



EMC[®] Data Domain[®] BoostFS

Version 1.0

Configuration Guide

302-003-051

REV. 01

Copyright © 2016 EMC Corporation. All rights reserved. Published in the USA.

Published September 2016

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (<https://support.emc.com>).

EMC Corporation
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.EMC.com

CONTENTS

Chapter 1	Introduction to BoostFS	5
	Introduction to BoostFS.....	6
	Supported Environments.....	6
	Supported Applications.....	6
Chapter 2	Preparing the Data Domain System for BoostFS	9
	Preparing the Data Domain System for BoostFS.....	10
	BoostFS and Existing Data Domain Commands.....	11
	Assigning Multiple Users to BoostFS.....	11
	Creating Storage Units.....	12
	Logical Stream Limits for Storage Units (Optional).....	13
	Client Groups and BoostFS.....	13
	Distributed Segment Processing Option.....	13
Chapter 3	Installing and Configuring BoostFS	15
	Installation Overview.....	16
	Components of the BoostFS Client.....	16
	BoostFS on Linux Systems	16
	The Role of FUSE in BoostFS.....	17
	Authentication Methods.....	17
	RSA Lockbox-Based Authentication.....	17
	Kerberos-Based Authentication.....	17
	BoostFS Command Overview.....	19
	boostfs mount.....	19
	boostfs lockbox.....	20
	boostfs kerberos.....	20
	The BoostFS Configuration File.....	21
	Configuring a BoostFS client and DD system for Kerberos.....	22
	Preparing the DD System for Kerberos.....	22
	Setting the host name and domain name for the DD system.....	22
	Configuring BoostFS for a Windows Active Directory environment.....	23
	Configuring BoostFS for a UNIX KDC environment.....	24
Chapter 4	Troubleshooting	27
	Log information.....	28
	Common issues.....	28
Chapter 5	Appendix	31
	About Puppet.....	32
	References.....	32

CONTENTS

CHAPTER 1

Introduction to BoostFS

- [Introduction to BoostFS](#)..... 6
- [Supported Environments](#)..... 6
- [Supported Applications](#)..... 6

Introduction to BoostFS

Data Domain Boost Filesystem (BoostFS) 1.0 provides a general file-system interface to the DD Boost library, allowing standard backup applications to take advantage of DD Boost features.

Advantages of BoostFS

By leveraging the DD Boost technology, BoostFS helps reduce bandwidth, can improve backup-times, offers load-balancing, allows in-flight encryption, and supports the Data Domain multi-tenancy feature set.

As a file server system implementation, the BoostFS workflow is similar to NFS but also leverages the DD Boost protocol. In addition, BoostFS improves backup times compared to NFS and various copy-based solutions.

BoostFS supports single-node Data Domain systems, Extended Retention systems, Data Domain Virtual Edition, and Extended Distance Protection.

Purpose

This document describes how to install and configure BoostFS on client systems.

Terminology

Term	Definition
FUSE	Filesystem in User Space (FUSE) is an open-source interface that enables non-privileged users to securely create and mount their own file-system implementations.
Puppet	An open-source configuration management tool.
Push	A process that involves using a centralized server to connect to specified clients and run commands remotely; for BoostFS, that means downloading and remotely installing the installation package on each client.

Supported Environments

Environments that use BoostFS must meet the following specifications.

- Data Domain Operating System version 6.0
- One of the following Linux distributions:
 - Red Hat Enterprise Linux versions 6 and 7
 - SUSE Linux Enterprise Server versions 11 and 12
 - Ubuntu 14.04 and 15
- FUSE 2.8 or greater

Supported Applications

BoostFS currently supports the following applications:

- CommVault Simpana versions 9, 10, and 11
- MySQL Community 5.6. and 5.7
- MySQL Enterprise Manager 5.6 and 5.7
- MongoDB Community 2.6, 3.0, and 3.2

Boost Features Not Supported by BoostFS

- Managed File Replication (MFR)
 - DD Boost-over-Fibre Channel (DFC)
 - Transport Layer Security (TLS) one-way and two-way authentication.
TLS anonymous encryption is supported.
-

Note

If you select TLS, be aware that there is no configuration option to enable TLS from the client. It must be enabled through the Data Domain System.

CHAPTER 2

Preparing the Data Domain System for BoostFS

- [Preparing the Data Domain System for BoostFS](#)..... 10
- [BoostFS and Existing Data Domain Commands](#).....11
- [Assigning Multiple Users to BoostFS](#)..... 11
- [Creating Storage Units](#).....12
- [Logical Stream Limits for Storage Units \(Optional\)](#) 13
- [Client Groups and BoostFS](#)..... 13
- [Distributed Segment Processing Option](#)..... 13

Preparing the Data Domain System for BoostFS

Every Data Domain system that is enabled for Data Domain Boost deduplication must have a unique name. You can use the DNS name of the Data Domain system, which is always unique.

Procedure

1. On the Data Domain system, log in as an administrative user.
2. Verify that the file system is enabled and running by entering:

```
# fileysys status
The file system is enabled and running.
```

3. Verify DD Boost is already enabled:

```
# ddbboost status
DD Boost status: enabled
```

If the DD Boost status is reported as disabled, enable it by entering:

```
# ddbboost enable
DD Boost enabled
```

4. Verify distributed segment processing is enabled:

```
ddbboost option show
```

You should see the following output:

Option	Value
distributed-segment-processing	enabled
virtual-synthetics	enabled
fc	disabled

If distributed segment processing is shown as disabled, enable it by entering:

```
ddbboost option set distributed-segment-processing enabled
```

Note

- If SMT is used, the user role must be set as `none`.
- Users who run backup applications that connect to Data Domain systems must have their user names configured on the Data Domain system. For more information, refer to the *EMC Data Domain Operating System Administration Guide*.
- Multiple applications can use DD Boost to access a Data Domain system, and multiple users can be configured for DD Boost access. The username, password, and role must have already been set up on the Data Domain system using the DD OS `user add` command:

```
user add <user> [password <password>]
[role {admin | security | user | backup-operator | data-access}]
[min-days-between-change <days>] [max-days-between-change <days>]
[warn-days-before-expire <days>] [disable-days-after-expire <days>]
[disable-date <date>]
```

For example, to add a user with a login name of `jsmith` and a password of `mP34$muk*E` with administrative privilege, enter:

```
# user add jsmith password mP34$muk*E role admin
```

Once the user has been created on the Data Domain system, the user must be made a DD Boost user. To add `jsmith` to the DD Boost user list, enter:

```
# ddbboost user assign jsmith
```

BoostFS and Existing Data Domain Commands

You must create one or more storage units on each Data Domain system enabled for BoostFS. Data Domain administrators can use existing DD OS CLI commands to create and manage storage units used by BoostFS.

Assigning Multiple Users to BoostFS

When, as a system administrator, you create the storage units that users employ with the backup applications, you associate a username with each storage unit. This associated username can be changed after creation of the storage unit.

Storage units are accessible only to applications with the username that owns the storage unit.

Each storage unit is owned by one username, and the same username can own multiple storage units. The application passes the username and password to BoostFS, and DD Boost passes them to the Data Domain system when attempting to connect to the Data Domain system. The Data Domain system then authenticates the username and password. The username and password can be shared by different applications.

When a storage unit is created with a valid Data Domain system local user but not assigned to DD Boost, the user is automatically added to the DD Boost users list in the same way that a user is added via the `ddbboost user assign` command.

Assign one or more users to the DD Boost users list:

```
# ddbboost user assign user1 user2
User "user1" assigned to DD Boost.
User "user2" assigned to DD Boost.
```

To verify and display the users in the users list, enter:

```
# ddbboost user show
```

DD Boost user	Default tenant-unit	Using Token Access
user1	Unknown	Yes
user2	Unknown	-
user3	Unknown	Yes
user4	Unknown	-
user5	Unknown	-
user6	Unknown	-
user7	Unknown	Yes
user8	Unknown	-

To unassign the user from the users list, enter:

```
# ddbboost user unassign user1
User "user1" unassigned from DD Boost.
```

Creating Storage Units

You need to create one or more storage units on each Data Domain system enabled for BoostFS.

Procedure

1. Create a storage unit on the Data Domain system:

```
# ddbboost storage-unit create NEW_STU1 user user1
Created storage-unit "NEW_STU1" for "user1".
```

A storage unit name must be unique on any given Data Domain system. However, the same storage unit name can be used on different Data Domain systems.

The username owns the storage unit and ensures that only connections with this username's credentials are able to access this storage unit. See the section on `ddbboost storage-unit` commands in the *EMC Data Domain Operating System Command Reference Guide* for details on command options.

2. Repeat the previous step for each storage-unit needed on the Data Domain system.
3. If you want to modify a storage unit on the Data Domain system, enter:

```
# ddbboost storage-unit modify NEW_STU1 user user2
Storage-unit "NEW_STU1" modified for user "user2".
```

The `ddbboost storage-unit modify` command allows the backup application to change the username ownership of the storage unit. Changing the username does not require that attributes of every file on the storage unit be changed.

4. Display the users list for the storage units:

```
# ddbboost storage-unit show
```

After entering the command, the output you see should be similar to the following:

```
# ddbboost storage-unit show
Name                Pre-Comp (GiB)  Status  User          Report Physical
                   -----
                   -----
                   -----
                   -----
                   -----
backup              3.0            RW      sysadmin      -
DDBOOST_STRESS_SU  60.0           RW      sysadmin      -
task2               0.0            RW      sysadmin      -
tasking1            0.0            RW      sysadmin      -
DD1                 0.0            RW      sysadmin      -
D6                  5.0            RW      sysadmin      -
TEST_DEST           0.0            D       sysadmin      -
STU-NEW             0.0            D       ddul          -
getevent            0.0            RW      ddul          -
DDP-5-7             120.0          RW      sysadmin      -
TESTME              150.0          RW      sysadmin      -
DDP-5-7-F           100.0          RW      sysadmin      -
testSU              0.0            RW      sysadmin      200
-----
D      : Deleted
Q      : Quota Defined
RO     : Read Only
RW     : Read Write
RD     : Replication Destination
```

Logical Stream Limits for Storage Units (Optional)

BoostFS supports the same stream limit and storage quota features as DD Boost. See the *EMC DD Boost for Partner Integration Administration Guide* for more information.

Note

If you are using BoostFS and you have a MySQL backup that is larger than your hard quota allows, MySQL Enterprise Backup might hang instead of exiting with an error message. When you use MySQL Enterprise Backup, you should monitor the hard limit to ensure the quota is not exceeded, which can cause an application error.

Client Groups and BoostFS

The Client Group feature identifies specific client loads when clients are associated with groups.

The `client group` command set is supported only for clients that use DD Boost or NFS protocols. For more information about Client Groups, see the *EMC Data Domain Operating System Command Reference Guide*.

Distributed Segment Processing Option

The distributed segment processing option is configured on the Data Domain system and applies to all the backup servers and all the client applications accessing the Data Domain system.

Note

Enabling or disabling the distributed segment processing option does not require a restart of the Data Domain file system.

CHAPTER 3

Installing and Configuring BoostFS

- [Installation Overview](#) 16
- [Components of the BoostFS Client](#)..... 16
- [The Role of FUSE in BoostFS](#)..... 17
- [Authentication Methods](#)..... 17
- [BoostFS Command Overview](#)..... 19
- [The BoostFS Configuration File](#)..... 21
- [Configuring a BoostFS client and DD system for Kerberos](#)..... 22

Installation Overview

There is a single RPM installation package for BoostFS that both enterprise and small-scale users can download. The BoostFS package contains all the files required to start the BoostFS process.

The RPM package includes the `boostfs` executable and RSA Lockbox.

Check the following before beginning the process:

- The FUSE version on the client must be 2.8 or higher

While the BoostFS process is running:

- You cannot upgrade BoostFS
- You cannot uninstall BoostFS

Components of the BoostFS Client

The BoostFS client is composed of the following:

- A daemon process that supports various commands
- Two shared libraries: `libDDBoost.so` and `libDDBoostFS.so`
- `.rsalib`: A hidden directory that contains redistributable RSA libraries
- A configuration file
- A manual page

`libDDBoost.so`, a FUSE-agnostic library built on the DD Boost library, provides such services as connection management, a retry mechanism, and client logging.

The packaging defaults to the Red Hat Package Manager (RPM) format, but the native packaging for other operating systems is also supported.

BoostFS on Linux Systems

Employing the Linux Filesystem Hierarchy Standard 3.0, the BoostFS client is installed in `/opt/emc/boostfs` and contains the following subdirectories:

- `bin`: `boostfs` command(s) are installed here.
- `lib`: Contains these libraries.
 - `libDDBoost.so`
 - `libDDBoostFS.so`
- `.rsalib`: Contains redistributable RSA libs.
- `etc`: Contains configuration files (sample and production).
- `man`: Contains standard man pages.

The actual installation package is offered in the Red Hat Package Manager (RPM) format.

The Role of FUSE in BoostFS

BoostFS uses FUSE, an open-source software interface that enables non-privileged users to securely create and mount their own file-system implementations.

FUSE allows you to export a virtual file system to the Linux kernel. Write operations through BoostFS and FUSE benefit from Data Domain's distributed segment processing.

Using FUSE and the DD Boost plug-in, BoostFS exports a storage unit on a Data Domain system to a mountpoint on a client. On the client, file system operations conducted on the mountpoint are captured by the kernel before being passed through FUSE to BoostFS.

BoostFS runs as a daemon on a client. As a software module, BoostFS serves as a layer between FUSE and DD Boost.

BoostFS is supported on some Linux systems in the initial release.

FUSE consists of three parts:

- A kernel module: `fuse.ko`
- A user space library: `libfuse`
- A mount utility: `fusermount`

Note

BoostFS requires the "user_allow_other" option for FUSE; it will add the option to the `/etc/fuse.conf` file if it is not already present. Be aware that this may change the behavior of other FUSE-based applications you are using.

Authentication Methods

BoostFS has two authentication options:

- RSA Lockbox (default)
- Kerberos

RSA Lockbox-Based Authentication

RSA Lockbox is the default password manager.

If you choose RSA Lockbox, you need to set the lockbox using the `boostfs lockbox set` command.

Kerberos-Based Authentication

BoostFS supports Kerberos-based authentication, which offers some important advantages.

Note

Before using Kerberos for BoostFS, you should verify that the Kerberos client libraries for your Linux distribution are installed on your machine.

These advantages include the following:

- Single sign-on capability

- Passwords never exchanged over the wire between client and server
- Automatic renewal of Kerberos tickets with strong encryption and configuration support

Note

There are tools such as `kinit -R` that administrators need to run as part of a Cron job to ensure the Kerberos credentials are renewed. This is part of a standard Kerberos configuration.

There are three main entities involved with Kerberos Authentication:

- BoostFS client
- Kerberos Key Distribution Center (KDC), which can be either one of the following:
 - An active directory on a domain controller in a Windows environment
 - A UNIX-based KDC
- Data Domain Restorer (DDR)

Firewall configurations

If you are creating a firewall configuration, you should use the BoostFS TCP Port 2049. Kerberos networking ports must also be added to the firewall.

Supported topologies

The following topologies support Kerberos:

- Data Domain systems running DD OS version 6.0 and later
- All Linux clients that run the `boostfs` executable
- KDC/Active Directory (AD) on a Windows 2003, 2008, or 2012 server
- KDC on a POSIX-based operating system with optional NIS lookups

The BoostFS client and Kerberos authentication

The BoostFS client uses the General Security Services (GSS) API to negotiate Kerberos authentication with the Data Domain system. GSS-API provides a generic interface which can be layered atop different security mechanisms, meaning that communicating peers that have GSS-API credentials for the same security mechanism can establish a security context.

To communicate using Kerberos authentication, the BoostFS client must meet the following conditions:

- Be configured as a Kerberos client for a specified domain or realm
- Be capable of acquiring Kerberos credentials for a specified node and access to the desired services

Kerberos tickets

You need two types of tickets to access different services through Kerberos:

- A Kerberos Ticket Granting Ticket (TGT)
- A Kerberos ticket for various services (service tickets) that the client will use (BoostFS, DNS, CIFS, NFS, etc.)

BoostFS Command Overview

You use the `boostfs` command to establish the FUSE mount, create the lockbox (if desired), and set up Kerberos credentials if you choose Kerberos as the authentication method.

boostfs mount

The `boostfs mount` command allows you to establish the BoostFS FUSE mount.

```
boostfs mount [-d|--data-domain-system] <data-domain-system>
[-s|--storage-unit] <storage-unit>
[[-o|--option <param>=<value>] ...] <mount-point>
```

Mount the BoostFS file system. Role required: none.

```
boostfs umount <mount-point>
```

Unmount the BoostFS file system.

Other mount options

You can use these `boostfs mount` options when configuring BoostFS through the Command Line Interface (CLI):

Option	Description
<code>-o security=<krb5 lockbox></code>	Security option used for authentication (default: lockbox).
<code>-o log-enabled=<true false></code>	Enable logging (default: true).
<code>-o log-level=<debug info warning error></code>	Log level (default: info).
<code>-o log-dir=/path/to/log</code>	Directory for log files (default: <code>/opt/emc/boostfs/log</code>).
<code>-o log-file=output.log</code>	Log file name (default: <code>ddbboostfs_<uid>_<gid>.log</code>).
<code>-o log-maxsize=100</code>	Maximum log size in MB (default: 100 MB).
<code>-o log-rotate-num=8</code>	Number of log files to save (default: 8).
<code>-o app-info="text_string"</code>	Text string that describes the application using boostfs with additional information such as the version (default: FUSE version). See the following note; for support purposes this setting is required

Note

The `app-info` string is used by the Data Domain system to help identify the application that is using BoostFS.

For support purposes, the `app-info` must be configured for systems operating in the production environment. This can be done by either setting it in the BoostFS configuration file or using `-o app-info=text_string` on the mount command line.

This string should represent the backup software being used and the corresponding version; for example, `Commvault_v11`, `MySQL_5.7`, etc.

boostfs lockbox

The `boostfs lockbox` command allows you to set, query, and remove the RSA lockbox values.

```
boostfs lockbox set [-d | --data-domain-system] <data-domain-system>
[-u | --storage-unit-username] <storage-unit-username>
[-s | --storage-unit] <storage-unit>
```

To store credentials in an RSA lockbox, the user specifies the Data Domain hostname, the storage-unit name, and the storage-unit user. After providing that information, the user is prompted for the password.

```
boostfs lockbox query [-d | --data-domain-system] <data-domain-system>
[-s | storage-unit] <storage-unit>
```

If the credentials have been stored in an RSA lockbox, this command returns the username after the query is submitted with the specified Data Domain hostname and storage-unit.

```
boostfs lockbox remove [-d | --data-domain-system] <data-domain-system>
[-s | storage-unit] <storage-unit>
```

Removes the stored RSA lockbox credentials in the specified Data Domain system and storage-unit.

boostfs kerberos

The `boostfs kerberos` command allows you to add, verify, and remove Kerberos credentials.

```
boostfs kerberos set [-u|--storage-unit-username] <storage-unit-username>
[-s|--storage-unit] <storage-unit>
[[-r|--kerberos-realm] <kerberos-realm>]
[[-m|--kerberos-username] <kerberos-username>]
```

Gets the Ticket Granting Ticket (TGT) for the storage-unit user.

```
boostfs kerberos query [[-u|--storage-unit-username] <storage-unit-username>
[-s|--storage-unit] <storage-unit>]
[[-m|--kerberos-username] <kerberos-username>]
```

Checks for Kerberos credentials.

```
boostfs kerberos remove [-s|--storage-unit] <storage-unit>
```

Removes Kerberos credentials.

The BoostFS Configuration File

The Boost Filesystem has two configuration options.

- Command-line interface (CLI)
- The configuration file: `boostfs.conf`

This file is located in `/opt/emc/boostfs/etc`, and can be edited by the "root" user or someone with sudo privileges.

Parameters can be specified either in the config file or on the command line, or both.

The configuration file has a global section and a mount-point specific section. Configuration parameters configured using the command line take the highest priority and override any values in the config file. The config file has mount-specific and global specific sections. Mount specific parameter values override global parameter values.

The following is a sample configuration file:

```
#####
# BoostFS 1.0 example input file
#
#
# The configuration file is divided into sections, delineated by brackets [].
# Options that are to apply to all mount points are in the [global] section.
# More details on the various configuration options can be found in the
# BoostFS manual. Command line options override what is in this file.
#
# Format:
# # - Identifies a comment line, and must be at the start. Configuration
# parameters can be disabled by adding a "#" to the start of the line.
#
# Values which contains spaces should use double quotations around the
# entire value.
#
# No whitespace is allowed between the option and the value, i.e.
# log-dir = /path is not allowed.
#
# Comments are not allowed after the option value pair.
#
#####

# [global]
# Data Domain Hostname or IP address
# data-domain-system=dd2500-1.yourdomain.com

# Storage Unit
# storage-unit=su-name

# Storage Unit Username
# storage-unit-username=sysadmin

# Subdirectory within the storage-unit to mount to
# directory-name=path/to/subdir

# Security option used for authentication (default: lockbox)
# security=<krb5|lockbox>

# Enable logging (default: true)
# log-enabled=<true|false>

# Log level (default: info)
# log-level=<debug|info|warning|error>

# Directory for log files (default: /opt/emc/boostfs/log)
# log-dir=/path/to/log
```

```
# Log file name (default: /opt/emc/boostfs/log/boostfs.log)
# log-file=output.log

# Maximum log size in MB (default: 100MB)
# log-maxsize=100

# Number of log files to save (default: 8)
# log-rotate-num=10

# Text string describing the application using boostfs (version, etc)
# app-info="text string"

# Allow users other than the owner of the mount to access the mount
# allow-others=<true|false>

#
# Mount point sections are delineated by [mountpoint]
#
# [/path/to/mount]
# Data Domain Hostname or IP address
# data-domain-system=dd2500-1.yourdomain.com

# Storage Unit
# storage-unit=su-name

# Storage Unit Username
# storage-unit-username=sysadmin

# Subdirectory within the storage-unit to mount to
# directory-name=path/to/subdir
```

Configuring a BoostFS client and DD system for Kerberos

You follow a specific sequence of steps to configure a BoostFS client and a Data Domain system to use Kerberos authentication.

Preparing the DD System for Kerberos

If you choose Kerberos as the authentication method, you should ensure that all your systems can access the Key Distribution Center (KDC).

If the systems cannot reach the KDC, check the domain name system (DNS) settings at `/etc/resolv.conf`.

Check the following to ensure your system is ready for Kerberos:

- Verify the KDC can ping both the Data Domain system and the client.
- Verify the client can ping the Data Domain system and the KDC.
- Verify the Data Domain system can ping the Client and the KDC.
- Check the NTP settings in the configuration.
All systems must have the same time stamp (+/- 5 minutes).

Setting the host name and domain name for the DD system

Set the host name and the domain name for the Data Domain system using the `net set` CLI command.

Procedure

1. Enter the following:

```
# net set hostname [host]
# net set {domain name [local-domain-name]}
```

For more information on `net` commands, see the *EMC Data Domain Operating System Command Reference Guide*.

Configuring BoostFS for a Windows Active Directory environment

In this environment, the Windows server that hosts the Active Directory (AD) service also acts as the Key Distribution Center (KDC).

Adding users to the AD KDC Server

Procedure

1. Log in the Windows AD system.
2. In Active Directory Users and Computers, add the user-owner of the storage-unit:

```
<su_username> Type: "User" Password: <user_password>
```

The password must match the user's password on the Data Domain system.

Configuring a Data Domain System for an Active Directory Realm

Procedure

1. Enter the following command on the Data Domain (DD) system: `# authentication kerberos set realm <realm> kdc-type windows`

You should see the following text and be prompted to enter additional information:

```
- Enter domain user: <enter KDC Domain username>
- Enter domain password: <enter KDC Domain password>
```

2. Optionally, configure the client access list for DD Boost on the DD system to use Kerberos authentication: `# ddboost client add <client-name> authentication-mode kerberos`

The configuration process is now complete; the required keytab has been generated and applied to the DD system.

Configuring the BoostFS Client in a Windows AD Environment

Procedure

1. Verify you are pointing to the correct Key Distribution Center (KDC) by checking the `/etc/krb5.conf` file.
2. Enter the `kerberos set` command to provide the Kerberos Ticket Granting Ticket (TGT) for the storage-unit user:

```
# boostfs kerberos set -u <su-username> -s <su-name>
```

3. Verify the Kerberos TGT has been granted for the storage-unit user:

```
# boostfs kerberos query -u <su-username> -s <su-name>
```

The client configuration is complete.

Mount a Storage-Unit Using BoostFS

Procedure

1. Enter the `boostfs mount` command to mount the storage unit:

```
# boostfs mount -d <dd-system> -s <su-name> -o security=krb5 -o
storage-unit-username=<su-username> <mount-point>
```

Note

BoostFS does not support files being executed on the mount point.

Configuring BoostFS for a UNIX KDC environment

In this type of environment, a UNIX server hosts the Key Distribution Center (KDC) service. The Kerberos file contains a "shared secret" (a password, pass phrase, or other unique identifier) between the KDC server and the DD system. When using a UNIX KDC, the DNS server does not have to be the KDC server; it can be a separate server.

For Kerberos, you must transfer a keytab file from the UNIX KDC server, where it is generated, to the Data Domain system. If you are using more than one DD system, you need a separate keytab file for each system. That means additional steps are required to configure multiple DD systems on a UNIX KDC.

Adding Data Domain Principals to the UNIX KDC

Procedure

1. Log in to the Key Distribution Center (KDC).
2. Enter KDC admin mode using the following command: `kadmin`
The commands in the subsequent steps apply to the KDC after entering `kadmin` mode.
3. Add Data Domain (DD) principals to the Key Distribution Center (KDC) using the Kerberos `addprinc` command: `# addprinc boostfs/<ddsystem-hostname>@<realm>`
4. Confirm the client principals have been added by entering the following Kerberos command: `listprincs`
5. Import host and BoostFS credentials to a temporary keytab file on the KDC by entering the Kerberos `ktadd -k` command: `# ktadd -k /tmp/<keytab-file-name-for-ddsystem> boostfs/<ddsystem-hostname>@<realm>`
The keytab file for the DD system is generated and needs to be imported to the Data Domain system.
6. Rename the file to `krb5.keytab` and copy it to `/ddr/var` folder.
7. Copy the keytab file generated in Step 3 from the KDC to the DD system directory `/ddr/var/releases`.

Adding Client Principals to the KDC

Procedure

1. Add the host and BoostFS service principals to the KDC using the Kerberos `addprinc` command:


```
# addprinc boostfs/<client-hostname>@<realm>
# addprinc host/<client-hostname>@<realm>
```

2. Confirm the client principals have been added using the following Kerberos command: **listprincs**
3. Import the host and BoostFS credentials to a temporary keytab file on the KDC by entering the Kerberos `ktadd -k` command:

```
# ktadd -k /tmp/<keytab-file-name-for-client> boostfs/<client-hostname>@<realm>
# ktadd -k /tmp/<keytab-file-name-for-client> host/<client-hostname>@<realm>
```

4. Copy the keytab file generated in Step 3 from the KDC to the client as `/etc/krb5.keytab` file.

Add users to the UNIX KDC server

The following command requires to access the `kadmin` interface on the Key Distribution Center (KDC) server.

Procedure

1. Add the storage-unit user from the Data Domain system to the KDC using the Kerberos `addprinc` command:

```
# addprinc <su-username>@<realm>
```

Configuring Data Domain Systems for the UNIX KDC

Procedure

1. Rename the `keytab_file_for_ddsystem` file located on the `/ddr/var/releases` directory to `krb5.keytab`.
See [Adding Client Principals to the KDC on page 24](#) for information on creating the keytab file for the Data Domain (DD) system.
2. On the DD system, import the keytab file moved in Step 2 to `/ddr/etc` using the following command: **# authentication kerberos keytab import**
3. Confirm the configuration using the `authentication` command: **# authentication kerberos show config**
4. Set the realm on the DD system using the `authentication` command: **# authentication kerberos set realm <realm> kdc-type unix kdcs <KDC-hostname>**
5. Optionally, configure the client access list for DD Boost to use Kerberos authentication: **# ddbboost client add <client-name> authentication-mode kerberos**

Note

If you perform this optional step, note that a BoostFS client configured to use Kerberos must use Kerberos for the connection to succeed. If that BoostFS client uses RSA Lockbox, the connection will fail.

Configuring the BoostFS client

Procedure

1. Check the `/etc/krb5.conf` file to make sure you point to the correct KDC server.
2. Enter the `boostfs kerberos set` command to get the Kerberos Ticket Granting Ticket (TGT) for the storage unit user: `# boostfs kerberos set -u <su-username> -s <su-name>`
3. Verify the Kerberos TGT is granted for the storage-unit user by using the following `boostfs kerberos query` command: `boostfs kerberos query -u <su-username> -s <su-name>`

The client configuration is complete.

Mount a Storage-Unit Using BoostFS

Procedure

1. Enter the `boostfs mount` command to mount the storage unit:

```
# boostfs mount -d <dd-system> -s <su-name> -o security=krb5 -o  
storage-unit-username=<su-username> <mount-point>
```

Note

BoostFS does not support files being executed on the mount point.

CHAPTER 4

Troubleshooting

- [Log information](#)..... 28
- [Common issues](#).....28

Log information

You can use the following logs to diagnose BoostFS problems:

- **BoostFS log file**
By default, the BoostFS log file is found in the directory `/opt/emc/boostfs/log`. The default name of the file is `ddbboostfs_<uid>_<gid>.log`, where:
 - `<uid>`
= user id of boostfs user
 - `<gid>`
= group id of boostfs user

A typical BoostFS log message appears in the following format:

```
Date + Time + ProcSS-ID + Thread-ID + [logging-level: E - error, W - warning, I - info, D - debug] + Message-Text
```

The following is an example information message:

```
May 23 12:53:51 2996 4014012160 [I] bfs_close_open_nodsp: File / 00000004 opened in non-DSP mode
```

- **DD Boost SDK precert log file**
- **DDFS logs**
DDFS logs are found on the Data Domain system in the directory `/ddr/var/log/debug`. See the *EMC Data Domain Operating System Administration Guide* for more information.

BoostFS generates a local log file that contains its internal status, activities, warnings, and errors. You can specify the logging level in addition to the name and location of the log file by using the CLI or the BoostFS configuration file.

If you want to initiate troubleshooting during BoostFS operations, you can use the `kill -s SIGUSR2 boostfs_pid` to rotate the BoostFS log level.

You might need to set a size limit on the log file to ensure that when the size of the log file reaches that limit, BoostFS will rotate log messages.

You can configure the maximum size of the BoostFS log file in the configuration file. You can also configure the number of older log files you wish to keep.

When the log file size reaches the maximum specified size (in MB), the log file is renamed by appending ".1" to the log file name. If there is already an existing log file that ends in ".1," that file is renamed to replace ".1" with ".2." As each log file reaches the maximum size, log files with numbers (n) appended are renamed .n+1 up to the maximum log rotate number.

Common issues

Unable to establish a BoostFS mount point

The following section describes other mount common errors and solutions:

The mount point `mount-point` is nonempty.

BoostFS cannot be mounted on a nonempty mount point. Try mounting BoostFS again on an empty mount point.

This error can occur if the user already has a mount point established. You might want to check to see if this might be the case. You can either use the already established mount point or use `boostfs umount` to unmount the existing mount point and establish a new one. This error can also appear if the directory on the

client being mounted to already contains files. In this case, these file need to be removed or an empty mount point directory must be selected instead.

Cannot mount mount-point: unexpected error, please see log for details.

Most often seen if the DD Boost protocol isn't enabled and configured on the Data Domain System. EMC recommends checking the boostfs log files for more details and confirming DD Boost is enabled using the `ddboost status` command on the Data Domain System.

**Invalid mount point option and value pair [option=key from config file] [value= value from config file]
/mnt/test: Configuration initialization failed**

This message can appear when errors occur during the processing of the BoostFS configuration file. EMC recommends reviewing the specific key and value printed out in the BoostFS configuration file and making any corrections.

Cannot mount mount-point: unexpected error

This error is most often seen when using Kerberos authentication and an error exists in the setup. EMC recommends reviewing the Kerberos instructions in this configuration guide and ensuring the values are set properly.

Insufficient access to or storage-unit storage-unit does not exist

This error is seen if you are using Kerberos and the Kerberos ticket has expired. EMC recommends using the `boostfs kerberos query` to confirm whether the user's ticket has expired, and using the `boostfs kerberos set` command to reconfigure the expired ticket.

Unable to unmount a BoostFS mountpoint

fusermount: failed to unmount mount-point: Invalid argument

This message can appear if the BoostFS mount point has not yet been established. There should be no issues if a mount point has already been unmounted, but this error can still appear.

Unable to access a BoostFS mount point

A permission or privileges error can appear when attempting to use the mount point. This error is most often seen when the user does not have the necessary permissions to access a mount point. By default, the only user allowed to access a mount point is the one that established it. To allow other users to share this mount point, you must include the `-o allow-others=true` option either on the command line or in the BoostFS configuration file.

Configuration values are not taking effect

Typically configuration parameters are not taking effect because the `[global]` label at the top of the BoostFS configuration file has not been uncommented. EMC recommends that you check the configuration file to confirm `# {global}` has been changed to `[global]`. The comment character `#` needs to be removed from the `[global]` label and any other parameters below it for them to be used.

CHAPTER 5

Appendix

- [About Puppet](#)..... 32
- [References](#)..... 32

About Puppet

If you have an enterprise/remote environment, you can install and configure Puppet to distribute the BoostFS configuration file – and any future updates – to clients. This is a best practice.

Puppet is open-source software that allows you to manage clients in a master-server manner. When you install Puppet, you designate one system as the master. BoostFS uses Puppet to distribute or “push” BoostFS to different machines in a large enterprise environment.

In smaller-scale environments, you can simply install BoostFS on individual machines.

For more information about Puppet, see the Puppet Labs website at <https://puppet.com/product/puppet-enterprise-and-open-source-puppet>.

References

The following documents, located at EMC Online Support, provide additional and relevant information. Access to these documents depends on your login credentials. If you do not have access to a document, contact your EMC representative.

- *EMC Data Domain Operating System Version 6.0 Administration Guide*
- *EMC Data Domain Boost for OpenStorage Version 3.3 Administration Guide*
- *EMC Data Domain Boost for Partner Integration Version 3.2.1 Administration Guide*
- *EMC Data Domain Boost SDK Version 3.3 Programmer's Guide*
- *EMC Data Domain Operating System Version 6.0 Initial Configuration Guide*