

CipherTrust Transparent Encryption

CTE Agent for Linux for CM

Advanced Configuration and Integration Guide

Release 7.2.0

Document Version 6

August 04, 2022



CTE Agent for Linux for CM
August 04, 2022

All information herein is either public information or is the property of and owned solely by Thales and/or its subsidiaries and affiliates who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales DIS France S.A. and any of its subsidiaries and affiliates (collectively referred to herein after as "Thales") information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any publicly accessible network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Thales reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

Copyright © 2009-2022 Thales Group. All rights reserved. Thales and the Thales logo are trademarks and service marks of Thales and/or its subsidiaries and affiliates and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the properties of their respective owners.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

Contents

Preface	16
The CTE Agent Documentation Set	16
Document Conventions	16
Typographical Conventions	16
Notes, Tips, Cautions, and Warnings	17
Hardware-Related Warnings	18
Sales and Support	18
Chapter 1: Overview of CTE	19
CTE Terminology	19
CTE Components	20
How to Protect Data with CTE	20
Chapter 2: Getting Started with CTE for Linux	21
Additional Considerations	21
Tracking and Preventing Local User Creation	21
Restricted Directories	21
Restricted Mode	22
CTE Agent Installation with UEFI Secure Boot	23
Public Certificate Naming Convention	23
Getting the Current Public Certificate	23
Adding the Certificate to the MOK List	23
Linux Package Installation	24
Installing the Unsigned RPM Package	24
Installing the Signed RPM Package with Yum	25
Installing the Signed RPM Package Manually	26
Verifying Kernel Compatibility	26
Installing CTE with No Key Manager Registration	27
Configuring CTE for Linux with CipherTrust Manager	28
Installation Overview	28
Installation and Registration Options	28
Installation Method Options	28
Hardware Association (Cloning Prevention) Option	29
Installation Prerequisites	29
Recommendations and Considerations	29
Network Setup Requirements	29
Port Configuration Requirements	30
Communication with Key Manager	30

Communication for LDT over CIFS/NFS	30
Interactive Installation on Linux	30
Before You Begin	30
Silent Installation on Linux	33
Guarding a Device with CipherTrust Manager	35
Access the CipherTrust Manager Domain	36
Create an Encryption Key	36
Create a Standard Policy	37
Create a GuardPoint	40
Chapter 3: Special Cases for CTE Policies	42
Re-Enabling Automatic Signing for Host Settings	42
Restricting Access Overrides from Unauthorized Identities	42
Chapter 4: Logs	44
Setting CTE Agent Logging Preferences	44
Audit Logs	46
Analyzing Audit log entries	46
File System Audit Log Effects Codes	46
Concise Logging	48
Using Concise Logging	48
Considerations	48
Configuring Concise Logging for CTE Clients or Client Groups with CipherTrust Manager	48
Chapter 5: Enhanced Encryption Mode	49
Compatibility	49
Difference between AES-CBC and AES-CBC-CS1	50
Disk Space	50
Encryption Migration	50
File Systems Compatibility	50
File System Requirements	51
Samba Share	51
Storing Metadata	51
Missing IV file	52
HDFS	52
Backups	52
Container Compatibility	53
Using the AES-CBC-CS1 Encryption Mode in CM	53
Exceptions and Caveats	53

Best Practices for AES-CBC CS1 Keys and Host Groups	53
---	----

Chapter 6: CTE and systemd	54
---	-----------

Overview of CTE and systemd	54
Linux Distributions that Support CTE and systemd	54
CTE Agent Control Changes on systemd	54
CTE Configuration Changes Required on systemd	55
About systemd Dependency Changes for Unit Configuration Files	56
Location of Application Unit Configuration Files	57
Adding Dependencies to systemd Unit Configuration Files	57
Adding Applications to the secfs-fs-barrier.service File	57
Adding Dependencies to the saslauthd.service File	58
Supported Use Cases	58
Manually Stop 3rd Party Applications on a Live System	59

Chapter 7: Utilities for CTE Management	60
--	-----------

secfsd Utility	60
secfsd syntax	60
secfsd Examples	61
Display GuardPoint Information	61
Display GuardPoint Information in a Different Format	62
Display Host Settings	62
Display Key Status	62
Display Lock Status	63
Agent Security Configuration Protection	63
Display CTE Log Status	63
Display Applied Policies	64
Display CTE Process Information	64
Display CTE Version Information	64
Display CTE Crypto Information	64
Manually Enable a GuardPoint in CipherTrust Manager	64
secfsd and Raw Devices	65
vmsec Utility	65
vmsec Syntax	65
vmsec Examples	65
Display CTE Challenge String	66
Display CTE Status	66
Entering a Password	66
Display Kernel Status	66

Display CTE Build Information	68
Display Contents of Conf files	68
Restricting Access Overrides with Client Settings	68
Using Advanced Encryption Set New Instructions (AES-NI)	69
vmd utility	69
Syntax	69
Display the Installed Version	70
Agent Health Utility	70
The Agent Health check script	70
Agent Health Return Codes	71
Help	71
Return Codes	71
Wait Time	72
agentinfo Utility (Java version)	73
check_host Utility	73
check_host Syntax	73
register_host Utility	73
fsfreeze and xfs_freeze	74
Restrictions	74
Platform Restrictions	74
Target Restrictions	74
File System Restrictions	74
LDT Restrictions	75
Offline Data Transformation Restrictions	75
Displaying Information for Nested File Systems with the DF tool	75
Chapter 8: Installing CTE on Hadoop	76
Overview	76
Overview of CTE on HDFS	76
HDFS Administrator	76
Administrator	77
HDFS Administrator	77
Implementing CTE on HDFS	77
CTE on HDFS Implementation Assumptions	77
Create an Encryption Zone in HDFS Name Space for AWS EMR	77
Using the Original Information from HDFS	78
Create an HDFS Host Group and GuardPoint in CipherTrust Manager	78
Notes for HDFS Cluster Policies	79
Additional Settings for HDFS (Linux Clients)	79
Adding a New DataNode to a CTE-protected HDFS	80

Install CTE on the Cluster Nodes before Ambari Installs Hadoop	80
Configure NameNodes	81
Configure the Ambari Hadoop NameNodes	81
Create and Configure the Cloudera Hadoop Namenodes and Datanodes	82
Take a DataNode Offline and Perform Data Transformation	82
Implementing CTE on HDFS on a Single Host	83
HDFS Upgrade with CTE	84
Upgrading one node at a time	84
Upgrade CTE with CTE-LDT in an HDFS Cluster	84
Rolling Upgrades	85
Configure the Hadoop Cluster for CTE	86
Create a CipherTrust Configuration Group	86
Update the Hadoop-env Template with CTE Settings	88
Modify the HDFS IOCTL	88
Change the HDFS File Rename Check	88
User Information Push	88
Uninstalling CTE for the Hadoop Cluster	89
CTE Installation and Configuration	89
Installing and Configuring CTE on an HDFS Node	89
Modifying host settings for HDFS hosts on the CipherTrust Manager	90
Simple Modification	90
Modifying Host Group for HDFS NameNodes HA on CipherTrust Manager	90
Configuring Hadoop to Use CTE	92
Verify secfsd is Running with Hadoop Environment	92
HDFS Name Cache	93
Enabling CTE on HDFS	93
Deleting Metadata in HDFS when Migrating Out of LDT	93
Chapter 9: Using CTE with Oracle	95
Oracle RAC ASM and ASMLib	95
Using CTE with an Oracle RAC ASM	95
ASMLib	95
Important ASM Commands and Concepts	95
Rebalancing Disks	95
Mapping Raw Devices	96
Checking Rebalance Status	96
Determining Best Method for Encrypting Disks	97
Online Method (No Application / Database Downtime)	97
Offline Method (Backup the DB)	97
General Prerequisites	98

Setup	98
Modify the UDEV Rules	98
Altering ASM_DISKSTRING on ASM	98
Specific Prerequisites	99
Establishing a Starting Point	99
The Importance of Device Mapping	99
Important Note about Raw Devices on UNIX	99
Oracle RAC ASMLib Multi-Disk Online Method	100
Assumptions	100
About Oracle RAC ASM Raw Devices	100
When Not Using ASMLib	100
Devices using Raw Bindings	100
Multipath I/O Devices	101
Standard Devices	101
Consistent Naming of Devices across RAC Nodes	101
Oracle RAC ASM Multi-Disk Online Method	101
Checking for Space	101
Adding a Disk to the Diskgroup	102
Troubleshooting	102
Oracle RAC ASM Multi-Disk Offline Method (Backup/Restore)	103
Troubleshooting	103
Surviving the Reboot and Failover Testing	104
Preparing for Failover Testing with ASMLib	104
CTE Load Order and Startup Scripts	104
Failover Testing	104
Issues with Device Mapper and Invalid Guard Path	104
Using CTE with Oracle ASM Filter Driver	105
About Oracle ASM Filter Driver	105
Audience	105
Prerequisites	105
Enable Oracle ASMFD for CTE	105
Configure CTE and create a guarded Oracle ASMFD	106
Convert a baseline ASMFD disk group to a CTE guarded ASMFD disk group using the Offline method	107
Convert a baseline ASMFD disk group to a CTE guarded ASMFD disk group using the Online method	109
Uninstall/ Upgrade CTE with active ASMFD Setup	111
For an Active ASMFD Standalone Setup	111
For an Active ASMFD RAC Setup	111
Basic Troubleshooting Techniques	112

Verifying Database Encryption	113
Option 1	113
Option 2	114
Option 3	114
Chapter 10: Integrating and Configuring EDB	115
Overview	115
Prerequisites	115
Preparing to Create GuardPoints	115
Integration with CTE	115
Chapter 11: Using CTE with Pacemaker	117
Overview	117
Considerations and Requirements	117
Creating GuardPoints	118
Chapter 12: Configuring Support for SAP HANA	120
Overview	120
Customizing CTE for SAP HANA in HA Mode	120
Using SAP HANA with LDT	122
Setting Memory Allocation	122
Chapter 13: Using CTE with GlusterFS	124
Overview	124
Considerations and Requirements	124
Configuring GlusterFS for CTE	124
Chapter 14: NetApp Snapshot Directory	126
Overview	126
Accessing snapshots	126
Enabling Snapshots	126
Dataxform Considerations	126
Best Practices	127
Chapter 15: Using CTE with Quantum StorNext	128
Overview of using CTE with Quantum StorNext	128
CTE and Quantum StorNext Compatibility	128
Supported StorNext Server and Client Configurations	128
Supported GuardPoint and Key Settings for SNFS File Systems	129

Supported Concurrent Access Read/Write Scenarios	129
Setting up CTE and Quantum StorNext Integration	130
Integration Task Overview	130
Installing and Configuring a Quantum StorNext MDC Server for Use with CTE	131
Installing and configuring Quantum StorNext DLC Clients for Use with CTE	131
Ensuring that the StorNext SNFS File System Starts Before secfs	131
Installing the CTE Agent on Each StorNext LAN client	131
Stop secfs Before Upgrading StorNext LAN Clients	132

Chapter 16: Using CTE with McAfee Endpoint Security for Linux

Threat Prevention	133
Supported McAfee Versions and Linux Operating Systems	133
Ensuring the Correct McAfee Service Startup and Shutdown Order	133
Excluding CTE protected directories with McAfee 10.5.x	133
Ensuring the Correct McAfee Service Order in systemd	134
Starting or Stopping McAfee and CTE Manually	134
Updating McAfee	135
Virus Scanning Behavior Differences for CIFS and NFS GuardPoints	135

Chapter 17: Using CTE with Trend Micro Deep Security Software

Chapter 17: Using CTE with Trend Micro Deep Security Software	136
Supported Deep Security Versions and Linux Operating Systems	136
Ensuring Correct Deep Security Service Startup Order	136
Ensuring Correct Deep Security Service Startup Order in systemd	136
Ensuring Correct Deep Security Service Startup Order Manually	137
Updating Deep Security	137

Chapter 18: CTE COS for Amazon S3

Chapter 18: CTE COS for Amazon S3	138
Overview	138
Supported operations	139
Limitations	139
Multi-part Upload Restrictions	140
System and Software Requirements	140
Client Software Requirements	140
CTE COS S3 Installation Overview	141
Install Required Linux Packages	141
Install CTE with COS Service	142
Configure the AWS CLI to use the COS Root CA Certificate	142
Configure the AWS CLI Network Proxy	143
Configure CTE COS S3	143

Setting the Default Chunk Size	144
Optionally Configure a CTE COS S3 Role for Guarded Buckets	144
Secure an S3 Bucket with the CTE COS S3 Role	147
Disable the CTE COS S3 Role for an S3 Bucket	147
Guard an AWS Bucket	148
Additional COS Proxy Root CA Certificate Information	148
Protecting Python Programs with CTE	150
Benefits of Using PyInstaller with CTE	150
Getting PyInstaller	150
Example Usage	151
Enable COS on an Agent with no COS Service	151
Uninstall COS from an Agent	152
Chapter 19: CTE-Efficient Storage for Linux	153
Introduction to CTE-Efficient Storage	153
Requirements and Considerations	153
CTE-Efficient Storage Enhanced Storage Arrays	153
Storage Arrays Compatible with CTE-Efficient Storage	154
Sharing Encryption Keys	154
Storage Array Registration	154
Efficient Storage Device Header and CTE Private Region	155
Device Size	155
ES GuardPoint Encryption Keys	156
Policy Requirements for ES GuardPoints	156
Guarding an Efficient Storage Device on Linux	156
Requirements for Efficient Storage GuardPoints on Linux	156
Initialize Linux Efficient Storage Devices	157
Initialize a New Linux Device	157
Initialize and Resize an Existing Linux Device	158
Guard the Linux Device with an ES GuardPoint	159
Data Relocation on Existing Linux Devices	161
Data Transformation on Existing Linux Devices	161
Example of Creating an ES GuardPoint on an Existing Linux Device	162
Guarding a CTE-Efficient Storage Device with Multiple IO Paths on Linux	163
Linux System and ES GuardPoint Administration	163
voradmin ESG Commands on Linux	164
Viewing Device Status and the ES Header on Linux	164
File System Mount Points on Linux	164
Auto Mount Options for File System Devices on Linux	165
Linux System Utilities for Signing	165

Changing the Encryption Key on Linux ESG Devices	166
Creating a New Policy for Key Rotation	167
Prepare an ES GuardPoint for Rekey	167
Resizing Guarded Efficient Storage Devices	168
Example: Resizing a Linux Device	169
Use Cases involving Efficient Storage GuardPoints	170
Use Case 1: Single Encryption Key	170
Use Case 2: Device-Level GuardPoints	170
Use Case 3: Directory-Level GuardPoints	171
Challenges with Root Access on Linux	172
Best Practices for the Migration of a legacy Raw Device Guardpoint to an Efficient Storage Guardpoint	176
Migration Prerequisites	176
Migration	176
Use Case 4: Using CTE-Efficient Storage with LVM	177
Alerts and Errors on Linux	178
Encryption key on device has not been made available	178
Specified policy disagrees with metadata set on the Guard Path	179
Device has not been configured for Efficient Storage	179
Device not resized for guarding as Efficient Storage	179
Data transformation failed	179
Data transformation in progress	179
Device <device-name> is configured to guard as Efficient Storage GuardPoint	179
Device <device-name> is configured as Efficient Storage GuardPoint	180
GuardPoint for device <device-name> still guarded on CipherTrust Manager	180
Failed to open device <device-name>, error Device or resource busy	180
Device <device-name> is not configured as Efficient Storage	180
Abort! Error: Could not stop secfs, secvm device(s) busy	180
Abort! Error: Could not unmount file systems	180
A dependency job for esg.mount failed. See 'journalctl -xe' for details	180
ESG-ALERT: IO error on header for [GuardPoint]	181
ESG-ALERT: Data transformation failure on [GuardPoint]	181
ESG-INFO: Data transformation complete on [GuardPoint]	181
ESG-ALERT: Failed to resize <device-name>	181
FSADM-ALERT: ESG required Signature Set for system utilities may have to be resigned	181
File System is not automatically mounted after IDT completes	181
Chapter 20: In-Place Data Transformation for Linux	182
Introduction to In-Place Data Transformation (IDT)	182
Requirements for IDT-Capable GuardPoints	182

IDT-Capable GuardPoint Encryption Keys	183
Key Attributes - Example	183
Policy Requirements for IDT-Capable GuardPoints	183
The CTE Private Region and IDT Device Header	184
CTE Private Region Location	184
Device Size	185
Guarding an IDT-Capable Device on Linux	186
Initializing an IDT-Capable Device	186
Initialize a New Linux Device	186
Initialize a Linux Device with Existing Data	187
Guard the Linux Device with an IDT-Capable GuardPoint	189
Data Relocation and Transformation on Existing Linux Devices	191
Example of Creating an IDT-Capable GuardPoint on an Existing Linux Device	192
Changing the Encryption Key on Linux IDT-Capable Devices	193
Guarding an IDT-Capable Device with Multiple IO Paths on Linux	196
Viewing Device Status and the IDT Device Header	197
Linux System and IDT-Capable GuardPoint Administration	197
voradmin IDT Commands on Linux	197
File System Mount Points on Linux	198
Auto Mount Options for File System Devices on Linux	198
Linux System Utilities for Signing	199
Resizing Guarded IDT Devices	199
Use Cases Involving IDT-Capable GuardPoints	200
Best Practices for the Migration of a legacy Raw Device Guardpoint to an In-Place Device Guardpoint	200
Alerts and Errors on Linux	200
Encryption key on device has not been made available	200
Specified policy disagrees with metadata set on the Guard Path	200
Device has not been configured for IDT-Capable or Efficient Storage	201
Device not resized for guarding as IDT-Capable or Efficient Storage	201
Data transformation failed	201
Data transformation in progress	201
Device <device-name> is configured to guard as Efficient Storage GuardPoint	201
Device <device-name> is configured as Efficient Storage GuardPoint	201
Device <device-name> is configured to guard as IDT-Capable GuardPoint	202
Device <device-name> is configured as IDT-Capable GuardPoint	202
GuardPoint for device <device-name> still guarded on DSM	202
Failed to open device <device-name>, error Device or resource busy	202
Device <device-name> is not configured as IDT-Capable	202
Abort! Error: Could not stop secfs, secvm device(s) busy	202

Abort! Error: Could not unmount file systems	203
A dependency job for idt.mount failed. See 'journalctl -xe' for details	203
ESG/IDT-ALERT: IO error on header for [GuardPoint]	203
ESG/IDT-ALERT: Data transformation failure on [GuardPoint]	203
ESG/IDT-INFO: Data transformation complete on [GuardPoint]	203
ESG/IDT-ALERT: Failed to resize <device-name>	203
FSADM-ALERT: ESG/IDT required Signature Set for system utilities may have to be resigned	203
File System is not automatically mounted after IDT completes	204
Chapter 21: CTE with Teradata Database Appliances	205
IDT-Capable GuardPoints and Teradata Database Appliances	205
Requirements and Considerations	205
Location of the CTE Private Region	205
Metadata File Access and Teradata Clusters	206
Additional Requirements and Considerations	206
Guarding a Teradata Database Device	207
Install CTE on the Teradata Database Appliance	207
Identify the Devices to Be Guarded	208
Select the Initial Configuration Method	209
Initialize and Guard the Database Devices Using the Standard Initialization Method	209
Guard the Devices as IDT-Capable GuardPoints on CM	210
Guard the Devices as IDT-Capable GuardPoints on DSM	210
Viewing Device and Data Transformation Status	212
Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method	212
Changing the Encryption Key on Teradata Devices	215
Access Rules to Apply on the Teradata Database Appliance	216
Replication of IDT Metadata Files Across Members of a Clique	217
Specific Issues to Consider	218
General PCL Error	218
Offline Node in Clique During Data Transformation	218
Adding a New Node to a Clique	218
Interoperability with Host Groups	219
Generate IDT-Capable metadata for Teradata Storage Expansion	220
Best Practices	221
Using a of Host/Client Group to Guard Metadata Directories	221
Using a Host/Client Group for Guarding Teradata Devices in a Clique	221
Best Practice for Preparation for Initial Data Transformation or Rekey	221
Uninstalling CTE from the Teradata Cluster	222
Alerts and Errors	223
General Errors	223

IDT-TD-ALERT: Node <node name> did not respond to pcl command	223
IDT-TD-ALERT: Node <node name> failed to perform voradmin task	223
IDT-TD-ALERT: Failed to find clique for disk <device>	223
IDT-TD-ALERT: Failed to move or rename IDT-Capable metadata file on remote nodes	224
IDT-TD-ALERT: Failed to get GuardPoint status on remote nodes	224
Operation Errors	224
IDT-TD-ALERT: Failed to distribute IDT-Capable metadata file to remote nodes	224
IDT-TD-WARNING: Failed to delete IDT-Capable metadata file on remote nodes	224
IDT-TD-ALERT: Failed to complete rekey on remote nodes	224
Chapter 22: Upgrading CTE on Linux	225
Upgrading CTE	225
Scheduled Upgrade Feature	225
Warnings for CTE for Linux	226
Using the Scheduled Upgrade Feature	226
Performing a Manual Upgrade When an Upgrade is Already Scheduled	227
Chapter 23: Uninstalling CTE from Linux	229
Considerations	229
Procedure	229

Preface

The CTE Agent for Linux for CM provides information about advanced installation, configuration, and integration options for CTE for Linux.

The CTE Agent Documentation Set

The following guides are available for CTE Agent:

- *CTE Agent for Linux Quick Start Guide*
- *CTE Agent for Linux Advanced Configuration and Integration Guide*
- *CTE Agent for Windows Quick Start Guide*
- *CTE Agent for Windows Advanced Configuration and Integration Guide*
- *CTE Agent for AIX Installation and Configuration Guide*
- *CTE Data Transformation Guide*
- *CTE-Live Data Transformation with Data Security Manager*
- *CTE-Live Data Transformation with CipherTrust Manager*
- *Compatibility Matrix for CTE Agent with CipherTrust Manager*
- *Compatibility Matrix for CTE Agent with Data Security Manager*
- *Compatibility Matrix for CTE Agent for AIX with CipherTrust Manager*
- *Compatibility Matrix for CTE Agent for AIX with Data Security Manager*
- *Release Notes for CTE for Linux Version 7.2.0.128*
- *Release Notes for CTE for Windows Version 7.2.0.128*
- *Release Notes for CTE for AIX Version 7.2.0.56*

To access any of these guides for the latest releases of CTE Agent, go to <https://thalesdocs.com/ctp/cte/index.html>.

Document Conventions

The document conventions describe common typographical conventions and important notice and warning formats used in Thales technical publications.

Typographical Conventions

This section lists the common typographical conventions for Thales technical publications.

Table 3-1: Typographical Conventions

Convention	Usage	Example
bold regular font	GUI labels and options	Click the System tab and select General Preferences .
<i>bold italic monospaced font</i>	Variables or text to be replaced	https://< <i>Token Server name</i> >/admin/ Enter password: < <i>Password</i> >

Table 3-1: Typographical Conventions (continued)

Convention	Usage	Example
regular monospacedfont	<ul style="list-style-type: none">• Commands and code examples• XML examples	<code>session start iptarget=192.168.253.102</code>
<i>italic regular font</i>	GUI dialog box titles	The <i>General Preferences</i> window opens.
	File names, paths, and directories	<i>/usr/bin/</i>
	Emphasis	<i>Do not</i> resize the page.
	New terminology	<i>Key Management Interoperability Protocol (KMIP)</i>
	Document titles	See <i>CTE Agent for Linux for CM</i> for information about CipherTrust Transparent Encryption.
quotes	<ul style="list-style-type: none">• File extensions• Attribute values• Terms used in special senses	"js", ".ext" "true" "false", "0" "1+1" hot standby failover

Notes, Tips, Cautions, and Warnings

Notes, tips, cautions, and warning statements may be used in this document.

A Note provides guidance or a recommendation, emphasizes important information, or provides a reference to related information. For example:

Note

It is recommended to keep tokenization keys separate from the other encryption/decryption keys.

A tip is used to highlight information that helps you complete a task more efficiently, such as a best practice or an alternate method of performing the task.

Tip

You can also use Ctrl+C to copy and Ctrl+P to paste.

Caution statements are used to alert you to important information that may help prevent unexpected results or data loss. For example:



CAUTION

Make a note of this passphrase. If you lose it, the card will be unusable.

A warning statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data. For example:



WARNING

Do not delete keys without first backing them up. All data that has been encrypted with deleted keys cannot be restored or accessed once the keys are gone.

Hardware-Related Warnings

The following warning statement is used to indicate the risk of electrostatic discharge of equipment:



ELECTROSTATIC DISCHARGE

If this warning label is affixed to any part of the equipment, it indicates the risk of electrostatic damage to the module. To prevent equipment damage, follow suitable grounding techniques.

The following warning statement is used to indicate the risk of hazardous voltages of equipment:



HAZARDOUS VOLTAGES

The warnings in this section indicate voltages that could cause serious danger to personnel.

Sales and Support

If you encounter a problem while installing, registering, or operating this product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or Thales Customer Support.

Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

For support and troubleshooting issues:

- <https://supportportal.thalesgroup.com>
- (800) 545-6608

For Thales Sales:

- <https://cpl.thalesgroup.com/encryption/contact-us>
- CPL_Sales_AMS_TG@thalesgroup.com
- (888) 267-3732

Chapter 1: Overview of CTE

For very large data sets, initial encryption deployments can affect data availability, require unacceptable maintenance windows or require cloning and synchronizing data. Encrypting millions of files can span hours or even days, which can delay encryption, or require extra disk space and data synchronization, which can be labor-intensive. Rekeying large data sets can demand significant processing time and lengthy maintenance windows. Security and IT teams face tough tradeoffs, having to choose between security and availability.

CipherTrust Transparent Encryption operates with minimal disruption, effort, and cost. Its transparent approach enables security organizations to implement encryption without changing application, networking, or storage architectures. CipherTrust Live Data Transformation builds on these advantages, offering patented capabilities that deliver breakthroughs in availability, resiliency and efficiency.

CTE includes several unique utilities to help you encrypt and manage your data. It also integrates with several third-party platforms such as Oracle, Teradata Database Appliances, and Amazon S3.

This document describes the installation and advanced configuration options for CTE, as well as detailed information about how to integrate CTE with the supported third-party products.

CTE Terminology

The CTE documentation set uses the following terminology:

Term	Description
CTE	<p>CipherTrust Transparent Encryption is a suite of products that allow you to encrypt and guard your data. The main software component of CTE is the CTE Agent, which must be installed on every host whose devices you want to protect.</p> <div style="border: 1px solid black; padding: 5px;"><p>Note</p><p>This suite was originally called Vormetric Transparent Encryption (VTE), and some of the names in the suite still use "Vormetric".</p><p>For example, the default installation directory is <code>/opt/vormetric/DataSecurityExpert/agent/</code>.</p><p>For example, the default installation directory is <code>/opt/vormetric/DataSecurityExpert/agent/</code> for Linux and AIX, and <code>C:\Program Files\Vormetric\DataSecurityExpert\agent\</code> for Windows.</p></div>
CTE Agent	<p>The software that you install on a physical or virtual machine in order to encrypt and protect the data on that machine. After you have installed the CTE Agent on the machine, you can use CTE to protect any number of devices or directories on that machine.</p>
key manager	<p>An appliance that stores and manages data encryption keys, data access policies, administrative domains, and administrator profiles</p>
host / client	<p>In this documentation, host and client are used interchangeably to refer to the physical or virtual machine on which the CTE Agent is installed.</p>
GuardPoint	<p>A device or directory to which a CTE data protection and encryption policy has been applied. CTE will control access to, and monitor changes in, this device and directory, encrypting new or changed information as needed.</p>

CTE Components

The CTE solution consists of two parts:

- The *CTE Agent software* that resides on each protected virtual or physical machine (host). The CTE Agent performs the required data encryption and enforces the access policies sent to it by the *key manager*. The communication between the CTE Agent and the key manager is encrypted and secure.

After the CTE Agent has encrypted a device on a host, that device is called a *GuardPoint*. You can use CTE to create GuardPoints on servers on-site, in the cloud, or a hybrid of both.

- A *key manager* that stores and manages data encryption keys, data access policies, administrative domains, and administrator profiles. After you install the CTE Agent on a host and register it with a key manager, you can use the key manager to specify which devices on the host that you want to protect, what encryption keys are used to protect those devices, and what access policies are enforced on those devices.

Thales recommends that you use the CipherTrust Manager unless you need a feature that is only supported by the DSM, as described below.

- Container Security
- CTE-Efficient Storage

For details about any of these features, see the *CTE Agent for Linux Advanced Configuration and Integration Guide* and *CTE-Live Data Transformation with Data Security Manager*.

Note

For a list of CTE versions and supported operating systems, see the [CTE Compatibility Portal](#) or the *Compatibility Matrix for CTE Agent with CipherTrust Manager* and the *Compatibility Matrix for CTE Agent with Data Security Manager*.

All CTE documentation is available at <https://thalesdocs.com/ctp/cte/index.html>.

How to Protect Data with CTE

CTE uses policies created in the associated key manager to protect data. You can create policies to specify file encryption, data access, and auditing on specific directories and drives on your protected hosts. Each GuardPoint must have one and only one associated policy, but each policy can be associated with any number of GuardPoints.

Policies specify:

- Whether or not the resting files are encrypted.
- Who can access decrypted files and when.
- What level of file access auditing is applied when generating fine-grained audit trails.

A Security Administrator accesses the key manager through a web browser. You must have administrator privileges to create policies using either key manager. The CTE Agent then implements the policies once they are pushed to the protected host.

CTE can only enforce security and key selection rules on files inside a guarded directory. If a GuardPoint is disabled, access to data in the directory goes undetected and ungoverned. Disabling a GuardPoint and then allowing unrestricted access to that GuardPoint can result in data corruption.

Chapter 2: Getting Started with CTE for Linux

This chapter section describes how to install CTE for Linux, register it with your selected key manager, and then create a simple GuardPoint on the protected host. It contains the following topics:

Additional Considerations	21
CTE Agent Installation with UEFI Secure Boot	23
Linux Package Installation	24
Verifying Kernel Compatibility	26
Installing CTE with No Key Manager Registration	27
Configuring CTE for Linux with CipherTrust Manager	28

Additional Considerations

The following sections describe some of the things to keep in mind when configuring CTE.

Tracking and Preventing Local User Creation

CTE audits any attempts to change user authentication files. It also allows you to prevent any change to user authentication files using the host settings `protect`. This includes, but is not limited to user creation, modification, and deletion, or to deny users.

- The `audit` setting is set to on by default. It logs access to the system credential files but does not prevent account modifications.
- The `protect` setting both audits and prevents local user account modifications. You must manually enable the `protect` setting for tracking and prevention of local user account creation.

The `protect` tag will prevent changes to the files mentioned below. In the absence of the `protect` tag in host/client settings, operations on these files are permitted. When a log entry is generated, it is tagged with an `[audit]` tag.

- `/etc/passwd`
- `/etc/group`
- `/etc/shadow`
- `/etc/gshadow`
- `/etc/ssh/sshd_config`
- `/etc/ssh/sshrd`

Notes

- You do not have to restart CTE after applying or removing these host settings.

Restricted Directories

Linux does not allow you to guard the following directories:

- `<secfs install root>/agent/secfs/`
- `/etc`
- `/tmp`
- `/usr`

- /usr/lib
- /usr/lib/pam
- /var/log/vormetric

Linux does not allow you to guard the following directories and all of their subdirectories:

- <install root>/agent/secfs/bin
- <secfs install root>/agent/vmd
- /etc/vormetric
- /etc/pam.d
- /etc/security
- /usr/lib/security
- /etc/rc*

Restricted Mode



CAUTION

If you install or upgrade in restricted mode, you cannot revert to unrestricted mode without uninstalling CTE.

You can install CTE in restricted mode. This mode prevents any user other than `root` from accessing the following directories:

- /var/log/vormetric
- /opt/vormetric/DataSecurityExpert

Restricted Mode also prevents non-`root` users from running the following utilities:

- agenthealth
- agentinfo
- check_host
- register_host
- secfsd
- vmd
- vmsec
- voradmin

Key Agents and Restricted Mode

- On systems where CTE is installed in restricted mode, you cannot install a key agent (pkcs11) or CipherTrust TDE Key Management.
- On systems where a key agent (pkcs11) or CipherTrust TDE are already installed, you cannot install CTE in restricted mode.

Restricted Mode Installation

To install in restricted mode, use the `-r` option.

```
# ./vee-fs-<release>-<build>-<system>.bin -r
```

For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -r
```

RPM Installation

If installing from an RPM directly, prior to installation, type:

```
# export VOR_RESTRICTED_INSTALL_MODE=yes
```

Upgrade in Restricted Mode

The upgrade mode is the same as the installation mode.

CTE Agent Installation with UEFI Secure Boot

If you want to install the CTE Agent software on a Linux system that has UEFI Secure Boot enabled, you must first download the appropriate Thales public certificate and add that certificate to the MOK (Machine Owner Key) list on the host.

Note

The Thales public certificate is valid for three years from the date of issuance. Six months before the current public certificate is set to expire, Thales will release an advisory along with the new certificate that will become valid after the six month grace period expires. You can add the new certificate to the MOK list on all UEFI Secure Boot hosts any time before the old certificate expires and CTE will automatically start using the new certificate when the old certificate expires.

Public Certificate Naming Convention

The Thales public certificate name is `CTE_Secure_Boot_Cert_MM-DD-YYYY.der`. For example, `CTE_Secure_Boot_Cert_01-11-2021.der`.

Getting the Current Public Certificate

You can get the current public certificate in any of the following ways:

- From the CTE Agent installation file using the `-e` option. For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -e
Contents extracted.
# ls | grep CTE_Secure_Boot_Cert
CTE_Secure_Boot_Cert_01-11-2021.der
```

- From the Thales public directory https://packages.vormetric.com/pub/CTE_Secure_Boot/ or from the [Thales Customer Support Portal](#) (under [KB0023449](#)). The certificate on these sites is in PEM format, and must be converted to DER format before it can be added to the MOK list.

For example, if the current certificate name is `CTE_Secure_Boot_Cert_01-11-2021.pem`, you could convert the certificate using the following command:

```
# openssl x509 -inform PEM -outform DER -in CTE_Secure_Boot_Cert_01-11-2021.pem \
-out CTE_Secure_Boot_Cert_01-11-2021.der
```

Adding the Certificate to the MOK List

Note

During this procedure, you will need to reboot the Linux host and then respond to a system prompt as soon as the host restarts. Make sure that all users accessing the host know that it will reboot and that you can respond to the system prompt as soon as the host restarts.

1. Log into the host as `root`.
2. Use the `mokutil --import <cert-name>` command to add the certificate to the MOK list. For example, if the certificate name is `CTE_Secure_Boot_Cert_01-11-2021.der`, you could enter:

```
# mokutil --import CTE_Secure_Boot_Cert_01-11-2021.der
```
3. Enter and confirm a password for this request when prompted.
4. Reboot the host and follow the instructions on the console when the host comes back online. You will need to enter the password you created in the previous step.
If you do not respond to the system prompt to update the MOK when the host restarts, the prompt will time out and you will need to run the `mokutil` command again.
5. When prompted, reboot the host again.
6. After the host has been rebooted the second time you can verify that the certificate has been properly added to the MOK list using the `mokutil --test-key` command. For example:

```
# mokutil --test-key CTE_Secure_Boot_Cert_01-11-2021.der  
CTE_Secure_Boot_Cert_01-11-2021.der is already enrolled
```

Linux Package Installation

This section describes how to access the Linux RPM installation package so that the CTE Agent installation integrates with the distribution software. To access the Linux RPM file, you can:

- Extract the RPM file from the CTE Agent install bin file. This is the easiest method, but the files in the package are not signed and therefore cannot be verified. For details, see ["Installing the Unsigned RPM Package" below](#).
- Download the package from the Yum repository. If you use Yum, the files in the package are signed and the signatures are automatically verified when the package is installed. For details, see ["Installing the Signed RPM Package with Yum" on the facing page](#).
- Manually download the RPM package outside of Yum and manually verify the package signatures. For details, see ["Installing the Signed RPM Package Manually" on page 26](#).



CAUTION

Do not use package installation for SUSE Linux. Instead, use the interactive or silent installation.

Installing the Unsigned RPM Package

The CTE installation `bin` files contain the unsigned native packages. Extract them by running the `bin` file with the `-e` flag.

1. Log on to the host system as `root` and copy or mount the installation file to the host system.
2. Extract the RPM file using the following command:

```
# ./vee-fs-<release>-<build>-<system>.bin -e
```

For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -e  
Contents extracted.  
# ls *rpm  
vee-fs-7.2.0-128-rh8-x86_64.rpm
```


3. To start the installation using the RPM file, use the following command:

```
# rpm -ivh vee-fs-7.2.0-128-rh8-x86_64.rpm
```

4. Follow the prompts to install and register CTE.

For details about the installation and registration process, see the appropriate installation procedure.

- If you are going to register the system with a CipherTrust Manager, see ["Configuring CTE for Linux with CipherTrust Manager" on page 28](#).

Installing the Signed RPM Package with Yum

Note

Before you can download and install the package using the Yum repository, you must contact Thales Customer Support to get the username and password for the package repository on which the package resides.

1. Create the repository file, `/etc/yum.repos.d/Vormetric-VTE.repo`, with the following contents:

```
[vormetric-vte]
releasever=REL_VERSION
name=Vormetric VTE Packages $releasever - $basearch - Source
baseurl=https://USER:PASSWORD@packages.vormetric.com/vte/VERSION/rhel-$releasever/
gpgkey=https://packages.vormetric.com/pub/PKG-GPG-KEY-vormetric
enabled=1
gpgcheck=1
repo_gpgcheck=1
sslverify=1
```

where:

- **REL_VERSION** is the the version of RHEL that you are running. This would be `rhel-6`, `rhel-7`, or `rhel-8`.
- **USER:PASSWORD** is the username and password for the CTE package repository you obtained from Thales Customer Support.
- **VERSION** is the CTE release version number. For example, `7.2.0`.

2. Clean up the Yum cache directory:

```
# yum clean all
```

3. Use Yum to install the CTE binary. For example, if the CTE binary name is `vee-fs-7.2.0-128-rh8-x86_64`, you would enter:

```
# yum install vee-fs-7.2.0-128-rh8-x86_64
```

Note: The first time you install CTE through Yum, you will be asked to import the GPG key. Make sure that you download this key or the install will fail. For example:

```
vormetric-vte/7Server/signature | 198 B 00:00:00
Retrieving key from https://packages.vormetric.com/pub/PKG-GPG-KEY-vormetric
Importing GPG key 0x628536B7:
  Userid      : "Vormetric (PKG-GPG-KEY) <support@vormetric.com>"
  Fingerprint: 7cb5 4f55 40d4 1b63 bf91 c896 f00a 13b0 6285 36b7
  From       : https://packages.vormetric.com/pub/PKG-GPG-KEY-vormetric
Is this ok [y/N]: y
```

4. Follow the prompts to install and register CTE.

For details about the installation and registration process, see the appropriate installation procedure.

- If you are going to register the system with a CipherTrust Manager, see ["Configuring CTE for Linux with CipherTrust Manager" on page 28](#).

Installing the Signed RPM Package Manually

If you want to manually verify the signed version of the CTE RPM package, you can download the public key from the Thales package repository and manually verify the rpm signature.

1. Import the public key from the Thales package repository:

```
# rpm --import https://packages.vormetric.com/pub/PKG-GPG-KEY-vormetric
# rpm -qa gpg-pub*
gpg-pubkey-628536b7-56f9887b : Imported CTE GPG public key.
gpg-pubkey-fd431d51-4ae0493b
```

2. Verify the signature of the package using the `rpm -Kv` command.

```
# rpm -Kv vee-fs-7.2.0-128-rh8-x86_64.rpm
vee-fs-7.2.0-128-rh8-x86_64.rpm:
  Header V4 DSA/SHA1 Signature, key ID 628536b7: OK
  Header SHA1 digest: OK (ed3d33dca580c66c70961cfee143e9877a09544c)
  MD5 digest: OK (95273b36ef1c205a7cea444e14bef15f)
  V4 DSA/SHA1 Signature, key ID 628536b7: OK
```

The output should show that the keys match is OK.

3. To start the installation using the RPM file, use the following command:

```
# rpm -ivh vee-fs-7.2.0-128-rh8-x86_64.rpm
```

4. Follow the prompts to install and register CTE.

For details about the installation and registration process, see the appropriate installation procedure.

- If you are going to register the system with a CipherTrust Manager, see ["Configuring CTE for Linux with CipherTrust Manager" on page 28](#).

Verifying Kernel Compatibility

Thales maintains a compatibility matrix in a JSON file that maps all CTE Agent releases to the Operating Systems and kernels that support those releases. The information in this file allows you to verify the compatibility between any Linux host and the version of the CTE Agent that you want to install on that host.

You can view the current compatibility information in CipherTrust Manager by downloading the most recent compatibility JSON file from Thales and then uploading it to CipherTrust Manager. After the JSON file has been uploaded, the CipherTrust Manager displays the compatibility of all Linux hosts registered with the CipherTrust Manager in the **Compatibility View** on the **Hosts** page.

For details about how to do this in CipherTrust Manager refer to: [Kernel Compatibility Matrix](#).

The following procedure describes how to download the compatibility JSON file and verify its authenticity.

1. Download the `cte_compatibility_matrix.tgz` file from <https://packages.vormetric.com/pub/> or from the [Thales Customer Support Portal](#).
2. Extract the files from the TGZ file. The resulting files are:
 - `CTE_Compatibility_Matrix_Cert_mm-dd-yyyy.pem` — The X.509 Public Key Certificate.
 - `cte_compatibility_matrix.json` — The compatibility JSON file.
 - `cte_compatibility_matrix.sign.sha256` — The SHA256 signature for the JSON file.

For example:

```
# tar -xvzf cte_compatibility_matrix.tgz
CTE_Compatibility_Matrix_Cert_12-17-2020.pem
cte_compatibility_matrix.json
cte_compatibility_matrix.sign.sha256
```

3. Extract the Public Key from the X.509 Public Key Certificate. For example:

```
# openssl x509 -in CTE_Compatibility_Matrix_Cert_12-17-2020.pem -pubkey \
-noout > cte_compatibility_matrix_public_key.pem
```

The Public Key is in PEM format in the file `cte_compatibility_matrix_public_key.pem`.

4. Verify the SHA signature using the Public Key. For example:

```
# openssl dgst -sha256 -verify cte_compatibility_matrix_public_key.pem \
-signature cte_compatibility_matrix.sign.sha256 cte_compatibility_matrix.json
Verified OK
```

Installing CTE with No Key Manager Registration

The following procedure installs the CTE Agent on the host but does not register it with a key manager. You cannot protect any data on the host until the CTE Agent is registered with one of the supported key managers. For a comparison of the available key managers, see ["CTE Components" on page 20](#).

1. Log on to the host where you will install the CTE Agent as `root`. You cannot install the CTE Agent without `root` access.
2. Copy or mount the installation file to the host system. If necessary, make the file executable with the `chmod` command.
3. Install the CTE Agent. A typical installation uses the following syntax:

```
# ./vee-fs-<release>-<build>-<system>.bin
```

For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin
```

To install the CTE Agent in a custom directory, use the `-d <custom-dir>` option. For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -d /home/my-cte-dir/
```

Note: If possible, Thales recommends that you use the default directory `/opt/vormetric`.

To view all installer options, use the `-h` parameter. For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -h
```

4. The Thales License Agreement displays. When prompted, type `y` and press Enter to accept.

The install script installs the CTE Agent software in either `/opt/vormetric` or your custom installation directory and then prompts you about registering the CTE Agent with a key manager.

```
Welcome to the CipherTrust Transparent Encryption File System Agent
Registration Program.
```

```
Agent Type: CipherTrust Transparent Encryption File System Agent
Agent Version: 7.2.0.128
```

```
In order to register with a Key Manager you need a valid registration
token from the CM.
```

```
Do you want to continue with agent registration? (Y/N) [Y]:
```

5. Type `n` and press Enter to end the installation procedure without registering the CTE Agent with either key manager.

When you are ready to register the CTE Agent with a key manager, see :

- ["Configuring CTE for Linux with CipherTrust Manager" below](#)

Configuring CTE for Linux with CipherTrust Manager

This section describes how to install and configure CTE on Linux systems that you plan to register with a CipherTrust Manager.

Installation Overview

The installation and configuration process when you are using CTE with a CipherTrust Manager consists of three basic steps:

1. Gather the information needed for the install and set up your network as described in ["Installation Prerequisites" on the facing page](#).
2. Install CTE on the protected host as described in ["Interactive Installation on Linux" on page 30](#) or ["Silent Installation on Linux" on page 33](#).
3. Register the protected host with CipherTrust Manager and make sure that they can communicate with each other. Registration can be done as part of the initial installation or at any point after the CTE Agent has been installed.

Installation and Registration Options

CTE provides the following installation and registration options. The options you choose determine the information you need to supply during the actual install procedure.

Installation Method Options

There are two methods for installing CTE on Linux platforms:

- **Typical:** Most common and recommended type of installation. Use this method for installing the CTE Agent on one host at a time. See "[Interactive Installation on Linux](#)" on the next page.
- **Silent:** Create pre-packaged installations by providing information and answers to a set of installation questions. Use silent installations when installing on a large number of hosts. See "[Silent Installation on Linux](#)" on page 33.

Hardware Association (Cloning Prevention) Option

CTE's hardware association feature associates the installation of CTE with the machine's hardware. When enabled, hardware association prohibits cloned or copied versions of CTE from contacting the key manager and acquiring cryptographic keys. Hardware association works on both virtual machines and hardware clients.

You can enable hardware association during CTE registration process. You can disable hardware association by re-running the registration program.

To verify if hardware association (cloning prevention) is enabled on a Linux host, on the command line enter the following:

```
cat /opt/vormetric/DataSecurityExpert/agent/vmd/etc/access
```

If you see `usehw:true`, then hardware association is enabled. If you see `usehw:false`, it's disabled.

Installation Prerequisites

This section lists the tasks you must complete, and the information you must obtain, before installing CTE.

Recommendations and Considerations

- The host on which you want to install CTE *must* support AES-NI hardware encryption. If it does not, any attempt to install or upgrade CTE to release 7.0.0 or later will fail.
- Thales recommends that you install CTE in the default location.
- Make the installation root directory `/opt` a real directory. If `/opt` is a symlink, you **must** use the `-d` option to specify the installation directory, which must be a real directory.

For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -d /home/hello/
```

- Ensure read/write permission is granted to other users accessing your shared resource.

Network Setup Requirements

- The IP addresses, routing configurations, and DNS addresses must allow connectivity of the Linux system on which you plan to install CTE to the CipherTrust Manager. After the Linux system is registered as a client with the CipherTrust Manager, the client must be able to poll the CipherTrust Manager in case there are any changes to the encryption keys, policies, or GuardPoints.
- It must also allow for connectivity of the CipherTrust Manager to all clients where you install CTE as well as communication between different CTE clients that need to communicate.
- If the system is a virtual machine, the VM must be deployed and running.

Port Configuration Requirements

Communication with Key Manager

The default port for http communication between CipherTrust Manager and the CTE Agent is **443**. If this port is already in use, you can set the port to a different number during the CTE Agent installation.

Communication for LDT over CIFS/NFS

All nodes that intend to use LDT over CIFS/NFS for GuardPoints must have the following ports open:

- 7024
- 7025

Interactive Installation on Linux

The Linux typical install is an interactive script that asks you a series of questions during the installation. You can also install CTE using a silent installer which pre-packages the install information. This allows you to install CTE on a large number of hosts. (For more information, see "[Silent Install](#)" on page 1).

After you install CTE, you are prompted to register it immediately with a key manager. CTE must be registered with a key manager before you can protect any of the devices on the host. However, you may postpone the registration if you plan to register CTE later.

Note

Do not install CTE on network-mounted volumes like NFS.

Before You Begin

The following prerequisites must be met for CTE to install and register to CipherTrust Manager properly:

- CipherTrust Manager installed and configured. See [CipherTrust Manager Documentation](#) for more information.
- CipherTrust Manager must contain a Client Profile. See [Changing the Profile](#) for more information.
- CipherTrust Manager must contain a registration token. See [Creating a Registration Token](#).
- Optionally, the name of the host group you want this client to be a part of.
- CipherTrust Manager must contain an LDT Communication Group if you will use CTE to guard data over CIFS/NFS shares using LDT policies. See [Managing LDT Communication Groups](#) for more information.

Procedure

1. Log on to the host where you will install the CTE Agent as `root`. You cannot install the CTE Agent without `root` access.
2. Copy or mount the installation file to the host system. If necessary, make the file executable with the `chmod` command.

3. Install the CTE Agent. A typical installation uses the following syntax:

```
# ./vee-fs-<release>-<build>-<system>.bin
```

For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin
```

To install the CTE Agent in a custom directory, use the `-d <custom-dir>` option. For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -d /home/my-cte-dir/
```

Note: If possible, Thales recommends that you use the default directory `/opt/vormetric`.

To view all installer options, use the `-h` parameter. For example:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -h
```

4. The Thales License Agreement displays. When prompted, type `y` and press Enter to accept.

The install script installs the CTE Agent software in either `/opt/vormetric` or your custom installation directory and then prompts you about registering the CTE Agent with a key manager.

```
Welcome to the CipherTrust Transparent Encryption File System Agent
Registration Program.
```

```
Agent Type: CipherTrust Transparent Encryption File System Agent
Agent Version: 7.2.0.128
```

```
In order to register with a Key Manager you need a valid registration
token from the CM.
```

```
Do you want to continue with agent registration? (Y/N) [Y]:
```

5. Enter `y` to continue with the registration process. The install script prompts you to enter the host name or IP address of the CipherTrust Manager with which you want to register CTE.

The default communication port is 443. If you want to specify a different communication port, enter it with the primary key manager host name in the format: `<hostName>:<port#>`

For example:

```
Do you want to continue with agent registration? (Y/N) [Y]: y
```

```
Please enter the primary key manager host name: 10.3.200.141:8445
```

```
You entered the host name 10.3.200.141
```

```
Is this host name correct? (Y/N) [Y]: y
```

6. Enter the client host name when prompted.

```
Please enter the host name of this machine, or select from the following
list.
```

```
[1] sys31186.qa.com
```

```
[2] 10.3.31.186
```

```
Enter a number, or type a different host name or IP address in manually:
```

```
What is the name of this machine? [1]: 2
```

```
You selected "10.3.31.186".
```

7. Enter the CipherTrust Manager registration token, profile name, host group and host description. If you omit the profile name, CipherTrust Manager associates the default client profile with this client.

```
Please enter the registration token: 12345
Please enter the profile name for this host: My-Profile
Please enter the host group name for this host, if any:
Please enter a description for this host: West Coast Datacenter server 5
```

```
Token           : 12345
Profile name    : My-Profile
Host Group      : (none)
Host description: West Coast Datacenter server 5
Are the above values correct? (Y/N) [Y]: Y
```

8. At the hardware association prompt, select whether you want to enable the hardware association feature (for details, see ["Hardware Association \(Cloning Prevention\) Option" on page 29](#)). The default is **y** (enabled):

```
It is possible to associate this installation with the hardware of this machine. If selected, the agent will not contact the key manager or use any cryptographic keys if any of this machine's hardware is changed. This can be rectified by running this registration program again.
Do you want to enable this functionality? (Y/N) [Y]: Y
```

9. At the LDT prompt, specify whether you want this client to use CipherTrust Transparent Encryption - Live Data Transformation (CTE-LDT).

```
Do you want this host to have LDT support enabled on the server? (Y/N) [N]: Y
```

10. If you are planning to create GuardPoints on NFS shares, enter the name of the LDT Communication Group that this node will join.

```
Enter the LDT Communication Group name: LCG1
```



WARNING

The registration token, profile name, client group name and LDT Communication Group name are case-sensitive. If any of these are entered incorrectly, the client registration will not succeed. If the registration fails, click Back in the installer and verify that the case is correct for all entries on this page.

11. At the Cloud Object Storage (COS) prompt, specify whether you want this client to use CTE COS.

```
Do you want to configure this host for Cloud Object Storage? (Y/N) [N]:
```

12. **Note:** You can only install the Cloud Object Storage feature during CTE installation. You cannot install it post installation.

13. CTE finishes the installation and registration process.

```
Generating key pair for the kernel component...done.
Extracting SECFS key
Generating EC certificate signing request for the vmd...done.
Signing certificate...done.
Enrolling agent with service on 10.3.200.141...done.
Successfully registered the CipherTrust Transparent Encryption File System Agent with the
CipherTrust Manager on 10.3.200.141.
```

```
Installation success.
```


14. If you are using CipherTrust Manager version 2.2 or later, you can now use CipherTrust Manager to administer CTE on the client.

If you are using CipherTrust Manager version 2.1 or earlier, change the client password using the manual password creation method. This password allows users to access encrypted data if the client is ever disconnected from the CipherTrust Manager. For details on changing the password, see the CipherTrust Manager documentation.

Silent Installation on Linux

This section describes how to perform a silent (unattended) installation of the CTE on a single host. The silent installation automates the installation process by storing the answers to installation and registration questions in a separate file that you create. You can also use the silent installation to install CTE on multiple hosts simultaneously.

The silent install method installs CTE on the host, and registers the host with the CipherTrust Manager you specify in the silent installation file.

Prerequisites

The following prerequisites must be met for CTE to install and register to CipherTrust Manager properly:

- CipherTrust Manager installed and configured. See [CipherTrust Manager Documentation](#) for more information.
- CipherTrust Manager must contain a Client Profile. See [Changing the Profile](#) for more information.
- CipherTrust Manager must contain a registration token. See [Creating a Registration Token](#).
- Optionally, the name of the host group you want this client to be a part of.
- CipherTrust Manager must contain an LDT Communication Group if you will use CTE to guard data over CIFS/NFS shares using LDT policies. See [Managing LDT Communication Groups](#) for more information.

Procedure

1. Log on as an administrator to the host where you will install CTE.
2. Create a parameter file and store it on your system, or copy an existing file from another location. The file can contain any of the following parameters:

SERVER_HOSTNAME

Required if you want to register CTE with a CipherTrust Manager.

REG_TOKEN

The registration token for the CipherTrust Manager with which you plan to register this client. Required for registration.

HOST_PROFILE

Specifies the client profile in the CipherTrust Manager that will be associated with this client. If this value is omitted, the CipherTrust Manager uses the default client profile.

TMPDIR

Specifies a custom temporary directory that the installer can use during the installation process. If this value is omitted, the installer uses the default temporary directory.

AGENT_HOST_NAME

FQDN of the host on which the CTE Agent is being installed. If this value is not specified, the installer uses the host's IP address.

AGENT_USEIP

Use the IP address of the protected host instead of host name. Used when AGENT_HOST_NAME is not supplied.

PKCS11_PASSWORD

The administrative password for the CTE Agent. Required if the CTE Agent is being registered with a CipherTrust Manager.

This password allows users to access encrypted data on the client if it is disconnected from the CipherTrust Manager.

AGENT_HOST_PORT

Specifies the port number this CTE Agent should use.

USEHWSIG

Set this value to 1 when you want to associate this installation with the machine hardware for cloning prevention.

ENABLE_LDT

Set this value to 1 to automatically enable and register CTE-LDT (Live Data Transformation) for this host on your key manager during the silent install.

ENABLE_ES

Set this value to 1 automatically enable and register CTE-Efficient Storage for this host on your key manager during the silent install.

CERT_FIELD_PARAM

Example 1: Registering with CipherTrust Manager

The following example contains just the required information for registration with CipherTrust Manager. In this case, the client will be registered with the CipherTrust Manager using its IP address instead of its host name:

```
SERVER_HOSTNAME=Key-Mgmt-Server.example.com  
REG_TOKEN=12345  
PKCS11_PASSWORD=ClientAdminP@$w0rd
```

Example 2: Registering with CipherTrust Manager

The following example specifies the required registration information, adds a host name and description, and enables hardware association. In this case, the client will be registered with the CipherTrust Manager using its host name instead of the IP address:

```
SERVER_HOSTNAME=Key-Mgmt-Server.example.com  
REG_TOKEN=12345  
PKCS11_PASSWORD=ClientAdminP@$w0rd  
AGENT_HOST_NAME=myagent.example.com
```

```
HOST_DESC="West Coast Server 12"  
USEHWSIG=1  
CERT_FIELD_PARAM="/C=US/ST=California/L=San Jose/O=Thales  
eSecurity/OU=Vormetric/CN=localhost/emailAddress=admin@thalegroup.com"  
SUBJECT_ALT_NAME_PARAM="DNS:www.thalesecurity.com,email:admin@thalesgroup.com"
```

3. Copy or mount the CTE installation file to the host system. The installation file is in the format `vee-fs-
<release>-<build>-<system>.bin`.

4. Run the installer using the following syntax:

```
# ./vee-fs-<release>-<build>-<system>.bin [-d <custom-dir>] -s <install-file>
```

where:

- `-d <custom-dir>` is an optional parameter that specifies the installation directory for CTE. If you omit this parameter, CTE is installed in `/opt/vormetric/DataSecurityExpert/agent/`.
- `-s <install-file>` indicates that you want to install silently using the installation options file `<install-file>`

For example, if the installation options file is called `/tmp/unattended.txt`, you would enter:

```
# ./vee-fs-7.2.0-128-rh8-x86_64.bin -s /tmp/unattended.txt
```

5. Verify the installation by checking CTE processes on the host:

- Run `vmc -v` to check the version of CTE matches that just installed.
- Run `vmsec status` to display CTE kernel status.
- Look at the log files in `/var/log/vormetric`, especially `install.fs.log.<date>` and `vorvmd_root.log`.

6. In CipherTrust Manager, change the client password using the manual password creation method. This password allows users to access encrypted data if the client is ever disconnected from the CipherTrust Manager. For details on changing the password, see the CipherTrust Manager documentation.

Guarding a Device with CipherTrust Manager

After you register a client with a CipherTrust Manager, you can create as many standard GuardPoints on the client as you need. These GuardPoints can protect an entire device or individual directories.

Note

For guarding using LDT on a local drive, or on a CIFS/Share drive, refer to the [CTE-Live Data Transformation with CipherTrust Manager](#) guide.

In order to guard a device or directory, you need to use the CipherTrust Manager Console to:

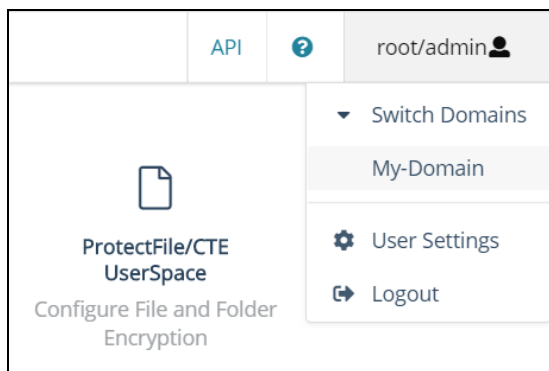
1. Access the CipherTrust Manager domain in which the client is registered.
2. Identify or create an encryption key that CTE will use to encrypt the data on the device or directory.
3. Identify or create a policy for the device or directory that specifies the access controls and the encryption keys to use for the device or directory.
4. Assign a GuardPoint to the device or directory.

The following example creates a simple policy and uses it to guard a directory on a registered client. For all of the following procedures, you must be logged into the CipherTrust Manager Console as a CipherTrust Manager Administrator, and you must be in the domain with which the client is registered.

For details about any of these procedures or the options for domains, encryption keys, policies, and GuardPoints, see the CipherTrust Manager documentation.

Access the CipherTrust Manager Domain

1. In a web browser, navigate to the URL of the CipherTrust Manager Console you want to use and log in with CipherTrust Manager Administrator credentials.
2. If the client you want to protect is registered to the default domain (root), proceed to ["Create an Encryption Key" below](#). If you need to change to a different domain, do the following:
 - a. In the top menu bar, click the user name **root/admin** on the right-hand side.
 - b. Select **Switch Domains**, then select the domain in which the client is registered.
 - c. The logged in user now shows the new domain name/user name.



Create an Encryption Key

Note

The following procedure is based on CipherTrust Manager version 2.2. If you are using a different version, see the CipherTrust Manager documentation for the version that you are using.

1. From the Products page in the CipherTrust Manager Console, click **Keys** in the left hand pane.

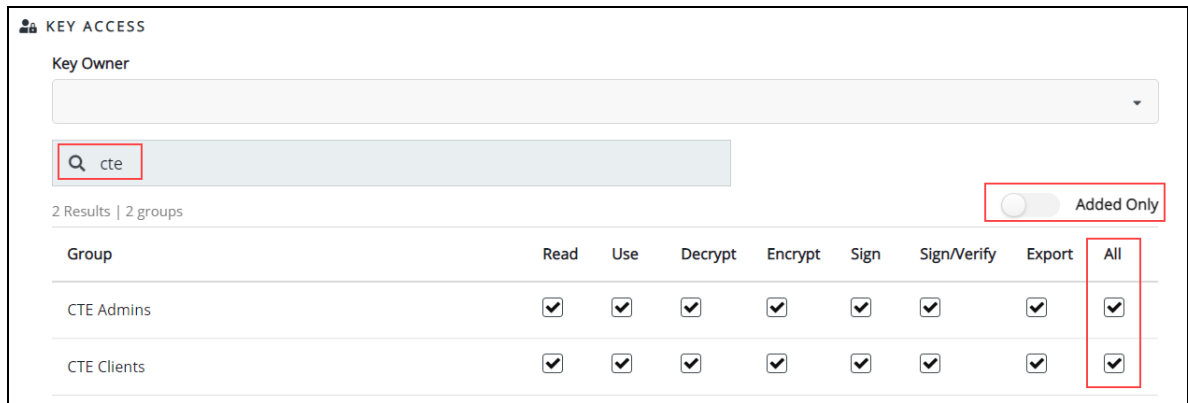
Tip: To navigate to the Products page from anywhere in the CipherTrust Manager Console, click the App Switcher icon in the top left corner.

2. Above the Key table, click **Create a New Key**.
3. In the **Key Name** field, add a name for the key. This name must be unique. For example, Simple-Key.
4. In the **Key Usage** section, make sure **Encrypt** and **Decrypt** are selected.
5. Click **Create**. CipherTrust Manager displays the properties for the new key.
6. In the general options area, enable the **Exportable** option.

You can also enable the **Deletable** option in this section if you want a CipherTrust Manager Administrator to be able to delete the key.

ID	2e58c582...61136313	Owner	Global	Object Type	Symmetric Key
UUID	e3ad9c3e...7fd47711	Created	05 Mar 2021, 05:13	Algorithm	AES
MUID	e3ad9c3e...f6333c9f	Last Modified	05 Mar 2021, 05:13	Size	256
KeyID	N/A	Exportable	<input checked="" type="checkbox"/>	Deletable	<input type="checkbox"/>

7. In the **Key Access** section, do the following:
 - a. In the Search Groups box, type "cte".
If no groups are displayed, make sure the **Added Only** option is *disabled*.
 - b. Click the **All** check box for both the CTE Admins and CTE Clients groups.



- c. When you are done, click **Update**.
 8. Click the **CTE** tab and set the following properties:
 - **CTE Versioned**: Specify whether the key is versioned. By default, the key is set as versioned.
For a standard policy, you should clear this check box. If you do not, the key will *not* appear in the keys list when you add the key rule to the standard policy.
 - **Persistent on Client**: Specify whether the key is stored in persistent memory on the client.
When the check box is selected, the key is downloaded and stored (in an encrypted form) in persistent memory on the client.
When the check box is left clear, the key is downloaded to non-persistent memory on the client. Every time the key is needed, the client retrieves it from the CipherTrust Manager. This is the default setting.
 - **Encryption Mode**: Encryption mode of the key. The options are:
 - CBC
 - CBC CS1
 - XTSEncryption using the XTS and CBC CS1 keys is known as enhanced encryption. For details, see the *CTE Agent for Linux Advanced Configuration and Integration Guide*.
- When you are done, click **Update**.

Create a Standard Policy

1. In the Applications page of the CipherTrust Manager Console, select the **Transparent Encryption** application.
2. In the sidebar on the Clients page, click **Policies**.
3. Click **Create Policy**. CipherTrust Manager displays the Create Policy Wizard.
4. On the General Info page, set the following options:

Field	Description
Name	A unique name for the policy. Make sure you use a name that is descriptive and easy to remember so that you can find it quickly when you want to associate it with a GuardPoint. This example uses "Simple-Policy".
Policy Type	The type of policy you want to create. In this example, we will create a Standard policy.
Description	A user-defined description to help you identify the policy later. For example: Standard policy for new GuardPoints
Learn Mode	Learn Mode provides a temporary method for disabling the blocking behavior of CTE policies. While useful for quality assurance, troubleshooting, and mitigating deployment risk, Learn Mode is not intended to be enabled permanently for a policy in production. This prevents the policy Deny rules from functioning as designed in the policy rule set. Ensure that the policy is properly configured for use in Learn Mode. Any Security Rule that contains a Deny effect must have Apply Key applied as well. This is to prevent data from being written in mixed states, resulting in the loss of access or data corruption. Apply Key will have no effect when combined with a Deny rule unless the policy is in Learn Mode.
Data Transformation	If you select Standard as the policy type, also select the the Data Transformation option to tell CTE that you want to change the current encryption key used on the data in the GuardPoint, or that you want to encrypt clear-text data for the first time. This option is only displayed for Standard policies.

When you are done, click **Next**.

5. On the Security Rules page, define the security rules that you want to use.

CipherTrust Manager automatically adds a default security access rule with an action of `key_op` and the effects `Permit` and `Apply Key`. This rule permits key operations on all resources, without denying user or application access to resources. This allows it to perform a rekey operation whenever the encryption key rotates to a new version.

To add additional security rules, click **Create Security Rule** and enter the requested information. For details about adding security rules, see the CipherTrust Manager documentation.

When you are done, click **Next**.

6. On the Create Key Rule page, click **Create Key Rule** and enter the following information:

Field	Description
Resource Set	If you want to select a resource set for this key rule, click Select and either choose an existing resource set or create a new one. Resource sets let you specify which directories or files will either be encrypted with the key or will be excluded from encryption with this key.
Current Key Name	Click Select to choose an existing key or create a new one. If the data has not yet been encrypted, select clear_key . Otherwise select the name of the non-versioned key that is currently being used to encrypt the data. In this example, select clear_key .
Transformation Key Name	Click Select to choose an existing versioned key or to create a new one. CTE uses the versioned key specified in this field to encrypt the data in the GuardPoint. If the data is currently encrypted, CTE decrypts it using the key specified in the Current Key Name field and re-encrypts it using the key specified in this field.

When you are done, click **Next**.

7. On the Data Transformation page, click **Create Data Transformation Rule** and enter the following information:

Field	Description
Resource Set	If you want to select a resource set for this key rule, click Select and either choose an existing resource set or create a new one. Resource sets let you specify which directories or files will either be encrypted with the key or will be excluded from encryption with this key.
Transformation Key Name	Click Select to choose an existing key or to create a new one. CTE uses the key specified in this field to encrypt the data in the GuardPoint. If the data is currently encrypted, CTE decrypts it using the key specified in the Current Key Name field and re-encrypts it using the key specified in this field. For this example, select the key Simple-Key you created in "Create an Encryption Key" on page 36 .

When you are done, click **Next**.

8. Click **Next**.

- On the confirmation page, review the information for the policy and click **Save**.

Create Policy

1 General Info 2 Security Rules 3 Key Rules 4 Data Transformation 5 Confirmation

Review the provided policy details.

1 General Info

Name: Simple-Policy
Policy Type: Standard
Description: Standard policy for new GuardPoints

2 Security Rules

Resource Set	User Set	Process Set	Action	Effect	Browsing
			key_op	permit,applykey	Yes
					Yes

3 Key Rules

Resource Set	Current Key Name
	clear_key

4 Data Transformation Rules

Resource Set	Transformation Key Name
	Simple-Key

Back Save

Create a GuardPoint

- Stop all applications that are accessing the device you want to protect. In this example, we are going to protect the following directories with the same policy and encryption key:
 - /dir/hr/files
 - /dir/accounting/files
 - /dir/shared/hr
 - /dir/shared/accounting

Tip: If you want to encrypt data without taking the device offline, you must use CipherTrust Transparent Encryption - Live Data Transformation.

- In the Applications page of the CipherTrust Manager Console, select the **CTE** application.
- In the Clients table, click on the name of the client you want to protect.
- Above the GuardPoints table, click **Create GuardPoint**.
- In the Create GuardPoint page:
 - In the **Policy** field, select the policy you created earlier.
 - In the Type field, select the type of device. You can guard a directory or a raw/block device. For this example, select **Auto Directory**.
 - In the **Path** field, enter the directories you want to protect with this policy or click **Browse** to select them from a explorer window.
If you want to enter multiple paths, put each path on its own line. For example:

Create GuardPoint [X]

Policy: *
Simple-Policy [Select]

Type: *
Auto Directory [v]

Path: *
/dir/hr/files/
/dir/accounting/files/
/dir/shared/hr/
/dir/shared/accounting/ [Browse]

- d. Click **Create**.
- e. If you want to use the same policy and GuardPoint type on another path, click **Yes** when prompted. Otherwise, click **No**. For this example, click No.

The CTE clients pull the GuardPoint configuration information from the CipherTrust Manager.

6. Type the following to transform the data:

```
# dataxform --rekey --print_stat --preserve_modified_time --gp <pathToGP>
```

When the data transformation has finished, applications can resume accessing the now-protected data. (See the “*CTE Data Transformation Guide*” for more information.)

Chapter 3: Special Cases for CTE Policies

This chapter section describes some CTE-specific configuration tasks related to configuring policies in the key manager. It contains the following topics:

Re-Enabling Automatic Signing for Host Settings	42
Restricting Access Overrides from Unauthorized Identities	42

Re-Enabling Automatic Signing for Host Settings

CTE blocks automatic re-signing of the host settings. Some users may have established procedures for updating system software that are based on the assumption that restarting the `vmd` will generate new signatures when signed software is updated. This is no longer true. However, you can re-enable automatic re-signing if your environment requires it.



CAUTION

Re-enabling the automatic regeneration of signatures exposes a potential security vulnerability for CTE Agents. When enabled, host setting binaries are re-signed when CTE receives a push from the associated key manager. If an attacker were to replace a binary with a Trojan, and then force a push from the key manager by, for example, restarting the CTE Agent, CTE could generate a signature for the malicious binary and pass it.

To re-enable automatic re-signing for host settings:

1. Change to the directory where the `agent.conf` file resides. For example, type:

```
# cd /opt/vormetric/DataSecurityExpert/agent/vmd/etc/
```

2. Edit the `agent.conf` file.

3. Change or add the following line:

```
AUTO_RESIGN_HOST_SETTINGS=TRUE
```

4. Save your changes and exit the file.

5. Restart the `vmd` to set the changes. Type:

```
# /etc/vormetric/secfs restart
```

6. Type the following to verify that the host settings is set to true:

```
# vmsec vmdconfig
```

Restricting Access Overrides from Unauthorized Identities

In some setups, system administrators can use the host settings `> |authenticator|` feature with `su` to change identities and gain access to restricted data. Now, you can instruct CTE to not trust any authentication attempt performed by certain identities by assigning restricted users to a user shell that CTE can block from authenticating other processes.

Any executable path that is marked with a `|path_no_trust|` host setting marks the process, and all child processes, as not trusted. Non-trusted processes are treated as "User Not Authenticated" to prevent access on user-based policies.

CTE prevents overrides from other host settings authenticators, using the `|path_no_trust|` status. If a user runs the `su` command from a non-trusted shell, that new shell is still marked as `|path_no_trust|`, even if `|authenticator|/usr/bin/su` is specified in the host-settings. The `|path_no_trust|` feature overrides any and all authenticators under host settings.

To restrict access overrides in CipherTrust Manager:

1. In the CipherTrust Manager Applications Page, click **CTE > Clients**.
2. Click on an existing client name to edit the client.
3. Click **Client Settings** tab.
4. Add the following to the client settings:

```
|path_no_trust|<path of the binary>
```

For example:

```
|path_no_trust|/bin/ksh
```

The above example indicates that no process under the kshell executable will be authenticated.

5. Click **OK**.

Chapter 4: Logs

This chapter contains the following sections:

Setting CTE Agent Logging Preferences	44
Audit Logs	46
Analyzing Audit log entries	46
File System Audit Log Effects Codes	46
Concise Logging	48

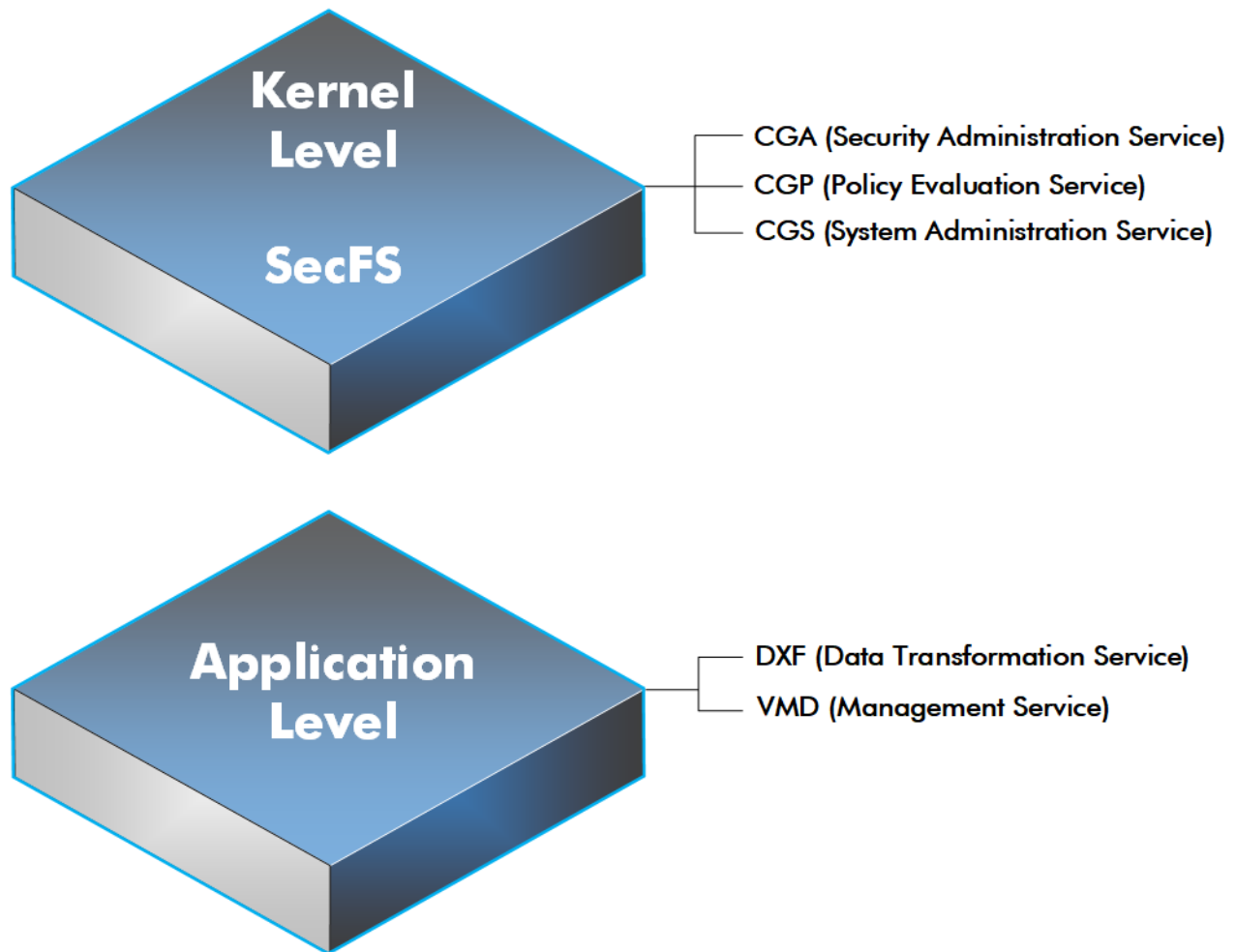
Setting CTE Agent Logging Preferences

You can configure the Agent process information that is entered into the Message Log. You can configure the process information globally, in which all the Agents that are added after the configuration change inherit the log attributes, while all current file system configurations remain intact. Alternatively, you can configure log attributes for individual Agent installations.

Always monitor log generation on new server and agent installations, and after changing logging preferences and options.

A variety of logging services are available and configured in the Log tab.

Logging Services



CTE log data may be sent to various different files such as:

- Sys log files, such as:

`/var/log/messages`

`/var/log/syslog`

Event log on

Note: The CM domain name can include spaces. However, Syslog does not allow spaces in header fields. Therefore, for Syslog purposes, the CTE client replaces the spaces with an underscore. For example: `My_Domain` instead of `My Domain`.

- CTE log files local to the agent, such as:

`/var/log/vormetric/vorvmd_root.log`

`C:\ProgramData\Vormetric\DataSecurityExpert\agent\log\vorvmd.log` (Windows)

- Uploaded to the Key Manager
- Uploaded to a Syslog server

Data Transformation log files are sent to:

- /var/log/vormetric/vordxf_path_usr.log

Audit Logs

Example audit log:

```
CGP2601I: [SecFS, 0] [AUDIT] Policy[allowAllOps_fs] User  
[root,uid=0,gid=0\root,bin,daemon,sys,adm,disk,wheel\] Process[/bin/cat] Action[write_  
app] Res[/opt/apps/apps1/doc/file2.txt] Key[aes128] Effect[PERMIT Code (1U,2U,3R,4M)]
```

Analyzing Audit log entries

The format of a File System Audit log entry is:

```
CGP2602I: [SecFS, 0] Level: Policy[policyName?] User[userID?] Process[command?] Access  
[whatIsItDoing?] Res[whatIsItDoingItTo?] Effect[allowOrDeny? Code (whatMatched?)]
```

Parameter	Description
Identifier	The TLA for the error message.
SECFS	Indicates that the message was generated by an Agent. You can enter <code>secfs</code> in the Search Message field in the Logs window to display the Agent policy evaluation and GuardPoint activity for all configured hosts.
Level	Indicates the importance of the message. For example, AUDIT indicates an informational message, whereas ALARM indicates a critical failure that you should not ignore.
Policy	Indicates the name of the policy that is being used to evaluate the access attempt.
User	Identifies the system user attempting to access data in the GuardPoint. It typically displays the user name, user ID, and group ID.
Process	Indicates the command, script, or utility being executed.
Access	Indicates what access is being attempted. Access may be <code>read_dir</code> , <code>remove_file</code> , <code>write_file_attr</code> , <code>write_app</code> , <code>create_file</code> , etc. These correspond to the Access methods that you configure in the policy. <code>Read_dir</code> corresponds to <code>d_rd</code> . <code>Remove_file</code> corresponds to <code>f_rm</code> , etc.
Res	Indicates the object/resource being accessed by the Process[].
Effect	Indicates the rule that matched and, based upon that rule, whether or not the CipherTrust Manager grants access. Access states may be either PERMIT or DENIED.

File System Audit Log Effects Codes

Codes are provided in the audit logs that identify actions by the policy enforcement engine. The code follows the number of the rule being processed.

Code	Definition
A	The Action component of a security rule failed to match.
M	All security rule components match and, unless overridden, the Effect for that security rule is applied.
P	The Process component of a security rule failed to match.
R	The Resource component of a security rule failed to match.
T	The time specified in the When component of a security rule failed to match.
U	The User component of a security rule failed to match.

Refer to the audit log example above:

- The first and second Security Rules fail because of a mismatch in the User component (1U, 2U).
- The third Security Rule fails because of a Resource component (3R) mismatch.
- All of the rules in the fourth Security Rule match (4M), and the actions defined in the policy, such as use an encryption key, are applied.

Concise Logging

Thales's standard operational logging sends audit messages for each file system operation each time a file is opened, read, updated, or written. Thales's standard logging can generate high volumes of log data. Most of these messages might not be useful or required by security administrators to monitor file system activity on the system.

Agent log data can be stored on the local host, sent to a syslog server, or uploaded to the Management Console. On an agent system, log entries can flood the local messages file or Event Log. Extreme logging can also affect network performance.

Concise Logging eliminates the following types of messages:

- Duplicate audit messages for each and every block read by the user or application. With Concise Logging, CTE only sends an audit message the *first* time a user or application performs a read/write activity. Subsequent read/write activity by that user or application is not logged.
- Audit messages that read the attributes, read the basic information of file-set attributes, and other event-based messages.
- Audit messages for directory open, read directory attributes, and directory close.

Using Concise Logging

You can enable and disable the Concise Logging option from the CipherTrust Manager for the following:

- All registered hosts in all domains
- A host that has registered with the CipherTrust Manager.

Considerations

- Concise Logging changes the set of log messages that are sent to Security Information and Event Management (SIEM) software systems. If this results in loss of data required for customer reports, then disable Concise Logging.
- Concise Logging is only supported by CTE `secfs`.
- Enable and disable Concise Logging on the host. CTE applies it to all GuardPoints and for all users on the host for which it is selected. There is no finer-grained control, such as per GuardPoint, user, or message type.
- When you enable this setting at the CipherTrust Manager level, it applies to all hosts in all domains, that are added to the CipherTrust Manager, but does not apply to any existing hosts. Hosts added after this setting is enabled inherit this setting. The default global setting is off.
- Do not use Learn mode with Concise Logging.

Configuring Concise Logging for CTE Clients or Client Groups with CipherTrust Manager

In CipherTrust Manager, when you create a Client Profile, you can select to Enable Concise Logging. Then, you can apply that Client Profile to a specific client, or to all clients in a Client Group. To enable Concise Logging, see the chapter *Managing Profiles* in the *CTE Administrator Guide*.

Chapter 5: Enhanced Encryption Mode

This chapter section describes the enhanced AES-CBC-CS1 encryption mode for keys. It contains the following topics:

- Compatibility 49
- Disk Space 50
- Encryption Migration 50
- File Systems Compatibility 50
- Container Compatibility 53
- Using the AES-CBC-CS1 Encryption Mode in CM 53
- Exceptions and Caveats 53
- Best Practices for AES-CBC CS1 Keys and Host Groups 53

The AES-CBC-CS1 encryption is superior to the existing AES-CBC mode because it uses a unique and unpredictable (random) IV (initialization vector) generated for each individual file. The per-file IV object is generated only at file creation time. It is stored as file metadata.

Note
AES-CBC-CS1 encryption does not require any additional license.

	AES-CBC	AES-CBC-CS1
Security Improvements		
Unique IV per-file	No	Yes
IV predictability	Yes	No
File System Support		
Local FS	EXT3/EXT4/XFS	EXT3/EXT4/XFS
Remote FS	NFS3/NFS4/CIFS	NFS3/NFS4
Block Device Support (secvm)	Fully supported	No. When a policy contains a key with CBC-CS1 encryption mode, the guarding fails on the CipherTrust Manager, and an error message displays.

Compatibility

- Starting with VTE for Linux version 6.1.0, CTE is backward compatible with, and fully supports, the existing AES-CBC mode for both new and existing datasets.
- Starting with VTE for Linux version 6.1.0, CTE fully supports AES-CBC-CS1 encryption for LDT and offline data transformation on CTE Linux environments.

Versions of VTE prior to version 6.1.0 are *not* backwards compatible with AES-CBC-CS1 encryption. On these earlier versions, attempting to guard a device using a policy containing an AES-CBC-CS1 key will fail.

- Protected hosts supporting AES-CBC-CS1 encryption can be added to host groups.

Difference between AES-CBC and AES-CBC-CS1

The two encryption modes are completely different from a file format standpoint.

- AES-CBC-CS1 encryption only applies to file system directories; AES-CBC encryption applies to both files and block devices.

Notes

- If you attempt to use an AES-CBC-CS1 key to guard a block device or partition, the guarding fails with an error reported on the CipherTrust Manager, similar to: Raw or Block Device (Manual and Auto Guard) GuardPoints are incompatible with Policy "policy-xxx" that contains a key that uses the CBC-CS1 encryption mode."
 - While AES-CBC-CS1 encryption is supported on both Linux and Windows environments, the file formats are incompatible. An encrypted file created with a specific AES-CBC-CS1 key on Windows cannot be read on Linux, even if that specific key were to be used and vice versa.
- AES-CBC-CS1 uses cipher-text stealing to encrypt the last partial block of a file whose size is not aligned with 16 bytes.
 - Each file encrypted with an AES-CBC-CS1 key is associated with a unique and random base IV.
 - AES-CBC-CS1 implements a secure algorithm to tweak the IV used for each segment (512 bytes) of a file.

Disk Space

Files encrypted with AES-CBC-CS1 keys consume additional disk space in contrast to files encrypted with AES-CBC keys. This is because AES-CBC-CS1 encryption requires file IVs to be created and persistently stored in contrast to AES-CBC encryption which does not consume any additional disk storage.

Therefore, administrators need to plan and provision additional disk capacity prior to deploying AES-CBC-CS1 encryption.

	AES-CBC	AES-CBC-CS1
Local Linux FS	No change to file size. No extended attribute allocation	Internal use of extended attribute per file. Extra 4KB increase in file size.
Remote Linux FS	No change to file size. No extended attribute allocation	Extra 4KB allocation in the form of an embedded header per file. With CTE guarding enabled, file size expansion is hidden.

Encryption Migration

You can use either LDT or offline dataxform to:

- Transform data encrypted by AES-CBC to AES-CBC-CS1 and vice versa.
- Transform AES-CBC-CS1 encrypted data to clear contents and vice versa.

File Systems Compatibility

On Linux, you can use AES-CBC-CS1 keys to guard currently supported file systems.

AES-CBC-CS1 encrypted files on Linux remote file systems like NFS and CIFS increase the file size compared to encrypted files on Linux local file systems which retain the original file size.

AES-CBC-CS1 encrypted files on Linux local file systems, in conjunction with LDT policies, can result in additional space consumption. Unlike the current AES-CBC encryption where encrypted files on all file systems, both remote or local, have the same file format, AES-CBC-CS1 encrypted file formats differ based on whether or not they were created on local or remote file systems.

AES-CBC-CS1 files on Linux remote file systems such as NFS and CIFS embed the IV in a 4K-byte header within the file. When these files are guarded, CTE masks the file header -- to applications and system utilities. The expanded file is only apparent when CTE guarding is disabled.

Note

The remote file system must have enough extra space to store the extra 4K bytes of the embedded header. You can run the following script in the GuardPoint to identify how much space to reserve before data transformation:

```
x=$(find . -type f | wc -l); y=$(echo "$x * 4 /1024" | bc); echo ${y}MB
```

File System Requirements

Unlike with AES-CBC encryption, files encrypted with AES-CBC-CS1 on remote file systems cannot be copied over to local file systems in the absence of CTE guarding. Similarly, AES-CBC-CS1 encrypted files on local file systems cannot be copied over to remote file systems in the absence of CTE guarding.

The fundamental reason for this incompatibility is the usage of extended attributes on local file systems to store the IV, in contrast to its storage as a part of the file metadata on remote file systems. This is why files cannot be transferred across these file system boundaries in the absence of CTE guarding.

	AES-CBC	AES-CBC-CS1
EXT3/EXT4 on RHEL6	None	EXT3/EXT4 must be mounted with <i>user_xattr</i> mount option
EXT3/EXT4 on other Linux distros	None	No limitation
XFS on Linux	None	No limitation
NFS v3,v4/CIFS on Linux	No limitation	No limitation

Samba Share

The remote Samba share server does not support ADS so you cannot use the CBC-CS1 key type on these GuardPoints.

Storing Metadata

AES-CBC-CS1 encrypted files on Linux store the base IV of a file in either the extended attributes or in the file metadata:

- On local FS (EXT3/EXT4/XFS), Linux saves it as an extended attribute associated with the file. The underlying file system requires that you mount it with the extended attribute mount option.
- On remote FS (NFS and CIFS), Linux saves it in the embedded header of the file.

To get the value of the base IV, type:

```
# voradmin secfs iv get <file-name>
```

Note

The base IV of a file is protected. It cannot be set/modified/removed by commands and applications. However, if a GuardPoint is unguarded, the files in the GuardPoint are no longer protected. An adversary can then corrupt the content of the files, as well as the IVs.

AES-CBC-CS1 depends on the physical file system's support for extended attributes in a manner similar to the CipherTrust Transparent Encryption - Live Data Transformation feature.

Missing IV file

If the IV for a file is missing, or CTE is unable to read the IV, then CTE denies access to the file. This access denied message may trigger an application to display an error message. This message may vary from application to application.

	AES-CBC	AES-CBC-CS1
Local Linux FS	No change	Internal extended attribute for each file
Remote Linux FS	No change	4KB embedded header for each file

HDFS

The AES-CBC-CS1 key is compatible with current Hadoop File System support.

Backups

Backups and other data protection utilities should take into account the extended attributes present in each AES-CBC-CS1 encrypted file on a Linux local file system to ensure that they are safely backed up. An AES-CBC-CS1 encrypted file whose IV is corrupted, renders the files to be corrupted and therefore unreadable. Hence all data protection software must preserve the file's extended attributes.

CTE Linux can inspect a file's IV using the following command:

```
# voradmin secfs iv get file
```

On Linux, the backup utility specified in the guarding policy should automatically backup/restore extended attributes as well. For example, you must use the options to preserve extended attributes when running `cp` or `rsync.normal`.

Due to the different file formats, the backup/restore across the local and remote file systems are not allowed. If you want to backup a GuardPoint from a local directory, you must restore it to a local directory. If a GuardPoint is backed up on a remote file system, you must restore it to a remote system.

	AES-CBC	AES-CBC-CS1
General backup utility requirement (all platforms)	Backup utility defined in guarding policy with clear view	Backup utility defined in guarding policy with clear view
Special requirement for backup/restore local fs on Linux	No	Backup utility must be run with user extended attribute enabled

	AES-CBC	AES-CBC-CS1
Special requirement for backup/restore remote fs on Linux	No	No
On Linux, backup local fs and restore to remote fs	Allowed	Not allowed, the restored files cannot be accessed with I/O error
On Linux, backup remote fs and restore to local fs	Allowed	Not allowed, the restored files cannot be accessed with I/O error

Container Compatibility

The CBC-CS1 key is compatible with current Docker and OpenShift support.

Using the AES-CBC-CS1 Encryption Mode in CM

When you create a key in CTE, you enable Encryption Mode by selecting CTE Key Properties. See *Creating a New Key* in the [Managing Policies](#) chapter in the [CTE Administrator Guide](#).

Exceptions and Caveats

Note the following when using AES-CBC-CS1 keys.

Ensure User Extended Attributes are Enabled on RHEL 6

On RHEL6, EXT3/EXT4 are not mounted with user extended attributes enabled, by default. If a GuardPoint is on EXT3/EXT4, then remount EXT3/EXT4 with `user_xattr` as an option. Otherwise, guarding fails with the error "Extended attribute not enabled for GuardPoint."

Guarding Existing Files Without Data Transformation

You must convert an existing file with clear text through offline data transformation or LDT. If you do not transform the file, then after you guard using an AES-CBC key, the file displays garbled characters.

If you use an AES-CBC-CS1 key, access to the file is blocked with an I/O error.

Best Practices for AES-CBC CS1 Keys and Host Groups

In a host group, do not deploy policies associated with AES-CBC and AES-CBC CS1 keys unless all hosts are running VTE for Linux version 6.1.0 or CTE version 7.0.0 or later.

Chapter 6: CTE and systemd

CipherTrust Transparent Encryption (CTE) for Linux is integrated with the `systemd` framework. To ensure that applications start after the CTE agent starts at startup, you must modify `systemd`. This is also true when the CTE agent is started and stopped manually.

This section contains the following topics:

- Overview of CTE and systemd 54
- CTE Agent Control Changes on systemd 54
- CTE Configuration Changes Required on systemd 55
- Supported Use Cases 58

Overview of CTE and systemd

You can use `systemd` to configure dependent applications to start up or shut down in the proper order when CTE starts up or shuts down on a live system. If applications start before CTE starts, those applications may not be guarded. This applies to cases when you manually start or stop the system, such as when you upgrade CTE.

`systemd` replaces `init` in Linux as a system and service manager. Linux inherited `init` from the UNIX System V `init`. `systemd`. The file is a collection of daemons, libraries, and utilities to provide central management and configuration for certain Linux distributions (see "[Linux Distributions that Support CTE and systemd](#)" below). In addition to providing enhanced features and performance, `systemd` is backwards compatible with System V and Linux Standard Base `init` scripts.

Linux Distributions that Support CTE and systemd

The entries in the following table are Linux distributions that are compatible with CTE. The table shows which distributions implement `systemd`, and those that do not. See the *Compatibility Matrix for CTE Agent with CipherTrust Manager* or the *Compatibility Matrix for CTE Agent with Data Security Manager* for the Linux distributions and kernels supported with your CTE version and key manager.

Distributions that support systemd	Distributions that do not support systemd
RHEL 7 RHEL 8 SLES 12 SLES 15 Ubuntu 16.04 Ubuntu 18.04	RHEL 6 SLES 11

CTE Agent Control Changes on systemd

The commands to start, stop, restart, and check CTE status on `systemd` are shown in the following table.

Command	Command syntax for distributions that support systemd
Start	<code>/etc/vormetric/secfs start</code>
Restart	<code>/etc/vormetric/secfs restart</code>

Command	Command syntax for distributions that support systemd
Stop	<code>/etc/vormetric/secfs stop</code>
Check status	<code>/etc/vormetric/secfs status</code>

The normal states of CTE services on a system with one or more active GuardPoints is shown in the following list. It is normal for `secfs-init` and `secfs-fs` to be listed as active (exited).

- `secfs-init`: active (exited) state
- `secfsd`: active (running) state
- `vmd`: active (running) state
- `secfs-fs`: active (exited) state

Example

To check the status of CTE, type:

```
# /etc/vormetric/secfs status
secfs-init service: active (exited) since Tue 2017-09-26 09:04:21 PDT; 1 day 4h ago
secfsd service : active (running) since Tue 2017-09-26 09:04:21 PDT; 1 day 4h ago
vmd service : active (running) since Tue 2017-09-26 09:04:44 PDT; 1 day 4h ago
secfs-fs service : active (exited) since Tue 2017-09-26 09:04:45 PDT; 1 day 4h ago
```

CTE Configuration Changes Required on systemd

The following table is a high-level overview of the required changes in `systemd` unit configuration files to control the applications that must be in sync with CTE during system startup and shutdown.

Typical applications that require `systemd` changes include `postgres`, `httpd`, `mongodb`, `mysqld` and `mariadb`. For example, `mongodb` requires that CTE to be running before `mongodb` starts.



CAUTION

Configuring the dependencies as recommended here mean that whenever you stop CTE or if CTE stops unexpectedly, the dependent applications will automatically stop shortly before CTE stops. Be sure you understand the implications of this behavior on your production environment.

Task	For more information
Compile a list of your applications that use GuardPoints.	See the application examples above.
Shut down the applications that use GuardPoints.	"Adding Dependencies to systemd Unit Configuration Files" on page 57
Add the following lines to the <code>[Unit]</code> section of the <code>systemd</code> unit configuration file for each application: <code>After=secfs-fs-barrier.service</code> <code>BindsTo=secfs-fs-barrier.service</code>	"Adding Dependencies to systemd Unit Configuration Files" on page 57

Task	For more information
To the <code>Before=</code> line in the <code>[Unit]</code> section of the <code>secfs-fs-barrier.service</code> file, add the following: <code>saslauthd.service</code> An entry for each application you added to the <code>systemd</code> unit configuration file.	"Adding Applications to the secfs-fs-barrier.service File" on the next page
Add the following lines to the <code>[Unit]</code> section of the <code>saslauthd.service</code> configuration file: <code>BindsTo=secfs-fs-barrier.service</code> <code>After=secfs-fs-barrier.service</code>	"Adding Dependencies to the saslauthd.service File" on page 58
Force the system to read the changed <code>systemd</code> configuration files by typing <code>systemctl daemon-reload</code> .	"Adding Dependencies to the saslauthd.service File" on page 58
Restart the applications that you shut down to make these changes.	"Adding Dependencies to the saslauthd.service File" on page 58

About systemd Dependency Changes for Unit Configuration Files

The previous table describes the effect of the required unit configuration file changes that are described in the following sections. See the [systemd documentation](#) for more information about the `After=` and `BindsTo=` unit configuration options.



CAUTION

In some cases, the unit configuration file for an application other than CTE may be overwritten when the application is upgraded. After upgrading an application, verify that changes to that application's unit configuration file are still in place. Changes to the `secfs-fs-barrier.service` file are retained after you upgrade CTE.

The `secfs-fs-barrier.service` service ensures that CTE services and dependent applications start after CTE services and stop before CTE services as needed during system startup, shutdown, and when starting and stopping services on a live system. This special service manages the dependencies between applications and the `secfs-fs`, `secfs-init`, `secfsd`, and `vmd` services that make up the CTE agent:

File and Change	Purpose
File: Application-specific unit configuration file Add this line to <code>[Unit]</code> section: <code>After=secfs-fs-barrier.service</code>	In conjunction with the <code>BindsTo=</code> option, this option ensures that the application starts after the CTE agent on system startup.
File: Application-specific unit configuration file Add this line to <code>[Unit]</code> section: <code>BindsTo=secfs-fs-barrier.service</code>	If the CTE agent shuts down, the application also shuts down.
File: <code>secfs-fs-barrier.service</code> Add each application to the existing <code>Before=</code> line in the <code>[Unit]</code> section. Add <code>saslauthd.service</code> to the existing <code>Before=</code> line in the <code>[Unit]</code> section.	Ensure that the <code>secfs-fs-barrier.service</code> starts before the application services mentioned in the <code>Before=</code> line.

Location of Application Unit Configuration Files

To configure the proper startup order of services, modify the unit configuration file for applications that require CTE to be running. You must also modify the `secfs-fs-barrier.service` file. The location of these files varies between Linux distributions as described in the following table.

Distribution	Location of systemd unit configuration files
RHEL 7 RHEL 8 SLES 12 SLES 15	<code>/usr/lib/systemd/system/<application_name>.service</code>
Ubuntu 16.04 Ubuntu 18.04	<code>/lib/systemd/system/<application_name>.service</code>

Adding Dependencies to systemd Unit Configuration Files

If your system supports `systemd` (see "[Linux Distributions that Support CTE and systemd](#)" on page 54), before you can safely reboot your protected host or use the files in the GuardPoint, you must perform the following steps to set the proper CTE dependencies for your applications.

1. Compile a list of your applications that use GuardPoints.
2. Prepare users of those applications for the interruption in service required to make these changes.
3. Shut down each of the affected applications.
4. For each application, log in as root and open the unit configuration file for that application using a text editor such as `vi`. See the table above to determine the location of the `systemd` unit file.

For example, on RHEL 7, the unit configuration file for a hypothetical `exampled` application would be located in `/usr/lib/systemd/system/exampled.service`.

5. Locate the `[Unit]` area in the file and add the following two lines at the end of the `[Unit]` section:

```
After=secfs-fs-barrier.service
BindsTo=secfs-fs-barrier.service
```

For example, on RHEL 7, the `exampled` application unit configuration file might have the following existing `[Unit]` section:

```
[Unit]
Description=Example server
After=syslog.target
After=network.target
```

In this case, you would add the two new lines after `After=network.target`.

6. Save and close the unit configuration file.
7. Repeat steps 3–6 for each application.

Continue to "[Adding Applications to the secfs-fs-barrier.service File](#)" below to make necessary changes to `secfs-fs-barrier.service`.

Adding Applications to the secfs-fs-barrier.service File

After editing the `systemd` unit configuration file for each dependent application, you must also add each application to the `secfs-fs-barrier.service` file that is installed as part of CTE. The `secfs-fs-barrier.service` file ensures that CTE starts before dependent applications during boot and CTE stops before dependent applications

when the system is shut down.

1. As a root user, open the `secfs-fs-barrier.service` file in a text editor such as `vi`.
For example, on RHEL 7, the `secfs-fs-barrier.service` file is located in `/usr/lib/systemd/system/secfs-fs-barrier.service`.
2. Add `saslauthd.service` along with the names of the unit configuration files that you edited in the last section to the end of the “Before=” clause.

For example on RHEL 7, the `secfs-fs-barrier.service` file might contain the following before editing:

```
Before=postgresql.service httpd.service mongod.service mongod.service  
mysqld.service mariadb.service nails.service
```

To add an entry for `saslauthd.service` and two custom application services called `example1.service` and `example2.service`, you would edit the `Before=` line to look like this:

```
Before=postgresql.service httpd.service mongod.service mongod.service  
mysqld.service mariadb.service nails.service saslauthd.service example1.service  
example2.service
```

3. Save and close the `secfs-fs-barrier.service` file.

In the future, CTE will start before each application listed in the `Before=` line and those applications will stop before CTE stops.

Continue to the next section to make the necessary changes to `saslauthd.service`.

Adding Dependencies to the `saslauthd.service` File

The `saslauthd` service must start after the `secfs-fs-barrier` service.

1. As a root user, open the `saslauthd.service` file in a text editor such as `vi`.
For example, on RHEL 7, the `saslauthd.service` file is located in `/usr/lib/systemd/system/saslauthd.service`.
2. Locate the `[Unit]` area in the file and add the following two lines at the end of the `[Unit]` section:

```
After=secfs-fs-barrier.service  
BindsTo=secfs-fs-barrier.service
```

For example, on RHEL 7, the `saslauthd.service` file might have the following existing `[Unit]` section:

```
[Unit]  
After=syslog.target  
After=network.target
```

In this case, you would add the two new lines after `After=network.target`.

3. Save and close the `saslauthd.service` file.
4. To force the system to re-read the `systemd` configuration files that you changed in this section and the previous section, type `systemctl daemon-reload`.
5. Start each application that you shut down to make the configuration changes.

Supported Use Cases

CTE supports the following use case.

Manually Stop 3rd Party Applications on a Live System

The applications must already be configured as dependencies in the unit configuration files using the `BindsTo` clause (required). For more information, see ["Adding Dependencies to systemd Unit Configuration Files" on page 57](#).

Use the following commands to stop CTE on a live system with dependent applications running.

```
$# systemctl stop secfs-fs
$# /etc/vormetric/secfs stop

Stopping the CipherTrust Encryption Expert File System Agent.
Stopping Live Data Transform processes (if any).
WARNING: Following dependent services will be stopped in order to stop secfs
services.
isecesp.service
isectpd.service
mysql.service
These services are required to start manually.
Successfully stopped the Encryption Expert CTE Agent.
```



WARNINGS

- The 3rd party application services using the `BindsTo` clause that automatically stop must be restarted manually after an upgrade.
- Only guard points that are busy due to 3rd party applications are freed using the new use case. Any guard point that is busy due to other I/O operations must be stopped manually.

Chapter 7: Utilities for CTE Management

Thales provides a variety of utilities that augment the standard Linux utilities. This combination of tools helps administrators manage CTE. The following utilities are described in this chapter section:

secfsd Utility	60
vmsec Utility	65
Restricting Access Overrides with Client Settings	68
Using Advanced Encryption Set New Instructions (AES-NI)	69
vmd utility	69
Agent Health Utility	70
agentinfo Utility (Java version)	73
check_host Utility	73
register_host Utility	73
fsfreeze and xfs_freeze	74
Displaying Information for Nested File Systems with the DF tool	75

secfsd Utility

The `secfsd` utility displays the following attributes of CTE:

- GuardPoints defined in the *GuardPoints* tab
- Authentication parameters defined in the *Host Settings* tab
- Lock status set by enabling **FS Agent Locked** and **System Locked**
- Web destination and SSL certificate for uploading log entries
- Policies applied in the **GuardPoints** tab
- Status of required processes (`secfsd` and `vmd`)
- Version of `secfs`

The `secfsd` utility is also used to mount GuardPoints for Directory (Manual Guard). Normally, CTE automatically mounts the `secfs` file system when you apply a GuardPoint to a directory. On Linux, the `secfsd` utility is located in `<install_dir>/secfs/.sec/bin` and a symbolic link to this file is placed in `/usr/bin/secfsd`.

secfsd syntax

Command	Description
<code>-help</code>	display <code>secfsd</code> options
Status Options	
<code>-status guard [-v -tree]</code>	list all GuardPoints
<code>-status keys</code>	show current encryption key state
<code>-status auth</code>	list authentication settings
<code>-status lockstat</code>	show CTE lock status

Command	Description
-status logger	list logging details
-status policy	list configured policies
-status plist	list protected processes
-status devmap	list guarded devices
Manual GuardPoint options	
-guard path [container ID]	manually guard path
-unguard path [container ID]	manually unguard path
Version option	
-version	list version of kernel module <code>secfs2</code>
Encryption Mode option information	
crypto	Displays the encryption modes that are supported.
Configuration Mode option information	
config <config_param> <value>	Displays the encryption modes that are supported.

secfsd Examples

Display GuardPoint Information

To display the GuardPoint paths, applied policies, policy type, and guard status, use the `secfsd -status guard` command. For example:

```
# secfsd -status guard
GuardPoint  Policy                Type                ConfigState  Status  Reason
-----
/opt/apl/lib allow AllOps_fs      local            guarded     guarded  N/A
/dev/sdb    watchaccess_rd       rawdevice        guarded     guarded  N/A
/dev/sdc    watchaccess_rd       manualrawdevice  guarded     guarded  N/A
/dev/sdd    watchaccess_rd       manualrawdevice  unguarded   not guarded  Inactive
/opt/apl/tmp MSSQL00123          manual           unguarded   not guarded  Inactive
```

Column	Description
GuardPoint	Full path of the GuardPoint.
Policy	Name of the policy applied to the GuardPoint.
Type	Can be local, automount, manual, raw device, or manual raw device. Configured in the GuardPoints tab.
ConfigState	Guard status of the GuardPoint, as recognized by the key manager. It can be guarded or unguarded.

Column	Description
Status	Current guard status, as recognized by CTE. State can vary.
Reason	Additional information about the status, if any.

Notes

- Config State and Status can vary. As an example, if you apply a GuardPoint and someone is currently working in the GuardPoint, the policy cannot be applied at that time. In this case, the ConfigState is guarded and the Status is not guarded.
- When the user removes an auto-mounted GuardPoint from CipherTrust Manager, the CTE Agent is only deleted after the configured `autoofs` timeout expires. This timeout does not start until the GuardPoint is free.

The timeout can be changed in the `auto.master` file on the host.

Display GuardPoint Information in a Different Format

To display the same information in a block format, use the `secfsd -status guard -v` command. For example:

```
# secfsd -status guard -v
GuardPoint: 1
  Policy:      allowAllOps_fs
  Directory:   /opt/apps/apps1/tmp
  Type:        local
  ConfigState: guarded
  Status:      guarded
  Reason:      N/A
GuardPoint: 2
  Policy:      allowAllRootUsers_fs
  Directory:   /opt/apps/apps1/lib
  Type:        local
  ConfigState: guarded
  Status:      guarded
  Reason:      N/A
```

Display Host Settings

To display the SHA2 hash signature for each protected host setting, use the `secfsd -status auth` command. For example:

```
# secfsd -status auth
/bin/su 3E765375897E04C39AB17D4C755F50A35195535B6747DBA28DF9BD4AA672DF9
|authenticator|/usr/sbin/sshd
98FC599D459EDEA52A60AB394B394803B5DAB96B53148DC608732DDA6777FA1A
/usr/sbin/in.rlogind 5C9A0EDD8BF54AE513F039476D21B3032507CF957AA0CB28C368EB8AB6E684FB
/bin/login 0D2EE0B995A30AE382B4B1CA5104715FC8902F457D283BDABAAD857B09259956
/usr/bin/gdm-binary 363780522E3CCF9ABF559F059E437743F9F97BBBB0EE85769007A464AD696BD1
/usr/bin/kdm BAD41BBCDD2787C7A33B5144F12ACF7ABC8AAA15DA9FDC09ECF9353BFCE614B5
```

Display Key Status

To display the status of CTE keys, use the `secfsd -status keys` command. For example:

```
# secfsd -status keys
Encryption keys are available
```

Display Lock Status

To display the status of CTE locks, use the `secfsd -status lockstat` command. For example:

```
# secfsd -status lockstat
FS Agent Lock: false
System Lock: false
```

The value is **true** if the lock is applied. The value is **false** if the lock is not applied. **System Lock** corresponds to **System Locked** in the *Host* window. **FS Agent Lock** corresponds to **FS Agent Locked** in the *Host* window.

Note

Before you upgrade, remove CTE software, or change operating system files, the status of FS Agent Lock and System Lock must be false.

Agent Security Configuration Protection

The Agent lock directory, `/opt/vormetric/DataSecurityExpert/agent/secfs/.sec` contains secfs secret files, configuration files, host setting signatures, etc. Thales recommends protecting the directory whenever secfs is online.

Applying improved directory protection ensures that only CTE applications (`vmd`, `secfsd`, `voradmin`, etc.) can modify the `.sec` directory and the files in it. All users, including root, are denied read/write access to the files. They also do not have permissions to modify `conf` and `bin` directories, using other tools.

A new command has been created to protect the directory: `voradmin secfs config`

Syntax

```
# voradmin secfs config <configuration_parameter> <value>
```

Example

```
# voradmin secfs config pagecache_writeback 1
```

Previously, you would have had to use the following command to achieve the same results as the example above:

```
# echo 1 > /opt/vormetric/DataSecurityExpert/agent/secfs/.sec/conf/pagecache_writeback
```

Note

When CTE is upgraded to v7.2.0 from the previous release, it may display 'Permission Denied' warnings which display when files are removed from subdirectories of the `.sec` directory. You can ignore these warnings. They are harmless.

Display CTE Log Status

To display the status of CTE log service, use the `secfsd -status logger` command. For example:

```
# secfsd -status logger
Upload URL: https://vmSSA06:8444/upload/logupload,https://vmSSA07:8444/upload/logupload,
\
https://vmSSA05:8444/upload/logupload
Logger Certificate directory: /opt/vormetric/DataSecurityExpert/agent/vmd/pem
```

This command sequence returns the URL to which the log service sends log data. It also returns the directory that contains the CTE certificate. CTE uses the certificate to authenticate CTE when it uploads the log data to the CipherTrust Manager.

Display Applied Policies

To display the policies that are applied to CTE, use the `secfsd -status policy` command. For example:

```
# secfsd -status policy
Policy: enc-audit
Type: ONLINE
```

Display CTE Process Information

To display CTE processes, use the `secfsd -status pslist` command. This command shows the process number associated with each CTE process. To show the details about a specific CTE process, use the `ps -fp <process #>` command, where `<process #>` is the process number from the `secfsd -status pslist` command.

For example:

```
# secfsd -status pslist
Protected pid list:      739    731
# ps -fp 739
UID      PID    PPID  C   STIME          TTY  TIME   CMD
root     739    1     0   11:04:56      -   0:00  /opt/vormetric/ \
        DataSecurityExpert/agent/vmd/bin/vmd
```

Display CTE Version Information

To display CTE version information, use the `secfsd -version` command. For example:

```
# secfsd -version
version: 7.2.0.128
```

Display CTE Crypto Information

To display CTE support information for encryption modes, use the `voradmin secfs crypto` command. For example:

```
># voradmin secfs crypto
```

```
AES CBC, CBC_CS1, XTS modes are supported
Encryption key protection is supported
```

Manually Enable a GuardPoint in CipherTrust Manager

To manually enable a GuardPoint on a Linux host:

1. Click **CTE > Clients > <clientName> GuardPoints**
2. Click **Create GuardPoint**.
3. In the Policy field, select a policy.
4. Set Type to **Manual Directory**.
5. Click **Browse** and enter the GuardPoint path.
6. Click **Create**.
7. Log onto the system hosting CTE as the root user.

8. To manually enable the GuardPoint, use the `secfsd -guard <path>` command. For example:

```
# secfsd -guard /opt/apps/etc
secfsd: Path is Guarded
```

9. To verify the change, use the `secfsd -status guard` command. For example:

```
# secfsd -status guard
GuardPoint      Policy          Type           ConfigState    Status         Reason
-----
/opt/apps/etc   allowAllOps_fs manual         guarded        guarded        N/A
```

secfsd and Raw Devices

CTE only creates block devices. To display them, use the `ls -l /dev/secvm/dev` command. For example:

```
# ls -l /dev/secvm/dev
brw----- 1 root system 38, 1 Jan 29 16:37 hdisk1
brw----- 1 root system 38, 2 Jan 29 16:37 hdisk2
crw----- 1 root system 38, 3 Jan 29 16:37 rhdisk1
crw----- 1 root system 38, 4 Jan 29 16:37 rhdisk2
```

vmsec Utility

The `vmsec` utility allows you to manage security aspects of CTE on the host. On Linux hosts, the `vmsec` utility is located in:

```
/opt/vormetric/DataSecurityExpert/agent/vmd/bin/vmsec
```

vmsec Syntax

<code>checkinstall</code>	Show vmd kernel status
<code>challenge</code>	Enter the dynamic host password
<code>vmdconfig</code>	Display the vmd configuration
<code>check_hwenc</code>	Display kernel configuration
<code>hwok</code>	Report status of hardware signature
<code>passwd [-p <password>]</code>	Enter the static host password
<code>version</code>	Display CTE version

vmsec Examples

Display CTE Challenge String

To display a CTE password challenge string and enter the response string when the CipherTrust Manager is not network accessible, use the `vmsec challenge` command. This command displays a challenge string that you can send to your key manager administrator, who will then send you back the correct response information.

For example:

```
# vmsec challenge
Contact a Security Server administrator for a response.
Your host name is "Host120" Your challenge is: HPTQ-ZYLK
Response -> IHFY-W7WG-PDAO-QKKQ
```

Contact your key manager administrator and give them the challenge string. The administrator will give you the response string. Enter the response string in the **Response** field and press **Enter**. You have 15 minutes to enter the response string.

Tip

If you are using CipherTrust Manager, the ability to change the contact string will be added in a future release. For CipherTrust Manager, the contact string says "Contact your CM administrator".

Display CTE Status

This utility shows you if CTE is configured and running. If it is not running, you might need to start it manually. To display CTE status, use the `vmsec checkinstall` command. For example:

```
# vmsec checkinstall
The kernel component is installed and running.
```

Entering a Password

To enter the CTE static host password, use the `vmsec passwd` command. For example:

```
# vmsec passwd
Please enter password:
OK passwd
```

To enter CTE static host password on the command line so you can specify it in a batch script, specify the password using the `-p` option. For example:

```
# vmsec passwd -p myPass123
OK passwd
```

Display Kernel Status

To display the kernel status, use the `vmsec status` command. For example:

```
# vmsec status
FILE_FORMAT=2
FILE_GENERATED=08/27/2019 18:54:10
SA_QOS_STATUS=0
SA_HOST_CPU_UTIL=0
GP_1_Policy=27
GP_1_Dir=/gp
GP_1_lock=1
GP_1_type=1
GP_1_gtype=manual
GP_1_opt=gtype=2,policy=27,lock=1,type=1,dir=/gp/
GP_1_config_state=unguarded
```

```
GP_1_status=not guarded
GP_1_statuschk_tm=0-00-00 00:00:00
GP_1_config_op_retry_cnt=0
GP_1_config_op_attempt_tm=0-00-00 00:00:00
GP_1_flags=0
GP_1_reason=Inactive
GP_1_usage=free
TOTAL_GP=1
KEYS_AVAILABLE=TRUE
sdk_version=7.2.0.128
sdk_builddate=2019-08-19 15:16:46 (PDT)
coreguard_locked=false
system_locked=false
logger_upload_url=https://thl602-2114.qa.com:8447/upload/logupload,https://thl602-2116.qa.com:8447/upload/logupload
logger_cert_dir=/opt/vormetric/DataSecurityExpert/agent/vmd/pem
hostname_for_logging=vmd
QOS_PAUSED=false
vmd_STRONG_ENTROPY=false
vmd_URL=https://thl602-2114.qa.com:8446
vmd_SRV_URLS=https://thl602-2114.qa.com:8446, https://thl602-2116.qa.com:8446
vmd_PRIMARY_URL=https://thl602-2114.qa.com:8446
vmd_SUPPORTS_F8P=TRUE
vmd_SUPPORTS_CR256=TRUE
vmd_RANDHP=TRUE
learn_mode=false
concise_logging=false
vmd_listening_port=7024
vmd_initialization_time=2019-07-25 12:07:14.514
vmd_last_server_update_time=2019-07-25 12:12:04.747 policy_name_27=aes256
policy_version_27=0
policy_keyvers_27=0
policy_type_27=ONLINE
policies=27
logger_suppression_VMD=SUPPRESS
logger_intervaltime_VMD=600
logger_repeat_max_VMD=5
logger_suppression_POL=SUPPRESS
logger_intervaltime_POL=600
logger_repeat_max_POL=5
CONFIG_SA_1=27
TOTAL_CONFIG_SA=1
SA_1_NAME=27
SA_1_ALIAS=aes256
SA_1_TYPE=0
SA_1_REF=1
SA_1_HIP_REG_TIME=0
SA_1_FLAGS=1
TOTAL_SA=1
TOTAL_AUTH=0
AUTHBIN_1=|authenticator|/usr/sbin/sshd
B92A3D7EEF67B82230F7F76097D65159FCF5722A4154A249EFDC22C20F1B572C
AUTHBIN_2=|authenticator|/bin/login
4F210D1B83ACD79B006BCF7DB247ED002A45FC892C42720390BFA6AE21AEA8DC
TOTAL_AUTHBIN=2
```

Display CTE Build Information

To see the CTE build version, use the `vmsec version` command. For example:

```
# vmsec version
Version 6
7.2.0
2022-03-17 15:15:23 (PDT)
Copyright (c) 2009-2022, Thales. All rights reserved.
```

Display Contents of Conf files

To display the contents of the `agent.conf` and `.agent.conf.defaults` files, use the `vmsec vmdconfig` command. For example:

```
# vmsec vmdconfig
appender_syslogdest_Syslog_Appender_0=127.0.0.1
VMSDK_AGENT_CONFIG_FILE=/opt/vormetric/DataSecurityExpert/agent/vmd/etc/agent.conf
appender_layout_Syslog_Appender_0=Syslog_Layout
VMSDK_AGENT_VERSION=7.2.0
VMSDK_AGENT_BUILD_ID=28
PREV_URLS=https://srv.my.thales.com:8443
syslog_appender_myhost name=dev.my.thales.com
VMD_PORT=7024
...
...
appenders=Upload_Appender, File_Appender, Syslog_Appender_0
layouts=Upload_Layout, File_Layout, Syslog_Layout, Simple
CONNECT_TIMEOUT=180000
URL=https://srv.my.thales.com:8443
STRONG_ENTROPY=false
```

Restricting Access Overrides with Client Settings

CTE host/client settings are the means by which an administrator configures user authorization. Users with root privileges, on Linux systems, have the unfettered ability to override all file access and execution permissions imposed by the system.

CTE access control allows you to restrict privileges of users, groups, application processes and binaries, including root users and `setuid` programs. By default, CTE agent **does NOT** trust any process as authenticated. Any attempt to access a resource, by any process, will therefore be flagged with a “User Not Authenticated” notification. The CTE agent must be instructed to trust the authenticator process progeny. For example, the `/usr/sbin/sshd` is a process that can be trusted to authenticate the user to the system and to CTE.

In some setups, when editing a host, system administrators can use the **host settings** > `[authenticator]` feature with `su` to change identities and gain access to restricted data. You can instruct CTE to not trust any authentication attempt performed by certain identities by assigning restricted users to a user shell that CTE can block from authenticating other processes.

Any executable path that is marked with a `[path_no_trust]` host setting marks the process, and all child processes, as not trusted. Non-trusted processes are treated as “User Not Authenticated” to prevent access on user-based policies.

CTE prevents overrides from other host settings authenticators, using the `[path_no_trust]` status. If a user runs the `su` command from a non-trusted shell, that new shell is still marked as `[path_no_trust]`, even if `[authenticator]/usr/bin/su` is specified in the host-settings. The `[path_no_trust]` feature overrides any and all authenticators under host settings.

Note

Using `|trust|*` before a `|path_no_trust|` host setting no longer disables the `|path_no_trust|` host setting.

For example, the following host setting denies authentication for users accessing through sshd:

```
|trust|*  
|path_no_trust|/usr/sbin/sshd
```

To restrict access overrides in **CM**:

1. In the CM products page, click **Transparent Encryption > Clients**.
2. Click on an existing Client name to edit the host.
3. Click **Client Settings** tab.
4. Add the following to the settings:

```
|path_no_trust|<path of the binary>
```

Example

```
|path_no_trust|/bin/ksh
```

The above example indicates that no process under the kshell executable will be authenticated.

5. Click **Apply**.

Using Advanced Encryption Set New Instructions (AES-NI)

To verify AES-NI hardware support, type:

```
# vmsec check_hwenc
```

Unlike the `-c algo` command above, CTE does not have to be running for this command to execute. This command displays one of the following messages to stdout:

```
"AES-NI hardware encryption is supported on this system."
```

```
"AES-NI hardware encryption is not supported on this system. Will default to software encryption."
```

vmd utility

The `vmd` utility displays CTE software version information.

The `vmd` utility is located in `/opt/vormetric/DataSecurityExpert/agent/vmd/bin` and a symbolic link to this file is placed in `/usr/bin/vmd`.

Syntax

```
vmd [OPTIONS...]
```

```
-h show utility syntax
```

```
-v display CTE version
```

```
-f runs vmd in the foreground
```

Display the Installed Version

To display the installed CTE version, type:

```
# vmd -v
Version 6
7.2.0.128
2022-02-04 11:09:40 (IST)
Copyright (c) 2009-2022, Thales.. All rights reserved.
```

Agent Health Utility

The `agenthealth` utility validates:

- Super-user privilege
- CTE Agent installation
- CTE registration to CipherTrust Manager Server
- CTE processes/ modules that are running
- Available disk resources
- Current GuardPoints
- Tests if the agent can reach the GuardPoints
- CTE log directory resource status

This directory contains pending CTE log files for upload. This utility reports the size and number of pending files for upload. These text files are logs that contain vmd/SecFS information. They are regenerated whenever secfs restarts. If the number of files is unexpectedly large, this can indicate a problem.

The Agent Health check script

By default, the `agenthealth` script is installed in `/opt/vormetric/DataSecurityExpert/agent/vmd/bin`.

To run the `agenthealth` check script, type:

```
# ./opt/vormetric/DataSecurityExpert/agent/vmd/bin/agenthealth
```

System Response

```
Checking for super-user privilege ..... OK
CipherTrust Agent installation ..... OK
CipherTrust policy directory ..... OK
Registration to server ..... OK
Kernel modules are loaded ..... OK
VMD is running ..... OK
SECFSD is running ..... OK
dsm4209.sjinternal.com is resolvable ..... OK
dsm4209.sjinternal.com port 8446 is reachable ... OK
dsm4209.sjinternal.com port 8447 is reachable ... OK
Can communicate to at least one server ..... OK
VMD is listening on port 7024 ..... OK
Time of last update from server ..... 2021-07-07 15:47:08.290
Checking available disk space ..... OK
Checking logging space ..... OK
    Log directory is "/var/log/vormetric"
    File system for log data is "/", 48G free (5% full)
```

```
Log directory contains 9 of maximum 200 files (4% full)
Log directory contains 1 of maximum 100 Mbytes used (1% full)
Testing access to /media ..... OK
Testing access to /usr/data/sub1 ..... OK
[root@agt4206 bin]#
```

Agent Health Return Codes

Previously, the agent health return codes were only available in `/var/log/vormetric/agenthealth.log`. Now, the following options are also available through the help pages:

Help

This agent health script checks various facets of the CipherTrust agent to make sure that everything is functioning properly. Results are also logged to `/var/log/vormetric/agenthealth.log`.

Syntax

```
# ./opt/vormetric/DataSecurityExpert/agent/vmd/bin/agenthealth --help
```

Return Codes

Use the return code option to get a list of the return codes and what they mean. The codes are returned if the Agent is not running.

Syntax

```
# ./opt/vormetric/DataSecurityExpert/agent/vmd/bin/agenthealth --return_codes
```

Response

Return Code	Definition
EPERM	User is not root.
ENOENT	One of the programs used in this script does not exist. See <code>/var/log/vormetric/agenthealth.log</code> for which program is missing.
ENOPKG	Agent software is not properly installed. Agent configuration directory is missing or corrupt. See <code>/var/log/vormetric/agenthealth.log</code> for more details.
EPROTO	Agent is not registered to a key manager. Register the agent to a key manager and try again. Try the wait option if the agent has never started correctly after registration. See <code>/var/log/vormetric/agenthealth.log</code> for more details.
EIO	Kernel modules are not loaded. To load a kernel module, type: <code>/etc/vormetric/secfs start</code>
ESRCH	VMD is not running. To start vmd manually, type: <code>/usr/bin/vmd</code>
SECFSD	Secfsd is not running. To start the secfsd manually, type <code>/usr/bin/secfsd</code>
EHOSTUNREACH	Unable to reach the Key Manager. Check network connectivity.
ECONNREFUSED	VMD is not listening. VMD did not finish initialization. See <code>/var/log/vormetric/vmd.log</code>

Return Code	Definition
EWOULDBLOCK	VMD is attempting to connect to the Key Manager but has exceeded the designated wait time. Check <code>/var/log/vormetric/vmd.log</code> to fix any issues and retry.

Wait Time

Use `--w` to set a maximum wait time in seconds. The minimum is 10 seconds to test for the VMD to Key Manager initial contact. The default setting is 0, which means that there is no wait. Maximum is 1200 seconds.

Syntax

```
[root@agt4206 bin]# ./opt/vormetric/DataSecurityExpert/agent/vmd/bin/agenthealth --w  
<value>
```

Example

```
[root@agt4206 bin]# ./opt/vormetric/DataSecurityExpert/agent/vmd/bin//agenthealth --w 60
```

Response

```
Checking for super-user privilege ..... OK  
CipherTrust Agent installation ..... OK  
CipherTrust policy directory ..... OK  
Registration to server ..... OK  
Kernel modules are loaded ..... OK  
VMD is running ..... OK  
SECFSD is running ..... OK  
dsm148.i.vormetric.com is resolvable ..... OK  
dsm148.i.vormetric.com port 8446 is reachable .... OK  
dsm148.i.vormetric.com port 8447 is reachable ... OK  
Can communicate to at least one server ..... OK  
VMD is listening on port 7024 ..... OK  
Time of last update from server ..... 2021-08-18 10:34:56.665  
Checking available disk space ..... OK  
Checking logging space ..... OK  
Log directory is "/var/log/vormetric"  
File system for log data is "/", 29G free (23% full)  
Log directory contains 1 of maximum 200 files (0% full)  
Log directory contains 0 of maximum 100 Mbytes used (0% full)
```

If the customer did not use the wait time options, the output would look similar to the following:

```
# /opt/vormetric/DataSecurityExpert/agent/vmd/bin/agenthealth
```

```
Checking for super-user privilege ..... OK  
CipherTrust Agent installation ..... OK  
CipherTrust policy directory ..... OK  
Registration to server ..... OK  
Kernel modules are loaded ..... OK  
VMD is running ..... OK  
SECFSD is running ..... OK  
Can communicate to at least one server ..... FAILED  
For more information consult the log file /var/log/vormetric/agenthealth.log
```


agentinfo Utility (Java version)

The `agentinfo` utility collects system and CTE configuration data. The `agentinfo` utility is used to take a configuration snapshot of the system that you will send to Thales Customer Support to debug an issue, (This section describes the Java version.)

The `agentinfo` utility is a Java Script file. You can open it in a text editor to see specific functions.

The `agentinfo` utility displays status information on the screen and outputs the results to a compressed tar file. The compressed tar file name format is `ai.<os_name_ver>.qa.com.tar.gz` and it is located in the current working directory.

To create an `agentinfo` file, type:

```
# /opt/vormetric/DataSecurityExpert/agent/vmd/bin/agentinfo
```

check_host Utility

If a CTE software installation fails during the certificate generation and exchange stage, use the `check_host` utility to list the network addresses for the host. The utility checks network interfaces, `/etc/hosts`, DNS, and so on, to compare, test, and evaluate possible addresses for the host, and weights them based upon their network efficiency. FQDNs are the most preferred and stand-alone IP addresses are the least preferred. Some applications, such as silent-mode installation, use `check_host` to determine the best host address to submit to the CipherTrust Manager during registration.

Run the `check_host` utility on a system that is hosting CTE to display available network host names, FQDNs, and IP numbers for the host.

Type:

```
# /opt/vormetric/DataSecurityExpert/agent/vmd/bin/check_host
```

check_host Syntax

```
check_host [[-h | -i | -a ] [-s name]] |  
-l name:port[,name:port] | -r name
```

-h	Print the best host name for this machine
-i	Print the best IP address
-a	Print all the host names and IP addresses
-s	The name of the server (optional hint)
-r	The name of the server for name resolution checks
-l	The name and port of the server for listening checks

register_host Utility

Use the `register_host` utility to create certificate requests, exchange certificates between the CipherTrust Manager and the host, and to register CTE on the CipherTrust Manager. After the host is registered, you can configure CTE, apply GuardPoints, or perform database backups. Run the `register_host` utility in text mode on a terminal window.



CAUTION

The default host registration timeout is 10 minutes. If the host is unable to reach the CipherTrust Manager within the allotted period because of an extremely slow network connection, set the `REGISTER_HOST_TIMEOUT` environment variable to extend the registration timeout. The variable value is an integer expressed in seconds. You might also have to extend the default TCP timeout.

fsfreeze and xfs_freeze

Users can freeze, snapshot, and unfreeze a file system with an SecFS GuardPoint using `fsfreeze|xfs_freeze` for both XFS and EXT3/4.

SecFS supports freezing with `fsfreeze|xfs_freeze` or by any other program issuing the same type of requests. Freezing SecFS results in freezing the underlying file system, as well as the primary file system.

Restrictions

There are restrictions for using `fsfreeze|xfs_freeze` support with CTE.

Platform Restrictions

The following platform restrictions occur with CTE and `fsfreeze|xfs_freeze`:

- CTE supports the `fsfreeze|xfs_freeze` utility for freezing SECFS GuardPoints on all Linux distributions for kernels ≥ 3.0 for Redhat, SLES, and Ubuntu platforms on EXT3/EXT4/XFS file systems. (Earlier Kernels do not contain the proper freeze_super VFS code).

Target Restrictions

The expected target of the `fsfreeze|xfs_freeze` command is the path of the GuardPoint.

For example, if `/dev/sdb` is mounted as `ext4` on `/data` and CTE contains the GuardPoint: `/data/protected`, then the target of `fsfreeze` must be `/data/protected`, not `/data`.

Valid: `# fsfreeze -f /data/protected`

Not valid: `# fsfreeze -f /data`

File System Restrictions

The following file system restrictions occur with CTE and `fsfreeze|xfs_freeze`:

- If multiple GuardPoints exist on the same file system, you only need to freeze one.

For example, if `/dev/sdb` is mounted as `ext4` on `/data` and the CTE GuardPoints are `/data/protected1` and `/data/protected2`, then issuing:

```
# fsfreeze -f /data/protected1
```

freezes `/data/protected1`, `/data/protected2` and the underlying `ext4` file system.



CAUTION

Do not unguard a GuardPoint, or restart the CTE Agent, while the file system is frozen. The only action permitted on a frozen file system is taking a snapshot or backing up.

- If you try to freeze `/data/protected2` after freezing `/data/protected1`, it returns as busy
- If you are not permitted to freeze one GuardPoint, then you cannot freeze any GuardPoints

LDT Restrictions

- You cannot freeze a file system while it is undergoing an LDT rekey operation. If it detects a rekey, the freeze returns as busy
- You cannot start an LDT rekey on a frozen file system

Offline Data Transformation Restrictions

Do **NOT** use `fsfreeze|xfs_freeze` while an offline transform policy is in effect.

Displaying Information for Nested File Systems with the DF tool

When a file system mounts on top of another file system, the command `df -a` does not display the attributes for the covered file system.

On Linux environments, this is the expected behavior for any file system that is overlaid by another file system.

The `secfs` driver properly handles the call, which is made by the `statfs` system call, which is issued by the `df` tool. When the system call returns, its structures are correctly populated with the details of the nested file system. On examining the source of the `df` tool, it is found that when the `-a` switch is on, it nullifies the stats which are received for overlaid mounts. When the `-a` option is not enforced, the stats are maintained.

Issue the `df` command for a specific mount point, for example `df /xfs/nested-xfs` (where `/xfs` is a `secfs` GuardPoint). It works correctly.

Chapter 8: Installing CTE on Hadoop

This chapter section describes how to protect an HDFS cluster with CTE. It contains the following topics:

Overview	76
Implementing CTE on HDFS	77
HDFS Upgrade with CTE	84
CTE Installation and Configuration	89
Deleting Metadata in HDFS when Migrating Out of LDT	93

Overview

The Hadoop Distributed File System (HDFS) is a file system that supports large files and directory structures distributed across hundreds, or even thousands, of commodity DataNode hosts in a cluster. Previously, CTE could only protect directories and files on the *local file system* rather than the actual *HDFS* files and directories. Now, CTE can protect *HDFS* files and directories.

A CipherTrust Manager can:

- Define an encryption policy for HDFS files and directories in HDFS name space.
- Selectively encrypt HDFS folders with different keys providing multi-tenancy support.
- Define user-based I/O access control rules for HDFS files in HDFS name space.

At the heart of an HDFS cluster is the *NameNode* that provides the framework to support a traditional hierarchical file and directory organization. The NameNode is a master server that manages the HDFS name space and regulates access to files by clients.

HDFS files are split into one or more data blocks that are distributed across *DataNode* hosts in a cluster. The NameNode maintains the namespace tree and the mapping of data blocks to DataNodes. To deploy CTE, install CTE on all of the NameNode and DataNode hosts in a cluster.

Overview of CTE on HDFS

The following sections list the high-level steps for implementing CTE protection on your HDFS. The process requires that the HDFS Administrator and CipherTrust Manager to work in tandem to complete separate tasks.

You can keep the HDFS cluster alive a active if you enable HDFS data replication and activate the nodes individually. Following are the high-level steps:

HDFS Administrator

1. Compile a list of directories specified by `dfs.datanode.data.dir`. If these directories do not already exist in the NameNode local file system, create them.
2. Pass the directory list to the CipherTrust Manager.
3. Ask CipherTrust Manager to:
 - a. Add the NameNode to the HDFS host/client group.
 - b. Create a GuardPoint for the HDFS host/client group on each of these directories.

Administrator

1. Create an HDFS host/client group to contain the HDFS nodes.
2. Create a host/client group GuardPoint on each of the datanode directories obtained from the previous step.
3. Add the NameNode to the HDFS host/client group.

HDFS Administrator

1. For each DataNode, take the node offline and perform a data transformation.
2. Ask the Administrator to add the DataNode to the host group. (After the DataNode is added to the host group, it can be brought online.)
3. Repeat this process until all of the nodes have been encrypted and added to the HDFS Host Group.
4. Modify the host group policies to protect specific HDFS files and directories, as needed.

Implementing CTE on HDFS

This section describes how to implement CTE protection on your HDFS NameNode or DataNode. If you enable HDFS data replication, protecting one node at a time allows you to maintain the HDFS cluster. When all of the nodes are configured, you can create GuardPoints on specific HDFS files and directories.

These instructions require that the HDFS Administrator and Administrator work in tandem to complete separate tasks.

Note

The instructions below assume that each NameNode and DataNode exist on their own separate host. If you have a NameNode and DataNode on the same host, see ["Implementing CTE on HDFS on a Single Host" on page 83](#).

CTE on HDFS Implementation Assumptions

- You have installed, configured and registered CTE on all of the NameNodes and DataNodes in the Hadoop cluster.
- The HDFS Administrator has knowledge and experience with HDFS and Ambari.
- The CipherTrust Manager Administrator has knowledge and experience with CTE.
- The two can work and communicate in tandem with each other.

Create an Encryption Zone in HDFS Name Space for AWS EMR

HDFS requires the following manual steps if you want to have an encryption zone in the HDFS name space for AWS EMR, (Elastic MapReduce).

1. Add the following properties to `hdfs-site.xml`.

Note: The default `.sec` folder is `/opt/vormetric/DataSecurityExpert/agent/secfs/.sec`

```
<property>
  <name>dfs.vte.ioctl.lib</name>
  <value>vorhdfs</value>
</property>
<property>
  <name>dfs.vte.rename.check</name>
  <value>>true</value>
</property>
<property>
  <name>dfs.vte.ioctl.device</name>
  <value><.sec folder name, up to the CTE installation location></value>
</property>
```

2. Save the file.
3. Restart HDFS NameNode and DataNode services.

Using the Original Information from HDFS

Update or add the following properties to `hdfs-site.xml` if you want CTE to use the original user information from HDFS.

1. Add the following properties to `hdfs-site.xml`.

```
<property>
  <name>dfs.block.access.token.enable</name>
  <value>>true</value>
</property>
<property>
  <name>dfs.client.read.shortcircuit</name>
  <value>>false</value>
</property>
<property>
  <name>dfs.vte.user.push</name>
  <value>>true</value>
</property>
```

2. Save the file.
3. Restart HDFS NameNode and DataNode services.

Create an HDFS Host Group and GuardPoint in CipherTrust Manager

After configuring the NameNodes, the next steps in activating CTE on HDFS is for the security administrator.

To create a client group:

1. Open the **Transparent Encryption** application.
2. Click **Clients > Client Groups**.
3. Click **Create Client Group**. The Create Client Group dialog box displays.
4. Enter a unique **Name** for the client group.

5. Select the **Cluster Type**: HDFS
 - a. **NON CLUSTER**: Creates a non-clustered client group.
 - b. **HDFS**: Creates a clustered client group. An HDFS client group is required to apply GuardPoints on CTE clients in an HDFS cluster.
6. (Optional, displayed if a profile already exists) From the Client Profile drop-down list, select the desired client profile. The default profile is DefaultClientProfile.
7. (Optional) Provide Description to identify the client group. The maximum length can be 256 characters.
8. Click **Create**. The client group is created.
9. Add the NameNode obtained from the HDFS Admin to the HDFS Client Group.

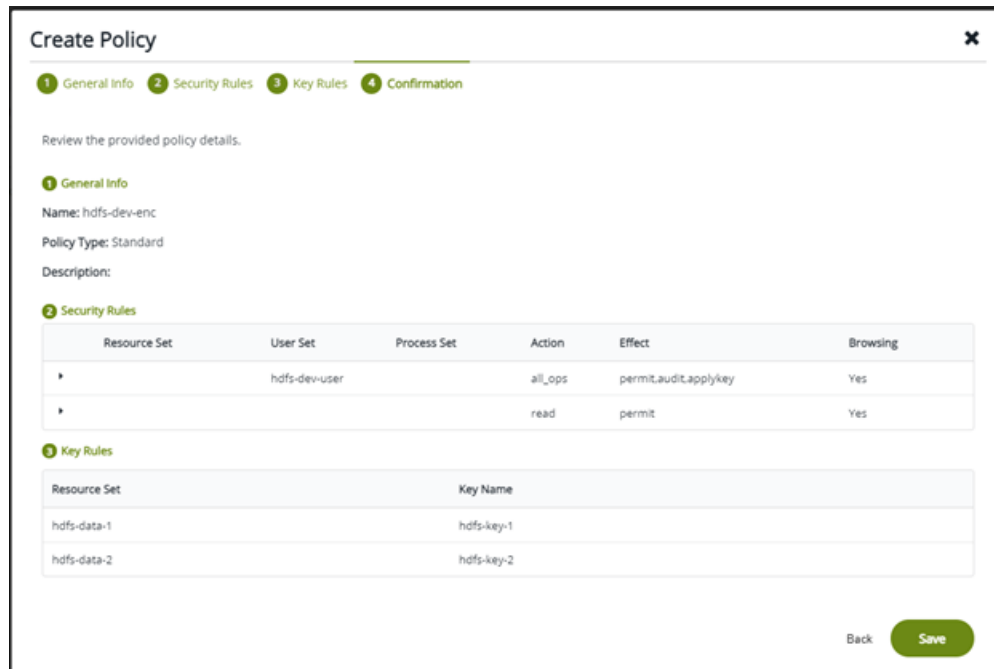
Note: Thales highly recommends using Auto Guard for HDFS. You can use manual guards, but this might result in data corruption if some nodes in a running cluster are guarded, while others are not.

Notes for HDFS Cluster Policies

The following screenshot depicts a valid HDFS policy. You can build a similar policy for your system:

The policy uses the following rules:

- For the user set `hdfs-dev-user`, the action is `all_ops`, the effect is Audit, Apply Key, Permit
- For other users the action is `READ`, the effect is Permit
- For the resource set `hdfs-data-1`, the key is `hdfs-key-1`
- For the resource set `hdfs-data-2`, the key is `hdfs-key-2`



Additional Settings for HDFS (Linux Clients)

Depending on the Hadoop security authentication mode, additional settings are needed for CTE clients in an HDFS cluster. Add the following settings as appropriate.

Note

`/usr/jdk64/jdk1.8.0_40/bin/java` is the Java executable used to launch the HDFS services. Change the Java jdk path to reflect your end-user environment.

- Sample setup when `hadoop.security.authentication` mode is simple.

```
|authenticator+arg+=+class=org.apache.hadoop.hdfs.server.namenode.NameNode|/usr/jdk64/jdk1.8.0_40/bin/java
```

```
|authenticator+arg+=+class=org.apache.hadoop.hdfs.server.datanode.DataNode|/usr/jdk64/jdk1.8.0_40/bin/java
```

- Sample setup when `hadoop.security.authentication` mode is Kerberos.

```
|authenticator+arg+=+class=org.apache.hadoop.hdfs.server.namenode.NameNode|/usr/jdk64/jdk1.8.0_40/bin/java
```

```
|authenticator+arg+=+class=org.apache.hadoop.hdfs.server.datanode.SecureDataNodeStarter|/usr/lib/bigtop-utils/jsvc
```

Adding a New DataNode to a CTE-protected HDFS

Use the following procedure to add a new DataNode to a CTE-protected HDFS. If not followed, HDFS encrypted files could be exposed in cleartext.

Note

If you already have CTE installed on the cluster nodes before Ambari installs the Hadoop software, see ["Install CTE on the Cluster Nodes before Ambari Installs Hadoop"](#) below.

1. Install the HDFS client on the host. This option is available in Ambari when adding a new DataNode to the cluster.
2. Add the new node to the CipherTrust Manager database and make sure that host/client settings of the new node is the same as existing nodes in the cluster. See the *CTE Installation for Hadoop* chapter in the *CTE Installation and Configuration Guide*.
3. Install CTE on the new node, register to CipherTrust Manager, and run `config-hadoop.sh` to prepare the libraries. See the *Configuring Hadoop to use CTE* section in the *CTE Installation and Configuration Guide*.
4. Make sure that the data directories (specified in `dfs.datanode.data.dir` property) exist on the new node. They must have the same permission and ownership as the other existing nodes in the cluster. If necessary, create them.
5. Add the host/client to the HDFS Host Group that is guarding the cluster. This is important: Do not rely on the DataNode to create the data directories as the data replication can occur before the GuardPoints are in effect.
6. Add the DataNode service to the new node. Again this option is available through Ambari.
7. If using Kerberos, check that the keytab files are created correctly.
8. Start the DataNode service on the new node.
9. Execute some `hdfs dfs` shell commands to ensure that encryption/decryption of data works correctly.

Install CTE on the Cluster Nodes before Ambari Installs Hadoop

If CTE is already installed on the cluster nodes before Ambari installs the Hadoop software, Ambari can mistakenly pick up the `.sec` directory in configuration steps to store the HDFS data. Make sure the following properties do not contain the `.sec` directory:

- DataNode data directory
- NameNode data directory
- Secondary NameNode checkpoint directory
- Zookeeper directory
- `yarn.nodemanager.local-dirs`
- `yarn.nodemanager.log-dirs`
- `yarn.timeline-service.leveldb-timeline-store.path`
- `yarn.timeline-service.leveldb-state-store.path`

Note

This list is not exhaustive. Depending on the Hadoop ecosystem packages installed, there can be others.

Configure NameNodes

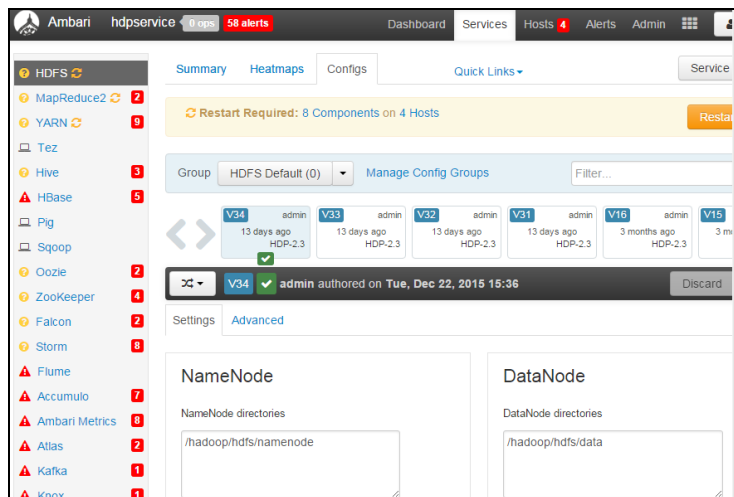
For both HDFS and Cloudera, you must configure the namenodes.

Configure the Ambari Hadoop NameNodes

The first step to implementing CTE on HDFS is for the **HDFS Administrator** to compile a list of the DataNode HDFS local file system directories, and create them on the NameNode local file systems. After this, the Administrator must add the NameNodes to an HDFS Host Group:

1. Compile a list of directories specified by `dfs.datanode.data.dir`. Obtain this from `hdfs-site.xml` or using Ambari go to:

HDFS > Configs > Settings > DataNode > DataNode directories



2. If these directories do not already exist in the NameNode local file system, create them on each NameNode in your Hadoop cluster.
3. Pass the following information to the Administrator:
 - The directory list and instructions to create a GuardPoint for the HDFS Host Group on each of these directories.
 - Instructions to add the NameNodes IP addresses or host names to the HDFS Host Group.

Create and Configure the Cloudera Hadoop Namenodes and Datanodes

Create the role groups CTE on Cloudera Manager:

1. Create a new role group for the Thales-namenode.
 - a. **Group Name:** thales-namenode
 - b. **Role Type:** NameNode
 - c. **Copy from:** NameNode Default Group
2. Move the name nodes to the group Thales-namenode.
3. Create a new role group for the Thales-datanode.
 1. **Group Name:** thales-datanode
 2. **Role Type:** DataNode
 3. **Copy from:** DataNode Default Group
4. Move the data nodes to the group thales-datanode.

Configure the Thales-namenode group

1. In the **NameNode Environment Advanced Configuration Snippet (Safety Valve)** dialog, in the thales-namenode field:

```
HADOOP_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent"
```

2. In the **NameNode Advanced Configuration Snippet (Safety Valve)** for hdfs-site.xml, add the following values:

```
<property><name>dfs.vte.ioctl.device</name><value>/opt/vormetric/DataSecurityExpert/agent/secfs/.sec</value></property>  
<property><name>dfs.vte.ioctl.lib</name><value>vorhdfs</value></property>  
<property><name>dfs.vte.rename.check</name><value>true</value></property>  
<property><name>dfs.block.access.token.enable</name><value>true</value></property>
```

Configure the Thales-datanode group

1. In the **DataNode Environment Advanced Configuration Snippet (Safety Valve)** dialog, in the thales-namenode field:

```
HADOOP_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent"
```

2. In the **DataNode Advanced Configuration Snippet (Safety Valve)** for hdfs-site.xml, add the following values:

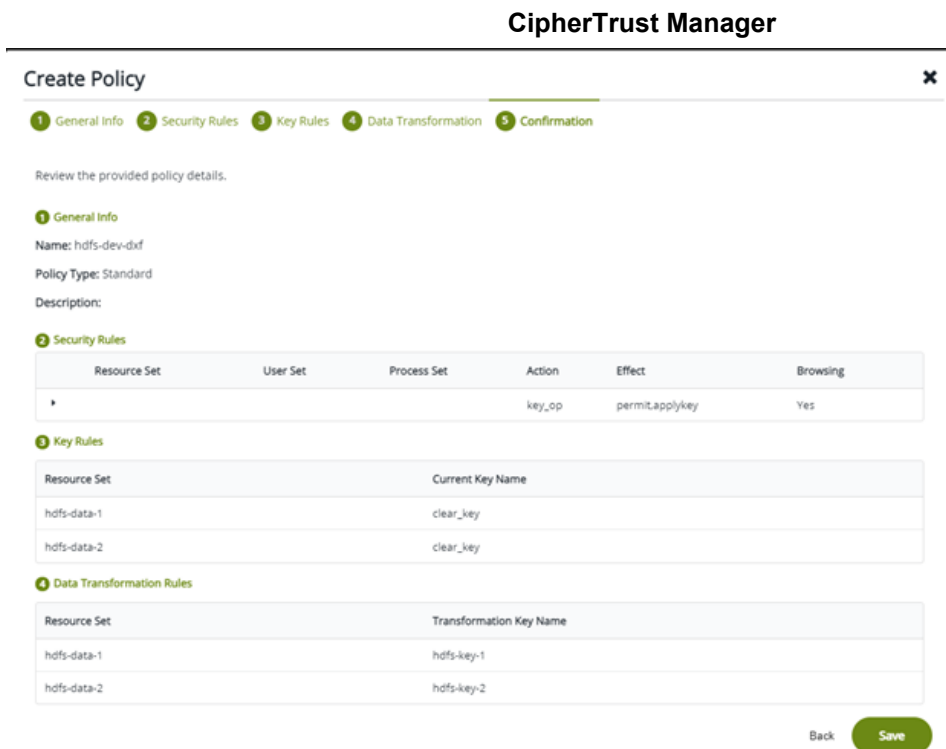
```
<property><name>dfs.vte.ioctl.device</name><value>/opt/vormetric/DataSecurityExpert/agent/secfs/.sec</value></property>  
<property><name>dfs.vte.ioctl.lib</name><value>vorhdfs</value></property>  
<property><name>dfs.vte.user.push</name><value>true</value></property>  
<property><name>dfs.block.access.token.enable</name><value>true</value></property>
```

Take a DataNode Offline and Perform Data Transformation

The next step in activating CTE on HDFS is to switch a DataNode to offline and transform (encrypt) its sensitive data. Once the data is transformed, the HDFS Admin can add the DataNode to the HDFS Host/Client Group. Then they can switch the DataNode back to online. Most of these procedures are completed by the **HDFS Administrator**

although one is done by the Administrator.

1. **HDFS Administrator:** Switch a DataNode to offline.
2. **HDFS Administrator:** Encrypt the files in the directories specified by `dfs.datanode.data.dir` (see ["Configure NameNodes" on page 81](#)).
3. **Administrator:** Create encryption keys and a data transformation policy to transform the data.



4. **Administrator:** After encrypting the data in those directories, add the DataNode host to the HDFS client group.
5. **HDFS Admin:** After the DataNode is added to the HDFS client group, activate the DataNode online.
6. Repeat this procedure for all of the DataNodes in your HDFS cluster.

For more information, see the [CipherTrust Manager documentation](#).

Implementing CTE on HDFS on a Single Host

It is possible, though not recommended, that an HDFS NameNode and DataNode exist as separate processes on the same host. If this is your deployment, use the following CTE deployment guidelines:

1. Configure the HDFS NameNodes (see ["Configure NameNodes" on page 81](#)):

Note: The directories specified by `dfs.datanode.data.dir` already exist on the local file system so you do not have to create them.

2. Pass the following information to the Administrator:
 - The `dfs.datanode.data.dir` directory list and instructions to create a GuardPoint for the HDFS Host Group on each of these directories.
 - Instructions to add the NameNodes IP addresses, or host names, to the HDFS Host Group.

3. Create an HDFS host/client group and host/client group GuardPoint (see "[Create a HDFS Host Group and Host Group GuardPoint](#)" on page 1):
 - a. Administrator must create an HDFS Host Group to contain the HDFS nodes.
 - b. The Administrator must create a GuardPoint for the Host Group on each of the directories specified by `dfs.datanode.data.dir`
4. Take the DataNode offline and perform a data transformation.
5. Add the NameNode/DataNode host/client to the host/client group.

HDFS Upgrade with CTE

To upgrade Hadoop, configure CTE to integrate with the new HDFS instance.

Upgrading one node at a time

Once CTE is installed and configured on the node:

1. Make sure that HDFS services are shut down on the node.
2. Upgrade **Hadoop**.
3. Type:

```
/opt/vormetric/DataSecurityExpert/agent/secfs/hadoop/bin/config-hadoop.sh -i -y
```
4. Start HDFS services on the node.

Upgrade CTE with CTE-LDT in an HDFS Cluster

If you are using CTE-LDT with HDFS cluster, follow these steps when upgrading CTE, in order to maintain your CTE-LDT GuardPoints.

1. Suspend rekey on all data nodes.
2. Shutdown your namenodes/datanodes.
3. Upgrade CTE in the namenode first.

Note: Always upgrade namenodes before datanodes.

4. After CTE upgrade succeeds, type:

```
# config-hadoop.sh -i -y
```
5. On the Ambari admin console, start the namenode.
6. Verify that the CipherTrust java process successfully launched in the namenode. (You should not see an error message.) Type:

```
# ps -ef | grep java | grep vormetric
```
7. Check the CipherTrust Manager status. It should show LDT rekeyed status.
8. Check the namenode status. It should display the GuardPoint status and match the state before upgrade.
Type:

```
# secfsd -status guard
GuardPoint      Policy          Type    ConfigState  Status  Reason
-----
```

```
/hadoop/hdfs/data LDT_HDFS_Sanity local guarded guarded N/A
```

9. Repeat the above steps for all of the datanodes in the HDFS cluster.

Rolling Upgrades

Hortonworks Data Platform has introduced rolling upgrades to automate the Hadoop upgrade process (<http://bit.ly/2pQrFo3>). The upgrade process is controlled by the Upgrade Pack (<http://bit.ly/2rkutvF>) that is predefined and certified by Hortonworks.

To integrate CTE with the upgrade, you need to temporarily change the Ambari scripts before performing the rolling upgrades and then restore the scripts after the upgrades.

1. On Ambari server machine, type:

```
# cd /var/lib/ambari-server/resources/common-services/HDFS/2.1.0.2.0/package/scripts
```

2. Copy the `utils.py` file, type:

```
# cp utils.py utils.py.org
```

3. Using a text editor, add the following commands to `utils.py`:

```
if action == "start":
if name == "namenode" or name == "datanode":
Execute(format("secfs/hadoop/bin/c onfig-hadoop.sh
-i -h {hadoop_bin}/../ -j <java home> -p hdp -d", not_if=service_is_up,
user=params.root_user)
# For Redhat 6.x, uncomment the following command
# Execute(format("/etc/init.d/secfs secfsd restart"),
not_if=service_is_up, user=params.root_user)
# For Redhat 7.x, uncomment the following command
# Execute(format("/secfs restart"), not_if=service_is_up,
user=params.root_user)
before
Execute(daemon_cmd, not_if=service_is_up, environment=hadoop_env_exports
The Java home of your HDFS instance should be used to replace <java home>:
if action == "start": if name == "namenode" or name == "datanode":
Execute(format("secfs/hadoop/bin/config-hadoop.sh -i -h {hadoop_bin}
../ -j <java home> -p hdp -d"), not_if=service_is_up,
user=params.root_user)
# For Redhat 6.x, uncomment the following command
# Execute(format("/etc/init.d/secfs secfsd restart"),
not_if=service_is_up, user=params.root_user)
# For Redhat 7.x, uncomment the following command
# Execute(format("/secfs restart"), not_if=service_is_up,
user=params.root_user)
Execute(daemon_cmd,
not_if=service_is_up,
environment=hadoop_env_exports)
```

4. Type:

```
# ambari-server restart
```

5. Perform rolling upgrades.
6. During the upgrade process, many of the intermediate service status checks can fail. Skip over them by clicking on **Proceed to Upgrade**.

7. Click **Finalize** to complete the upgrade. If the active NameNode fails to activate due to the incompatible HDFS layout version, manually start the NameNode with '-upgrade' option to correct the layout version file.

```
# /var/lib/ambari-server/ambari-sudo.sh su hdfs -l -s /bin/bash -c 'ulimit -c unlimited ; /usr/hdp/current/hadoop-client/sbin/hadoop-daemon.sh --config /usr/hdp/current/hadoop-client/conf start namenode -upgrade'
```

8. If there are excessive under-replicated blocks, run the following command to isolate them and manually start the replication:

```
# su - <$hdfs_user>
# hdfs fsck / | grep 'Under replicated' | awk -F':' '{print $1}' >> /tmp/under_replicated_files
# for hdfsfile in `cat /tmp/under_replicated_files`; do echo "Fixing $hdfsfile :"; hadoop fs -setrep 3 $hdfsfile; done
```

9. Restart the HDFS services. Wait for the replication to complete and the NameNodes to exit safe mode.
10. When Hbase is restarted after upgrades, it tries to rename from:
`/apps/hbase/data/.tmp/data/hbase/namespace` to: `/apps/hbase/data/data/hbase/namespace`, which may cause key conflict if the GuardPoint is set incorrectly (for example, `/apps/hbase/data/data` is guarded, but not `/apps/hbase/data/.tmp`). This results in Hbase shutting down.
Before re-starting Hbase, make sure that the GuardPoint policies on the Hbase files are set correctly to cover all Hbase-related files. A broader GuardPoint (`/apps/hbase/data` instead of just `/apps/hbase/data/data` and other folders) could fix this issue.
11. Check cluster upgrade by verifying the `hadoop version`.
12. Run a few map reduce jobs and Hbase commands to make sure that the entire Hadoop stack is working properly.
13. Rename `utils.py.org` to `utils.py`

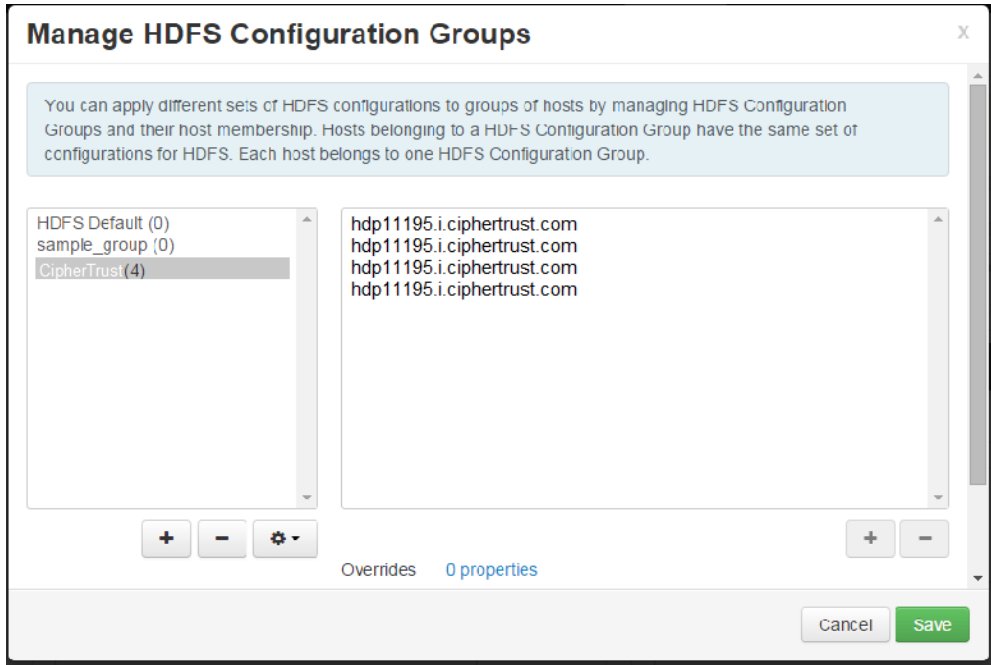
Configure the Hadoop Cluster for CTE

Configure the Hadoop cluster to use CTE before installing and configuring CTE on the nodes. Use Ambari to perform this configuration.

Create a CipherTrust Configuration Group

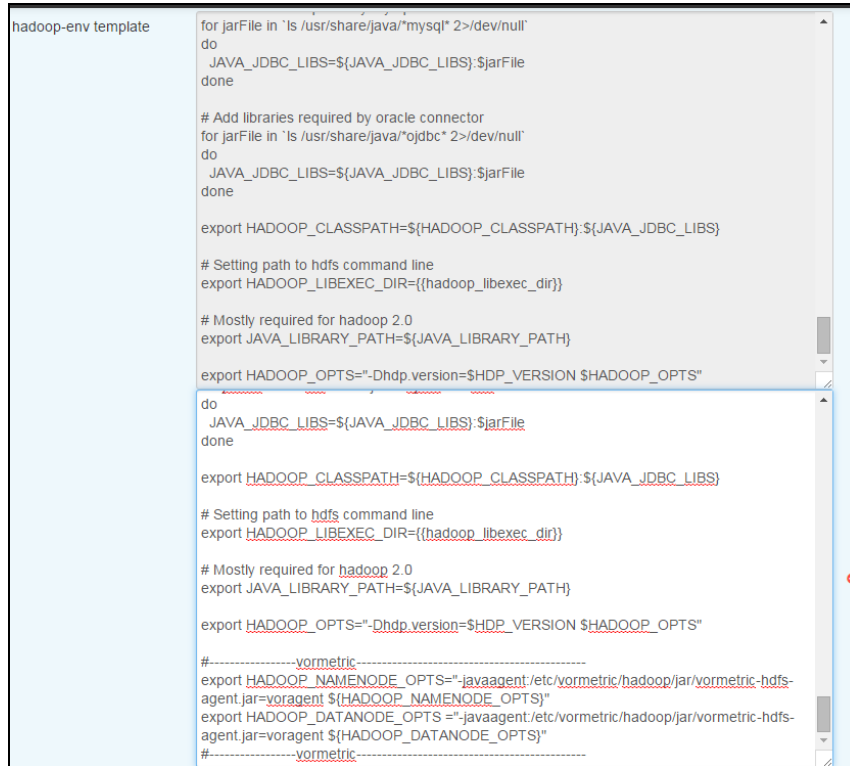
The CipherTrust Configuration Group will eventually contain all of the hosts in your Hadoop cluster. At first you create an empty group. Later you populate it with the hosts on which you will install and configure agents.

1. On Ambari, go to **HDFS > Configs > Manage Config Groups**.
2. Add a new configuration group: *CipherTrust*.
3. Make the group, *CipherTrust*, the current group.



Update the Hadoop-env Template with CTE Settings

1. Go to **HDFS > Configs > Advanced > Advanced hadoop-env > hadoop-env template**
2. Copy and paste the original `hadoop-env` templates into the Thales template and add the following two export lines to specify that the CTE Java agent is instrumented into NameNode and DataNode.



```
hadoop-env template
for jarFile in `ls /usr/share/java/*mysql* 2>/dev/null`
do
  JAVA_JDBC_LIBS=$(JAVA_JDBC_LIBS):$jarFile
done

# Add libraries required by oracle connector
for jarFile in `ls /usr/share/java/*ojdbc* 2>/dev/null`
do
  JAVA_JDBC_LIBS=$(JAVA_JDBC_LIBS):$jarFile
done

export HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:${JAVA_JDBC_LIBS}

# Setting path to hdfs command line
export HADOOP_LIBEXEC_DIR=${hadoop_libexec_dir}

# Mostly required for hadoop 2.0
export JAVA_LIBRARY_PATH=${JAVA_LIBRARY_PATH}

export HADOOP_OPTS="-Dhdp.version=$HDP_VERSION $HADOOP_OPTS"

do
  JAVA_JDBC_LIBS=$(JAVA_JDBC_LIBS):$jarFile
done

export HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:${JAVA_JDBC_LIBS}

# Setting path to hdfs command line
export HADOOP_LIBEXEC_DIR=${hadoop_libexec_dir}

# Mostly required for hadoop 2.0
export JAVA_LIBRARY_PATH=${JAVA_LIBRARY_PATH}

export HADOOP_OPTS="-Dhdp.version=$HDP_VERSION $HADOOP_OPTS"

#-----vormetric-----
export HADOOP_NAMENODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent ${HADOOP_NAMENODE_OPTS}"
export HADOOP_DATANODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent ${HADOOP_DATANODE_OPTS}"
#-----vormetric-----
```

```
export HADOOP_NAMENODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent ${HADOOP_NAMENODE_OPTS}"
export HADOOP_DATANODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent ${HADOOP_DATANODE_OPTS}"
```

Modify the HDFS IOCTL

1. Go to **HDFS > Configs > Advanced > Custom hdfs-site**.
2. In the `dfs.vte.ioctl.lib` field, type: `vorhdfs`
3. In the `dfs.vte.ioctl.device` field, type:
`# /opt/vormetric/DataSecurityExpert/agent/secfs/.sec`

Change the HDFS File Rename Check

1. Go to **HDFS > Configs > Advanced > Custom hdfs-site**.
2. Set `dfs.vte.rename.check` to `true`

User Information Push

You only need this configuration if you want to use the original user information with HDFS operation for IO access check.

Note

It has a performance cost.

1. Go to **HDFS > Configs > Advanced > Advanced hdfs-site** and uncheck **HDFS Short-circuit read**.
2. Set **dfs.block.access.token.enable** to **true**.
3. Go to **HDFS > Configs > Advanced > Custom hdfs-site** and set **dfs.vte.user.push** to **true**.

Uninstalling CTE for the Hadoop Cluster

This section explains how to remove CTE and restore the environment back to the non-CTE cluster environment. Thales recommends uninstalling the agent from HDFS nodes one by one, starting from the DataNode. Uninstall the agent from the NameNode last.

1. Shut down one DataNode.
2. Perform the normal agent uninstall procedure.
3. Go to Ambari, remove the DataNode host from the CipherTrust Configuration Group.
4. Start the DataNode.
5. Delete the CipherTrust Configuration Group from Ambari when agent is uninstalled from all DataNodes.
6. Repeat these steps for each DataNode.
7. Repeat these steps for each NameNode.

CTE Installation and Configuration

After configuring the Hadoop cluster for CTE:

1. Install and register CTE on the HDFS nodes.
 - You can do this to all the nodes at once, but the HDFS is unavailable during CTE installation and configuration.
 - You can also do this one node at a time. If you install and register CTE nodes one at a time, you must start from NameNode, then DataNode, and always keep NameNode service up once NameNodes are configured.
2. In either case, add the FQDN of the node to the CipherTrust Configuration Group, then proceed with agent installation and configuration. See "[Installing and Configuring CTE on an HDFS Node](#)" below.
 - Modify the Host Group. See "[Modifying host settings for HDFS hosts on the CipherTrust Manager](#)" on the [facing page](#).
 - Configure CTE by running `config-hadoop.sh` on the HDFS node. See "[Configuring Hadoop to Use CTE](#)" on [page 92](#).
 - Review the `SecFS` configuration variables that support the HDFS name cache. "[HDFS Name Cache](#)" on [page 93](#).

Installing and Configuring CTE on an HDFS Node

1. Using Ambari, add the FQDN of the node to the CipherTrust Configuration Group. See "[Create a CipherTrust Configuration Group](#)" on [page 86](#).
2. Install, configure, and register CTE as described in [Chapter 1: "Installing CTE for "](#) on [page 1](#) the CTE Agent for

Linux Quick Start Guide.

3. Modify the host settings for each node. See "[Modifying host settings for HDFS hosts on the CipherTrust Manager](#)" below.

Modifying host settings for HDFS hosts on the CipherTrust Manager

The Hadoop service can start as root and then downgrade to an unprivileged user. If the unprivileged user is not authenticated by password, CTE flags the user as fake. CTE cannot allow a user to access a resource protected by a user rule when the user is faked, even if the user matches the permit rule. Because of this, modify the CipherTrust Manager host/client setting as follows:

- On Ambari, go to **HDFS > Configs > Advanced > Advanced core-site** and find out if **hadoop.security.authentication** mode is set to **simple** (no authentication) or **Kerberos**.

Simple Modification

To use **simple**, ask the Administrator to add the following lines to the host/client group:

Note

`/usr/jdk64/jdk1.8.0_40/bin/java` is the Java executable used to launch the HDFS services. Change the Java `jdk` path to reflect your end-user environment.

```
|authenticator+arg+=+class=org.apache.hadoop.hdfs.server.namenode.NameNode |  
/usr/jdk64/jdk1.8.0_40/bin/java  
|authenticator+arg+=+class=org.apache.hadoop.hdfs.server.datanode.DataNode |  
/usr/jdk64/jdk1.8.0_40/bin/java
```

The entire host/client settings will look like this:

```
|authenticator|usr/sbin/sshd  
|authenticator|usr/sbin/in.rlogind  
|authenticator|bin/login/  
|authenticator|usr/bin/gdm-binary  
|authenticator|usr/bin/kdm  
|authenticator|usr/sbin/vsftpd  
  
|authenticator+arg+=+class=org.apache.hadoop.hdfs.server.namenode.NameNode|usr/jdk64/jdk1.8.0_40/bin/java  
|authenticator+arg+=+class=org.apache.hadoop.hdfs.server.datanode.DataNode|usr/jdk64/jdk1.8.0_40/bin/java
```

Modifying Host Group for HDFS NameNodes HA on CipherTrust Manager

To enable high availability (HA) for your HDFS NameNodes, ask the Administrator to add the following lines to the host/client group.

Note

`/usr/jdk64/jdk1.8.0_40/bin/java` is the Java executable used to launch the HDFS services. Change the Java `jdk` path to reflect your end-user environment.

```
|authenticator+arg+class=org.apache.hadoop.hdfs.qjournal.server.JournalNode|usr/jdk64/j  
dk1.8.0_40/bin/java  
|authenticator+arg+class=org.apache.hadoop.yarn.server.applicationhistoryservice.Applica  
tionHistoryServer|usr/jdk64/jdk1.8.0_40/bin/java  
|trust+arg+=+class=org.apache.hadoop.hdfs.tools.DFSZKFailoverController|usr/jdk64/jdk1.8  
.0_40/bin/java
```

The entire host/client group for HA (in this example, with Kerberos) will look like this:

```
|authenticator|/usr/sbin/sshd  
|authenticator|/usr/sbin/in.rlogind  
|authenticator|/bin/login/  
|authenticator|/usr/bin/gdm-binary  
|authenticator|/usr/bin/kdm  
|authenticator|/usr/sbin/vsftpd  
  
|authenticator+arg+class=org.apache.hadoop.hdfs.server.namenode.NameNode|/usr/jdk64/jdk1.8.0_40/bin/java  
|authenticator+arg+=class=org.apache.hadoop.hdfs.server.datanode.SecureDataNodeStarter|/usr/lib/bigtop-utils/jsvc  
|trust+arg+class=org.apache.hadoop.hdfs.qjournal.server.JournalNode|/usr/jdk64/jdk1.8.0_40/bin/java  
|trust+arg+=class=org.apache.hadoop.yarn.server.applicationhistoryservice.ApplicationHistoryServer| \  
/usr/jdk64/jdk1.8.0_40/bin/java  
|trust+arg+=class=org.apache.hadoop.hdfs.tools.DFSZKFailoverController|/usr/jdk64/jdk1.8.0_40/bin/java
```

Configuring Hadoop to Use CTE

1. On the HDFS node, type:

```
# /opt/vormetric/DataSecurityExpert/agent/secfs/hadoop/bin/config-hadoop.sh -i.
```

HDFS prompts you for the following information:

- **Hadoop product name:** (i.e. *hdp*)
- **Hadoop product version:** (i.e. *2.6.0.2.2.0.0-2041*)
- **Path to JAVA_HOME used by Hadoop:** (i.e. */usr/jdk64/jdk1.8.0_40*)

Note: Alternatively, you can use the automated installation option:

```
/opt/vormetric/DataSecurityExpert/agent/secfs/hadoop/bin/config-hadoop.sh -i  
-p hdp -v 2.6.0.2.2.0.0-2041 -j /usr/jdk64/jdk1.8.0_40 2.
```

2. Verify the configuration the using the `-s` option:

```
# /opt/vormetric/DataSecurityExpert/agent/secfs/hadoop/ bin/config-hadoop.sh -s
```

```
Vormetric-Hadoop Configuration Status  
PRODUCT_NAME=hdp- PRODUCT_VERSION=3.0.0.0-2557 HADOOP_HOME=/usr/hdp/current/hadoop-  
client/sbin/./- HADOOP_VERSION=2.7.1 HADOOP_PRODUCT_VERSION=2.7.1.2.3.0.0-2557-  
HADOOP_VERSION_MAJOR=2.7 LIBVORHDFS_  
SO=/usr/hdp/current/hadoopclient/sbin/./lib/native/libvorhdfs.so LIBHDFS_  
SO=/etc/vormetric/hadoop/lib/libhdfs.so VORMETRIC_HADOOP=/etc/vormetric/hadoop-  
#-----vormetric-----  
export HADOOP_NAMENODE_OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-  
agent.jar=voragent ${HADOOP_NAMENODE_OPTS}"v6 . . . . 62 export HADOOP_DATANODE_  
OPTS="-javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent  
${HADOOP_DATANODE_OPTS}"  
#-----vormetric-----  
/etc/vormetric/hadoop/lib/libhdfs.so ...ok  
/usr/hdp/current/hadoop-client/sbin/./lib/native/libvorhdfs.so ...ok  
/etc/vormetric/hadoop/gen-vor-hadoop-env.sh ...ok  
/etc/vormetric/hadoop/vor-hadoop.env ...ok  
Looks ok.  
Vormetric Transparent Encryption Agent 6.0.3 Installation and Configuration Guide  
v6 . . . . 62 export HADOOP_DATANODE_OPTS="-  
javaagent:/etc/vormetric/hadoop/jar/vormetric-hdfs-agent.jar=voragent ${HADOOP_  
DATANODE_OPTS}"  
#-----vormetric -----  
/etc/vormetric/hadoop/lib/libhdfs.so ...ok /usr/hdp/current/hadoop  
client/sbin/./lib/native/libvorhdfs.so ...ok /etc/vormetric/hadoop/gen-vor-hadoop-  
env.sh ...ok /etc/vormetric/hadoop/vor-hadoop.env ...ok  
Looks ok
```

Verify secfsd is Running with Hadoop Environment

Use a text editor to view the `/etc/init/secfsd-upstart.conf` file. The file should contain env entries, type:

```
# cat /etc/init/secfsd-upstart.conf
```

HDFS Name Cache

Obtaining the HDFS file name from the NameNode is network intensive, so the map from HDFS block file name to HDFS file name is cached in a hash table. The following `secfs` configuration variables are used to support the hash cache. They are provided in case you need to tune the memory management of the name cache for better performance.

hdfs_cache_entry: Default is 1,024,000, which could cover up to 125TB HDFS data because the default HDFS block size is up to 128MB (128MB * 1024000 = 125TB).

hdfs_cache_bucket: Default is 10240.

hdfs_cache_timeout: Default to 30 minutes.

hdfs_cache_interval: Default wake up interval for a worker thread to update the cache entry whose timeout has expired is 10 seconds.

On Linux, you can configure each `secfs` configuration variable using the `voradmin` commands:

For example, to configure the variable, type:

```
# voradmin secfs config hdfs_cache_interval 30
```

Note

Because HDFS rename is a metadata operation inside the HDFS NameNode and does not call into the local file system, the hash cache might contain expired data. The HDFS NameNode is coded to prevent renamed data from crossing a key boundary to prevent data corruption. However, other access checks based on the HDFS file name may give incorrect information if the name is expired. Understand this risk before using this feature.

Enabling CTE on HDFS

To enable CTE on your HDFS:

1. Restart CTE agent on the node.
 - For Redhat 6.x, type:

```
# /etc/init.d/secfs restart
```
 - For Redhat 7.x or 8.x, type:

```
# /etc/vormetric/secfs restart
```
2. Restart the Hadoop Services in the cluster.

You can now create GuardPoints to protect your entire HDFS.

Deleting Metadata in HDFS when Migrating Out of LDT

In an HDFS deployment, if you migrate from an LDT to a non-LDT environment, the administrator must delete the LDT `mdstore` file.

In the following example:

```
/hadoop/hdfs is the mount point  
/hadoop/hdfs/data is the guardpoint
```

To manage the migration:

1. In the CipherTrust Manager, click **Client > Client Groups**.
2. Click `<host/client group name>`. The Edit host/client group - `<host group name>` window opens.
3. Click **GuardPoints**.
4. Select the appropriate HDFS directory with an LDT GuardPoint, and click **Unguard**.
5. Using the Ambari admin console, shutdown all NameNode/DataNode one by one. Ensure that no HDFS guardpoints are busy.
6. Ensure that no guardpoints are configured on any HDFS node in the cluster, type:

```
# secfsd -status guard
No guardpoints configured
```
7. On the node running secfs, type:

```
# voradmin ldt attr delete <guard path>
# voradmin ldt attr delete /hadoop/hdfs/data
LDT metadata has been removed from all files in guardpoint /hadoop/hdfs/data
```
8. On the system, verify that the metadata store has been removed from the secfs mount points, type:

```
# voradmin ldt rmstore <mount_point>
# voradmin ldt rmstore /hadoop/hdfs
Enter YES if /hadoop/hdfs does not include any guardpoints associated with an LDT
policy ->YES
MDS file /hadoop/hdfs/vorm-prs has been removed.
```
9. Verify that the metadata store has been removed from the secfs mount points, type:

```
# ls -altr <mount_point>
# ls -altr /hadoop/hdfs
```

You should not see `/hadoop/hdfs/vorm-prs` listed.
10. Repeat the above steps for each node in the HDFS cluster.

Chapter 9: Using CTE with Oracle

This chapter section describes how to install and configure CTE on Oracle RAC ASM, how to use an ASM Filter DriverLinux. It contains the following topics:

Oracle RAC ASM and ASMLib	95
Oracle RAC ASMLib Multi-Disk Online Method	100
About Oracle RAC ASM Raw Devices	100
Oracle RAC ASM Multi-Disk Online Method	101
Oracle RAC ASM Multi-Disk Offline Method (Backup/Restore)	103
Surviving the Reboot and Failover Testing	104
Using CTE with Oracle ASM Filter Driver	105
Basic Troubleshooting Techniques	112

Oracle RAC ASM and ASMLib

This section describes how to install and configure CTE on an Oracle RAC ASM and ASMLib.

Using CTE with an Oracle RAC ASM

You can apply CTE when the Oracle DB is active or inactive. If you choose to use it while the Oracle DB is active, it eliminates any downtime. You can apply CTE during low volume traffic time frames. If you choose to use this option, then use the **rebalance** function of ASM. This allows you to:

1. Migrate data off of a disk so that it can be dropped/removed from a **Diskgroup**.
2. Apply CTE protection.
3. Add the disk back into the diskgroup.



CAUTION

If you drop a disk from an ASM diskgroup, then add it back to the diskgroup without cleanly wiping the disk, the ASM diskname will be corrupted. To avoid this problem, clear out the disk before you add it back to diskgroup. Example: `dd if=/dev/zero of=/dev/secvm/dev/mapper/asmdg-asmlv002 bs=32k`

ASMLib

ASMLib is an optional support library for the Automatic Storage Management feature of the Oracle Database. If the customer is using ASMLib, then management is performed through an Oracle ASM command line. Using this can be simpler than the setup required for standard ASM. The commands and details of the procedure differ as well.

Important ASM Commands and Concepts

Rebalancing Disks

When you drop/remove a disk from the diskgroup, it is important to apply the proper value for the power setting for rebalance and to use the `WAIT` command.

Example ASM Command:

```
SQL> ALTER DISKGROUP <DiskGroupName> DROP DISK <diskName> REBALANCE POWER 8 WAIT;
```

- The **rebalance** command moves the data off of the disk that you are removing from the diskgroup, distributing the data across the remaining DISKS.
- The **power** setting is a number from 1 to 11. It determines how much processing power is dedicated to the rebalance, versus normal operations. Unless the encrypting occurs during heavy traffic volume, the minimum value you should use is 6. Otherwise, consult the customer's DBA for the proper setting. An appropriate value to start with is 8.

Mapping Raw Devices

You can map raw devices for this configuration using:

- **Multipath I/O**

This is typically evident when the path for the mapped devices is set to: `/dev/mapper/<device-name>`.

- **Raw devices**

Some customers use raw devices to map a name like `raw3` to a specific device name. You usually find this mapping in a file called: `/etc/sysconfig/rawdevices`.

Note

It is important to understand how the device names are used and if they are the same across all of the RAC nodes.

Checking Rebalance Status

The `wait` command is very important when ASM performs a rebalance. When you specify `wait`, the command prompt does not display until all of the data is rebalanced and migrated off of the disk. If you do not specify `wait`, the command prompt returns immediately, and you must issue the following ASM command to check the status of the rebalance:

```
SQL> select * from v$asm_operation;
```

This command returns information about the:

- State
- Current power level
- Current amount rebalanced
- Estimated work until completion
- Rate
- Estimated minutes
- Any error codes

Note

It is highly recommended that you always specify the `wait` command when performing a **Drop Disk** with Rebalance. If it is not specified, ASM may prematurely release the disk, thereby allowing CTE to place a GuardPoint on the disk before the rebalance completes. This action may corrupt the data.

Oracle cautions against this issue:



CAUTION

The `ALTER DISKGROUP...DROP DISK` statement returns before the drop and rebalance operations complete. Do not reuse, remove, or disconnect the dropped disk until the `HEADER_STATUS` column in the `V$ASM_DISK` view for this disk changes to `FORMER`. You can query the `V$ASM_OPERATION` view to determine the amount of time remaining for the drop/rebalance operation to complete. For more information, refer to the *Oracle Database SQL Language Reference* and the *Oracle Database Reference*.

Determining Best Method for Encrypting Disks

A diskgroup can contain one or multiple disks. You must determine if the diskgroup contains enough disks and free space for encryption. If the diskgroup contains only one disk, or multiple disks but not enough free space, then you must use the **Offline** (backup/restore) method for encryption.

If the diskgroup contains more than one, you can use the **Online** (rebalancing) method. During rebalancing, additional disks allow for migrating data from the original disk so that it can be encrypted, added back into the diskgroup, and then migrated back to the source disk. Therefore, if the customer does not want to permanently add extra disks, they can add disks temporarily, just for rebalancing.

In general, once you have completed the initial setup for the operating system with which you are working, for both ASM or ASMLib, the high-level process is the same for applying CTE protection to raw devices and using them.

Online Method (No Application / Database Downtime)

Typically, when using the online method, follow these steps:

1. Make an ASM disk available for protection by either removing a disk from an existing diskgroup, or allocating a new disk.
2. Apply CTE encryption to the disk.
3. Add each protected disk to the diskgroup.
4. Restart the nodes and the failover test.
5. Repeat the previous steps for each disk in the diskgroup.

Offline Method (Backup the DB)

Typically, when using the offline method, follow these steps:

1. Backup the database.
2. Make an ASM disk available for protection by either removing a disk from an existing diskgroup, or allocating a new disk.
3. Stop the Oracle database.
4. Delete the diskgroup.
5. Apply CTE encryption to the disk.
6. Recreate the diskgroup.
7. Add the protected disk to the diskgroup.

- Restart the nodes and the failover test.
- Repeat the previous steps for each disk in the diskgroup.

General Prerequisites

Setup

- Verify that you have a current backup of the database
- Install and register CTE agents on all RAC node Hosts
- Create a **Host Group** and add all RAC node hosts as members
- Create an encryption key for the Oracle RAC Database / Application
- Create an Oracle policy using the proper encryption key

Note

If the raw device mappings for the disk(s) are **not** identical across all nodes in the RAC, then you cannot use a Host Group for managing the GuardPoint within the CipherTrust Manager. You **must** apply the GuardPoint to each Host individually. This is typically not optimal, as a Host Group is the most effective and consistent way to manage GuardPoints for Oracle RAC environments.

Modify the UDEV Rules

Note

This applies to RHEL 5 operating systems.

Update the UDEV rules to ensure that they recognize the CTE protected raw devices:

- Edit the file `/etc/udev/rules.d/90-dm.rules`.
- Find and comment out the following line by adding a hash tag (**#**) to the beginning of the line.

```
# KERNEL=="dm-[0-9]*", ACTION=="add", OPTIONS+="ignore_device"
```
- Restart the UDEV rules by executing the following command:

```
# /sbin/start_udev
```

Note

The CTE guide states that a restart is usually required once the UDEV rules have changed. However, that would defeat the purpose of an online method for conversion to CTE. Using the above `start_udev` command removes the need to restart the Host. If this command does not work, then a system restart may be required.

Altering ASM_DISKSTRING on ASM

ASM uses the `asm_diskstring` setting to identify the path where ASM will attempt to locate available disks to use. If you are using device names when adding the disk, you must modify the string to include the path to SecVM.

1. To retrieve the `ASM_DISKSTRING` setting, type:

```
SQL> SHOW PARAMETER ASM_DISKSTRING
```

2. To modify the setting, type:

```
SQL> ALTER SYSTEM SET ASM_DISKSTRING='/dev/mapper/*', '/dev/secvm/dev/mapper/*';
```

Where the path added is the path to SecVM.

ASMLib manages the binding, not ASM. ASMLib creates ASMLib devices on the SecVM devices and presents it to ASM. ASM automatically recognizes the new device. This creates the need to alter diskstrings for ASM. In addition, Oracle ASM sees a new device created using ASMLib and Raw, by default.

For example:

Using Oracle ASMLib to bind the device:

```
# oracleasm createdisk <devicename> /dev/secvm/dev/<blockdev>
```

Using the raw command to bind the device:

```
# raw /dev/raw/rawN /dev/secvm/dev/<blockdev>
```

Specific Prerequisites

Establishing a Starting Point

In many production environments, you may find that it has been a very long time since the RAC nodes have had the services restarted or have been completely rebooted. This can result in a lack of understanding of the actual state of the RAC cluster and its ability to survive a reboot on its own, prior to installing CTE.

Restarts can uncover issues in the RAC environment that are unrelated to CTE. To avoid issues after a CTE installation, Thales recommends that you restart each RAC node **AFTER** CTE is installed and **PRIOR** to establishing any GuardPoints. This may not be feasible in a single node configuration. However, by doing so, CTE is installed but inactive, and you can ensure that the platform is in a workable state prior to getting started.

The Importance of Device Mapping

It is important to use device naming and mapping in a multi-node RAC configuration. Verify the device names to ensure that the disks are mapped to the same disks on each RAC node before applying any GuardPoints. Thales recommends that RAC nodes use the same device names across all nodes. If they do not match, then problems can occur.

If the RAC nodes use the same device names, use a Host Group to create GuardPoints. If they do not match, do not use a Host Group to create GuardPoints. Set them up independently on each Host.

Important Note about Raw Devices on UNIX

In general, raw devices are created as either character or block mode devices. Any I/O performed on character devices is non-buffered, while I/O on block devices is buffered and performed in defined block sizes (that is, 4K bytes).

While the Oracle documentation for using ASM with raw devices indicates that you can use either character or block devices, **CTE REQUIRES a block device for guarding.**

Notes

- Attempting to apply a GuardPoint on a character device that *does not* have a corresponding block device may result in a GuardPoint that never encrypts data. The status of the GuardPoint never shows as guarded.
- The WebUI does not support browsing for the character devices. You would need to manually paste the name into the WebUI.

Oracle RAC ASMLib Multi-Disk Online Method

The online method describes how to remove, protect and add disks to a diskgroup.

Assumptions

Using the Online Method assumes that there is enough free space in the diskgroup so that you can drop/remove, protect and then add the disk back into the diskgroup.

Note

During the initial investigation, you may want to ensure that you have the correct raw device name for each disk that you plan on protecting. Before making any changes to the ASM configuration, obtain the definitive device names for each disk by running the following from the command prompt:

```
# oracleasm querydisk -p <diskName>
```

To add the disk to the diskgroup using the online method and make it ready for use:

1. Open a terminal session on both RAC Nodes.
2. On **RAC Node 1**, perform the following:
 - a. On the ASM, type the following to remove a disk for the diskgroup.

```
SQL> ALTER DISKGROUP <diskGroupName> DROP DISK <diskName> REBALANCE POWER 11 WAIT;
```

- b. Delete the disk from ASM, type:

```
# oracleasm deletedisk <diskName>
```

- c. Verify that the disk is deleted from the ASM and therefore, it is not listed, type:

```
# oracleasm listdisks
```

Note

If you are planning to apply a GuardPoint to a raw device that is currently in an ASM diskgroup, you must remove and delete the disk from the diskgroup before you apply the GuardPoint. ASMLib will not see the guarded disk if you skip this step. When deleting the disk, make sure that the deletion completes before continuing.

About Oracle RAC ASM Raw Devices

When Not Using ASMLib

Before starting the CTE implementation, investigate how the customer is using raw devices for their ASM configuration.

Devices using Raw Bindings

Typically, a device that uses a raw binding looks like the following to ASM:

```
/dev/raw/raw1
```

If the device is mapped this way, you must locate where the mapping is performed. Typically, you can find this in the following configuration file:

```
/etc/sysconfig/rawdevices
```

The underlying binding could be to either a **standard device** name or a **multipath I/O** device name. Either way, you must find where the bind commands are run so that you can modify them for SecVM.

Note

If raw bindings are in use, then typically no changes are needed for the `asm_diskstring`. Because the binding to the actual device is created through the `bind` command, locate where the binding occurs and change the binding to SecVM.

Multipath I/O Devices

Devices using multipath I/O are typically found with the name:

```
/dev/mapper/mpath1
```

Generally, when using multipath I/O, you create SecVM on the multipath device name.

Note

If you use multipath I/O devices in the ASM configuration to add its disk, you must modify the `asm_diskstring` parameter to include the `/dev/secvm/dev/*` path.

Standard Devices

In many cases the ASM configuration may be using plain device names, like the following:

```
/dev/sda1
```

Note

If you use standard device names in the ASM configuration to add a disk, you must modify the `ASM_DISKSTRING` parameter to include the `/dev/secvm/dev/*` path.

Consistent Naming of Devices across RAC Nodes

As previously stated, if the raw device mappings for the disk(s) are **NOT** identical across all nodes in the RAC, then you **CANNOT** use a Host Group and you **MUST** apply the GuardPoints to each Host individually. This is typically NOT optimal, as a Host Group is the most effective way to manage an Oracle RAC environment.

Oracle RAC ASM Multi-Disk Online Method

Performing encryption with the online rebalancing method requires sufficient free space to allow the drop of the largest ASM disk.

Checking for Space

In the Oracle system, use the following commands to check for available disk space:

1. Check total free space in the disk group:

```
SQL> SELECT name, free_mb, total_mb, free_mb/total_mb*100 as percentage FROM v$asm_
diskgroup;
```

NAME	FREE_MB	TOTAL_MB	PERCENTAGE
DATA	7	2109	.331910858

2. Check individual ASM disk size and usage:

```
SQL> select a.name DiskGroup, b.disk_number Disk#, b.name DiskName, b.total_mb, b.free_mb, b.path, b.header_status FROM v$asm_disk b,
v$asm_diskgroup a where a.group_number (+) =b.group_number order by b.group_number, b.disk_number, b.name
```

DISKGROUP	DISK#	DISKNAME	TOTAL_MB	FREE_MB	PATH	HEADER_STATUS
DATA	0	DATA_0000	1874	1273	/dev/oracleasm/disks/DATA3	MEMBER
DATA	1	DATA_0001	1992	608	/dev/oracleasm/disks/DATA4	MEMBER
DATA	3	DATA_0003	117	0	/dev/oracleasm/disks/DATA2	MEMBER
	0	DATA_ENC_0000	109	28	/dev/oracleasm/disks/DATA1_ENC	MEMBER

Adding a Disk to the Diskgroup

Using the Online Method assumes that there is enough free space in the diskgroup so that you can drop/remove a disk, protect it with CTE, and then add it back into the diskgroup.

To add the disk to the diskgroup:

1. Open a terminal session on both RAC Nodes.
2. On **RAC Node 1**, on the ASM, remove the disk from the disk group, type.
3. On the CipherTrust Manager, in the Host Group, apply a GuardPoint to the Raw Device: `<rawDevice1Name>`.
4. From **RAC Node 1**, to display the status of the guarded disks, type:

```
# secfsd -status guard
```

5. On both **RAC Node 1 and 2** type:

```
# chown oracle:oinstall /dev/secvm/<rawDevice1Name>
# chmod 660 /dev/secvm/<rawDevice1Name>
```

6. From **RAC Node**, on the ASM, add the protected disk to the disk group:

```
SQL> ALTER DISKGROUP <diskGroupName> ADD DISK /dev/secvm/<rawDevice1Name> NAME
<disk1Name>;
```

The disk is now added to the diskgroup and ready for use.

7. The system is now ready for a reboot and failover test. For details, see ["Surviving the Reboot and Failover Testing" on page 104](#).

Troubleshooting

Occasionally, settings do not persist when the system is rebooted. To ensure they do persist, edit the `/etc/rc.local` file and add the following lines:

```
Echo "Changing Permission for secvm devices"
chown oracle:oinstall /dev/secvm/dev/<rawDevice1Name>
chmod 660 /dev/secvm/dev/<rawDevice1Name>
```

Oracle RAC ASM Multi-Disk Offline Method (Backup/Restore)

Using the Offline Method assumes that there is not enough free space in the diskgroup.

1. Open a terminal session on both RAC Nodes.
2. On RAC Node 1, on the ASM, type the following to remove the disk group.
`SQL> DROP DISKGROUP <diskGroupName> FORCE INCLUDING CONTENTS;`

Note: Make sure that the disk is removed before guarding the raw devices.

3. On the CipherTrust Manager, in the Host Group, apply GuardPoints to the three raw devices:

```
<rawDeviceName1>  
<rawDeviceName2>  
<rawDeviceName3>
```

4. On **RAC Node 1**, to display the status of the guarded disks, type:

```
# secfsd -status guard
```

5. On both **RAC Node 1** and **2**, type:

```
# chown oracle:oinstall /dev/secvm/<rawDeviceName1>  
# chmod 660 /dev/secvm/<rawDeviceName1>  
# chown oracle:oinstall /dev/secvm/<rawDeviceName2>  
# chmod 660 /dev/secvm/<rawDeviceName2>  
# chown oracle:oinstall /dev/secvm/<rawDeviceName3>  
# chmod 660 /dev/secvm/<rawDeviceName3>
```

6. From **RAC Node 1**, on the **ASM**, add the protected disk to the disk group, type:

```
SQL> ALTER DISKGROUP <diskGroupName> ADD DISK /dev/secvm/<rawDeviceName1> NAME  
<diskName1>;  
SQL> ALTER DISKGROUP <diskGroupName> ADD DISK /dev/secvm/<rawDeviceName2> NAME  
<diskName2>;  
SQL> ALTER DISKGROUP <diskGroupName> ADD DISK /dev/secvm/<rawDeviceName3> NAME  
<diskName3>;
```

The disks are now added to the diskgroup and ready for use.

7. On **RAC Node 1**, restore the database.
8. The system is now ready for a reboot and failover test. Go to the section ["Surviving the Reboot and Failover Testing" on the next page.](#)

Troubleshooting

Occasionally, settings do not persist when the system is rebooted. To ensure they do persist, edit the `/etc/rc.local` file and add the following lines:

```
Echo "Changing Permission for secvm devices"  
# chown oracle:oinstall /dev/secvm/dev/<rawDeviceName1>  
# chmod 660 /dev/secvm/dev/<rawDeviceName1>  
# chown oracle:oinstall /dev/secvm/dev/<rawDeviceName2>  
# chmod 660 /dev/secvm/dev/<rawDeviceName2>  
# chown oracle:oinstall /dev/secvm/dev/<rawDeviceName3>  
# chmod 660 /dev/secvm/dev/<rawDeviceName3>
```

Surviving the Reboot and Failover Testing

Preparing for Failover Testing with ASMLib

When using ASMLib with the `createdisk` command, there is no requirement to make additional changes in `rc.local` or other areas for mapping device names or to use `chmod` or `chown` for SecVM. This is because it is managed for you by the `createdisk` function and you can verify this by running the following command:

```
# ls -l /dev/oracleasm/disks
```

CTE Load Order and Startup Scripts

The last change is to ensure that CTE starts before ASM starts in the startup scripts.

Failover Testing

Confirm that everything is functional:

- Ensure that the GuardPoints are all operational.
- Ensure that you receive valid results when you query the database.
- Verify that the load order ensures that CTE starts before ASM .

Once verified, you can start the failover testing for each RAC Node.

1. Reboot the RAC Node 1 and monitor the startup.
2. Once the restart is clean, reboot RAC Node 2 and monitor the startup.

Issues with Device Mapper and Invalid Guard Path

If CTE is unable to apply a GuardPoint on a raw device, the logs may generate an error similar to the following:

```
[SecFS, 0] EVENT: Failed to guard /dev/mapper/devicename (reason: Invalid Guard Path  
flags 0x2 gyped 0x4 status 0x11) - Will retry later
```

If you receive this error, use the `setup` command to check the status of the disks, type:

```
# setup info <deviceName>
```

Before attempting to establish a GuardPoint, look closely at the open count value and ensure that it is 0 on all nodes.

Using CTE with Oracle ASM Filter Driver

This chapter describes how to configure CTE on Oracle Standalone and RAC ASM Filter Driver for Linux. It contains the following topics:

- How to enable CTE for Oracle ASMFD
- How to configure CTE and create a guarded Oracle ASMFD
- How to convert a baseline ASMFD disk group to a guarded ASMFD disk group
- How to uninstall/ upgrade CTE with an active ASMFD Setup

About Oracle ASM Filter Driver

Oracle ASM Filter Driver (Oracle ASMFD) is a kernel module that resides in the I/O path of the Oracle ASM disks. Oracle ASM uses the filter driver to validate write I/O requests to Oracle ASM disks.

The Oracle ASMFD simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

The Oracle ASM Filter Driver rejects any I/O requests that are invalid. This action eliminates accidental overwrites of Oracle ASM disks that would cause corruption in the disks and files within the disk group.

Audience

Users must be a Senior Oracle DBA administrator who is comfortable setting up a baseline ASMFD environment.

Prerequisites

The following setup allows CTE to correctly load the guarded ASMFD disks correctly after CTE is installed, upgraded, or restarted:

1. Ensure that the ASM `$ORACLE_BASE` variable is assigned a value in `/root/.bash_profile`, if Oracle is already installed.
2. Ensure that a line starting with **+ASM** exists in the `/etc/oratab`.

Note: This line is added by Oracle when Oracle is configured with ASM.

Enable Oracle ASMFD for CTE

To enable Oracle ASMFD:

1. Log in as the root user and stop the Oracle Grid Infrastructure for a standalone server, type:
`$GRID_HOME/bin/crsctl stop has`
2. Configure Disk Discovery for AFD. Modify the following files by adding the content of `/dev/secvm/dev/sd*` to the OS files in `/etc/oracleafd.conf` and `/etc/afd.conf` with `afd_filtering` enabled, type:
 - a. `$ cat /etc/afd.conf`
`afd_diskstring='/dev/sd*,/dev/secvm/dev/sd*'`
`afd_filtering=enable`
 - b. `$ cat /etc/oracleafd.conf`
`afd_diskstring='/dev/sd*,/dev/secvm/dev/sd*'`
`afd_filtering=enable`
 - c. To prevent the file from changing back to `afd_diskstring='/dev/sd*'` make it readonly, type:
`chmod 444 /etc/oracleafd.conf /etc/afd.conf`
3. Restart ACFS before proceeding, type:
`$ acfsload stop`
`$ acfsload start`
4. Restart the Oracle Grid Infrastructure, type:
`$GRID_HOME/bin/crsctl start has`
5. Verify that `/dev/secvm/dev/sd*` is part of the SQL output and the state is enabled as a grid, type:
`SQL> SELECT SYS_CONTEXT('SYS_ASMFD_PROPERTIES', 'AFD_DISKSTRING') FROM DUAL;`

Expected Response

```
SYS_CONTEXT('SYS_ASMFD_PROPERTIES', 'AFD_DISKSTRING')  
-----  
/dev/sd*,/dev/secvm/dev/sd*
```

Note

If using Oracle RAC (Real Application Clusters), repeat the previous steps for all of the nodes in the related RAC cluster.

Configure CTE and create a guarded Oracle ASMFD

To configure CTE and create a Guarded Oracle ASMFD:

1. Create a raw device disk, for example:
`# /dev/sdc`
 2. Partition the targeted raw device, for example:
`# /dev/sdc1`
 3. Guard the partitioned raw device, that you just created, using CTE.
 4. Create an ASMFD disk on the guarded partitioned raw device, type:
`# GRID_HOME/bin/asmcmd afd_label '<disk_label>' '<guarded_raw_partition>'`
- Example:**
- ```
$GRID_HOME/bin/asmcmd afd_label 'DISK1' '/dev/secvm/dev/sdc1'
```

5. Verify that the new disk was created, type:

```
GRID_HOME/bin/asmcmd afd_lsdisk
```

The disk should display in the list of disks, similar to the following:

```
Label Filtering Path
=====
DISK1 ENABLED /dev/secvm/dev/sdc1
```

6. If using Oracle RAC (Real Application Clusters), repeat the previous steps for all of the nodes in the RAC cluster.
7. Now you can use the Oracle Grid ASM Configuration Assistant GUI to create the targeted guarded ASMFD disk group on the above created guarded ASMFD disk, DISK1. See the Oracle GRID documentation for more information.

## Convert a baseline ASMFD disk group to a CTE guarded ASMFD disk group using the Offline method

We will use the following example scenario in this section:

### Objective:

- Convert the baseline ASMFD disk group **DATA1**, to a CTE guarded ASMFD disk group.
- Current configured baseline **DATA1** disk group consists of ASMFD disk **DISK1** on a raw partitioned device `/dev/sdc1`.

To convert:

1. If the targeted raw disk partition to be guarded does not yet exist, create it. For example: `/dev/sdd1`.

**Note:** The capacity of `/dev/sdd1` must be large enough to transfer all of the data from ASMFD disk DISK1 on `/dev/sdc1`.

2. Using CTE, guard the raw partitioned device.

3. Create an ASMFD disk on the guarded raw partition:

```
$ GRID_HOME/bin/asmcmd afd_label '<disk_label>' '<guardedRawPartition>
```

### Example

```
$ GRID_HOME/bin/asmcmd afd_label 'DISK2' '/dev/secvm/dev/sdd1'
```

4. Ensure that there are current backups of all of the database(s) residing on the ASMFD disk group DATA.
5. Shutdown all databases residing on ASMFD disk group DATA1.
6. In the ASM database, run the following SQL command with the rebalance option to add a guarded ASMFD disk DISK2 to ASMFD disk group DATA1, type:

```
SQL> ALTER DISKGROUP <ASMFD diskgroup name> ADD DISK 'AFD:<ASMFD disk name>'
REBALANCE POWER 11 WAIT;
```

### Example

```
ALTER DISKGROUP DATA1 ADD DISK 'AFD:DISK2' REBALANCE POWER 11 WAIT;
```

7. Verify the results, type:

```
SQL>
```

```
col PATH for a15
```

```
col DG_NAME for a15
col DG_STATE for a10
col FAILGROUP for a10
select dg.name dg_name, dg.state dg_state, dg.type, d.disk_number dsk_no,d.path,
d.mount_status, d.FAILGROUP, d.state
from v$asm_diskgroup dg, v$asm_disk d
where dg.group_number=d.group_number
and dg.name = 'DATA1'
```

**System Response:**

| DG_NAME | DG_STATE | TYPE   | DSK_NO | PATH      | MOUNT_S | FAILGROUP | STATE  |
|---------|----------|--------|--------|-----------|---------|-----------|--------|
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK1 | CACHED  | DISK1     | NORMAL |
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK2 | CACHED  | DISK2     | NORMAL |

- Run the following SQL command, in the ASM database, to transfer all data from ASMFD disk DISK1 to ASMFD disk DISK 2 and then drop ASMFD disk DISK1 from ASMFD disk group DATA1, type:

```
SQL> ALTER DISKGROUP <GRPNAME> DROP DISK 'AFD:<DISKNAME>' REBALANCE POWER <POWER #>
WAIT;
```

**Example:**

```
ALTER DISKGROUP DATA1 DROP DISK 'AFD:DISK1' REBALANCE POWER 11 WAIT;
```

**Verify the results:**

```
SQL>
col PATH for a15
col DG_NAME for a15
col DG_STATE for a10
col FAILGROUP for a10
select dg.name dg_name, dg.state dg_state, dg.type, d.disk_number dsk_no,d.path,
d.mount_status, d.FAILGROUP, d.state
from v$asm_diskgroup dg, v$asm_disk d
where dg.group_number=d.group_number
where dg.name = 'DATA1'
```

**System response**

| DG_NAME | DG_STATE | TYPE   | DSK_NO | PATH      | MOUNT_S | FAILGROUP | STATE  |
|---------|----------|--------|--------|-----------|---------|-----------|--------|
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK2 | CACHED  | DISK2     | NORMAL |

At this point, all data in the ASMFD disk group DATA1 has been transferred from the baseline ASMFD disk DISK1, to the guarded ASMFD disk DISK2. At this time, the databases that are residing in ASMFD disk group DATA1 can now be restarted.

## Convert a baseline ASMFD disk group to a CTE guarded ASMFD disk group using the Online method

We will use the following example scenario in this section:

- Number of RAC nodes: 2
- Name of Diskgroup: DATA1
- Number of disks: 3
- Name of baseline disks assigned to raw device:
  - DISK1: /dev/sdb1
  - DISK2: /dev/sdc1
  - DISK3: /dev/sdd1

ASMFD disk group **DATA1** was created from baseline ASMFD disks DISK1, DISK2, and DISK3.

| DG_NAME | DG_STATE | TYPE   | DSK_NO | PATH      | MOUNT_S | FAILGROUP | STATE  |
|---------|----------|--------|--------|-----------|---------|-----------|--------|
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK1 | CACHED  | DISK1     | NORMAL |
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK2 | CACHED  | DISK2     | NORMAL |
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK3 | CACHED  | DISK3     | NORMAL |

### Objective:

- Convert all of the database data on ASMFD disk group DATA1 from baseline to a CTE guarded ASMFD disk group.

**Note:** The combined capacity of /dev/sdc1 on DISK2 and /dev/sdd1 on DISK3 must be large enough to hold all of the data transferring from ASMFD disk DISK1 on /dev/sdb1.

To convert:

From RAC Node 1:

1. Connect locally, as the system administrator, to an Oracle ASM instance using OS authentication, type:

```
$ sqlplus 'as sysasm'
```

2. At the ASM prompt, drop the ASMFdisk DATA1 from the disk group DISK1, type:

```
SQL> ALTER DISKGROUP DATA1 DROP DISK DISK1 REBALANCE POWER 11 WAIT;
```

3. Verify that the ASMFdisk DISK2 is no longer part of the ASMFdisk disk group DATA1, type:

```
sqlplus '/as sysasm'
col PATH for a15
col DG_NAME for a15
col DG_STATE for a10
col FAILGROUP for a10
select dg.name dg_name, dg.state dg_state, dg.type, d.disk_number dsk_no,d.path,
d.mount_status, d.FAILGROUP, d.state
from v$asm_diskgroup dg, v$asm_disk d
where dg.group_number=d.group_number
and dg.name = 'DATA1';
```

### System Response

| DG_NAME | DG_STATE | TYPE   | DSK_NO | PATH      | MOUNT_S | FAILGROUP | STATE  |
|---------|----------|--------|--------|-----------|---------|-----------|--------|
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK2 | CACHED  | DISK2     | NORMAL |
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK3 | CACHED  | DISK3     | NORMAL |

**Note:** Depending on the size of your data, DISK1 might take some time to no longer show in the above query.

4. From the OS command prompt, drop ASMFdisk DISK1:

```
$ORACLE_HOME/bin/asmcmd afd_unlabel DISK1
```

5. From the command prompt, verify DISK1 is no longer visible on both RAC Node 1 and 2.

```
$ORACLE_HOME/bin/asmcmd afd_lslbl
```

6. Using CTE, guard the DISK1 raw partitioned device /dev/sdb1.

7. From the command prompt, recreate ASMFdisk DISK1, with the guarded raw device /dev/sdb1.

```
$GRID_HOME/bin/asmcmd afd_label 'DISK1' '/dev/secvm/dev/sdb1'
```

8. From the command prompt, verify that DISK1 is now visible on both RAC Node 1 and 2 with the newly guarded path: '/dev/secvm/dev/sdb1'

```
$ORACLE_HOME/bin/asmcmd afd_lslbl
```

9. From the ASM prompt, add the guarded ASMFdisk DISK1 back to the ASMFdisk disk group DATA1, type:

```
$ sqlplus 'as sysasm'
```

```
SYS> ALTER DISKGROUP DATA1 ADD DISK 'AFD:DISK1' REBALANCE POWER 11 WAIT;
```

10. Verify that ASMFd disk DISK1 is now part of ASMFd diskgroup DATA1, type:

```
$ sqlplus '/as sysasm'
col PATH for a15
col DG_NAME for a15
col DG_STATE for a10
col FAILGROUP for a10
select dg.name dg_name, dg.state dg_state, dg.type, d.disk_number dsk_no,d.path,
d.mount_status, d.FAILGROUP, d.state
from v$asm_diskgroup dg, v$asm_disk d
where dg.group_number=d.group_number
and dg.name = 'DATA1';
```

### System Response

| DG_NAME | DG_STATE | TYPE   | DSK_NO | PATH      | MOUNT_S | FAILGROUP | STATE  |
|---------|----------|--------|--------|-----------|---------|-----------|--------|
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK1 | CACHED  | DISK1     | NORMAL |
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK2 | CACHED  | DISK2     | NORMAL |
| DATA1   | MOUNTED  | EXTERN | 0      | AFD:DISK3 | CACHED  | DISK3     | NORMAL |

#### Note

Depending on the size of your data, DISK1 might take some time to display in the above query.

11. Repeat the previous steps for Baseline DISK 2 and DISK 3

## Uninstall/ Upgrade CTE with active ASMFd Setup

### For an Active ASMFd Standalone Setup

1. Shutdown all databases running on guarded devices.
2. Shutdown the CRS and ACFS, type:  

```
$ crsctl stop resource -all
$ crsctl stop has
$ afdload stop
```
3. Perform CTE maintenance, uninstall or upgrade.
4. Restart ACFS and CRS, type:  

```
$ afdload start
$ crsctl start has
$ crsctl start resource -all
```
5. Restart related databases.

### For an Active ASMFd RAC Setup

To perform CTE maintenance on all ASMFd RAC nodes in a cluster:

1. Shutdown all databases running on guarded devices.
2. Shutdown CRS, ACFS, and AFD on all nodes, type:  

```
$GRID_HOME/bin/crsctl stop cluster -all
$GRID_HOME/bin/acfsload stop cluster -all
$GRID_HOME/bin/afdload stop cluster -all
```
3. Perform CTE maintenance, uninstall, or upgrade on all targeted RAC nodes.
4. Restart AFD, ACFS, and CRS on all nodes, type:  

```
$GRID_HOME/bin/afdload start cluster -all
$GRID_HOME/bin/acfsload start cluster -all
$GRID_HOME/bin/crsctl start cluster -all
```
5. Restart related databases.

To perform CTE maintenance on a ASMFD RAC cluster in a rolling fashion:

1. Shutdown all instances running on guarded devices on the targeted RAC node only.
2. Shutdown CRS, ACFS, and AFD on the targeted RAC node only. Type:  

```
$GRID_HOME/bin/crsctl stop has
$GRID_HOME/bin/acfsload stop
$GRID_HOME/bin/afdload stop
```
3. Perform CTE maintenance, uninstall, or upgrade on the targeted RAC node only.
4. Restart AFD, ACFS, and CRS on a targeted RAC node only, type:  

```
$GRID_HOME/bin/afdload start
$GRID_HOME/bin/acfsload start
$GRID_HOME/bin/crsctl start has
```
5. Restart related instances on the targeted RAC node.

**Note**

If AFD fails to stop, try the following OS command:

```
modprobe -r oracleafd
```

## Basic Troubleshooting Techniques

The following are some of the most common configuration issues that prevent the Oracle ASM configuration from working properly.

If you encountering errors similar to:

- ORA-15075: disk(s) are not visible cluster-wide
- ORA-15032: not all alterations performed

This could be the result of improper settings for the I/O layer, meaning that your disks are not properly configured.



Perform the following tasks to verify that the settings are correct:

1. On the CipherTrust Manager, in the Host Group that was created for the RAC cluster, verify that the host group for this configuration does **NOT** have the Cluster Group option set ( this option is only for GPFS, which is not supported with CTE).
2. Ensure that the GuardPoints for the block devices are set at the Host Group level. This ensures that each node receives identical GuardPoints.
3. Verify that the GuardPoints are active on all nodes. When the GuardPoints are set, go to each node and verify that they are set and guarded, using the WebUI or the `secfsd -status guard` command. If they do not guard correctly:
  - Verify that the udev rules are set correctly, as described in ["Modify the UDEV Rules" on page 98](#)
  - Make sure the device names are the same across all nodes.
4. From ASM, make sure that the `asm_diskstring` parameter is modified to include the CTE devices and that the proper pathname is used, see ["Altering ASM\\_DISKSTRING on ASM" on page 98](#).

## Verifying Database Encryption

### Option 1

The best way to verify the state of the data, without impacting anything in the existing environment, is to use the Oracle `kfed` command. You can run this command against the native path of the existing GuardPoints and make sure it returns with valid header information. If it returns valid information with the GuardPoint in place, then this confirms that the data is properly encrypted. If it returns with invalid header information, then that indicates that the data is either clear, double encrypted, or not in the expected encrypted state. The syntax for running this command would look similar to the following but will vary based on your environment.

```
/app/oracle/grid/product/11.2.0/grid/bin/kfed read /dev/rdisk/<diskName>
```

If the location is properly encrypted, following is an example of the viewable output:

```
/app/oracle/grid/product/11.2.0/grid/bin/kfed read /dev/rdisk/<diskName>
```

#### System Response:

```
kfbh.endian: 1 ; 0x000: 0x01
kfbh.hard: 242 ; 0x001: 0xf2
kfbh.type: 124 ; 0x002: *** Unknown Enum ***
kfbh.datfmt: 66 ; 0x003: 0x42
kfbh.block.blk: 1088904227 ; 0x004: blk=1088904227
kfbh.block.obj: 1558192170 ; 0x008: file=8234
kfbh.check: 3321251423 ; 0x00c: 0xc5f6465f
kfbh.fcn.base: 932956641 ; 0x010: 0x379bc9e1
kfbh.fcn.wrap: 3040493590 ; 0x014: 0xb53a4016
kfbh.spare1: 3806015223 ; 0x018: 0xe2db2ef7
kfbh.spare2: 3794962182 ; 0x01c: 0xe2328706
6000000000D8000 01F27C42 40E75C23 5CE0202A C5F6465F
[. .|B@.\#\ . *..F_]
6000000000D8010 379BC9E1 B53A4016 E2DB2EF7 E2328706 [7.....:@.....2..]
6000000000D8020 CA2F30AD 522B4D21 99292639 004EBB34 [./0.R+M!.)&9.N.4]
6000000000D8030 A3896BE8 BD839D23 2204E19E 946C575C [...k....#"....lW\]
6000000000D8040 4CE2218F 35E1B101 AF751A70 780E6D6E [L.!5....u.px.mn]
6000000000D8050 5E7E6A38 C600ED5F 929047C4 DF372A8E [^~j8..._..G..7*.]
6000000000D8060 E103152D BA87CC03 11A7D963 9D72FCE1
[...-.....c.r..]
6000000000D8070 1EC6B48B 03EE869F 61D651F9 E7614957 [... ..a.Q..aIW]
```

```
60000000000D8080 810E0353 9C461F49 69569733 501D19EF [...S.F.IiV.3P...]
60000000000D8090 B268002B 4F9457B6 BDB04AC5 D3D07446 [.h.+O.W...J...tF]
60000000000D80A0 FD9EE5E0 9B46CB66 30D10B22 F63AB77E [.....F.f0..":.~]
60000000000D80B0 6FF79075 4BBD1FAD 8F226188 7774300D [o..uK...."a.wt0.]
60000000000D80C0 A809B6FB E1F1C80B B5760E68 9747D97D [.....v.h.G.}]
KFED-00322: Invalid content encountered during block traversal: [kfbtTraverseBlock]
[Invalid OSM block type][][124]
```

## Option 2

The second option to verify the state of the data is to use the `dd` command. This requires you to specify some blocks and write it out to a file. After this completes, read the file using the `strings` command. You can do this while the device is in use. In the example below some sectors are skipped and it only dumps 10000 counts.

**For example:**

```
dd if=/dev/mapper/asm_data2p1 of=/tmp/dd2.out skip=1047 count=10000
```

## Option 3

The third option to verify the state of the data without impacting the existing environment is to use the `strings` command.

### Note

The `strings` command cannot read a busy or large device.

You can run this command against the native path (`/dev/<deviceName>`) of the existing GuardPoints (`/dev/secvm/dev/<deviceName>`). The new path to SecVM would be similar to `/dev/secvm/dev/<deviceName>`. By executing the `strings` command against the native path **`strings /dev/devicename | more`**, this does not go through the SecVM device and therefore is not be decrypted. If it is encrypted the output should contain illegible text.

# Chapter 10: Integrating and Configuring EDB

---

This chapter describes how to configure and integrate Enterprise DB Postgres Advanced Server (EDB) with CTE in Linux. It contains the following sections:

|                                       |     |
|---------------------------------------|-----|
| Overview .....                        | 115 |
| Prerequisites .....                   | 115 |
| Preparing to Create GuardPoints ..... | 115 |
| Integration with CTE .....            | 115 |

## Overview

EDB provides database management software to get more functionality from PostgreSQL. EDB offers secure, scalable, advanced and enterprise-class PostgreSQL solutions. Advanced Server extends PostgreSQL with the security and performance features that enterprises need. It's also compatible with Oracle databases.

## Prerequisites

Refer to [EDB documentation](#) for information on how to setup and configure EnterpriseDB.

## Preparing to Create GuardPoints

Before you can safely reboot your machine, create GuardPoints, and administer any of the services, you must perform the following steps:

1. Stop the EDB service.
2. Open the `/lib/systemd/system/edb-as-13.service` file.
3. Below the [Unit] section, add the following line: `Requires=secfs-fs-barrier.service`.
4. Save the file and exit.
5. Open the file: `/lib/systemd/system/secfs-fs-barrier.service`.
6. At the end of the [Before] clause, add `edb-as-13.service`.
7. Save the file and exit.
8. Type: `systemctl daemon-reload`.
9. Type: `systemctl start edb-as-13.service`.
10. Reboot the system.
11. Stop the EDB.
12. Restart SecFS.
13. Start the EDB service again.

## Integration with CTE

The following describes how to integrate EDB with CTE using an offline dataxform policy:

1. Enable and start EDB.
2. Create a database with sample data.
3. Install the latest CTE build. Refer to the [CTE Agent Linux Quick Start Guide](#) for more information.
4. Register CTE with a CipherTrust Manager.
5. Stop the EDB service.
6. Guard the EDB data and log directories with a Dataxform policy.  
For example, create a policy that transforms from Clear key to AES256.
7. Perform a data transformation on both directories.
8. After Dataxform finishes, guard both directories with a policy using the correct key.  
For example, use AES256 on both directories.
9. After the database and log directories are guarded, start the EDB service again.

# Chapter 11: Using CTE with Pacemaker

---

This chapter describes how to configure CTE with Pacemaker and MySQL on Red Hat 7 or Red Hat 8. It contains the following sections:

|                                                       |     |
|-------------------------------------------------------|-----|
| <a href="#">Overview</a> .....                        | 117 |
| <a href="#">Considerations and Requirements</a> ..... | 117 |
| <a href="#">Creating GuardPoints</a> .....            | 118 |

## Overview

Pacemaker is a high-availability cluster resource manager that runs on a set of hosts (a cluster of nodes) in order to preserve integrity and minimize downtime of desired services (resources). Every resource has a resource agent that abstracts the service it provides and present a consistent view to the cluster.

Thales provides a resource agent for CTE that allows CTE to guard nodes running MySQL databases over an NFS share in a Pacemaker environment.

## Considerations and Requirements

- System Requirements:
  - Red Hat 7 or Red Hat 8. Other versions of Red Hat are not supported.
  - Pacemaker with Corosync and the MySQL database service configured. Make sure that the Pacemaker properties settings for features such as STONITH and quorum are correct based on your Pacemaker environment.

**Note:** Other database applications may be used instead of MySQL, but Thales has only tested the CTE resource agent with MySQL.

- The CTE resource agent supports manual GuardPoints on a device or folder. Automatic GuardPoints are not supported.
- The CTE resource agent supports GuardPoints created from Standard or Live Data Transformation policies. It does *not* support IDT-Capable GuardPoints or CTE-Efficient StorageGuardPoints.
- If you install Pacemaker *after* you have installed CTE, you must copy the CTE resource agent to the `ocf` directory.

```
cp /opt/vormetric/DataSecurityExpert/agent/secfs/.sec/bin/pacemaker_ra_mgp \
/usr/lib/ocf/resource.d/heartbeat/mgp
```
- Upgrades to the CTE resource agent will require using Pacemaker to put each node into maintenance mode while the upgrade is performed.

## Creating GuardPoints

1. Install the CTE Agent on each node in the Pacemaker cluster, and register that node in the same domain in your key manager. For details, see [Chapter 2: "Getting Started with CTE for Linux" on page 21](#).
2. In your key manager, do the following:
  - Create a manual GuardPoint for the MySQL data directory `/var/lib/mysql/` on all of the nodes in the cluster. You can use either a Standard or a Live Data Transformation policy to create the GuardPoint, but you must use the same policy on the GuardPoint for each node.
  - Create any other manual GuardPoints you want to use. Each GuardPoint must be created on all nodes in the cluster and each set of GuardPoints must use the same policy.

For example, if you have three nodes in the cluster and you want to add a Standard GuardPoint for `/hr/shared/files` and a Live Data Transformation GuardPoint for `/dir/accounting/data`, on *each* of the three nodes in the cluster you would create:

- A Standard or Live Data Transformation GuardPoint for `/var/lib/mysql/`.
  - A Standard GuardPoint for `/hr/shared/files`.
  - A Live Data Transformation GuardPoint for `/dir/accounting/data`.
3. Select one of the nodes in your cluster and log into that node as `root`.
  4. On the selected node, add a Pacemaker resource for the manual GuardPoint using the following command:

```
pcs resource create mysql-mgp ocf:heartbeat:mgp mgpdir=/var/lib/mysql \
[start-services=true] [stop_services=false]
```

where:

- `start_services=true` is an optional command that tells the resource agent to start the CTE services before enabling any manual GuardPoints. The options are `true` or `false`, and the default is `true`.
  - `stop_services=false` is an optional command that tells the resource agent to stop the CTE services after disabling any manual GuardPoints. The options are `true` or `false`, and the default is `false`.
5. Create the required resource groups, colocation settings, and constraints.



### WARNING

If the resource groups and colocation constraints are not configured properly, the PCS cluster could run resources on multiple nodes. If the ordering constraints are not set properly, the cluster could fail to start because the system tries to mount the GuardPoint before the file system has been mounted or tries to unmount the file system before the GuardPoint has been disabled. Make sure that the following resource groups and constraints are set properly.

To do so:

- a. Create a MySQL file system group (`mysql-fsg`) that contains `mysql-fs` and `mysql-mgp` using the following command:

```
pcs resource group add mysql-fsg mysql-fs mysql-mgp
```
- b. Create a MySQL service group (`mysql-sg`) that contains `mysql-server` and `mysql-vip` using the following command:

```
pcs resource group add mysql-sg mysql-vip mysql-server
```

c. Configure the colocation and ordering constraints so that:

- `mysql-sg` and `mysql-fsg` are colocated.
- `mysql-fs` starts before `mysql-mgp`.
- `mysql-mgp` stops before `mysql-fs`.
- `mysql-fsg` starts before `mysql-sg`.

To do so, use the following commands:

```
pcs constraint colocation add mysql-sg with mysql-fsg
pcs constraint order start mysql-fs then start mysql-mgp
pcs constraint order stop mysql-mgp then stop mysql-fs
pcs constraint order start mysql-fsg then start mysql-sg
```

6. To check the status of the cluster after the configuration is complete, use the `pcs status` command.

# Chapter 12: Configuring Support for SAP HANA

This chapter section describes SAP HANA, which provides automatic host-failover support. It contains the following topics:

|                                               |     |
|-----------------------------------------------|-----|
| Overview .....                                | 120 |
| Customizing CTE for SAP HANA in HA Mode ..... | 120 |
| Using SAP HANA with LDT .....                 | 122 |
| Setting Memory Allocation .....               | 122 |

## Overview

SAP HANA provides automatic host-failover support. CTE works with HANA fibre storage systems to enable and disable GuardPoints when a protected host starts, stops, or fails over to standby host.

CipherTrust Transparent Encryption - SAP HANA (CTE-SAP HANA) supports non-shared storage where each HANA node has its own separate storage volumes. CTE-SAP HANA provides customized scripts to support startups, shutdowns, and fail overs.

HANA attaches logical unit number (LUNs) or logical volume management (LVMs) using a Fibre Storage Connector (fcClient) providers. Thales provides hooks that are called by the HANA fcClient providers that manage guarding or unguarding of storage locations.

### Note

Thales recommends using host groups to manage configuring in a clustered host environment.

## Customizing CTE for SAP HANA in HA Mode



### CAUTION

The following procedure only applies if multiple HANA nodes are configured in a high availability (HA) environment. If you are installing CTE on a single HANA node, do not change the default HANA or CTE settings.

1. Go to the installation directory:

```
cd /opt/vormetric/DataSecurityExpert/agent/secfs/saphana
```

2. If required, edit the appropriate CTE `fcClient` refined script:

| Script file                           | Use                           |
|---------------------------------------|-------------------------------|
| <code>fcClientRefinedVTE.py</code>    | fcClient provider for LUN     |
| <code>fcClientLVMRefinedVTE.py</code> | fcClientLVM provider for LVMs |

3. Copy the appropriate script file to a shared location that is accessible to all nodes. In a HANA cluster environment, all nodes require access to the CTE scripts.
4. Edit the storage section of the `global.ini` file to indicate the corresponding CTE `fcClient` as the High Availability (HA) provider and to point to the location of the CTE script.



### LUN Example

```
[storage]
 ha_provider = fcClientRefinedVTE
 ha_provider_path = /hana/shared/myFcClient
```

If necessary, enable debug tracing:

```
[trace]
 ha_provider = debug
 ha_fcclient = debug
 ha_fcclientrefinedvte = debug
```

### LVM Example

```
[storage]
 ha_provider = fcClientLVMRefinedVTE
 ha_provider_path = /hana/shared/myFcClient
```

If necessary, enable debug tracing:

```
[trace]
 ha_provider = debug
 ha_fcclientlvm = debug
 ha_fcclientlvmrefinedvte = debug
```

5. Use the same CTE agent with all hosts, including standby hosts.

6. Ensure that `/etc/sudoers` includes the following:

```
<sid>adm ALL=NOPASSWD: /usr/bin/secfsd
```

7. Enable the guard paths at the mount-point level.

For example, individual guards were placed on `/hana/data/HAN/mnt00001`, `/hana/data/HAN/mnt00002`, and so forth.

- a. Use a similar naming practice for log partitions, such as `/hana/log/HAN/mnt00001`, and so forth.
- b. Place the guard at the mount-point level. The guarded paths must match the corresponding data and log mount paths.

8. Configure GuardPoints as type `manual`. You must enable and disable the GuardPoints immediately after the device is attached, or just prior to detachment.

The reason for manual GuardPoints is that it invokes guarding and unguarding from within the HANA during the startup, shutdown, or failover process. The process resembles that of mounting and unmounting guarded auto-mount points.

9. Configure all GuardPoints so that they are available in the standby host, so that any data and log partitions that fail over from any host can be guarded on the standby.

Thales recommends that you configure all GuardPoints on all hosts, because a failed-over active host can then become the new standby, and will require all available GuardPoints.

## Example

The following is an example of the data and log volumes for the host that are mounted.

```
/dev/mapper/VG_HAN_DATA_1-LV_HAN_DATA_1 793971096 3059712 750579916 1%
/hana/data/HAN/mnt00001
 /hana/data/HAN/mnt00001 793971096 3059712 750579916 1%
/hana/data/HAN/mnt00001
 /dev/mapper/VG_HAN_LOG_1-LV_HAN_LOG_1 496233160 2461764 468564152 1%
/hana/log/HAN/mnt00001
 /hana/log/HAN/mnt00001 496233160 2461764 468564152 1%
/hana/log/HAN/mnt00001
```

Note that partition `mnt00002` is also configured, although not currently mounted by HANA. The `secfsd` status output should show the GuardPoint configuration as follows:

| GuardPoint              | Policy | Type   | ConfigState | Status      | Reason   |
|-------------------------|--------|--------|-------------|-------------|----------|
| -----                   | -----  | ----   | -----       | -----       | -----    |
| /hana/data/HAN/mnt00001 | my-pol | manual | guarded     | guarded     | N/A      |
| /hana/log/HAN/mnt00001  | my-pol | manual | guarded     | guarded     | N/A      |
| /hana/data/HAN/mnt00002 | my-pol | manual | unguarded   | not guarded | Inactive |
| /hana/log/HAN/mnt00002  | my-pol | manual | unguarded   | not guarded | Inactive |

For more information, see the *SAP HANA Fiber Channel Storage Connector Admin Guide* at <http://www.sap.com/documents/2016/06/84ea994f-767c-0010-82c7-eda71af511fa.html>

## Using SAP HANA with LDT

SAP HANA is compatible with LDT with the following changes:

- You must add additional CTE commands to the HANA administrator entry. Using a text editor, edit `/etc/sudoers` and add entries for `/usr/bin/voradmin` and `/usr/bin/vmsec`:

```
hanadm ALL=NOPASSWD:
/usr/bin/secfsd, /usr/bin/voradmin, /usr/bin/vmsec, /sbin/multipath, /sbin/multipathd, /etc/init.d/multipathd, /usr/bin/sg_persist, /bin/mount, /bin/umount, /bin/kill, /usr/bin/lsof, /sbin/vgchange, /sbin/vgscan
```
- If you are using an `ext3` file system, you must mount it with extended attributes. Using a text editor, edit the storage section of the `global.ini` file, type:

```
partition_*_data__mountOptions = -o user_xattr
partition_*_log__mountOptions = -o user_xattr
```

## Setting Memory Allocation

There is a limitation in memory allocations for SAP HANA with asynchronous direct I/O. When you use CTE in conjunction with applications like SAP HANA that can process large numbers of direct I/O writes through the Linux AIO interface, CTE can allocate more memory than is desirable.

To limit the amount of memory that CTE consumes for AIO buffers, use the following configuration to limit the amount of memory CTE consumed for AIO buffers:

```
max_aio_memory_limit <MB>
```

The MB value specifies how much memory to allocate to temporary DIO buffers.

### Note

If you do not specify a value, the default is 0, which has no memory bounding effect.

Set the option by echoing a value into the

`/opt/vormetric/DataSecurityExpert/agent//secfs/.sec/conf/` configuration file. For example:

```
echo 1024 > /opt/vormetric/DataSecurityExpert/agent/ secfs/.sec/conf/max_aio_memory_
limit
```

This limits the memory consumed by AIO buffers to 1GB.

**Note**

You **must restart** CTE after changing any values in the configuration directory to make the changes effective.

# Chapter 13: Using CTE with GlusterFS

---

This chapter describes how to configure CTE with GlusterFS. It contains the following sections:

|                                       |     |
|---------------------------------------|-----|
| Overview .....                        | 124 |
| Considerations and Requirements ..... | 124 |
| Configuring GlusterFS for CTE .....   | 124 |

## Overview

GlusterFS is a scalable network file system suitable for data-intensive tasks such as cloud storage and media streaming. The Gluster share can also be mounted as an NFS client.

Thales provides support for GuardPoints in a Gluster environment.

## Considerations and Requirements

- Gluster is supported on Red Hat 7, Red Hat 8, or Ubuntu 18. Other versions of Red Hat or Ubuntu are not supported.
- Clients must have gluster-fuse installed.
- Encryption keys for GlusterFS GuardPoints must use the CBC or CBC-CS1 encryption mode.
- The following CTE features are *not* supported in GlusterFS:
  - Automounted file systems.
  - CTE-Live Data Transformation.
  - IDT-Capable GuardPoints.
  - CTE-Efficient Storage GuardPoints.

### Note

In a multi-node configuration, Thales has tested only the close-to-open consistency for GlusterFS. Other consistencies offered by GlusterFS are not guaranteed to work with CTE.

## Configuring GlusterFS for CTE

1. On the Gluster server, configure the following Gluster volume properties for all configurations:

- `performance.flush-behind` should be off.
- `network.remote-dio` should be disabled.
- `performance.strict-o-direct` should be on.

For example, if the Gluster volume name is `MyGlusterVolume`, you would enter the following commands:

```
gluster vol set MyGlusterVolume performance.flush-behind off
gluster vol set MyGlusterVolume network.remote-dio disable
gluster vol set MyGlusterVolume performance.strict-o-direct on
```

2. If you want to use CBC-CS1 encryption keys, configure the following additional Gluster volume properties on the Gluster server:
  - `performance.read-ahead` should be off.
  - `performance.cache-size` should be 1GB.
  - `performance.write-behind` should be off.
  - `performance.client-io-threads` should be on.

For example:

```
gluster vol set MyGlusterVolume performance.read-ahead off
gluster vol set MyGlusterVolume performance.cache-size 1GB
gluster vol set MyGlusterVolume performance.write-behind off
gluster vol set MyGlusterVolume performance.client-io-threads on
```

3. On the clients, do the following based on your configuration:
  - If the kernel version is *earlier* than version 4.0.0, the Gluster share must be mounted with the `use-readdir=no` mount option.

```
mount -t glusterfs -o acl,use-readdirp=no MyGlusterVolume /gluster
```
  - If you want to use CBC CS1 encryption keys, the Gluster share must be mounted with the `direct-io-mode=enable` mount option.

```
mount -t glusterfs -o acl,direct-io-mode=enable MyGlusterVolume /gluster
```
  - If you are using only CBC encryption keys with kernel versions 4.0.0 and later, no changes need to be made on the clients.

# Chapter 14: NetApp Snapshot Directory

---

This chapter section describes SecFS support for NetApp .snapshot directory over NFS. It contains the following topics:

|                                |     |
|--------------------------------|-----|
| Overview .....                 | 126 |
| Accessing snapshots .....      | 126 |
| Enabling Snapshots .....       | 126 |
| Dataxform Considerations ..... | 126 |

## Overview

The NetApp snapshot directory contains ONTAP snapshot data entries for a specific live volume. Each snapshot is a read-only volume that is automatically mounted over NFS.

A snapshot copy is a read-only image of a traditional, or FlexVol volume, or an aggregate, that captures the state of the file system at a specific point in time.

Data ONTAP maintains a configurable snapshot copy schedule that creates and deletes snapshot copies automatically for each volume.

## Accessing snapshots

By default, every volume contains a directory named .snapshot through which users can access previous versions of files. Users can gain access to snapshot copies depending on the file-sharing protocol used, NFS or CIFS. You can also prevent access to snapshot copies.

Snapshot files carry the same read permissions as the original file. A user who has permission to read a file in the volume, can also read that file in a snapshot copy. A user without read permission to the volume cannot read that file in a snapshot copy.

### Note

Snapshot copies do not have write permissions.

Snapshot directories only display at the mount point, although they actually exist in every directory in the tree. This means that the .snapshot directory is accessible by name in each directory, but is only seen in the output of the `ls` command at the mount point. The snapshots are stamped with the date and time.

## Enabling Snapshots

The NetApp storage administrator, or the OnTap device, must configure this feature. No configuration is required through CTE. CTE guards the client directory mounting the OnTap data volume over NFS.

### Note

The NetApp documentation is located here: <https://nt-ap.com/2vEnEeJ>

## Dataxform Considerations

You cannot transform snapshot directory entries with Dataxform with a new key, because the snapshots are read only. You must keep previous keys and alter the running security policy accordingly to maintain access to the older snapshot entries alongside any new snapshots taken with the new key.

Also, any snapshots that get created during the data transform process (this may take a long time) have to be discarded/deleted as it may contain a mix of data blocks encrypted with both old and new keys.

## Best Practices

Maintaining keys for access to older snapshots can be tedious and cumbersome. Therefore, the simplest and safest practice is to delete all old snapshots once the data is transformed with a new key.

This allows for all new snapshots to be readable with the new key while old keys can be discarded, unless used in other security policies.

# Chapter 15: Using CTE with Quantum StorNext

This chapter describes how to configure CTE and Quantum StorNext devices to interoperate to allow CTE policies to apply to storage managed by Quantum StorNext.

This section contains the following topics:

|                                                        |     |
|--------------------------------------------------------|-----|
| Overview of using CTE with Quantum StorNext .....      | 128 |
| CTE and Quantum StorNext Compatibility .....           | 128 |
| Setting up CTE and Quantum StorNext Integration .....  | 130 |
| Stop secfs Before Upgrading StorNext LAN Clients ..... | 132 |

## Overview of using CTE with Quantum StorNext

Quantum StorNext Fibre Channel-connected devices provide shared file access to third party storage for workstation clients and are optimized for simultaneous access to very large files such as video files. The Quantum StorNext file system is known as SNFS or by its older name, CVFS.

You can encrypt and control access to SNFS files with policies by installing CTE Agents on Linux clients that are configured for access to the SNFS file system. Some limitations apply to this integration, such as supported operating systems, supported SNFS features, concurrent read/write access by multiple clients, and GuardPoint settings (see the next section for more information about these limitations).

## CTE and Quantum StorNext Compatibility

The following sections list the supported operating systems and CTE settings supported for use with Quantum StorNext file systems. Important unsupported configuration parameters are also listed.

### Supported StorNext Server and Client Configurations

The CTE integration with SNFS file systems works only with certain SNFS versions, SNFS storage policies, and client operating systems.

| Configuration parameter                           | Linux                               |
|---------------------------------------------------|-------------------------------------|
| StorNext (SNFS) operating system version          | 6.x                                 |
| StorNext metadata controller (MDC) server OS type | Linux MDC supported                 |
| StorNext replication policy                       | Not supported                       |
| StorNext deduplication policy                     | Not supported                       |
| StorNext truncation                               | Supported                           |
| StorNext full and partial backup                  | Supported                           |
| StorNext expand file system                       | Supported                           |
| StorNext data migration                           | Supported                           |
| StorNext read-ahead cache                         | Disable for use with CTE            |
| Client operating systems                          | Red Hat Enterprise Linux (RHEL) 7.x |



| Configuration parameter                          | Linux         |
|--------------------------------------------------|---------------|
| StorNext LAN client                              | DLC           |
| StorNext mount method: locally mounted directory | Supported     |
| StorNext mount method: CIFS                      | Not supported |
| StorNext mount method: NFS                       | Not supported |

## Supported GuardPoint and Key Settings for SNFS File Systems

When configuring CTE GuardPoints and keys for SNFS, keep in the mind the compatibility limitations listed in the following table.

### Note

Because AES-CBC-CS1 keys are not supported on Windows, do not create a policy on Linux that uses AES-CBC-CS1 keys if access to the same SNFS GuardPoint is required by both Windows and Linux LAN clients.

| Configuration element                                     | Linux                                                                                                                                                                         |
|-----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Offline data transformation                               | Supported                                                                                                                                                                     |
| Live Data Transformation (LDT)                            | Not supported                                                                                                                                                                 |
| Key manager compatibility                                 | See the <i>Compatibility Matrix for CTE Agent with Data Security Manager</i> or the <i>Compatibility Matrix for CTE Agent with Data Security Manager</i> for your CTE version |
| Guard unstructured data                                   | Supported                                                                                                                                                                     |
| Guard structured data                                     | Not supported                                                                                                                                                                 |
| GuardPoint type: Directory (including entire SNFS volume) | Supported                                                                                                                                                                     |
| GuardPoint type: Raw device                               | Not supported                                                                                                                                                                 |
| GuardPoint type: Block device                             | Not supported                                                                                                                                                                 |
| GuardPoint mount option: manual guard                     | Supported                                                                                                                                                                     |
| GuardPoint mount option: auto guard                       | Supported                                                                                                                                                                     |
| GuardPoint mount option: automount                        | Not supported                                                                                                                                                                 |
| AES-CBC key type                                          | Supported                                                                                                                                                                     |
| AES-CBC-CS1 key type                                      | Supported                                                                                                                                                                     |

## Supported Concurrent Access Read/Write Scenarios

If you want to allow access by multiple clients (users) to CTE-protected SNFS files under the same GuardPoint, just read-only access is supported. StorNext file locking is not implemented in CTE, so there is currently no way to prevent concurrent conflicting writes to the same file. As a result, Thales does not support write access to the same GuardPoint from multiple clients.

To enable read access to the same GuardPoint from multiple clients, ensure that all clients are configured to use the same policy and key.

| Configuration parameter                                                    | Linux         |
|----------------------------------------------------------------------------|---------------|
| Read/write access from a single LAN client to a GuardPoint                 | Supported     |
| Read/write access from two or more LAN clients to the same GuardPoint      | Not supported |
| Read-only access from one, two, or more LAN clients to the same GuardPoint | Supported     |

## Setting up CTE and Quantum StorNext Integration

For the most part, CTE integration with Quantum StorNext is the same as for any standard file system. The next section provides an overview of the steps involved in making CTE work with SNFS. Later sections provide more information about the steps that are new or differ significantly from a typical CTE setup.

### Integration Task Overview

The table below provides an overview of the steps involved in setting up SNFS and CTE to work together. As noted in the table, some of these tasks are described in the documentation for your selected key manager. Some of these steps may need to be performed by other staff members at your organization if you have divided the security administration duties as recommended by Thales and you don't have access to the key manager.

| Task                                                                          | Key configuration notes                                                                                                                                                                                                                                                                          | For more information                                                                                                                                                 |
|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Install and configure a Quantum StorNext MDC server for use with CTE          | Disable the StorNext read-ahead cache. Only certain StorNext policies, features, and mount types are supported. See <a href="#">"Supported StorNext Server and Client Configurations"</a> on page 128.                                                                                           | See <a href="#">"Installing and Configuring a Quantum StorNext MDC Server for Use with CTE"</a> on the next page.                                                    |
| Install and configure Quantum StorNext clients for use with CTE               | Ensure that SNFS starts before <code>secfs</code> . See <a href="#">"Ensuring that the StorNext SNFS File System Starts Before secfs"</a> on the next page. Only certain operating systems are supported. See <a href="#">"Supported StorNext Server and Client Configurations"</a> on page 128. | See <a href="#">"Installing and configuring Quantum StorNext DLC Clients for Use with CTE"</a> on the next page.                                                     |
| Create a domain for one or more SNFS hosts, or add them to an existing domain | No difference from standard CTE agent configuration.                                                                                                                                                                                                                                             | See "Domain Management" in your key manager documentation.                                                                                                           |
| Add the host to the key manager                                               | No difference from standard CTE agent configuration.                                                                                                                                                                                                                                             | See "Configuring Hosts and Host Groups" in your key manager documentation.                                                                                           |
| Install and register the CTE Agent on the host system                         | No difference in installation.                                                                                                                                                                                                                                                                   | See <a href="#">Chapter 2: "Getting Started with CTE for Linux"</a> on page 21                                                                                       |
| Create encryption keys (optional)                                             | No difference from standard CTE agent configuration.                                                                                                                                                                                                                                             | See "Managing Keys" in your key manager documentation. For information about AES-CBC-CS1 keys, see <a href="#">Chapter 5: "Enhanced Encryption Mode"</a> on page 49. |

| Task                                                                                                   | Key configuration notes                                                                                                                    | For more information                                                       |
|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Configure host groups containing one or more StorNext LAN clients (optional)                           | No difference from standard CTE agent configuration.                                                                                       | See “Configuring Hosts and Host Groups” in your key manager documentation. |
| Configure policies (including user, process, and resource sets) to control access or enable encryption | No difference from standard CTE agent configuration.                                                                                       | See “Configuring Policies” in your key manager documentation.              |
| Configure one or more GuardPoints                                                                      | Some GuardPoint settings are not supported. See <a href="#">"Supported GuardPoint and Key Settings for SNFS File Systems"</a> on page 129. | See “Managing GuardPoints” in your key manager documentation               |

## Installing and Configuring a Quantum StorNext MDC Server for Use with CTE

Install and configure a Quantum StorNext metadata controller (MDC) server using the [Quantum StorNext documentation](#) as a guide. The CTE integration works with Linux StorNext MDCs. Ensure that you configure the StorNext server to work with the settings supported by CTE as listed in ["Supported StorNext Server and Client Configurations"](#) on page 128. For example, you must disable the StorNext read-ahead cache and only certain StorNext policies, features, and mount types are supported.

## Installing and configuring Quantum StorNext DLC Clients for Use with CTE

Install and configure Quantum StorNext DLC clients using the [Quantum StorNext documentation](#) as a guide. The CTE integration works with Linux StorNext DLCs.

Ensure that you configure DLC clients to work with the settings supported by CTE as listed in ["Supported StorNext Server and Client Configurations"](#) on page 128. For example, only certain operating systems are supported.

### Note

Just read-only access is supported if multiple StorNext LAN clients will access files in the same GuardPoint. For more information, see ["Supported Concurrent Access Read/Write Scenarios"](#) on page 129.

## Ensuring that the StorNext SNFS File System Starts Before secfs

For CTE to function properly for Linux SNFS clients, the SNFS service must start before the CTE `secfs` service. Add an entry for the SNFS file system to `/etc/fstab` on each Linux client that has a CTE agent installed on it. Use the following format:

```
/snfs_share /stornext/snfs1 cvfs defaults,diskproxy=client 0 0
```

In this example, `/snfs_share` should be a share that has been exported from the StorNext Server. It should not be a local disk. You may have completed this configuration step as part of the StorNext LAN client installation. See the [Quantum StorNext documentation](#) for more details.

## Installing the CTE Agent on Each StorNext LAN client

Install a CTE Agent on each computer that is set up as a StorNext LAN client and for which you want to set policies. For supported operating systems, see the table in ["Supported StorNext Server and Client Configurations"](#) on page 128.

Use any installation method supported for your operating system. For details, see [Chapter 2: "Getting Started with CTE for Linux" on page 21](#).

## Stop secfs Before Upgrading StorNext LAN Clients

Before you upgrade StorNext Linux LAN clients, disable or stop the CTE `secfs` service. To stop the `secfs` service, follow these steps:

1. Log in as root on the computer that contains the LAN client that you want to upgrade.
2. Type the following command: `/etc/vormetric/secfs stop`

After you upgrade the StorNext client, start `secfs` again by logging in as root and running the `/etc/vormetric/secfs start` command.

# Chapter 16: Using CTE with McAfee Endpoint Security for Linux Threat Prevention

---

McAfee Endpoint Security for Linux Threat Prevention detects malware such as viruses and handles the malware according to polices that you configure in McAfee ePO. This chapter describes how to configure McAfee and CTE to work together.

This section contains the following topics:

|                                                                        |     |
|------------------------------------------------------------------------|-----|
| Supported McAfee Versions and Linux Operating Systems .....            | 133 |
| Ensuring the Correct McAfee Service Startup and Shutdown Order .....   | 133 |
| Updating McAfee .....                                                  | 135 |
| Virus Scanning Behavior Differences for CIFS and NFS GuardPoints ..... | 135 |

## Supported McAfee Versions and Linux Operating Systems

In general, Thales has verified that CTE is compatible with McAfee version 10.6.5 and later on Red Hat Enterprise Linux (RHEL) 7 and RHEL 8. See the most recent *Compatibility Matrix for CTE Agent with CipherTrust Manager* or *Compatibility Matrix for CTE Agent with Data Security Manager* for details about the versions of McAfee Endpoint Security for Linux Threat Prevention that have been verified to work with CTE. All CTE documentation is available at <https://thalesdocs.com/ctp/cte/index.html>.

## Ensuring the Correct McAfee Service Startup and Shutdown Order

CTE services and McAfee services must be started and stopped in the correct order to prevent problems with any data that is guarded by CTE. This order is important any time these services need to be started or stopped, such as:

- During the normal startup and shutdown of your Linux host.
- Before enabling a scheduled upgrade of CTE.
- Before performing a manual upgrade of CTE.
- As needed for maintenance or troubleshooting.

## Excluding CTE protected directories with McAfee 10.5.x

Starting with CTE v7.2.0, installation of McAfee breaks communication between VMD and CipherTrust Manager. This happens when starting CTE v7.2.0, because the initial startup directory, `/opt/vormetric/DataSecurityExpert/agent/secfs/.sec/` is a protected directory. CTE uses this directory for processes like VMD, secfsd etc. McAfee tries to scan files in this directory and blocks access to VMD. When VMD cannot create files in this directory, communication with CM fails.

To add the CTE protected directory to the exclusion list:

1. At the command line, type:

```
/opt/isec/ens/threatprevention/bin/
```

2. Append the file with the following:

```
'--addexclusionrw --excludepaths
"/opt/vormetric/DataSecurityExpert/agent/secfs/.sec/" --excludesubfolder'
```

The entire exclusion list command should look like the following:

```
/opt/isec/ens/threatprevention/bin/isecav --setoasprofileconfig --profile standard --setmode sor -
-filetypestoscan all --onscanerror deny --onscantimeout deny --networkscan enable --scanarchive
disable --scanmime enable --scanunknownprograms enable --scanunknownmacros disable --primaryaction
clean --secondaryaction delete --primaryactionpup clean --secondaryactionpup delete --
addexclusionrw --excludepaths "/opt/vormetric/DataSecurityExpert/agent/secfs/.sec/" --
excludesubfolder
```

**Note:** You only need to run this command once.

## Ensuring the Correct McAfee Service Order in systemd

Configuring the proper startup and shutdown order of CTE and McAfee services in `systemd` ensures that the services start in the right order during system startup and shutdown. This is also important if you configure a scheduled upgrade of CTE, as CTE services will need to be shut down during the upgrade.

The following McAfee services must be configured to start after CTE services:

- For McAfee 10.6.6 or later:
  - `mfetpd.service`
  - `mfeespd.service`
- For McAfee 10.6.5 or earlier:
  - `isectpd.service`
  - `isecespd.service`

To configure this behavior, add these services to the `Before=` line in the `secfs-fs-barrier.service` file on your system. The order of these services on the `Before=` line in the `secfs-fs-barrier.service` file does not matter. For more information, see ["Location of Application Unit Configuration Files" on page 57](#) and ["Adding Applications to the secfs-fs-barrier.service File" on page 57](#).

## Starting or Stopping McAfee and CTE Manually

When you manually start or stop McAfee and CTE, you must do so in the correct order.

### To manually stop McAfee and CTE:

1. Stop McAfee services using one of the following:

For McAfee 10.6.6 or later:

```
systemctl stop mfetpd.service mfeespd.service
```

For McAfee 10.6.5 or earlier:

```
systemctl stop isecesp.service isectpd.service
```

2. Stop CTE:

|                                                              |                                        |
|--------------------------------------------------------------|----------------------------------------|
| Linux distributions that support <code>systemd</code>        | <code>/etc/vormetric/secfs stop</code> |
| Linux distributions that do not support <code>systemd</code> | <code>service secfs stop</code>        |

**To manually start McAfee and CTE:**

1. Start CTE:

|                                                              |                                         |
|--------------------------------------------------------------|-----------------------------------------|
| Linux distributions that support <code>systemd</code>        | <code>/etc/vormetric/secfs start</code> |
| Linux distributions that do not support <code>systemd</code> | <code>service secfs start</code>        |

2. Start McAfee services using one of the following:

For McAfee 10.6.6 or later:

```
systemctl start mfectpd.service mfeesp.service
```

For McAfee 10.6.5 or earlier:

```
systemctl start isecesp.service isectpd.service
```

## Updating McAfee

It is not necessary to shut down CTE services when you update McAfee Endpoint Security to a new version. Follow the update procedure described by McAfee. Before updating, ensure that the new version of McAfee is compatible with CTE as described in ["Supported McAfee Versions and Linux Operating Systems" on page 133](#).

**Note**

When you upgrade McAfee, make sure that the current McAfee services are configured to start *after* the CTE services in `systemd`. The McAfee service names depend on the version of McAfee that you are using. For details, see ["Ensuring the Correct McAfee Service Order in systemd" on the previous page](#).

## Virus Scanning Behavior Differences for CIFS and NFS GuardPoints

By default, on McAfee Endpoint Security, on-access virus scanning for remotely mounted file systems such as CIFS and NFS, is disabled. However, for GuardPoints configured on CIFS and NFS volumes, this default is ignored. So on-access virus scanning is always on for GuardPoints configured on CIFS and NFS volumes. This means that if a process attempts to save an infected file to a GuardPoint configured on a CIFS or NFS volume, the infected file will be discovered immediately, if it matches the McAfee malware detection algorithm and handled according to the appropriate malware policy in McAfee ePO.

# Chapter 17: Using CTE with Trend Micro Deep Security Software

---

Trend Micro's Deep Security software provides comprehensive security in a single solution that is purpose-built for virtual, cloud, and container environments. Thales has verified certain versions of this Trend Deep product for compatibility with CTE on Red Hat Enterprise Linux (RHEL) 7 and RHEL 8.

This section contains the following topics:

|                                                                                    |     |
|------------------------------------------------------------------------------------|-----|
| <a href="#">Supported Deep Security Versions and Linux Operating Systems</a> ..... | 136 |
| <a href="#">Ensuring Correct Deep Security Service Startup Order</a> .....         | 136 |
| <a href="#">Updating Deep Security</a> .....                                       | 137 |

## Supported Deep Security Versions and Linux Operating Systems

CTE and Deep Security can be used with RHEL 7 and RHEL 8. See the most recent *Compatibility Matrix for CTE Agent with CipherTrust Manager* or *Compatibility Matrix for CTE Agent with Data Security Manager* for the current versions of Trend Micro Deep Security that have been verified to work with CTE. You can find that document on the Thales support website <https://supportportal.thalesgroup.com> (login account required).

## Ensuring Correct Deep Security Service Startup Order

CTE services and Deep Security services must be started and stopped in the correct order to prevent problems with your data that is guarded by CTE. This order is important any time these services need to be started or stopped, such as:

- During normal startup and shutdown of your Linux host.
- Before enabling a scheduled upgrade of CTE.
- Before performing a manual upgrade of CTE.
- As needed for maintenance or troubleshooting.

## Ensuring Correct Deep Security Service Startup Order in systemd

Configuring the proper startup and shutdown order of CTE and Trend Micro's Deep Security services in `systemd` ensures that the services start in the right order during system startup and shutdown. This is also important if you configure a scheduled upgrade of CTE.

The following Deep Security service must be configured to start after CTE services:

```
ds_agent.service
```

To configure this behavior, add this service to the `Before=` line in the `secfs-fs-barrier.service` file on your system. The order of these services on the `Before=` line in the `secfs-fs-barrier.service` file does not matter. See "[Location of Application Unit Configuration Files](#)" on page 57 for the location of the `secfs-fs-barrier.service` file on your system. See "[Adding Applications to the secfs-fs-barrier.service File](#)" on page 57 for information about how to add services to the `secfs-fs-barrier.service` file.



## Ensuring Correct Deep Security Service Startup Order Manually

Perform the following commands in this order if you need to stop Deep Security and CTE services manually:

1. Stop Deep Security services:

```
systemctl stop ds_agent.service
```

2. Stop CTE:

|                                                 |                                        |
|-------------------------------------------------|----------------------------------------|
| Linux distributions that support systemd        | <code>/etc/vormetric/secfs stop</code> |
| Linux distributions that do not support systemd | <code>service secfs stop</code>        |

Perform the following commands in this order if you need to start Deep Security and CTE services manually:

1. Start CTE:

|                                                 |                                         |
|-------------------------------------------------|-----------------------------------------|
| Linux distributions that support systemd        | <code>/etc/vormetric/secfs start</code> |
| Linux distributions that do not support systemd | <code>service secfs start</code>        |

2. Start Deep Security services:

```
systemctl start ds_agent.service
```

## Updating Deep Security

It is not necessary to shut down CTE services when you update Trend Micro Deep Security to a new version. Follow the update procedure described by Trend Micro. Before updating, ensure that the new version of Deep Security is compatible with CTE as described in "[Supported Deep Security Versions and Linux Operating Systems](#)" on the [previous page](#).

# Chapter 18: CTE COS for Amazon S3

---

This chaptersection discusses how to configure CTE COS for Amazon S3 buckets. It contains the following topics:

|                                                                  |     |
|------------------------------------------------------------------|-----|
| Overview .....                                                   | 138 |
| System and Software Requirements .....                           | 140 |
| Client Software Requirements .....                               | 140 |
| CTE COS S3 Installation Overview .....                           | 141 |
| Install Required Linux Packages .....                            | 141 |
| Install CTE with COS Service .....                               | 142 |
| Configure the AWS CLI to use the COS Root CA Certificate .....   | 142 |
| Configure the AWS CLI Network Proxy .....                        | 143 |
| Configure CTE COS S3 .....                                       | 143 |
| Optionally Configure a CTE COS S3 Role for Guarded Buckets ..... | 144 |
| Guard an AWS Bucket .....                                        | 148 |
| Additional COS Proxy Root CA Certificate Information .....       | 148 |
| Protecting Python Programs with CTE .....                        | 150 |
| Enable COS on an Agent with no COS Service .....                 | 151 |
| Uninstall COS from an Agent .....                                | 152 |

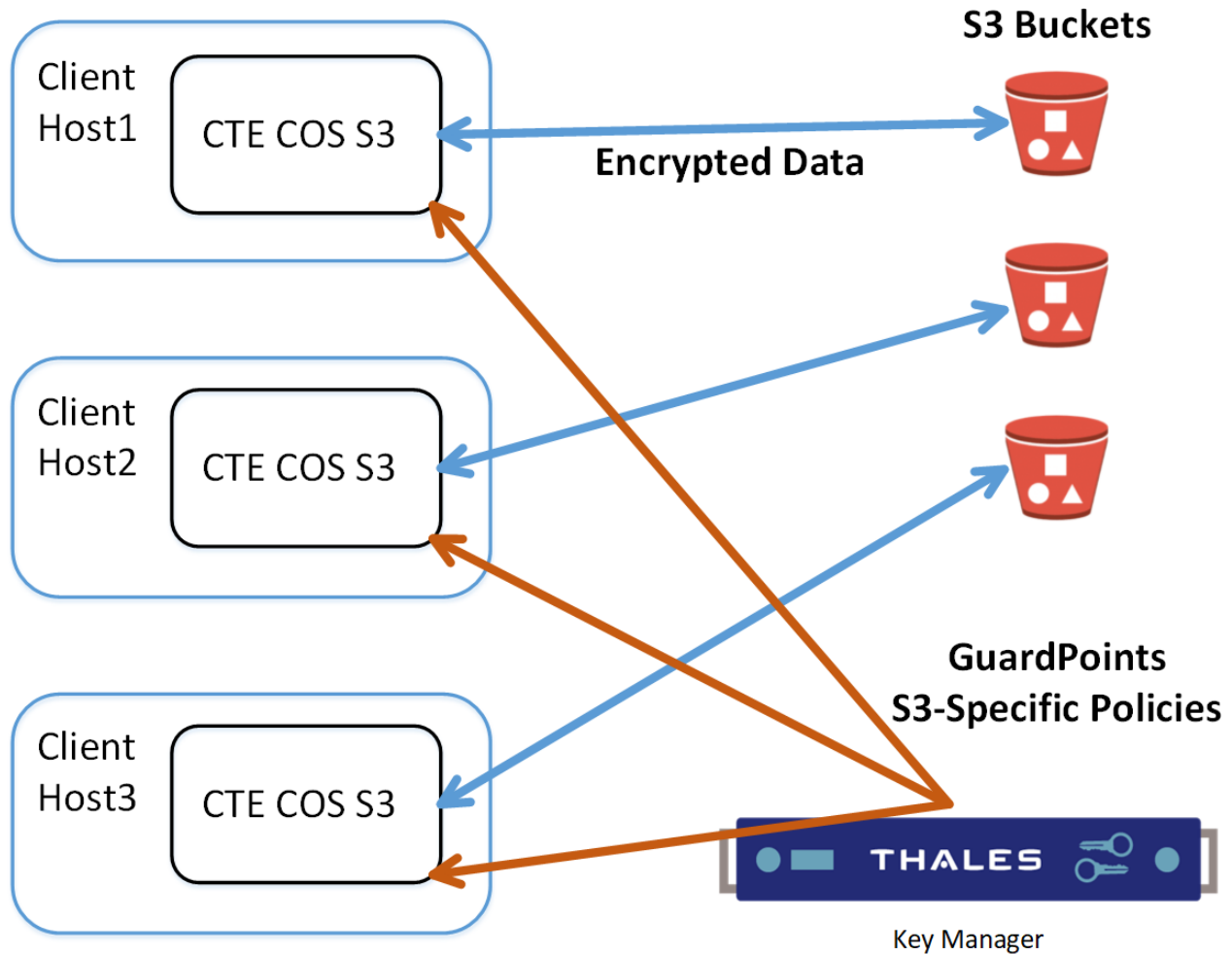
## Overview

CipherTrust Transparent Encryption for Cloud Object Storage (CTE COS) is an object storage service that you can use to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. CTE COS provides management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements.

CTE COS for Amazon Simple Storage Service (CTE COS S3) is an extension to CTE for inline encryption and local host access controls involving applications performing REST-API-based operations to S3 object stores.

CTE COS S3 is not a replacement for AWS IAM access controls which are enforced independently at the AWS server end.

The following diagram shows the high-level CTE COS S3 architecture:



## Supported operations

- CTE COS S3 will process only AWS S3 REST https calls issued by applications. The interception is done by a bundled TLS proxy with additional CTE services.
- CTE's traditional access controls and inline encryption are transparently applied during the following operations:
  - Create a bucket
  - Write an object
  - Read an object
  - Delete an object
  - List objects

## Limitations

- CTE COS S3 is supported only on RHEL 7 and RHEL 8.
- Only REST API based S3 protocol-aware applications are supported.
- Only locally generated self-signed COS Proxy CA Certificates are supported.

- AWS S3 URL path validations are not currently implemented. CTE COS S3 requires that the user must specify the correct URLs for bucket paths.
- CTE COS S3 permits only AES CBC-CS1 encryption. Encrypted files in protected buckets therefore will prepend the 4K embedded header used in CTE's AES CBC-CS1 encryption. The Key Manager will enforce usage of AES CBC-CS1 keys within S3 bucket policies.

## Multi-part Upload Restrictions

- All upload operations for a multi-part upload must be conducted from the same host.
- Maximum file size upload is 5TB.
- Part sizes during uploads must be identical. The part numbering must be in sequence starting with 1, e.g. 1,2,3,4,... Not 10, 20, 30, etc.
- The part size specified in CTE S3's AWS credential file must match the part size specified within the application.
- Thales recommends that the Content-MD5 header be included in the request message.

## System and Software Requirements

- The RHEL 7 or RHEL 8 Linux host must meet the standard CTE requirements.
- You must have an AWS account
- We recommend that you guard the bucket by restricting it with a CTE COS S3 Role to prevent accidental data corruption from connections outside the control of CTE COS S3. For details, see ["Optionally Configure a CTE COS S3 Role for Guarded Buckets" on page 144](#).
- You must download and install the pre-requisite rpm packages before you install CTE COS S3. For details, see ["Install Required Linux Packages" on the next page](#).
- The CTE Agent must be installed in the default directory on the host. You cannot change the default path if you also enable CTE COS S3.

## Client Software Requirements

- Clients must exist on the same host as CTE COS S3. External Client connections will be rejected by the COS Proxy.
- Clients must be AWS S3 protocol aware, and can directly connect to a S3 bucket without the use of an intermediary service, such as the AWS S3 Management Console website.
- Clients must be configured to divert outgoing network connection to a COS proxy (default is localhost:3128 or 127.0.0.1:3128). For more information, see ["Configure the AWS CLI Network Proxy" on page 143](#).
- Clients must use the TLS 1.2 / TLS 1.3 network encryption protocol over TCP/IP to establish connections to the COS proxy.
- Clients must be configured to use the COS CA Root Certificate of the COS proxy to verify and authenticate TLS connections. For more information, see ["Configure the AWS CLI to use the COS Root CA Certificate" on page 142](#).

## CTE COS S3 Installation Overview

In order to configure CTE COS S3, you must complete the following tasks.

### Note

The AWS CLI is used as a sample application to explain the detailed installation procedure.

1. Install the required packages for CTE COS S3. For details, see ["Install Required Linux Packages" below](#).
2. Install CTE and generate the local COS Proxy CA Certificate. For details, see ["Install CTE with COS Service" on the facing page](#).

**Note:** You can only install the Cloud Object Storage feature during CTE installation. You cannot install it post installation.

3. Configure the client to use the COS Proxy CA Certificate. For details, see ["Configure the AWS CLI to use the COS Root CA Certificate" on the facing page](#).
4. Configure the client to use the COS proxy port. For details, see ["Configure the AWS CLI Network Proxy" on page 143](#).
5. Configure CTE COS S3 with AWS Credentials. For details, see ["Configure CTE COS S3" on page 143](#).
6. Optionally configure Role & IAM policies for CTE COS S3 for guarded buckets. For details, see ["Optionally Configure a CTE COS S3 Role for Guarded Buckets" on page 144](#).
7. Configure the guarded buckets. For details, see ["Guard an AWS Bucket" on page 148](#).

## Install Required Linux Packages

CTE COS S3 requires the following pre-requisite packages:

- `boost-regex`
- `boost-system`
- `boost-thread`
- `lib-curl`
- `libtool-ltdl`
- `libxml2`
- `epel-release`.
- `cryptopp` // This package must be installed *after* `epel-release`.
- `log4cpp` // This package must be installed *after* `epel-release`.

For example:

```
$ sudo yum install boost-regex boost-system boost-thread libcurl libtool-ltdl libxml2 epel-release
$ sudo yum install cryptopp log4cpp
```

CTE COS S3 supports both Python2 and Python3. If both versions of Python are available, CTE COS S3 will use Python3. For either Python package, you also need to install the Python modules `"boto3"` and `"future"` after you install the main python package.

- Example for Python2:

```
$ sudo yum install python-pip
$ sudo pip install boto3 future
```

- Example for Python3:

```
$ sudo yum install python3 python3-pip
$ sudo pip3 install boto3 future
```

## Install CTE with COS Service

When you install CTE COS S3 for use with S3, use the instructions in [Chapter 1: "Installing CTE for " on page 1](#) the *CTE Agent for Linux Quick Start Guide*. In addition to those instructions, answer the following prompts as described in this section.

1. When the installer asks about Cloud Object Storage, type `Y` and follow the prompts as shown.

**Note:** You can only install the Cloud Object Storage feature during CTE installation. You cannot install it post installation.

Do *not* change the default CTE installation directory. CTE must be installed in the default location if you enable Cloud Object Storage.

Do you want this host to have Cloud support enabled on the server? (Y/N) [N]: **Y**

```
CTE COS CA Cert is located in
/opt/vormetric/DataSecurityExpert/agent/squid/etc/cosCA.crt
Clients must be updated to use the new CA Certificate
```

```
Generating certificate signing request for the kernel component...done.
Signing certificate...done.
Generating EC certificate signing request for the vmd...done.Signing
certificate...done.
Generating EC certificate signing request for the vmd...done.
Signing certificate...done.
```

```
The following is the fingerprint of the EC CA certificate. Please verify that it
matches the fingerprint shown on the Dashboard page of the Management Console.
If they do not match, it can indicate an unsuccessful setup or an attack.
B0:93:C7:67:07:C9:CB:09:E2:21:F1:5C:8A:C8:79:8F:03:86:21:F2
Do the fingerprints match? (Y/N) [N]: Y
```

```
Successfully registered the CipherTrust Encryption Expert File System Agent with the
primary CipherTrust Data Security Server on security.manager.example.com.
Starting CTE Cloud Service
Installation success.
```

## Configure the AWS CLI to use the COS Root CA Certificate

You must configure the AWS CLI to use the COS root CA certificate. To do so, edit `~/.aws/config` and add the following line to the AWS cli configuration file:

```
ca_bundle = /opt/vormetric/DataSecurityExpert/agent/squid/etc/cosCA.crt
```

For example:

```
$ cat ~/.aws/config
[default]
output = json
Region = us-west-1
ca_bundle = /opt/vormetric/DataSecurityExpert/agent/squid/etc/cosCA.crt
```

## Configure the AWS CLI Network Proxy

All communications between client applications and the AWS server must be done through the COS proxy and the environment variable `HTTPS_PROXY` or `https_proxy` should be set. If both variables are defined, then the AWS CLI will use `https_proxy`.

For example:

```
Export HTTPS_PROXY=localhost:3128
```

## Configure CTE COS S3

In order for CTE COS S3 to do transparent encryption and decryption, all requests sent to the AWS S3 server must be generated and signed using valid AWS credentials. In order to retrieve these credentials for CTE COS S3, you can use any of the following methods:

- **The User Supplies the Credentials for the User's AWS Account**

The application can send the long-term credentials from the user's AWS account to CTE COS S3. These credentials do not expire.

For the AWS CLI, the credentials are in the credential file `~/.aws/credentials`. The user credentials consists of an Access Key ID and a Secret Access Key. For details about accessing the user's credentials, see your AWS documentation.

To add the AWS credentials to CTE COS S3, use the `voradmin cos s3 cred add` command:

```
voradmin cos s3 cred add [<aws_key_id> <aws_secret_key>] where:
```

- `<aws_key_id>` is the AWS secret key ID from the `.aws/credentials` file.
- `<aws_secret_key>` is the AWS secret key from the same file.

For example:

```
voradmin cos s3 cred add AKIA****P KQSm****D
```

- **The User Supplies Temporary Security Credentials**

You can use temporary security credentials, which expire after a short period of time. Usually temporary security credential are obtained through IAM roles and other features of the AWS Security Token Service.

Use the `voradmin cos s3 cred add` command, described above, to add the temporary credential to CTE COS S3.

- **CTE COS S3 Captures Temporary Credentials**

CTE COS S3 can capture a temporary, newly generated, security credential and automatically add it to CTE COS S3 if the application generates the temporary security credential using the AWS Security Token Service with one of the following 3 actions *and* `HTTPS_PROXY` is set to `localhost:3128`.

- `AssumeRole`
- `AssumeRoleWithSAML`
- `AssumeRoleWithWebIdentity`

No other action is required from the application or user.

- **CTE COS S3 Retrieves Credentials from EC2 Instance Metadata Service**

When CTE COS S3 is installed on AWS EC2 instance with an attached role, CTE COS S3 automatically retrieves the credential from Instance Metadata Service and uses it. However, if CTE COS S3 already has a valid credential given by the user or admin using the `voradmin` command, then that credential will be used instead. For information about setting up an IAM Role, see your AWS EC2 documentation.

No action is required from application or user.

## Setting the Default Chunk Size

### Note

If a chunk size is configured in the AWS CLI configuration, you must configure the same chunk size for CTE COS S3.

The default chunk size for multi-part uploads is 8 MB. To change the chunk size, use the following command:

```
voradmin cos s3 chunk [<aws_key_id> <aws_secret_key>] [<chunk_size>] where:
```

- <aws\_key\_id> is the AWS secret key ID from the `.aws/credentials` file.
- <aws\_secret\_key> is the AWS secret key from the same file.
- <chunk\_size> is the number of MB per chunk that you want to use for multi-part uploads. Enter an integer between 5 and 5120.

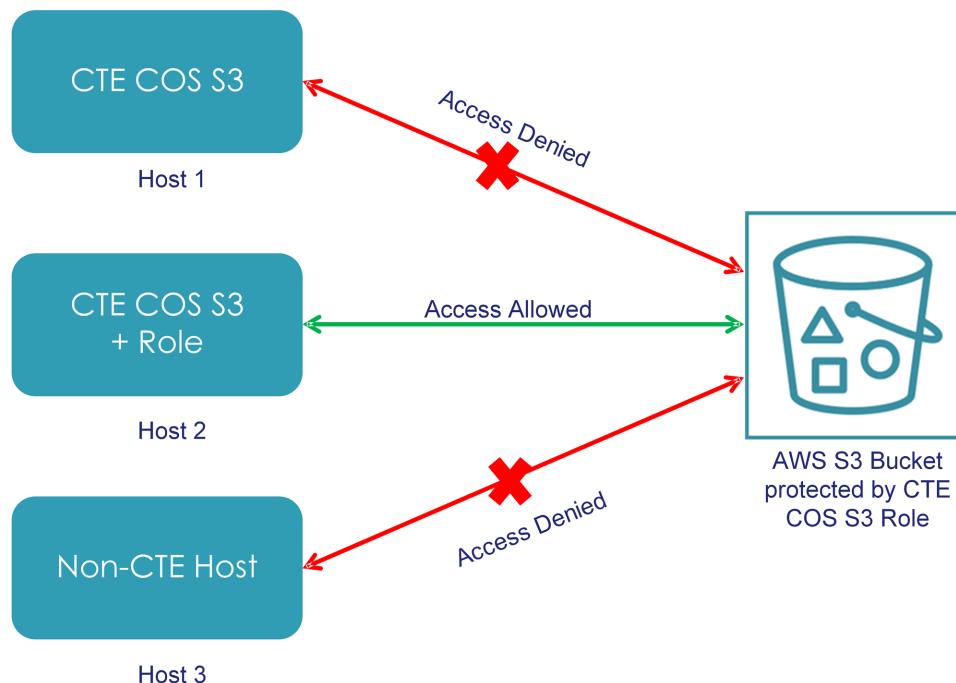
The `voradmin` command prompts for any of the optional parameters that you do not specify on the command.

For example, to set a chunk size of 250MB, you would enter:

```
voradmin cos s3 chunk AKIA****P KQSm****D 250
```

## Optionally Configure a CTE COS S3 Role for Guarded Buckets

AWS provides the IAM Role feature that contains certain specific permissions. A user can assume the IAM Role and therefore take on those permissions. CTE COS S3 provides a special feature using this IAM Role to prevent access to one or more buckets outside the CTE COS S3 protection as shown in the following diagram:



Only hosts that are configured with the CTE COS S3 Role can access the protected buckets. All other access attempts from any hosts where the CTE COS S3 Role is not configured, including attempts made through the AWS S3 Console, are blocked for the protected buckets.

This is a one-time configuration process. After the CTE COS S3 Role is configured on a host by the system or security administrator, all authorized users in the host can access the protected buckets.



## Prerequisites

To set up a CTE COS S3 Role, you need a delegated IAM user, role, and policy. The delegated IAM user should be created by the AWS Administrator without any specific privileges. The role and policy must be created by a user who has at least the following privileges:

- iam:ListPolicies
- iam:CreatePolicy
- iam:GetPolicyVersion
- iam:ListRoles
- iam:ListRolePolicies
- iam:ListAttachedRolePolicies
- iam:AttachRolePolicy
- iam:UpdateAssumeRolePolicy
- iam:CreateRole
- iam:GetRole

## Procedure

1. In the IAM Management Console, create a policy that allows access to specific S3 resources. You can leave the policy open to include all S3 resources in the account or include only those buckets that require CTE protection. Make sure you name the policy something that you will remember.

**Tip:** You can also create the policy as an inline policy after you create the CTE COS S3 Role later in this procedure.

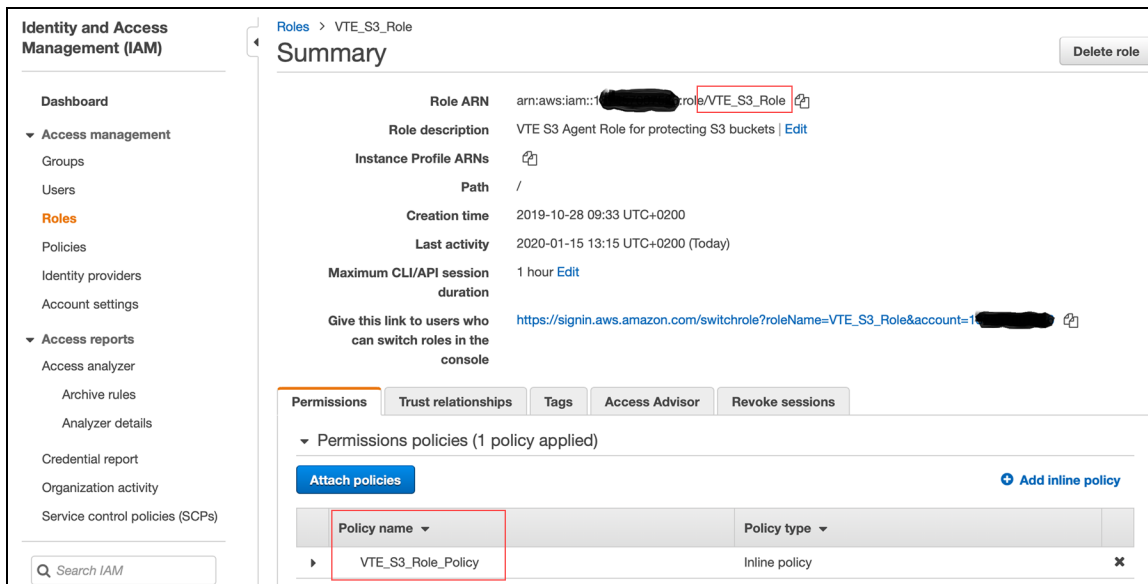
For example, you can create a policy called `VTE_S3_Role_Policy` that allows access to the single S3 bucket `vte-cos-s3-rtb`. To verify that the policy restricts access to that bucket, you can look at the Resource allocation in the Policy summary.

The full JSON for the `VTE_S3_Role_Policy` is:

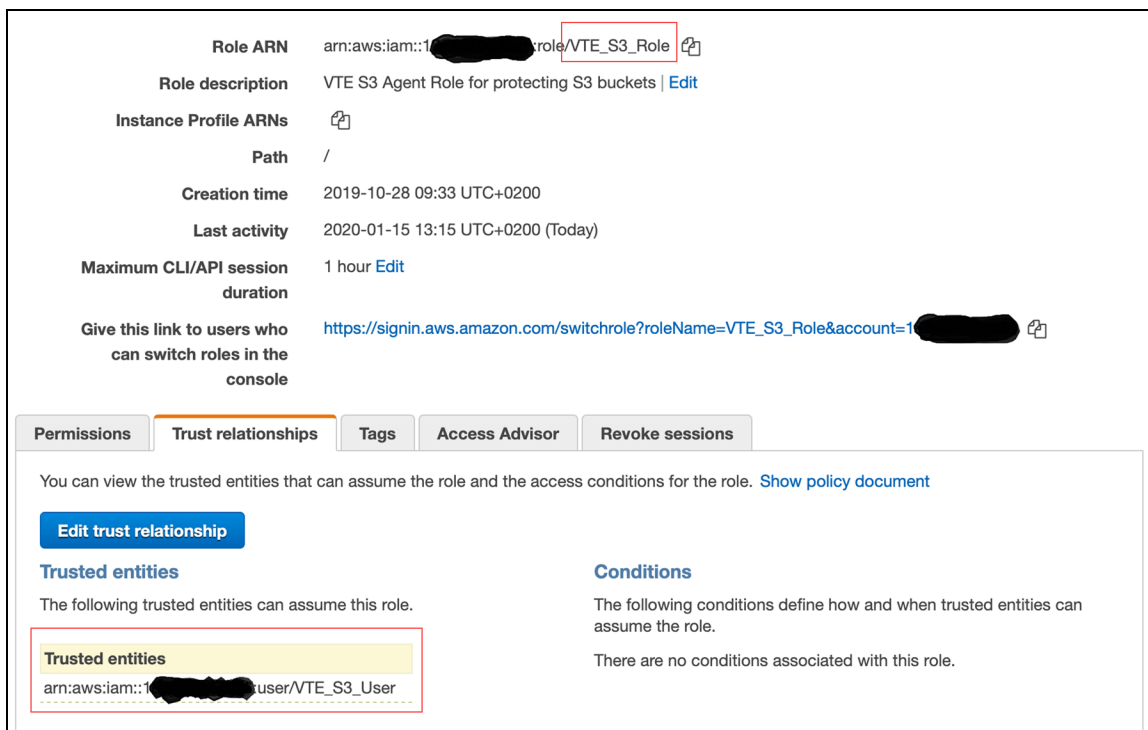
```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor1",
 "Effect": "Allow",
 "Action": "s3:*",
 "Resource": [
 "arn:aws:s3::vte-cos-s3-rtb",
 "arn:aws:s3::vte-cos-s3-rtb/*"
]
 }
]
}
```

2. Create a new role that you can use for the CTE COS S3 Role. For example, you could call the role `VTE_S3_Role`.

3. Assign the CTE COS S3 policy you created to the role, or click **Add inline policy** to create a new policy. For example:



4. Create a delegated IAM user for the CTE COS S3 Role. The user does not require any privileges because its only job is to assume the CTE COS S3 Role. The delegated IAM user can either be in the same account as the role or it can be in a different trusted account. For example, you could create a user called VTE\_S3\_User with no privileges.
5. Set "Trusted Entities" to the delegated IAM user on the **Trust relationships** tab. For example:



6. Configure the CTE COS S3 Role with the credentials of the delegated IAM user you created earlier by entering the following command in the command line on the host system:

```
voradmin cos s3 role config [<aws_key_id> <aws_secret_key> <user_arn> <role_arn>]
where:
```

- <aws\_key\_id> is the AWS secret key ID for the delegated IAM user that you created.
- <aws\_secret\_key> is the AWS secret key or the delegated IAM user that you created.
- <user\_arn> is the Amazon Resource Name for the delegated IAM user that you created.
- <role\_arn> is the Amazon Resource Name for the delegated IAM Role that you created.

If you omit any of the optional parameters, the `voradmin` command prompts you for that information.

For example, if the AWS account number for the delegated IAM user is 1XXXXXXXXXXXX, the user name is `VTE_S3_User`, and the CTE COS S3 Role is `VTE_S3_Role`, you would enter:

```
voradmin cos s3 role config AKIA****P KQSm****D arn:aws:iam::1XXXXXXXXXXXX:user/VTE_
S3_User arn:aws:iam::1XXXXXXXXXXXX:role/VTE_S3_Role
```

After you configure the user and assign the CTE COS S3 Role, CTE will access the S3 bucket through the delegated IAM user account using temporary credentials that CTE regenerates periodically. These credentials are maintained entirely by CTE and are never exposed to end users.

## Secure an S3 Bucket with the CTE COS S3 Role

When you enable the CTE COS S3 Role for a bucket, the associated bucket policy prevents unauthorized users from accessing the contents of the bucket. To enable the CTE COS S3 Role for a bucket, use the following command:

```
voradmin cos s3 role secure-bucket <key_id> <secret key> <bucket_name> where:
```

- <aws\_key\_id> is the AWS secret key ID for the delegated IAM user that you created.
- <aws\_secret\_key> is the AWS secret key for the delegated IAM user that you created.
- <bucket\_name> is the name of the S3 bucket on which you want to enable the CTE COS S3 Role.

For example, if the bucket name is `vte-cos-s3-rtb`, you would enter:

```
voradmin cos s3 role secure-bucket AKIA****P KQSm****D vte-cos-s3-rtb
```

## Disable the CTE COS S3 Role for an S3 Bucket

To remove the CTE COS S3 Role restrictions associated with a bucket, use the following command:

```
voradmin cos s3 role release-bucket <key_id> <secret key> <bucket_name> where:
```

- <aws\_key\_id> is the AWS secret key ID for the delegated IAM user that you created.
- <aws\_secret\_key> is the AWS secret key for the delegated IAM user that you created.
- <bucket\_name> is the name of the S3 bucket on which you want to disable the CTE COS S3 Role.

For example, if the bucket name is `vte-cos-s3-rtb`, you would enter:

```
voradmin cos s3 role release-bucket AKIA****P KQSm****D vte-cos-s3-rtb
```

## Guard an AWS Bucket

To Guard an AWS bucket, you must:

1. Create a CBC\_CS1 key.
  - For CM, see [“Creating a New Key”](#) in the [CM Administrator Guide](#)
2. Create a Cloud Object Storage (COS) policy.
  - For CM, see [“Creating Policies”](#) in the [CTE Administrator Guide](#)
3. Apply the policy to the host.
  - For CM, see [Managing GuardPoints](#) in the [CTE Administrator Guide](#)

## Additional COS Proxy Root CA Certificate Information

The CTE COS CA Certificate, not to be confused with the Kernel and VMD Kernel Certificates, is used with the COS internal Proxy Certificate Authority and must be used by Clients to validate Certificates received during their TLS connection handshake. The default COS CA Self-Signed root CA is automatically created using a locally generated Public/Private Key with the following parameters:

- `CERT_FIELD_PARAM="/C=OZ/ST=Munchkin-land/L=Emerald City/O=ACME Inc/OU=ACME Deliveries/CN=localhost"`
- `SUBJECT_ALT_NAME_PARAM="DNS.1:localhost,IP.1:127.0.0.1"`

To view the currently installed Certificate for the COS Proxy CA, use the `voradmin cos ca_cert display` command.

In the context of the internal COS Proxy CA, the FQDN of 'localhost' would be the correct value, as well as the loop-back IP address of 127.0.0.1 This results in the following locally generated Root CA Certificate.

```
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
2a:28:2c:c5:d6:3b:05:11:fe:6e:32:1d:aa:35:29:44:e5:0d:ce:bf
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=OZ, ST=Munchkin-land, L=Emerald City, O=ACME Inc, OU=ACME Deliveries,
CN=localhost
Validity
Not Before: Feb 11 18:19:33 2020 GMT
Not After : Feb 10 18:19:33 2021 GMT
Subject: C=OZ, ST=Munchkin-land, L=Emerald City, O=ACME Inc, OU=ACME Deliveries,
CN=localhost
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:b9:e6:60:c9:00:f8:00:83:b7:1b:ff:b2:31:eb:
66:5a:eb:21:87:1c:aa:3d:71:b8:08:42:4d:82:6c:
9a:5c:c7:d0:ad:ec:11:9b:be:80:15:55:ab:bc:38:
11:9c:80:c4:1e:63:31:ae:b7:33:8f:88:0b:c2:ca:
e9:e8:0d:78:5a:19:e3:d9:45:fd:4c:b4:81:24:ea:
d3:d4:b9:d2:14:07:e0:33:df:b9:75:36:57:16:4d:
6e:ee:bf:5f:1d:13:14:10:d1:ba:29:0e:1e:11:38:
84:78:8a:e8:ed:1a:24:f7:6a:ac:87:66:9b:21:23:
7b:2c:44:b3:33:6c:04:b7:aa:8c:d3:64:d2:5e:b6:
56:b5:46:54:a9:37:06:c8:e5:30:5f:2a:ba:78:00:
4a:2f:f1:66:a0:1f:fd:26:05:8d:e0:da:23:1e:1b:
1e:a8:ee:77:73:76:32:3c:5e:01:aa:0f:d5:8b:ac:
a9:08:7e:50:63:5e:88:95:e5:5f:dc:1d:7b:b0:59:
50:c1:56:ba:e6:11:da:c6:c5:79:3e:a6:46:f2:39:
db:6a:9d:aa:da:ff:68:d0:39:9c:fd:5a:d5:0e:3e:
41:07:62:32:c0:be:4f:92:56:34:92:c8:1d:bd:87:
ec:e5:3b:44:a0:8f:8c:09:f9:37:40:df:b3:24:bb:
8d:67
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints: critical
CA:TRUE, pathlen:0
X509v3 Key Usage:
Certificate Sign, CRL Sign
X509v3 Subject Key Identifier:
C5:19:E5:41:B7:69:E7:10:27:2D:F6:49:0D:46:0A:4B:FE:8C:7E:CB
X509v3 Authority Key Identifier:
keyid:C5:19:E5:41:B7:69:E7:10:27:2D:F6:49:0D:46:0A:4B:FE:8C:7E:CB
X509v3 Subject Alternative Name:
DNS:localhost, IP Address:127.0.0.1
X509v3 Issuer Alternative Name:
DNS:localhost, IP Address:127.0.0.1
Signature Algorithm: sha256WithRSAEncryption
2d:3c:2b:93:c0:61:1d:35:d7:f2:5f:5c:e8:0d:61:57:f2:a8:
e0:ec:98:74:02:b5:c4:78:a4:2f:b5:2b:b4:96:56:17:93:89:
eb:45:ac:df:1e:1b:e0:d5:38:da:55:62:61:97:5b:d9:9e:31:
9b:71:f1:17:37:31:5d:12:0f:5e:c1:ea:29:ee:b2:97:6e:7c:
```

```
c0:97:a9:8d:a9:2c:c0:68:e4:fa:b1:21:f8:50:b8:c0:2e:51:
fd:f2:5b:4d:41:72:0c:48:a2:db:47:14:66:20:c7:62:bd:33:
e8:a4:f4:22:c9:07:0f:0d:58:a0:9e:a1:f9:96:9c:97:c1:28:
6a:18:6f:ea:b9:28:42:48:5a:5c:da:98:22:9f:05:59:27:82:
3f:3d:4e:0b:9d:37:04:76:0e:ec:d9:f1:25:c8:78:78:fc:31:
d0:cb:24:db:47:96:7c:fa:dc:0d:14:6c:13:44:8d:87:5b:82:
d2:0f:a9:8c:48:bd:a6:b1:b9:0c:bb:50:14:70:d0:8b:7b:8c:
a5:e5:52:83:47:25:15:d6:d0:17:e0:9f:f7:99:d0:2e:17:93:
c5:38:e0:b8:c8:d4:f2:ed:39:99:ec:19:cf:5e:39:78:7b:5f:
07:48:4b:df:ec:d9:94:c5:aa:df:4d:a9:a5:a9:e3:88:74:0e:
d7:74:83:87
```

If you want to change the defaults, you can use the silent install option with the `CERT_FIELD_PARAM` and `SUBJECT_ALT_NAME_PARAM` set to the desired values, or you can replace the default Certificate using the `voradmin cos ca_cert` command. For more details, see the `voradmin` manpage.

## Protecting Python Programs with CTE

CTE uses SHA-256 to generate binary signatures that can be used to allow certain trust levels for a customer application. Capturing the signature of a binary allows CTE to determine if the application's binary has changed in any way, because those changes might mean the security of the binary has been compromised.

While this mechanism works well for executable binaries to verify an application's validity at run time, this mechanism does *not* work for applications implemented with interpreter languages because CTE can only access the interpreter's signature to validate the file. The program itself is something that is executed by the interpreter and thus CTE cannot hook into this process. Scripting languages like Python are particularly difficult because many Python programs include other Python files, which means there is no single program signature that defines the execution path.

While CTE cannot use signature sets to protect traditional Python programs, tools like PyInstaller allow for the creation of a self-contained Python executable that CTE can sign and protect.

## Benefits of Using PyInstaller with CTE

Using PyInstaller allows the creation of self-contained Python binary packages. These packages contain the Python interpreter, the Python program, and all required modules and libraries. Because each package includes all of the files it needs, it is immune to changes to the Python modules or libraries on the system. This not only allows CTE to sign the binary but it also adds extra protection against an attacker trying to gain access through modifications of the underlying libraries.

Another benefit is that it provides security administrators with a stable process that reasonably immune to system updates done by a system administrator. Once the application binary has been created and signed, the behavior of the program will not change because of updates made through the OS packaging system or Python's PIP package manager.

## Getting PyInstaller

PyInstaller is an OpenSource project. The instruction for installing PyInstaller as well as the documentation are available through the project page at:

<https://PyInstaller.readthedocs.io>

## Example Usage

The exact usage of PyInstaller arguments varies based on the details of the Python program being packaged into an executable. This example show what is needed to generate a self-contained AWS CLI binary with PyInstaller. It uses the `--onefile` flag to generate a single, self-contained binary. It assumes that the pip3 Python package manager is already installed on the system.

### Installing PyInstaller

The following command installs PyInstaller in `${HOME}/.local/bin/PyInstaller`:

```
pip3 install --user PyInstaller
```

### Installing the Amazon AWS Command Line Utility

The following command installs the AWS utility in `${HOME}/.local/bin/aws`:

```
pip3 install --user awscli
```

### Creating a Runtime Hook

The following command creates a runtime hook that changes the Python `botocore` `root` directory within the packaged binary. This is necessary to make sure that we use the `botocore` packaged by PyInstaller. The hook is save to a file named `change-botocore-root.py`. For more information on runtime hooks, see the PyInstaller documentation.

```
cat <<EOF > ${HOME}/change-botocore-root.py
import sys
import botocore

if getattr(sys, 'frozen', False):
 botocore.BOTOCORE_ROOT = sys._MEIPASS
EOF
```

In order to get the AWS CLI utility working with PyInstaller, we need to specify some hidden imports as well as add some folders from the `awscli` and `botocore` packages into the executable. For more information about hidden imports and adding files to the executable, see the PyInstaller documentation.

```
${HOME}/.local/bin/PyInstaller --onefile \
 --hiddenimport=awscli.handlers \
 --hiddenimport=pipes \
 --add-data=${HOME}/.local/lib/python3.6/site-packages/awscli/data:data \
 --add-data=${HOME}/.local/lib/python3.6/site-packages/botocore/data:data \
 --runtime-hook=${HOME}/change-botocore-root.py \
 ${HOME}/.local/bin/aws
```

The resulting binary will be saved in the current working directory as `dist/aws`.

## Enable COS on an Agent with no COS Service

To enable COS on a system that has CTE installed but no COS Service:

1. Uninstall CTE completely.
2. Reinstall CTE and select **Yes** when asked:

```
Do you want to configure this host for Cloud Object Storage? (Y/N) [N]: Y
```

## Uninstall COS from an Agent

Currently, the only method for uninstalling COS is as follows:

1. Uninstall CTE completely.
2. Delete the host/client from DSM or CM.
3. Re-register the host/client.
4. Reinstall CTE and select **NO** when asked:

```
Do you want to configure this host for Cloud Object Storage? (Y/N) [N]: N
```



# Chapter 19: CTE-Efficient Storage for Linux

---

Storage arrays offer features such as compression and deduplication for storage efficiency. They also provide simplistic encryption that fails to deliver the requisite levels of security. CipherTrust Transparent Encryption (CTE) offers a much higher level of security. However, in deployments where storage arrays receive encrypted data streams from hosts protected by a CTE instance (protected hosts), the encrypted data streams make the efficiency of storage array systems ineffective. This is because every block written by CTE is different resulting in zero deduplication and compression on these blocks.

CTE offers CTE-Efficient Storage as the solution to the storage efficiency challenge in storage arrays on Linux.

## Introduction to CTE-Efficient Storage

CTE-Efficient Storage for Linux is a licensed feature available with CTE 7.0.0 or later. In addition, all CTE Agents must be registered with a CipherTrust Manager v2.6, or a subsequent version.

### Note

All references to 'storage array' in this document assume storage array systems capable of supporting CTE-Efficient Storage functionality.

With CTE-Efficient Storage, CTE offers the same degree of security for the data stored on the arrays while offering a new type of encryption key and GuardPoint. The new type of key enables storage array systems to achieve storage efficiencies with encrypted data streams. The coordination between the storage array and CTE is essential for achieving storage efficiency with encrypted data.

In the context of this solution, a LUN exported from a storage array system to a CTE-managed host, is referred to as an *ES GuardPoint*. An ES GuardPoint is a guarded device configured with the CTE-Efficient Storage capability.

## Requirements and Considerations

- CTE-Efficient Storage requires XTS-AES mode of the AES algorithm for encryption.
- CTE only supports CTE-Efficient Storage on servers with microprocessors integrated with Advanced Encryption Standard instruction set (AES-NI).
- CTE-Efficient Storage requires that the encryption keys be stored in the KMIP server on the CipherTrust Manager. Therefore, all hosts on which you want to create ES GuardPoints must be registered with a KMIP-enabled domain in the CipherTrust Manager.

**Note:** You can also use In-Place Data Transformation to protect storage arrays without needing to store the encryption keys in the KMIP server. For details, see [Chapter 20: "In-Place Data Transformation for Linux" on page 182](#).

## CTE-Efficient Storage Enhanced Storage Arrays

CTE shares the encryption key associated with the LUN with a storage system that exports the LUN to a protected host. In this solution, the LUN is a device configured for CTE-Efficient Storage that can be guarded as an ES GuardPoint. When the device is guarded, the storage system and protected hosts coordinate operations for sharing the encryption key applied to the ES GuardPoint.

By sharing the key, the storage system decrypts the encrypted data streams that the protected host writes to the LUN, and then performs the data reduction process on the clear data before encrypting and storing the final encrypted data in the storage array system. The storage system does the reverse operations when the protected host reads data from Efficient Storage devices.

## Storage Arrays Compatible with CTE-Efficient Storage

You can use CTE-Efficient Storage with:

- FlashArray from Pure Storage

### FlashArray

FlashArray from Pure Storage is enhanced with CTE-Efficient Storage capabilities and inter-operates with CTE to provide CTE-Efficient Storage on Linux. The Pure Storage system is a client of the CipherTrust Manager and shares the encryption keys protecting the LUNs exported from the storage system to the protected hosts registered with the same CipherTrust Manager.

#### Note

See the EncryptReduce Installation Guide from Pure Storage for information on setting up interoperability with CTE.

### Sharing Encryption Keys

CTE shares the encryption key for a LUN with the storage system using the KMIP protocol. In this solution, the CipherTrust Manager is the KMIP server, and the storage system is a KMIP client registered with the CipherTrust Manager. Any host accessing and protecting the LUNs from the storage system is a CTE managed host registered with the same CipherTrust Manager. The protected hosts register with the CipherTrust Manager using the `register_host` script executed on the protected hosts.

With the host and the storage system registered with the same CipherTrust Manager, the protected host continues enforcing policy and security rules on device and directory GuardPoints. The protected host stores an Efficient Storage Device Header, (ES Header) on each LUN configured as ES GuardPoint. The ES Header includes the UUID of the encryption key applied to the LUN and identifies the LUN as an ES GuardPoint to the storage array exporting the LUN. The storage array recognizes the ES Header on the LUN when the protected host writes the header, and then uses the UUID of the key from the header to retrieve the key attributes and material from the CipherTrust Manager (KMIP Server). This process enables the storage array and the protected host to share and apply the same key for encryption and decryption of data streams exchanged between them.

#### Note

The hosts accessing a shared LUN must be protected hosts registered with the same CipherTrust Manager as the storage array.

When the LUN is permanently de-configured as an ES GuardPoint, the ES Header must be removed. The storage array also detects the removal of the ES Header from the LUN and de-configures the LUN as an CTE-Efficient Storage device. Both the protected host and the storage array stop encryption and decryption of data streams exchanged between hosts and the storage array.

### Storage Array Registration

The storage array administrator creates a certificate for the storage array and communicates the certificate to the Administrator. The Administrator produces a corresponding certificate specific to the array which is given to the storage array administrator. The CipherTrust Manager and the storage array system register both certificates and uses them each time they establish a secured session.

#### Note

See the EncryptReduce Installation Guide from Pure Storage for detailed instructions on registering the FlashArray as KMIP client with the with the CipherTrust Manager's KMIP server.

## Efficient Storage Device Header and CTE Private Region

The key sharing aspect of an ES GuardPoint requires a small amount of disk space in the storage device reserved for CTE private use. The reserved space is where CTE shares information with the storage array that is exporting the device to the protected host. The reserved space starts the beginning of the device.

The protected host writes the ES Header to the device when the device is guarded for the first time. The storage array recognizes the header written to the LUN and begins the key sharing process and encrypting/decrypting data streams transferred between the protected host and the storage array on the LUN.

CTE allocates a small amount of storage space on each device configured as CTE-Efficient Storage. This region is reserved for exclusive use by CTE and is referred to as the CTE Private Region. The CTE Private Region is 63 megabytes.

CTE stores the ES Header and other metadata information to allow CTE and the storage array to exchange information. The ES Header occupies the first sector on the device. The method that CTE uses to claim the CTE Private Region on a device depends on whether the device is new (holds no data) or has existing data that you want to preserve. CTE writes the ES Header when guarding the device for the first time. The storage array recognizes the header written to the device and begins the key sharing process for exchange of encrypted data streams between the protected host and the storage array on the device.

## Device Size

After you create the ES GuardPoint on the device, the device size reported to applications is the size of the device minus the space reserved for the CTE Private Region. This can lead to a discrepancy between the disk size reported by some applications versus the size reported by system utilities such as `fdisk`.



### WARNING

**Do not shrink ES GuardPoints. Due to relocation of user data from CTE Private Region, if you shrink the device, you may corrupt data on the device.**

The ES Header contains both the available device size and the size of the CTE Private Region. To view the ES Header, use the `voradmin esg status <device-name>` command. The **Exported Disk Size** field shows the disk size available for use by other applications. The **Private Region Size** field shows the disk size reserved for CTE. For example:

```
voradmin esg status /dev/sdh
ESG Header on /dev/secvm/dev/sdh
 Version: 1
 Change: 0
 Notifications: None
 Storage Status: None
 Private Region Size: 129024 sectors
 Exported Device Size: 20842496 sectors
 Key UUID: 2c3ee4af-010c-3593-898e-f2f62d29222d
 Mount Path: None
```

The `voradmin esg status` command also reports the UUID of the XTS-AES 256 key applied to the device. The reported UUID matches the UUID of the key as shown on the CipherTrust Manager key property and KMIP key object.

## ES GuardPoint Encryption Keys

ES GuardPoints must be encrypted using XTS-AES 256 keys. An XTS-AES 256 type key is a 512-bit key composed of two components:

- The first 256 bits of the key is the AES 256 encryption key component
- The second 256 bits is the tweak component

You create XTS keys on the CipherTrust Manager using the “Add Key” function, similar to non-XTS keys.

### Note

You *must* check the **KMIP Accessible** check box on the “Add Agent Key” page to make the key available to KMIP clients through KMIP. If you do not check the **KMIP Accessible** option, storage arrays cannot get keys for CTE-Efficient Storage devices from the CipherTrust Manager (KMIP server) for sharing those keys with protected hosts.

The CipherTrust Manager also generates a UUID for a newly added key. The CipherTrust Manager provides the key and its attributes to the protected hosts for guarding the CTE-Efficient Storage device. The protected host for the device writes the ES Header to the device, including the UUID of the key, before the initial attempt to enable an ES GuardPoint. The storage array recognizes the ES Header written to the device, retrieves the UUID from the ES Header, and retrieves information and attributes of the AES 256 component of the XTS-AES 256 key from the CipherTrust Manager using KMIP.

## Policy Requirements for ES GuardPoints

Initializing an ES GuardPoint requires a Standard or In-Place Data Transformation policy with a KMIP-accessible XTS-AES 256 key as described in ["Initialize Linux Efficient Storage Devices" on the facing page](#).

Rekeying an ES GuardPoint requires an In-Place Data Transformation policy with a KMIP-accessible XTS-AES 256 key as described in ["Changing the Encryption Key on Linux ESG Devices" on page 166](#).

You may add security rules to restrict certain user/process access to protected devices. For suggestions about what security rules you may want to use, see ["Use Cases involving Efficient Storage GuardPoints" on page 170](#).

## Guarding an Efficient Storage Device on Linux

The following sections discuss how to guard an efficient storage device on Linux. If you want to guard an efficient storage Windows device, see the *CTE Agent for Windows Advanced Configuration and Integration Guide*.

In order to guard a Linux efficient storage device, you need to:

1. Make sure the devices you intend to guard meet the requirements for ES GuardPoints. For details, see ["Requirements for Efficient Storage GuardPoints on Linux" below](#).
2. Register the protected host with the CipherTrust Manager with CTE-Efficient Storage enabled.
3. Initialize the storage device to create a CTE Private Region for the Efficient Storage Device Header. For details, see ["Initialize Linux Efficient Storage Devices" on the facing page](#).
4. Log on to the CipherTrust Manager to apply the ES GuardPoint to the storage device. For details, see ["Guard the Linux Device with an ES GuardPoint" on page 159](#).

## Requirements for Efficient Storage GuardPoints on Linux

A LUN must meet the following requirements before it can be protected as an ES GuardPoint:

- The LUN is exported from a storage array must be enhanced with the CTE-Efficient Storage capability.
- The storage array exporting the LUN to the protected host must be a KMIP client registered with the same CipherTrust Manager as the protected host.
- The protected host must have direct physical access to the LUN through Fiber Channel Protocol (FCP) or iSCSI.
- The entire LUN must be protected as one and only one ES GuardPoint.
- In an ESXi environment:
  - The LUN added to a virtual machine must be configured for Raw Device Mapping in physical mode or the LUN must be part of a VVol datastore.
  - The LUN *cannot* be a VMDK or a disk in a datastore.
- Devices protected by an ES GuardPoint cannot currently be initialized/added as physical volumes for use by LVM. When LVM support is added, it will be announced in the CTE Release Notes.
- Existing devices divided into one or more logical partitions cannot be guarded as ES GuardPoints. Logical partitions in such devices cannot be accessed or separately guarded after guarding the device.  
For example, the logical partition `/dev/sda1` or `/dev/sda2` inside `/dev/sda` cannot be accessed after guarding `/dev/sda` as an ES GuardPoint. Using `/dev/securem/dev/sda1` is invalid as `/dev/securem/dev/sda1` is not a GuardPoint and cannot be guarded, and, as such, would not provide access to clear-text data on `/dev/sda1`.

**Note:** If you want to use VVol datastores with CTE-Efficient Storage devices, see the available documentation from the Storage Array system vendor.

## Initialize Linux Efficient Storage Devices

When you initialize a Linux efficient storage device, the process creates a private region on the device for CTE to write the Efficient Storage header along with metadata that identifies the ES device as a guarded device. The CTE private region also contains the metadata for the initial transformation of clear-text data on device to cipher-text, and for the subsequent transformation of cipher-text on the device to another encryption key as needed.

How you initialize the device depends on whether it is a new device or an existing device that already has data that needs to be transformed into cipher-text. For details, see:

- ["Initialize a New Linux Device" below](#)
- ["Initialize and Resize an Existing Linux Device" on the next page](#)

### Initialize a New Linux Device

Run the `voradmin esg config new` command to initialize a new device. The `new` option specifies that the device does not hold user data, and that CTE can reserve the first 63MB of storage on the device for the CTE private region. The remaining storage space is available for new user data. The device size reported to applications is the actual device size minus CTE private region size.

For a shared device that is accessed from multiple protected hosts, you must initialize the device only once and on only one protected host.

**Note**

To configure devices with multiple IO paths for Linux, see ["Guarding a CTE-Efficient Storage Device with Multiple IO Paths on Linux" on page 163](#).

1. Log into the device as root.
2. Run the `voradmin esg config new [-c <n>] <device-name>` command, where:
  - `new` (required) indicates that the device contains no data (it is a new disk). CTE will create the CTE private region at the beginning of the disk and the rest of the disk will be available for user data.
  - `-c <n>` (optional). If you use this option on Linux, CTE sets the number of data transformation jobs to run in parallel to the number specified in `<n>`. `<n>` can be an integer between 1 and 60 (default: 8).

Each data transformation job transforms 1MB worth of data and requires CPU resources in addition to three I/O operations as part of data transformation. Each job reads 1MB of data from the device, preserves the data in the CTE private region, rekeys the data to cipher-text, and writes the transformed data to the device. If you increase the number of parallel jobs, the data transformation process will complete faster but there will be an increased performance impact on the system. Only increase the `-c` option if you are certain that the system resources are available to handle the additional load.

The value for the `-c` option you specify here remains in effect for all subsequent data transformations (such as any data rekeys) until you specify a new value.

- `<device-name>` (required). Specifies the device name. For example, `/dev/sdh`.

For example, if you want to initialize a new Linux disk named `/dev/sdh` using 10 parallel data transformation jobs, you would specify:

```
voradmin esg config new -c 10 /dev/sdh
```

3. To verify that the disk has been initialized, run the `voradmin esg status` command.

```
voradmin esg status /dev/sdh
```

Device `/dev/sdh` is configured to guard as an Efficient Storage GuardPoint.
4. At this point the Administrator can protect the device as an ES GuardPoint through the CipherTrust Manager console. For details, see ["Guard the Linux Device with an ES GuardPoint" on the facing page](#).

#### Note

The initialization process prepares the device to be guarded but does not actually guard it. You need to assign an ES GuardPoint to the device in the CipherTrust Manager before the device is actually protected. In addition, the initialization process is only kept in memory until the device is guarded or rebooted. If the device is rebooted before you guard it, you will need to perform the initialization procedure again.

## Initialize and Resize an Existing Linux Device

If the device has existing data, you need to use the `voradmin esg config xform` command to initialize the disk for CTE-Efficient Storage. This command examines the current disk size and computes the size required to hold the existing data plus the CTE Private Region at the beginning of the device. After the CTE initialization is complete, you then need to resize the device before you can guard it with an ES GuardPoint.

The following procedure describes how to initialize the device for CTE. Note that the existing data is not altered in any way until after you perform this procedure and you guard the data with an ES GuardPoint. CTE does *not* begin transforming the data from clear-text to cipher-text until the ES GuardPoint has been applied and the encryption key has been pushed to the device through the GuardPoint Policy.

1. Log into the device as root.
2. Run the `voradmin esg config xform [-c <n>] <device-name>` command, where:
  - `xform` (required) indicates that the device contains existing data. CTE will transform all existing data on the device from clear-text to cipher-text as soon as you guard the device. The device will be inaccessible until the transformation is complete, and the device must remain offline during the entire transformation process. No user access will be permitted until all data has been transformed.



- `-c <n>` (optional). If you use this option on Linux, CTE sets the number of data transformation jobs to run in parallel to the number specified in `<n>`. `<n>` can be an integer between 1 and 60 (default: 8).

Each data transformation job transforms 1MB worth of data and requires CPU resources in addition to three I/O operations as part of data transformation. Each job reads 1MB of data from the device, preserves the data in the CTE private region, rekeys the data to cipher-text, and writes the transformed data to the device. If you increase the number of parallel jobs, the data transformation process will complete faster but there will be an increased performance impact on the system. Only increase the `-c` option if you are certain that the system resources are available to handle the additional load.

The value for the `-c` option you specify here remains in effect for all subsequent data transformations (such as any data rekeys) until you specify a new value.

- `<device-name>` (required). Specifies the device name. For example, `/dev/sdh`.

For example, if you want to initialize an existing Linux disk named `/dev/sdh` using 10 parallel data transformation jobs, you would specify:

```
voradmin esg config xform -c 10 /dev/sdh
```

Device `/dev/sdh` must be resized to at least 21100544 sectors (20606 MBs) before guarding as Efficient Storage GuardPoint.

In this case you must manually resize the Linux disk by at least 20606 MBs before you can guard it. After you guard the disk, you can expand it again later but you cannot shrink it unless you remove the GuardPoint.

3. To verify that the disk has been initialized, run the `voradmin esg status` command.

```
voradmin esg status /dev/sdh
```

Device `/dev/sdh` is configured to guard as an Efficient Storage GuardPoint.

4. At this point, you need to resize the device using your standard disk management tools before you can guard it. Make sure you increase the device size by at least the amount shown in the `voradmin esg config xform` message.

You cannot assign an ES GuardPoint to the device until it has been resized. If you do not resize the device, the GuardPoint assignment will fail.

5. After the device has been resized, the Administrator can protect the device as an ES GuardPoint through the CipherTrust Manager console, as described in "[Guard the Linux Device with an ES GuardPoint](#)" below.

#### Note

The initialization process prepares the device to be guarded but does not actually guard it. You need to assign an ES GuardPoint to the device in the CipherTrust Manager before the device is actually protected. In addition, the initialization process is only kept in memory until the device is guarded or rebooted. If the device is rebooted before you guard it, you will need to perform the initialization procedure again.

## Guard the Linux Device with an ES GuardPoint

After the device has been initialized, you can guard the device as an ES GuardPoint from the CipherTrust Manager console. For existing devices, as soon as the GuardPoint configuration has been pushed to the host and the status changes to guarded, CTE begins transforming the data on the disk using the encryption key associated with the GuardPoint Policy.

1. Log on to the CipherTrust Manager as an administrator of type Security with Host role permissions, type Domain and Security, or type All.
2. Make sure that you know what Policy you want to associate with the ES GuardPoint or create a new standard policy if needed. The policy you use for CTE-Efficient Storage must use an XTS-AES 256 key as the key rule. For details, see "[ES GuardPoint Encryption Keys](#)" on page 156.

3. Select **Hosts > Hosts** on the menu bar. The *Hosts* window opens.
4. Click the target host in the **Host Name** column. The Edit Host window opens to the General tab for the selected host.
5. Click the **GuardPoints** tab and then click **Guard**. The Guard window opens.
6. In the **Policy** field, select the Policy you identified or created earlier in this procedure. CTE will use the XTS-AES 256 key associated with this policy to encrypt the data on the device.
7. In the **Type** field, select either **Raw or Block Device (Auto Guard)** or **Raw or Block Device (Manual Guard)**.

If you select **Auto Guard**, CTE starts the guard process as soon as the policy is pushed to the host. You enable, disable, guard, and unguard the GuardPoint in the CipherTrust Manager. If you want to have the device automatically guarded and mounted at system start up, add the device to `/etc/fstab`. For details, see ["Auto Mount Options for File System Devices on Linux" on page 165](#).

If you select **Manual Guard**, You guard the GuardPoint on the protected host with the `secfsd -guard <path>` and `secfsd -unguard <path>` commands. At system startup, you must guard the device and then mount it. This gives you more control over when data transformations occur because CTE will not start encrypting or rekeying the device until you manually start the process.

8. In the **Path** field, add the path for the device you want to guard. For example, `/dev/sdh`.  
If you specify multiple paths in this field, all specified devices will be guarded and all will be encrypted with the encryption key specified in the associated policy.
9. Make sure the **Efficient Storage** check box is checked. If this option is not selected, the host will *not* enable the device as an ES GuardPoint.
10. Click **OK**.

The CipherTrust Manager pushes the policy and the GuardPoint configuration to the host and the CTE agent on the host writes the ES Header into the CTE private region for the specified devices. If this is a new device, the status changes to guarded and the disk is available for user access immediately.

If there is existing data on the device, CTE begins transforming the data from clear-text to cipher-text as soon as the GuardPoint configuration is available and the device status changes to guarded. The device will remain inaccessible until this data transformation completes. The length of time required to transform the data depends on the amount of existing data and the number of parallel data transformation jobs specified on the `voradmin config` command. For details, see ["Data Relocation on Existing Linux Devices" on the facing page](#) and ["Data Transformation on Existing Linux Devices" on the facing page](#).

To see the data transformation progress, use the `voradmin esg xform status <device-name>` command, as described in ["Viewing Device Status and the ES Header on Linux" on page 164](#).

After the device is initialized and guarded, the protected device must be accessed through the CTE device pathname. This pathname corresponds to the `secvm` device. For example, the Linux device pathname `/dev/sdh` becomes `/dev/secvm/dev/sdh` as soon as the process is complete.

#### Notes

- Be sure to use the `secvm` device name when using file system management tools such as `mkfs` and `fsck`.
- Do not use the device mapper names corresponding to ES GuardPoints for GuardPoint administration on protected hosts.



## Data Relocation on Existing Linux Devices

When you add an ES GuardPoint to a device that has been initialized with the `xform` option, CTE first relocates existing data in the region of the device designated as CTE Private Region. The data is relocated to the end of the LUN, into the new space allocated. The relocation occurs once when the device is guarded for the first time.

Relocation of data is transparent to applications accessing data through the ES GuardPoint. CTE will map application I/O requests over the CTE Private Region to the relocated region. After guarding the device, you can grow the device size further if necessary. However, you cannot shrink the device size.

## Data Transformation on Existing Linux Devices

As the ES Header is written before data transformation begins, the data transformed to cipher-text and written back to the device during data transformation process is subject to data reduction process through the storage array.

Existing devices populated with data are transformed from clear-text to cipher-text using the encryption key applied to each device. Data transformation is also called In-Place Data Transformation (CTE-IDT).

CTE-IDT is not the same as the legacy offline data transformation. CTE-IDT is a block level data transformation with built-in resiliency to recover from system crashes during the data transformation process. CTE-IDT uses the CTE Private Region on the device to manage the entire transformation process. CTE-IDT partitions the data on a device in segments of 1MB in size and transforms one or multiple segments, up to 60 segments, in parallel. The CTE-IDT process preserves existing data in a segment during transformation in the private region of the device, and then transforms the data in-place. CTE-IDT also maintains the segments undergoing transformation in the private region. In the event of system crash, CTE-IDT will recover the segments undergoing transformation at the time of crash and then resume the transformation process.

You can specify the number of segments to transform concurrently using `-c` option of the `voradmin` command when initializing the device. Choose a concurrency level that does not affect performance of your production workload. By default, CTE-IDT transforms 8 segments concurrently, if the concurrency level has not been specified through the `voradmin` command.

### Note

When choosing the concurrency level for your system you must consider the number of CPU cores, the total IOPS of your storage system and production workload, the size of the device to transform, and the duration for the data transformation.

Another advantage of CTE-IDT over legacy offline data transformation is that CTE-IDT does not require a separate policy for data transformation. With the same production policy applied to the device, CTE-IDT determines whether the device is in need of data transformation, per specification of `xform` option when device was initialized, and starts the CTE-IDT process when transformation is required. During the CTE-IDT process, access to the device is blocked until the CTE-IDT process completes.

```
voradmin esg status xform /dev/sdh
 Status: In-Progress
 Relocation Zone 22939648 (relocated = 1)
 SegSpc 27, Xformation Range: 3217 ... 4799, SegIDs: 4795 4796 4791 4792 4797 4798 4799
 KeyID: 988905 Key Name: ESG_DEMO_KEY_1
 Old KeyID: 0 Old Key Name: clear_key
dd if=/dev/secvm/dev/sdh of=/dev/null bs=512 count=1
dd: failed to open '/dev/secvm/dev/sdh': Resource temporarily unavailable
voradmin esg status xform /dev/sdh
 Status: Complete
 Relocation Zone 22939648 (relocated = 1)
 SegSpc 27, Xformation Range: 3217 ... 20189, SegIDs: none
 KeyID: 988905 Key Name: ESG_DEMO_KEY_1
 Old KeyID: 0 Old Key Name: clear_key
dd if=/dev/secvm/dev/sdh of=/dev/null bs=512 count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.000989039 s, 518 kB/s
```

## Thin-Provisioned Devices

CTE-IDT skips transforming thin-provisioned regions of a device. Data returned to CTE-IDT as sequence of clear-text zeros, in sector size granularity, is indication of possible sparse or un-allocated regions of the device that do not have to be transformed.

## CTE-IDT Recovery From Crash

CTE-IDT is fault tolerant in the event of system crashes. CTE-IDT keeps track of the transformation process over the entire device. In the event of a crash, CTE-IDT will automatically resume transformation from the point of failure as soon the GuardPoint is enabled after system startup.

If you find the transformation status set to **In-Progress** when the GuardPoint is not enabled, the **In-Progress** state reflects an earlier system crash after which the GuardPoint has not been enabled to recover from the interruption in the CTE-IDT process.

## Example of Creating an ES GuardPoint on an Existing Linux Device

The following figure shows an example of initializing an existing Linux device using `voradmin esg config xform` and guarding it as an ES GuardPoint from the viewpoint of the Linux root user.

```
voradmin esg status /dev/sdh
Native device /dev/sdh is not labeled as ESG

fdisk -l /dev/sdh

Disk /dev/sdh: 21.1 GB, 21103640576 bytes, 41218048 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 4194304 bytes

mkfs.xfs /dev/sdh
meta-data=/dev/sdh isize=512 agcount=4, agsize=1288064 blks
 = sectsz=512 attr=2, projid32bit=1
 = crc=1 finobt=0, sparse=0
data = bsize=4096 blocks=5152256, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal log bsize=4096 blocks=2560, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
mount -t xfs /dev/sdh /xfs
cp /bin/* /xfs
umount /xfs
voradmin esg config xform /dev/sdh
Device /dev/sdh must be resized to at least 41347072 sectors (40378 MBs) before guarding as Efficient Storage GuardPoint
```

Now you must resize the device on the storage array to the specified size or larger, and then guard the device on the CipherTrust Manager as ES GuardPoint using Manual Guard. For details about guarding the device on the CipherTrust Manager, see ["Guard the Linux Device with an ES GuardPoint" on page 159](#).

After resizing the device and configuring the GuardPoint in the CipherTrust Manager:

```
fdisk -l /dev/sdh
Disk /dev/sdh: 21.2 GB, 21169700864 bytes, 41347072 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 4194304 bytes

secfsd -guard /dev/sdh
secfsd: Path is guarded
```

At this point CTE begins data relocation and transformation. Data transformation will require some amount of time to complete. You can monitor the transformation process with the `voradmin esg status xform <device-name>` command.

```
voradmin esg status xform /dev/sdh
Status: In-Progress
Relocation Zone 41218048 (relocated = 1)
SegSpc 27, Xformation Range: 228 ... 229, SegIDs: 228
KeyID: 988905 Key Name: ESG_DEMO_KEY_1
Old KeyID: 0 Old Key Name: clear_key

Wait for data transformation to complete:

voradmin esg status vform /dev/sdh
Status: Complete
Relocation Zone 41218048 (relocated = 1)
SegSpc 27, Xformation Range: 7881 ... 20189, SegIDs: none
KeyID: 988905 Key Name: ESG_DEMO_KEY_1
Old KeyID: 0 Old Key Name: clear_key

voradmin esg status /dev/sdh
ESG Header on /dev/secvm/dev/sdh
Version: 1
Change: 0
Notifications: None
Storage Status: None
Private Region Size: 129024 sectors
Exported Device Size: 41218048 sectors
Key UUID: 2c3ee4af-010c-3593-898e-f2f62d29222d
Mount Path: None

mount -t xfs /dev/secvm/dev/sdh /xfs
for file in `bin/ls /xfs`; do cmp /bin/$file /xfs/$file; done
umount /xfs
```

## Guarding a CTE-Efficient Storage Device with Multiple IO Paths on Linux

Each individual IO path from a server node to a storage controller is treated as a separate device on the host. DM-Multipath on a Linux host provides a management framework to group the individual IO paths to the same LUN into a single multipath device. If you use DM-Multipath to manage devices on the protected host, the individual devices that correspond to each IO path to the LUN cannot be configured for guarding as ES GuardPoints, as those devices are under control of DM-Multipath. To guard such devices, you must guard the device mapper generated by DM-Multipath (multipathd) under the `/dev/mapper` directory.

### Note

CTE-Efficient Storage is the only feature of CTE that exclusively supports guarding of device mapper generated devices under DM-Multipath framework.

The following example illustrates the procedure for guarding a device mapper generated device with the alias name `/dev/mapper/mpathA`.

1. Create a standard policy using an XTS key as the key rule.
2. On the host, prepare the device to be configured as ESG using the `voradmin` command with `new` or `xform` option. For example:  

```
voradmin esg config new /dev/mapper/mpathA
```
3. On the CipherTrust Manager, guard `/dev/mapper/mpathA` as Device GuardPoint using the policy created above. Be sure to check the **Efficient Storage** check box.
4. For Manual-Guard configuration, enable the GuardPoint using the `secfsd` command as follows:  

```
secfsd -guard /dev/mapper/mpathA
```
5. For Auto-Guard, wait for the `/dev/mapper/mpathA` device to be guarded on the protected host.
6. Once the device is guarded, provide the pathname of the `secvm` device to applications and/or file system operations. For example, `/dev/secvm/dev/mapper/mpathA`.

## Linux System and ES GuardPoint Administration

## voradmin ESG Commands on Linux

The `voradmin` command is a command line utility for management of CTE specific configuration and status reporting. The `voradmin` command also supports configuration management related to ES GuardPoints (ESG).

For details about the Linux `voradmin esg` command options, see the man page for the `voradmin` command.

## Viewing Device Status and the ES Header on Linux

After you guard a device, you can view the status of that device using the `voradmin esg [xform] status <device-name>` command, where:

- `xform` (optional). If you specify this option, CTE shows the status of any data transformation processes happening on the device. If you do not specify this option, CTE displays the ES Header for the device.
- `<device-name>` (required). The standard Linux name of the device whose status you want to view. (For example, `/dev/sdh`.)

For example, if you want to view the ES Header for the Linux device `/dev/sdh`, you would enter:

```
voradmin esg status /dev/sdh
ESG Header on /dev/secvm/dev/sdh
 Version: 1
 Change: 0
 Notifications: None
 Storage Status: None
 Private Region Size: 129024 sectors
 Exported Device Size: 20842496 sectors
 Key UUID: 2c3ee4af-010c-3593-898e-f2f62d29222d
 Mount Path: None
```

If you want to view the data transformation status on `/dev/sdh`, you would enter:

```
voradmin esg xform status /dev/sdh
 Status: In-Progress
 Relocation Zone 22939646 (relocated = 1)
 SegSpc 27, Xformation Range: 3217 ... 4799, SegIDs: 4795 4796 4791 4792...
 KeyID: 988905 Key Name: ESG_DEMO_KEY_1
 Old KeyID: 0 Key Name: clear_key
```

The **Status** field displays **In-Progress** if a data transformation process is running, and **Completed** if the process has finished.

## File System Mount Points on Linux

You can create and mount a file system on an ES GuardPoint. CTE imposes one restriction on the mount point pathname selected for a device. Once you mount the device on a pathname, you cannot change the mount point to a different pathname. This restriction is enforced to allow the file system mount point to be guarded using a separate policy to enforce access control rules on the mounted file system namespace. For a use case involving a directory GuardPoint guarded over a mounted ES GuardPoint, see ["Use Case 3: Directory-Level GuardPoints" on page 171](#).

The following example shows the mount point of the ES GuardPoint as the `/xfs` directory. The example also shows a failed attempt to mount the file system on a different directory pathname.

```
voradmin esg status /dev/sdh
ESG Header on /dev/secvm/dev/sdh
 Version: 1
 Change: 0
 Notifications: None
 Storage Status: None
 Private Region Size: 129024 sectors
 Exported Device Size: 41218048 sectors
 Key UUID: b16445bd-dble-3a8f-b829-5893dd2fd0b0
 Mount Path: /xfs
umount /xfs
mkdir /other-xfs
mount -t xfs /dev/secvm/dev/sdh /other-xfs
mount: permission denied
mount -t xfs /dev/secvm/dev/sdh /xfs
```

## Auto Mount Options for File System Devices on Linux

ES GuardPoints containing file systems can also be added to the `/etc/fstab` configuration file for auto mount at startup or unmount at shutdown. An entry can be for an ES GuardPoint configured for Auto Guard or Manual Guard. For more information about Auto and Manual Guard options, see ["Guard the Linux Device with an ES GuardPoint" on page 159](#).

Use the device path corresponding to an ES GuardPoint device when specifying `fstab` entries, such as `/dev/secvm/dev/sdh`. Do not use the native device pathnames, such as `/dev/sdh`, or device mapper device names. You must also include several settings in the `fstab` entry for each ES GuardPoint, as shown in the following table:

| Option                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x-systemd.requires=secvm-barrier.service</code>     | Ensure that the ES GuardPoint is enabled before the device is mounted at startup and disabled after the device is unmounted at shutdown. The <code>secvm-barrier.service</code> service is a proxy for all the services that make up CTE.                                                                                                                                                                                                                                                                        |
| <code>nofail</code>                                       | The system boot will proceed without waiting for the ES GuardPoint device if it can't be mounted successfully.                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>x-systemd.wanted-by=&lt;es device&gt;.device</code> | Required for Linux distributions running <code>systemd</code> 242 or later. Instructs <code>systemd</code> to add a <code>Wants=</code> dependency on the ES device to ensure that, when the device becomes available, this mount operation is executed. <code>&lt;es device&gt;.device</code> is the name of the device specified in <code>fstab</code> with the <code>'/'</code> replaced with <code>'.'</code> . For example, <code>/dev/secvm/dev/sdb</code> becomes <code>dev-secvm-dev-sdb.device</code> . |

This is an example of an entry in `/etc/fstab` for an ES GuardPoint with an `xfs` file system that is mounted on `/xfs`:

```
/dev/secvm/dev/sdh /xfs xfs x-systemd.requires=secvm-barrier.service, \
x-systemd.wanted-by=dev-secvm-dev-sdh.device,nofail 0 0
```

For information about configuring `systemd` for CTE, see [Chapter 6: "CTE and systemd" on page 54](#).

## Linux System Utilities for Signing

The following table includes recommendations on the system and file system specific utilities for inclusion in the signature set to allow or deny root execution.

| EXT Utilities         | Deny/Allow | XFS                   | Deny/Allow | Generic Utilities  | Deny/Allow |
|-----------------------|------------|-----------------------|------------|--------------------|------------|
| <code>badblock</code> | Allow      | <code>fsck.xfs</code> | Allow      | <code>mount</code> | Allow      |

| EXT Utilities | Deny/Allow | XFS           | Deny/Allow | Generic Utilities | Deny/Allow |
|---------------|------------|---------------|------------|-------------------|------------|
| debugfs       | Deny       | mkfs.xfs      | Allow      | umount            | Allow      |
| e2freefrag    | Allow      | xfs_repair    | Allow      | dmsetup           | Allow      |
| e2fsck        | Allow      | xfs_admin     | Allow      |                   |            |
| e2image       | Allow      | xfs_bmap      | Allow      |                   |            |
| e2label       | Allow      | xfs_check     | Allow      |                   |            |
| e2undo        | Allow      | xfs_copy      | Deny       |                   |            |
| filefrag      | Allow      | xfs_db        | Deny       |                   |            |
| fsck.ext2     | Allow      | xfs_estimate  | Allow      |                   |            |
| fsck.ext3     | Allow      | xfs_freeze    | Allow      |                   |            |
| fsck.ext4     | Allow      | xfs_fsr       | Allow      |                   |            |
| logsave       | Allow      | xfs_growfs    | Allow      |                   |            |
| mke2fs        | Allow      | xfs_info      | Allow      |                   |            |
| mkfs.ext2     | Allow      | xfs_logprint  | Allow      |                   |            |
| mkfs.ext3     | Allow      | xfs_mdrestore | Allow      |                   |            |
| mkfs.ext4     | Allow      | xfs_metadump  | Allow      |                   |            |
| resize2fs     | Allow      | xfs_mkfile    | Deny       |                   |            |
| tune2fs       | Allow      | xfs_ncheck    | Allow      |                   |            |

## Changing the Encryption Key on Linux ESG Devices

To meet various compliance requirements, you may want to change the key that CTE has used to encrypt ES GuardPoints. Thales refers to this changing of encryption keys as “Key rotation” or “Rekey”. Unlike the CipherTrust Transparent Encryption - Live Data Transformation product offered by Thales for file systems on traditional storage devices, to change the encryption key on an ES GuardPoint, the device must be taken offline. The data on the device will be inaccessible during the key rotation process.

The key rotation process involves the following:

- Creating a new policy for key rotation
- Preparing the CTE-Efficient Storage device for key rotation
- Applying the new policy to the CTE-Efficient Storage device on the CipherTrust Manager

See the following sections for details of key rotation. If your organization has separated security duties, some of the steps below may need to be completed by different people.

## Creating a New Policy for Key Rotation

As part of rekeying the data on a CTE-Efficient Storage device, you must create a new policy with a key rule specifying the new key and include a second key rule that specifies the current key. The screenshot below depicts a policy named ESG\_DEMO\_POLICY\_2 for rekey:

Similar to the policy ESG\_DEMO\_POLICY\_1, the above production policy applies the key specified in the first key rule, ESG\_DEMO\_KEY\_2, on the device guarded with the policy. The second key rule is actually ineffective, because the first key rule is always selected for IO operations to devices guarded with ESG\_DEMO\_POLICY\_2, hence never selecting the second key rule for key operations.

The reason for inclusion of the second key rule is the availability of the key specified in the second key rule (ESG\_DEMO\_KEY\_1) to the protected host. Note that the name and the ID of the key applied to the data on ES GuardPoints are persistently recorded in the CTE private region on the device. However, the recording of the key name and ID in the private region may not be sufficient for availability of ESG\_DEMO\_KEY\_1 to the protected host without the second key rule listed in ESG\_DEMO\_POLICY\_2 policy. If another GuardPoint on the protected host guarded with a policy that specifies ESG\_DEMO\_KEY\_1 as a key rule, the key ESG\_DEMO\_KEY\_1 is available to the protected host, and the policy ESG\_DEMO\_POLICY\_2 does not have to include the second key rule listing ESG\_DEMO\_KEY\_1. Absence of any GuardPoint not guarded with a policy specifying ESG\_DEMO\_KEY\_1 causes the CipherTrust Manager not to push the key ESG\_DEMO\_KEY\_1 to the protected host.

Notice the resource set empty-∅ listed as the resource set in the second key rule. This is an empty resource set that does not associate any object in a guarded GuardPoint with the key specified in the key rule. An empty resource set is specified in the second key rule because a resource set is required for adding the second or additional key rules to the policy.

## Prepare an ES GuardPoint for Rekey

Before you apply the new policy, you must shutdown any applications accessing the ES GuardPoint. This also includes unmounting file system if the ES GuardPoint is a device mounted as a file system. The next step is to unguard the guarded device on the protected host and then on the CipherTrust Manager. Unguarding the device on the CipherTrust Manager deletes the ES GuardPoint on the host.

The following screenshot enumerates the administrative steps for rekeying ES GuardPoints. The following figure shows the transformation from ESG\_DEMO\_POLICY\_1 using ESG\_DEMO\_KEY\_1 to the new policy ESG\_DEMO\_POLICY\_2 using ESG\_DEMO\_KEY\_2.

```
secfsd -status guard | grep sdh
/dev/sdh ESG_DEMO_POLICY_1 manualrawdevice guarded guarded N/A
secfsd -unguard /dev/sdh
secfsd: Path is not guarded
voradmin esg status xform /dev/sdh
Status: Complete
Relocation Zone 22939648 (relocated = 1)
SegSpc 27, Xformation Range: 2017 ... 20183, SegIDs: none
KeyID: 988905 Key Name: ESG_DEMO_KEY_1
Old KeyID: 0 Old Key Name: clear_key
```

Notice the key name and the old key name associated with the device as recorded in the device private region.

Next, unguard the device on the CipherTrust Manager and then run the `voradmin` command to prepare the device for rekey. Be sure the device is unguarded on the CipherTrust Manager before running the `voradmin` command, otherwise the `voradmin` command refuses to prepare the device for rekey. Notice that the ES Header is temporarily deleted from the device in preparation for rekey.



```
secfsd -status guard | grep sdh
voradmin esg rekey /dev/sdh
Enter YES to prepare device /dev/sdh for rekey -> YES
voradmin esg status /dev/sdh
Native device /dev/sdh is not labeled as ESG
```

Next, guard the device on the CipherTrust Manager as an Efficient Storage device using the policy ESG\_DEMO\_POLICY\_2. After the CipherTrust Manager pushes the configuration on the ES GuardPoint, proceed with guarding the device on the protected host. The guard operation detects the policy/key change on the device and begins the CTE-IDT process to transform data from ESG\_DEMO\_KEY\_1 to ESG\_DEMO\_KEY\_2. During the CTE-IDT process, user access to the ES GuardPoint is blocked until IDT completes the transformation process.

```
secfsd -status guard | grep sdh
/dev/sdh ESG_DEMO_POLICY_2 manualrawdevice unguarded not guarded Inactive
secfsd -guard /dev/sdh
secfsd: Path is guarded
voradmin esg status xform /dev/sdh
Status: In-Progress
Relocation Zone 22939648 (relocated = 1)
SegSpc 27, Xformation Range: 3308 ... 3314, SegIDs: 3312 3314 3308 3309 3310 3311
KeyID: 988969 Key Name: ESG_DEMO_KEY_2
Old KeyID: 988905 Old Key Name: ESG_DEMO_KEY_1
```

After CTE-IDT completion you can restart application workloads on the guarded device.

```
voradmin esg status xform /dev/sdh
Status: Complete
Relocation Zone 22939648 (relocated = 1)
SegSpc 27, Xformation Range: 20189 ... 20189, SegIDs: none
KeyID: 988969 Key Name: ESG_DEMO_KEY_2
Old KeyID: 988905 Old Key Name: ESG_DEMO_KEY_1
mount -t xfs /dev/secvm/dev/sdh /xfs
```

## Resizing Guarded Efficient Storage Devices

Devices configured for Efficient Storage can be resized using the system-provided resizing utilities. If you are using a file system on the GuardPoint, you can mount the file system after resizing the device and then grow the file system to the new size using the appropriate utility such as `xfs_growfs` or `resize2fs`.



### WARNING

**Do not shrink ES GuardPoints. Due to relocation of user data from CTE private region, if you shrink the device, you may corrupt data on the device.**

1. Stop applications from accessing the GuardPoint.
  - Unmount the file system if the device is mounted.
  - Disable GuardPoints: auto-guard if it is enabled on the CipherTrust Manager, or manual-guard if it is enabled on the protected host.
2. Use the native disk management tools to resize the device.
3. After resizing the device, check the size of the device. For Linux, you can use the `fdisk -l` command.
4. If the reported size does not match what you expect, you may need to rescan your storage devices using the command appropriate for the device's connection type.
5. Once the expected size is achieved, enable the GuardPoint and restart your applications.



## Example: Resizing a Linux Device

The following example show the administrative steps to grow an XFS file system mounted on an ES GuardPoint. The example shows a device of size 41,347,072 sectors resized to 43,395,072 sectors .

```
fdisk -l /dev/sdh

Disk /dev/sdh: 21.2 GB, 21169700864 bytes, 41347072 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 4194304 bytes

xfs_info /dev/secvm/dev/sdh
meta-data=/dev/secvm/dev/sdh isize=512 agcount=4, agsize=1288064 blks
= sectsz=512 attr=2, projid32bit=1
= crc=1 finobt=0, spinodes=0
data = bsize=4096 blocks=5152256, imaxpct=25
= sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal bsize=4096 blocks=2560, version=2
= sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0

voradmin esg status /dev/sdh
ESG Header on /dev/secvm/dev/sdh
 Version: 1
 Change: 0
 Notifications: None
 Storage Status: None
 Private Region Size: 129024 sectors
 Exported Device Size: 41218048 sectors
 Key UUID: b1c445bd-d81e-3a8f-b829-5893dd2fd0b0
 Mount Path: N/A (GuardPoint not enabled)

umount /xfs
secfsd -unguard /dev/sdh
secfsd: Path is not guarded
```

In this example we resize the device on the storage array to 43395072 sectors. Notice that the `voradmin` command reports the old device size until the device is guarded.

```
fdisk -l /dev/sdh

Disk /dev/sdh: 22.2 GB, 22218276864 bytes, 43395072 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 4194304 bytes

voradmin esg status /dev/sdh
ESG Header on /dev/secvm/dev/sdh
 Version: 1
 Change: 0
 Notifications: None
 Storage Status: None
 Private Region Size: 129024 sectors
 Exported Device Size: 41218048 sectors
 Key UUID: b1c445bd-d81e-3a8f-b829-5893dd2fd0b0
 Mount Path: N/A (GuardPoint not enabled)

secfsd -guard /dev/sdh
secfsd: Path is guarded
voradmin esg status /dev/sdh
ESG Header on /dev/secvm/dev/sdh
 Version: 1
 Change: 0
 Notifications: None
 Storage Status: None
 Private Region Size: 129024 sectors
 Exported Device Size: 43266048 sectors
 Key UUID: b1c445bd-d81e-3a8f-b829-5893dd2fd0b0
 Mount Path: /xfs
```

The final step shown in below is to mount and resize the XFS file system to include the extended space into the file system.

```
mount -t xfs /dev/secvm/dev/sdh /xfs
xfs_growfs /xfs
meta-data=/dev/secvm/dev/sdh isize=512 agcount=4, agsize=1288064 blks
 = sectsz=512 attr=2, projid32bit=1
 = crc=1 finobt=0 spinodes=0
data = bsize=4096 blocks=5152256, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal bsize=4096 blocks=2560, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
data blocks changed from 5152256 to 5408256
xfs_info /dev/secvm/dev/sdh
meta-data=/dev/secvm/dev/sdh isize=512 agcount=5, agsize=1288064 blks
 = sectsz=512 attr=2, projid32bit=1
 = crc=1 finobt=0 spinodes=0
data = bsize=4096 blocks=5408256, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal bsize=4096 blocks=2560, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
```

## Use Cases involving Efficient Storage GuardPoints

ES GuardPoints support three use cases for managing customer's data in GuardPoints. This section describes those potential use cases.

### Use Case 1: Single Encryption Key

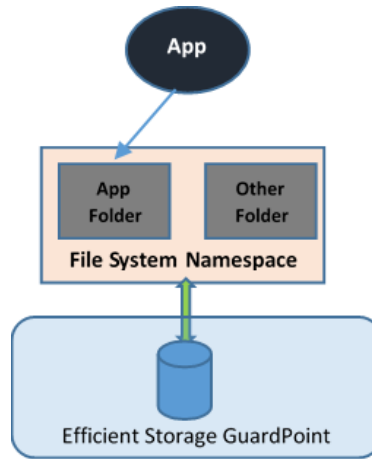
Applications, such as an Oracle Database, store structured data in one or multiple LUNs guarded as an ES GuardPoint. In this use case, a LUN may be an independent datastore or a member of a disk group managed by an application, for example an Oracle ASM disk group. In this use case, the policy applied to the GuardPoints specifies one key rule for encryption, and, potentially, a second key rule with an empty resource set for rekey. The policy may include access rules for user or process level access control.



### Use Case 2: Device-Level GuardPoints

Protect structured or unstructured data stored in data files. The data files are organized inside one or more directories or folders within a file system namespace, such as ext4 or XFS, without any protection on the folders or the file system namespace. In this use case, the file system resides in the device guarded as Efficient Storage using a policy with a key rule and *no user specified access rule*. *Access rules are not applicable in this use case and should not be used*. Similar to use case 1, Linux policies supporting this use case can also specify the second key rule with an empty resource set for rekey.

Figure 19-1: File system resides in device guarded as ES GuardPoint



## Example

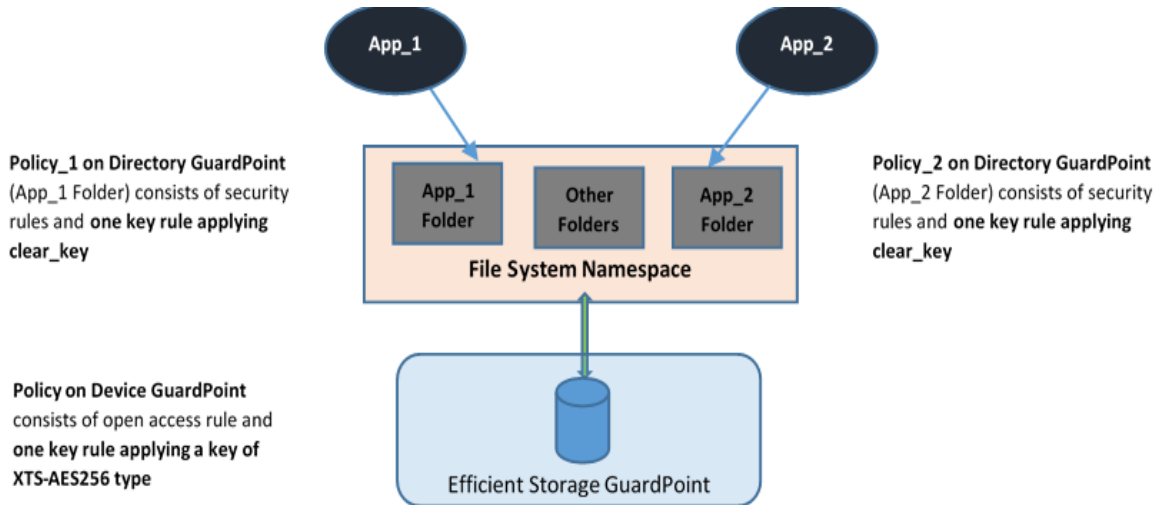
Below is an example of this use case where a Linux file system is created in an ES GuardPoint and then mounted. The policy used for the GuardPoint does not specify user or process-level access rules because I/O operations to the GuardPoint are from the file system module accessing the device on behalf of application I/O operations to the files inside the mounted file system.

```
secfsd -status guard | grep sdh
/dev/sdh ESG_DEMO_POLICY_2 manualrawdevice unguarded not guarded Inactive
voradmin esg config new /dev/sdh
secfsd -guard /dev/sdh
secfsd: Path is guarded
voradmin esg status /dev/sdh
ESG Header on /dev/secvm/dev/sdh
 Version: 1
 Change: 0
 Notifications: None
 Storage Status: None
 Private Region Size: 129024 sectors
 Exported Device Size: 43266048 sectors
 Key UUID: b16445bd-dble-3a8f-b829-5893dd2fd0b0
 Mount Path: None
mkfs.xfs /dev/secvm/dev/sdh
meta-data=/dev/secvm/dev/sdh isize=512 agcount=4, agsize=1352064 blks
= sectsz=512 attr=2, projid32bit=1
= crc=1 finobt=0, sparse=0
data = bsize=4096 blocks=5408256, imaxpct=25
= sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal log bsize=4096 blocks=2640, version=2
= sectsz=512 sunit=0 blks, lazy-count=1
realtime =none ctxtsz=4096 blocks=0, rtextents=0
mount -t xfs /dev/secvm/dev/sdh /xfs
mount | grep xfs
/dev/secvm/dev/sdh on /xfs type xfs (xw,relatime,seclabel,attr2,inode64,noquota)
```

## Use Case 3: Directory-Level GuardPoints

Protect structured or unstructured data stored in data files. The data files are organized inside one or multiple directories or folders within a file system namespace, such as ext4 or XFS, where the entire file system namespace is guarded with one policy as a Directory GuardPoint. In this use case, the file system resides in a device guarded as ES GuardPoint. Similar to use case 1, Linux policies supporting this use case can also specify the second key rule with an empty resource set for rekey.

**Figure 19-2: All Data in file system Device Encrypted through an ES GuardPoint**



The second policy protecting the device is the same policy as use case 2.

## Example

Below is an example of this use case where a file system created in a guarded device and mounted on `/xfs` is protected under a policy that denies root access to the files under `/xfs/dir1`:

```
mount | grep xfs
/dev/secvm/dev/sdh on /xfs type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
find /xfs -print
/xfs
/xfs/non-secret
/xfs/dir1
/xfs/dir1/secret
cat /xfs/dir1/secret
This file holds highly sensitive data.
cat /xfs/non-secret
This file does not hold sensitive information.
secfsd_guard /xfs
secfsd: Path is guarded
secfsd -status guard | grep sdh
/dev/sdh ESG_DEMO_POLICY_2 manualrawdevice guarded guarded N/A

ls /xfs
dir1 non-secret
cat /xfs/dir1/secret
cat: /xfs/dir1/secret: Permission denied
cat /xfs/non-secret
This file does not hold sensitive information.
```

As depicted above, the root user is denied access to read/write the files associated with the resource set representing files under `/xfs/dir1` subdirectory.

## Challenges with Root Access on Linux

As demonstrated in the example in use case 3, data is protected at two levels using two separate policies and GuardPoints. The data is encrypted at the device level through the policy on the device guarded as Efficient Storage, and user access controls are enforced at the file system level through the policy on the mount point directory. Splitting the full protection through separate GuardPoints poses new challenges with respect to root privilege on Linux.

With the GuardPoint on the file system mount point enabled, the access rule(s) denying root access is enforced. However, when the GuardPoint on the file system mount point is disabled, root gains full access to the files in the file system. As shown below, the file holding secret information and protected against root is exposed as soon as the GuardPoint on file system mount point is disabled.

```
secfsd -status guard | grep ESG
/dev/sdh ESG_DEMO_POLICY_2 manualrawdevice guarded guarded N/A
/xfS ESG_MOUNTED_FS_POLICY manual guarded guarded N/A
cat /xfS/dir1/secret
cat: /xfS/dir1/secret: Permission denied

secfsd -unguard /xfS
secfsd: Path is not guarded
cat /xfS/dir1/secret
This file holds highly sensitive data.
```

The next two sections describe the challenges with root access and solutions to overcome these challenges.

### Challenge 1 - Deny root access to the files in mounted Linux file system

As the above example shows, the policy ESG\_MOUNTED\_FS\_POLICY denies the root user access to the files associated with the resource set dir1. Enforcement of the rule become ineffective as soon as the GuardPoint on the file system mount point is disabled. Since data is in clear-text at the file system level, root would gain access to clear-text in the files associated with the resource set, which includes files with sensitive information.

The solution to this problem is to force the file system to unmount when the GuardPoint on the file system mount point is disabled. Basically, the file system is guarded/enabled immediately when the file system mounts, and the file system is unmounted as soon as the GuardPoint is disabled. To enforce this, the GuardPoint on the file system mount point directory must be guarded with **Auto Mount** option checked on the CipherTrust Manager. With this option, CTE immediately guards the mount point directory as soon as the file system mounts, and similarly, CTE disables the GuardPoint before unmounting the file system. As shown below, the file system mount point is guarded with **Auto Mount** option checked.

This solution imposes one policy protecting the entire file system namespace. Enforcement of a single policy over entire file system namespace may seem restrictive if you wish to impose different sets of access rules to different directories within the file system name. Basically, your option of enforcing one policy with a set of specific access rules for guarding a specific directory within the mounted file system namespace is not possible with this solution. Instead, you have to create a resource set for each directory, which you would have guarded, and specify the desired access rules specific to the directory through association of the rule with the resource set. Let's see the effect of auto-mount on root user attempts to view files that root users are not allowed to read. As shown below, the GuardPoint /xfS is automatically mounted as soon as the file system mounts, and the file system unmounts as soon as the GuardPoint is disabled, hence there is no opportunity for root, or any other privileged user, to read the protected files.

```
mount -t xfS /dev/secvm/dev/sdh /xfS
secfsd -status guard | grep ESG
/dev/sdh ESG_DEMO_POLICY_2 manualrawdevice guarded guarded N/A
/xfS ESG_MOUNTED_FS_POLICY automount guarded guarded N/A

cat /xfS/dir1/secret
cat: /xfS/dir1/secret: Permission denied

umount /xfS
secfsd -status guard | grep ESG
/dev/sdh ESG_DEMO_POLICY_2 manualrawdevice guarded guarded N/A
/xfS ESG_MOUNTED_FS_POLICY automount guarded not guarded Inactive

cat /xfS/dir1/secret
cat: /xfS/dir1/secret: No such file or directory
```

As depicted above, the policy ESG\_MOUNTED\_FS\_POLICY enforces a single rule to block root access to the files under the dir1 subdirectory under the mounted file system GuardPoint. We need to add another security rule to the policy to grant root user access to read the files under dir2 subdirectory under the guarded file system mount point. Note that dir1 and dir2 may have been guarded separately under different policies.



The resource set ESG\_FS\_ResourceSet consists of the binary files listed in the next screenshot. Let's walk through the steps to add the security rule to limit root access:

1. Select the system utilities that must be granted access to Efficient Storage devices.
2. Click **Add** to add a signature set.
3. On the **Add Signature Sets** page, enter the name of the signature set to create and then click on OK. This creates an empty signature set with the name you have selected for the signature set. The name appears on the **Signature Sets** page.
4. On the **Signature Sets** page, click on the name of the signature set to get to the **Edit Signature Set** page to edit your signature set.
5. On the **Edit Signature Set** page, click on the **Source** tab to select the protected host where the file system utilities are located. Click **Select** to select the protected host.
6. Click **Add** to add each system utility from the selected host to the signature set. After adding the system utilities to the set, then click **Sign** to sign the binary files on the selected protected host.

At this point, you have created a signature set consisting of the system utilities that root is allowed to execute on Efficient Storage devices. Next, create a process set from the signature set. The process set will be included in a security rule in the policy protecting the ES GuardPoint. The security rule will allow root or other privileged users to access the device only through the system utilities in the signature set. Continue with the following steps:

1. For Policies in the top menu bar, then click on **Manage Policies**, and then click on **Process Sets** to get to **Process Sets** page.
2. Click **Add** to get to **Add Process Set** page to add a process set.
3. Enter the name of the process set in the **Name** entry and then click **Add** to get to **Add Process** page.
4. Click **Select** next to the **Signature Set** entry to select a signature set. Select the signature set that you just created.

After clicking **Select**, you will be on the **Select Signature Reference** page, and you will see the signature set you just created, for example, the ESG\_FS\_Uilities signature set. Click on the **Select** button to the left of the signature set and then click on **Select Signature Reference**. You will go back to **Add Process** page with the selected signature set name entered in **Signature Set** box.

5. Click **Select** next to the **Host** entry to select the protected host the process set. On the Select a Host to Continue page, select the protected host and then click on **Select**.
6. On the **Add Process Set** page, click OK to create the process set associated with the selected signature set and the protected host.
7. You have selected the signature set and the protected host. Skip **Directory** and **File** entries, and then click OK to go back to **Add Process Set**.
8. Edit the policy protecting ES GuardPoints to add a security rule. In the following figure, the process set that you just created is included in the security rule. The rule allows only the processes listed in the process set to access the ES GuardPoints. This rule prevents any privileged user from reading or dumping the content of an ES GuardPoint.

After adding the security rule, the policy protecting ES GuardPoints will be the same policy as ESG\_DEMO\_POLICY\_2.

After applying the revised policy over the ES GuardPoint, root can no longer dump the contents of the device.



```
secfsd -status guard | grep ESG
/dev/sdh ESG_DEMO_POLICY_2 manualrawdevice unguarded not guarded Inactive
/xfs ESG_MOUNTED_FS_POLICY automount guarded not guarded Inactive
secfsd -guard /dev/sdh
secfsd: Path is guarded
mount -t xfs /dev/secvm/dev/sdh /xfs

cat /xfs/dirl/secret
cat: /xfs/dirl/secret: Permission denied

dd if=/dev/secvm/dev/sdh bs=1048576 | grep --binary-files=text "file holds highly sensitive data"
dd: failed to open '/dev/secvm/dev/sdh': Permission denied
```

## Best Practices for the Migration of a legacy Raw Device Guardpoint to an Efficient Storage Guardpoint

Starting with CTE Agent v7.1.1, users can migrate an existing Raw Device GuardPoint to an Efficient Storage GuardPoint (ESG). The migration steps described in this section make use of the ESG In-Place Data Transformation process to re-encrypt existing data, within a Device GuardPoint that was using an AES-CBC encryption key, to a new ESG GuardPoint using an AES-XTS encryption key, all while preserving the existing data on the device.

You must use an In-Place Data Transformation policy for migration of legacy Raw Device GuardPoints to ESG.

Once the migration to an ESG GuardPoint is completed, the new ESG GuardPoint automatically makes use of the data reduction feature if the following conditions are met:

- The host/agent is registered to a CipherTrust Manager domain that is KMIP-enabled
- The XTS keys used with the In-Place policy as part of migration has KMIP Accessible enabled

## Migration Prerequisites

As part of the migration of a Raw Device GuardPoint to an Efficient Storage GuardPoint, consider the following prerequisites:

- Access to the device, for all users and applications, is blocked during the entire migration process
- Devices that will be migrating to an Efficient Storage GuardPoint are resized to provide sufficient disk space for ESG metadata
- Agent is running CTE v7.1.1 or a successive version
- An In-Place Data Transformation policy is required to guard the device as an ESG GuardPoint. When creating the policy, set the AES-CBC key that is used in the Standard policy as the Current Key. Set the AES-XTS key as the New Key in the key rule.

## Migration

Before beginning the migration of a Raw Device GuardPoint to an ESG GuardPoint, stop all user and application access to the Raw Device GuardPoint.

1. Backup the device if possible.
2. On the CipherTrust Manager, navigate to the client tab and select the target client Name that contains the raw device that you want to migrate.
3. In the **GuardPoints** tab, select the target GuardPoint and click **Unguard** to unguard the raw device GuardPoint.



4. On the client, run the following command to configure the device to be guarded as an ESG GuardPoint, type:

```
voradmin esg config xform <device-name>
```

**Note:** Make sure that you specify the native Linux device name of your device, such as `/dev/sdh` in the `voradmin` command, and resize the device before guarding the device using the IDT policy.

5. On the CipherTrust Manager, navigate to the client tab and select the target Host Name.
6. Under the **GuardPoints** tab, click **Guard** to set a Raw or Block Device GuardPoint using the In-Place Data Transformation policy created for this device for migration. Make sure that the Efficient Storage option is selected. If this option is not selected, the host will not enable the device as an ESG GuardPoint.
7. Click **OK**.

CTE begins transforming the data using the previous AES-CBC key and encrypting to the new AES-XTS key as soon as the device is guarded. During data transformation, the device remains inaccessible until this process completes. The length of time required to transform the data depends on the amount of existing data and the number of parallel data transformation jobs specified during the `voradmin config` command.

For details, see ["Data Relocation on Existing Linux Devices" on page 161](#) and ["Data Transformation on Existing Linux Devices" on page 161](#).

8. To see the data transformation progress, use the `voradmin esg xform status <device-name>` command, as described in ["Viewing Device Status and the ES Header on Linux" on page 164](#).
9. After transformation is completed and the device is guarded, the protected device must be accessed through the CTE device pathname that corresponds to the raw device.

For example, the Linux device pathname `/dev/sdh` becomes `/dev/secvm/dev/sdh` as soon as the guard process completes.

## Use Case 4: Using CTE-Efficient Storage with LVM

Security rules for access control enforcement on I/O operations are based on the context of real users and/or processes requesting the I/O operations. For ESG devices, security rules are checked and enforced at the CTE `secvm` layer. If the context of the real user or process that requested the I/O operation is not available at the `secvm` layer, then the enforcement of access rules is invalid and may have unpredictable results and potentially cause system failures.

The previous use cases have described environments where CTE can get the context of the real users or processes requesting I/O operations at the `secvm` layer and can properly enforce the access rules in those environments. For example, as described in ["Use Case 3: Directory-Level GuardPoints" on page 171](#), CTE can detect when an ESG device is mounted with a file system and it can apply the access rules in the policy on the ESG device without enforcing the rules on I/O operations from the file system layer.

However, there are other environments where CTE cannot determine how the ESG device is actually used and whether a real user or process context is available at the `secvm` layer. In these environments, CTE cannot properly enforce the access control rules on the ESG device. An example of such an environment is Linux LVM (Logical Volume Manager). A physical device under LVM control is a shared storage resource that can be fully or partially allocated to a logical volume. Adding an ESG device as a physical device to LVM is fully supported for data encryption but not for access control enforcement. In this case, access rules cannot be enforced because when the logical volume with file system is mounted, the mount operation is on the logical volume, not the ESG device. Consequently, CTE cannot determine that the ESG device is part of mounted file system, and therefore, it cannot get the real user or process context for applying access rules.

In order to use CTE in an environment like LVM, the CTE system administrator needs to explicitly tell CTE whether or not it should enforce the access rules for I/O operations on the device when they initialize the device using the `voradmin esg config` command. As described in ["Initialize a New Linux Device" on page 157](#), this initialization must be done *before* the device is guarded as an ES GuardPoint for the first time.

In addition, for LVM specifically, the device must be a new device with no existing data. Existing LVM disks cannot be used with CTE unless CTE can get the proper context at the `secvm` layer.

When you initialize the new device, use the `voradmin esg config -noacc new [-c <n>] <device-name>` command where:

- `-noacc` (required) tells CTE to disable access control rules on this device at the `secvm` layer level.
- `new` (required) indicates that the device contains no data (it is a new disk).
- `<device-name>` (required). Specifies the device name. For example, `/dev/sdh`.

For example, if you want to initialize a new LVM Linux disk named `/dev/sdh`, you would specify:

```
voradmin esg config -noacc new /dev/sdh
```



#### CAUTION

When you use `-noacc`, CTE will not perform *any* access checks on read/write operations to the specified device, even if the policy for guarding the device includes security rules enforcing read and/or write access checks. Before you use this option, make sure that your environment requires it. You should always use the options described in the first three use cases if at all possible, so that CTE can apply access controls to all I/O operations and therefore perform all required read/write access checks.

The following example shows the steps for initializing and guarding the device `/dev/sdb`, then adding or using the guarded device under LVM to create a file system.

1. Initialize the device using the `-noacc` option:

```
voradmin esg config -noacc new /dev/sdb
```

2. Guard the device as described in ["Guard the Linux Device with an ES GuardPoint" on page 159](#).
3. Create and mount the file system.

```
pvcreate /dev/secvm/dev/sdb
vgcreate secvm_sdb_vg /dev/secvm/dev/sdb
lvcreate -y -l 100%FREE -n my_volume
mkfs.xfs /dev/mapper/my_volume
mount /dev/mapper/my_volume /mnt
```

## Alerts and Errors on Linux

This section lists the alerts and errors that may be encountered during system operations.

### Encryption key on device has not been made available

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the encryption key specified in the policy for the GuardPoint has not been made available to the protected host.

**Solution:** Check the host's connectivity with the CipherTrust Manager.

## Specified policy disagrees with metadata set on the Guard Path

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the key specified in the policy for the GuardPoint does not match with the key stored in the ES Header.

**Solution:** Un-guard the device and check the name and UUID of the key in the ES Header using `voradmin esg status <device-name>` and `voradmin esg status xform <device-name>` and, compare the name and UUID with the key name specified in the policy. Correct the discrepancy and re-guard the device.

## Device has not been configured for Efficient Storage

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the device has been properly guarded as ES GuardPoint on the CipherTrust Manager but the device has not been initialized for guarding as Efficient Storage.

**Solution:** The most probable cause of this error is that you did not initialize the device for guarding as Efficient Storage on protected host. It's also possible that the guarded device has already configured for guarding as ES GuardPoint. If the device needs to be initialized for guarding as Efficient Storage, follow the steps to initialize a device for guarding as Efficient Storage.

## Device not resized for guarding as Efficient Storage

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the newly guarded ES GuardPoint, which has been initialized with `xform` option of `voradmin`, has not been resized to accommodate storage space for CTE private region.

**Solution:** Unguard the GuardPoint from the CipherTrust Manager, resize the LUN and verify that the host sees the expanded size, and then guard the GuardPoint from the CipherTrust Manager.

## Data transformation failed

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the protected host encountered an error while transforming the data on the device during IDT.

**Solution:** Consult with the system and/or storage admin to check on the health of the LUN in the storage array. You may contact Thales Support for troubleshooting and recovery if there has not been a report of any error on the LUN.

## Data transformation in progress

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the protected host is transforming the data on the device.

**Solution:** You must wait for data transformation to complete. Check the status of transformation by running `voradmin esg status xform <device name>`. Access to the device is blocked until transformation completes.

## Device <device-name> is configured to guard as Efficient Storage GuardPoint

This error message is reported by the `voradmin` command when initializing a device that has already been initialized for guarding as Efficient Storage.

**Solution:** The device has already been initialized for guarding as Efficient Storage and is probably waiting to be guarded as an ES GuardPoint. Alternatively, if the device was incorrectly initialized for guarding as Efficient Storage, you can remove the configuration by running `voradmin esg delete <device-name>`.

### Device <device-name> is configured as Efficient Storage GuardPoint

This error message is reported by the `voradmin` command when initializing a device that is already being guarded as an Efficient Storage GuardPoint.

**Solution:** None. The device is already an Efficient Storage device.

### GuardPoint for device <device-name> still guarded on CipherTrust Manager

This error message is reported by the `voradmin` command when attempting to initialize a device for rekey or removing the device as Efficient Storage.

**Solution:** Unguard the GuardPoint from the CipherTrust Manager, wait for the protected host to process the update, and then rerun the `voradmin` command.

### Failed to open device <device-name>, error Device or resource busy

This message occurs when `voradmin` detects that the target device is busy.

**Solution:** The device may already be in use by other application. Rerun the `voradmin` command when the device is no longer in use.

### Device <device-name> is not configured as Efficient Storage

This error message is reported by the `voradmin` command when attempting to delete a device as ES GuardPoint.

**Solution:** This message indicates that the target device has been not initialized for guarding as Efficient Storage. The message may also be reported if the specified device has been initialized for guarding as Efficient Storage but it has not been guarded yet. In this case, it removes the preparation made by `voradmin`. You can remove the Efficient Storage configuration status on the device by running `voradmin esg delete <device-name>`. You may see the same error message again, and if you do, you can ignore it.

### Abort! Error: Could not stop secfs, secvm device(s) busy

This error occurs during CTE shutdown when there is a busy CTE protected device.

**Solution:** Verify that all applications directly accessing `secvm` protected GuardPoints have been shut down. Ensure that all file systems on top of a `secvm` protected devices are under the control of `systemd` and have been unmounted before attempting to shut down the agent.

### Abort! Error: Could not unmount file systems

This error occurs during CTE shutdown when a file system under the control of `systemd` fails to unmount.

**Solution:** Verify that file systems on top of an ES GuardPoint devices are not busy and then rerun the agent shut down command.

### A dependency job for `esg.mount` failed. See '`journalctl -xe`' for details

This error occurs during CTE startup when system fails to mount a file system. This error message is typically accompanied by a long timeout during the CTE startup process.

**Solution:** Check that the underlying device is available and that the ES GuardPoint was successfully applied on the device. Once the device is available, the file system will automatically finish mounting.

### **ESG-ALERT: IO error on header for [GuardPoint]**

This is an alert message to the CipherTrust Manager. It occurs when CTE encounters a general error when attempting to access the private region on an Efficient Storage device.

**Solution:** An I/O error attempting to read or write to the device may have been caused by errors on the host or storage array. Consult with the system and/or storage admin to check on the health of the LUN in the storage array. You may contact Thales Support for troubleshooting and recovery if there has been no report of errors on the LUN.

### **ESG-ALERT: Data transformation failure on [GuardPoint]**

This is an alert message to the CipherTrust Manager. It occurs when protected host encounters an error transforming the data on device during IDT process.

**Solution:** Contact Thales Support for troubleshooting and recovery.

### **ESG-INFO: Data transformation complete on [GuardPoint]**

This message is a notification to the CipherTrust Manager admin that the protected host has completed the data transformation of the specified ES GuardPoint.

### **ESG-ALERT: Failed to resize <device-name>**

This alert is a notification to the CipherTrust Manager admin that the protected host has failed to update the change to the device size in the ES Header on the device.

**Solution:** Contact Thales Support for troubleshooting and recovery.

### **FSADM-ALERT: ESG required Signature Set for system utilities may have to be resigned**

This alert is a notification to the CipherTrust Manager admin that the recent system upgrade to the protected host may have updated the binary files listed in the signature set for restricting root access.

**Solution:** Upon this notification you must immediately re-sign the affected or all the binary files to prevent them from accessing protected data. Refer to "[Linux System Utilities for Signing](#)" on page 165.

### **File System is not automatically mounted after IDT completes**

An Efficient Storage device with a file system that has been configured to automatically mount may fail to automatically mount while the device undergoes data transformation through IDT. Following is a sample log message in the kernel ring buffer that reports a failed attempt to access a device during IDT. You can ignore this message.

```
Vormetric SecVM: secvm_map during IDT ((dc 00000000c3537611))
```

When access to a device is to mount the file system, the kernel ring buffer may also report a second error message indicating that a file system failed to mount due to data corruption. You can ignore this message. This occurs because the mount command is issued after the device has been guarded but before data transformation through IDT is complete. While IDT is in progress, the device cannot be used, so any attempt to mount the file system will fail.

**Solution:** Manually mount the file system after IDT is complete.

# Chapter 20: In-Place Data Transformation for Linux

---

This section contains the following topics:

|                                                                                                           |     |
|-----------------------------------------------------------------------------------------------------------|-----|
| Introduction to In-Place Data Transformation (IDT) .....                                                  | 182 |
| Requirements for IDT-Capable GuardPoints .....                                                            | 182 |
| IDT-Capable GuardPoint Encryption Keys .....                                                              | 183 |
| The CTE Private Region and IDT Device Header .....                                                        | 184 |
| Guarding an IDT-Capable Device on Linux .....                                                             | 186 |
| Changing the Encryption Key on Linux IDT-Capable Devices .....                                            | 193 |
| Guarding an IDT-Capable Device with Multiple IO Paths on Linux .....                                      | 196 |
| Linux System and IDT-Capable GuardPoint Administration .....                                              | 197 |
| Resizing Guarded IDT Devices .....                                                                        | 199 |
| Use Cases Involving IDT-Capable GuardPoints .....                                                         | 200 |
| Best Practices for the Migration of a legacy Raw Device Guardpoint to an In-Place Device Guardpoint ..... | 200 |
| Alerts and Errors on Linux .....                                                                          | 200 |

## Introduction to In-Place Data Transformation (IDT)

CTE offers In-Place Data Transformation (IDT) Capable Device GuardPoints on Linux. IDT-Capable GuardPoints allow you to guard devices by transforming the plain-text data to cipher-text on the host device. The data transformation process is called In-Place Data Transformation (IDT). The term “IDT-Capable” refers to the data transformation capability available on IDT-Capable GuardPoints.

IDT is not the same as the legacy offline data transformation. IDT is a block level data transformation with built-in resiliency to recover from system crashes during the data transformation process. IDT uses the CTE Private Region to manage the entire transformation process (For details, see ["The CTE Private Region and IDT Device Header" on page 184](#)).

IDT partitions the data on a device in segments of 1MB in size and transforms one or multiple segments, up to 60 segments, in parallel. The IDT process preserves existing data in a segment during transformation in the CTE Private Region, and then transforms the data in-place. IDT also maintains the segments undergoing transformation in the CTE Private Region. In the event of system crash, IDT will recover the segments undergoing transformation at the time of crash and then resume the transformation process.

Another advantage of IDT over legacy offline data transformation is that IDT does not require a separate policy for data transformation. Instead, IDT allows you to initialize each device as either a “new device” with no existing data or as an “existing device” with existing data that needs to be transformed. You can then apply any IDT policy to any combination of new and existing devices and IDT will immediately guard the new devices while starting the IDT transformation process on the existing devices. New devices are immediately available for use while existing devices are inaccessible until the IDT process completes and all data has been converted from plain-text to cipher-text.

## Requirements for IDT-Capable GuardPoints

- All versions of CM work with IDT-Capable GuardPoints.
- The host server must use the Advanced Encryption Standard instruction set (AES-NI).



- The policy assigned to the IDT-Capable GuardPoint must be an “In-Place Data Transformation” and use an XTS-AES 256 encryption key.
- In order to create an IDT-Capable GuardPoint on a raw device, the device must be either:
  - Exported from an external storage system to the host device.
  - On a locally-attached disk.
- Devices protected by an IDT-Capable GuardPoint cannot currently be initialized/added as physical volumes for use by LVM. When LVM support is added, it will be announced in the CTE Release Notes.
- Existing devices divided into one or more logical partitions *cannot* be guarded as IDT-Capable Device GuardPoints. Logical partitions in such devices cannot be accessed or separately guarded after guarding the device.

For example, the logical partition `/dev/sda1` or `/dev/sda2` inside `/dev/sda` cannot be accessed after guarding `/dev/sda` as IDT-Capable GuardPoint. Using `/dev/securevm/dev/sda1` is invalid as `/dev/securevm/dev/sda1` is not a GuardPoint and cannot be guarded, and, as such, would not provide access to clear-text data on `/dev/sda1`. However, you can guard individual partitions, such as `/dev/sda1` or `/dev/sda2`, as IDT-Capable GuardPoints without guarding the entire `/dev/sda` device.

## IDT-Capable GuardPoint Encryption Keys

IDT-Capable GuardPoints must be encrypted using XTS-AES 256 keys. An XTS-AES 256 type key is a 512-bit key composed of two components:

- The first 256 bits of the key is the AES 256 encryption key component.
- The second 256 bits is the tweak component.
- For CM, refer to “[Creating a New Key](#)” for more information.

The Key Manager (CipherTrust Manager) generates a UUID, along with other relevant attributes, for each newly-added key. It then provides the key and its attributes to the CTE protected host when the policy containing the key is pushed to the host device. CTE stores the key and its attributes, including the key's UUID, in the IDT Device Header. The first time a device is guarded as an IDT-Capable GuardPoint, CTE writes the IDT Header on the device before data transformation takes place, if data transformation is required.

### Key Attributes - Example

The following describes the parameters you should specify to add a new XTS-AES 256 key named `IDT_DEMO_KEY_1`. Note the algorithm and encryption mode specified for the key.

- Algorithm: **AES**
- Size: **256**
- Encryption Mode: **XTS**

Note the UUID of the key. This UUID is stored in the IDT Device Header on all devices encrypted with this key, allowing you to verify which key is being used on each device.

## Policy Requirements for IDT-Capable GuardPoints

IDT-Capable GuardPoints require a policy of type **In-Place Data Transformation** with a single key rule specifying the key names for **Current Key** and **New Key**. The **Current Key** name is `clear_key` or an XTS-AES 256 key name, depending on whether the data on the device has already been encrypted.

- If there is no existing data on the device or if the existing data on the device has not yet been encrypted, specify `clear_key` for **Current Key**. In the **New Key** field, specify the name of the XTS-AES 256 key that you want to use to encrypt the data on the device.
- If the existing data on the device has already been encrypted, specify the name of the key used to encrypt the data in the **Current Key** field and the name of the new XTS-AES 256 key you want to use to rekey the data in the **New Key** field. When the policy is pushed to the host, CTE will rekey the data on the device using the key specified in **New Key**. In other words, the **New Key** field specifies the XTS-AES 256 production key name to apply to device.

In all cases, the **New Key** field specifies the XTS-AES 256 production key name that you want to use to encrypt the data on the device. After you guard an existing device with an In-Place Data Transformation policy, CTE transforms the existing data using the New Key. When the process is finished, the New Key becomes the Current Key for that device, and all data will be encrypted or decrypted with that key. The IDT Device Header contains the UUID of the key currently being used to encrypt/decrypt data on the device. To view the current key UUID, use the `voradmin idt status <device-name>` command.

You may add security rules to restrict certain user/process access to protected devices. For suggestions, see ["Use Cases Involving IDT-Capable GuardPoints" on page 200](#).

A simple IDT policy requires:

- Policy type: In-Place Data Transformation
- Simple security rule that permits access to all users and programs
- Current key: `clear_key`
- New key: an `IDT_KEY` for encrypting the data on any device associated with the policy

## The CTE Private Region and IDT Device Header

IDT-Capable GuardPoints require a small amount of disk space in the standard CTE Private Region. The reserved space is where CTE stores metadata information to identify IDT-Capable GuardPoints and to perform all data transformation and rekey operations in a resilient manner to avoid data loss or integrity issues due to system failures. The IDT-specific reserved space within the CTE Private Region is known as the IDT Device Header. By default, when you initialize a device as an IDT-Capable GuardPoint, CTE reserves 63 MB of space starting at the first sector on the device for the CTE Private Region.

CTE writes the IDT Device Header into the CTE Private Region when the device is guarded for the first time. If there is existing data on the device, the data at the start of the device is relocated to the available free space on the device and CTE creates the CTE Private Region starting at the first sector. For details, see ["Initialize a Linux Device with Existing Data" on page 187](#).

## CTE Private Region Location

Normally, CTE requires that the CTE Private Region be embedded at the beginning of the device. IDT, however, allows you to specify that the CTE Private Region for an IDT-Capable GuardPoint should be located in a central CTE metadata directory on the host called `/vte/vte-metadata-dir` (default: `/opt/vte/vte-metadata-dir`). If you use this option, CTE stores the CTE Private Region and IDT Device Header for the device in this directory. The location of the CTE Private Region for a device is determined when you first initialize the device as an IDT-Capable GuardPoint. For details, see ["Initializing an IDT-Capable Device" on page 186](#).





### WARNING

Access to the CTE metadata directory is local to the CTE protected host. Devices whose access is shared across multiple CTE protected hosts in a cluster *must* be configured with the CTE Private Region embedded in those devices. Using a centralized metadata directory for shared devices will lead data corruption.

The location of the CTE Private Region does not affect CTE's functionality, but there are some considerations if you choose to use the centralized metadata directory `/vte/vte-metadata-dir`:

- Thales recommends that you keep the metadata for the device on the device if at all possible. You should only use the centralized metadata directory if the device cannot be expanded to accommodate the CTE Private Region.
- The centralized CTE metadata directory must be guarded by the Administrator to prevent accidental modification or deletion of the CTE metadata. If the CTE metadata directory is not guarded, any attempt to configure or enable an IDT-Capable GuardPoint using the centralized metadata directory will be rejected.

The policy associated with the metadata directory must:

- Deny all users (including the root user) the ability to modify or remove any files in the metadata directory.
- Use the key rule `clear_key` so that the metadata is stored in clear text.
- You must back up this directory whenever you back up a device that uses the directory. You will not be able to restore a protected device without access to its corresponding metadata in `/vte/vte-metadata-dir`.
- Devices with existing data do not need to be resized to accommodate the CTE Private Region, so there are no disk size discrepancies between system utilities such as `fdisk` and any other applications. However, Thales still recommends that you do not shrink an IDT-Capable GuardPoint even if the CTE Private Region is not embedded on the device.

## Device Size

If you embed the CTE Private Region on the device itself, after configuring and guarding the IDT-Capable GuardPoint on the device, the device size reported to applications is the size of the device minus the space reserved for the CTE Private Region. This can lead to a discrepancy between the disk size reported by some applications versus the size reported by system utilities such as `fdisk`.



### WARNING

Do not shrink IDT-Capable GuardPoints. Due to the relocation of user data from the CTE Private Region, if you shrink the device, you may corrupt data on the device.

The IDT Device Header contains both the available device size and the size of the CTE Private Region. To view the IDT Device Header, use the `voradmin idt status <device-name>` command. The **Exported Disk Size** field shows the disk size available for use by other applications. The **Private Region Size** field shows the disk size reserved for CTE. For example:

```
voradmin idt status /dev/sdc2
IDT Header on /dev/secvm/dev/sdc2
 Version: 1
 Change: 0
 Private Region Size: 129024 sectors
 Exported Device Size: 9627648 sectors
 Key UUID: 9cc3c8e4-7ea7-310f-85c7-6f911de1ab52
 Mount Path: None
```

The `voradmin idt status` command also reports the UUID of the XTS-AES 256 key applied to the device.

## Guarding an IDT-Capable Device on Linux

In order to guard an IDT-Capable device, you need to:

1. Make sure the devices you intend to guard meet the requirements for IDT-Capable GuardPoints. For details, see ["Requirements for IDT-Capable GuardPoints" on page 182](#).
2. Install the CTE Agent and register the host with the Key Manager (DSM or CipherTrust Manager) if it is not already registered. IDT does not require any special registration options or licenses. For details, see [Chapter 1: "Installing CTE for " on page 1](#) the *CTE Agent for Linux Quick Start Guide*.
3. Initialize the device using the `voradmin idt config [new|xform]` command to specify whether there is any existing data on this device that needs to be encrypted and to configure the location of the CTE Private Region. For details, see ["Initializing an IDT-Capable Device" below](#).
4. Log on to the Key Manager (DSM or CipherTrust Manager) to apply an IDT-Capable GuardPoint to the device. For details, see ["Guard the Linux Device with an IDT-Capable GuardPoint" on page 189](#).



### WARNING

**For devices with shared access across multiple CTE Protected hosts in a cluster, you must designate one and only one of the nodes in the cluster as the node on which you plan to initialize and guard the device for the first time. The designated node must be the only one that accesses the device until the entire initial data transformation process has completed. This requires guarding each shared device at the designated host level rather than at the host group level if you are using a host group to manage the CTE Protected nodes in your cluster. DO NOT initialize or guard any device on multiple nodes in the cluster simultaneously, because multiple nodes attempting to transform the same data can corrupt the data on the entire device.**

## Initializing an IDT-Capable Device

When you initialize an IDT-Capable storage device, the process specifies:

- Whether there is existing data on the device that needs to be encrypted.
- Where you want to store the CTE Private Region, which contains the IDT Device Header along with metadata that identifies the IDT-Capable device as a guarded device. You can embed the CTE Private Region on the device itself or in the central CTE metadata directory on the host. (For details, see ["The CTE Private Region and IDT Device Header" on page 184](#).)

How you initialize the device depends on whether it is a new device or an existing device that already has data that needs to be transformed into cipher-text. For details, see:

- ["Initialize a New Linux Device" below](#)
- ["Initialize a Linux Device with Existing Data" on the next page](#)

## Initialize a New Linux Device

Run the `voradmin idt config new` command to initialize a new device. The `new` option specifies that the device does not hold user data so no initial data transformation is required. For a shared device that is accessed from multiple protected hosts, you must initialize the device only once and on only one protected host.

### Note

To configure devices with multiple IO paths for Linux, see ["Changing the Encryption Key on Linux IDT-Capable Devices" on page 193](#).

1. Log into the device as `root`.
2. Run the `voradmin idt config [-external] new [-c <n>] <device-name>` command, where:
  - `-external` is an optional parameter that tells CTE you want to use the centralized CTE metadata directory instead of embedding the CTE Private Region on the device itself. If you use this option, you must have configured and guarded the centralized CTE metadata directory as described in ["CTE Private Region Location" on page 184](#).
  - `new` (required) indicates that the device contains no data (it is a new disk). As soon as you push the IDT policy, the device will be available as a guarded IDT-Capable GuardPoint.
  - `-c <n>` (optional). If you use this option, CTE sets the number of data transformation jobs to run in parallel to the number specified in `<n>`. `<n>` can be an integer between 1 and 60 (default: 8).

Each data transformation job transforms 1MB worth of data and requires CPU resources in addition to three I/O operations as part of data transformation. Each job reads 1MB of data from the device, preserves the data in the CTE Private Region, rekeys the data to cipher-text, and writes the transformed data to the device. If you increase the number of parallel jobs, the data transformation process will complete faster but there will be an increased performance impact on the system. Only increase the `-c` option if you are certain that the system resources are available to handle the additional load.

The value for the `-c` option you specify here remains in effect for all subsequent data transformations (such as any data rekeys) until you specify a new value.

- `<device-name>` (required). Specifies the device name. For example, `/dev/sdc2`.

For example, if you want to initialize a new Linux disk named `/dev/sdc2` using 10 parallel data transformation jobs with the CTE Private Region embedded on the device, you would specify:

```
voradmin idt config new -c 10 /dev/sdc2
```

If you want to initialize a new Linux disk named `/dev/sdc2` using the default number of parallel data transformations but with the CTE Private Region in the centralized CTE metadata directory, you would specify:

```
voradmin idt config -external new /dev/sdc2
```

3. To verify that the disk has been initialized, run the `voradmin idt status` command.

```
voradmin idt status /dev/sdc2
```

Device `/dev/sdc2` is configured to guard as IDT-Capable GuardPoint
4. At this point the Administrator can protect the device as an IDT-Capable GuardPoint through the Key Manager (DSM or CipherTrust Manager). For details, see ["Guard the Linux Device with an IDT-Capable GuardPoint" on page 189](#).

### Note

The initialization process prepares the device to be guarded but does not actually guard it. You need to assign an IDT-Capable GuardPoint to the device in the Key Manager (DSM or CipherTrust Manager) before the device is actually protected.

## Initialize a Linux Device with Existing Data

If the device has existing data, you need to use the `voradmin idt config xform` command to initialize the disk for CTE. Unless you are using the centralized CTE metadata directory, this command examines the current disk

size and computes the size required to hold the existing data plus the CTE Private Region at the beginning of the device. After the CTE initialization is complete, you then need to resize the device before you can guard it with an IDT-Capable GuardPoint.



**WARNING**

**If access to the device is shared access across multiple CTE Protected hosts in a cluster, be sure to initialize the device on one and only one of the CTE hosts.**

The following procedure describes how to initialize the device for CTE. Note that the existing data is not altered in any way until after you perform this procedure and you guard the data with an IDT-Capable GuardPoint. CTE does *not* begin transforming the data from clear-text to cipher-text until the IDT-Capable GuardPoint has been applied and the encryption key has been pushed to the device through the GuardPoint Policy.

1. Log into the device as `root`.
2. Run the `voradmin idt config [-external] xform [-c <n>] <device-name>` command, where:
  - `-external` is an optional parameter that tells CTE you want to use the centralized CTE metadata directory instead of embedding the CTE Private Region on the device itself. If you use this option, you will not have to resize the device but you must have configured and guarded the centralized CTE metadata directory as described in "[CTE Private Region Location](#)" on page 184.
  - `xform` (required) indicates that the device contains existing data. CTE will transform all existing data on the device from clear-text to cipher-text as soon as you guard the device. The device will be inaccessible until the transformation is complete, and the device must remain offline during the entire transformation process. No user access will be permitted until all data has been transformed.
  - `-c <n>` (optional). If you use this option on Linux, CTE sets the number of data transformation jobs to run in parallel to the number specified in `<n>`. `<n>` can be an integer between 1 and 60 (default: 8).

Each data transformation job transforms 1MB worth of data and requires CPU resources in addition to three I/O operations as part of data transformation. Each job reads 1MB of data from the device, preserves the data in the CTE Private Region, rekeys the data to cipher-text, and writes the transformed data to the device. If you increase the number of parallel jobs, the data transformation process will complete faster but there will be an increased performance impact on the system. Only increase the `-c` option if you are certain that the system resources are available to handle the additional load.

The value for the `-c` option you specify here remains in effect for all subsequent data transformations (such as any data rekeys) until you specify a new value.

- `<device-name>` (required). Specifies the device name. For example, `/dev/sdc3`.

For example, if you want to initialize an existing Linux disk named `/dev/sdc3` using 10 parallel data transformation jobs with the CTE Private Region embedded on the device, you would specify:

```
voradmin idt config xform -c 10 /dev/sdc3
```

```
Device /dev/sdc3 must be resized to at least 9893888 sectors (4831 MBs) before
guarding as IDT-Capable GuardPoint
```

In this case you must manually resize the Linux disk by at least 9893888 sectors before you can guard it. After you guard the disk, you can expand it again later but you cannot shrink it unless you remove the GuardPoint.

If you want to initialize the same device using the centralized CTE metadata directory, you would specify:

```
voradmin idt config xform -external -c 10 /dev/sdc3
```

Note that you do not get a message about resizing the device because the CTE Private Region will not be embedded on the device.

3. To verify that the disk has been initialized, run the `voradmin idt status` command.

```
voradmin idt status /dev/sdc3
```

```
Device /dev/sdh is configured to guard as an IDT-Capable GuardPoint
```

4. If you are embedding the CTE Private Region on the device, at this point, you need to resize the device using your standard disk management tools before you can guard it. Make sure you increase the device size by at least the amount shown in the `voradmin idt config xform` message.

You cannot assign an IDT-Capable GuardPoint to the device until it has been resized. If you do not resize the device, the GuardPoint assignment will fail.

5. After the device has been resized or the centralized CTE metadata directory has been configured and guarded, the Administrator can protect the device as an IDT-Capable GuardPoint through the Key Manager (DSM or CipherTrust Manager) as described in "[Guard the Linux Device with an IDT-Capable GuardPoint](#)" below.

#### Note

The initialization process prepares the device to be guarded but does not actually guard it. You need to assign an IDT-Capable GuardPoint to the device in the Key Manager (DSM or CipherTrust Manager) before the device is actually protected. In addition, the initialization process is only kept in memory until the device is guarded or rebooted. If the device is rebooted before you guard it, you will need to perform the initialization procedure again.

## Guard the Linux Device with an IDT-Capable GuardPoint

#### Note

For details about how to create a GuardPoint in CM, see, "[Managing GuardPoints](#)", [CTE Administration Guide](#).

After the device has been initialized, you can guard the device as an IDT-Capable GuardPoint from the Key Manager (DSM or CipherTrust Manager). For existing devices, as soon as the GuardPoint configuration has been pushed to the host and the status changes to guarded, CTE begins transforming the data on the disk using the encryption key associated with the GuardPoint Policy.



#### WARNING

**If access to the device is shared access across multiple CTE Protected hosts in a cluster, be sure to guard the device on one and only one of the CTE hosts.**

1. Log on to the DSM.
2. Make sure that you know what Policy you want to associate with the GuardPoint or create a new **In-Place Data Transformation** policy if needed. The policy you use for IDT must use an XTS-AES 256 key as the key rule.
3. Select **Hosts > Hosts** on the menu bar. The *Hosts* window opens.
4. Click the target host in the **Host Name** column. The Edit Host window opens to the General tab for the selected host.
5. Click the **GuardPoints** tab and then click **Guard**. The Guard window opens.
6. In the **Policy** field, select the In-Place Data Transformation policy you identified or created earlier in this procedure. CTE will use the XTS-AES 256 key associated with this policy to encrypt the data on the device.

7. In the **Type** field, select either **Raw or Block Device (Auto Guard)** or **Raw or Block Device (Manual Guard)**.

If you select **Auto Guard**, CTE starts the guard process as soon as the policy is pushed to the host. You enable, disable, guard, and unguard the GuardPoint in the DSM. If you want to have the device automatically guarded and mounted at system start up, add the device to `/etc/fstab`. For details, see ["Auto Mount Options for File System Devices on Linux" on page 198](#).

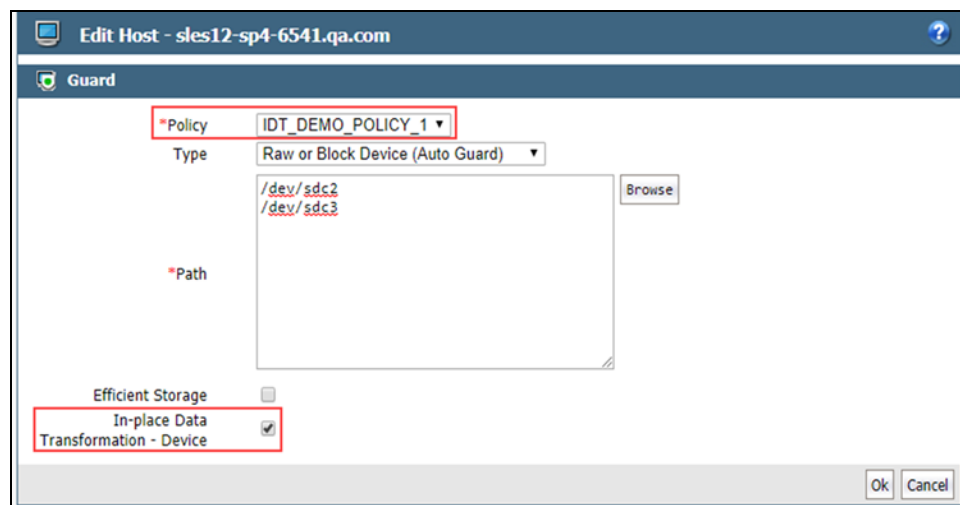
If you select **Manual Guard**, You guard the GuardPoint on the protected host with the `secfsd -guard <path>` command and unguard it with the `secfsd -unguard <path>` command. At system startup, you must guard the device and then mount it. This gives you more control over when data transformations occur because CTE will not start encrypting or rekeying the device until you manually start the process.

8. In the **Path** field, add the path for the device you want to guard. For example, `/dev/sdc2` and `/dev/sdc3`.

If you specify multiple paths in this field, all specified devices will be guarded and all will be encrypted with the encryption key specified in the associated policy.

The devices you specify here must already have been initialized as described in ["Initializing an IDT-Capable Device" on page 186](#). They can be new devices, devices with existing data, or a mix of both.

9. Make sure the **In-Place Data Transformation** check box is checked. If this option is not selected, the host will *not* enable the device as an IDT-Capable GuardPoint.



10. Click **OK**.

The DSM pushes the policy and the IDT-Capable GuardPoint configuration to the host and the CTE Agent on the host writes the IDT Device Header into the CTE Private Region for the specified devices. If this is a new device, the status changes to guarded and the disk is available for user access immediately.

If there is existing data on the device, CTE begins transforming the data from clear-text to cipher-text as soon as the IDT-Capable GuardPoint configuration is available and the device status changes to guarded. The device will remain inaccessible until this data transformation completes. The length of time required to transform the data depends on the amount of existing data and the number of parallel data transformation jobs specified on the `voradmin config` command. For details, see ["Data Relocation and Transformation on Existing Linux Devices" on the next page](#).

To see the data transformation progress, use the `voradmin idt xform status <device-name>` command, as described in ["Viewing Device Status and the IDT Device Header" on page 197](#).

After the device is initialized and guarded, the protected device must be accessed through the CTE device pathname. This pathname corresponds to the `secvm` device. For example, the Linux device pathname `/dev/sdc2` becomes `/dev/secvm/dev/sdc2` as soon as the process is complete.



### Notes

- Be sure to use the `secvm` device name when using file system management tools such as `mkfs` and `fsck`.
- Do not use the device mapper names corresponding to IDT-Capable GuardPoints for GuardPoint administration on protected hosts.

## Data Relocation and Transformation on Existing Linux Devices

When you add an IDT-Capable GuardPoint to a device that has been initialized with the `voradmin idt xform` command and you opted to embed the CTE Private Region on the device, CTE first relocates existing data in the region of the device designated as CTE Private Region. The data is relocated to the end of the device, into the new space allocated when you resized the device. The relocation occurs once when the device is guarded for the first time. No relocation is necessary for subsequent rekeys on the device.

Relocation of data is transparent to applications accessing data through the IDT-Capable GuardPoint. CTE will map application I/O requests over the private region to the relocated region. After guarding the device, you can grow the device size further if necessary. However, you cannot shrink the device size.

IDT does not require a separate policy for data transformation. If you initialized the device with the `xform` option, CTE starts the IDT process when transformation is required. During the IDT process, access to the device is blocked until the IDT process completes and all the data on the device has been encrypted.

```
voradmin idt status xform /dev/sdc3
Status: In-Process
 Relocation Zone 9764864 (relocated = 1)
 SegSpc 27, Xformation Range: 3217 ... 4799, SegIDs: 4795 4796 4791 4792 4797
4798 4799
 KeyID: 2793 Key Name: IDT_DEMO_KEY_1
 Old KeyID: 0 Old Key Name: clear_key

dd if=/dev/secvm/dev/sdc3 of=/dev/null bs=512 count=1
dd: failed to open 'dev/secvm/dev/sdc3': Resource temporarily unavailable

voradmin idt status xform /dev/sdc3
Status: Complete
 Relocation Zone 9764864 (relocated = 1)
 SegSpc 27, Xformation Range: 3217 ... 20189, SegIDs: none
 KeyID: 2793 Key Name: IDT_DEMO_KEY_1
 Old KeyID: 0 Old Key Name: clear_key

dd if=/dev/secvm/dev/sdc3 of=/dev/null bs=512 count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.000989039 s, 518 kB/s
```

### Thin-Provisioned Devices

IDT skips transforming thin-provisioned regions of a device. Data returned to IDT as sequence of clear-text zeros, in sector size granularity, is indication of possible sparse or un-allocated regions of the device that do not have to be transformed.

### IDT Recovery From Crash

IDT is fault tolerant in the event of system crashes. IDT keeps track of the transformation process over the entire device. In the event of a crash, IDT will automatically resume transformation from the point of failure as soon the GuardPoint is enabled after system startup.

If you find the transformation status set to **In-Progress** when the GuardPoint is not enabled, the **In-Progress** state reflects an earlier system crash after which the GuardPoint has not been enabled to recover from the interruption in the IDT process.

## Example of Creating an IDT-Capable GuardPoint on an Existing Linux Device

The following example shows the process of initializing an existing Linux device using `voradmin idt config xform` and guarding it as an IDT-Capable GuardPoint from the viewpoint of the Linux root user. In this example, all files in `/bin/*` are copied to a temporary location outside the device, then compared with the corresponding files on the device after the device has been resized and encrypted. The comparison proves that the file system is unchanged after the encryption process has completed.

First, we verify that the device is not protected, then we check the current size of the disk and create the copy of the files in `/bin/*`. After that, we run the `voradmin idt config xform` command to initialize the device.

```
voradmin idt status /dev/sdc1
Device /dev/sdc1 is not configured as IDT-Capable
fdisk -l /dev/sdc1
Disk /dev/sdc1: 21.1 GiB, 21103640576 bytes, 41218048 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 4194304 bytes
mkfs.xfs /dev/sdc1
meta-data=/dev/sdc1 isize=256 agcount=4, agsize=1288064 blks
 = sectsz=512 attr=2, projid32bit=1
 = crc=0 finobt=0, sparse=0
data = bsize=4096 blocks=5152256, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal log bsize=4096 blocks=2560, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
mount -t xfs /dev/sdc1 /xfs
cp /bin/* xfs
voradmin idt config xform /dev/sdc1
Device /dev/sdc1 must be resized to at least 41347072 sectors (40378 MBs) before
guarding as IDT-Capable GuardPoint
```

At this point, you need to resize the device using your device management tools. You must increase the size by at least 41347072 sectors (40378 MBs). After the device has been resized, you can verify the new size:

```
fdisk -l /dev/sdc1
Disk /dev/sdc1: 21.2 GiB, 21169700864 bytes, 41347072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 4194304 bytes
```

After the device has been resized, the Administrator can guard the device with the desired In-Place Data Transformation policy. If the Administrator chooses Auto Guard, data transformation begins as soon as the policy is pushed to the host. If the Administrator chooses Manual Guard, data transformation does not begin until the Linux root user initiates it with the `secfsd -guard` command. Once data transformation begins, the Linux root user can check the progress using the `voradmin idt status xform` command.

```
secfsd -guard /dev/sdc1
secfsd: Path is guarded
voradmin idt status xform /dev/sdc1
 Status: In-Process
```



```
Relocation Zone 9764864 (relocated = 1)
SegSpc 27, Xformation Range: 3217 ... 4799, SegIDs: 4795 4796 4791 4792 4797
4798 4799
KeyID: 2793 Key Name: IDT_DEMO_KEY_1
Old KeyID: 0 Old Key Name: clear_key
```

After the status has changed to completed, you can compare the current version of the files in `/bin/*` with the ones you copied earlier.

```
voradmin idt status xform /dev/sdc1
Status: Complete
Relocation Zone 9764864 (relocated = 1)
SegSpc 27, Xformation Range: 3217 ... 20189, SegIDs: none
KeyID: 2793 Key Name: IDT_DEMO_KEY_1
Old KeyID: 0 Old Key Name: clear_key

voradmin idt status /dev/sdc1
IDT Header on /dev/secvm/dev/sdc1
Version: 1
Change: 0
Private Region Size: 129024 sectors
Exported Device Size: 41218048 sectors
Key UUID: 9cc3c8e4-7ea7-310f-85c7-6f911de1ab52
Mount Path: None

mount -t xfs /dev/secvm/dev/sdc1 /xfs
for file in '/bin/ls /sfx'; do cmp /bin/$file /xfs/$file; done
unmount /xfs
```

## Changing the Encryption Key on Linux IDT-Capable Devices

To meet various compliance requirements, you may want to change the key that CTE has used to encrypt IDT-Capable GuardPoints. Thales refers to this changing of encryption keys as “Key rotation” or “Rekey”. Unlike the Live Data Transformation product offered by Thales for file systems on traditional storage devices, to change the encryption key on an IDT-Capable GuardPoint, the device must be taken offline. The data on the device will be inaccessible during the key rotation process.

For CipherTrust Manager, see [Key Version Modify Request](#) for more information.



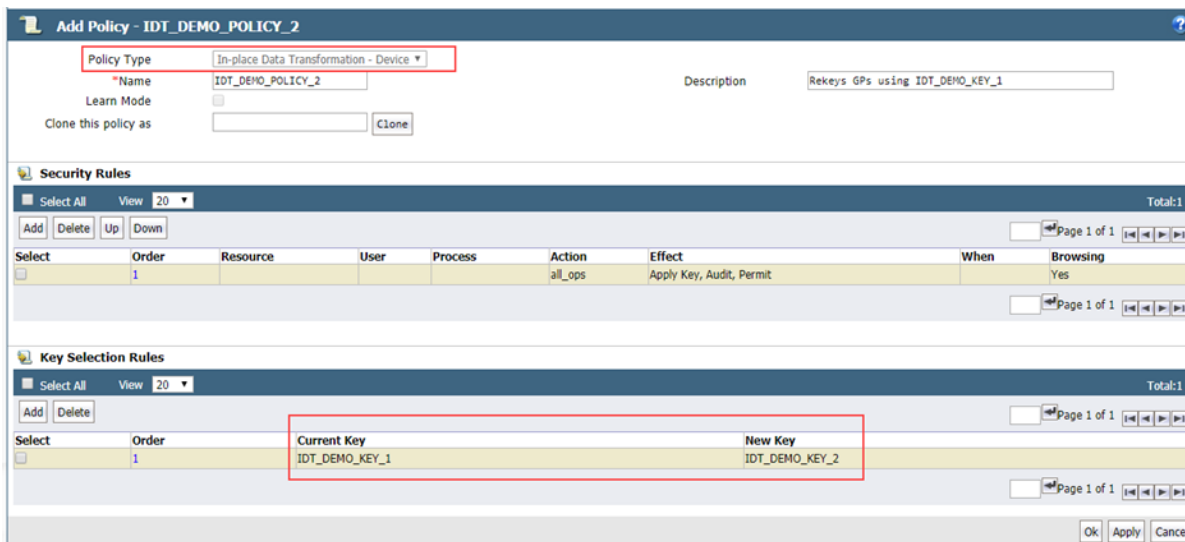
### WARNING

**For devices with shared access across multiple CTE Protected hosts in a cluster, you must designate one and only one of the nodes in the cluster as the node on which you plan to initialize and guard the device for rekey. The designated node must be the only one that accesses the device until the entire rekey process has completed. This requires guarding each shared device at the designated host level rather than at the host group level if you are using a host group to manage the CTE Protected nodes in your cluster. DO NOT initialize or guard any device on multiple nodes in the cluster simultaneously, because multiple nodes attempting to transform the same data can corrupt the data on the entire device.**

1. Log on to the DSM Management Console as an administrator of type Security with Host role permissions, type Domain and Security, or type All.
2. Make sure that you know what Policy you want to associate with the GuardPoint or create a new **In-Place Data Transformation** policy if needed. The key rule must specify the current XTS-AES 256 key that the GuardPoint is currently using as well as the new XTS-AES 256 key that you want to use to transform the protected data.

You can either create a new In-Place Data Transformation policy or you can change the keys assigned to an existing In-Place Data Transformation policy. If you use an existing policy however, the new key you specify cannot have been previously used to encrypt the IDT-Capable GuardPoint. If you want to rekey the GuardPoint using a previously-used key, you must create a new policy in order to do so.

The following screenshot shows a policy named `IDT_DEMO_POLICY_2` for rekey that uses `IDT_DEMO_KEY_1` as the current key and `IDT_DEMO_KEY_2` as the new key:



Including both the current key and the new key in the policy ensures that both keys will be available during the rekey process, even if something happens and the key information in stored in the CTE Private Region becomes unavailable.

Do not push this policy to the host yet.

3. Shut down any applications accessing the GuardPoint you are planning to rekey. If the GuardPoint is mounted as a file system, you must also unmount the file system.
4. Once the data can no longer being accessed, you can unguard the GuardPoint on the DSM.

- a. If the GuardPoint is a manual device GuardPoint, you must first unguard it using the `secfsd -unguard` command on the CTE host before you unguard it on the DSM. If it is an automatic GuardPoint, you can skip this step and simply unguard the GuardPoint in the DSM.

The following example checks the guard status of `/dev/sdc1` and gets the current key name, then unguards the device.

```
secfsd -status guard
GuardPoint Policy Type ConfigState Status Reason

/dev/sdc2 IDT_DEMO_POLICY_1 rawdevice guarded guarded N/A
/dev/sdc3 IDT_DEMO_POLICY_1 rawdevice guarded guarded N/A
/dev/sdc1 IDT_DEMO_POLICY_1 manualrawdevice guarded guarded N/A
voradmin idt status xform /dev/sdc1
Status: Complete
Relocation Zone 0 (relocated = 0)
SegSpc 27, Xformation Range: 4294967295 ... 4294967295, SegIDs: none
KeyID: 2793 Key Name: IDT_DEMO_KEY_1
Old KeyID: 0 Old Key Name: clear_key
secfsd -unguard /dev/sdc1
secfsd: Path is not guarded
secfsd -status guard
GuardPoint Policy Type ConfigState Status Reason

/dev/sdc2 IDT_DEMO_POLICY_1 rawdevice guarded guarded N/A
/dev/sdc3 IDT_DEMO_POLICY_1 rawdevice guarded guarded N/A
/dev/sdc1 IDT_DEMO_POLICY_1 manualrawdevice unguarded not guarded
Inactive
```

- b. Return to the DSM Management Console, select the GuardPoint in the GuardPoints table, and click **Unguard** to unguard the device in the DSM.

Wait until the GuardPoint has been removed from the DSM.

5. Return to the device and run the `voradmin idt rekey` command. After you run the `voradmin` command, the IDT Device Header is temporarily removed from the device.

```
voradmin idt rekey /dev/sdc1
Enter YES to prepare device /dev/sdc3 for rekey -> YES
voradmin idt status /dev/sdc1
Device /dev/sdc1 is configured to guard as IDT-Capable GuardPoint
```

**Note:** For manual GuardPoints, you must unguard the device both using `secfsd -unguard` and the DSM Management Console before you can use the `voradmin idt rekey` command.

6. In the DSM, guard the device with the new **In-Place Data Transformation** policy you created earlier. If you selected a Manual GuardPoint, use `secfsd -guard` to activate the new policy and start the data transformation to the new key.

During the IDT process, user access to the GuardPoint is blocked until IDT completes the transformation process.

The following example shows how to use `secfsd -guard` on manual GuardPoint `/dev/sdc1`, and the status messages that occur during the rekey process. If you are using an automatic GuardPoint, you do not need to use the `secfsd -guard` command. Instead, the rekey process starts as soon as you push the new policy from the DSM.

```
secfsd -guard /dev/sdc1
secfsd: Path is guarded
secfsd -status guard
GuardPoint Policy Type ConfigState Status Reason
```

```

/dev/sdc2 IDT_DEMO_POLICY_1 rawdevice guarded guarded N/A
/dev/sdc3 IDT_DEMO_POLICY_1 rawdevice guarded guarded N/A
/dev/sdc1 IDT_DEMO_POLICY_2 manualrawdevice guarded guarded Data
transformation in progress
voradmin idt status xform /dev/sdc1
 Status: In-Progress
 Relocation Zone 0 (relocated = 0)
 SegSpc 27, Xformation Range: 3987 ... 3993, SegIDs: 3991 3987 3988 3992 3989
3990 3993
 KeyID: 2921 Key Name: IDT_DEMO_KEY_2
 Old KeyID: 2793 Old Key Name: IDT_DEMO_KEY_1

```

7. After the `xform` status shows as completed, you can restart all application workloads on the guarded device.

```

voradmin idt status xform /dev/sdc1
 Status: Complete
 Relocation Zone 0 (relocated = 0)
 SegSpc 27, Xformation Range: 4768 ... 4768, SegIDs: none
 KeyID: 2921 Key Name: IDT_DEMO_KEY_2
 Old KeyID: 2793 Old Key Name: IDT_DEMO_KEY_1
secfsd -status guard
GuardPoint Policy Type ConfigState Status Reason

/dev/sdc2 IDT_DEMO_POLICY_1 rawdevice guarded guarded N/A
/dev/sdc3 IDT_DEMO_POLICY_1 rawdevice guarded guarded N/A
/dev/sdc1 IDT_DEMO_POLICY_2 manualrawdevice guarded guarded N/A

```

## Guarding an IDT-Capable Device with Multiple IO Paths on Linux

Each individual IO path from a server node to a storage controller is treated as a separate device on the host. DM-Multipath on a Linux host provides a management framework to group the individual IO paths to the same LUN into a single multipath device. If you use DM-Multipath to manage devices on the protected host, the individual devices that correspond to each IO path to the LUN cannot be configured for guarding as IDT-Capable, as those devices are under control of DM-Multipath. To guard such devices, you must guard the device mapper generated by DM-Multipath (multipathd) under the `/dev/mapper` directory.

### Note

IDT and Efficient Storage are the only features of CTE that exclusively support guarding of device mapper generated devices under DM-Multipath framework.

The following example illustrates the procedure for guarding a device mapper generated device with the alias name `/dev/mapper/mpathA`.

1. Create a **In-Place Data Transformation** policy using an XTS-AES 256 key as the key rule.
2. On the host, prepare the device to be configured as IDT-Capable using the `voradmin idt config [-external] new|xform [-c n] <mapper-alias-name>` command.

For example, if the disk is a new disk with no existing data, you would enter:

```
voradmin idt config new /dev/mapper/mpathA
```

If the disk has existing data that you want to encrypt, you would enter:

```
voradmin idt config xform /dev/mapper/mpathA
```

3. Guard `/dev/mapper/mpathA` as Device GuardPoint using the policy created above. Be sure to check the **In-Place Data Transformation** check box.

4. For Manual Guard configuration, enable the GuardPoint using the `secfsd` command as follows:  

```
secfsd -guard /dev/mapper/mpathA
```
5. For Auto Guard, wait for the `/dev/mapper/mpathA` device to be guarded on the protected host.
6. Once the device is guarded, provide the pathname of the `secvm` device to applications and/or file system operations. For example, `/dev/secvm/dev/mapper/mpathA`.

## Viewing Device Status and the IDT Device Header

After you guard a device, you can view the status of that device using the `voradmin idt [xform] status <device-name>` command, where:

- `xform` (optional). If you specify this option, CTE shows the status of any data transformation processes happening on the device. If you do not specify this option, CTE displays the IDT Device Header for the device.
- `<device-name>` (required). The standard Linux name of the device whose status you want to view. (For example, `/dev/sdc2`.)

For example, if you want to view the IDT Device Header for the Linux device `/dev/sdc2`, you would enter:

```
voradmin idt status /dev/sdc2
IDT Header on /dev/secvm/dev/sdc2
 Version: 1
 Change: 0
Private Region Size: 129024 sectors
Exported Device Size: 9627648 sectors
 Key UUID: 9cc3c8e4-7ea7-310f-85c7-6f911de1ab52
 Mount Path: None
```

If you want to view the data transformation status on `/dev/sdc2`, you would enter:

```
voradmin idt status xform /dev/sdc3
 Status: In-Process
 Relocation Zone 9764864 (relocated = 1)
 SegSpc 27, Xformation Range: 3217 ... 4799, SegIDs: 4795 4796 4791 4792 4797
4798 4799
 KeyID: 2793 Key Name: IDT_DEMO_KEY_1
 Old KeyID: 0 Old Key Name: clear_key
```

The **Status** field displays **In-Progress** if a data transformation process is running, and **Completed** if the process has finished.

## Linux System and IDT-Capable GuardPoint Administration

### Note

For details about how to create a GuardPoint in CM, see the chapter, [“Managing GuardPoints”, CTE Administration Guide](#).

## voradmin IDT Commands on Linux

The `voradmin` command is a command line utility for management of CTE specific configuration and status reporting. The `voradmin` command also supports configuration management related IDT-Capable GuardPoints (IDT).

For details about the Linux `voradmin idt` command options, see the man page for the `voradmin` command.

## File System Mount Points on Linux

You can create and mount a file system on an IDT-Capable GuardPoint. CTE imposes one restriction on the mount point pathname selected for a device. Once you mount the device on a pathname, you cannot change the mount point to a different pathname. This restriction is enforced to allow the file system mount point to be guarded using a separate policy to enforce access control rules on the mounted file system namespace.

The following example shows the mount point of the IDT-Capable GuardPoint as the `/xfs` directory. The example also shows a failed attempt to mount the file system on a different directory pathname.

```
voradmin idt status /dev/sdc1
IDT Header on /dev/secvm/dev/sdc1
 Version: 1
 Change: 0
 Private Region Size: 129024 sectors
 Exported Device Size: 9627648 sectors
 Key UUID: 9cc3c8e4-7ea7-310f-85c7-6f911de1ab52
Mount Path: /xfs
umount /xfs
mkdir /other-xfs
mount -t xfs /dev/secvm/dev/sdc2 /other-xfs
mount: permission denied
mount -t xfs /dev/secvm/dev/sdc2 /xfs
```

## Auto Mount Options for File System Devices on Linux

IDT-Capable GuardPoints containing file systems can also be added to the `/etc/fstab` configuration file for auto mount at startup or unmount at shutdown. An entry can be for a GuardPoint configured for either Auto Guard or Manual Guard. For more information about Auto and Manual Guard options, see ["Guard the Linux Device with an IDT-Capable GuardPoint" on page 189](#).

Use the device path corresponding to an IDT-Capable GuardPoint device when specifying `fstab` entries, such as `/dev/secvm/dev/sdh`. Do not use the native device pathnames, such as `/dev/sdh`, or device mapper device names. You must also include several settings in the `fstab` entry for each IDT-Capable GuardPoint, as shown in the following table:

| Option                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x-systemd.requires=secvm-barrier.service</code>      | Ensure that the IDT-Capable GuardPoint is enabled before the device is mounted at startup and disabled after the device is unmounted at shutdown. The <code>secvm-barrier.service</code> service is a proxy for all the services that make up CTE.                                                                                                                                                                                                                                                                         |
| <code>nofail</code>                                        | The system boot will proceed without waiting for the IDT-Capable device if it can't be mounted successfully.                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>x-systemd.wanted-by=&lt;idt device&gt;.device</code> | Required for Linux distributions running <code>systemd</code> 242 or later. Instructs <code>systemd</code> to add a <code>Wants=</code> dependency on the IDT-Capable device to ensure that, when the device becomes available, this mount operation is executed. <code>&lt;idt device&gt;.device</code> is the name of the device specified in <code>fstab</code> with the <code>'/'</code> replaced with <code>'.'</code> . For example, <code>/dev/secvm/dev/sdb</code> becomes <code>dev-secvm-dev-sdb.device</code> . |

This is an example of an entry in `/etc/fstab` for an IDT-Capable GuardPoint with an `xfs` file system that is mounted on `/xfs`:

```
/dev/secvm/dev/sdh /xfs xfs x-systemd.requires=secvm-barrier.service, \
x-systemd.wanted-by=dev-secvm-dev-sdh.device,nofail 0 0
```

For information about configuring systemd for CTE, see [Chapter 6: "CTE and systemd" on page 54](#).

## Linux System Utilities for Signing

The following table includes recommendations on the system and file system specific utilities for inclusion in the signature set to allow or deny root execution.

| EXT Utilities | Deny/Allow | XFS           | Deny/Allow | Generic Utilities | Deny/Allow |
|---------------|------------|---------------|------------|-------------------|------------|
| badblock      | Allow      | fsck.xfs      | Allow      | mount             | Allow      |
| debugfs       | Deny       | mkfs.xfs      | Allow      | umount            | Allow      |
| e2freefrag    | Allow      | xfs_repair    | Allow      | dmsetup           | Allow      |
| e2fsck        | Allow      | xfs_admin     | Allow      |                   |            |
| e2image       | Allow      | xfs_bmap      | Allow      |                   |            |
| e2label       | Allow      | xfs_check     | Allow      |                   |            |
| e2undo        | Allow      | xfs_copy      | Deny       |                   |            |
| filefrag      | Allow      | xfs_db        | Deny       |                   |            |
| fsck.ext2     | Allow      | xfs_estimate  | Allow      |                   |            |
| fsck.ext3     | Allow      | xfs_freeze    | Allow      |                   |            |
| fsck.ext4     | Allow      | xfs_fsr       | Allow      |                   |            |
| logsave       | Allow      | xfs_growfs    | Allow      |                   |            |
| mke2fs        | Allow      | xfs_info      | Allow      |                   |            |
| mkfs.ext2     | Allow      | xfs_logprint  | Allow      |                   |            |
| mkfs.ext3     | Allow      | xfs_mdrestore | Allow      |                   |            |
| mkfs.ext4     | Allow      | xfs_metadump  | Allow      |                   |            |
| resize2fs     | Allow      | xfs_mkfile    | Deny       |                   |            |
| tune2fs       | Allow      | xfs_ncheck    | Allow      |                   |            |

## Resizing Guarded IDT Devices

Devices configured as IDT-Capable can be resized using the system-provided resizing utilities. If you are using a file system on the GuardPoint, you can mount the file system after resizing the device and then grow the file system to the new size using the appropriate utility such as `xfs_growfs` or `resize2fs`.



### WARNING

**Do not shrink IDT-Capable GuardPoints. Due to relocation of user data from the CTE Private Region, if you shrink the device, you may corrupt data on the device.**

1. Stop applications from accessing the IDT-Capable GuardPoint.
  - Unmount the file system if the device is mounted.
  - Disable the IDT-Capable GuardPoint if using Auto Guard or on the host with the `secfsd -unguard <device-name>` command if using Manual Guard.
2. Use the native disk management tools to resize the device.
3. After resizing the device, check the size of the device with the `fdisk -l` or similar command. Note that you cannot use the `voradmin idt status` command to verify the new size of the device at this point because the size information is not updated in CTE until the IDT-Capable GuardPoint is re-enabled.
4. If the reported size does not match what you expect, you may need to rescan your storage devices using the command appropriate for the device's connection type.
5. Once the expected size is achieved, enable the IDT-Capable GuardPoint and restart your applications.

This process is identical to resizing an ES GuardPoint. For an example of that process with CTE-Efficient Storage, see ["Example: Resizing a Linux Device" on page 169](#).

## Use Cases Involving IDT-Capable GuardPoints

IDT-Capable GuardPoints support all of the use cases that apply to ES GuardPoints. For details, see ["Use Cases involving Efficient Storage GuardPoints" on page 170](#).

## Best Practices for the Migration of a legacy Raw Device Guardpoint to an In-Place Device Guardpoint

Migrating a raw device to an In-Place Device GuardPoint uses the same methods as converting to an ESG GuardPoint. For more, see ["Best Practices for the Migration of a legacy Raw Device Guardpoint to an Efficient Storage Guardpoint " on page 176](#).

## Alerts and Errors on Linux

This section lists the alerts and errors that may be encountered during system operations.

### Encryption key on device has not been made available

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the encryption key specified in the policy for the GuardPoint has not been made available to the protected host.

**Solution:** Check the host's connectivity with the DSM.

### Specified policy disagrees with metadata set on the Guard Path

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the key specified in the policy for the GuardPoint does not match with the key stored in the IDT Device Header.

**Solution:** Un-guard the device and check the name and UUID of the key in the IDT Device Header using `voradmin idt status <device-name>` and `voradmin idt status xform <device-name>` commands and compare the name and UUID with the key name specified in the policy. Correct the discrepancy and re-guard the device.



## Device has not been configured for IDT-Capable or Efficient Storage

This alert may appear as the reason for a GuardPoint not to have been enabled. The message appears in the output of `secfsd -status guard` command. The status indicates that the device has been properly guarded as an IDT-Capable GuardPoint on the DSM but the device has not been initialized for guarding as IDT-Capable.

**Solution:** The most probable cause of this error is that you did not initialize the device for guarding as IDT-Capable on protected host. It's also possible that the guarded device has already configured for guarding as Ean IDT-Capable GuardPoint. If the device needs to be initialized for guarding as IDT-Capable, see ["Initializing an IDT-Capable Device" on page 186](#).

## Device not resized for guarding as IDT-Capable or Efficient Storage

This message appears in the output of `secfsd -status guard` command and indicates that the IDT-Capable GuardPoint was not successfully enabled. The status indicates that the newly guarded IDT-Capable GuardPoint, which has been initialized with `xform` option of `voradmin`, has not been resized to accommodate storage space for the CTE Private Region.

**Solution:** Unguard the IDT-Capable GuardPoint from the DSM, resize the LUN and verify that the host sees the expanded size, and then guard the IDT-Capable GuardPoint from the DSM.

## Data transformation failed

This message appears in the output of `secfsd -status guard` command and indicates that the IDT-Capable GuardPoint was not successfully guarded. The status indicates that the protected host encountered an error while transforming the data on the device during IDT.

**Solution:** Consult with the system and/or storage admin to check on the health of the LUN in the storage array. You may contact Thales Support for troubleshooting and recovery if there has not been a report of any error on the LUN.

## Data transformation in progress

This message appears in the output of `secfsd -status guard` command and indicates that the IDT-Capable GuardPoint was not successfully enabled. The status indicates that the protected host is transforming the data on the device.

**Solution:** You must wait for data transformation to complete. Check the status of transformation by running `voradmin idt status xform <device name>`. Access to the device is blocked until transformation completes.

## Device *<device-name>* is configured to guard as Efficient Storage GuardPoint

This error message is reported by the `voradmin` command when initializing a device as an IDT-Capable GuardPoint that has already been initialized for guarding as an Efficient Storage GuardPoint.

**Solution:** The device has already been initialized for guarding as Efficient Storage and is probably waiting to be guarded as an ES GuardPoint. If you want to change the device to an IDT-Capable GuardPoint, you can remove the ES configuration by running `voradmin esg delete <device-name>` and then re-initializing the device as an IDT-Capable GuardPoint as described in ["Initializing an IDT-Capable Device" on page 186](#).

## Device *<device-name>* is configured as Efficient Storage GuardPoint

This error message is reported by the `voradmin` command when initializing a device as IDT-Capable that is already being guarded as an Efficient Storage GuardPoint.

**Solution:** The device is already guarded by an Efficient Storage GuardPoint. If you want to change it to an IDT-Capable GuardPoint, you would need to decrypt the data, delete the Efficient Storage GuardPoint, then reinitialize the device as an IDT-Capable GuardPoint.

### Device <device-name> is configured to guard as IDT-Capable GuardPoint

This error message is reported by the `voradmin` command when initializing a device as an IDT-Capable GuardPoint that has already been initialized for guarding as an IDT-Capable GuardPoint.

**Solution:** The device has already been initialized for guarding as IDT-Capable and is probably waiting to be guarded as an IDT-Capable GuardPoint through the DSM. Alternatively, if you want to remove the IDT-Capable configuration, use the `voradmin idt delete <device-name>` command.

### Device <device-name> is configured as IDT-Capable GuardPoint

This error message is reported by the `voradmin` command when initializing a device that is already being guarded as an IDT-Capable GuardPoint.

**Solution:** The device is already guarded by an IDT-Capable GuardPoint.

### GuardPoint for device <device-name> still guarded on DSM

This error message is reported by the `voradmin` command when attempting to initialize a device for rekey or removing the device as IDT-Capable.

**Solution:** Unguard the IDT-Capable GuardPoint from the DSM, wait for the protected host to process the update, and then rerun the `voradmin` command.

### Failed to open device <device-name>, error Device or resource busy

This message occurs when `voradmin` detects that the target device is busy.

**Solution:** The device may already be in use by other application. Rerun the `voradmin` command when the device is no longer in use.

### Device <device-name> is not configured as IDT-Capable

This error message is reported by the `voradmin` command when attempting to delete a device as an IDT-Capable GuardPoint.

**Solution:** This message indicates that the target device has been not initialized for guarding as IDT-Capable. The message may also be reported if the specified device has been initialized for guarding as IDT-Capable but it has not been guarded yet. In this case, it removes the preparation made by `voradmin`. You can remove the IDT-Capable configuration status on the device by running `voradmin idt delete <device-name>`. You may see the same error message again, and if you do, you can ignore it.

### Abort! Error: Could not stop secfs, secvm device(s) busy

This error occurs during CTE shutdown when there is a busy CTE protected device.

**Solution:** Verify that all applications directly accessing `secvm` protected GuardPoints have been shut down. Ensure that all file systems on top of a `secvm` protected devices are under the control of `systemd` and have been unmounted before attempting to shut down the agent.

## Abort! Error: Could not unmount file systems

This error occurs during CTE shutdown when a file system under the control of `systemd` fails to unmount.

**Solution:** Verify that file systems on top of an IDT-Capable GuardPoint devices are not busy and then rerun the agent shut down command.

## A dependency job for `idt.mount` failed. See '`journalctl -xe`' for details

This error occurs during CTE startup when system fails to mount a file system. This error message is typically accompanied by a long timeout during the CTE startup process.

**Solution:** Check that the underlying device is available and that the IDT-Capable GuardPoint was successfully applied on the device. Once the device is available, the file system will automatically finish mounting.

## ESG/IDT-ALERT: IO error on header for [GuardPoint]

This is an alert message to the DSM. It occurs when CTE encounters a general error when attempting to access the private region on an IDT-Capable device.

**Solution:** An I/O error attempting to read or write to the device may have been caused by errors on the host or storage array. Consult with the system and/or storage admin to check on the health of the LUN in the storage array. You may contact Thales Support for troubleshooting and recovery if there has been no report of errors on the LUN.

## ESG/IDT-ALERT: Data transformation failure on [GuardPoint]

This is an alert message to the DSM. It occurs when protected host encounters an error transforming the data on device during IDT process.

**Solution:** Contact Thales Support for troubleshooting and recovery.

## ESG/IDT-INFO: Data transformation complete on [GuardPoint]

This message is a notification to the DSM admin that the protected host has completed the data transformation of the specified IDT-Capable GuardPoint.

## ESG/IDT-ALERT: Failed to resize *<device-name>*

This alert is a notification to the DSM admin that the protected host has failed to update the change to the device size in the IDT Device Header on the device.

**Solution:** Contact Thales Support for troubleshooting and recovery.

## FSADM-ALERT: ESG/IDT required Signature Set for system utilities may have to be resigned

This alert is a notification to the DSM admin that the recent system upgrade to the protected host may have updated the binary files listed in the signature set for restricting root access.

**Solution:** Upon this notification you must immediately re-sign the affected or all the binary files to prevent them from accessing protected data. See "[Linux System Utilities for Signing](#)" on page 199.

## File System is not automatically mounted after IDT completes

An IDT-Capable device with a file system that has been configured to automatically mount may fail to automatically mount while the device undergoes data transformation through IDT. Following is a sample log message in the kernel ring buffer that reports a failed attempt to access a device during IDT. You can ignore this message.

```
Vormetric SecVM: secvm_map during IDT ((dc 00000000c3537611))
```

When access to a device is to mount the file system, the kernel ring buffer may also report a second error message indicating that a file system failed to mount due to data corruption. You can ignore this message. This occurs because the mount command is issued after the device has been guarded but before data transformation through IDT is complete. While IDT is in progress, the device cannot be used, so any attempt to mount the file system will fail.

**Solution:** Manually mount the file system after IDT is complete.

# Chapter 21: CTE with Teradata Database Appliances

---

CipherTrust Transparent Encryption (CTE) offers IDT-Capable GuardPoints on Linux for protecting data on raw devices. The IDT solution offers data encryption, data transformation, and access control on storage devices. The principles behind CTE's IDT-Capable GuardPoints can also be applied to protect Teradata Database Appliances. (For details about IDT, see [Chapter 20: "In-Place Data Transformation for Linux" on page 182.](#))

This section contains the following topics:

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| <a href="#">IDT-Capable GuardPoints and Teradata Database Appliances</a>     | 205 |
| <a href="#">Requirements and Considerations</a>                              | 205 |
| <a href="#">Guarding a Teradata Database Device</a>                          | 207 |
| <a href="#">Changing the Encryption Key on Teradata Devices</a>              | 215 |
| <a href="#">Access Rules to Apply on the Teradata Database Appliance</a>     | 216 |
| <a href="#">Replication of IDT Metadata Files Across Members of a Clique</a> | 217 |
| <a href="#">Generate IDT-Capable metadata for Teradata Storage Expansion</a> | 220 |
| <a href="#">Best Practices</a>                                               | 221 |
| <a href="#">Uninstalling CTE from the Teradata Cluster</a>                   | 222 |
| <a href="#">Alerts and Errors</a>                                            | 223 |

## IDT-Capable GuardPoints and Teradata Database Appliances

The Teradata Database Appliance must be protected by an IDT-Capable GuardPoint. The In-Place Data Transformation (CTE-IDT) feature offers capabilities such as initial data transformation and access control on protected raw devices. The transformation capability transforms existing clear-text data on a device to cipher-text, and allows for subsequent re-transformation using another encryption key.

You must be familiar with CTE and CipherTrust Manager support for IDT-Capable GuardPoints before you can configure CTE to work with Teradata. For details about CTE-IDT, see [Chapter 20: "In-Place Data Transformation for Linux" on page 182.](#)

## Requirements and Considerations

### Location of the CTE Private Region

The CTE Private Region contains the metadata CTE requires in order to support initial transformation of data on the device and subsequent data transformation to other encryption keys. By default, CTE creates the CTE Private Region at the beginning of the guarded device. If data already exists on the device, CTE requires that the device be expanded by 63 MB to make room for the CTE Private Region. The existing data in the first 63MB of the device is then migrated into the expanded space and the beginning of the device is reserved for the CTE metadata. This data relocation is completely transparent to applications and users.

With a Teradata Database Appliance, however, CTE cannot create the CTE Private Region at the beginning of the Teradata pdisk devices because the disks in the Appliance cannot be expanded. Therefore, for Teradata Databases, CTE stores the metadata in a special directory called the CTE Metadata Directory located at: `/var/opt/teradata/vormetric/vte-metadata-dir`. This directory contains all of the metadata for every Teradata Database device that is protected by CTE.

While this does not affect the functionality of CTE, it does affect the way administrators need to back up the Teradata Database because both the Teradata Database and the metadata directory must be backed up together. You will not be able to restore a Teradata Database without access to the associated metadata in the CTE metadata directory.

## Metadata File Access and Teradata Clusters

A Teradata *cluster* can contain multiple hosts. The members of a cluster that share access to pdisk devices belong to a *clique*. When you create an IDT-Capable GuardPoint on a pdisk, the metadata for that GuardPoint must be available to all members of the clique. CTE automatically replicates the metadata files across the members of a clique when the metadata is created or changed. For details, see "[Replication of IDT Metadata Files Across Members of a Clique](#)" on page 217.

## Additional Requirements and Considerations

- The Teradata kernel must be at minimum version 4.4.140-96.54.TDC-default on every node of the Teradata Database Appliance on which you plan to install CTE. Refer to the release notes document for CTE releases for compatibility requirements between the kernel releases from Teradata, and the CTE releases.



### CAUTION

**Be sure the version of the Teradata Database is fully compatible with CTE.**

**The Parallel Upgrade Tool (PUT) component of Teradata Database has been enhanced to discover CTE protected devices. The Parallel Upgrade Tool (PUT) component of Teradata (TDput) must be version `TDput-03.09.06.09` or higher. This PUT component must be available in your Teradata Database.**

**The Parallel Database Extensions (PDE) component of Teradata (ppde) must be version `ppde-16.20.53.07` or higher. This PDE component must be available in your Teradata Database.**

**The Teradata I/O Scheduler (tdsched) component must be version `01.04.02.02-1` or higher. This I/O Scheduler component must be available in your Teradata Database.**

**Contact your Teradata Customer Support Representative if you are unsure of the availability of this functionality in your Teradata cluster.**

- The CTE Agent must be installed in `/opt/teradata/vormetric` on every node in the Teradata cluster. In order to specify this location, use the `-d` option when installing CTE. For example:

```
./vee-fs-7.2.0-128-sles12-x86_64.bin -d /opt/teradata/vormetric
```

**Note:** The CTE Agent can be installed without stopping the Teradata Database service.

- The CTE metadata directory `/var/opt/teradata/vormetric/vte-metadata-dir` must be guarded by the DSM/CM Administrator to prevent accidental modification or deletion of the CTE metadata files. If the CTE metadata directory is not guarded, any attempt to configure or enable an IDT-Capable GuardPoint on the Teradata appliance will be rejected.

The standard CTE policy associated with the metadata directory must:

- Deny all users (including the root user) the ability to modify or remove any files in the metadata directory.
- Specify the key `clear_key` in the key rule so that the metadata is stored in clear text.
- The following table displays how to set up your policy for guarding the CTE metadata directory:

| Security Rules | Effect        |
|----------------|---------------|
| Action         |               |
| read           | Permit, Audit |
| all_ops        | Deny, Audit   |

| Key Selection Rules |
|---------------------|
| Key                 |
| clear_key           |

- Stop the Teradata Database service when upgrading an existing CTE Agent installation, type:  

```
tpareset -x Stopping Database
```
- Stop the Teradata Database service so that CTE can rekey any guarded devices. The service must remain stopped until CTE has finished rekeying *all* of the guarded devices on the appliance.

## Guarding a Teradata Database Device

To guard a Teradata Database device:

1. Install CTE on every node in your Teradata Database Appliance. During this procedure you will register each 'protected host' (DSM) / 'protected client' (CM) with the DSM. For details, see ["Install CTE on the Teradata Database Appliance" below](#).
2. Identify the devices that you want to guard as IDT-Capable GuardPoints. For details, see ["Identify the Devices to Be Guarded" on the next page](#).
3. Select the initial configuration method that you want to use. For details, see ["Select the Initial Configuration Method" on page 209](#).
4. Initialize and guard the devices using the selected initialization method. For details, see ["Initialize and Guard the Database Devices Using the Standard Initialization Method" on page 209](#) or ["Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method" on page 212](#).

### Note

You must use the **In-Place Data Transformation** policy type when guarding those devices.

## Install CTE on the Teradata Database Appliance

CTE must be installed on every node in the Teradata cluster, including any Hot Standby Nodes. When you install CTE on every node in your Teradata cluster and register each node with the CipherTrust Manager, the following are required:

- You must have AES-NI key available on the host.
- The Teradata kernel must be at version 4.4.140-96.54.TDC-default, or higher, on the Teradata Database Appliance on which you plan to install CTE. Refer to CTE release notes for compatibility between CTE and the Teradata kernel releases.
- Your version of Teradata Database must be fully compatible with CTE for supporting the pdisk devices protected as IDT-Capable GuardPoints.

- The CTE Agent must be installed in `/opt/teradata/vormetric`. In order to specify this location, use the `-d` option when installing CTE. For example:

```
./vee-fs-7.2.0-128-sles12-x86_64.bin -d /opt/teradata/vormetric
```



#### WARNING

Before you install or upgrade CTE, you must stop the Teradata Database and any other applications that access the database devices directly or through the database service. Failure to stop the applications will result in a failure to install or upgrade CTE.



#### CAUTION

Be sure to install the CTE Agent in a file system with sufficient free storage space. The minimum available free space is the number of pdisks in your cluster multiplied by 63MB.

For details about installing the CTE Agent and registering it with the DSM, see ["Configuring CTE for with a DSM" on page 1](#).

## Identify the Devices to Be Guarded

Storage devices in Teradata clusters are known as pdisks. A pdisk is a slice of a LUN exported from the backend storage system attached to the nodes in a cluster. For example, the pdisk named `/dev/pdisk/dsk304` is a symbolic link to the slice on the LUN called `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`, as shown below:

```
ls -l /dev/pdisk/dsk304
lrwxrwxrwx 1 root root 70 May 18 12:21 /dev/pdisk/dsk304 -> /dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

CTE guards the devices represented by the pdisks in the Teradata cluster. In the example above, the device that you must configure and guard as an IDT-Capable GuardPoint is `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`.

#### Note

You cannot use the symbolic link name when initializing or guarding the storage devices.

To identify the disks available to the Teradata cluster and require CTE protection, look in `/dev/pdisks` on each node in the cluster. Any symbolic links that point to a Linux device path are the disks that should be initialized.



#### CAUTION

Do not enter the path name of the pdisk device when initializing it for guarding or in the CipherTrust Manager when you actually guard the device.



## Select the Initial Configuration Method

Thales supports two methods for the initial configuration of Teradata Database devices:

- **Standard Initialization Method**

This method configures devices for initial encryption of existing clear-text data on the database devices. The time required to encrypt may take hours or even days, depending on the volume of data, the number of database devices and nodes, and the bandwidth of the storage back-end of database devices. For details, see ["Initialize and Guard the Database Devices Using the Standard Initialization Method" below](#).

- **Backup/Restore Initialization Method**

This method skips the initial encryption step but requires a full backup of the database and the engagement of Teradata Customer Support to assist you with some configuration steps at the Teradata database level in preparation for the full restore of the database after the database devices are protected. The length of time required for this method depends on the length of time it takes to back up, and then fully restore, your Teradata database. For details about using this method, see ["Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method" on page 212](#).

## Initialize and Guard the Database Devices Using the Standard Initialization Method

The following procedure describes how to perform the initial configuration of your database devices using the Standard Initialization Method. This procedure encrypts the existing data in place and does not require you to backup your Teradata database when initially deploying CTE. It may, however, take several hours, or even days, to complete depending on the volume of data, the number of database devices and nodes, and the bandwidth of the storage back-end of database devices.



### WARNING

**For each device, you must designate *one and only one* of the nodes in the cluster as the initial node. This is the node on which you plan to initialize and guard the device for the first time. The designated node must be the only one that guards the device until the entire initial data transformation process has completed. DO NOT initialize or guard any device on multiple nodes in the cluster simultaneously, because multiple nodes attempting to transform the same data can corrupt the data on the entire device.**

1. Shut down the Teradata Database. You cannot initialize an online database device.
2. For each device, designate one of the nodes in the cluster as the initial node that you will use for the initial data transformation when the device is guarded for the first time.
3. Log into the designated node in the Teradata Database Appliance. Type:

```
voradmin idt config [-external] [new|xform] [-c <n>] <device-name>
```

- **[new|xform]** (required)

Indicates whether data already exists on the device. If the device contains no data, specify `new`. If the device contains data that you want to keep, specify `xform`. Most installations of Teradata Appliance are expected to have pdisks populated with data, therefore, most often you will use the `xform` option. When you use `xform`, CTE will transform all existing data on the device from clear-text to cipher-text as soon as you guard the device on the Appliance. The device will be inaccessible until the transformation is complete, and the device must remain offline to the Teradata Database service during the entire transformation process. No user access will be permitted until all of the data has been transformed.

- **-external** (required)

You *must* use this option when initializing any Teradata device. With this option, CTE writes the CTE Private Region to a metadata file located in the CTE metadata directory. For details, see "[Location of the CTE Private Region](#)" on page 205.

- **-c <n>** (optional)

If you use this option, CTE sets the number of data transformation jobs to run in parallel to the number specified in <n>. <n> can be an integer between 1 and 60, (default: 8).

Each data transformation job transforms 1MB worth of data and requires CPU resources in addition to three I/O operations as part of data transformation. Each job reads 1MB of data from the device, preserves the data in the CTE private region, rekeys the data to cipher-text, and writes the transformed data to the device. If you increase the number of parallel jobs, the data transformation process will complete faster but there will be an increased performance impact on the system. Only increase the **-c** option if you are certain that the system resources are available to handle the additional load.

The value for the **-c** option you specify here remains in effect for all subsequent data transformations (such as any data rekeys) until you specify a new value.

- **<device-name>** (required)

The name of the Teradata Database device that you want to initialize.

**For example:**

```
voradmin idt config -external xform -c 20 \
/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

4. Repeat the `voradmin idt config` command for each device that you want to initialize. If you want to distribute the initial data transformation or subsequent rekey load on all of the disks across all the nodes in the cluster, make sure that you run the `voradmin idt config` command for each device on the node you designated for data transformation, *excluding* the HSN node. The node on which you run the `voradmin idt config` command is the node that performs the data transformation on the device. Do *not* designate the HSN node for data transformation because the devices on the HSN node may be reserved and therefore not available for IO operations.
5. After you initialize the devices, you need to use the DSM Management Console to designate those devices as IDT-Capable GuardPoints and assign the In-Place Data Transformation policy (or policies) you want to use on those devices as described in "[Guard the Devices as IDT-Capable GuardPoints on CM](#)" below.

## Guard the Devices as IDT-Capable GuardPoints on CM

For Guarding Devices on CipherTrust Manager, go to [Creating IDT GuardPoints](#) on [Thalesdocs.com](#).

## Guard the Devices as IDT-Capable GuardPoints on DSM



### WARNING

For each device, you must designate a node in the cluster as the initial node on which you plan to initialize and guard the device for the first time. The designated node must be the only one that accesses the device until the entire initial data transformation process has completed. This requires guarding each device at the designated host level rather than at the host group level. **DO NOT** initialize or guard any device on multiple nodes in the cluster simultaneously, because multiple nodes attempting to transform the same data can corrupt the data on the entire device.

1. Log on to the Management Console as a Security administrator with Host role permissions, type Domain and Security, or type All.
2. Select **Hosts > Hosts** on the menu bar. The *Hosts* window opens.
3. Click the target host in the **Host Name** column. The Edit Host window opens to the General tab for the selected host.
4. In the **Policy** field, select a In-Place Data Transformation policy from the list of available policies. The policy must meet the requirements described in ["Policy Requirements for IDT-Capable GuardPoints" on page 183](#).

**Note:** For information about how to create a GuardPoint, see the chapter, "Managing GuardPoints", in the *DSM Administration Guide*.

5. In the *Type* field, select **Raw or Block Device (Auto Guard)**.
6. In the *Path* field, add the Linux device name for the device you want to guard. For example, `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`. Do *not* enter the symbolic `pdisk` device name for a Teradata disk.
7. Make sure the In-place Data Transformation - Device check box is selected.
8. Click **OK**.

The DSM pushes the policy and the GuardPoint configuration to the node in the cluster and the CTE Agent on the node writes the IDT Device Header into the CTE Private Region in the local CTE metadata directory on the node.

If this is a new device, the status changes to guarded immediately. If there is existing data on the device, CTE begins transforming the data from clear-text to cipher-text as soon as the GuardPoint configuration is available and the device status changes to guarded. The device will remain inaccessible until this data transformation completes. The length of time required to transform the data depends on the amount of existing data and the number of parallel data transformation jobs specified on the `voradmin config` command. To see the data transformation progress, use the `voradmin idt xform status <device-name>` command, as described in ["Viewing Device and Data Transformation Status" on the next page](#).

Devices with existing data are transformed from clear-text to cipher-text using the encryption key specified in the selected policy through the In-Place Data Transformation (CTE-IDT) process. For details on how CTE does data transformation on IDT-Capable GuardPoints, see [Chapter 20: "In-Place Data Transformation for Linux" on page 182](#).

After the device is initialized and guarded, the protected device must be accessed through the device pathname corresponding to the `secvm` device. For example, if you guard the Linux device `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`, the pathname becomes `/dev/secvm/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3` as soon as the process is complete.

9. After all the Teradata Database devices have been guarded, disable the guarded IDT-Capable GuardPoints on each designated node and then enable those GuardPoints at the Host Group level on all the nodes in the clique.

- After guarding your devices and before starting your database, you must change the current configuration of your cluster to reflect the status of the pdisk devices guarded as IDT-Capable GuardPoints. See the **Rebuild Vconfig Only** option of the Teradata Parallel Upgrade Tool (PUT) service to apply the guarded status of the pdisk devices into your database configuration. The **Rebuild Vconfig** process applies the guarded configuration status of each pdisk and resets the pdisk symbolic link to the CTE `sesvm` device.



#### WARNING

You must complete this step before starting your database. Failure to do so will result in database failures and potential corruption of your database.

For example, the pdisk `/dev/pdisk/dsk304` is linked to the `sesvm` device after **Rebuild Vconfig Only** commits the guarded status of pdisk on each node:

```
ls -l /dev/pdisk/dsk304 \
lrwxrwxrwx 1 root root 70 May 18 12:21 /dev/pdisk/dsk304 -> /dev/disk/by-id/tdmp-
360080e500043092c0000b46f5c34c018-part
```

- After all the Teradata Database devices have been guarded in the clique and the **Rebuild Vconfig Only** on TDput has been executed, start the Teradata Database, type:

```
/etc/init.d/tpa start
```

## Viewing Device and Data Transformation Status

After you guard a Teradata Database device, you can view the status of that device using the `voradmin idt status [xform] <device-name>` command, where:

- `xform` (optional)

If you specify this option, CTE shows the status of any data transformation processes happening on the device. If you do not specify this option, CTE displays the IDT Device Header for the device.

- `<device-name>` (required)

The name of the Teradata Database device that you want to initialize. This must be the actual device name and not the symbolic pdisk name.

For example, if you want to view the status of device `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`, you would enter:

```
voradmin idt status /dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

If you want to check the data transformation progress on the device `/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3`, you would enter:

```
voradmin idt status xform \
/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

The **Status** field displays **In-Progress** if a data transformation process is running, and **Completed** if the process has finished.

## Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method

The Standard Initialization Method encrypts the Teradata database devices using In-place data transformation. Because Teradata Appliance models range in both storage and node sizes within a cluster, the time it takes to encrypt the existing data in-place on these large scale models may exceed the desired time frame.

To address this issue on an existing Teradata database, you can configure CTE on Teradata Appliances using the Backup and Restore method to retain existing data. The total time required to configure your Teradata Appliances with CTE using this method depends on how long your backup/restore application takes to restore the database.

To use this method, you must:

1. Backup the entire Teradata database, including all meta files, required for a full restore of that database.
2. Install and configure CTE on the Teradata database devices.
3. Work with Teradata Customer Support to initialize a `sysinit` on the Teradata Appliance cluster in preparation for a full database restore.
4. Perform a full restore of the Teradata database from the backup to the CTE protected devices. CTE automatically encrypts the data as it is restored to each protected device.

## Prerequisites

Before starting this process, complete the following prerequisites:

- Make sure you have met the requirements described in ["Requirements and Considerations" on page 205](#).
- Create or identify the XTS-AES 256 key that you will use for the In-Place Data Transformation policy that you will apply to the IDT-Capable GuardPoints.
- Create a **Standard** policy that will be used to guard the CTE Metadata Directory (`/var/opt/teradata/vormetric/vte-metadata-dir`).
- Create an **In-Place Data Transformation** policy that will be used to guard the Teradata Database Devices. This policy must use an XTS-AES 256 key.
- Identify all of the devices that need to be guarded as described in ["Identify the Devices to Be Guarded" on page 208](#).

## Procedure

1. Using a Teradata certified backup application, backup your *entire* Teradata database. The backup needs to include the Data Dictionary, the user database, and everything else that is required for a successful restore of your database. Contact your Teradata Customer Support representative for the requirements for a full backup.
2. After confirming that your backup of the Teradata database completed successfully, shutdown the Teradata database, type:  

```
tpareset -x -f "shutting down for CTE installation"
```
3. Install CTE in `/opt/teradata/vormetric` on all nodes in the Teradata cluster and register them to a CipherTrust Manager server as described in ["Install CTE on the Teradata Database Appliance" on page 207](#).  

```
./vee-fs-7.2.0-128-sles12-x86_64.bin -d /opt/teradata/vormetric
```
4. Create the CTE metadata directory (`/var/opt/teradata/vormetric/vte-metadata-dir`) on all nodes in the cluster.  

```
. pcl -shell "mkdir -p /var/opt/teradata/vormetric/vte-metadata-dir"
```
5. If you are using the Teradata Intelibase model, complete the following step. If you are using the Teradata Inteliflex model, proceed to Step 6.
  - a. Log on to the CipherTrust Manager, click the **Hosts/Clients** tab, and set the GuardPoint to the CTE Metadata Directory (`/var/opt/teradata/vormetric/vte-metadata-dir`) using the Standard policy that was created for the metadata directory on all the nodes in the Teradata cluster.

- b. On each node, identify the list of disks that will be configured and guarded as "new" devices as described in ["Identify the Devices to Be Guarded" on page 208](#).
- c. On each node, configure each disk to be guarded as a new IDT-Capable device using the `voradmin idt config -external new` command.

```
voradmin idt config -external new \
/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

**Note:** Make sure that you configure each device using the `new` option. This option tells CTE to guard the device without transforming the data.

- d. Log on the CipherTrust Manager, click the **Hosts/Clients** tab, and select one of the host in the clique to set the GuardPoints for all the devices that have been initialized as new IDT-Capable devices using the In-Place Data Transformation policy that you created.

Repeat this step for all other hosts until all initialized devices have been guarded with an In-Place Data Transformation policy.

6. If you are using the Teradata Inteliflex model, complete the following step. Otherwise, proceed to Step 7.

- a. Designate one of the nodes in a clique as the node on which you will perform the first-time initialization and guarding procedures for each device. For example, if you have a 4 clique cluster, you must designate a total of 4 nodes, one from each clique.

- b. Log on to the CipherTrust Manager, click the **Host/Client Groups** tab, and create a host group for each clique. For example, if your Inteliflex cluster has 4 cliques, you would create 4 host groups, one for each clique. For example, cluster-clique-1, cluster-clique-2, cluster-clique-3, and cluster-clique-4.

- c. On the **Host/Client Groups** tab, click the name of one of the client groups you just created. Click on the **Members** tab and add the designated node from the clique as a member of this client group.

Repeat this step until all designated nodes have been added to the client group that matches their clique. In the previous example, you would have one designated node in each of the 4 client groups that you created.

- d. On the **Host/Client Groups** tab, click the name of one of the host groups that you just created. Click on the **GuardPoints** tab and add a GuardPoint for the CTE Metadata Directory (`/var/opt/teradata/vormetric/vte-metadata-dir`) using the Standard policy that was created for the metadata directory on all the nodes in the Teradata cluster.

Repeat this step for each Host/Client Group you created for your cliques. In the previous example, you would add the metadata directory GuardPoint in each of the 4 host groups that you created.

- e. On the designated node for each clique, identify the list of disks that will be configured and guarded as a "new" IDT device as described in ["Identify the Devices to Be Guarded" on page 208](#).

- f. Configure each disk device on the designated node to be guarded as a new IDT-Capable device, type:

```
voradmin idt config -external new \
/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

**Note:** Make sure that you configure each device using the `new` option. This option tells CTE to guard the device without transforming the data.

- g. For each of the other designated nodes you identified, identify the list of devices to be guarded and configure them using the `voradmin idt config -external new` command.

- h. Repeat this configuration process until all devices on all designated nodes that you want to guard have been configured as new IDT-Capable devices.

- i. On the **Host/Client Groups** tab, click the name of one of the clientgroups you just created. Click on the **GuardPoints** tab and add a GuardPoint for each device that you initialized as a new IDT-Capable device using an In-Place Data Transformation policy.



- j. Repeat this step on each one of the host groups you created.

The guarded devices will immediately be set as guarded on the designated node through the client group. However, as part of Teradata cluster support, each guarded device will have its metadata file replicated across all nodes in a clique as described in ["Replication of IDT Metadata Files Across Members of a Clique" on page 217](#). You must wait for metadata replication to complete for each clique in a cluster before continuing to the next step. This process takes approximately 1 second per disk that was guarded. So if you guarded 10 disks, you need to wait approximately 10 seconds. Verify that the number of metadata files on each clique matches with the designated node using the following command:

```
pcl -shell "ls /var/opt/teradata/vormetric/vte-metadata-dir | wc -l"
```

The number of metadata files on each node in the clique should also match the number of devices in that clique. For example, if you have a clique with 10 devices in it, then the results of the above `pcl` command should show 10 metadata files on each node in the clique.

- k. After metadata replication is completed on all cliques, log on the DSM and click the **Hosts/Client Group** tab, then click the name of one of the client groups you just created. Click the **Members** tab and add the rest of the nodes that belong to that clique to the Host Group. After all the nodes have been added, the DSM automatically pushes the existing GuardPoints out to the newly added members.
  - l. Repeat this step for each one of the Host Groups that you created for your cliques.
7. Verify that all of the nodes in a clique show that all of the devices as guarded.  

```
pcl -shell "secfsd -status guard"
```
  8. Work with Teradata Customer Support to perform a `sysinit` of the Teradata appliances in preparation for the full database restore.
  9. Restore your Teradata database from backup.

## Changing the Encryption Key on Teradata Devices

To meet various compliance requirements, you may want to change the key that CTE used to encrypt the Teradata Database devices. This process is called “Key rotation” or “Rekey”.

If you want to rekey a Teradata Database device, you can either:

- Follow the process for standard IDT-Capable GuardPoints. For details, see ["Changing the Encryption Key on Linux IDT-Capable Devices" on page 193](#).
- Use the Backup/Restore method as described in ["Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method" on page 212](#). The only differences between the initial configuration described in that section and rekey process are:
  - You do not need to change the policy assigned to the CTE Metadata Directory.
  - You need to create a new In-Place Data Transformation policy, that specifies a key rule with the existing key used to encrypt the Database devices in the **Current Key** field, and the new XTS-AES 256 key that you want to use to encrypt the data in the **New Key** field.
  - When you re-guard the designated node in each clique, make sure that you use the new policy so that CTE knows it needs to re-encrypt the data with the new key after you restore the devices from the backup.

### Note

Make sure that you stop the Teradata database before you rekey the devices.



### WARNING

Rekeying a device must be executed only on *one and only one* of the nodes in the cluster. **DO NOT** prepare the device for rekey or guard the device with the new rekey policy on multiple nodes in the cluster because doing so initiates the data transformation process on the device on multiple nodes. The simultaneous attempts to rekey the data can cause data corruption of all of the data on the entire device.

## Access Rules to Apply on the Teradata Database Appliance

This section provides the instructions for creating a sample policy and signature sets specific to a Teradata Database Appliance to deny unprivileged users access to clear-text data on guarded devices. Such a policy requires the inclusion of specific security rules to restrict access to the Teradata Device GuardPoints for specific set of users, groups, and/or processes.

1. In the CipherTrust Manager, edit the host/client entry for the Teradata Database Appliance.
2. Go to the **Host/Client Settings** tab and add the following entries to the list of binaries:
  - |authenticator|/usr/pde/bin/pdmain
  - |authenticator|/opt/teradata/sm3g/bin/tdsmagent
  - |authenticator|/usr/pde/bin/pcl
3. Create a signature set for system processes that require access to guarded devices. The following system processes on the Teradata Database Appliance *must* be permitted access:
  - /usr/lib/systemd/systemd-udev
  - /sbin/dmsetup
  - /opt/teradata/vormetric/agent/vmd/bin/get\_disks
  - /sbin/pvdisplay
  - /sbin/lvdisplay
  - /usr/sbin/parted

You can include additional processes as needed.

4. Create a signature set for Teradata Database processes that require access to guarded devices. The following directories contain the Teradata binaries that require access:
  - /usr/pde/bin/\*
  - /usr/tdbms/bin/\*
  - /opt/teradata/gsctools/bin/\*
  - /opt/teradata/TDput/bin/\*

You can include additional processes as needed.



5. Create a system-level process set to associate the system-level processes listed in step 3, with the signature set you created in step 3. In the following example, this process set is called TD-Demo-system-process.

**2 Processes**

| Directory                             | File         | Signature              |  |
|---------------------------------------|--------------|------------------------|--|
| /opt/teradata/vormetric/agent/vmd/bin | get_disks    | TD-Demo-System-Process |  |
| /sbin/                                | dmsetup      | TD-Demo-System-Process |  |
| /sbin/                                | lvdisplay    | TD-Demo-System-Process |  |
| /sbin/                                | pwdisplay    | TD-Demo-System-Process |  |
| /usr/lib/systemd                      | systemd-udev | TD-Demo-System-Process |  |

6. Create a Teradata Database process set to associate the signature set for the Teradata Database binaries created in step 4.

**2 Processes**

| Directory                   | File | Signature           |     |
|-----------------------------|------|---------------------|-----|
| /opt/teradata/TDput/bin     | *    | TD-Demo-TD-Binaries | ... |
| /opt/teradata/gsctools/bin/ | *    | TD-Demo-TD-Binaries | ... |
| /usr/pde/bin/               | *    | TD-Demo-TD-Binaries | ... |
| /usr/tdbms/bin/             | *    | TD-Demo-TD-Binaries | ... |

7. Create the user set for the Teradata trusted group on your appliance.
8. If you have any other trusted groups that you want to include in your security rule, add a user set for each one of those trusted groups. You can add as many user sets to the policy as you need.
9. Create a user set for the root user and any other privileged users you want to add.
10. Verify that you have the complete signature sets, process sets, and user sets as shown in the examples below.
11. Apply the policy to the Teradata Device GuardPoints.

## Replication of IDT Metadata Files Across Members of a Clique

For Teradata, metadata for IDT-Capable GuardPoints is stored in external files in the CTE metadata directory. Because the Teradata appliance is a cluster of multiple hosts/clients that share access to the same devices across multiple nodes, a metadata file must be replicated across the nodes that are in the same clique within the cluster, including Hot Standby nodes. Replication is required when the initial data transformation has been completed, and it

is required again during subsequent rekeys. The availability of the metadata files on all members of the clique is critical for high availability of Teradata database in the event of a node failure because the data on the device cannot be accessed without the encryption key stored in the metadata.

Upon completion of data transformation on each device, the CTE Agent on that device automatically replicates the metadata file for that device to the other nodes in the clique using the Teradata `pcl` command. CTE uses `pcl` to both determine the other members of the clique and to replicate the metadata to those other members. When the metadata is replicated on the remote nodes, any existing metadata for the recently-transformed device already on those nodes is replaced with the updated metadata files sent by the CTE Agent. This replacement is achieved by sending the updated metadata file to all of the remote nodes through `pcl` and replacing each existing metadata file on the remote nodes with the most recent metadata files.

## Specific Issues to Consider

This section describes specific problems that may be encountered during data transformation and metadata replication. Manual user intervention will be required to recover from the reported issues. For troubleshooting and recovery steps, see ["Alerts and Errors" on page 223](#).

### General PCL Error

If the `pcl` command fails during the replication process, a message indicating the error will be logged. The messages are tagged with IDT-TD-ALERT. For example:

- IDT-TD-ALERT: Node <node name> did not respond to `pcl` command
- IDT-TD-ALERT: Failed to distribute IDT-Capable metadata file to remote nodes

These errors indicate that the node specified in the first error did not respond to a `pcl` command during metadata distribution. As a result, the metadata distribution must be manually performed across the clique before access to the device is possible.

### Offline Node in Clique During Data Transformation

If a node is offline during data transformation and metadata replication process, CTE will log a message that metadata replication to the target node failed. The administrator will be required to manually replicate the metadata file to the offline node when the node comes online. The metadata file must be replicated before the database is brought up.

To do so:

- Run the `voradmin td distribute <device name>` command to distribute the metadata file of each device. The command will copy, or update, the metadata file on the agent that has come online. For example:  

```
voradmin td distribute \
/dev/disk/by-id/tdmp-360080e500043092c0000b46f5c34c018-part3
```

### Adding a New Node to a Clique

When you add a new node to an existing clique, the metadata files of all disks shared in the clique must be manually replicated to the newly added node before any of the guarded devices on the new node are enabled. This should be done manually using the `voradmin td distribute` command, as described above, by the Teradata administrator as part of joining the cluster.

## Interoperability with Host Groups

For Teradata, when you create a new IDT-Capable GuardPoint using the `voradmin idt config xform` command, or when you subsequently rekey an existing IDT-Capable GuardPoint with the `voradmin idt rekey` command, the device must be initialized and guarded on one and only one of the nodes in the cluster. That means the IDT-Capable GuardPoint cannot be part of a client group when an IDT-Capable GuardPoint is created or rekeyed because membership in a client group means that any data transformation on any member of the client group is initiated for that member on all nodes in the client simultaneously. In the case of a Teradata cluster, multiple nodes simultaneously trying to perform data transformation on a particular device can lead to data corruption of all data on the entire device.

After the device has been guarded, or rekeyed, and the metadata files have been replicated to other members of the Teradata clique, then you can then rejoin the client with the client group.

### To configure a new device whose host/client is part of a host/client group:

1. Make sure that there is no GuardPoint for the device at the client group level.
2. Designate one of the nodes in the cluster as the node that you will use to initialize the GuardPoint and for the initial data transformation when the device is guarded for the first time.
3. On the designated node, initialize the device using the `voradmin idt config -external xform <device name>` command. For details, see ["Initialize and Guard the Database Devices Using the Standard Initialization Method" on page 209](#).
4. Guard the device on the designated node using an In-Place Data Transformation policy. For details, see ["Guard the Devices as IDT-Capable GuardPoints on CM" on page 210](#).
5. Wait for the data transformation to complete on the host and for CTE to replicate the metadata to the other members of the clique.
6. You can verify that the metadata file has been distributed to the other nodes in the clique by running `md5sum /var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata` on each node in the clique.
7. Remove the IDT-Capable GuardPoint you added earlier in this procedure.
8. Guard the device through the client group to make sure that all nodes in the cluster recognize it as guarded.

### To rekey a new device whose host/client is part of a host/client group:

1. Unguard the IDT-Capable GuardPoint through the client group and make sure that it has been removed from all nodes in the cluster.
2. Designate one of the nodes in the cluster as the node that you will use to prepare the GuardPoint for rekeying and to perform the data transformation when the device is guarded with the new policy.
3. On the designated node, prepare the device for rekey using the `voradmin idt rekey <device name>` command.
4. On the other nodes in the clique, make sure that the device metadata has been renamed running `ls /var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata* on` each of the other nodes. On these other nodes, the metadata file for the device should be renamed to `/var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata_xforming`.
5. Guard the device on the designated node with the In-Place Data Transformation policy that specifies the new key you want to use for the device.

6. Wait for the data transformation process to complete, and then make sure that the metadata for the device has been updated on the other nodes in the clique and the renamed metadata files have been removed. Each node should have identical copies of `/var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata` and `/var/opt/teradata/vormetric/vte-metadata-dir/vormetric/secvm_<device>_metadata_xforming` should *not* exist on any node.
7. Remove the IDT-Capable GuardPoint that you created earlier in this procedure on the designated node.
8. Guard the device through the client group to make sure that all nodes in the cluster recognize it as guarded.

## Generate IDT-Capable metadata for Teradata Storage Expansion

You can use the `voradmin td expand` command to configure new disks as IDT-capable devices during Teradata disk expansion.

### Note

Guard each new device using the same policy applied to an existing guarded IDT-Capable device.

### Note

Only use this option on new disks added during Teradata disk expansion.

After the new disk is visible to all members of the clique, perform the following steps to configure the new disk:

1. Configure the new disk using the following command on any node in the clique.

```
voradmin td expand <newDiskName> <existingDiskName>
```

**Note:** Do not run this command on the same device on multiple nodes.

2. Guard the new device on the DSM/CipherTrust Manager on all members of the clique.

For disk expansion that includes the addition of a new node to the clique, perform the following steps:

1. Install CTE on the new node and register the node to the DSM/CipherTrust Manager, but **DO NOT** add the node to the host/client group, if using one.

2. On the new node, create the CTE Metadata directory, type:

```
mkdir -p /opt/teradata/vormetric/vte-metadata-dir
```

3. On one of the existing nodes in the clique, run the `voradmin td distribute` option to copy all existing metadata to the new node, type:

```
voradmin td distribute < node1 name, node2 name, etc..> <device>
```

4. On the same node selected in step 3, use the `voradmin td expand` option to configure the new disk as an IDT-Capable device, type.

```
voradmin td expand <newDiskName> <existingDiskName>
```

**Note:** The `td expand` option must be run on each new disk.

5. On DSM/CipherTrust Manager, guard the new disk in the host/client group. You can verify that the new disk is now guarded on all nodes except the new node.
6. On the DSM/CipherTrust Manager, add the new node as a member to the host/client group. Once the new node is added as a member, DSM/CipherTrust Manager will push the security policy to the new node and guard all of the disks with metadata files on the new node.

## Best Practices

### Using a of Host/Client Group to Guard Metadata Directories

Thales recommends that customers use a host group exclusively for guarding and management of the CTE external metadata directories across all nodes in Teradata cluster. (For details about the CTE metadata directory, see "[Location of the CTE Private Region](#)" on page 205.)

By using a Host/Client Group, you can apply the same policy to the external metadata directory (`/opt/vte/vte-metadata-dir`) on all of the CTE protected nodes in the cluster.

### Using a Host/Client Group for Guarding Teradata Devices in a Clique

Thales recommend that customers use a Host/Client Group to guard and manage the devices shared by all CTE protected nodes in a clique. Each clique in the cluster should be associated with a separate Host/Client Group. When configuring the Host/Client Group, the members for that client group are the nodes that belong within a clique in the cluster.

**Example 1:** If you have a cluster consisting of 4 cliques where there are 2 nodes sharing disks in each clique, you would create 4 Host/Client Groups, one for each clique. In each Host/Client Group, you would add the 2 nodes that are associated with the clique.

**Example 2:** If you have a cluster consisting of 1 clique where there are 6 nodes sharing disks in the clique, you want to create 1 client group with all 6 members that belong within that clique.

Below is an example of the Host Groups for a cluster consisting of three cliques and one metadata group:

| Select                   | Name                     | Cluster Group | Description                             |
|--------------------------|--------------------------|---------------|-----------------------------------------|
| <input type="checkbox"/> | TD-cluster1-clique-1     |               | host/client group for cluster1 clique-1 |
| <input type="checkbox"/> | TD-cluster1-clique-2     |               | host/client group for cluster1 clique-2 |
| <input type="checkbox"/> | TD-cluster1-clique-3     |               | host/client group for cluster1 clique-3 |
| <input type="checkbox"/> | TD-cluster1-metadata-dir |               | host/client group for cluster1 metadata |

### Best Practice for Preparation for Initial Data Transformation or Rekey

With large amounts of data managed in Teradata clusters, duration of initial data transformation and/or subsequent rekey can be very long and challenging because the database must be stopped during the data transformation process. Although you cannot transform data without shutting down the database, you may be able to reduce the transformation time by distributing the transformation of the guarded devices across multiple nodes in your cluster.

Because multiple nodes in a cluster within a clique share access to the same devices, you can designate a subset of devices to each node within the clique for initial data transformation or subsequent rekey. This requires unguarding all of the IDT-Capable devices in the Host/Client Group associated with the clique, and re-enabling each IDT-Capable GuardPoint on the node designated for the data transformation of that IDT-Capable GuardPoint. The device then remains unguarded on other nodes until the data transformation completes.

After the data transformation completes, you can then enable the guarded devices at the Host/Client Group level. Enabling the guarded device at Host/Client Group level enables the guarded devices on all of the nodes within the clique to share access to the device.

After transforming data and guarding each device on all of the nodes, you can then restart the database.

As described in ["Interoperability with Host Groups" on page 219](#), the metadata file for each device is replicated to the hosts in the cluster that share access to the device. As noted, replication of metadata files is automatic.

## Uninstalling CTE from the Teradata Cluster

The following procedure describes how to remove CTE from your Teradata cluster. Note that uninstalling CTE will unprotect the pdisk devices that CTE is protecting/encrypting. Because the data on the pdisks remain encrypted after uninstalling CTE, the entire clear-text data must be restored to the pdisks from backup. Therefore, you must have a full backup of your database to restore to your cluster before you uninstall CTE. See the Teradata Administration Guide or your Teradata Customer Support Representative for information about backing up your database in full.

### Note

You must be familiar with CTE operations for protecting Teradata clusters and IDT-Capable GuardPoints.

To uninstall CTE:

1. Make sure you have access to a full database backup so that you can restore your Teradata database after uninstalling CTE.
2. Shut down your Teradata database.
3. Use the `secfsd -status guard` command to view the full list of device GuardPoints guarded by CTE on each node of your cluster. From this information, create a consolidated list of GuardPoints with one entry per device in the consolidated list. Note that the devices accessed by the nodes in the same clique are shared devices on each node. The consolidated list must include all devices that are protected in your cluster.
4. Perform the following steps on the CipherTrust Manager:
  - a. Click on the client group name associated with your Teradata cluster and then click the **GuardPoints** tab.
  - b. Select the device names listed in GuardPoints table. The devices listed in GuardPoints table are the same devices listed in the consolidated list of devices compiled in step 3.
  - c. Disable the selected device GuardPoints on all nodes in the cluster.
  - d. Wait for device status to change from green circle to red circle in the Status column for all the selected devices.
  - e. Select the same devices.
  - f. Click **Unguard** to remove the selected device GuardPoints on all nodes in the cluster.
5. Perform the following steps on one of the nodes of every clique in your cluster:
  - a. Run the `voradmin idt delete` command to delete the IDT-Capable GuardPoint configuration recorded for each device. This command automatically deletes the GuardPoint configuration across the nodes in the same clique.

```
voradmin idt delete <device name>
```
  - b. Repeat the previous step on the other devices in your cluster, if any.
6. Go back to the DSM and do the following:
  - a. Select the Metadata Directory GuardPoint in the GuardPoints table of the same host group.
  - b. Click **Disable** to disable the Metadata GuardPoint directory on every node in the cluster.
  - c. Select the Metadata Directory GuardPoint in the GuardPoints table again and then click **Unguard** to remove the Metadata Directory GuardPoint on all the nodes in the cluster.

7. On every node in the cluster:
  - a. Run the following command to stop CTE service on each node:

```
/etc/vormetric/secfs stop
```
  - b. Run the following command to uninstall the CTE Agent on each node:

```
/opt/teradata/vormetric/agent/vmd/bin/uninstall
```

To restore a full backup of the database to your cluster, contact your Teradata Customer Support Representative (CSR) to perform System Initializer and do a full restore.

## Alerts and Errors

This section lists the alerts and errors that may be encountered during system operations on Teradata Appliance. Refer to Alerts and Errors listed in the chapter on IDT-Capable Device GuardPoints for additional Alerts and Errors.

### General Errors

The errors in this section indicate what step failed during CTE system operations for Teradata clusters and will always be paired with an operation error which indicates which CTE system operation failed. For example:

- IDT-TD-ALERT: Node <node name> did not respond to pcl command
- IDT-TD-ALERT: Failed to complete rekey on remote nodes

This pairing of errors indicates that the steps after data transformation were unsuccessful and that the `voradmin` command `voradmin td rekeyed <device name>` is the one that failed.

For details about the `voradmin td rekeyed` command, see the `voradmin` manpage.

#### IDT-TD-ALERT: Node <node name> did not respond to pcl command

The node listed did not respond to a `pcl` command to perform an operation.

**Solution:** Check if the node is still online and responding. If it is, run the `voradmin` command to repeat the failed operation. This error message will always be accompanied by one of the other IDT-TD-ALERTS which will determine which `voradmin` command to run to correct the error.

#### IDT-TD-ALERT: Node <node name> failed to perform voradmin task

The node listed failed to perform a `voradmin` command invoked through `pcl`.

**Solution:** Correct the issue blocking `voradmin` and run the `voradmin` command to repeat the failed operation. This error message will always be accompanied by one of the other IDT-TD-ALERTS which will determine which `voradmin` command to run to correct the error.

#### IDT-TD-ALERT: Failed to find clique for disk <device>

CTE failed to locate the clique for the device specified.

**Solution:** Verify that `/usr/lib/tmpfiles.d/pdisk.conf` is properly configured on all nodes in the Teradata cluster.



## IDT-TD-ALERT: Failed to move or rename IDT-Capable metadata file on remote nodes

This error can be due to two issues; one, metadata files that have been distributed to the remote nodes cannot be moved into the metadata directory, or two, metadata files on remote nodes cannot be renamed in preparation for rekey.

**Solution:** For the first issue, rerun `voradmin td distribute <node name> <device name>` command once any other problems have been resolved. For the second issue rerun `voradmin td rekey <device name>` to prepare for rekey again once any other problems have been resolved.

For details about the `voradmin td` commands, see the `voradmin` manpage.

## IDT-TD-ALERT: Failed to get GuardPoint status on remote nodes

The GuardPoint for the device is still present on the CipherTrust Manager for remote nodes or CTE failed to ascertain the GuardPoint status for remote nodes.

**Solution:** Verify that the GuardPoint for the device has been removed from the CipherTrust Manager for remote nodes and then rerun the `voradmin` command. This error message will always be accompanied by one of the other IDT-TD-ALERTS which will determine which `voradmin` command to run to correct the error.

## Operation Errors

These errors indicate which CTE system operation failed and what command should be run upon resolution of the issue that caused the error.

## IDT-TD-ALERT: Failed to distribute IDT-Capable metadata file to remote nodes

CTE failed to copy the IDT-Capable metadata file from the current node to all nodes in the clique.

**Solution:** Correct any issues blocking `pcl` or `voradmin` and then run `voradmin td distribute <node name> <device name>` to attempt copying the metadata file over to the remote node.

For details about the `voradmin td distribute` command, see the `voradmin` manpage.

## IDT-TD-WARNING: Failed to delete IDT-Capable metadata file on remote nodes

CTE was unable to remove the IDT-Capable metadata file for a device from a remote node.

**Solution:** Run `voradmin idt delete <device name>` on the remote node after correcting issues impeding `voradmin`. Members of the clique that have successfully removed the metadata file will not be adversely affected if an attempt to remove an already de-configured device occurs.

## IDT-TD-ALERT: Failed to complete rekey on remote nodes

CTE was unable to complete the rekey operation on remote nodes to make the device available for guarding again.

**Solution:** Correct the issue and run `voradmin td rekeyed <node name> <device name>` to signal the remote node that rekey is complete and provide an updated copy of the metadata file to that remote node.



# Chapter 22: Upgrading CTE on Linux

---

This chapter describes how to upgrade an existing VTE for Linux host to CipherTrust Transparent Encryption (CTE) for Linux.

This chapter contains the following sections:

|                                 |     |
|---------------------------------|-----|
| Upgrading CTE .....             | 225 |
| Scheduled Upgrade Feature ..... | 225 |

## Upgrading CTE

This section describes the generic instructions for interactively upgrading CTE. If there are any changes to this procedure for the current release of CTE, those changes will be documented in the CTE Release Notes.

If you want to schedule an upgrade to occur the next time the system boots, see "[Scheduled Upgrade Feature](#)" below.

1. Stop any application accessing files in the GuardPoint.
2. Log on to the host where you will upgrade CTE. You must have root access.
3. Copy or mount the installation file onto the host system.
4. Start the upgrade by executing the install program for the release to which you want to upgrade. If you want to automatically accept the CTE License Agreement, you can include the `-y` parameter.

For example, the following command upgrades the product to version 7.2.0.128 after the user manually reviews and accepts the CTE License Agreement:

```
./vee-fs-7.2.0-128-rh8-x86_64.bin
```

The following command upgrades the product to version 7.2.0.128 but automatically accepts the CTE License Agreement:

```
./vee-fs-7.2.0-128-rh8-x86_64.bin -y
```

5. Follow the prompts. During an upgrade, the following message displays. Enter Y at the prompt:

```
Upgrade detected: this product will be stopped and restarted.
Do you wish to proceed with the upgrade? (Y/N) [Y]: Y
Installation success.
```

You will not do the registration steps since CTE is already registered with the DSM.

6. To verify that the upgrade was successful, use the `vmd -v` command:

```
$ vmd -v
Version 6, Service Pack 2
7.2.0.128
2022-02-04 09:45:20 (IST)
Copyright (c) 2009-2022, Thales. All rights reserved.
```

## Scheduled Upgrade Feature

This section describes how to run the scheduled upgrade feature on CTE for Linux systems. This option is available for CTE version 6.1.3 or later. It contains the following topics:

**Note**

Scheduled upgrade on reboot is not supported on HDFS nodes.

## Warnings for CTE for Linux

- Prior to upgrading your system, perform a backup or take a snapshot of your system.
- As with prior CTE versions, DSM connectivity is required during upgrade.
- Yum updates, or OS patches, should be done prior to CTE upgrade on reboot.
- If you upgrade from a compatible kernel to an incompatible kernel, the `secfs` module will fail to load on the next reboot.
- You may see the following behavior if the upgrade on reboot fails due to a crash, or a power failure, (this is similar to a failure during a normal upgrade).
  - If a crash, or power failure, occurs before the upgrade executes, the upgrade will not take place, and the currently installed CTE version continues to run after the reboot. Restart the system to upgrade successfully.
  - If a crash, or power failure, occurs during the upgrade, CTE may enter an inconsistent state. Perform a restore from your backup, or roll back to the snapshot that you just took. Then, start the upgrade again.
  - If a crash, or power failure, occurs after a successful upgrade, then the new version will run on the next reboot. No user intervention is required in this case.
- During reboot or shutdown, all applications and services dependent on CTE services must be stopped before a scheduled update takes place. Failure to stop these services can result in an aborted scheduled upgrade during the system reboot. Examples of situations that may cause an aborted upgrade are applications with open files in a CTE GuardPoint, or a third party anti-virus software doing periodic scans.

For examples of how to set up CTE start/stop dependencies with other programs, see [Chapter 6: "CTE and systemd" on page 54](#).

See the `systemd` chapter in the *CTE Agent for Linux Advanced Configuration and Integration Guide* for examples of how to set up CTE start/stop dependencies with other programs.

## Using the Scheduled Upgrade Feature

The following procedure describes how to use `voradmin` to schedule an upgrade that will be applied the next time the machine reboots.

1. If you want to check which version of CTE for Linux you currently have installed, use the `vmd -v` command:

```
$ vmd -v
Version 6, Service Pack 2
7.1.0.66
2022-02-04 11:09:40 (IST)
Copyright (c) 2009-2022, Thales. All rights reserved.
```

2. To schedule an upgrade on reboot, use the following commands:

```
voradmin upgrade schedule <path_to_CTE_installer_binary> -y [-t <custom_extraction_path>]
```

where:

- `<path to CTE installer>` is the full path to the CTE installation file for the release to which you want to upgrade. For example, `./vee-fs-7.2.0-128-rh8-x86_64.bin`.
- `-y` is an optional parameter that automatically accepts the CTE License Agreement. If you do not specify this parameter, the installer displays the CTE License Agreement and you must manually accept it before the upgrade can be scheduled.
- `[-t <custom_extraction_path>]` is an optional parameter that specifies the path to a custom binary extraction path directory in which you want CTE to store the temporary files it needs during the upgrade. The default is `/var/tmp/`, but in some systems, `/var/tmp/` is restricted and not available for use.

**Exceptions:**

- Do not use the `-t` option on protected paths, GuardPoint paths, or paths which do not have sufficient permissions to copy/extract the target binary.

For example, if you are upgrading to version 7.2.0.xx and you want to automatically accept the license agreement and use a custom directory, you would type:

```
$ voradmin upgrade schedule ./vee-fs-7.2.0-98-rh8-x86_64 -y -t /my_custom_dir
```

**Note:** The `[-t]` option is only supported by CTE v7.2 and subsequent versions.

3. If you want to verify that the upgrade was successfully scheduled, use the `voradmin upgrade show` command:

```
$ voradmin upgrade show
Upgrade on reboot is currently scheduled.
Current CTE version is 7.1.0.66, upgrade on reboot scheduled for CTE 7.2.0-128.
```

4. Reboot the machine, then log in and verify that the upgrade was successful.

```
$ vmd -v
Version 6, Service Pack 2
7.2.0-128
2022-02-04 09:45:20 (IST)
Copyright (c) 2009-2022, Thales. All rights reserved.
```

**Note:** Appropriate logs will be logged in syslog.

## Performing a Manual Upgrade When an Upgrade is Already Scheduled

If an administrator runs a manual upgrade after an upgrade has already been scheduled, the installer displays the following warning:

```
WARNING: upgrade on reboot is already scheduled for 7.2.0-128.
Do you want to cancel scheduled upgrade on reboot ? (Y/N) [Y] :
```

If the administrator does *not* cancel the scheduled upgrade, the scheduled upgrade takes precedence and the manual upgrade fails with the message:

```
Already scheduled upgrade on reboot remains intact.
Installation failure.
```

If the administrator wants to proceed with the manual upgrade immediately, they must enter Y at the prompt to cancel the scheduled upgrade:

```
WARNING: upgrade on reboot is already scheduled for 7.2.0-128.
Do you want to cancel scheduled upgrade on reboot ? (Y/N) [Y] : Y
Removed symlink /etc/systemd/system/multi-user.target.wants/secfs-upgrade.service.
WARNING: upgrade on reboot is cancelled for 7.2.0-128. Proceeding with manual upgrade.
Upgrade detected: this product will be stopped and restarted.
Do you wish to proceed with the upgrade? (Y/N) [Y]: Y
.....
Upgrade success.
```

To verify that the upgrade succeeded, the administrator can use the `vmd -v` command:

```
$ vmd -v
Version 6, Service Pack 2
7.2.0-128
2022-02-04 09:45:20 (IST)
Copyright (c) 2009-2022, Thales. All rights reserved.
```

To cancel an existing scheduled upgrade on reboot:

```
$ voradmin upgrade cancel
Removed symlink /etc/systemd/system/multi-user.target.wants/secfs-upgrade.service.
Successfully cancelled upgrade on reboot
```

# Chapter 23: Uninstalling CTE from Linux

This chapter describes how to upgrade an existing VTE for Linux host to CipherTrust Transparent Encryption (CTE) for Linux.

This chapter contains the following sections:

|                      |     |
|----------------------|-----|
| Considerations ..... | 229 |
| Procedure .....      | 229 |

## Considerations

- The CTE Agent must be removed from the Linux host before the host is removed from the key manager with which it is registered.
- Database applications like DB2 and Oracle can lock the user space while they run. If the uninstall fails because a GuardPoint is in use, determine which applications are using the files in the GuardPoint and stop them. Then run the uninstall again.
- Commands like `fuser` and `lsof` might not reveal an active GuardPoint because they detect active usage, not locked states. Although it may appear that a GuardPoint is inactive, it may be in a locked state. Under this condition, software removal may fail with an error similar to the following:

```
/home: device is busy.
```

## Procedure

1. Stop any application from accessing files in the GuardPoint.
2. In the key manager with which this host is registered, do the following:
  - Decrypt any data you want to use after uninstall. After the CTE Agent software is removed, access to data is no longer controlled. If data was encrypted, it will remain encrypted. If decrypted or copied out of the GuardPoint, the data is visible as clear text.  
  
This decryption must be done on *every* GuardPoint on the host if you want to access all existing data on the host.
  - Make sure the Agent and System locks have been disabled for the host.
  - Thales recommends that you remove all GuardPoints from the host before you uninstall the CTE Agent.

*Do not* remove the host from the key manager yet.
3. Log on to the host as `root`.
4. Change the directory to an unguarded location (for example, `/`).



### CAUTION

**Do not change ( `cd` ) into the `/opt/vormetric` directory or into any directory below `/opt/vormetric`. If you run the uninstaller from `/opt/vormetric` or any of its subdirectories, the package removal utility may fail and return the following message:**

```
You are not allowed to uninstall from the /opt/vormetric directory or any of its sub-directories.
```

```
Agent uninstallation was unsuccessful.
```

5. Start the uninstall. Type:

```
/opt/vormetric/DataSecurityExpert/agent/vmd/bin/uninstall
Would you like to uninstall the vee-fs package? (Y/N) [Y]: Y
.....
Success!
```

6. Remove the host record from the key manager.

# THALES

## Contact us

For office locations and contact information,  
visit [cpl.thalesgroup.com/contact-us](https://cpl.thalesgroup.com/contact-us)

> [cpl.thalesgroup.com](https://cpl.thalesgroup.com) <

