS Client and CIFS Client security attributes (if local administration is selected). /etc/passwd contains entries identifying a user's UNIX Username, password, UNIX User UID, Primary UNIX User GID and user's real name.
/etc/usermap.cfg file	A TOE resident appliance file that contains NT Username and UNIX Username mappings used for cross protocol access for both CIFS Clients and UNIX Clients.
wabl.default_unix_user file	A TOE resident appliance file that contains a default UNIX Username used to resolve a CIFS Client's NT Username for cross protocol access.
wabl.default_nt_user file	A TOE resident appliance file that contains a default NT Username used to resolve an NFS Client's UNIX Username for cross protocol access.

### 2.6.1.4 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.

### 2.6.1.4.1 Client Security Attributes

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

### 2.6.1.4.2 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 NTFS-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 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. The rwx triplet identifies the permission for that set (owner, group, user). 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. An SD contains an 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 Functionality Excluded from the TOE's Logical Boundary

The TOE, Data ONTAP, is self contained in the WAFL and System Administration components and does not contain any security functionality not claimed in the ST.

## 2.8 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 an operator 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.

## 2.9 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.

## **2.10 TOE Evaluated Configuration**

### **2.10.1 TOE Evaluated Configuration Systems**

The evaluated configuration will include one or more instances of a product in one of the following Network Appliance families: Filer, V-Series Virtual Filer, and NearStore. Because the security functionality of Data ONTAP Versions 7.0.3 and 7.0.4 are equivalent and both versions are evaluated, either version may be used in the evaluated configuration.

### **2.10.2 TOE Evaluated Configuration Options**

The following sections describe the evaluated configuration options.

#### **2.10.2.1 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.

#### **2.10.2.2 Name Service Options**

The evaluated configuration supports both local and remote resolution (NIS or LDAP) of TSF Data used to support the DAC SFP, but does not support remote resolution of authentication data (nsswitch.conf passwd file).

#### **2.10.2.3 Miscellaneous**

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

- B) Shared level ACLs are not evaluated
- C) The password field in the /etc/groups file is not used (should be blank).
- D) The evaluated configuration will not include the bypass traverse checking option.
- 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 style once the qtree is configured.
- G) The evaluation configuration disables CIFS and NFS access to the /etc directory

## 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 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 2 - 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 operate under the same security policy constraints.

##### 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 3 - 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 careless, willfully negligent or 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 4 - Physical Assumptions**

Assumption	Assumption Description (Physical)
A.LOCATE	The processing resources of the TOE will be located within controlled access facilities that will prevent unauthorized physical access.
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.
A.CONNECT	All devices on which the TOE resides and their connections will be housed within a controlled access facility.

### 3.2 Threats

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

**Table 5 - Threats**

Threat	Threat Description
T.ADMIN_ERROR	Improper administration may result in defeat of specific security features.
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.
T.USER_CORRUPT	User data may be tampered with by other users.

### 3.3 Organizational Security Policies

There are no Organizational Security Policies identified for this TOE.



## 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 6 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 6 - Security Objectives for the TOE**

Security Objective (TOE)	TOE Security Objective Description
O.ACCESS	The TOE will ensure that users gain only authorized access to the TOE and to the data the TOE manages.
O.ADMIN_ROLES	The TOE will provide administrative roles to isolate administrative actions.
O.DAC_ACC	The TOE will control accesses 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 organizational policies are enforced in the target environment.
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 7 identifies the security objectives for the TOE's IT Environment.

**Table 7 - 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 allow authorized users to access only appropriate TOE functions and data.
O.E.SUBJECTDATA	The IT Environment will provide the TOE with the appropriate subject security attributes.

#### 4.2.2 Security Objectives for the Non-IT Environment

Table 8 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 8 - 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 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 2.1* and all National Information Assurance Partnership (NIAP) and 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: indicated in underlined text.

Assignments within selections: *indicated in italics and underlined text*.

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 9 are described in more detail in the following subsections.

**Table 9 - Security Functional Requirements of the TOE**

Security Functional Requirement (TOE)	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
FIA_USB.1	User-Subject Binding
FMT_MSA.1	Management of Security Attributes
FMT_MSA.3	Static Attribute Initialisation

Security Functional Requirement (TOE)	Security Functional Requirement Name
FMT_MTD.1	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1(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 User Data Protection (FDP)

#### 5.1.1.1 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**

<i>Subject</i>	<i>Object</i>			<i>Operation among Subject and Object covered by the DAC SFP</i>
	<i>File Style</i>	<i>File Type</i>	<i>Qtree Type</i>	
<i>NFS Client</i>	<i>UNIX-Style file</i>	<i>Directory, Symbolic link, Regular file</i>	<i>UNIX Qtree or Mixed Qtree</i>	<i>Create, read, write, execute, delete, change permissions, change ownership</i>
	<i>NTFS-Style file</i>	<i>Directory, Regular file</i>	<i>NTFS Qtree or Mixed Qtree</i>	<i>Create, read, write, execute, delete, change permissions, change ownership</i>
<i>CIFS Client</i>	<i>NTFS-Style file</i>	<i>Directory, Regular file</i>	<i>NTFS Qtree or Mixed Qtree</i>	<i>Create, read, write, execute, delete, change permissions, change ownership</i>
	<i>UNIX-Style file</i>	<i>Directory, Regular file</i>	<i>UNIX Qtree or Mixed Qtree</i>	<i>Create, read, write, execute, delete, change permissions, change ownership</i>

<i>Subject</i>	<i>Object</i>			<i>Operation among Subject and Object covered by the DAC SFP</i>
	<i>File Style</i>	<i>File Type</i>	<i>Qtree Type</i>	
<i>Administrators</i>	<i>UNIX-Style file</i>	<i>Directory, Symbolic link, Regular file</i>	<i>UNIX Qtree or Mixed Qtree</i>	<i>Create, read, write, execute, delete, change permissions, change ownership</i>
	<i>NTFS-Style file</i>	<i>Directory, Regular file</i>	<i>NTFS Qtree or Mixed Qtree</i>	<i>Create, read, write, execute, delete, change permissions, change ownership</i>

**5.1.1.2 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**

<i>Operation</i>	<i>Subject</i>	<i>Object</i>	<i>Subject</i>		<i>Object (file) Security Attribute</i>	<i>Other Objects and Security Attribute used for DAC SFP</i>
			<i>Security Attribute</i>	<i>Other TSF Data</i>		
<i>Create</i>	<i>NFS Client</i>	<i>UNIX-Style file</i>	<i>UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs</i>	<i>None</i>	<i>N/A</i>	<i>Parent directory's UNIX file UID, UNIX file GID and access mode</i>

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

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

<i>Operation</i>	<i>Subject</i>	<i>Object</i>	<i>Subject</i>		<i>Object (file) Security Attribute</i>	<i>Other Objects and Security Attribute used for DAC SFP</i>
			<i>Security Attribute</i>	<i>Other TSF Data</i>		
		<i>UNIX-Style file stored in a Mixed qtree</i>				
		<i>UNIX- Style file stored in a Mixed qtree</i>	<i>UNIX User UID, NT User SID, NT User GID, Primary UNIX User GID and Secondary UNIX User GIDs</i>	<i>UNIX Username, NT Username</i>	<i>None</i>	<i>Qtree type, Parent directory's SID and ACEs or Parent directory's UNIX File UID, UNIX file GID and access mode.</i>
<i>Change Permission</i>	<i>NFS Client</i>	<i>UNIX- Style file</i>	<i>UNIX User UID, Primary UNIX User GID and Secondary UNIX User GIDs</i>	<i>None</i>	<i>UNIX File UID, UNIX file GID and access mode</i>	<i>Parent directory's UNIX File UID, UNIX file GID and access mode</i>
	<i>CIFS Client</i>		<i>NT User SID, NT User GID</i>	<i>NT Username</i>		
	<i>NFS Client</i>	<i>NTFS- Style file</i>	<i>NT User SID, NT User GID</i>	<i>NT Username</i>	<i>SID and ACEs</i>	<i>Parent directory's SID and ACEs</i>



<i>Operation</i>	<i>Subject</i>	<i>Object</i>	<i>Subject</i>		<i>Object (file) Security Attribute</i>	<i>Other Objects and Security Attribute used for DAC SFP</i>
			<i>Security Attribute</i>	<i>Other TSF Data</i>		
	<i>CIFS Client</i>		<i>NT User SID, NT User GID</i>	<i>NT Username</i>	<i>SID and ACEs</i>	<i>Parent directory's SID and ACEs</i>
<i>Change Owner</i>	<i>Change Owner</i>	<i>UNIX- Style file</i>	<i>UNIX User UID</i>	<i>None</i>	<i>None</i>	<i>None</i>

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**

<i>Subject</i>	<i>Operation</i>	<i>Object</i>				<i>DAC Rule</i>
		<i>Qtree Style</i>	<i>File Style</i>	<i>Parent Directory has an ACL</i>	<i>File (object) has an ACL</i>	
<i>NFS Client or CIFS Client</i>	<i>Create</i>	<i>UNIX</i>	<i>N/A</i>	<i>N/A</i>	<i>N/A</i>	<p>1. The subject is the owner of the parent directory and the owner has been granted Write and Execute access (UNIX-Style security attributes).</p> <p>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).</p> <p>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).</p>
		<i>NTFS or Mixed</i>	<i>N/A</i>	<i>Yes</i>	<i>N/A</i>	<p>4. There is no parent directory ACE that denies Write or Execute access to the subject and parent directory ACEs exists that grant Write and Execute permission to the subject (NTFS-Style security attributes).</p> <p>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).</p>
		<i>NTFS or Mixed</i>	<i>N/A</i>	<i>No</i>	<i>N/A</i>	<p>6. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory).</p>

<i>Subject</i>	<i>Operation</i>	<i>Object</i>				<i>DAC Rule</i>
		<i>Qtree Style</i>	<i>File Style</i>	<i>Parent Directory has an ACL</i>	<i>File (object) has an ACL</i>	
	<i>Read, Write, Execute</i>	<i>UNIX, NTFS or Mixed</i>	<i>UNIX-Style file</i>	<i>N/A</i>	<i>No</i>	<p>7. The subject is the owner of the file and the owner has been granted access for the specific operation (UNIX-Style security attributes).</p> <p>8. 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).</p> <p>9. 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)</p>
		<i>NTFS or Mixed</i>	<i>NTFS-Style file</i>	<i>N/A</i>	<i>Yes</i>	<p>10. 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).</p> <p>11. 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).</p>
	<i>Delete</i>	<i>UNIX</i>	<i>UNIX-Style file</i>	<i>N/A</i>	<i>N/A</i>	12. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory).

Subject	Operation	Object				DAC Rule
		Qtree Style	File Style	Parent Directory has an ACL	File (object) has an ACL	
		NTFS	NTFS-Style file	Yes	Yes	<p>13. Rule 10 or 11 above is true for Delete operation (subject has Delete NTFS-Style permission for object).</p> <p>14. Rule 13 above fails and Rule 15 or 16 below is true (subject has Delete Child NTFS-Style permission for parent directory).</p>
		NTFS	UNIX-Style file	Yes	No	<p>15. 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).</p> <p>16. 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).</p>
		NTFS	NTFS-Style file	No	Yes	<p>17. Rule 10 or 11 above is true for Delete operation (subject has Delete NTFS-Style permission for object).</p> <p>18. Rule 17 above fails and Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory).</p>
		NTFS	UNIX-Style file	No	No	19. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory).
		Mixed	NTFS-Style file	Yes	Yes	<p>20. Rule 10 or 11 above is true (subject has Delete NTFS-Style access for the object).</p> <p>22. Rule 15 or 16 above is true (subject has Delete Child NTFS-Style permission for the parent directory).</p>

<i>Subject</i>	<i>Operation</i>	<i>Object</i>				<i>DAC Rule</i>
		<i>Qtree Style</i>	<i>File Style</i>	<i>Parent Directory has an ACL</i>	<i>File (object) has an ACL</i>	
		<i>Mixed</i>	<i>UNIX-Style file</i>	<i>Yes</i>	<i>No</i>	<p>23. Rule 10 or 11 above is true (subject has Delete NTFS-Style access for the object).</p> <p>24. Rule 15 or 16 above is true (subject has Delete Child NTFS-Style permission for the parent directory).</p> <p>25. Rule 3, 4 or 5 above is true (subject has Write and Execute UNIX-Style permission for parent directory).</p>
		<i>Mixed</i>	<i>NTFS-Style file</i>	<i>No</i>	<i>Yes</i>	26. Rule 10 or 11 above is true (subject has Delete NTFS-Style access for the object).
		<i>Mixed</i>	<i>UNIX-Style file</i>	<i>No</i>	<i>No</i>	27. Rule 3, 4 or 5 above is true (subject has Write and Execute UNIX-Style permission for parent directory).
	<i>Change Permission</i>	<i>UNIX</i>	<i>UNIX-Style file</i>	<i>N/A</i>	<i>N/A</i>	28. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory) and rule 7, 8 or 9 above is true for Write operation (UNIX-Style permission for object).
		<i>NTFS or Mixed</i>	<i>NTFS-Style file</i>	<i>Yes</i>	<i>Yes</i>	29. Rule 4 or 5 above is true (subject has Write and Execute NTFS-Style permission for parent directory) and rule 10 or 11 above is true for Change Permission operation (NTFS-Style permission for object).
		<i>NTFS or Mixed</i>	<i>UNIX-Style file</i>	<i>Yes</i>	<i>No</i>	30. Rule 4 or 5 above is true (subject has Write and Execute NTFS-Style permission for parent directory) and rule 7, 8 or 9 above is true for Write operation (UNIX-Style permission for object).

<i>Subject</i>	<i>Operation</i>	<i>Object</i>				<i>DAC Rule</i>
		<i>Qtree Style</i>	<i>File Style</i>	<i>Parent Directory has an ACL</i>	<i>File (object) has an ACL</i>	
		<i>NTFS or Mixed</i>	<i>UNIX-Style file</i>	<i>No</i>	<i>Yes</i>	<i>31. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory) and rule 10 or 11 above is true for Change Permission operation (NTFS-Style permission for object).</i>
		<i>NTFS or Mixed</i>	<i>UNIX-Style file</i>	<i>No</i>	<i>No</i>	<i>32. Rule 1, 2 or 3 above is true (subject has Write and Execute UNIX-Style permission for parent directory) and rule 7, 8 or 9 above is true for Write operation (UNIX-Style permission for object).</i>
<i>CIFS Client</i>	<i>Change Owner</i>	<i>NTFS or Mixed</i>	<i>NTFS-Style file</i>	<i>N/A</i>	<i>Yes</i>	<i>33. Rule 10 or 11 above is true for Change Owner operation (subject has Change Owner NTFS-Style permission for object).</i>

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**

<i>Subject</i>	<i>Operation</i>	<i>Object</i>	<i>DAC Rule</i>
<i>Administrator</i>	<i>All</i>	<i>All</i>	<i>Access is allowed</i>
<i>NFS Client</i>	<i>Change Owner</i>	<i>UNIX-Style file</i>	<i>The subject is root</i>
<i>CIFS Client</i>	<i>Change Owner</i>	<i>NTFS-Style file without an ACL</i>	<i>Access is allowed.</i>

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**

<i>Subject</i>	<i>Operation</i>	<i>Object</i>	<i>DAC Rule</i>
<i>NFS Client</i>	<i>Change Owner</i>	<i>NTFS-Style file stored in a NTFS qtree</i>	<i>Request denied</i>
<i>CIFS Client</i>	<i>Change Owner</i>	<i>UNIX-Style file stored in a UNIX qtree</i>	<i>Request denied</i>

## **5.1.2 Identification and Authentication (FIA)**

### **5.1.2.1 FIA\_ATD.1(1) User Attribute Definition**

FIA\_ATD.1.1(1) The TSF shall maintain the following list of security attributes belonging to individual users: *UNIX User UID and Primary UNIX User GID.*

*Application note: the evaluated configuration provides an option to maintain UNIX username/UNIX User UID/UNIX User GID mapping either via a local TOE managed file (/etc/passwd) or via the IT Environment (NIS or LDAP). This SFR applies when the evaluated configuration is selected for local administration.*

### **5.1.2.2 FIA\_ATD.1(2) User Attribute Definition**

FIA\_ATD.1.1(2) The TSF shall maintain the following list of security attributes belonging to individual users: *Secondary UNIX User GIDs.*

*Application note: the evaluated configuration provides an option to maintain Secondary UNIX User GIDs either via a local TOE managed file (/etc/groups) or via the IT Environment (NIS or LDAP). This SFR applies when the evaluated configuration is selected for local administration.*

### **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.

#### **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.

#### **5.1.2.5 FIA\_USB.1 User-Subject Binding**

FIA\_USB.1.1 The TSF shall associate the appropriate user security attributes with subjects acting on behalf of that user.

### **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 *UNIX User UID and Primary UNIX User GID maintained locally by the TOE to an administrator.*

*Application note: the evaluated configuration provides an option to maintain UNIX username/UNIX User UID/UNIX User GID mapping either via a local TOE managed file (/etc/passwd) or via the IT Environment (NIS or LDAP). This SFR applies when the evaluated configuration is selected for local administration.*

#### **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 UNIX User GIDs maintained locally by the TOE to an administrator.*

*Application note: the evaluated configuration provides an option to maintain Secondary UNIX User GIDs either via a local TOE managed file (/etc/groups) or via the IT Environment (NIS or LDAP). This SFR applies when the evaluated configuration is selected for local administration.*

#### **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 *UNIX Username stored in /etc/passwd to administrators.*

#### **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 *UNIX User Password stored in /etc/passwd to administrators.*



**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 *NT Username and UNIX Username mapping stored in /etc/usermap.cfg* to administrators.

**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 file* to administrators.

**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 *NT Username stored in the wafl.default\_nt\_user file* to administrators.

**5.1.3.9 FMT\_SMF.1 Specification of Management Functions**

FMT\_SMF.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 an administrator to manage TSF Data and configure TSF Functions.*
- 2) *Provides I&A functions that require administrators to identify and authenticate themselves to the TOE before allowing any modifications of TSF Data.*
- 3) *Provides a CLI function that enables a user to specify DAC subject security attribute resolution via locally maintained TOE files or via the IT Environment (NIS or LDAP).*

**5.1.3.10 FMT\_SMR.1(1) Security Roles**

FMT\_SMR.1.1(1) The TSF shall maintain the roles *administrator and non-administrator*.

FMT\_SMR.1.2(1) The TSF shall be able to associate users with roles.

**5.1.4 Protection of the TSF (FPT)****5.1.4.1.1 FPT\_RVM\_SW\_EXP.1 Non-Bypassability of the TSP for Software TOEs**

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

**5.1.4.1.2 FPT\_SEP\_SW\_EXP.1 TSF Domain Separation for Software TOEs**

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 Table 15 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_SMR.1	Security Roles
FPT_RVM_HW_EXP.1	Non-Bypassability of the TSP
FPT_SEP_HW_EXP.1	Domain Separation environment

### 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 *IT Administrators (subjects), files containing the UNIX username/UNIX User UID/Primary UNIX User GID/Secondary UNIX User GID/ NT User SID/NT User GID/NT 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 IT Administrators can have read and write access to files containing the UNIX username/UNIX User UID/Primary UNIX User GID/Secondary UNIX User GID/ NT User SID/NT User GID/NT 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 IT Administrators can have read and write access to files containing the UNIX username/UNIX User UID/Primary UNIX User GID/Secondary UNIX User GID/ NT User SID/NT User GID/NT 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(3) 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.*

*Application note: the evaluated configuration provides an option to maintain UNIX username/UNIX User UID/Primary UNIX User GID mapping either via a local TOE managed file (/etc/passwd) or via the IT Environment (NIS or LDAP). This SFR applies when the evaluated configuration is selected for IT Environment administration.*

### 5.2.2.2 FIA\_ATD.1(4) 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.*

*Application note: the evaluated configuration provides an option to maintain Secondary UNIX User GIDs either via a local TOE managed file (/etc/groups) or via the IT Environment (NIS or LDAP). This SFR applies when the evaluated configuration is selected for IT Environment administration.*

### 5.2.2.3 FIA\_ATD.1(5) User Attribute Definition

- FIA\_ATD.1.1(5) The **IT Environment** shall maintain the following list of security attributes belonging to individual users: *NT User SID and NT 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.

### 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.

### 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 an IT 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 an IT administrator*.

*Application note: the evaluated configuration provides an option to maintain UNIX username/UNIX User UID/Primary UNIX User GID mapping either via a local TOE managed file or via the IT Environment (NIS). This SFR applies when the evaluated configuration is selected for IT administration.*

#### 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 an IT administrator*.

*Application note: the evaluated configuration provides an option to maintain Secondary UNIX User GIDs either via a local TOE managed file or via the IT Environment. This SFR applies when the evaluated configuration is selected for IT administration.*

#### 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 *NT User SID and NT User GID maintained by the domain controller to an IT 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 *IT 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 an IT administrator*.

*Application note: the evaluated configuration provides an option to maintain UNIX username/UNIX User UID/Primary UNIX User GID mapping either via a local TOE managed file (/etc/passwd) or via the IT Environment (NIS or LDAP). This SFR applies when the evaluated configuration is selected for IT administration.*

#### **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 *NT Username maintained by a CIFS Client* to an *IT 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 *NT Username maintained by the IT Environment's Domain Controller* to an *IT administrator*.

#### **5.2.3.9 FMT\_SMR.1(2) Security Roles**

FMT\_SMR.1.1(2) The **IT Environment** shall maintain the roles *IT 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 Os's**

FPT\_RVM\_HW\_EXP.1.1 The security functions of the host OS shall ensure that host OS security policy enforcement functions are invoked and succeed before each function within the scope of control of the host OS is allowed to proceed.

##### **5.2.4.1.1 FPT\_SEP\_HW\_EXP.1 TSF Domain Separation for Os's**

FPT\_SEP\_HW\_EXP.1.1 The security functions of the host OS shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects in the scope of control of the host OS.

FPT\_SEP\_HW\_EXP.1.2 The security functions of the host OS shall enforce separation between the security domains of subjects in the scope of control of the host OS.

### **5.3 TOE Security Assurance Requirements**

The TOE meets the assurance requirements for EAL2. These requirements are summarised in Table 16 below.

**Table 16 - Assurance Requirements**

Assurance Class	Component ID	Component Description
Configuration Management	ACM_CAP.2	Version Numbers and Configuration Items

Assurance Class	Component ID	Component Description
Delivery and Operation	ADO_DEL.1	Delivery Procedures
	ADO_IGS.1	Installation, Generation, and Start-up Procedures
Development	ADV_FSP.1	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
Tests	ATE_COV.1	Evidence of Coverage
	ATE_FUN.1	Functional Testing
	ATE_IND.2	Independent Testing
Vulnerability Assessment	AVA_SOF.1	Strength of Function
	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 an administrator to manage and support the TSF.

The TOE maintains two roles for users: administrators and non-administrators.

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

The TOE ensures that only 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-bypassibility of functions within the TOE's scope of control (TSC).

#### 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. The following sections describe the DAC SFP and provide the Security Functional Requirement that meet 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. In general (), 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 style of files and in addition, the files may have both UNIX-Style security attributes and NTFS-Style security attributes associated with them or they may have only one type of security attributes associated with them. 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. The following sections describe the security attributes associated with the objects.

#### **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. The rwx triplet identifies the permission for that set (owner, group, user). 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 user 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 NT 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 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	Add, delete or rename a file in the directory.	Write to the file the symbolic link contains the name of.	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	Become the directory's owner.	Become the file'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. 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 owner of files in a UNIX qtree or mixed qtree. Only CIFS Clients can change permission or change owner of files in an NTFS qtree or mixed qtree.
- 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 determines the type of security attributes required by the client. If a file has NTFS-Style security attributes (an ACL) they are considered first for create, read, write, execute, delete and change permission - regardless of client type.
- 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 directory operations.

**Operation** The operations supported by the DAC (Create, Read, Write, Execute, Change Permissions, 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 administrators, NFS Clients and CIFS Clients (FDP\_ACC.1(1)). The latter two subjects access the TOE via remote systems (process acting on behalf of a user) (FIA\_USB.1). To determine if access is permitted for an object, the TOE requires the security attributes associated with the client. These security attributes may be resolved by the TOE, the IT Environment or both.

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. Administrators are always UNIX-style clients. The following sections describe the TOE's subject security attribute resolution used to enforce the DAC SFP.

##### **6.2.1.4.1 UNIX-Style Client 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 administrator, the TOE determines the UNIX User UID and username from the I&A functionality. The TOE then searches the `/etc/passwd` file (or IT Environment) to get the Primary UNIX User GID. The TOE then uses the UNIX username to search the `/etc/group` file (or IT Environment) to obtain 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 (IT Environment). The TOE then searches the `/etc/passwd` file (or IT Environment) to get the Primary UNIX User GID and UNIX username. The TOE then uses the UNIX username to search the `/etc/group` file (or IT Environment) to obtain 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 (NT username) when the client logged onto the system and joined the NT 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 NT

username. If a match is found, the UNIX username is used. Otherwise, the TOE converts the NT username to lowercase and uses this name as a UNIX username. The TOE then looks up the UNIX username in the `/etc/passwd` file (or IT Environment) to find the UNIX User UID and Primary UNIX User GID. If no entry exists for the name, the TOE uses the UNIX User username specified in the `waf.default_unix_user` file. The TOE then uses this name to obtain the UNIX User UID and Primary UNIX User GID from the `/etc/passwd` file (or IT Environment). The TOE then searches the `/etc/group` file (or IT Environment) for the UNIX username to find any Secondary UNIX User GIDs (FIA\_ATD.1).

For the remainder of this document, when the documents states 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 NTFS-Style Client 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: an NT User SID and an NT User GID.

If the access request is initiated by a CIFS Client, the TOE obtained the CIFS Client's username (NT username) when the client logged onto the remote system and joined the NT Domain. In addition to this, the IT Environment queried the domain controller to obtain the NT User SID and the NT 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 first finds the UNIX username that maps to the UNIX User UID by searching the `/etc/passwd` file (or IT Environment) (FIA\_ATD.1(1)). If a match is not found, access is denied. If a match is found, the TOE uses the UNIX username to find the NT username in the `/etc/usermap.cfg` file. If a match is not found, the TOE uses the NT username specified in the `waf.default_nt_user` file. Given the NT username, the TOE finds the NT User SID and NT User GID by querying the IT Environment's Domain Controller.

For the remainder of this document, when the documents states 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) file are used to determine access and the type of qtree the file is stored in 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 has an ACL (an NTFS-Style file stored in an NTFS qtree or stored in a mixed qtree with and ACL), the TOE uses NTFS-Style security attributes for both subject and object to determine if access is permitted. If the client has write and execute privileges for the parent directory, the file is created (FDP\_ACF.1(1)). The new file inherits the NTFS-Style security attributes from the parent directory (FMT\_MSA.3(1)). If the client does not have write and execute privileges to the parent directory, the request is denied.

If the client requests to create a file in a directory that does not have an ACL (UNIX qtree or mixed qtree without an ACL), the TOE uses UNIX-Style security attributes for both subject and object to determine access. If the client has write and execute privileges for the parent directory, the file is created with the UNIX-Style security attributes inherited from the parent directory (FDP\_ACF.1(1), FMT\_MSA.3(1)). If the client does not have write and execute privileges to the parent directory, the request is denied.

### **Read, Write, Execute Access Requests**

To determine if a client has permission to read, write or execute a file, the TOE first determines if the file has an ACL. If the file has an ACL (the file is an NTFS-Style file stored in an NTFS qtree or mixed qtree), 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.

If the file does not have an ACL (the file is a UNIX-Style file stored in a UNIX qtree or stored in an NTFS or mixed qtree without an ACL), the TOE uses UNIX-Style security attributes for both subject and object to determine if the 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.

### **Client Delete Access Request**

To determine if a client has permission to delete a file, the TOE first looks at the type of qtree the file is stored in.

#### UNIX qtree

If the file is stored in a UNIX qtree, 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.

#### NTFS qtree

If the file is stored in an NTFS qtree, the TOE first determines if the parent directory has an ACL. If the parent directory has an ACL, the TOE then determines if the file has an ACL. If the file does and the client has delete access to the file, access is granted (FDP\_ACF.1(1)). If the file doesn't have an ACL or has an ACL but the 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)). If the parent doesn't have a DC ACE, access is denied.

If the file is stored in an NTFS qtree and the parent doesn't have an ACL, the TOE determines if the file has an ACL. If the file does and the client has delete access to the file, access is granted (FDP\_ACF.1(1)). If the file doesn't have an ACL or has an ACL but the ACEs do not grant delete permission to the client, the TOE uses UNIX-Style security attributes for both subject and object to determine if delete access is permitted. The TOE 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.

### Mixed qtree

If the file is stored in a mixed qtree, the TOE first determines if the parent directory has an ACL. If the parent directory has an ACL, the TOE then determines if the file has an ACL. If the file does and the client has delete access to the file, access is granted (FDP\_ACF.1(1)). If the file doesn't have an ACL or has an ACL but the 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. If the parent doesn't have a DC ACE, the TOE then determines, using UNIX-Style security attributes for both subject and object, 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.

If the file is stored in a mixed qtree and the parent does not have an ACL, the TOE determines if the file has an ACL. If the file has an ACL and the client has delete access to the file, access is granted (FDP\_ACF.1(1)). If the client does not have delete access, access is denied. If the file does not have an ACL, the TOE then determines, using UNIX-Style security attributes for both subject and object, 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 change permission access permission the TOE first determines if the client has write and execute permission for the file's parent directory. The TOE uses NTFS-Style security attributes for a parent directory with an ACL and UNIX-Style security attributes for a parent directory without an ACL. If the TOE determines if the client does not have write and execute permissions for the file's directory, access is denied.

If the client has write and execute permission for the directory and the file is stored in a UNIX qtree or mixed qtree without an ACL, the TOE, using UNIX-Style security attributes for both subject and object, determines if the client has write access for the file. If the client doesn't, access is denied. If the client does, access is permitted (FDP\_ACF.1(1)).

If the file has an ACL, the TOE uses NTFS-Style security attributes for both subject and object to determine if access is allowed. The TOE determines if the client has Change Permission permissions for the file. If the client doesn't, access is denied. If the client does, access is permitted and the file's security attributes are overwritten by the new attributes (FDP\_ACF.1(1)).

## 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 a file and the file is stored in an NTFS qtree, the request is denied (FDP\_ACF.1(1)). If the file is stored in a UNIX qtree or mixed qtree, 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 file stored in a UNIX qtree, the request is denied (FDP\_ACF.1(1)). If the file is stored in an NTFS qtree (the file has an ACL) or stored in a mixed qtree and has an ACL, 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 file is stored in a mixed qtree, the TOE also creates UNIX-Style security attributes for the file. If the CIFS Client does not have Change Owner privileges, the request is denied.

If the file is stored in a mixed qtree and does not have an ACL, access is allowed (FDP\_ACF.1(1)). The TOE creates an ACL including a new owner ACE and creates UNIX-Style security attributes for the file.

## 6.3 Administrative Security Function

The Administrative Security Function provides the necessary functions to allow an administrator to manage and support the TSF. Included in this functionality is the rules enforced by the TOE that defines access to TOE maintained TSF Data and TSF Functions. The TSF Data includes both authentication data (used to authenticate 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 an administrator to manage and support the TSF (FMT\_SMF.1).

### 6.3.2 Roles

The TOE maintains two roles for users: administrators and non-administrators (FMT\_SMR.1(1)).

An administrator is any local human user who accesses the TOE via the serial port. 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. 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 (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)). Administrator's authentication data is maintained by the TOE in a local file (/etc/passwd). The file contains the username, password, username and the full name of each 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) UNIX Username Management
- 2) UNIX User secondary groups
- 3) UNIX Username to NT Username Mapping
- 4) Default usernames for cross protocol access requests.
- 5) Deny unauthorized administrative login attempts via Data ONTAP,
- 6) 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.

#### UNIX Username Management

The TOE maintains authentication data locally that is used to authenticate administrators (local human users) connecting via the serial port. This file, /etc/passwd, contains entries identifying a user's username, password, UNIX User UID, Primary UNIX User GID and a user's real name. The TOE uses this file to access authentication data for users and to optionally resolve DAC SFP security attributes (if local administration is selected). Specifically, the TOE uses this file to:

- 1) Validate a local human user's login id and password to allow local administration,
- 2) Find a Primary UNIX User GID given a UNIX User UID (UNIX-Style security attribute resolution for an NFS Client),
- 3) Find a UNIX User UID and Primary UNIX User GID given a UNIX Username (UNIX-Style security attribute resolution for a CIFS Client) and to

- 4) Find a UNIX Username given a UNIX User UID (NTFS-Style security attribute resolution for an NFS Client),

Only administrators may modify this file (FMT\_MSA.1(1), FMT\_MTD.1(1), FMT\_MTD.1(2)).

#### Secondary UNIX GIDs

The TOE maintains a file, /etc/groups that contains names of groups, the group's GID (Secondary UNIX User GID), and the UNIX usernames of users who belong to the group. The TOE uses this file to find any Secondary UNIX User GIDs associated with a user.

Only authorized local human users administrators may modify this file (FMT\_MSA.1(2), FMT\_MTD.1(1), FMT\_MTD.1(2)).

#### UNIX Usernames to NT Username Mapping

The TOE maintains a local /etc/usermap.cfg file that contains a mapping of UNIX Usernames to NT 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 TOE 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 resolve a client's username for cross protocol access. Waf.default\_unix\_user contains the default UNIX username for CIFS Client. Waf.default\_nt\_user contains the default NT username for NFS Client. Section 8.3.1.3.1 and 8.3.1.2.1 describe the access to the files. Only 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 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.ACCESS	O.ADMIN_ROLES	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.N.CREDEN	O.N.INSTALL	O.N.PHYSICAL	O.N.TRAINED	
T.ADMIN_ERROR		X				X										X
T.CONFIG_CORRUPT	X	X			X		X	X		X						
T.UNAUTH_ACCESS	X	X		X	X		X	X	X	X						
T.USER_CORRUPT	X	X	X		X		X	X		X	X					
A.CONNECT															X	
A.COOP												X				
A.LOCATE														X		
A.MANAGE													X			
A.NO_EVIL_ADM												X				

Threat/Assumption	TOE, IT Environment and Non-IT Environment Objectives														
	O.ACCESS	O.ADMIN_ROLES	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.N.CREDEN	O.N.INSTALL	O.N.PHYSICAL	O.N.TRAINED
A.PEER													X		
A.PROTECT														X	

**Table 20 - Threats and Assumptions to Security Objectives Rationale**

Threat/Assumption	Security Objective Rationale (TOE, IT Environment and Non-IT Environment)
T.ADMIN_ERROR	<p>O.N.TRAINED – this object addresses this threat by requiring users to be trained. Therefore, reducing the threat of improper administration by an unknowledgeable or incompetent administrator.</p> <p>O.ADMIN_ROLES – this objective addresses this threat by isolating the amount of damage a user can perform by requiring authorized roles for administrators to perform administrative procedures.</p> <p>O.MANAGE – this objective addresses this threat by providing the necessary functions that enable proper administrative support of the TOE’s security functionality.</p>

Threat/Assumption	Security Objective Rationale (TOE, IT Environment and Non-IT Environment)
T.CONFIG_CORRUPT	<p>O.ACCESS – 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.</p> <p>O.I&amp;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.</p> <p>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.</p> <p>O.E.ACCESS – 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.</p> <p>O.E.I&amp;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.</p> <p>O.E.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.</p>



Threat/Assumption	Security Objective Rationale (TOE, IT Environment and Non-IT Environment)
T.UNAUTH_ACCESS	<p>O.ACCESS – this objective addresses this threat by enforcing a DAC SFP that defines and enforces restrictive access and modification rules for security attributes and TSF Data managed by the TOE.</p> <p>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.</p> <p>O.I&amp;A - this objective builds on the O.ACCESS objective by requiring users to identify and authenticate themselves to the TOE before attempting to modify TSF data or security attributes via a telnet or serial connection.</p> <p>O.ADMIN_ROLES – this objective supports the O.ACCESS by requiring authorized roles for users to perform administrative procedures therefore, isolating the amount of damage a user can perform.</p> <p>O.E.ACCESS – this objective builds on O.ACCESS 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.</p> <p>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.</p> <p>O.E.I&amp;A - this objective builds on the O.E.ACCESS objective by requiring users to identify and authenticate themselves to the IT Environment before attempting to modify TSF data or security attributes managed by the IT Environment.</p> <p>O.E.ADMIN_ROLES – this objective supports the O.E.ACCESS objective by requiring authorized roles for users to perform administrative procedures therefore, isolating the amount of damage a user can perform.</p>

Threat/Assumption	Security Objective Rationale (TOE, IT Environment and Non-IT Environment)
T.USER_CORRUPT	<p>O.DAC_ACC – this objective addresses this threat by defining a DAC SFP that defines and enforces access rules for user data managed by the TOE based on subjects, objects, subject security attributes, object security attributes and operations.</p> <p>O.ACCESS – this objective supports the O.DAC_ACC objective by enforcing a DAC SFP that defines and enforces restrictive access and modification rules for security attributes and TSF Data managed by the TOE .</p> <p>O.I&amp;A - this objective supports the O.ACCESS objective by requiring users to identify and authenticate themselves to the TOE before attempting to modify TSF data and security attributes managed by the TOE.</p> <p>O.ADMIN_ROLES – this objective supports the O.ACCESS by requiring authorized roles for administrators to perform administrative procedures therefore, isolating the amount of damage a user can perform.</p> <p>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.</p> <p>O.E.ACCESS – this objective builds on O.SUBJDATA 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.</p> <p>O.E.I&amp;A - this objective builds on the O.E.ACCESS objective by requiring users to identify and authenticate themselves to the IT Environment before attempting to modify TSF data or security attributes managed by the IT Environment and used by the TOE to enforce the DAC SFP.</p> <p>O.E.ADMIN_ROLES – this objective supports the O.E.ACCESS objective by requiring authorized roles for users to perform administrative procedures therefore, isolating the amount of damage a user can perform.</p>
A.CONNECT	O.N.PHYSICAL – this objective provides for the physical protection of the TOE.
A.COOP	O.N.CREDEN - this objective provides for the physical protection of the TOE's authentication data.
A.LOCATE	O.N.PHYSICAL – this objective provides for the physical protection of the TOE.
A.MANAGE	O.N.INSTALL - this objective ensures that the TOE will be managed appropriately.
A.NO_EVIL_ADM	O.N.INSTALL - this objective ensures that the TOE will be managed appropriately.
A.PEER	O.N.INSTALL - this objective ensures that the TOE will be managed appropriately.
A.PROTECT	O.N.PHYSICAL – this objective provides for the physical protection of the TOE.

## 8.2 Security Functional Requirements Rationale

### 8.2.1 Rationale for Security Functional Requirements of the TOE Objectives

This section provides 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, the TOE security objective(s) identified in Section 4.1 that address it. Table 22 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.ACCESS	O.ADMIN_ROLES	O.DAC_ACC	O.ENFORCE	O.I&A	O.MANAGE
FDP_ACC.1(1)			X			
FDP_ACF.1(1)			X			
FIA_ATD.1(1)			X			
FIA_ATD.1(2)			X			
FIA_UAU.2(1)					X	
FIA_UID.2(1)					X	
FIA_USB.1			X			
FMT_MSA.1(1)	X					X
FMT_MSA.1(2)	X					X
FMT_MSA.3(1)			X			
FMT_MTD.1(1)	X					X
FMT_MTD.1(2)	X					X
FMT_MTD.1(3)	X					X
FMT_MTD.1(4)	X					X
FMT_MTD.1(5)	X					X
FMT_SMF.1						X
FMT_SMR.1(1)		X				
FPT_RVM.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.ACCESS		<p>FMT_MSA.1(1) – Defines the restrictions to modify a client’s UNIX-Style security attributes (UNIX User UID and Primary UNIX User GID) managed by the TOE and used to enforce the DAC SFP.</p> <p>FMT_MSA.1(2) – Defines the restrictions to modify a client’s UNIX-Style security attributes (Secondary UNIX User GIDs) managed by the TOE and used to enforce the DAC SFP.</p> <p>FMT_MTD.1(1) – Defines the restrictions to modify UNIX usernames (/etc/passwd) managed by the TOE and used to enforce the DAC SFP and I&amp;A.</p> <p>FMT_MTD.1(2) – Defines the restrictions to modify UNIX passwords (/etc/passwd) managed by the TOE and used to enforce the I&amp;A.</p> <p>FMT_MTD.1(3) – Defines the restrictions to modify UNIX username and NT username mappings (/etc/usermap.cfg) managed by the TOE and used to enforce the DAC SFP</p> <p>FMT_MTD.1(4) – Defines the restrictions to modify the default UNIX username (wafl.default_unix_user) managed by the TOE and used to enforce the DAC SFP.</p> <p>FMT_MTD.1(5) – Defines the restrictions enforced to modify the default NT username (wafl.default_nt_user) managed by the TOE and used to enforce the DAC SFP.</p>
O.ADMIN_ROLES		FMT_SMR.1(1) – Defines the user roles implemented by the DAC SFP requiring authorized roles for administrators to perform administrative procedures.
O.DAC_ACC	DAC Subjects	<p>FDP_ACC.1(1) – Identifies the subjects covered by the DAC SFP.</p> <p>FDP_ACF.1(1) – Identifies the subject security attributes used to enforce the DAC SFP.</p> <p>FIA_ATD.1(1) – Identifies the TOE maintained subject security attributes (UNIX User UID and Primary UNIX User GID) used to enforce the DAC SFP.</p> <p>FIA_ATD.1(2) – Identifies the TOE maintained subject security attributes (Secondary UNIX User GID) used to enforce the DAC SFP.</p> <p>FIA_USB.1 – Ensures that the appropriate security attributes are used for the subjects (client processes) acting on behalf of the users.</p>
	DAC Objects	<p>FDP_ACC.1(1) – Identifies the objects covered by the DAC SFP.</p> <p>FDP_ACF.1(1) – Identifies the object security attributes used to enforce the DAC SFP.</p> <p>FMT_MSA.3(1) – Ensures restrictive default values are defined for the TOE’s object security attributes used to enforce the DAC SFP.</p>

Objective (TOE)	TOE Security Objectives Rationale	
	Note	Rationale
	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		<p>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.</p> <p>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.</p>
O.I&A		<p>FIA_UID.2(1) – Ensures that users must identify themselves before any TSF mediated access to the TOE functions or TSF data is allowed.</p> <p>FIA_UAU.2(1) – Ensures that users must authenticate themselves before any TSF mediated access to the TOE functions or TSF data is allowed.</p>
O.MANAGE		<p>FMT_SMF.1 – Defines the TSF management functions provided by the TOE that ensures the TOE’s SFPs can be enforced.</p> <p>FMT_MSA.1(1) Only authorized administrators responsible for the management of TOE security may modify, delete or add the UNIX-style security attributes (UNIX User UID and Primary UNIX User GID) maintained locally by the TOE and used to enforce the DAC SFP.</p> <p>FMT_MSA.1(2) Only authorized administrators responsible for the management of TOE security may modify, delete or add the UNIX-style security attributes (Secondary UNIX User GIDs) maintained locally by the TOE and used to enforce the DAC SFP.</p> <p>FMT_MTD.1(1) – Defines the restrictions enforced by the DAC SFP to modify UNIX usernames (/etc/passwd) managed by the TOE and used to enforce the DAC SFP and I&amp;A.</p> <p>FMT_MTD.1(2) – Defines the restrictions enforced by the DAC SFP to modify UNIX passwords (/etc/passwd) managed by the TOE and used to enforce the I&amp;A.</p> <p>FMT_MTD.1(3) – Defines the restrictions enforced by the DAC SFP to modify UNIX username and NT username mappings (/etc/usermap.cfg) managed by the TOE and used to enforce the DAC SFP</p> <p>FMT_MTD.1(4) – Defines the restrictions enforced by the DAC SFP to modify the default UNIX username (waf.default_unix_user) managed by the TOE and used to enforce the DAC SFP.</p> <p>FMT_MTD.1(5) – Defines the restrictions enforced by the DAC SFP to modify the default NT username (waf.default_nt_user) managed by the TOE and used to enforce the DAC SFP</p>

## 8.2.2 Rationale for Security Functional Requirements of the IT Environment

This section provides 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. Table 23 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 24 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
FIA_ATD.1(3)					X
FIA_ATD.1(4)					X
FIA_ATD.1(5)					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_SMR.1(2)		X			
FPT_RVM_HW_EXP.1			X		

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
FPT_SEP_HW_EXP.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	<p>FDP_ACC.1(2) and FDP_ACF.1(2) – Defines the IT Environment UAC SFP to access TSF data managed by the IT Environment</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>FMT_MSA.1(6) – Defines the restrictions enforced by the IT Environment’s UAC SFP to modify NT User SIDs and NT User GIDs maintained by the IT Environment’s Domain Controller and used by the TOE to enforce the DAC SFP.</p> <p>FMT_MSA.3(2) – Defines the restrictions that the IT Administrator is only able to set default values or modify the default values for the files containing the TSF data.</p> <p>FMT_MTD.1(6) – Defines the restrictions enforced by the IT Environment’s UAC SFP to modify UNIX usernames (NFS or LDAP) managed by the IT Environment and used to enforce the DAC SFP.</p> <p>FMT_MTD.1(7) – Defines the restrictions enforced by the IT Environment’s UAC SFP to modify NT usernames managed by the IT Environment’s CIFS Server and used to enforce the DAC SFP.</p>

Objective (IT Environment)	IT Environment Security Objectives Rationale
	FMT_MTD.1(8) – Defines the restrictions enforced by the IT Environment’s UAC SFP to modify NT usernames (from Domain Controller) managed by the IT Environment’s Domain Controller and used to enforce the DAC SFP.
O.E.ADMIN_ROLES	FMT_SMR.1(2) – Defines the user roles implemented by the IT Environment’s UAC SFP requiring authorized roles for administrators to perform administrative procedures.
O.E.ENFORCE	<p>FPT_RVM_HW_EXP.1 – Ensures that the TOE cannot be bypassed, therefore allowing the TOE to perform its policy enforcement.</p> <p>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.</p>
O.E.I&A	<p>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.</p> <p>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.</p>
O.E.SUBJECTDATA	<p>FIA_ATD.1(3) - 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.</p> <p>FIA_ATD.1(4) - Identifies the subject security attributes (Secondary UNIX User GIDs) maintained by the IT environment and used by the TOE to enforce the DAC SFP.</p> <p>FIA_ATD.1(5) - Identifies the subject security attributes (NT User SID and NT User GID) maintained by the IT environment and used by the TOE to enforce the DAC SFP.</p>

### 8.3 TOE Summary Specification Rationale

This section demonstrates that the TOE’s Security Functional Requirements identified in Section 5.1 completely and accurately meet the TOE’s two Security Functions identified in Sections 6.1. Table 25 demonstrates the correspondence between the two Security Functions and the TOE Security Functional Requirements. The subsequent sections describe the rationale proving that the Security Functional Requirements provide the functionality of the Security Functions. The table is also shows how the Security Functional Requirements meet different areas of the two Security Functions.



**Table 25 - Security Functional Requirements to Security Functions Mapping**

Security Functional Requirements (TOE)	Security Functions		
	ADMIN	DAC	Self Protection
FDP_ACC.1(1)		X	
FDP_ACF.1(1)		X	
FIA_ATD.1(1)		X	
FIA_ATD.1(2)		X	
FIA_UAU.2(1)	X		
FIA_UID.2(1)	X		
FIA_USB.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 26 - Security Functional Requirements to Security Functions Rationale**

Security Functional Requirements (TOE)	Security Function Rationale
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_ATD.1(1)	DAC – The TOE maintains the client security attributes used to enforce file access rules.

Security Functional Requirements (TOE)	Security Function Rationale
FIA_ATD.1(2)	DAC - The TOE maintains the client security attributes used to enforce file access rules.
FIA_UAU.2(1)	ADMIN – The TOE requires administrators to authenticate themselves before allowing any access to any TSF mediated actions.
FIA_UID.2(1)	ADMIN - The TOE requires administrators to identify themselves before allowing any access to any TSF mediated actions.
FIA_USB.1	DAC – The TOE associates users with the security attributes used to enforce the DAC SFP.
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 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 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 administrators to create and modify UNIX username used to support the DAC SFP. The TOE restricts access to UNIX usernames managed by the TOE to users with an administrator role.
FMT_MTD.1(2)	ADMIN - The TOE provides administrative support that enables administrators to create and modify UNIX user passwords used to support the DAC SFP. The TOE restricts access to UNIX passwords managed by the TOE to users with an administrator role.
FMT_MTD.1(3)	ADMIN - The TOE provides administrative support that enables administrators to create and modify NT Username and UNIX Username mappings used to support the DAC SFP. The TOE restricts access to NT Username/UNIX Usernames managed by the TOE to users with an administrator role.

Security Functional Requirements (TOE)	Security Function Rationale
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 file. Access to wafl.default_unix_user file is limited by the TOE to administrators.
FMT_MTD.1(5)	ADMIN - The TOE provides administration functionality that enables a user to modify the NT Username stored in wafl.default_nt_user file. Access to wafl.default_nt_user file is limited by the TOE to administrators.
FMT_SMF.1	ADMIN - The TOE provides an CLI management interface that enables users to modify TSF Data used to enforce the TOE's SFPs. Access to the CLI requires users to identify and authenticate themselves to the TOE.
FMT_SMR.1(1)	ADMIN - The TOE supports two user roles: administrator and non-administrator. The TOE uses these two 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 each are each are hierarchical to and dependent upon and 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. Table 27 lists the TOE Security Functional Components and the Security Functional Components each are hierarchical to and dependent upon and any necessary rationale. Table 28 lists the IT Environment Security Functional Components and the Security Functional Components each 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
FDP_ACC.1	No Components	FDP_ACF.1	Satisfied
FDP_ACF.1	No Components	FDP_ACC.1, FMT_MSA.3	Satisfied Satisfied
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 TOE therefore, this dependency is satisfied.
FIA_UID.2	FIA_UID.1	No dependencies	N/A
FIA_USB.1	No Components	FIA_ATD.1	Satisfied
FMT_MSA.1	No Components	[FDP_ACC.1 or FDP_IFC.1], FMT_SMR.1	Satisfied N/A Satisfied
FMT_MSA.3	No Components	FMT_MSA.1, FMT_SMR.1	Satisfied Satisfied
FMT_MTD.1	No components.	FMT_SMR.1	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.1	No Components	No dependencies	N/A
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_SMR.1	Satisfied N/A Satisfied
FMT_MSA.3	No components.	FMT_MSA.1, FMT_SMR.1	Satisfied Satisfied
FMT_MTD.1	No components.	FMT_SMR.1	Satisfied
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_SEP.1	No Components	No dependencies	N/A
FPT_RVM_HW_EXP.1	No Components	No dependencies	N/A
FPT_SEP_HW_EXP.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.

Table 29 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 7.0.3 and 7.0.4 Ontap703filelist.txt Ontap704filelist.txt
Delivery and Operation	ADO_DEL.1	Delivery and Operations
	ADO_IGS.1	Installation, Generation and Start Up Procedures
Development	ADV_FSP.1	DataONTAP 7.0.3 and 7.0.4 Functional Specification
	ADV_HLD.1	DataONTAP 7.0.3 and 7.0.4 High Level Design
	ADV_RCR.1	DataONTAP 7.0.3 and 7.0.4 Correspondence Mapping
Guidance Documents	AGD_ADM.1	Administrator and User guidance for DataONTAP Common Criteria deployments for DataONTAP 7.0.3 and 7.0.4  Net App Man Pages

Assurance Class	Component ID	Documentation Satisfying Component
	AGD_USR.1	Administrator and User guidance for DataONTAP Common Criteria deployments for DataONTAP 7.0.3 and 7.0.4  Net App Man Pages
Tests	ATE_COV.1	DataONTAP 7.0.3 Test Documentation for Common Criteria EAL2 Evaluation  DataONTAP 7.0.4 Test Documentation for Common Criteria EAL2 Evaluation
	ATE_FUN.1	DataONTAP 7.0.3 Test Documentation for Common Criteria EAL2 Evaluation  DataONTAP 7.0.4 Test Documentation for Common Criteria EAL2 Evaluation
	ATE_IND.2	DataONTAP 7.0.3 Test Documentation for Common Criteria EAL2 Evaluation  DataONTAP 7.0.4 Test Documentation for Common Criteria EAL2 Evaluation
Vulnerability Assessment	AVA_SOF.1	Strength of Function Analysis for Common Criteria EAL2, Data ONTAP 7.0.3 and 7.0.4
	AVA_VLA.1	Developer Vulnerability Analysis Data ONTAP 7.0.3 and 7.0.4 Common Criteria