thalescpl.com

THALES

VTE Agent Installation and Configuration Guide

Version 6.3.1

Document version 9

January 29, 2021

VTE Agent Installation and Configuration Guide
6.3.1
Document version 9
January 29, 2021

Notices and License

All information herein is either public information or is the property of and owned solely by Thales DIS France S.A. and/or its subsidiaries or 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 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.

Contents

PREFACE

The VTE Agent Installation and Configuration Guide describes how to install and configure Vormetric Transparent Encryption. This guide describes how to install and configure Vormetric Transparent Encryption (VTE) on hosts. After you install VTE and configure the VTE with the DSM appliance, VTE can protect data on hosts.

AUDIENCE

The Agent Installation and Configuration Guide is intended for system administrators who install and configure VTE on host machines

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 1-1: Typographical Conventions

Convention	Usage	Example
bold regular font	GUI labels and options.	Click the **System** tab and select **General Preferences**.
bold italic monospaced font	Variables or text to be replaced	https://<***Token Server name***>/admin/ Enter password: <***Password***>
`regular monospaced font`	Command and code examples XML examples	Example: `session start iptarget=192.168.253.102`
italic regular font	GUI dialog box titles	The *General Preferences* window opens.
	File names, paths, and directories	*/usr/bin/*
	Emphasis	*Do not resize the page.*
	New terminology	*Key Management Interoperability Protocol (KMIP)*
	Document titles	Refer to the *DSM Administrators Guide* for information on how to administer your DSM Appliance.

Table 1-1: Typographical Conventions (Continued)

Convention	Usage	Example
quotes	File extensions Attribute valuesTerms used in special senses	".js", ".ext" "true" "false", "0" "1+1" hot standby failover

Notes, cautions, and warnings

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

A Note provides a tip, guidance, or 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.

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

For support and troubleshooting issues:

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

For Thales Sales:

- https://enterprise-encryption.vormetric.com/contact-sales.html
- sales@thalesesecurity.com
- (888) 267-3732

Chapter 1: Overview

This chapter contains the following sections:

Introduction

This document describes how to install and configure Vormetric Transparent Encryption for Linux and Windows to protect data on host computers. A host computer protected with a VTE agent is referred to as a *protected host* in this document. VTE supports multiple operating system environments, and you can deploy it on physical devices as well as virtual machines.

VTE secures data with little impact to application performance. It requires no changes to your existing infrastructure and supports separation of duties between data owners, system administrators, and security administrators.

The VTE solution consists of a *Data Security Manager* (DSM) and VTE residing on your hosts, your protected hosts, or servers.

- The DSM is the central component of the VTE solution. You can set it up as either a security-hardened physical appliance or a virtual appliance. The DSM stores and manages data encryption keys, data access policies, administrative domains, and administrator profiles.
- The VTE communicates with the DSM and implements the security policies on their GuardPoint systems.
- You can apply VTE to GuardPoints on servers on-site, in the cloud, or a hybrid of both.

Note: Refer to the VDS Compatibility Matrix for a list of VTE versions and supported operating systems.

VTE protects data at rest. VTE protects data residing on Direct Attached Storage (DAS), Network Attached Storage (NAS) or Storage Area Networks (SAN). This can be a mapped drive or mounted disk, as well as through Universal Naming Convention paths.

The VTE installation and configuration process:

1. Install VTE on the protected host.
2. Add the protected host's fully qualified domain name (FQDN) or IP address to the DSM.
3. Register the protected host with the DSM so they can communicate with each other.

How to protect data with VTE

VTE uses policies to protect data. You can create policies to specify file encryption, data access, and auditing on specific directories and drives on your protected hosts. These directories are called *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

An administrator accesses the Management Console through a web browser. You must have administrator privileges to create policies using the DSM Management Console. VTE implements policies once they are pushed to the protected host.

VTE 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 or not guarding a directory can result in corrupt data.

Chapter 2: Installing VTE for Windows

This chapter describes how to install and configure VTE on Windows computers.

Both VTE or host installer and the DSM administrator assist in the installation.

This chapter contains the following sections:

Installation Overview

The installation and configuration process consists of three basic steps:

1. Install VTE on the protected host.
2. Manually add the protected host to the DSM using the fully qualified domain name (FQDN), or IP address. (This is automatically performed using the *Shared Secret Registration* method.)
3. Register the protected host with the DSM so they can communicate with each other.

Assumptions

- The IP addresses, routing configurations, and DNS addresses allow connectivity between the DSMs and all hosts which contain VTE installations.
- If the protected host is a virtual machine, the VM is deployed and running.
- For all types of upgrades, including interactive (GUI-based) and scheduled upgrades, the protected host must be able to connect to the DSM that it is registered to or the upgrade will fail.

Requirements

- You must install VTE on the system drive. Do *not* install VTE on a network share volume.
- The host on which you want to install VTE *must* support AES-NI hardware encryption. If it does not, any attempt to install or upgrade VTE to release 6.3.0 or later will fail.

Installation Prerequisites

This section lists tasks you must complete and information you must gather before installing VTE:

- "Hardware Association" on page 25
- "One-way Communication" on page 25.
- "Determine the Installation Method" on page 26

Determine your VTE Registration Method

You can register the protected hosts with the DSM using either the *Fingerprint method* or the default *Shared Secret method*.

- **Fingerprint method** requires the DSM administrator to add the FQDN, or IP address, of each protected host to the DSM before registering VTE.

  During the registration, the DSM generates the certificate and passes it down to VTE along with the fingerprint. The security administrator must verify the fingerprint to make sure the certificate is valid.
- **Shared Secret method** requires the DSM Administrator to create a *shared secret* password—a case-sensitive string of characters—for auto-registering a domain or host group.

  VTE installers use the shared secret to add and register protected hosts to the DSM for a domain or host group. The DSM administrators can optionally add host names or IP addresses to the DSM. There is no need to verify that the protected host and DSM share valid certificates. You can add multiple protected hosts dynamically with a single shared secret password during VTE installation and registration.

  After the DSM Administrator creates a shared secret for the domain or host group in which the new protected host will reside, obtain it and the validity period (one hour, day, week, or month) and register within that period.

Host Name Resolution

Host name resolution is the method of mapping a host name to an IP address. During this configuration process, enter either the FQDNs, or IP addresses, of your DSM and protected hosts. If you use FQDNs, your protected hosts must be able to resolve their DSM host names, and the DSM must be able to resolve its protected hosts.

Note: The exception to this requirement is if you approve of only VTE-initiated communication between the DSM and the protected host. See "One-way Communication" on page 25 for more discussion.

A Domain Name Service (DNS) server is the preferred method of host name resolution. Use the following guidelines for host name resolution:

1. If you use DNS, use the FQDNs for the DSM and host for the installation and configuration procedures in this chapter.
2. If you do NOT use a DNS server, complete one of the following tasks on the DSM and the protected hosts:
   - Use the IP addresses of the DSM and protected hosts.
   - Add an entry for the DSM in the `C:\WINDOWS\system32\drivers\etc\hosts` file on the Windows protected hosts.

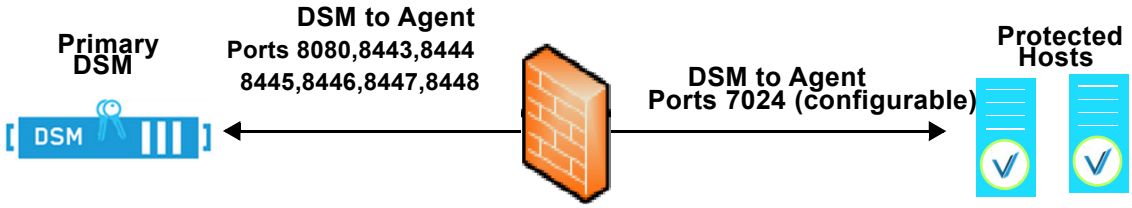
Note: You may have to obtain the DSM server names and IP addresses from the DSM administrator.

3. Repeat this procedure for each protected host and add an entry for all DSM nodes.

Port Configuration

If a protected host must communicate with a DSM through a firewall, open the ports in the firewall as shown in the following figure.

Figure 2-1: Ports to open between DSM and protected host



Note: For a complete list of ports required for the DSM, see the *VDS Administration Guide*.

Port Usage in One Way Communications Mode

By default, polling from the agent host to the DSM when running in one-way communications mode uses HTTP via port 8080. If the agent is configured to use secure polling, then polling is performed using HTTPS via port 8448 (in suite B mode) or port 8445.

Hardware Association

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

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

To verify if hardware association (cloning prevention) is enabled on the protected host, launch the Windows command line and run the following command:

```
C:\Program Files\Vormetric\DataSecurityExpert\agent\vmd\bin> vmsec.exe hwok
```

To change the status from enable to disable or vice versa:

1. Open the system tray and right-click on the Vormetric icon.
2. Select **Register Host**.
3. Follow the prompts to re-register VTE with the DSM.
4. Select **Enable hardware association** in the wizard.

One-way Communication

In some deployments, VTE might not be visible to the DSM through normal network communications. For example, when the host on which VTE is installed:

- is behind NAT

- is behind a firewall
- is not permanently connected to a communication channel to the DSM
- is unable to resolve the host name to an IP address

In these situations, VTE can initiate VTE-only communication to the DSM. This feature is called one-way communication and works by having VTE poll the DSM for any policy messages or changes, then downloading changes as required.

The downside of one-way communication is that the DSM cannot issue any queries to VTE. For example, the DSM administrator cannot browse host directories or User IDs.

If you want the DSM to be able to contact VTE directly, abandon the Register Host process if you get the following message during the registration process:

Determine the Installation Method

There are two methods for installing VTE:

- **Typical installation:** This is the most common and recommended type of installation. Use this for installing VTE on one host at a time. See “Windows Typical Install” on page 27.
- **Silent installation:** Create pre-packaged installations by providing information and answers to the installation questions. Use silent installations when installing on a large number of hosts. See “Windows Silent Install Using MSI or Self-Extracting .exe” on page 30.

VTE Install Checklist

Use this table to verify prerequisites and collect the information you need for the installation.

Checklist item	Status
Obtain VTE installation binary from Vormetric support, either self-extracting .exe file or MSI. The format for self-extracting executable VTE file names is: vee-<*agent_type-build-system*>.exe Example: `vee-fs-6.3.1.38-win64.exe` The format for MSI VTE file names is: vee-<*agent_type-build-system*>.msi Example: `vee-fs-6.3.1.38-win64.msi`	
Fully Qualified Domain Name (FQDN) of the DSM	
IP address or FQDN of the host	
Administrator password for the host	
If using Shared Secret Registration, obtain the following from the DSM administrator: 1) Shared secret 2) Domain 3) Host group, if applicable 4) (Optional) A description for the host.	

Checklist item	Status
If using the Fingerprint Registration ask the DSM administrator to add the host to the DSM and check the **Registration Allowed** check box. After checking the fingerprint, select the **Communication Enabled** check box.	
Addressed "Host Name Resolution" on page 24 for the protected hosts and DSM?	
Set "Port Configuration" on page 25	
Do you want "Hardware Association" on page 25?	
Is "One-way Communication" on page 25 required?	
Synchronize host clock to DSM clock.	
Set network subnet mask on the host (unless you are using one-way communication)	
Preferred DNS Server (if using FQDNs):	

Windows Typical Install

Use the *typical install* feature to install VTE on a host by manually answering each installation question. A *silent install* pre-packages installations by providing information with scripts that answer the installation questions. Use silent installation when installing on a large number of hosts (see "Windows Silent Install Command Line Options" on page 30).

Typically, you register VTE with the DSM as part of the installation process as described below. However, you may postpone registration if you have a plan to register VTE later. VTE cannot protect data on the host until you register VTE with the DSM.

Install VTE on the Protected Host

Note: If you are upgrading a protected host, stop all activity on the application before upgrading.

1. Log on to the host as a Windows user with administrative privileges.
2. Copy the installation file onto the Windows system.
3. Double-click the installation file.
4. The **InstallShield Wizard for Vormetric File System Agent** window opens. Verify the version of VTE you are installing and click **Next**.
5. The *License Agreement* appears. Accept the *License Agreement* and then click **Next**.
6. The *Destination Folder* window opens. Click **Next** to accept the default folder or click **Change** to select a different folder.

Note: If you have VTE already installed on the system and are upgrading, you can not change the Destination Folder.

7. The **Ready to Install the Program** window opens. Click **Install**. When the installation is finished, the **Install Shield Wizard Completed** window opens.
8. To register now using the *Certificate Fingerprint* method, go to "To Register VTE Using the Certificate Fingerprint Method" on page 29.

   To register now using the *Shared Secret* method, go to "To Register VTE Using the Shared Secret Registration Method" on page 28.

9. If you want to register VTE later, clear the check box for **Register Vormetric File System now**, then click **Finish**.

To Register VTE Using the Shared Secret Registration Method

1. If the installer is not currently running, start the registration wizard by running `C:\Program Files\Vormetric\DataSecurityExpert\agent\shared\bin\register_host.exe`. Otherwise, click **Next** in the install wizard.
2. Click **Next** to confirm that you want to register the host shown in the dialog box.
3. VTE prompts you to enter the host name of the Primary DSM. If necessary, ask the DSM administrator for the **Server name** on the dashboard of the DSM Management Console. Type the name of the DSM and click **Next**.
4. VTE prompts you to enter the protected host name or IP address. If you use a host name, it must be resolvable by the DNS server (see "Host Name Resolution" on page 24). You can type the name/IP address or select it from the drop-down menu. To enable cloning prevention, select the check box called **Enable Hardware Association**, see "Hardware Association" on page 25. Make sure the **Use Shared Secret Registration** check box is enabled and click **Register.**

Note: If you get the message "Only VTE-initiated communication is possible ...", read "One-way Communication" on page 25 and choose accordingly.

5. In the **Select components to register** window, verify that you have selected the correct VTE type.

   Register your host by using a domain-specific shared secret created by the DSM administrator. Alternatively, use the fingerprint method that requires you to verify fingerprints displayed by this registration program with those shown in the DSM Dashboard (see "Determine the Installation Method" on page 26).
6. Click **Next**. You are prompted for the shared secret registration information:
   - **Shared secret**: Password for the domain to which the host, or host group, will be added. Contact the DSM administrator for this value.
   - **Domain name**: Name of the DSM domain to which the host will be added. Contact the DSM administrator for this value.
   - **Host group** (optional): Name of the host group to which the host will be added. Contact the DSM administrator for this value.
   - **Host description** (optional): Description of the host to be registered.

WARNING

Be sure to enter the case-sensitive shared secret, domain name, and host group correctly. If any of these are entered incorrectly, an error message displays. If you exceed the number of tries defined in the **Maximum Number of Login Tries** setting on the DSM **Password Preferences** page (System > General Preferences > Password), you are locked out of the system for a period define in the **User Lockout Time** setting.

7. Click **Next**. VTE prompts you to enter the host name of the Primary DSM. If necessary, ask the DSM administrator for the **Server name** on the dashboard of the DSM Management Console. Type the name of the DSM and click **Next**.
8. VTE prompts you to enter the protected host name. Enter the protected host name or IP address that the DSM Administrator entered in the DSM. If it is a name, it must be resolvable by the DNS server (see "Host Name Resolution" on page 24). You can type the name/IP address or select it from the drop-down menu. To enable cloning prevention, select the check box called **Enable Hardware Association**, see "Hardware Association" on page 25. Click **Register.**

Note: If you get the message "Only VTE-initiated communication is possible ...", read "One-way Communication" on page 25 and choose accordingly.

9. If successful, the *Register Host* window displays that the File System Component was successfully registered with a Management Console.
10. Verify the installation by checking VTE processes.
    a. In the system tray of the protected host, right-click the Vormetric icon.
    b. Select **Status**. Review the information in the **Vormetric Status** window to confirm that the correct VTE are installed and registered.

To Register VTE Using the Certificate Fingerprint Method

1. If the installer is not currently running, start the registration wizard by running `C:\Program Files\Vormetric\DataSecurityExpert\agent\shared\bin\register_host.exe`. Otherwise, click **Next** in the install wizard.
2. In the **Select components to register** window, verify that you have selected the correct VTE type.

   Register your host by using a domain-specific shared secret created by the DSM administrator. Alternatively, use the fingerprint method that requires you to verify fingerprints displayed by this registration program with those shown in the DSM Dashboard (see "Determine the Installation Method" on page 26).
3. Uncheck the **Use Shared Secret Registration** checkbox and click **Next**.
4. VTE prompts you to enter the host name of the Primary DSM. If necessary, ask the DSM Administrator for the **Server name** on the dashboard of the DSM Management Console. Type the name of the DSM and click **Next**.
5. VTE prompts you to enter the protected host name. Enter the protected FQDN host name or IP address, or select it from the drop-down menu, that the DSM Administrator entered in the DSM (if it is a name, it must be resolvable by the DSM server).
6. To enable hardware association (cloning prevention), select the check box called **Enable Hardware Association**, see "Hardware Association" on page 25. Click **Register.**

Note: If you get the message "Only agent-initiated communication is possible ...", read "One-way Communication" on page 25 and choose accordingly.

If successful, the *Register Host* window displays the fingerprint of the elliptic curve (EC) Certificate Authority (CA) certificate.

Note: If you get the error message *File System component service stopped 'Couldn't resolve host name',* it means the DSM host name could not be resolved by the protected host. See "Host Name Resolution" on page 24 to fix.

At this stage of the installation, you and the DSM Administrator must exchange information to confirm that protected host and DSM share valid certificates. This is done to verify that nobody is intercepting and modifying traffic between the DSM and VTE.

7. **VTE Installer**: Send the fingerprint to the DSM Administrator and wait for confirmation.
8. DSM **Security Admin**:
   a. Log on to the DSM **Management Console** and navigate to the domain where the host was added.
   b. Click the **Dashboard** tab.
   c. Match the fingerprint from VTE Installer with the EC CA fingerprint on the Dashboard.
   d. Advise the Host Admin of the results.
9. **VTE Installer**: If the fingerprints match, click **Yes**. A fingerprint for the protected host certificate displays.
10. Click **Ok**. Host has been successfully registered with the DSM. Click **Finish**. Restart the System.
11. Ask the DSM Administrator to select the **Communication Enabled** checkbox on the Management Console for the new protected host.
12. Verify the installation by checking VTE processes.
    a. In the system tray of the protected host, right-click the Vormetric icon.
    b. Select **Status**. Review the information in the **Vormetric Status** window to confirm that the correct VTE are installed and registered.

Windows Silent Install Using MSI or Self-Extracting .exe

Silent install refers to using the command line to install VTE in a non-interactive session. Use silent install to roll out VTE installations or upgrades to large numbers of hosts or to reduce your time and interaction as an administrator. Thales provides two types of installation binaries for silent installation:

- Windows Installer Package (MSI)
- Self-extracting `.exe`

Note: Thales supports installing or upgrading VTE agents with Microsoft System Center Configuration Manager (SCCM) using MSI installation binaries.

Windows Silent Install Command Line Options

The MSI and self-extracting `.exe` silent install methods require different command line options to specify a silent installation. However, the parameters specific to VTE installation are the same for both MSI and self-extracting `.exe` install methods. See the following tables for required and optional command line options.

General Format for Self-extracting .exe Silent Install Commands

For self-extracting .exe silent installations, be sure to escape the quotes surrounding the parameters passed in *REGISTERHOSTOPTS_Options* using a backslash (\) as shown:

Installation_executable `/s/v"/qn` **REGISTERHOSTOPTS=**`\"`*REGISTERHOSTOPTS_Options*`\""`

For example:

```
vee-fs-6.3.1.38-win64.exe /s /v"/qn REGISTERHOSTOPTS=\" SecSrv.demo.com -useip\""
```

General Format for MSI Silent Install Commands

Use the following syntax for MSI silent installations:

`msiexe.exe /i` *Installation_executable* `/qn` **REGISTERHOSTOPTS="***REGISTERHOSTOPTS_Options***"**

For example:

```
msiexec.exe /i vee-fs-6.3.1.38-win64.msi /qn REGISTERHOSTOPTS=" SecSrv.demo.com -
useip"
```

Command Line Options for MSI and .exe Silent Installations

Refer to the following tables for required and optional command line options for MSI and self-extracting `.exe` VTE silent installations. The following table lists the command line options for self-extracting `.exe` binary types.

Table 2-1: Command line options exclusive to self-extracting `.exe` silent install

Windows Attribute	Description	Required?
`/s`	Run the installation in silent mode.	Yes
`/v`	Pass command-line options and values of public properties through to the installer.	Yes

The following table lists the command line options for MSI binary types.

Table 2-2: Command line options exclusive to MSI silent install and uninstall

Windows Attribute	Description	Required?
/i	Enable installation/configuration	Yes, for installation
REINSTALLMODE=voums	Required REINSTALLMODE codes for upgrades	Yes, for upgrade
REINSTALL=ALL	Required REINSTALL setting for upgrades	Yes, for upgrade
/x	Required for uninstallation. See “To uninstall VTE silently” on page 39.	Yes, for uninstall

The following table lists the command line options common to both MSI and self-extracting .exe binary types.

Table 2-3: Common options for both MSI and self-extracting `.exe` silent installs

Windows Attribute	Description	Required?
`/qn`	Run the command in non-interactive mode (no GUI).	Yes
`REGISTERHOSTOPTS`	Options for registering VTE with the DSM. The full set of options is specified below.	No

Windows Attribute	Description	Required?
`INSTALLDIR`	Specifies an alternate location for installing VTE. NOTE: If syntax is incorrect, silent install will fail. See examples. NOTE: The alternate location must be on the system drive.	No
`REBOOT=ReallySuppress`	By default, the machine reboots after installation. Supplying this option prevents that. However, a reboot is still required to fully install VTE.	No

The following table lists the REGISTERHOSTOPTS parameters that determine log settings and VTE configuration settings. The REGISTERHOSTOPTS parameters are common to both MSI and self-extracting .exe binary types.

Table 2-4: REGISTERHOSTOPTS options for both MSI and self-extracting .exe silent install

Windows Attribute	Description	Required?
DSM host name	FQDN of the DSM. Example: `thl.example.com`	Yes
`-useip`	Use the IP address of the protected host instead of host name. Used when `-agent` is not supplied.	No
`-agent=`*your.agent.name.com*	FQDN of protected host.	No
`-onewaycomms`	Set when VTE-initiated-only communication is required. See “One-way Communication” on page 25	No
`-usehwsig`	Sets when you want to associate this installation with the machine hardware for cloning prevention. See “Hardware Association” on page 25	No
`-log=`*filename*	Sets a log file to output results to.	No
`-port=`*port*	Specifies the port number.	No
`-secret=`*password*	Specifies the password for a shared secret registration. See “Determine your VTE Registration Method” on page 24	No
`-domain=`**domain_name**	Specifies domain for the shared secret.	No
`-hostgroup=`*hostgroup_name*	Specifies the optional host group for the shared secret.	No
`-description="`*description*`"`	Specifies a description for the protected host with a Shared Secret registration, however, it does not overwrite an existing description. It does not work for the Fingerprint method, NOTE: If syntax is incorrect, silent install fails. See examples.	No
`-log=%temp%\vor-agent-reg.log`	Specifies a log file particular to the registration. Useful for documenting why the registration failed (if it does).	No
`-enableldt`	Automatically enable and register LDT during silent install.	No

Windows Attribute	Description	Required?
`-enablees`	Automatically enable and register Efficient Storage during silent install	No

Windows Silent Install Examples

These examples fall into two categories: installations using the *Shared Secret* registration method and installations using the *Certificate Fingerprint* method (see "Determine your VTE Registration Method" on page 24). The Fingerprint method requires that you enter the host name, or IP address, in the DSM.

Silent Install Using the Shared Secret Registration Method

The Shared Secret method requires that the DSM administrator create a shared secret password for a domain/host group and share the password with you. Collect the following information from the DSM administrator:

Table 2-5: Shared Secret Information for VTE Registration

Shared secret password	
Domain assigned to host in the DSM for the shared secret password	
Host group for the shared secret password (if created)	
Description for protected host (optional)	
Shared secret validity period (registration must be completed in this period)	

Note: If either the `-description` or `-INSTALLDIR` syntax is incorrect, silent install fails and you must correct the syntax and reinstall. See examples for correct syntax.

Example 1: Default install without description and using 2-way communications. Installs VTE using the Shared Secret method, specifies the DSM named `thl.example.com`, host uses IP address to register (`-useip` option), supports hardware association for cloning prevention (`-usehwsi` option), specifies a secret key for the DSM domain `domain1`. This method also automatically registers LDT (`-enableldt` option) and Efficient Storage (`-enablees` option) during install.

Self-extracting .exe Example

```
vee-fs-6.3.1.38-win64.exe /s /v" /qn registerhostopts=\"thl.example.com -useip
-enableldt -enablees -usehwsig -secret=MaCarena45# -domain=domain1 \""
```

MSI Example

```
msiexec.exe /i vee-fs-6.3.1.38-win64.msi /qn registerhostopts="thl.example.com
-useip -enableldt -enablees -usehwsig -secret=MaCarena45#  -domain=domain1"
```

Example 2: Installs VTE using the Shared Secret method, specifies the DSM named `thl.example.com`, host uses IP address to register, supports 1-way communications, specifies a shared secret for the DSM domain `domain1` with custom installation directory.

Self-extracting .exe Example

```
vee-fs-6.3.1.38-win64.exe /s /v" /qn INSTALLDIR=\"c:\opt\vormetric2\"
registerhostopts=\"thl.example.com -useip -onewaycomms -usehwsig
-secret=MaCarena45# -domain=domain1 -description=\"\"Silent Install\"\"\""
```

MSI Example

```
msiexec.exe /i vee-fs-6.3.1.38-win64.msi /qn registerhostopts=" thl.example.com
-useip -onewaycomms -usehwsig -secret=MaCarena45#
-domain=domain1 -description=Silent Install"
```

Example 3: Installs VTE using the Shared Secret method, specifies the DSM named `thl.example.com`, host uses IP address to register, supports hardware association (cloning prevention), specifies a secret key for the DSM domain `domain1`, installs in default directory, and has a host description “Silent Install”.

Self-extracting .exe Example

```
vee-fs-6.3.1.38-win64.exe /s /v" /qn registerhostopts=\"thl.example.com -useip
-usehwsig -secret=MaCarena45# -domain=domain1
-description=\"\"Silent Install\"\"\""
```

MSI Example

```
msiexec.exe /i vee-fs-6.3.1.38-win64.msi /qn registerhostopts=" thl.example.com
-useip  -usehwsig -secret=MaCarena45#
-domain=domain1 -description=Silent Install"
```

Silent Install Using the Fingerprint Registration Method

The Fingerprint Registration method requires that the DSM Administrator add all hosts on which you will install VTE to the DSM with the Registration Allowed and Communication Enabled features enabled. Once those are added, you may proceed with the silent install.

Example: Installs VTE by FQDN host name.

Self-extracting .exe Example

```
vee-fs-6.3.1.38-win644.exe /s /v" /qn registerhostopts=\"30181.example.com
-onewaycomms \""
```

MSI Example

```
msiexec.exe /i vee-fs-6.3.1.38-win64.msi /qn registerhostopts=" 30181.example.com
-onewaycomms"
```

Example 2: Installs VTE using Fingerprint method, specifies the DSM named `thl.example.com`, host uses IP address to register, supports hardware association (cloning prevention), and installs in a custom directory.

Self-extracting .exe Example

```
vee-fs-6.3.1.38-win64.exe /s /v" /qn INSTALLDIR=\"c:\abc\vormetric2\"
registerhostopts=\"thl.example.com -useip -usehwsig \""
```

MSI Example

```
msiexec.exe /i vee-fs-6.3.1.38-win64.msi /qn registerhostopts=" thl.example.com
-useip  -usehwsig"
```

To Install in Windows Silently

Follow this procedure to manually use silent install. Refer to Microsoft documentation for silent install using SCCM.

Note: Before performing a silent install, disable the firewall because the firewall may block the installation. After installation, enable the firewall.

1. Either gather the appropriate Shared Secret information (for Shared Secret Registration) or have the DSM administrator add the protected host name or IP address to the DSM (Fingerprint registration method).
2. Add the target host to DSM and enable communication for silent mode
3. Log on to the protected host as an administrator and run the silent install command on each host from the Windows command shell.
4. After running the silent install command, the system reboots.
5. Repeat for additional silent installs.

To Upgrade in Windows Silently

If you have already installed VTE on a Windows computer and want to upgrade it silently, use the appropriate command below for the type of installation binary that you are using.

Note: The protected host must be able to connect to the DSM that it is registered to or the upgrade will fail.

Upgrade using self-extracting .exe

```
vee-fs-6.3.1.38-win64.exe /s /v" /qn"
```

Upgrade using MSI

```
msiexec.exe /i vee-fs-6.3.1.38-win64.msi /qn REINSTALLMODE=voums REINSTALL=ALL
```

Before upgrading using MSI, you must rename the installation file to the name of the previously used MSI installation file. See “Workaround for MSI VTE Typical, Silent, and Scheduled Upgrades” on page 37 for more information.

Verify the Windows Installation

After running a silent install, verify the installation by checking VTE processes.

1. In the system tray, right-click the Vormetric icon.
1. Select **Status**. Review the information in the Vormetric Status window to confirm the correct VTE are installed and registered.

Resolving Problems that Prevent Silent Install

If you encounter problems using MSI or self-extracting `.exe` silent install commands, first check the syntax of the command. To further investigate installation issues, you can use Microsoft diagnostics software:

- For desktop versions of Windows: Microsoft Diagnostics Troubleshooting Wizard
- For server versions of Windows: Microsoft Automatic Troubleshooting Services (MATS)

Refer to the Microsoft documentation on the linked pages for more information. See the Vormetric Transparent Encryption Agent Compatibility Matrix for a list of versions of Windows that are supported for use with VTE.

VTE Scheduled Upgrade

Scheduled upgrade allows you schedule an upgrade of the VTE agent to occur after the next time the server hosting the agent reboots normally. Scheduled upgrade can minimize VTE service interruptions. Also, scheduled upgrade can reduce coordination issues in organizations where the security roles are separated.

Note: Install all Microsoft update patches before scheduling a VTE agent upgrade. Otherwise, the upgrade will fail.

Note: The VTE scheduled upgrade feature is compatible with Windows Server 2008 R2 and higher versions.

Note: For all types of upgrades, including interactive (GUI-based) and scheduled upgrades, the protected host must be able to connect to the DSM that it is registered to or the upgrade will fail.

Scheduling a VTE Upgrade on the Command Line

To schedule VTE to upgrade the next time the system reboots, type:

```
> voradmin upgrade schedule <VTE setup executable path>
```

Self-extracting .exe Example

```
> voradmin upgrade schedule C:\6.0.3.15\vee-fs-6.3.1.38-win64.exe
```

MSI Example

```
> voradmin upgrade schedule C:\6.3.1.38\vee-fs-6.3.1.38-win64.msi
```

System Response

```
Creating and installing service to upgrade. VTE agent will be upgraded on next
reboot.
```

WARNING

If you have scheduled an upgrade on reboot and the system crashes or is not shutdown gracefully, you must restart the system again to upgrade the agent.

Scheduling a VTE Upgrade Interactively (self-extracting .exe only)

When you open the self-extracting `.exe` VTE installation binary and a version of VTE is already installed, you have the option of upgrading immediately or initiating a scheduled upgrade. See the procedure below for details. This is an alternative to scheduling an update on the command line using `voradmin` (see "Scheduling a VTE Upgrade on the Command Line" on page 36).

1. Move the self extracting `.exe` VTE installation binary to the computer on which you want to initiate the scheduled upgrade.
2. Double-click the self extracting `.exe` VTE installation binary to run it.
3. Click through the standard initial dialog boxes like the license dialog box.
4. On the **UPGRADE - Install now or Later** dialog box, click **Schedule Upgrade on next reboot** and then click **Next**.
5. Click **Schedule** on the confirmation dialog box.

   The next time the computer reboots, the upgrade will occur.

This interactive method of scheduling an update is not available for MSI VTE installation binaries. You must use the `voradmin` command line scheduled upgrade method.

Show Scheduled VTE Upgrades

To display all scheduled VTE agent upgrades, type:

```
> voradmin upgrade show
```

System Response

```
Current version:          6.0.3.12
Target upgrade version:   6.0.3.15
Upgrade on reboot:        Enabled
```

Cancel a Scheduled VTE Upgrade

To cancel/cleanup a scheduled VTE agent upgrade, type:

```
> voradmin upgrade cancel
```

System Response

```
VTE agent upgrade canceled successfully.
```

Workaround for MSI VTE Typical, Silent, and Scheduled Upgrades

When performing an upgrade, Windows Installer Package (MSI) expects the name of the installation binary to be the same as the binary that you used to install VTE. Because Thales includes the software version and build number in the binary file name, you must rename the installation binary to the name of the previously used MSI

installation binary before upgrading using MSI. This applies to any MSI VTE upgrade method: typical (interactive), silent upgrade, and scheduled upgrade. If the name does not match the previous binary file name, the upgrade will fail with error code 1316.

In the following hypothetical MSI VTE installation binary, the software version is 6.3.1 and the GA build number is 38:

```
vee-fs-6.3.1.38-win64.msi
```

If you used this binary to install or upgrade VTE, the next time you want to upgrade VTE the installation binary that you download from Thales eSecurity might have the following file name:

```
vee-fs-7.0.0-120-win64.msi
```

To upgrade successfully using MSI, you would need to rename the new installation binary to the previous file name of `vee-fs-6.3.1.38-win64.msi` before upgrading.

Finding The Name Used For A Previous MSI Installation or Upgrade

If you want to upgrade VTE using an MSI installation binary but don't know the file name that you used during the previous installation or upgrade, you can look it up by using one of the following methods on the computer where you installed VTE:

MSI File Name Lookup Method 1: PowerShell

Run the following command in PowerShell:

```
PS> (Get-WmiObject Win32_Product | where { $_.Name -match "Vormetric File System
Agent" }).PackageName
```

Rename the new VTE setup installation binary to the file name output from the PowerShell command and proceed with the upgrade.

MSI File Name Lookup Method 2: Windows Registry

Find the following registry entry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Installer\Products\905CB36BF7940894995701C8690
1D14F
```

Rename the new VTE installation binary to the file name in the registry and proceed with the upgrade.

Uninstall VTE from a Windows Host

CAUTION

After removing VTE software, encrypted data remains encrypted. If decrypted or copied out of the host, the data is visible as clear text.

Before removing VTE from a Windows host, verify the following:

1. All applications protected by hosts are stopped.
2. The DSM Administrator has evaluated the current hosts in the *Guard FS* tab to avoid data loss or compromise.

3. The DSM Administrator removed **System Locked** and **FS Agent Locked** settings for this host (if set).
4. All hosts have been removed.
5. VTE is removed from the host before the host is removed from the DSM.
6. Any data you want to use after uninstall is decrypted.

Note: If VTE installation fails because a host is in use, determine which applications are using the hosts and stop them. Then run the uninstall again.

To uninstall VTE interactively

Note: VTE for Windows must be removed from the host before the host is removed from the DSM **Hosts** pane.

If multiple VTE types are installed on the Windows host, they must be removed or uninstalled separately. Remove all non-VTE items before VTE.

1. Stop any application accessing files in the host.
2. Log on to the host that is running VTE as the system administrator.
3. Run the Windows **Add** or **Remove Programs** utility to remove VTE software.
4. Click **Yes** to reboot the host.

To uninstall VTE silently

You can use the command line to uninstall VTE in a non-interactive session or silent uninstall. For more information about command line installation on Windows, see "Windows Silent Install Using MSI or Self-Extracting .exe" on page 30.

Prepare your environment for uninstallation as described in "Uninstall VTE from a Windows Host" on page 38. For more information about supported command line options, see "Windows Silent Install Command Line Options" on page 30.

Note: Do not attempt to uninstall VTE silently using the MSI method if you initially installed VTE with the self-extracting `.exe` method. See "To uninstall VTE interactively" on page 39 to use the interactive uninstallation instead.

If you have installed VTE using the MSI installation binary, you can uninstall it silently using the following MSI command:

`msiexe.exe /x` *`Installation_executable`* `/qn`

For example: `msiexec.exe /x vee-fs-6.3.1.38-win64.msi /qn`

Chapter 3: Installing VTE for Linux

This chapter describes how to install and configure VTE on Linux systems. It contains the following sections:

Installation Overview

The installation and configuration process consists of three basic steps:

1. Making sure the host system meets the requirements described in "Installation Prerequisites" on page 41.
2. Installing VTE on the host.
3. Adding the host fully qualified domain name (FQDN) or IP address to the DSM. The DSM Administrator can perform this task manually, using the Fingerprint registration method, or automatically, using the Shared Secret registration method.
4. Registering the host with the DSM so they can communicate.

Installation Prerequisites

This section lists the tasks you must complete, and the information you must obtain, before installing VTE:

Recommendations and Considerations

- The host on which you want to install VTE *must* support AES-NI hardware encryption. If it does not, any attempt to install or upgrade VTE to release 6.3.0 or later will fail.
- Vormetric recommends that you install VTE in the default location.
- Do not install VTE on network-mounted volumes such as NFS.
- If you want to install VTE on a host that uses UEFI Secure Boot, you must download the Thales public certificate and add that certificate to the MOK (Machine Owner Key) list on the host. For details, see "VTE Agent Installation with UEFI Secure Boot" on page 44.
- 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.

Example:
```
# ./vee-fs-6.3.1-45-sels12-x86_64.bin -y -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 DSM(s) to all hosts where you install VTE.
- If the host is a virtual machine, the VM must be deployed and running.

Host Name Resolution

Host name resolution maps host names to an IP address. During this configuration process, enter either the FQDNs, or IP addresses, of your DSMs and hosts. If you use FQDNs, your hosts must be able to resolve their DSM host names, and the DSMs must be able to resolve to their hosts.

Note: The exception is if you approve of only VTE-initiated communication between the DSM and the Host. See "Random Number Generation Method" on page 46 for more information.

Setting the host name with DNS

The Domain Name Service (DNS) is the preferred method of host name resolution. If you use DNS, use the DSM and host name FQDNs for the installation and configuration procedures in this chapter.

Setting the host name without DNS

If you do not use a DNS server, perform one of the following tasks on all of the DSMs and the hosts:

- Request that the DSM administrator add an entry in the `/etc/hosts` file on the DSMs for each Host. The administrator must use the DSM Admin CLI, and add an entry to *each* DSM in an HA deployment because entries in the `/etc/hosts` file are not replicated across DSMs.
- Use the IP addresses of the DSMs and hosts.

Port configuration

If a protected host must communicate with a DSM through a firewall, open the ports in the firewall as shown in the following figure.

Note: See the following table to determine which of the above ports must be opened through the firewall.

The default port for communication between the DSM and the agent is 7025. If this port is already in use, set the port to a different number by specifying the new port during agent installation.

Table 3-1: Ports to Configure

Port	Protocol	Communication Direction	Purpose
	ICMP	All ICMP	Used for Ping
22	TCP	Management Console → DSM	CLI SSH Access
161	TCP/UDP	SNMP Manager →DSM	SNMP queries from an external manager
443	TCP	Browser → DSM DSM ↔ DSM Agent → DSM	Redirects to either port 8445 or 8448 depending on the security mode. (8445 is used in compatible & RSA modes; 8448 is used in Suite B mode, for secure communication between DSMs in an HA cluster and for LDT registration.)
1792	TCP	DSM ↔ network HSM	DSM communication with Luna HSM.
5432	TCP	DSM (HA node 1) ↔ DSM (HA node n)	HA information exchange.
5696	TCP	KMIP client → DSM	Allows communication between the KMIP client and DSMs
7025	TCP/UDP	DSM↔ DSM	Uses SNMP to get HA node response time.
8080	TCP	Agent → DSM DSM↔ DSM	Port 8080 is no longer used for registration, but you can manually close/open this legacy port for new deployment, for backward compatibility if you use previous versions of the agent and need to register to 8080. Default is on (open). Syntax `0001:system$ security legacyregistration [ on | off | show ]`
8443	TCP	Agent → DSM	RSA TCP/IP port through which the agent communicates with the DSM, in case 8446 is blocked. The agent establishes a secure connection to the DSM, through certificate exchange, using this port.
8444	TCP	Agent → DSM	RSA port via which the Agent log messages are uploaded to DSM, in case 8447 is blocked.
8445	TCP	Browser → DSM DSM ↔ DSM (fall back)	Management Console, VMSSC, and fall back for HA communication in case port 8448 is dropped.
8446	TCP	Agent → DSM	Configuration Exchange using Elliptic Curve Cryptography (Suite B)
8447	TCP	Agent → DSM	Agent uploads log messages to DSM using Elliptic Curve Cryptography (ECC) and RSA
8448	TCP	Browser → DSM DSM ↔ DSM Agent → DSM	GUI Management during enhanced security using Elliptic Curve Cryptography (Suite B). Also for secure communication between DSMs in an HA cluster. Also used for communication between host with LDT host and DSM during Agent registration.
8449	TCP	Smart Card → DSM	Smart card used with RSA mode

Table 3-1: Ports to Configure

Port	Protocol	Communication Direction	Purpose
8450	TCP	Smart Card → DSM	Smart card used with ECC/compatible mode
9004	TCP	DSM ↔ network HSM	DSM communication with nShield Connect and its associated RFS
9005	TCP	DSM ↔ remote admin	Used by Remote Administration Service process to accept connections from the Remote Administration Client.

In the following example, the default communication port between the DSM and agent was changed from 7025 to 8000:

```
Please enter the host name of this machine, or select from the following
list. The name you provide must precisely match the name used on the "Add
Host" page of the Management Console.
[1] centos-6-0
[2] 192.168.1.160
Enter a number, or type a different host name or IP address in manually:
What is the name of this machine? [1]: centos-6-0:8000
You entered the host name "centos-6-0:8000"
Is this host name correct? (Y/N) [Y]:
```

Port Usage in One Way Communications Mode

By default, polling from the agent host to the DSM when running in one-way communications mode uses HTTP via port 8080. If the agent is configured to use secure polling, then polling is performed using HTTPS via port 8448 (in suite B mode) or port 8445.

VTE Agent Installation with UEFI Secure Boot

If you want to install the VTE 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-7.0.0-47-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
```

Installation and Registration Options

VTE 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

There are two methods for installing VTE on Linux platforms:

- **Typical:** Most common and recommended type of installation. Use this method for installing VTE on hosts concurrently. See "Typical Install" on page 50.
- **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 Install" on page 55.

If you are upgrading, not installing, VTE, you can configure the upgrade to occur at system boot time. See "Scheduled Upgrade Feature" on page 62.

VTE Registration Method

You can register protected hosts with the DSM using either the *Fingerprint method* or the default *Shared Secret method*, and you will be prompted to choose a method by the VTE installer. While you can install VTE without registering it with the DSM, you cannot use any VTE functions until VTE is registered.

- **Fingerprint method**: Requires the DSM administrator to add the FQDN or IP address of each host to the DSM before registering VTE.

  During the registration, the DSM generates the certificate and passes it down to the VTE with the fingerprint. The security administrator must verify the fingerprint to make sure the certificate is valid.
- **Shared Secret method**: Requires the DSM administrator to create a *shared secret* registration password—a case-sensitive string of characters—for auto-registering a host in a domain or host group.

  VTE installers use the shared secret to add and register hosts to the DSM for a domain or host group. This method can automatically add host names or IP addresses to the DSM. This eliminates the need to verify that the host and DSM share valid certificates. You can add multiple hosts dynamically, during VTE installation and registration, with a single shared secret password.

  If you choose the Shared Secret method, ask the DSM Administrator to create a shared secret for the domain, or host group, in which the new host will reside. Then, obtain the shared secret and the validity period (one hour, day, week, or month) and register within that period.

Note: In the DSM Registration Shared Secret page (Hosts > Registration Shared Secret) there is an option: **Require that hosts are first added**. If you select this option, you must manually add the hosts to the DSM first.

Random Number Generation Method

On Linux systems, you can choose from two methods for generating the random number used to create a certificate:

- `/dev/urandom`: This method is recommended. It is efficient and provides an acceptable level of security.
- `/dev/random`: This method provides more security but it can lengthen installation times and decrease startup speeds.

One-way Communication Option

In some deployments, VTE might not be visible to the DSM through the normal network communications. For example, when the host that contains VTE:

- uses NAT so its IP address will change.
- is protected behind a firewall.
- is not permanently connected to a communication channel on the DSM.
- is unable to resolve the host name to an IP address.

In these situations, VTE can initiate VTE-only communication to the DSM. This feature is called one-way communication and works by making VTE poll the DSM for any policy messages or changes. Then it downloads the required changes.

The downside of one-way communication is that the DSM cannot issue any queries to VTE. For example, the DSM Administer cannot browse host directories or User IDs.

VTE Install Checklist

Use the following table to verify prerequisites and collect the information you need for the installation.

Checklist item	Status

Obtain VTE installation image from Vormetric. The format for VTE file names is:
`vee-<agent_type-build-system>.bin`
Example:
`vee-fs-6.3.1-45-sels12-x86_64.bin`	

Fully Qualified Domain Name (FQDN) of the DSM	
IP address or FQDN of the host	
Administrator password for the host	
If using Shared Secret registration, obtain the following from the DSM administrator: 1) Shared secret password 2) Domain 3) Host group, if applicable 4) (Optional) Description of the host	
If using the Fingerprint registration, ask the DSM Administrator to add the host to the DSM and check the **Registration Allowed** check box. After checking the fingerprint, select the **Communication Enabled** check box.	
Resolved “Host Name Resolution” on page 42 for the hosts and DSM	
Set “Port configuration” on page 42	

Checklist item	Status
Select your installation method and registration options as described in "Installation and Registration Options" on page 46	
Synchronize the host clock to the DSM clock.	
Set the network subnet mask on the host (unless you are using one-way communication)	
Preferred DNS Server (if using FQDNs):	

Determining Kernel Compatibility With VTE

As new Red Hat, SUSE, and Ubuntu Linux kernels are published, Thales regularly publishes new patch versions of VTE that are compatible with selected versions of these new kernels. Thales recommends that you check that the kernel on which you want to install VTE is verified to be compatible with VTE. VTE may appear to run correctly on an unsupported kernel. However, Thales strongly advises against running VTE on an unsupported kernel due to potential stability issues.

The following table describes several ways to verify that a kernel version is compatible with the version of VTE that you plan to install or are running. For more information, see the following sections.

Table 3-2: Methods to check kernel compatibility with VTE

Kernel compatibility check method	How to check	Description
Vormetric Transparent Encryption Agent Compatibility Matrix document	Download from Thales support site	Includes kernel compatibility and other types of compatibility such as file system, database, antivirus, and DSM version compatibility.
supported_kernel_list file	Extract this text file from VTE installation binary	This text file contains the compatible kernels for the distribution and OS version of the VTE installation binary from which it has been extracted. The standard format of the text file is intended to be parsable by orchestration software. See "Extracting a List of Supported Kernels From a VTE Install Binary" on page 49
Automated kernel compatibility check during installation or upgrade	Kernel compatibility is checked during installation or upgrade	If an incompatible kernel is detected, a message is printed to the screen and logged to `syslog`. See "Automatic Checks for Supported Kernels" on page 50
Automated run-time kernel compatibility check	Kernel compatibility is checked when the VTE services start	If an incompatible kernel is detected, a message is logged to `syslog`. See "Automatic Checks for Supported Kernels" on page 50

Extracting a List of Supported Kernels From a VTE Install Binary

As of VTE version 6.2.0, you can extract a text file from a Linux installation binary that lists the supported kernels for the VTE version of that installation binary. As a hypothetical example, the text file `supported_kernel_list` in `vee-fs-6.3.1-45-sels12-x86_64.bin` contains a list of the Red Hat 8 kernels that are compatible with version VTE version 6.3.1.

Although you can view it in a text editor, the standard format of the `supported_kernel_list` text file is also intended to be parsable by orchestration software. The file format is readily apparent upon viewing the extracted file. The `supported_kernel_list` file only contains kernel compatibility information. For other types of compatibility information, such as file system, database, antivirus, and DSM version, see the *Vormetric Transparent Encryption Agent Compatibility Matrix* document available from the Thales support site.

The `supported_kernel_list` file has the following MIME type:

```
text/plain; charset=us-ascii
```

See the next section for the steps to extract the file from an installation binary without installing VTE. Once VTE is installed, the `supported_kernel_list` file is placed in <*install_path*>`/agent/vmd/supported_kernel_list`. For default installations, the path is:

```
/opt/vormetric/DataSecurityExpert/agent/vmd/supported_kernel_list
```

Extracting the supported_kernel_list Text File

Follow these steps to extract the `supported_kernel_list` text file from a VTE installation binary without running the installation.

1. Download the installation binary for your operating system and OS version from the Software Download Portal, such as `vee-fs-6.3.1-45-sels12-x86_64.bin` for Red Hat 8 systems.
2. Change to the directory where you stored the downloaded file. For example:

   ```
   cd ~/downloads
   ```

3. Use the `-e` option to extract the `supported_kernel_list` file from the installation binary. For example:

   ```
   ./vee-fs-6.3.1-45-sels12-x86_64.bin -e
   ```

   This file extraction can be done as a normal user (non-root), assuming the user account has write permission in the current working directory.
4. The `supported_kernel_list` file is extracted from the installation binary. Other contents of the installation binary are also extracted. These extra files can be deleted or ignored. To list the extracted files:

   ```
   # ls
   installer    manifest.txt           vee-fs-6.3.1-45-sels12-x86_64.bin
   license.txt  supported_kernel_list  vee-fs-6.3.1-45-sels12-x86_64.rpm
   ```

5. Open the `supported_kernel_list` file with a text editor or deploy it to your orchestration software.

If the kernel is compatible and you want to proceed with installing or upgrading VTE, start with the downloaded installation binary rather than the package file that is extracted along with the `supported_kernel_list` file. For more information about installating or upgrading VTE, see one of the following sections in this chapter:

- “Typical Install” on page 50
- “Silent Install” on page 55
- “Upgrading VTE” on page 61

- “Scheduled Upgrade Feature” on page 62

Automatic Checks for Supported Kernels

As of VTE version 6.1.3, when installing or upgrading VTE, the installer checks the kernel on the host for compatibility with the version of VTE that you are installing or upgrading. If the kernel is not compatible, the installer prints a warning on the screen and logs a message to `syslog`. Thales eSecurity recommends that you exit installation if you see this error. However, the installer does not prevent you from installing VTE on an unsupported kernel. The message printed on the console during an attended installation when an incompatible kernel is detected looks similar to the following:

```
Do you accept this license agreement? (Y/N) [N]: Y
WARNING: Linux Kernel 3.10.0-957.el7.x86_64 is not supported with the VTE version
6.1.3.12 being installed.
Do you want to proceed? (Y/N) [N]:
```

For silent (unattended) installations or if you are using orchestration tools like Chef or Puppet, the installer will proceed with the installation even if the running kernel is unsupported. The installer will log a warning to `syslog` that the kernel is unsupported.

The check for supported kernels is also done whenever the VTE services start up, just before loading the VTE kernel modules. If the currently running kernel is unsupported, a warning message is logged to `syslog` and the startup of VTE continues.

Checking for supported kernels is an advisory feature. Thales eSecurity does not recommend that you use VTE on an unsupported kernel, but if you ignore the warning, VTE installation and startup of services proceeds as if the kernel were supported. In some cases, the VTE kernel module loading might fail or encounter some unexpected error. If this occurs, you are advised to run VTE on supported kernels. To check for supported kernels before installing VTE, see the options in “Determining Kernel Compatibility With VTE” on page 48.

Typical Install

This section describes the typical install and registration process of VTE on a Linux system.

Typically, you register VTE with the DSM as part of the installation process. However, you can postpone registration if you have a specific plan to register VTE later.

The data on the host is not protected until you complete the configuration of the host. Communication to the DSM (and retrieval of any policies and keys) cannot happen until you register VTE on the DSM, and enable communication between VTE and the DSM.

Note: Do not install VTE on network-mounted volumes like NFS.

Before You Begin

Verify that the DSM administrator has added all of the hosts that you need for VTE installation to the DSM with the following functionality enabled:

- Registration Allowed
- Communication Enabled

Note: If registration appears to freeze, verify that the DSM and VTE can communicate with each other over the network.

Note: If you are installing VTE using the shared secret method, you do not need to add the hosts to the DSM before installation.

Installation

1. Log on to the host where you will install VTE.

Note: You must have 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 VTE. A typical installation and registration, in the non-verbose mode, uses the following syntax:

```
# ./vee-<product>-<version>-<build-system>.bin
```

Example:

```
# ./vee-fs-6.3.1-45-sels12-x86_64.bin
```

4. The Vormetric License Agreement displays. Enter '**Y**' and **Enter** to accept.

Note: VTE is installed on the host, but not yet registered. The Vormetric Encryption Expert File System prompts you to register.

5. Enter the host name of the machine running the Security Server (the host name displays on the Dashboard window of the Management Console)

Note: If you are using the Fingerprint method, VTE's host machine must be pre-configured on the DSM with the **Reg. Allowed** option enabled.

6. Enter '**Y**' and **Enter** to accept.
7. During host registration, you can automatically register and enable LDT. The installation script prompts you to enable the automatic registration.
8. Select one of the following registration options:
   - Register VTE using the *Certificate Fingerprint* method. See "To register VTE using the Certificate Fingerprint method:" on page 52.
   - Register VTE now using the *Shared Secret* method. See "To register VTE for Linux using the Shared Secret Registration method:" on page 53.
   - Register VTE later by entering N.

Note: Use the command `register_host` at `/opt/vormetric/DataSecurityExpert/agent/vmd/bin/` to register without the installation program.

Register Using the Certificate Fingerprint

To register VTE using the Certificate Fingerprint method:

1. Enter Y when you see the following prompt:

```
Do you want to continue with agent registration? (Y/N) [Y]: Y
Please enter the primary Security Server host name:
```

2. Enter the DSM FQDN and then `Y`. Ask the DSM Administrator to obtain the FQDN from the dashboard of the DSM Management Console.

   Example:

```
thl1490.i.vormetric.com
You entered the host name thl1490.i.vormetric.com
Is this host name correct? (Y/N) [Y]: Y
```

3. Enter the host name when prompted:

```
Please enter the host name of this machine, or select from the following
list. If using the "fingerprint" registration method, the name you provide
must precisely match the name used on the "Add Host" page of the Management
Console.

[1] host14.i.example.com
[2] Host-RHEL-14.i.example.com
[3] 10.3.14.90
[4] 192.168.122.

Enter a number, or type a different host name or IP address in manually:
What is the name of this machine? [1]: 1
```

4. Enter the host name. This must match the name used on the **Add Host** page of the Management Console. Once completed, the installation prompts you for the registration method:

```
You selected "host14.i.example.com".
Would you like to register to the Security Server using a registration shared secret
(S) or using fingerprints (F)? (S/F) [S]: F
```

5. Enter **F** (fingerprints). At the prompt, select **Y** to enable hardware association (see “Random Number Generation Method” on page 46).

```
It is possible to associate this installation with the hardware of this
machine. If selected, the agent will not contact the DSM (GDE Appliance) 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]:
```

6. Enter **Y** or press **Enter** to enable. The following prompt displays:

```
Do you want to configure the agent to use /dev/random for the source of random
numbers? Doing so has security benefits but could cause significant delays during
installation and startup. The default behavior (answering No) is to use /dev/
urandom, which has no associated delay.

Would you like to use /dev/random for random numbers? (Y/N) [N]:
```

```
Generating certificate...done.
Signing certificate...done.
```

7. Enter **N**. If everything is working, the install program generates certificate signing requests and lists the fingerprint of the EC (elliptic curve) CA (Certificate Authority) certificate:

```
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.

A5:6D:4B:DE:1C:ED:F7:E5:8C:C7:F3:21:58:31:F2:27:15:C5:8C:C9

Do the fingerprints match? (Y/N) [N]:
```

Note: If you see the error message *File System component service stopped 'Couldn't resolve host name',* it means that the DSM could not resolve the host name. See the *Vormetric* DSM Guide for information.

8. This fingerprint must match the certificate on the DSM dashboard. This verifies that nobody is intercepting and modifying traffic between the DSM and VTE. Verify this match with the DSM Administrator, then enter **Y**. VTE fingerprint for the host displays:

```
The following is the fingerprint for this agent on this host.
Please verify that it matches the fingerprint shown for this host on the
Edit Host window of the Management Console.

01:FE:F9:37:93:36:F7:74:DD:D5:5D:EA:C8:4A:9B:9C:D0:58:73:8C

Successfully registered the Vormetric Encryption Expert File System Agent with the
primary Vormetric Data Security Server on thl1490.i.vormetric.com.
```

9. Verify with the DSM Administrator that VTE fingerprint matches with the fingerprint shown for this host on the (Hosts > Hostname) **Edit Host** window of the Management Console. VTE is installed and registered.
10. Verify the installation by checking VTE processes on the host:
    a. Run `vmd -v` to check the version of VTE.
    b. Run `vmsec status` to display VTE processes.
    c. Look at the log files in `/var/log/vormetric/install.fs.log.<date>`, especially `vorvmd_root.log`.

Register with Shared Secret

To register VTE for Linux using the Shared Secret Registration method:

1. Verify that the DSM Administrator created a shared secret for the domain or host group in which the new host resides.
2. Enter Y when you see the following prompt:

```
Do you want to continue with agent registration? (Y/N) [Y]: Y
Please enter the primary Security Server host name:
```

3. Enter the DSM FQDN and then `Y`. Ask the DSM Administrator to obtain the FQDN from the dashboard of the DSM Management Console.

   Example: `thl1490.i.vormetric.com`

```
You entered the host name thl1490.i.vormetric.com
Is this host name correct? (Y/N) [Y]: Y
```

4. Enter the host name when prompted:

```
Please enter the host name of this machine, or select from the following
list. If using the "fingerprint" registration method, the name you provide must
precisely match the name used on the "Add Host" page of the Management Console.

[1] host14.i.example.com
[2] Host-RHEL-14.i.example.com
[3] 10.3.14.90
[4] 192.168.122.

Enter a number, or type a different host name or IP address in manually:
What is the name of this machine? [1]: 1
```

5. Enter the host name. This host name must match the name used on the **Add Host** page of the Management Console (adding the host name is not needed for the shared secret method). The installation prompts you for the registration method:

```
You selected "host1490.i.vormetric.com".
Would you like to register to the Security Server using a registration shared secret
(S) or using fingerprints (F)? (S/F) [S]: S
```

6. Enter S (Shared Secret). You are prompted for the following information (examples are in italics—use your own system information):

```
What is the registration shared secret?
Please enter the domain name for this host: <assigned-domain-name-in-DSM>
Please enter the host group name for this host, if any:
Please enter a description for this host: Linux RH-6

Shared secret  : *********
Domain name  : <assigned-domain-name-in--DSM>
Host Group  : (none)
Host description  : Linux RH-6
Are the above values correct? (Y/N) [Y]:Y
```

7. If the Shared Secret information is correct, enter **Y**. Enter appropriate information when prompted for enabling hardware association (see “Random Number Generation Method” on page 46).

```
It is possible to associate this installation with the hardware of this
machine. If selected, the agent will not contact the DSM 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]:
```

8. Enter Y or press **Enter** to enable. If everything is working properly, the install program generates certificate signing requests and then generates the signed certificates. Unlike the fingerprint method, the fingerprints do not display for verification:

```
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.
Successfully registered the Vormetric Encryption Expert File System Agent with the
primary
Vormetric Data Security Server on thl1490.i.vormetric.com.
Installation success.
[root@host15101 Downloads]#
```

9. Verify the installation by checking the VTE version on the host, type:

```
# vmd -v
```

Silent Install

This section describes how to perform a silent installation of the VTE 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 VTE on multiple hosts simultaneously.

Before You Begin

The silent install method installs VTE on the host, and registers the host with the DSM you specify in the silent installation file.

For the Fingerprint Registration method, the DSM Administrator must add all hosts on which you will install VTE to the DSM. The following functionality must be enabled:

- Registration Allowed
- Communication Enabled

For the Shared Secret Registration method, you may not need to add hosts to the DSM.

Create the Silent Installation File

The following table shows the required and optional environment variables to be entered in the silent installation file. You can store this file anywhere on your system.

- Initiate a silent install by using the `-s` option in the install command.
- Use `-i` to install VTE and suppress the license text so that it is non-interactive.

Table 3-3: Register host options for silent install

Variable	Description	Required?
`AGENT_HOST_NAME`	FQDN of this VTE's host	Yes, if HOST IP is being registered.
`AGENT_HOST_PORT`	The port number for VTE. Ignored for other agents.	No
`AGENT_USEIP`	Uses IP address instead of host name	No
`ENABLE_DOCKER`	Set to 1 to automatically enable and register Docker during silent install.	No
`ENABLE_LDT`	Set to 1 to automatically enable and register LDT during silent install.	No
`HOST_DESC`	Specifies a host Description on the **Hosts** page of the DSM Management Console. Works only with `SHARED_SECRET`.	No
`HOST_DOMAIN`	Specifies domain for the shared secret. Required if using Shared Secret method.	Yes
`HOST_GROUP`	Specifies the optional host group for the shared secret.	No
`ONEWAY_COMMS`	Set to '1' when VTE-initiated-only communication is required	No
`SERVER_HOSTNAME`	FQDN of DSM	Yes
`SHARED_SECRET`	Specifies the passphrase for a shared secret registration. See "Host Name Resolution" on page 42	Yes

Variable	Description	Required?
STRONG_ENTROPY	Use /dev/random on Linux. Set to '1' if desired	No
USEHWSIG	Associate hardware to keys+certs. Set to '1' if desired.	No (default: false)

Silent Install with Shared Secret Registration Method

1. Create a parameter file and store it on your system. Following is an example file containing the FQDN of the DSM and the FQDN of the host on which you will install VTE. In this example, the file is called unattended.txt.

   Example:

```
SERVER_HOSTNAME=linux64.example.com
AGENT_HOST_NAME=RH6.example.com
SHARED_SECRET=Shallacl12345#
USEHWSIG=1
HOST_DESC="Linux RH-6"
```

2. Log on as an administrator to the host where you will install VTE.
3. Copy or mount the installation file to the host system.
4. Start the installation. Type:

   # ./vee-<product-version-build-system>.bin -s -i <dir>/unattended.txt

   Example:

```
# ./vee-fs-6.3.1-45-sels12-x86_64.bin -s /tmp/unattended.txt
```

5. Verify the installation by checking VTE processes on the host:
   a. Run vmd -v to check the version of VTE matches that just installed.
   b. Run vmsec status to display VTE kernel status.
   c. Look at the log files in /var/log/vormetric, especially install.fs.log.<date>, vorvmd_root.log.

Silent Install with Fingerprint Registration Method

1. Create a parameter file and store it on your system. Following is an example file containing the FQDN of the DSM and the FQDN of the host on which you will install VTE. In this example, the file is called unattended.txt.

   Example:

```
SERVER_HOSTNAME=linux64.example.com
AGENT_HOST_NAME=RH6.example.com
```

2. Log on as an administrator to the host on which you will install VTE.
3. Copy or mount the installation file to the host system.
4. Start the installation. Type:

   # ./vee-<product-version-build-system>.bin -s <dir>/unattended.txt

   Example:

```
#  ./vee-fs-6.3.1-45-sels12-x86_64.bin -s /tmp/unattended.tx
```

   The fingerprint displays:

```
D9:0E:B5:FF:51:F8:8F:2F:C9:F1:B0:74:5C:09:5B:45:BF:DA:01:9E
```

5. Verify the installation by checking VTE processes on the host:
   a. To check the version of VTE, type:

      ```
      # vmd -v
      ```

   b. To display VTE processes, type:

      ```
      # vmsec status
      ```

   c. Look at the log files in `/var/log/vormetric/install.fs.log.<date>`, especially `vorvmd_root.log`.

Automatically Registering LDT and Docker

The registration process now lets you register LDT and Docker during registration, instead of after.

For Linux: there are two new questions during registration:

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

For Windows, there is an option with the following label:

```
Enable LDT Feature (FS agent only)
```

Automatically Registering LDT and Docker During Silent Installation

If you want Docker or LDT to be automatically registered during a silent install, you can set the register_host script to do that.

1. Change to the following directory:

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

2. In a terminal, edit the `register_host` file.
3. Find the section entitled: `ask_enable_feature()`
4. Set the following:

   ```
   ENABLE_DOCKER = 1

   ENABLE_LDT = 1
   ```

Note: Docker and LDT are not compatible. You cannot use them both on the same system.

5. Save the file.
6. Run the Silent Install as described in the section: “Silent Install” on page 55.

Tracking and Preventing Local User Creation

VTE allows you to track attempts to change user authentication files. This includes, but is not limited to user creation, modification, and deletion, or to deny users.

All VTE versions enable detection and prevention of user accounts on the local host. You can deploy any 5.x or 6.x DSM for protection of the Linux host.

VTE provides the host settings, `protect` and `audit` for this purpose. The `audit` setting is set to on, by default. It audits access to the system credential files but does not prevent account creation. The `protect` setting both audits and prevents local user account creation. You must manually enable the `protect` setting for tracking and prevention of local user account creation.

You can tag the following files with either `audit` or `protect`:

```
/etc/passwd
/etc/group
/etc/shadow
/etc/gshadow
```

The **Protect** setting supersedes the **Audit** setting if both tags are applied to the same file.

Note: You do not have to restart VTE after applying or removing these host settings.

This VTE for Linux feature does not require a matching DSM version. You can use a VTE for Linux installation with a v5.x DSM. However, Vormetric highly recommends that you use the this VTE feature with a v6.x DSM. Although a VTE 6.x Linux installation can use this protection feature with a v5.x DSM, audit messages are absent on the v5.0 DSM.

Linux Package Installation

This section describes how to extract and run Linux packages directly so that the Vormetric VTE installation integrates with the distribution software.

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

To Extract and Run the RPM File

The VTE installation `bin` files contain the 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. Type:

   # ./vee-<product-version-build-system>.bin -e

   Example:

   ```
   # ./vee-fs-6.3.1-45-sels12-x86_64.bin -e
   Contents extracted.
   # ls *rpm
   vee-fs-6.3.1-45-sels12-x86_64.rpm
   ```

3. To start the installation using the RPM file, Type:

   # rpm -ivh vee-fs-6.3.1-45-sels12-x86_64.rpm

4. Follow the prompts until installation and registration are complete.

Restricted Mode

You can install VTE in restricted mode. This mode prevents users, other than root, from accessing the following directories:

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

Accessing Utilities

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

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

VTE Permissions in Restricted Mode

The following addresses VTE permissions in restricted mode on systems that also use key agents.

Key Agent or VKM

- On systems where VTE is installed in restricted mode, you cannot install key agent (pkcs11) or VKM.
- On systems where key agent (pkcs11) or VKM are already installed, you cannot install VTE in restricted mode.

Restricted Mode installation

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

```
# . /vee-<product><version><build-system>.bin -r
```

Example

```
# . /vee-fs-6.3.1-45-sels12-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.

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

Restrictions

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*
```

Uninstalling VTE

This section describes how to uninstall VTE on a Linux host.

Before Removing VTE from a Linux Host

Consider the following before removing a VTE from a host machine.

- Stop all applications from running on locations where hosts are applied.
- Before you remove VTE, decrypt any data you want to use after uninstall. Once VTE 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 host, the data is visible as clear text.
- The DSM Administrator must evaluate the current hosts in the *Guard FS* tab to avoid data loss or compromise.
- The DSM Administrator must remove **System Locked** and **FS Agent Locked** settings for this host (if set).
- Vormetric recommends that you remove all GuardPoints.
- VTE for Linux must be removed from the host before the host is removed from the DSM.
- Database applications like DB2, and Oracle can lock the user space while they run. If a VTE installation fails because a host is in use, determine which applications are using the hosts and stop them. Then run the uninstall again.

- Commands like `fuser` and `lsof` might not reveal an active host because they detect active usage, not locked states. Although it may appear that a host is inactive, it may be in a locked state. Under this condition, software removal may fail and an error like the following may display:

  ```
  /home: device is busy.
  ```

To Remove VTE from a Linux Host

1. Stop any application accessing files in the host.
2. Log on to the host as root with system administrator privileges.
3. Change directory to an unguarded location (for example, `/..`)

CAUTION

Do not change (`cd`) into the `/opt/vormetric` directory or any directory below `/opt/vormetric`. If you are in `/opt/vormetric`, or any directory below `/opt/vormetric`, 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.
```

4. 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!
```

Upgrading VTE

This section describes the generic instructions for interactively upgrading VTE. If there are any changes to this procedure for the current release of VTE, those changes will be documented in the VTE Release Notes.

If you want to schedule an upgrade to occur the next time the system boots, see “Scheduled Upgrade Feature” on page 62.

1. Stop any application accessing files in the host.
2. Log on to the host where you will upgrade VTE. 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 VTE License Agreement, you can include the `-y` parameter.

   For example, the following command upgrades the product to version 6.3.1.127 after the user manually reviews and accepts the VTE License Agreement:

   `# ./vee-<product-version-build-system>.bin`

   The following command upgrades the product to version 6.3.1.127 but automatically accepts the VTE License Agreement:

   `# ./vee-<product-version-build-system>.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 VTE 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
6.3.1.127
2020-08-20  09:45:20 (IST)
Copyright (c) 2009-2020 Vormetric. All rights reserved.
```

Scheduled Upgrade Feature

This section describes how to run the scheduled upgrade feature on VTE for Linux systems. This option is available for VTE version 6.1.3 or later. It contains the following topics:

Note: Scheduled upgrade on reboot is not supported on HDFS nodes.

Warnings for VTE for Linux

- As with prior VTE versions, DSM connectivity is required during upgrade.
- Yum updates or OS patches should be done prior to VTE 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 crash or 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 VTE version continues to run after the reboot. Restart the system to upgrade successfully.
  - If a crash or power failure occurs during the upgrade, VTE may enter an inconsistent state. The only way to restore is to enter single-user mode and delete all the files or directories related to VTE.
  - 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 VTE 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 VTE GuardPoint, or a third party anti-virus software doing periodic scans.

  See the "VTE and systemd" on page 315 and "Ubuntu Upstart Service Support" on page 321 for examples of how to set up VTE 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 VTE for Linux you currently have installed, use the `vmd -v` command:

```
$ vmd -v
Version 6, Service Pack 2
6.3.1.38
2020-05-31 11:09:40 (IST)
Copyright (c) 2009-2020, Vormetric. All rights reserved.
```

2. To schedule an upgrade on reboot, use the `voradmin upgrade schedule <path to VTE installer> -y [-p <path to temp upgrade directory>]` command, where:
   - `<path to VTE installer>` is the full path to the VTE installation file for the release to which you want to upgrade. For example, `./vee-fs-6.3.1-38-rh6-x86_64.bin`.
   - `-y` is an optional parameter that automatically accepts the VTE License Agreement. If you do not specify this parameter, the installer displays the VTE License Agreement and you must manually accept it before the upgrade can be scheduled.
   - `-p <path to temp upgrade directory>` is an optional parameter that specifies the path to a directory in which you want VTE to store the temporary files it needs during the upgrade. The default is `/var/tmp/`.

     To use this option:
     - The currently installed version of VTE must be 6.3.1 or later.
     - The version of VTE to which you are upgrading must be version 6.3.1 or later.
     - The specified path must not be inside a GuardPoint or be a symbolic link.

   For example, if you are upgrading from version 6.3.1.38 to version 6.3.1.127 and you want to automatically accept the license agreement and use the temporary directory `/temp_dir/`, you would enter:

```
$ voradmin upgrade schedule ./vee-fs-6.3.1-38-rh6-x86_64 -y -p /temp_dir/
Linux Kernel 3.10.0-862.el7.x86_64 is supported with the VTE version 6.3.1.127
being installed.
Created symlink from /etc/systemd/system/multi-yuser.target.wants/secfs-
upgrade.service to /usr/lib/systemd/system-secfs-upgrade.service.
Successfully scheduled upgrade on reboot to build 6.3.1.127.
```

   If you want to manually accept the VTE License Agreement and let the installer use the default temporary directory `/var/tmp/`, you would enter:

```
$ voradmin upgrade schedule ./vee-fs-6.3.1-38-rh6-x86_64
```

   The installation script runs and displays the license agreement. When prompted, enter “Y” to proceed with the installation, or “N” to cancel it.

```
Do you accept this license agreement? (Y/N) [N] Y
Linux Kernel 3.10.0-862.el7.x86_64 is supported with the VTE version 6.3.1.127
being installed.
Upgrade on reboot detected: this product will be upgraded on shutdown/reboot.
Do you wish to proceed with the upgrade? (Y/N) [Y]: Y
Created symlink from /etc/systemd/system/multi-user.target.wants/secfs-
upgrade.service to /usr/lib/systemd/system/secfs-upgrade.service.
Successfully scheduled upgrade on reboot to build 6.3.1.127.
```

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 VTE version is 6.3.1.38, upgrade on reboot scheduled for VTE 6.3.1.127.
```

4. Reboot the machine, then log in and verify that the upgrade was successful.

```
$ vmd -v
Version 6, Service Pack 2
6.3.1.127
2020-08-20  09:45:20 (IST)
Copyright (c) 2009-2020 Vormetric. 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 6.3.1.127.
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 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 6.3.1.127.
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 6.3.1.127. 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
6.3.1.127
2020-08-20  09:45:20 (IST)
Copyright (c) 2009-2020 Vormetric. All rights reserved.
```

Voradmin Commands for the Scheduled Upgrade Feature

To view the current VTE version, use the `vmd -v` command:

```
$ vmd -v
Version 6, Service Pack 2
6.3.1.38
2020-05-31 11:09:40 (IST)
```

To schedule an upgrade on reboot, use the `voradmin upgrade schedule <path to VTE installer> [-y] [-p <path to temp upgrade directory>]` command, where:

- `<path to VTE installer>` is the full path to the VTE installation file for the release to which you want to upgrade. For example, `./vee-fs-6.3.1-38-rh6-x86_64.bin`.
- `-y` is an optional parameter that automatically accepts the VTE License Agreement.
- `-p <path to temp upgrade directory>` is an optional parameter that specifies the path to a directory in which you want VTE to store the temporary files it needs during the upgrade. The default is `/var/tmp/`.

To use this option:

- The currently installed version of VTE must be 6.3.1 or later.
- The version of VTE to which you are upgrading must be version 6.3.1 or later.
- The specified path must not be inside a GuardPoint or be a symbolic link.

For example, to schedule an upgrade on reboot to version 6.3.1.127 and automatically accept the VTE License Agreement and use the temporary install directory `/temp_dir/`, you would enter:

```
$ voradmin upgrade schedule ./vee-fs-6.3.1-45-sels12-x86_64.bin -y -p /temp_dir/
```

To get information about a scheduled upgrade:

```
$ voradmin upgrade show
Upgrade on reboot is currently scheduled.
Current VTE version is 6.3.1.38, upgrade on reboot scheduled for VTE 6.3.1.127.
```

If no upgrade has been scheduled, this command displays the following:

```
$ voradmin upgrade show
Upgrade on reboot has not been scheduled.
```

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 4: Special Cases for VTE Policies

This chapter describes some VTE-specific configuration tasks related to configuring policies in the DSM and contains the following sections:

More Information About Configuring VTE Policies

This chapter describes some special cases that apply only to VTE agent policy configuration. See the DSM Administration Guide for general information about configuring polices. The following chapters in the DSM Administration Guide pertain specifically to the information in this chapter:

- "Creating and Configuring Signature Sets"
- "Configuring Hosts and Host Groups"
- "Configuring Policies"

Re-Signing Executable Files on Secfs GuardPoints

In VTE 6.1.0 for Linux and VTE 6.1.3 for Windows and previous versions, an issue affected signed executables in LDT policies. In these VTE versions, any executable that is part of either a host setting, or Signature set, and resides in a GuardPoint that uses an LDT policy will have a different SHA signature after every LDT key rotation. So after each LDT key rotation the host settings executables will no longer be authenticated, or the Signature Set policy rules that include those executables will no longer match them as expected. This problem occurred because VTE generated an SHA signature of the encrypted executable which changes after each LDT key rotation. To work around these issues on these older VTE versions, the security administrator must manually re-sign each affected executable after each key rotation.

Starting with VTE release 6.1.2 for Linux and 6.2.0 for Windows, the SHA signature is created from the unencrypted executable. This new SHA signature does not change with a key rotation.

If upgrading or installing a new machine using the same signature sets that you used previously, do the following:

1. Install the current release of the VTE agent. The previous signatures will be used until the next key rotation.
2. Before the next key rotation, the security administrator must resign the binaries.
3. Do not remove the old signatures on the DSM until all agents have been upgraded to the latest VTE release. Refer to the DSM Installation and Configuration Guide for information on how to perform a manual re-sign.
4. After all agents have been upgraded, then you can remove the old signatures.

If you are installing the VTE agents for the first time with VTE 6.1.2 (or later) for Linux or 6.2.0 (or later) for Windows, there are no special steps if no signatures have been defined. New signatures will sign using the new method so it is not necessary to manually re-sign each affected executable after each key rotation.

Note: In previous releases, if the executable was in a GuardPoint protected directory, but was the same as an unguarded executable, the administrator could restrict only the guarded executable. With the change in 6.1.2 (or later) for Linux or 6.2.0 (or later) for Windows, the unguarded executable matches the guarded executable with regards to policies.

Enable Automatic Signing for Host Settings (Linux)

A new feature of VTE blocks automatic re-signing of the host settings. Some users may have established procedures for updating system software. The user created these procedures based on their assumption that restarting the `vmd` will generate new signatures when signed software is updated. This is no longer true. To restore this behavior for updating system software, you must disable this new feature.

Disabling on Linux

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
   ```

   Previously this setting was known as `RE_SIGN_HOST_SETTINGS`. As of VTE 6.1.3 and later, the attribute name is `AUTO_RESIGN_HOST_SETTINGS` as shown here.
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
   ```

Enabling the automatic regeneration of signatures exposes a potential security vulnerability for agents. When enabled, host setting binaries are re-signed when it receives a push from the DSM. If an attacker were to replace a binary with a Trojan, and then force a push from the DSM by, for example, restarting the agent, VTE could generate a signature for the malicious binary and pass it.

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 VTE to not trust any authentication attempt performed by certain identities by assigning restricted users to a user shell that VTE 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.

VTE 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:

1. At the DSM management console, click **Hosts > Hosts**.
2. Click on an existing host name to edit the host.
3. Click **Host Settings** tab.
4. Add the following to the host 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 **OK.**

Behavior of Hard Links Inside and Outside of GuardPoints (Windows)

When using hard links on Windows, all the hard links to a file must be within the boundary of a GuardPoint and must use the same key. The following scenarios provide additional details:

- If hard links to the same file are inside a GuardPoint and outside a GuardPoint, the effect on the file depends on what process accesses which hard link first. If the hard link within the GuardPoint is opened first, the file is transformed. If the hard link outside the GuardPoint is opened first, the file won't be transformed.
- If hard links to the same file exist in different GuardPoints with different keys, the file will be corrupted.
- If hard links to the same file exist in the same GuardPoint but with different keys, such as if folder-based rules are used, there will be a conflict in the key.

Chapter 5: Installing VTE on Hadoop

This chapter describes how to protect an HDFS cluster with VTE. It contains the following sections:

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, VTE could only protect directories and files on the *local file system* rather than the actual *HDFS* files and directories. Now, VTE can protect files and directories.

DSM Administrators 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 VTE, install VTE on all the NameNode and DataNode hosts in a cluster.

Overview of VTE on HDFS

This section lists the high-level steps for implementing VTE protection on your HDFS. The process requires that the HDFS Administrator and DSM Security Administrator 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 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 DSM Security Administrator.
3. Ask DSM Security Administrator to:
   a. Add the NameNode to the HDFS Host Group.
   b. Create a GuardPoint for the HDFS Host Group on each of these directories.

DSM Security Administrator:

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

HDFS Administrator:

1. For each DataNode, take the node offline and perform a data transformation (see the *VTE Live Data Transformation Guide*).
2. Ask the DSM Security 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 VTE on HDFS

This section describes how to implement VTE 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 the nodes are configured, you can create GuardPoints on specific HDFS files and directories.

These instructions require that the HDFS Administrator and DSM Security 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 VTE on HDFS on a Single Host" on page 77.

VTE on HDFS Implementation Assumptions

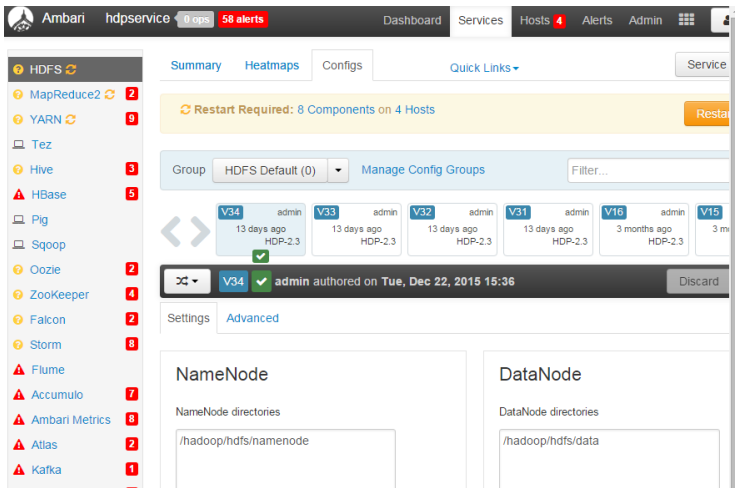
- You have installed, configured and registered VTE on all of the NameNodes and DataNodes in the Hadoop cluster.
- The HDFS Administrator has knowledge and experience with HDFS and Ambari.
- The DSM Administrator has knowledge and experience with VTE
- The two can work and communicate in tandem with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

Configure the HDFS NameNodes

The first step to implementing VTE 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 DSM **Security 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**

Figure 5-1: DataNode directories on Ambari



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 DSM Security Administrator:
   a. The directory list and instructions to create a GuardPoint for the HDFS Host Group on each of these directories.
   b. Instructions to add the NameNodes IP addresses or host names to the HDFS Host Group.

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.rename.check</name>
    <value>true</value>
</property>
<property>
    <name>dfs.vte.ioctl.device</name>
    <value><.sec folder name, up to the VTE 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 VTE 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>
```

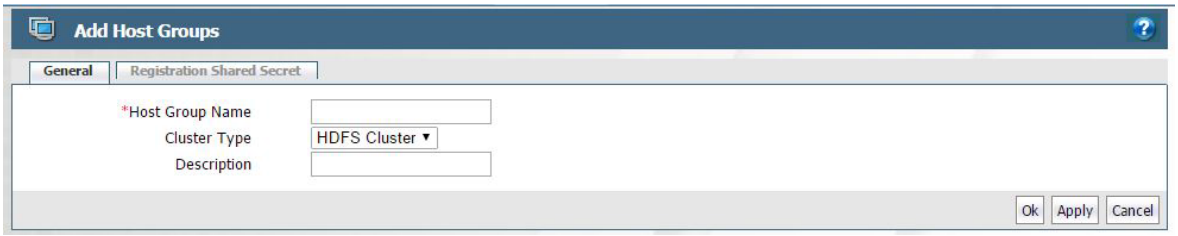
2. Save the file.
3. Restart HDFS NameNode and DataNode services.

Create a HDFS Host Group and Host Group GuardPoint

After configuring the NameNodes, the next steps in activating VTE on HDFS is for the DSM **Security Administrator.**

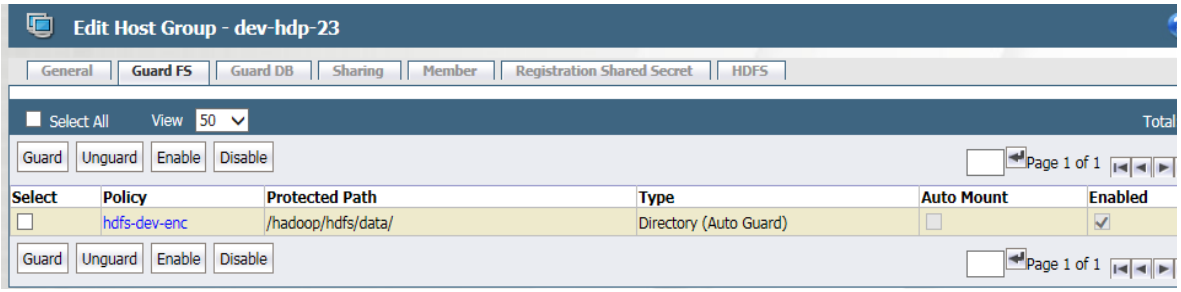
Create an HDFS Host Group to contain the HDFS nodes:

1. In the DSM Management Console, click **Hosts** > **Host Groups** > **Add**.
2. Enter a **Host Group Name** for the Hadoop cluster.
3. Select **HDFS Cluster** for the Cluster type.
4. (Optional) Enter a description and click **Ok**.



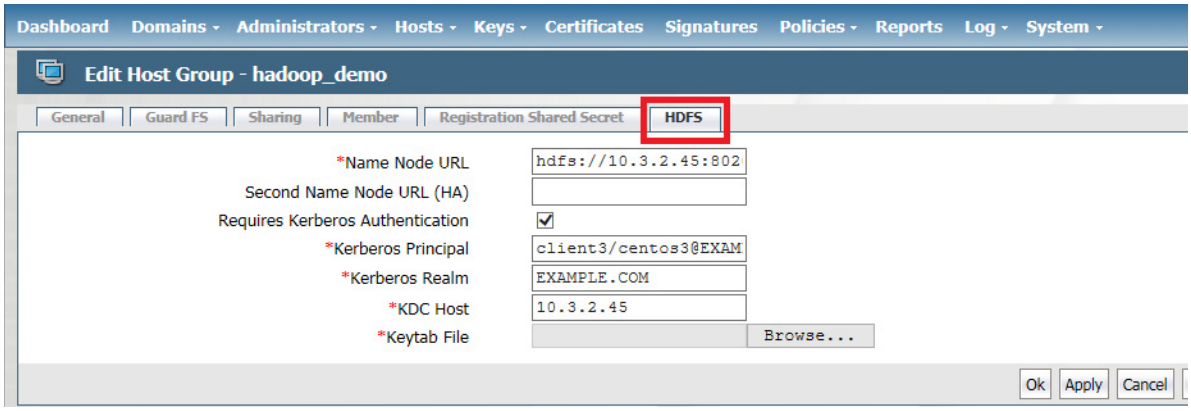
5. In the **Edit Host Group** page, click the HDFS tab.
6. For a Hadoop authentication configured as **Simple mode**, enter the NameNode URL information in the URL format: `hdfs://<host>:<port>`.

By default the port number is 8020. Check the HDFS configuration to verify this. HDFS HA cluster requires the URLs for both active and standby.



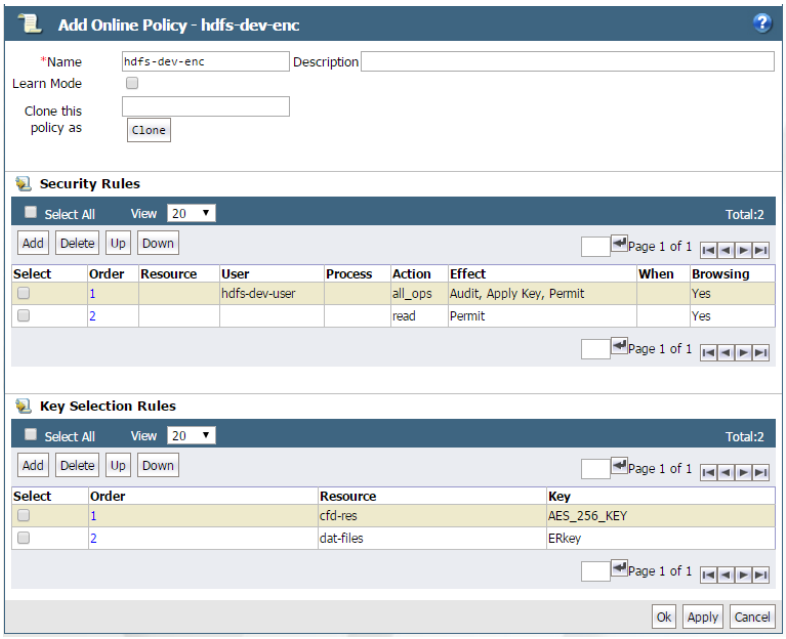
7. For Hadoop authentication configured as **Kerberos**, enter the NameNode URL information in the URL format: host name (not IP address).
8. Check the **Requires Kerberos Authentication** option and enter the following Kerberos information used for authentication:

- **Kerberos Principal:** Unique identity to which Kerberos can assign tickets. Format is: `primary/instance@REALM`.
- **Kerberos Realm**: Typically your domain name.
- **KDC Host**: Hostname of your domain controller.
- **Keytab File**: File containing pairs of Kerberos principals and encrypted keys.



9. You can create any policy you choose, but the example encryption policy, `hdfs-dev-enc`, uses the following rules:
   - For the user set `hdfs-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-dev-data-1`, the key is `hdfs-dev-key-1`.

- For the resource set `hdfs-dev-data-2`, the key is `hdfs-dev-key-2`.



10. If you haven't already done so, add the NameNode obtained from the HDFS Admin to the HDFS Host Group.

Note: Vormetric highly recommends 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 other are not.

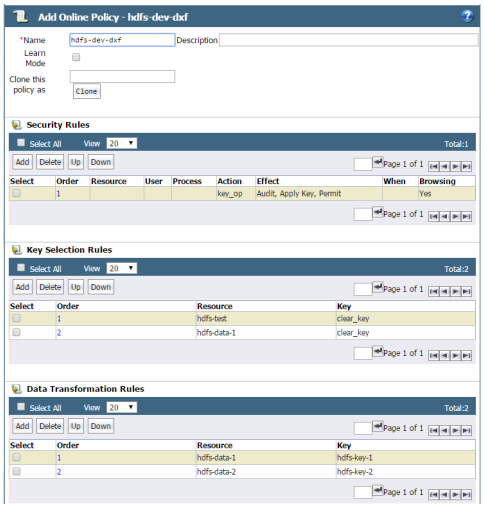
Take a DataNode Offline and Perform Data Transformation

The next step in activating VTE 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 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 DSM **Security 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 the HDFS NameNodes" on page 72). Read the VTE *Data Transformation Guide* for instructions on how to encrypt files and directories.
3. DSM **Security Administrator**: Create encryption keys and a data transformation policy to transform the data.

   The following figure shows an example of a data transformation policy that transforms the Resource Set `hdfs-dev-data-1` from `clear_key` to `hdfs-dev-key-1`. It also transforms the Resource Set `hdfs-`

`dev-data-2` from `clear_key` to `hdfs-dev-key-2`. Resource Set `hdfs-dev-data-1` consists of `/tmp/data1` and the Resource Set `hdfs-dev-data-2` consists of `/tmp/data2`.



4. **HDFS Admin**: After encrypting the data in those directories, ask the DSM Security Administrator to add the DataNode to the HDFS Host Group.
5. DSM **Security Admin**: Add the DataNode host to the HDFS Host Group.
6. **HDFS Admin**: After the DataNode is added to the HDFS Host Group, bring the DataNode online.
7. Repeat this procedure for all the DataNodes in your HDFS cluster.

Implementing VTE 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 VTE deployment guidelines:

1. Configure the HDFS NameNodes (see "Configure the HDFS NameNodes" on page 72):

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 DSM Security 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 Group and Host Group GuardPoint (see "Create a HDFS Host Group and Host Group GuardPoint" on page 74):
   a. DSM Security Administrator must create an HDFS Host Group to contain the HDFS nodes.
   b. The DSM Security 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 to the Host Group.

Adding a New DataNode to a VTE-protected HDFS

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

Note: If you already have VTE installed on the cluster nodes before Ambari installs the Hadoop software, see "VTE installed on the cluster nodes before Ambari installs Hadoop" on page 78.

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 DSM database and make sure that the host settings of the new node is the same as existing nodes in the cluster. See the *VTE Installation for Hadoop* chapter in the *VTE Installation and Configuration Guide.*
3. Install VTE on the new node, register to DSM, and run `config-hadoop.sh` to prepare the libraries. See the *Configuring Hadoop to use VTE* section in the *VTE 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 to DSM 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 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.

VTE installed on the cluster nodes before Ambari installs Hadoop

If VTE 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.

HDFS Upgrade with VTE

To upgrade Hadoop from version 2.6.0 to version 2.7.0 and higher, configure VTE to integrate with the new HDFS instance.

Upgrading one node at a time

Once VTE is installed and configured on the node:

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

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

4. Start HDFS services on the node.

Upgrade VTE with LDT in an HDFS Cluster

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

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

Note: Always upgrade namenodes before datanodes.

4. After VTE upgrade succeeds, type:

```
# config-hadoop.sh -i -y
```

5. On the Ambari admin console, start the namenode.
6. Verify that the Vormetric java process successfully launched in the namenode. (You should not see an error message.) Type:

```
# ps -ef | grep java | grep vormetric
```

7. Check the DSM status. The DSM 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 VTE 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("/opt/vormetric/DataSecurityExpert/agent/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("/etc/vormetric/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("/opt/vormetric/DataSecurityExpert/agent/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("/etc/vormetric/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 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 mapreduce 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 VTE

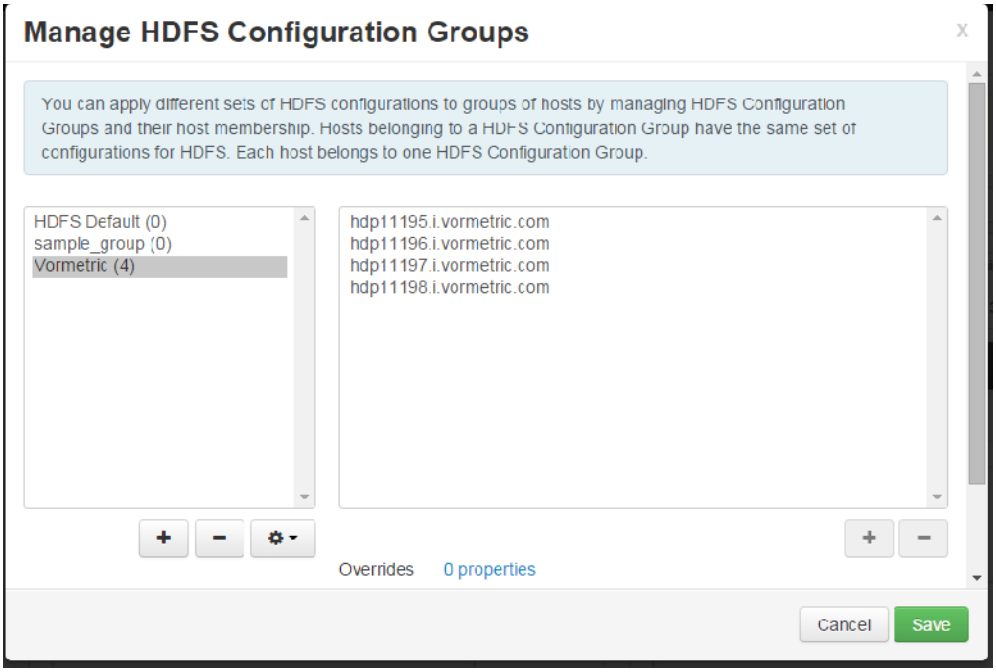
Configure the Hadoop cluster to use VTE before installing and configuring VTE on the nodes. Use Ambari to perform this configuration.

Create a Vormetric Configuration Group

The Vormetric Configuration Group will eventually contain all the hosts in your Hadoop cluster. At first you will create an empty group and later 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 *Vormetric*.

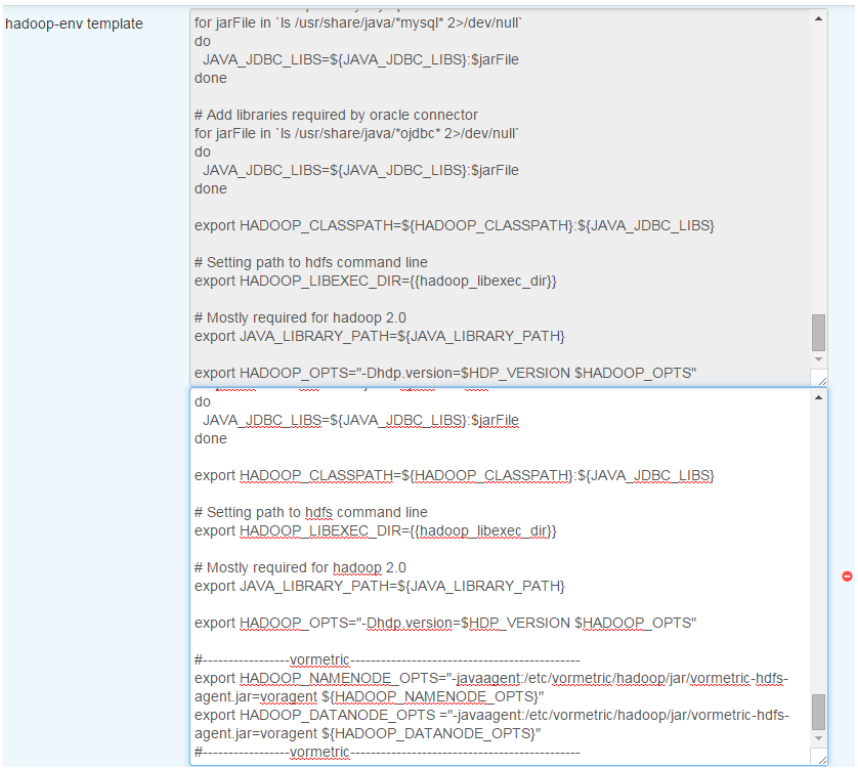
3. Make group *Vormetric* the current group.



Update the Hadoop-env Template with VTE Settings

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

```
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**.

Create Kerberos Principal for VTE

If Hadoop is configured in the secure mode with Kerberos enabled (`hadoop.security.authentication=Kerberos`), you need to create the Kerberos principal for VTE. Make the principal renewable with the `maxlife` property (Maximum ticket life) larger than 1 hour and smaller than 1 day, and Maximum ticket life smaller than Maximum renewable life (which by default is 7 days).

Configure the *keytab* file and principal with HDFS using these steps:

1. Go to **HDFS->Configs->Custom hdfs-site**
2. Set **dfs.vte.keytab.file**=<*VTE keytab file*>.
3. Set **dfs.vte.kerberos.principal**=<*VTE principal name*>.

VTE calls `kinit` to initialize the Kerberos ticket and renew the ticket once per hour. You can execute the following steps from the command line to verify that the Kerberos principal was created and configured for VTE correctly:

```
kinit -r 1440m -k -t <VTE keytab file> <VTE principle>

kinit -R
```

Uninstalling VTE for the Hadoop Cluster

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

Follow the steps below:

1. Shut down one DataNode.
2. Perform the normal agent uninstall procedure
3. Go to Ambari, remove the DataNode host from the Vormetric Configuration Group.
4. Start the DataNode.
5. Delete the Vormetric 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.

VTE Installation and Configuration

After configuring the Hadoop cluster for VTE:

1. Install and register VTE on the HDFS nodes.
   - You can do this to all the nodes at once, but the HDFS is unavailable during VTE installation and configuration.
   - You can also do this one node at a time. If you install and register VTE notes 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 Vormetric Configuration Group, then proceed with agent installation and configuration. See “Installing and configuring VTE on an HDFS node” on page 85.
   - Modify the Host Group. See “Modifying host settings for HDFS hosts on the DSM” on page 85.
   - Configure VTE by running `config-hadoop.sh` on the HDFS node. See “Configuring Hadoop to Use VTE” on page 87.

- Review the `SecFS` configuration variables that support the HDFS name cache. "HDFS name cache" on page 88.

Installing and configuring VTE on an HDFS node

1. Using Ambari, add the FQDN of the node to the Vormetric Configuration Group. See "Create a Vormetric Configuration Group" on page 81.
2. Install, configure, and register VTE as described in "Installing VTE for Linux" on page 41.
3. Modify the host settings for each node. See "Modifying host settings for HDFS hosts on the DSM" on page 85.

Modifying host settings for HDFS hosts on the DSM

The Hadoop service can start as root and then downgrade to an unprivileged user. If the unprivileged user is not authenticated by password, VTE flags the user as fake. VTE 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 DSM host 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 DSM Administrator to add the following lines to the Host Group in the DSM Management Console:

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 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
```

Using Kerberos

To use **Kerberos**, ask the DSM Administrator to add the following two lines to the **Host Settings** in the DSM Management Console:

Note: `/usr/jdk64/jdk1.8.0_40/bin/java` and `/usr/lib/bigtop-utils/jsvc` are the Java executables used to launch the HDFS services. Change the versions accordingly to fit your 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.SecureDataNo
deStarter|/usr/lib/bigtop-utils/jsvc
```

The entire Host Group 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

|authenicator+arg=+class=org.apache.hadoop.hdfs.server.namenode.NameNode|/usr/
jdk64/jdk1.8.0_40/bin/java
|authenicator+arg=+class=org.apache.hadoop.hdfs.server.datanode.SecureDataN
odeStarter|/usr/lib/bigtop-utils/jsvc
```

Modifying Host Group for HDFS NameNodes HA on DSM

To enable high availability (HA) for your HDFS NameNodes, ask the DSM Administrator to add the following lines to the Host Group in the DSM Management Console

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/jdk1.8.0_40/bin/java
|authenticator+arg+class=org.apache.hadoop.yarn.server.applicationhistoryservice.A
pplicationHistoryServer|/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 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
```

```
|authenicator+arg=+class=org.apache.hadoop.hdfs.server.datanode.SecureDataN
odeStarter|/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.Applicat
ionHistoryServer|/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 VTE

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:c(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.`

Verify the configuration the using ?s option:

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

System Response:

```
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 in the following file:

```
# /opt/vormetric/DataSecurityExpert/agent/secfs/.sec/ conf/<variable name>
```

For example, you can configure the variable `hdfs_cache_entry`

```
../.sec/conf/hdfs_cache_entry
```

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 VTE on HDFS

To enable VTE on your HDFS:

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

     ```
     # /etc/init.d/secfs restart
     ```

   - For Redhat 7.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 DSM console, click **Host > Host Groups.**
2. Click <*host group name*>. The Edit Host Group - <*host group name*> window opens.
3. Click **Guard FS**.
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`

   System Response:

   ```
   No guardpoints configured
   ```

7. On the node running secfs, type:

   `# voradmin ldt attr delete <guard path>`

   Example:

   `# voradmin ldt attr delete /hadoop/hdfs/data`

   System Response:

   ```
   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>`

   Example:

   `# voradmin ldt rmstore /hadoop/hdfs`

   System Response:

   ```
   Enter YES if /hadoop/hdfs does not include any guardpoints associated with
   an LDT policy ->YES
   MDS file /hadoop/hdfs/::vorm:mds:: has been removed.
   ```

9. Verify that the metadata store has been removed from the secfs mount points, type:

   `# ls -altr <mount_point>`

Example:

```
# ls -altr /hadoop/hdfs
```

You should not see `/hadoop/hdfs/::vorm:mds::` listed.

10. Repeat the above steps for each node in the HDFS cluster.

Chapter 6: Using VTE with Oracle

This chapter describes how to install and configure VTE on Oracle RAC ASM, Linux/It contains the following sections:

Oracle RAC ASM and ASMLib

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

Using VTE with an Oracle RAC ASM

You can apply VTE 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 VTE 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 VTE 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.

- **EMC PowerPath**

  If using EMC PowerPath then the device names are similar to the following: `/dev/emcpowerXXXX`.

  When browsing the DSM through the local host, you cannot find Power Path devices. You must manually input the paths. The guarded disk names are prepended with: `/dev/secvm`.

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 VTE 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 VTE 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 VTE 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 VTE encryption to the disk.
6. Recreate the diskgroup.
7. Add the protected disk to the diskgroup.
8. Restart the nodes and the failover test.
9. Repeat the previous steps for each disk in the diskgroup.

General Prerequisites

Follow these guidelines for best results.

Setup

- Verify that you have a current backup of the database
- Install and register VTE 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 DSM. 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: For RHEL 5 operating systems.

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

1. Edit the file `/etc/udev/rules.d/90-dm.rules`.
2. 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"
```

3. Restart the UDEV rules by executing the following command:

```
/sbin/start_udev
```

Note: The VTE 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 VTE. 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.

Example

Use Oracle ASMlib to bind the device:

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

Using 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 VTE.

Restarts can uncover issues in the RAC environment that are unrelated to VTE. To avoid issues after a VTE installation, Thales recommends that you restart each RAC node **AFTER** VTE is installed and **PRIOR** to establishing any GuardPoints. This may not be feasible in a single node configuration. However, by doing so, VTE 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, **VTE REQUIRES a block device for guarding**.

Note: 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.

Note: 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;
```

a.Delete the disk from ASM, type:

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

b.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 VTE 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;
   ```

   System Response:

   ```
   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
   ```

   System Response:

   ```
   DISKGROUP  DISK# DISKNAME   TOTAL_MB    FREE_MB PATH  HEADER_STATU
   ---------- ---------- ------------- ---------- ---------- ---------
   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 VTE, 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.

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

3. On the DSM, in the Host Group, apply a GuardPoint to the Raw Device: <*rawDevice1Name*>
4. From **RAC Node 1**, 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. Go to the section "Surviving the Reboot and Failover Testing" on page 100.

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.

   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.

2. On the DSM, in the Host Group, apply GuardPoints to the three raw devices:

   <*rawDeviceName1*>

   <*rawDeviceName2*>

   <*rawDeviceName3*>

3. On **RAC Node 1**, perform the following:
   a. Display the status of the guarded disks, type:

      # secfsd -status guard

4. 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>
```

5. 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.

6. On **RAC Node 1**, restore the database.
7. The system is now ready for a reboot and failover test. Go to the section "Surviving the Reboot and Failover Testing" on page 100.

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
```

VTE Load Order and Startup Scripts

The last change is to ensure that VTE 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 VTE 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 VTE 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 gypped 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.

Basic Troubleshooting Techniques

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, etc.

Perform the following tasks to verify that the settings are correct:

1. On the DSM **WebUI**, 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 (it is only for GPFS).
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:
   - The udev rules are not set correctly, see “Modify the UDEV Rules” on page 94
   - The device names are not the same across all nodes

4. From ASM, make sure that the `asm_diskstring` parameter is modified to include the VTE devices and that the proper pathname is used, see "Altering ASM_DISKSTRING on ASM" on page 95.

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, 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
60000000000D8000 01F27C42 40E75C23 5CE0202A C5F6465F
[..|B@.\#\. *..F_]
60000000000D8010 379BC9E1 B53A4016 E2DB2EF7 E2328706  [7....:@......2..]
60000000000D8020 CA2F30AD 522B4D21 99292639 004EBB34  [./0.R+M!.)&9.N.4]
60000000000D8030 A3896BE8 BD839D23 2204E19E 946C575C  [..k....#".....lW\]
60000000000D8040 4CE2218F 35E1B101 AF751A70 780E6D6E  [L.!.5....u.px.mn]
60000000000D8050 5E7E6A38 C600ED5F 929047C4 DF372A8E  [^~j8..._..G..7*.]
60000000000D8060 E103152D BA87CC03 11A7D963 9D72FCE1
[...-.......c.r..]
60000000000D8070 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.."0.:.~]
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 7: Configuring Support for SAP HANA

This chapter describes SAP HANA, which provides automatic host-failover support. It contains the following sections:

Overview

SAP HANA provides automatic host-failover support. VTE works with HANA fibre storage systems to enable and disable GuardPoints when a protected host starts, stops, or fails over to standby host.

SAP HANA supports non-shared storage where each HANA node has its own separate storage volumes. VTE 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. Vormetric provides hooks that are called by the HANA fcClient providers that manage guarding or unguarding of storage locations.

Note: Vormetric recommends using host groups to manage configuring in a clustered host environment.

Customizing VTE 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 VTE on a single HANA node, do not change the default HANA or VTE settings.

1. Go to the installation directory:

```
# cd /opt/vormetric/DataSecurityExpert/agent/secfs/saphana
```

2. If required, edit the appropriate VTE `fcClient` refined script:

Script file	Use
`fcClientRefinedVTE.py`	fcClient provider for LUN
`fcClientLVMRefinedVTE.py`	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 VTE scripts.

4. Edit the storage section of the `global.ini` file to indicate the corresponding VTE `fcClient` as the High Availability (HA) provider and to point to the location of the VTE 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 VTE 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. From the DSM, 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.

   Vormetric 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:

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* http://www.sap.com/documents/2016/06/84ea994f-767c-0010-82c7-eda71af511fa.html

Using SAP HANA with LDT

SAP HANA is compatible with LDT. You must add additional VTE commands to the HANA administrator entry.

- Using a text editor, edit `/etc/sudoers` and add entries for `/usr/bin/voradmin` and `/usr/bin/vmsec`:

  Example:

  ```
  # 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 VTE in conjunction with applications like SAP HANA that can process large numbers of direct I/O writes through the Linux AIO interface, VTE can allocate more memory than is desirable.

To limit the amount of memory that VTE consumes for AIO buffers, use the following configuration to limit the amount of memory VTE 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** VTE after changing any values in the configuration directory to make the changes effective.

Chapter 8: Using VTE with Microsoft SQL

This chapter discusses using VTE with Microsoft SQL AlwaysOn and SQL File Tables. It contains the following sections:

Using VTE with SQL

You must stop the SQL service before guarding the SQL DB. When this occurs, the SQL Server replication may become unsynchronized. When restarting, it may take a brief period of time for the SQL Server replication to resynchronize with the other node. The SQL Server issues a warning against any attempted failovers during that brief period.

Note: Minimizing the duration for which the SQL Server service is stopped is beneficial for reducing the resynchronization period.

Using LDT with SQL FILESTREAM

When applying a Live Data Transformation (LDT) GuardPoint to SQL Server with FILESTREAM enabled, a rekey may be triggered which never finishes. This can occur if SQL Server is renaming files when the GuardPoint is applied, which causes the rekey to start the scan process again. If the rekey seems to be taking a long time, stop the SQL service until the rekey finishes and then restart the SQL service.

Using VTE with SQL FileTables

SQL FileTables allows you to store files and documents in special tables in the SQL Server called FileTables, but access them from Windows applications as if they were stored in the file system, without making any changes to your client applications. For some of the use cases, you can use FileTables with VTE.

Considerations

- The VTE Agent must be installed on the same server where the FileTables reside. If the FileTables reside on your SQL server, then you should install the VTE Agent on your SQL server.
- If multiple servers access the SQL FileTables:
  - Install VTE agent on all of the servers.
  - Protect all of the FileTable folders with the same VTE policy.

CAUTION
Accessing the FileTable without VTE may corrupt the data.

- When you create a new FileTable, alter, or drop FileTables, this may require applying a new GuardPoint.
- Every FileTable has a separate FileTable Folder so you must apply separate GuardPoints for each FileTable.
- You must apply a unique GuardPoint to each VNN path.

  For example, if you configure two FileTables on an SQL Server, then the remote SQL administrator system must apply one GuardPoint to each configured VNN name.
- Guarding on a VNN name is similar to guarding a network path with VTE.
- If you want to access the FileTables from multiple remote systems, you must install VTE agent on those systems and apply the GuardPoints.

CAUTION
LDT is **not** supported with SQL FileTables. Only use offline Data Transformation to transform the initial SQL data.

Advantages

- System administrator cannot see the data locally on the SQL server because no VTE Agent is installed on the SQL server.
- The data transferring between servers is also encrypted.

Supported FileTables Use Cases

VTE supports the following FileTables use cases:

VTE Data Transformation of existing files in FileTables

Configuration guidelines:

1. Install VTE agent on the remote server.
2. Create a new FileTable, or Identify the FileTable folder for the existing FileTable.
3. Create an offline Data Transformation policy and apply to the GuardPoint on the FileTable folder.
4. Run the Dataxform utility to transform the data.

Protect files in SQL FileTables with VTE

Configuration guidelines:

1. Install VTE agent on the remote server.
2. Create a new FileTable, or Identify the FileTable folder for the existing FileTable.
3. Create a production policy and apply the GuardPoint on the FileTable folder.
4. Once the GuardPoint is active, you can use the file table to load and access files.

Protect files with SQL AlwaysOn Availability Groups with VTE

When the database that contains the FILESTREAM, or FileTable data, belongs to an AlwaysOn availability group, the FILESTREAM and FileTable functions accept or return virtual network names (VNNs) instead of computer names.

Configuration guidelines:

1. Install VTE agent on the remote server.
2. Create a new FileTable, or Identify the virtual network names (VNNs) for the existing FileTable.
3. Create a production policy and apply the GuardPoint to the VNN name
4. Once the GuardPoint is active, you can use the FileTable to load and access files.
5. When you enable FILESTREAM on an instance of SQL Server, it creates an instance-level share to provide access to the FILESTREAM data. Access this share by using the computer name in the following format:

```
\\<computer_name>\<filestream_share_name>
```

6. In an AlwaysOn availability group, the computer name is virtualized by using a Virtual Network Name, (VNN). When the computer is the primary replica in an availability group, and databases in the availability group contain FILESTREAM data, then SQL creates a VNN-scoped share to provide access to the FILESTREAM data. Applications that use the file system APIs have to use the VNN-scoped share, which has a path in the following format:

```
\\<VNN>\<filestream_share_name>
```

Install VTE on remote systems and guard the SQL Server VNN names

In this use case, VTE is installed on the SQL administrator system (a separate system from where the SQL Server resides) and a GuardPoint is applied to the VNN name.

Unsupported FileTables Use Cases

VTE does not support the following use cases:

1. Install VTE agent on the SQL Server and locally apply the GuardPoint on the SQL Server storage.
2. Access FileTables with Transact-SQL.
3. Access FileTables with File I/O APIs on the SQL server. Perform all file I/O on the remote system running the VTE agent.

Installing VTE on Microsoft SQL AlwaysOn

This section describes how to implement VTE with Microsoft SQL AlwaysOn in a variety of configurations for primary and secondary replica servers, and assumes that you have a basic understanding of Microsoft SQL database.

You may want to keep the primary server decrypted to serve all users, and use the secondary database for running reports or backups.

- If the database is encrypted, then the Volume Shadow copy-related backups will snapshot and backup encrypted protected data.
- Administrators with the `apply_key` permission can run a query and pull down reports from the secondary database server without affecting the performance of the primary database server.

- The secondary server could be in a remote Data Recovery location. You may want to secure it with encryption.
- LDT is supported with SQL AlwaysOn. See “Using LDT with SQL AlwaysOn” on page 115 for more information.

Methods for Initial Encryption

There are multiple methods for performing the initial encryption of the databases. Decide on which of the following methods best fits your environment. For more information on transforming data, see the “*VTE Data Transformation Guide*.”

- Data Transformation – Encrypt data in place
- Backup and Restore to a GuardPoint
- Copy and paste the data into a GuardPoint

Configuration 1

- Databases on primary server and secondary replica servers require encryption
- Database name and location of secondary replica server are the same as the primary server

To perform the procedure:

1. Perform a full backup of the primary database.
2. Change the primary database to offline mode.
3. Confirm the creation of a data transformation and/or operational policy.
4. Guard the folder containing the primary database files with that policy:
   a. If using 'Encrypt data in place' as the selected method of encryption, execute the data transformation and then apply the operational policy.
   b. If using the 'Copy/Restore ' method of encryption, apply the operational policy on an empty folder/device.
5. On the secondary server, create a new folder to store the replicated database.

Note: The folder name and the path must be the same as the primary server.

6. Guard the folder with the operational policy.
7. Perform step 4 above for additional secondary server(s).
8. Put the primary database back into online mode.
9. Setup SQL AlwaysOn High Availability group to perform FULL Data Synchronization.
   This copies the primary database and replicates it to secondary replica servers.
10. Verify that the databases in the secondary server are in “Synchronized” mode.

Configuration 2

- Database on the primary server does not require encryption, but the secondary replica database requires it
- Database names and locations for the secondary replica servers are the same as the primary server

To perform the procedure:

1. Perform a full backup of the primary database.
2. Confirm the creation of a data transformation and/or operational policy.
3. On the secondary server, create a new folder to store the replicated database.

Note: The folder name and the path must be the same as the primary server.

4. Guard the folder with the operational policy.
5. Perform step 3 & 4 above for additional secondary server(s).
6. Setup SQL AlwaysOn High Availability group to perform **FULL Data Synchronization**.

   This copies the primary database and replicates it to secondary replica servers.
7. Verify that the databases in the secondary server are in “Synchronized” mode.

Configuration 3

- Databases on the primary and secondary replica servers require encryption
- Database name is the same, but the location of the secondary replica server is in a different location from that of the primary server

To perform the procedure:

1. Perform a full backup of the primary database.
2. Change the primary database to offline mode.
3. Confirm the creation of a data transformation and/or operational policy.Guard the folder containing the primary database files with that policy:
   a. If using 'Encrypt data in place' as the selected method of encryption, execute the data transformation and then apply the operational policy.
   b. If using the 'Copy/Restore ' method of encryption, apply the operational policy on an empty folder/device.
4. On the secondary server, create a new folder to store the replicated database.

Note: The folder name and the path must be the same as the primary server.

5. Guard the folder with the encryption policy.
6. From secondary server, perform the restore to the primary database.
   a. Select the options **Restore with norecovery** and **Relocate all files to folder**.
   b. Specify the path of the new folder from step 5.
7. Repeat steps 4 & 5 above for any additional secondary server(s).
8. Setup SQL AlwaysOn High Availability group to perform **JOIN ONLY Data Synchronization.**

   This joins the secondary database to the SQL Always High Availability Group. It also establishes replication of new data and logs from the primary to the secondary replicated server.
9. Verify that the databases in the secondary server are in **Synchronized** mode.

Configuration 4

- Database on the primary server does not require encryption, but the secondary replica database requires encryption
- Database name is the same, but the location on the secondary replica server is in a different location than that of the primary server

To perform the procedure:

1. Perform a full backup of the primary database.
2. Confirm the creation of a data transformation and/or operational policy.On the secondary server, create new folder to store the replicated database.
3. Guard the folder with the operational policy.
4. From secondary server, perform restore the primary database:
   a. Select the options **Restore with norecovery** and **Relocate all files to folder**.
   b. Specify the path of the new folder from step.
5. Setup SQL AlwaysOn High Availability group to perform **JOIN ONLY Data Synchronization.**

   Joins the secondary database to the SQL Always HA Group. It also establishes replication of new data and logs from the primary to the secondary replicated server.
6. Verify that the databases in the secondary server are in **Synchronized** mode.

Data Transformation (Encryption in place)

For more information on transforming and encrypting data-in-place, see the *VTE Data Transformation Guide*.

Copy/Restore

For more information on transforming data using the copy and replace method, see the *VTE Data Transformation Guide*.

SQL Server Policy Tuning

In this section, you created and defined a process set for SQL Server that grants certain executables –in this case `sqlservr.exe-` unrestricted access to the database files. The need may arise to allow other executables, and/or users, access to the files.

You can grant this access by:

- Adding to the existing process set
- Creating a new one

The best option depends on the access requirements. The key decision is whether or not to select the **Apply Key** effect along with **Permit** or not. Omitting **Apply Key** on a security rule that still contains **Permit** allows the specified user or process to access to the data, but does not apply the encryption key, so therefore only shows them the data in its encrypted, cypher-text format. This is useful for anti-virus or backup software that may need to scan or copy the file, but does not necessarily need to see the contents.

Using LDT with SQL AlwaysOn

To guard a directory with an LDT (Live Data Transformation) policy, you must temporarily close all files in that directory. In a SQL Server AlwaysOn environment, this may entail temporarily stopping the SQL Server service on the node that is being guarded. Once the directory is guarded, then you can start the SQL Server service immediately.

It is important to remember the SQL Server AlwaysOn replication standard operating procedures.

- If one SQL Server service is taken offline for any reason, then once it is brought back on line, it takes the SQL Server a moment to re-synchronize the database nodes.
- The longer that secondary service was down, and the more inserts/updates and deletes that occurred on the still active node during that downtime, then the longer the synchronization period takes.
- During that synchronization period, any attempted fail over results in the SQL Server warning that data loss may occur if the fail over continues. However, once the SQL Server has completed re-synchronizing that secondary node, then any fail over is safe and does not result in loss of data.

Chapter 9: Concise Logging

This section describes Concise Logging and selective filtering.

This chapter contains the following sections:

Overview of Concise Logging

Thales's standard operational logging sends audit messages for each file system operation. An audit message is sent 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.

Concise Logging allows you to focus on relevant audit messages and important actionable messages, such as errors and warnings. It can eliminate the repetitive and less important audit messages generated by read and write activity on a file, reading and writing directory attributes, and other file system activity.

Concise Logging eliminates the following types of messages:

- Audit messages for each and every block read by the user or application. It sends only one audit message for each read/write activity.
- 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 DSM. You can configure Concise Logging for the following:

- All registered hosts in all domains; see "Do not use Learn mode with Concise Logging." on page 117
- A host that has registered with the DSM; see "Configuring Concise Logging for a registered host" on page 118

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 VTE `secfs`.
- Enable and disable Concise Logging on the host. VTE 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 DSM level, it applies to all hosts in all domains, that are added to the DSM, 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 global Concise Logging

You can enable or disable Concise Logging at any time. The DSM controls the function. Any change in the Concise Logging is reflected on any newly registered hosts and their domains.

To configure global Concise Logging:

1. Login to the DSM with System Admin privileges.
2. Click **System > Log Preferences**. Your system may contain multiple log tabs.
3. Click on a **Log** tab.
4. In the Duplicate Message Suppression Settings field, click **Enable Concise Logging**.
5. Click **Apply**.
6. Repeat steps for any other logs, as appropriate.

The host sends the following message after the administrator has enabled Concise Logging for an individual host:

```
DAO00821: Administrator “voradmin” updated Security Server configuration “Concise
Logging Enabled” from “true” to “false”.
```

Configuring Concise Logging for a registered host

You can enable Concise Logging for a host after you have registered it with the DSM. Hosts that are added to the DSM after enabling Concise Logging inherit the global settings from the DSM. This setting can be customized at any time.

To enable Concise Logging on the DSM for a registered host:

1. Log into your host with DSM security admin privileges.
2. Select the host that you would like to customize.
3. Select a **Log** tab.
4. In the Duplicate Message Suppression Settings, click **Enable Concise Logging**.
5. Click **Apply**.

After you enable or disable Concise Logging, VTE generates a log message to record that event:

```
"[CGA] [INFO] [CGA3201I] [11/11/2016 10:57:18] Concise logging enable
"[CGA] [INFO] [CGA3202I] [11/11/2016 10:57:27] Concise logging disabled
```

Chapter 10: Container Security

This chapter describes securing data in container environments, Docker or RedHat OpenShift, using VTE. It contains the following sections:

Note: Docker and OpenShift are for Linux only.

Installing Docker Automatically

A new option allows for automatically registering the host for Docker during registration. This allows for easily enabling Docker for a large number of hosts.

During the installation phase, the agent indicates to the DSM that it wants DOCKER. Only registration is performed at this time. After registration, the DSM will validate the Docker host to ensure that the agent and the DSM use compatible versions, and that LDT is not enabled for the host.

Note: LDT and Docker are not compatible.

During the installation phase, the agent indicates to the DSM that it wants DOCKER. Only registration is performed at this time.

Container Security Overview

VTE provides data security for container environments. You can set up data protection policies for container images. In addition to data encryption, VTE also provides container-level access control and audit logging. DSM Administrators can create GuardPoints in container images through the DSM Management Console. Users can use either of the following container options:

- **Docker Container**: CLI tool for creating containers.
- **Red Hat OpenShift with Customized Docker Container**: GUI tool for creating containers and persistent storage that mounts like NFS.

Container Terminology

- **Containers**

  The basic unit of OpenShift/Docker Application are called containers. A container runs one or more processes inside of a portable Linux environment. Containers are started from an Image and are usually isolated from other containers on the same machine.

- **Image**

  A layered Linux file system that contains application code, dependencies, and any supporting operating system libraries. An image is identified by a name that can be local to the current cluster or point to a remote Docker registry.

- **Pods (OpenShift only)**

  A POD is a set of one or more containers that reside on a host/node and share a unique IP address and volume, (persistent storage). OpenShift leverages the Kubernetes concept of a pod.

- **Project and Users (OpenShift only)**

  A namespace that provides a mechanism to scope resources in a cluster. Users interact with OpenShift. It grants permission to access applications.

Docker Containers with VTE

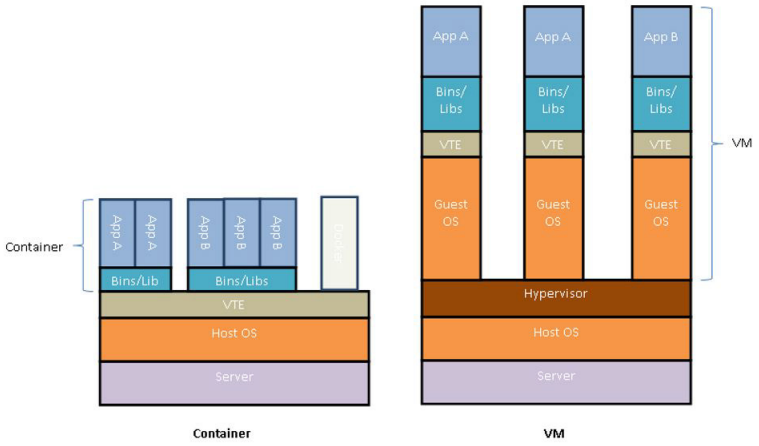
Docker allows for containerizing an environment for application deployment. Docker is an infrastructure built on top of Linux containers and various namespace components. Typically, an application deployment is bundled with all of the dependent packages in a single image called a Docker image. Docker images are ready to run applications in containers. A user can instantiate any number of Docker containers using from one or more Docker images. This makes it easy for users to move applications around on their servers. The containerization removes the pain of setting up an environment for applications and provides isolation for applications.

A Docker container can run any application, for example, a Postgres database server. Docker containers are used widely to run micro-services, which are stateless in nature. But a micro-service, when it is running, might generate a log trail inside the Docker container. This log trail might contain sensitive information. This warrants encrypting directories inside a Docker container. If a Docker container runs a database-server type of service, it will need data protection including encryption of data, granular access control and the ability to audit the log that accesses the data.

VTE: Virtual Machine versus Docker

VTE installs on the Docker container host and orchestrates security from the Docker host. It does not leave a footprint inside the Docker image or containers. You setup the GuardPoints inside individual containers at runtime. The diagram below depicts the difference between VTE deployment on a virtual machine and a Docker container host.

Figure 10-1: VTE in Docker container environment versus virtual machine environment



Using the VTE Agent

In order to use the VTE Agent to protect Docker images and containers, you must obtain a VTE Agent license for Docker. Contact Vormetric Support for information on obtaining a license.

There is no change to the VTE installation, however VTE agent must be installed on the Docker host system.

Docker containers in custom paths won't start after VTE agent is installed

This problem occurs if you have installed one or more Docker containers on a host in a directory path other than the Docker default of `/var/lib/docker`. After installing the VTE agent on such a host, any Docker containers installed on that host will no longer start if the VTE `secfs` service is running. This problem occurs even to docker containers not configured to work with the VTE agent.

The following error message appears when you attempt to start a Docker container:

```
exec user process caused "permission denied"
```

To work around this issue, create a symbolic link from the current installation path of the docker container to the default Docker path of `/var/lib/docker`:

```
ln -s /custom/path /var/lib/docker
```

After creating this symbolic link, affected Docker containers will start up normally even if `secfs` is running.

Set the Docker Storage Driver

On RHEL 7.x operating systems, the Docker engine default storage driver has changed from devicemapper to overlayfs for v1.13.1 and later. Currently, VTE only supports devicemapper. If your Docker engine uses overlayfs or any other storage driver, then you must change it to devicemapper before using VTE to protect your data.

Before you change your storage driver, verify your Docker version:

1. Login to your Docker agent CLI.
2. Check your Docker version, type:

   ```
   # docker version
   ```

   System Response:

   ```
   Client:
    Version:      17.04.0-ce
    API version:  1.28
    Go version:   go1.7.5
    Git commit:   4845c56
    Built:        Mon Apr  3 18:01:50 2017
    OS/Arch:      linux/amd64

   Server:
    Version:      17.04.0-ce
    API version:  1.28 (minimum version 1.12)
    Go version:   go1.7.5
    Git commit:   4845c56
    Built:        Mon Apr  3 18:01:50 2017
    OS/Arch:      linux/amd64
    Experimental: false
   ```

3. Check to what option the storage driver is set, type:

```
# docker Info
```

System Response:

```
Containers: 2
 Running: 1
 Paused: 0
 Stopped: 1
Images: 2
Server Version: 17.04.0-ce
Storage Driver: overlay
```

Change the Storage Driver

To change the storage driver:

1. Login to your Docker agent CLI.
2. Stop the Docker daemon, type:

```
# systemctl stop docker.service
```

3. Open another CLI session of agent so you can monitor Docker.
4. In `/etc/docker/` directory, create a file called `daemon.json` if the file does not yet exist.
5. Add a storage driver to the file, type:

```
{
  "storage-driver": "devicemapper"
}
```

6. Save the file.
7. Restart the Docker daemon, type:

```
# systemctl start docker.service
```

8. Verify that the storage driver is set to device mapper, type:

```
# docker Info
```

System Response:

```
Containers: 2
 Running: 1
 Paused: 0
 Stopped: 1
Images: 2
Server Version: 17.04.0-ce
Storage Driver: devicemapper
```

Verify that the docker containers are active, type:# docker ps

System Response:

```
CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES
bb953565e866  abc/rh:v1   "bash" 4 weeks ago Up 4 weeks jdoe
```

Administering the Docker Host

To protect data inside Docker images or containers, you need to create GuardPoints in the DSM, inside a Docker image or container, to which a VTE Agent policy is applied.

This section describes the administrative tasks related to Docker hosts; creating policies, creating GuardPoints, configuring audit logs, and generating reports.

Creating Policies

Policy creation for Docker hosts is largely the same as VTE's existing policy creation procedure with a few differences described below.

The basic procedure to create a policy is as follows:

1. Log on to your DSM as an administrator of type Security, Domain and Security, or All.
2. Navigate to **Policies > Manage Policies**.
3. Click **Add** to open the *Add Online Policy* page
4. From the **Policy Type** drop down list, select **Standard**.
5. Type in a name for the policy in the **Name** field.
6. Add a description for the policy (optional).

Refer to "Configuring Policies" in the *DSM Administrators Guide* for details about creating a policy. Once you have created a policy, you must add rules to the policy to encrypt data and control access to files and directories.

Adding Security Rules

This section describes how to create security rules in the context of Docker images and containers.

Create Resource Set

A *Resource Set* specifies the hosts, files, and directories to which a user or process will be permitted or denied access.

1. After creating the policy in the previous section, click **Add** in the **Security Rules** panel to open the *Add Security Rule* page.
2. Click **Select** next to the **Resource** field to open the *Select Resource Set* page. The page displays resource sets if any currently exist.
3. Select an existing resource set that meets your requirements, otherwise click **Add** to open the *Add Resource Set* page.
4. Enter a name for the resource you want to create.
5. Click **Add** again to open the *Add Resource* page to add a resource to the resource set you want to create.
6. Click **Select** next to the **Host** field to select a Docker host from which to choose resources. Select the radio button next to the host and click **Select**.
7. Since this a Docker host, another field displays: **Docker Image**. Click **Browse** to open the **Remote Docker Browser** to select a Docker image or container, from which to select a resource.
8. Click **Browse** next to the **Directory** field to open the **Remote File Browser**.
9. Browse the directories and files on the image or resource that you want to add to the resource set. (Select **Directory Only** or **Directory and File** to browse only directories or files and directories.)
10. Select the resources to add and click **Ok**.
11. Click **Ok** to add the resource set. The *Select Resource Set* page opens
12. Select the resource you just created and click **Select Resource Set**.
13. Check the **Exclude** box, to the right of the **Resource** field.

This excludes the resources in the resource set and includes all other host resources. Uncheck the box to include just the resources in the resource set.

Create User Set

A *User Set* specifies users that are permitted or denied access to files and directories in a GuardPoint.

1. After creating the policy in the previous section, click **Add** in the **Security Rules** panel to open the *Add Security Rule* page.
2. Click **Select** next to the **User** field to open the *Select User Set* page. The page displays user sets if any currently exist.
3. Select an existing user set if it meets your requirements, otherwise click **Add** to open the *Add User Set* page.
4. Enter a name for the user set and click **Add** to open the *Add User* page.
5. Enter information for the **uname**, **uid**, **gname**, **gid**, and **osDomains** fields. Refer to the online help for more details.
6. If you click **Browse Users**, the *Add Users* page opens, you can select users from an LDAP server if configured, or from a selected Host.
7. To select users from docker images or containers, use the default **Agents** selection and select the host name (FQDN) of the Docker host from the list.

   Since this a Docker host, another field displays: **Docker Image/Container**.
8. Click **Browse** to open the **Remote Docker Browser** to select a Docker image or container from which to select users.
9. From the **Remote Docker Browser**, expand the file icon to view the Docker image and containers from which to select users to add to the User Set.
10. Once you've made your selections, click **Ok**, a tabulated list of available users is displayed.
11. Select the appropriate users. Click **Ok** to return to the *Add User Set* page.
12. Select users. Click **Ok** to return to the *Select User Set* page.
13. Select the newly created user set and click **Ok**.

Create Process Set

A *Process Set* specifies the executables that are permitted or denied access to GuardPoint data.

1. After creating a policy in the previous section, click **Add** in the **Security Rules** panel to open the *Add Security Rule* page.
2. Click **Select** next to the **Process** field to open the *Select Process Set* page. The page displays process sets if any currently exist.
3. Select an existing process set if it meets your requirements, otherwise click **Add** to open the *Add Process Set* page.
4. Enter a name for the process set you are about to create and click **Add** to open the *Add Process* page.
5. Click **Select** next to the **Signature Set** field, a list of existing signature sets is displayed.
6. Select an existing signature set if it meets your requirements, otherwise click **Add** to open the *Add Signature Set* page.
7. If you selected **Add** then you need to provide a **Name** for the signature set.
8. Enter the name and click **Ok**, the *Signature Reference* page opens.
9. Click the name of your signature set to edit the signature set, the *Edit Signature Set* page opens.
10. Click the **Source** tab, click **Select** next to the Host field, the **Select a host to continue** page opens, select the Docker host and click **Select** to go back to the **Source** tab.

11. Click **Browse** next to the **Docker Image/Container** field, select a Docker image or container from the **Remote Docker Browser**.
12. Select a binary from the Docker image and sign it similar to a binary on a protected host.
13. Click **Back**, the *Signature Reference* page opens.
14. Select the signature set you just created and click **Select Signature Reference**, the *Add Process* page opens.
15. Select a host. Once you select a Docker host, the **Docker Image/Container** field displays.
16. Select a Docker image or container.
17. Click **Browse** and select a directory from the **Remote File Browser**, fill in the file name field as required. Click **Ok** to return to the *Add Process Set* page.
18. Choose the appropriate (newly created) process set and click **Ok**. This returns you to *Select Process Set* page.
19. Select the process set and click **Select Process Set**. This returns you to the Add Security Rule page.
20. Click **Ok** to add the process set to the security rule.

Enable Docker through Host Settings

You must enable Docker in the Agent.

1. At the DSM management console, click **Hosts > Hosts**.
2. Click **Add Host**, or click on an existing Host name to edit the host.
3. In the General tab, select **Docker Enabled**.
4. Click **Apply**.
5. Click **Host Settings** tab.
6. Add the Docker Daemon to the host settings:

   For Docker v1.12 and above, type:

```
|authenticator|/usr/bin/dockerd
```

   For Docker v1.12 and below, type:

```
|authenticator|/usr/bin/docker
```

7. Click **Apply**.

VTE Docker GuardPoints

Docker typically provides two types of containers:

- **Transient**: Run micro services which are stateless applications.
- **Long running**: Host stateful applications similar to a database application.

The VTE Docker security feature protects data in both type of containers. You can use VTE to protect either type of containers. You select the container type while configuring the GuardPoint.

VTE provides two types of GuardPoints:

- Image-based
- Container-based

Image-based GuardPoints

You can set up a data protection policy on a Docker image. After an image-based GuardPoint is created, all of the instances running from the protected Docker image inherit the policy and its settings. Any change to the policy is reflected across Docker containers that are started from protected Docker images. You can also refer to Image-based protection as templatized protection. The GuardPoints set up on a Docker image serves as a template for protection of all the Docker containers created as VTE protected Docker images.

Users can browse a Docker image to select the path for protection and configure security rules using information from a Docker image. This process is described in the DSM Guide.

Container-based GuardPoints

You can set up a GuardPoint for a specific Docker container. You can browse a Docker image to select the path for protection, and to configure security rules using information from a Docker image.

GuardPoints for Docker Containers

Before creating GuardPoints on Docker images and containers, the following must be taken into consideration:

- You must add the Docker engine process to the Host Settings.
- When applying GuardPoint policies to Docker containers, users must ensure that the root user has at least 'permit' effect on the GuardPoint. Otherwise, the GuardPoint is inaccessible to all users, even for users with ‘apply_key’, and ‘permit’ effects.

Creating GuardPoints

1. Log on to your DSM as an administrator of type Security, Domain and Security, or All.
2. Navigate to **Hosts**.
3. On the *Hosts* page, click the name of the host in the **Host Name** column, the *Edit Host* page opens.
4. Click the **Guard Docker** tab.
5. Click **Guard** to open the *Guard File System* page.
6. Select a policy to apply to the GuardPoint you are about to create.
7. Click **Browse** next to the **Docker Image/Container** field to browse the Docker host for an image or container to which to apply the policy.
8. Select the type of directory to guard.
9. Click **Browse** next to the **Path** text box to browse the image or container for a file path to add the GuardPoint.
10. Click **Ok**, the *Edit Host* page opens with the newly created GuardPoint listed in the table.

Note: Automount is not supported in a Docker environment.

Viewing GuardPoints

You can view GuardPoints from the Management Console GUI and from the Docker host using the VTE Agent `secfsd` utility. To view GuardPoints using the `secfsd` utility:

1. Log on to your Docker Host as root.
2. At the prompt, type:

```
# secfsd -status guard -tree
```

The output is displayed in a tabular format. The table displays the following information:

- GuardPoint location on the image or container
- Name of the policy applied to the GuardPoint
- Type of directory being guarded
- Container ID
- GuardPoint configuration status; whether or not the GuardPoint has been enabled
- GuardPoint status; whether or not the GuardPoint is currently guarded or not
- Reason for the GuardPoint not being guarded

To view information for each GuardPoint;

1. Log on to your Docker Host as root.
2. At the prompt, type the following;

```
# secfsd -status guard -v
```

The output is displayed for each GuardPoint configured on the host. The following information is displayed for each GuardPoint;

- Name of the policy applied to the GuardPoint
- Directory to which the GuardPoint is applied
- Type of directory being guarded
- GuardPoint configuration status; whether or not the GuardPoint has been enabled
- GuardPoint status; whether or not the GuardPoint is currently guarded or not
- Reason for the GuardPoint not being guarded
- Space usage on the GuardPoint location
- Container ID

Data Security for Docker Images and Containers

The VTE Agent supports data security for directories within Docker images and containers. If a new GuardPoint is added to a directory withing an image or container, and that GuardPoint contains data, that data must be transformed before VTE can apply an encryption policy. Therefore, before creating a GuardPoint, determine which of the following conditions is applicable to your situation.

Setting up an image based GuardPoint

If you are setting up an image based GuardPoint:

1. Create a container from that image using the `docker run` command. For example;

   ```
   # docker run Ubuntu
   ```

   VTE creates a container <*container_name*>.
2. Export the container to a TAR file in a directory using the `docker export` command. The following example creates a directory and exports the TAR file:

   ```
   # mkdir -p /tmp/export/GP1

   # docker export <container_name> > /tmp/export/GP1/<container_name>.tar
   ```
3. Extract TAR file using the command *tar -xvf*. The following example creates a directory and extracts the TAR file to that directory;

   ```
   # mkdir -p /tmp/extract/GP1

   # tar -xfv /tmp/export/GP1/<container_name>.tar -C /tmp/extract/GP1/
   ```
4. Guard extracted folder with a data transform policy, for example guard a folder with sensitive data under /tmp/extract.
5. Transform files using VTE.
6. Unguard the transformed folder.
7. Create a TAR file from the extracted files using the command `tar -czf`. The following example creates a directory, a TAR file and places it in that directory;

   ```
   # mkdir -p /tmp/import/GP1

   # cd /tmp/extract/GP1

   # tar -czf  /tmp/import/GP1/<container_name>.tar *
   ```
8. Import the TAR file back to the image using the Docker command `docker import`. For example,

   ```
   # cat /tmp/import/GP1/<container_name>.tar | docker import - <image>

   sha256:d303c8b357084291f02f5e033660f469705f8e2c1744bf14516783b58b0f3bd7
   ```

   The image <*imageName*> is created in this example, and its ID is;
   sha256:d303c8b357084291f02f5e033660f469705f8e2c1744bf14516783b58b0f3bd7
9. You can now guard directories within this image or container with a production policy.

Note: This procedure only works with an image's local file system and cannot transform a Docker data volume or NFS mount.

Note: Data transformation only occurs on the directory that is guarded, and not the entire Docker image.

Setting up a Container-Based GuardPoint

If you are setting up a container based GuardPoint, stop the Docker container before setting up GuardPoints:

1. Export the container to a TAR file in a directory using the `docker export` command. The following example creates a directory and exports the TAR file;

```
# mkdir -p /tmp/export/GP1

# docker export <container_name> > /tmp/export/GP1/<container_name>.tar
```

2. Extract TAR file using the command `tar -xvf`. The following example creates a directory and extracts the TAR file to that directory;

```
# mkdir -p /tmp/extract/GP1

# tar -xfv  /tmp/export/GP1/<container_name>.tar -C /tmp/extract/GP1/
```

3. Guard extracted folder with a security policy.
4. Transform files using the VTE.
5. Unguard the transformed folder.
6. Create a TAR file from the extracted files using the command `tar -czf`. The following example creates a directory and creates a TAR file and places it in that directory:

```
# mkdir -p /tmp/import/GP1

# cd /tmp/extract/GP1

# tar -czf  /tmp/import/GP1/<container_name>.tar *
```

7. Import the TAR file back to the image using the Docker command `docker import`. For example,

```
# cat /tmp/import/GP1/<container_name>.tar | docker import - <image>

sha256:d303c8b357084291f02f5e033660f469705f8e2c1744bf14516783b58b0f3bd7
```

   The image, *<image>* is created in this example, and its ID is;
   sha256:d303c8b357084291f02f5e033660f469705f8e2c1744bf14516783b58b0f3bd7
8. You can now guard this image or container with a production policy.

Note: This procedure only works with an image's local file system and cannot transform a Docker data volume or NFS mount.

Note: Data transformation only occurs on the directory that is guarded, and not the entire Docker image.

Setting up a GuardPoint for an Exported Docker Volume

If you are setting up a GuardPoint for a Docker volume exported from a Docker host:

1. Guard a folder on a host with a data transform policy.
2. Transform files using the Vormetric executable dataxform.

3. Unguard the folder on the host.
4. You can now guard this image or container with a production policy.

This approach works with Docker data volume and NFS mount, and can not transform the image/container local file system.

Configuring Audit Logging

Configure Log settings for the VTE Agent (FS Agent Log) at the System level on the DSM. These settings are inherited by all the domains on the DSM. However, you can fine tune log settings for a specific host, and those settings will override the system settings. With the introduction of Docker support, you can now configure log settings for Docker images and containers. Docker logs evaluate GuardPoint policies.

Configure Docker Log Settings

1. Log on to your DSM and switch to a domain. Alternately, log on to a DSM as a local domain administrator of type Security with a Host role.
2. Navigate to the **Hosts** page.
3. Click the name of your Docker host in the **Host Name** column, the *Edit Host* page opens.
4. Click **Docker Log**.
5. Enter the following information in the **Configure Docker Log Setting** panel:
   a. **Docker Image/Container**: Click **Browse** to select an image or container from the Docker host. If you select an image, the **Docker Image ID** field displays the image ID. If you select a container, the **Docker Image ID** field displays the image from which the container was created, and the **Docker Container ID** displays the container ID. You can use these IDs to search for Docker specific logs on the *Logs* page later.
   b. **Policy Evaluation Level**: Select a log message level. For more information about log levels, refer to the *DSM Administrators Guide*.
   c. **Policy Evaluation Duplicated**: Select to suppress or allow duplicate messages. The default is SUPPRESS.
6. Click **Ok**. VDS saves the Policy Evaluation settings in a tabular format under the **Configure Docker Log Setting** panel.

Searching for Docker Log Messages

Docker log messages display on the Logs page. To search for Docker specific log messages:

1. Click **Logs > Logs**.
2. Enter the following information in the **Search** panel:

- **Log Type**: Select whether you want to display logs from both the DSM and the agents, only the DSM, or only the agents. The default is All, which means from both DSM and agents.
- **Source**: Enter the hostname of the DSM server, or agent, for which you want to return log files.
- **Last Refreshed**: Displays the date and time of when the displayed log files were last refreshed. Format is YYYY-MM-DD HH:MM:SS
- **Message Contains**: Type in the text string that you want to search for in the log messages.
- **Docker Host**: Click **Browse** to select the Docker Host for which you want to return log files.
- **Docker Image/Container**: Click **Browse** to select an image or container for which you want to display logs.
- **Docker Image ID**: Displays the ID for the selected Docker image.
- **Docker Container ID**: Displays the ID of the selected Docker container.

3. Click **Go**. The relevant logs display in the table under the **Search** panel.

Generating Reports

The following DSM reports have been updated to include Docker information:

System Level Reports

To view system level reports, log on to a DSM as an administrator of type System or All. The system level reports that contain Docker information are:

- **System License Usage Summary**: Includes information about the total number of Docker licenses in use for the entire DSM.
- **License Usage by Domain**: Includes information about the total number of hosts with the Docker license enabled.

Domain Level Reports

To view domain level reports, log on to a DSM as an administrator of type Domain, Domain and Security, or All. Administrators of type Domain and Security and type Security must have AUDIT role privileges to access the reports. The domain level reports that contain Docker information are:

- **License Usage by Domain Summary**: Includes information about the total number of hosts with the Docker licenses enabled.
- **Host with GuardPoint Status**: Includes identification information about Docker Images and Docker containers that have GuardPoints. The columns are **Docker Image ID** and **Docker Container ID**.

You can download and save all reports locally in CSV format by clicking **Download**.

RedHat OpenShift Containers with VTE

Red Hat OpenShift Container Platform (OCP) provides an immutable, container-based platform to deploy and run applications and micro services. It is Red Hat's on-premise private PaaS product. It is built around a core of application containers powered by Docker Container Packages and Kubernetes Container Cluster Management, on a foundation of Red Hat Enterprise Linux.

Using the VTE Agent

In order to use the VTE Agent to protect OCP images and containers, you must obtain a VTE Agent license for Docker. The Docker license covers both Docker and OCP. Contact Vormetric Support for information on obtaining a license.

There is no change to the VTE installation, however, VTE agent must be installed on the OCP host system.

VTE: Virtual Machine versus OCP

Similarly to Docker, VTE installs on the OCP container host and orchestrates security from the OCP host. It does not leave a footprint inside the OCP image or containers. You setup the GuardPoints inside the individual containers at runtime.

Set the OpenShift Storage Driver

Although needed for Docker, this is not needed for OCP.

Administering the OpenShift Host

To protect data inside OCP containers, you need to create GuardPoints in the DSM, inside an OCP image or container, to which a VTE Agent policy is applied.

The administrative tasks related to OCP hosts are the same as for Docker hosts. See “Administering the Docker Host” for more information.

Enable OpenShift through Host Settings

If you have obtained a Docker license and enabled it through the DSM Host Settings, then OCP is also enabled.

VTE OCP GuardPoints

OCP typically provides two types of containers:

- **Transient**: Run micro services which are stateless applications
- **Long running**: Host stateful applications similar to a database application.

The VTE OCP security feature protects data in both type of containers. You can use VTE to protect either type of containers. You select the container type while configuring the GuardPoint.

Types of GuardPoints

VTE provides two types of GuardPoints:

- Image-based
- Container (POD)-based

Image-based GuardPoints

You can set up a data protection policy on an OCP image. After an image-based GuardPoint is created, all of the instances, running from the protected OCP image, inherit the policy and its settings. Any change to the policy is reflected across OCP containers that are started from protected OCP images. You an also refer to Image-based protection as templatized protection. The GuardPoints set up on an OCP image serves as a template for protection of all of the OCP containers created as VTE protected OCP images.

Users can browse an OCP image to select the path for protection and configure security rules using information from an OCP image. This process is described in the DSM Guide. It is the same procedure as for a Docker image.

Container/POD-based GuardPoints

You can set up a GuardPoint for a specific OCP container in the same manner that you do for a Docker container. You can browse an OCP image to select the path for protection, and to configure security rules using information from an OCP image.

Creating GuardPoints

Create GuardPoints in the same manner as for Docker.

Viewing GuardPoints

View GuardPoints in the same manner as for Docker.

Data Security for OpenShift Images and Containers

The VTE Agent supports data security for directories within OCP images and PODS/containers. If a new GuardPoint is added to a directory withing an image or container, and that GuardPoint contains data, that data must be transformed before VTE can apply an encryption policy. Therefore, before creating a GuardPoint, determine which of the following conditions is applicable to your situation.

Setting up an Image-based GuardPoint

Set up a GuardPoint in the same manner as for Docker.

Setting up a POD-based GuardPoint

Set up a POD GuardPoint in the same manner as for Docker GuardPoints. Select the POD as you would a container.

Setting up a GuardPoint for an exported OCP volume

Set up a GuardPoint in the same manner as for exported Docker volume.

Configuring Audit Logging

Configure Audit logging in the same manner as for Docker.

Generating Reports

Generate Reports in the same manner as for Docker.

Creating an OCP Project in CLI with API Commands

Creating an OCP Project with a Template

You can create an OCP project in the CLI as well as in the GUI.

Note: RedHat OpenShift is OpenSource technology. Therefore, commands, references and documentation are subject to change. Thales is providing CLI commands that are current at this time. Thales cannot guarantee the integrity of these commands. If commands no longer work, consult the OpenShift developer documentation located at: https://docs.openshift.com/enterprise/3.2/cli_reference/basic_cli_operations.html#cli-reference-basic-cli-operations

1. Login to the OCP server with a user name and password, type:

```
# oc login <ocp-server> -u <username> -p <password>
```

2. Create a new project, type:

```
# oc new-project <project-name>
```

3. Add an instant application template to your project and deploy it, type:

```
# oc process openshift <instant-app-template-path> | oc create -f -
```

Example

```
# oc process openshift//django-psql-example | oc create -f -
```

Deploying an OCP Project

Run the following commands after deployment completes.

1. Get pod details, type:

```
# oc get -o name pods
```

2. Parse JSON output and get container details, type:

```
# oc get -o json <pod-name>
```

3. Create directory to be guarded in containers

```
# oc exec <pod-name> -c <container-name> -- <command>
```

```
# oc exec postgresql-1-bag25 -c postgresql -- mkdir /var/tmp/gp
```

4. Guard path inside all containers and inside all pods, type:
   - Container-based guarding

     ```
     # vmssc host addgp -d <guard-path> -p <policy> -c <container-id> -i <image-id> <hostname>
     ```

   - Image-based guarding

     ```
     # vmssc host addgp -d <guard-path> -p <policy> -i <image-id> <hostname>
     ```

Note: You can also guard paths in the DSM UI.

5. Unguard all guarded paths, type:
   - Container-based guarding

     ```
     # vmssc host delgp -d <guard-path> -p <policy> -c <container-id> -i <image-id> <hostname>
     ```

   - Image-based guarding

     ```
     # vmssc host delgp -d <guard-path> -p <policy> -i <image-id> <hostname>
     ```

Note: You can also guard paths in the DSM UI.

6. Delete project, type:

```
# oc delete project <project-name>
```

Available OpenShift commands

Commands	Function
clusterresourcequota	Create cluster resource quota resource.
configmap	Create a configmap from a local file, directory or literal value
deployment	Create a deployment with the specified name.
deploymentconfig	Create deployment config with default options that uses a given image.
identity	Manually create an identity (only needed if automatic creation is disabled).
imagestream	Create a new empty image stream.
namespace	Create a namespace with the specified name
policybinding	Create a policy binding that references the policy in the targetted namespace.
quota	Create a quota with the specified name.
route	Expose containers externally via secured routes
secret	Create a secret using specified subcommand
service	Create a service using specified subcommand.
serviceaccount	Create a service account with the specified name
user	Manually create a user (only needed if automatic creation is disabled).
useridentitymapping	Manually map an identity to a user.

Available OPC Options

Options	Parameter	Function
-f	--filename	Filename or URL to file to read a template.
-l	-labels	Label that you can set in all resources for this template.
-o	-o, --output='json'	Output format. It is either: describe|json|yaml|name|template|templatefile.
-o	--output-version	Output the formatted object with the given version (default api-version).
-o	--parameters=false	Do not process but only print available parameters.
-o	--raw=false	If true, output the processed template instead of the template's objects. Implied by -o describe.
-t	--template	Template string or path to template file to use when -o=template or -o=templatefile. The template format is golang templates. [http://golang.org/pkg/text/template/#pkg-overview]
-v	-v, --value=[]	Specify a key-value pair (ex: -v FOO=BAR) to set/override a parameter value in the template.

Container secfsd Utilities

Move to the Container-appropriate name space for the instance.

Use the following commands for more information.

```
# secfsd -[command] [option]
```

Commands	Function
`-status guard [-v/-tree]`	list all GuardPoints
`-status keys`	show current encryption key state
`-status auth`	list authentication settings
`-status lockstat`	show status of system and agent lock
`-status logger`	list logging details
`-status policy`	list configured policies
`-status pslist`	list protected process
`-status devmap`	list guarded devices
`-guard path [containerID]`	manually guard path
`-unguard path [containerID]`	manually unguard path
`-version`	show version of kernel module secfs2
`cmd -c debug.<level>.[on|off]>`	set debug logging on/off
`-debug <on|off>`	enable verbose logging
`-help`	Displays this help message

Chapter 11: NetApp Snapshot Directory

This chapter describes SecFS support for NetApp .snapshot directory over NFS. It contains the following sections:

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 exists 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 VTE. VTE guards the client directory mounting the OnTap data volume over NFS.

Note: 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 12: Secure Start

This chapter describes encrypting an Microsoft Active Directory (AD) with the Secure Start feature. It contains the following sections:

Secure Start Overview

Secure Start offers data protection for applications which start earlier in the boot sequence than VMD (Vormetric Daemon). For example, the Microsoft Active Directory system service starts very early in the boot sequence. To determine if another application qualifies, contact Thales technical support.

Note: Secure Start is included with VTE v6.0.2. You do not have to purchase it separately.

Note: Secure Start is supported on Windows Server 2008 R2 and later versions. It is not supported on Linux.

In VTE v6.0.1 and below, VTE cannot encrypt the AD system service because it boots earlier in the boot sequence than the VMD agent service. After it boots, VMD makes a secure connection with the DSM and retrieves the encryption keys from the DSM. Therefore, since AD boots before that, the AD system service is not encrypted.

VTE offers Secure Start through a new type of GuardPoint. A Secure Start GuardPoint starts before the AD services, and can, therefore, encrypt those services.

There are three methods for encrypting the directory:

- "Encrypt by Moving the AD Service into a Guarded Directory" on page 142
- "Encrypt Data in Place with Offline Transformation" on page 144
- "Encrypt with an LDT Transformation Policy" on page 144

Prerequisites

Prior to using Secure Start to guard your AD database:

1. Backup your AD database:
   a. Navigate to **Administrative Tools**.
   b. Click **Windows Server Backup.**
   c. Click **Action > Backup Once**.
   d. Follow the instructions in the Backup Wizard to create a backup of the server in a local drive.

Note: When the backup operation completes, it saves the server backup in <*backup drive*>:\WindowsImageBackup\<*BackupComputerName*>.

2. Perform a system state backup.
3. Obtain the Microsoft DSRM (Data Services Restore Mode) password.
4. Ensure that your AD database is not in `c:\Windows\NTDS`.

WARNING
Do not put your AD database in `c:\Windows` or `c:\Program files`. Secure Start cannot encrypt or decrypt any files in those folders.

Encrypt by Moving the AD Service into a Guarded Directory

You can move the AD service into a directory protected by a standard or LDT production policy. This method does not require the initial data transformation step. When you move the AD service into this directory, VTE immediately encrypts the data with either policy.

Note: This step occurs when the system is in DSRM mode, so users have no access to the AD service.

Create the AD GuardPath directory

Create the directory in which the AD service will reside.

1. Log in to the Active Directory Server in DSRM mode using the DSRM password. User ID is Administrator.
2. Create a folder to which you will move the AD database.

Apply Secure Start GuardPoints to a Directory

Access to a Secure Start GuardPoint is only permitted during the boot sequence and for a short period of time. Once the VMD is up and running, it performs the normal agent initialization and communicates with the DSM to access files within a GuardPoint location.

To apply Secure Start GuardPoints:

1. In the DSM, click **Hosts > Hosts >** *<hostName>*
2. In the General host information section, select the option: **Secure Start GuardPoint**.
3. Click **Guard FS.**
4. Select the directory and click **Guard**.
5. In the Policy field, select an **LDT** or **Standard Production** policy.
6. Set Type to **Directory (Auto Guard)**.
7. Click **Browse** and navigate to the folder that you just created for the AD database.
8. Select the option: **Secure Start**.
9. Click **OK**.
10. Select the GuardPoint and click **Secure Start On.**

Verify the Secure Start GuardPoint with CLI

After the DSM pushes the policy to the Active Directory Server, verify the GuardPoints.

To verify the GuardPoints, type:

```
> voradmin ss verify <GuardPoint_path>
```

System Response

```
Successfully completed the command verify
Success from kernel -Successfully verified the secure start GP
```

Move the AD Database into the Secure Start GuardPoint

Move your AD database from the default location (`c:\windows\NTDS`) to this newly created protected folder.

To move the AD database:

1. In DSRM mode, login using the DSRM password. User ID is Administrator.
2. Start NTDSUTIL utility, type:

   ```
   > activate instance ntds
   ```

3. Type:

   ```
   > files
   ```

4. Type:

   ```
   > move db to \<GuardPoint>
   ```

5. Type:

   ```
   > move logs to \<GuardPoint>
   ```

6. Exit NTDSUTIL utility.
7. Reboot the system into normal mode. The Active Directory Services automatically starts after rebooting.

Note: This step occurs when the system is in DSRM mode, so users have no access to the AD service.

Encrypt Data in Place with Offline Transformation

Encrypting the AD database with a standard (production), or offline policy is very similar to encrypting other data with a standard (production), or offline policy. For information on standard policies and encrypting data offline or with a production policy, see the *VTE Data Transformation* guide.

The advantage to encrypting data in place is that it saves space. When you copy/move a directory into a guarded directory, you will need twice as much space to store the data because you leave a copy of the data in the original folder, as a precaution, until the original directory has been successfully moved and encrypted. Once the data is transformed, then you can delete the directory that contains the decrypted/clear data.

Using this method, you perform an Initial Data Transformation using the `dataxform` command line utility. During this transformation, access to the GuardPoint data is blocked. After initial transformation, you remove the initial policy, and then apply a production policy, so users can access the data.

Note: This step occurs when the system is in DSRM mode, so users have no access to the AD service.

Note: If your AD service is installed in the default directory, `C:\Windows\NTDS`, you must move it to another directory before you can encrypt it. See "Encrypt by Moving the AD Service into a Guarded Directory" on page 142 for more information.

To encrypt the data:

1. In DSRM mode, login using the DSRM password. User ID is Administrator.
2. Create and apply a `dataxform` policy to the GuardPoint directory.
3. Run the `dataxform` command.
4. Remove the `dataxform` policy on the GuardPoint and replace it with a production policy.
5. Reboot out of DSRM mode.

Encrypt with an LDT Transformation Policy

Encrypting the AD database with an LDT policy uses the same steps as encrypting with a standard production policy. The only difference is that you select an LDT policy instead of a standard one. See "Encrypt by Moving the AD Service into a Guarded Directory" on page 142 for more information.

Note: If your AD service is installed in the default directory, `C:\Windows\NTDS`, you must move it to another directory before you can encrypt it.

Configure the Time Out Failure

During the initial access to a Secure Start GuardPoint, the VTE agent sets a timer. The default duration is 120 seconds, but you can configure the duration. Minimum duration is one second, maximum duration is 300 seconds.

Data inside the GuardPoint is accessible without DSM connectivity until the timeout is reached. VMD service activates and makes a secure connection to the DSM. After the VMD makes a secure connection, the agent verifies that it is connected to correct DSM. If the VMD fails to connect to the DSM, the timeout is reached, and if AD is installed, the agent shuts down the system for data security purposes.

Note: In DSRM mode, when the timeout occurs, VTE removes the keys from memory. However, VTE does not shut down the system.

In normal mode, VTE shuts down the AD server. For any other application, or if AD is not installed, Secure Start does not shut down the server. However, the data inside the GuardPoint becomes inaccessible until DSM connectivity is restored, or you issue a challenge/response, or password. After the timer has expired, VTE denies any further access to the Secure Start GuardPoint.

1. To configure the timeout duration in seconds, type:

   `C:\> voradmin ss settimeout <`*timeout*`>`

   Example:

   `C:\> voradmin ss settimeout 220`

   System Response:

   ```
   Successfully completed the command settimeout
   Successfully set the Secure Start timeout value to 220 seconds
   ```

   To verify the timeout duration, type: **`C:\> voradmin ss gettimeout`**

   System Response:

   ```
   Successfully set the Secure Start timeout value to 220 seconds
   ```

Recover a Server After it Loses Connection to the DSM

User must unlock the GuardPoint by entering the Challenge/Response or password to restore the DSM connectivity. Once the GuardPoint is unlocked, you can start the AD services manually.

Note: The challenge response pop up dialog does not display in DSRM mode when the host loses DSM connectivity.

To activate the challenge/response:

1. Navigate to:

   ```
   C:\Program Files\Vormetric\DataSecurityExpert\agent\shared\bin
   ```

2. Double-click `etray.exe` to start it manually.

If the AD server does not connect to the DSM, then the AD system automatically shuts down. The Administrator must enter DSRM mode and restore the DSM connectivity to recover the server.

Prerequisites

Before rebooting your active directory servers:

- Ensure that DSM connectivity is strong. If it is not strong, restore the DSM connectivity.

Note: When trying to fix a DSM connectivity issue, you can log in to DSRM mode. In DSRM mode, there is no requirement to increase the timeout, because in DSRM mode, the AD system does not shut down after timeout expires.

DSRM Mode

The first method for recovering a server relies on manual DSM connection troubleshooting:

1. Boot into DSRM mode.
2. Attempt to resolve why the server is not connecting to the DSM.
3. Fix that DSM connectivity issue.
4. Reboot into normal mode.

Other Use Cases

Using Secure Start GuardPoints, you can also secure an SQL Server on Microsoft Azure in certain scenarios. SQL system services in Azure also boot earlier in the boot sequence than the VMD (Vormetric Daemon) agent service.

Note: To determine if another application qualifies, contact Thales technical support.

Boot a Windows Server in Azure

To move and guard the AD database, you must boot the AD server into DSRM mode.

To boot a Windows Server 2012/2016 Domain Controller into DSRM remotely in Azure:

Note: The Windows Server 2012/2016 domain controller must be running and accessible through Windows Remote Desktop.

1. Establish a Remote Desktop session on the domain controller.
2. Open an command prompt as Administrator and type:

```
> bcdedit /set safeboot dsrepair
```

3. Reboot the domain controller. The Remote Desktop session disconnects.

4. Wait a few minutes, then establish a new Remote Desktop session. The domain controller will be running in DSRM.
5. To reboot into normal mode, open an command prompt as Administrator and type:

```
> bcdedit /deletevalue safeboot
```

6. Reboot the domain controller.

Best Practices for Encrypting and Protecting the AD Service

Thales recommends the following best practices when using Secure Start with an AD service.

Access Control with Secure Start

User can setup a restricted access control policy with encryption to prevent the unauthorized access of AD database files. The restricted policy with Secure Start:

1. Prevents a rogue user from logging into the system, and moving or copying the AD database files to another directory and tampering with it.
2. Denies permissions, after you setup and guard files, so that no one can move a file from the guarded directory. Plus it restricts any other unwanted/unnecessary process or users from tampering with AD files.
3. Provides permission for an authorized user who needs access to AD services and files.

Creating a Minimal Policy Required for AD with Access Control

When creating a normal, strict policy for access control, you must allow access to the following processes and directories for Active Directory.

Processes

```
secfsd.exe (C:\Program Files\Vormetric\DataSecurityExpert\agent\secfs\ sec\bin\)
lsass.exe (C:\Windows\System32\)
vds.exe (C:\Windows\System32\)
vssvc.exe (C:\Windows\System32\)
wbengine.exe (C:\Windows\System32\)
ntoskrnl.exe (C:\Windows\System32\)
```

Users

```
NT AUTHORITY\SYSTEM
```

To create a minimal policy:

1. Create a User Set named: **AD_Minimum_User_Set**.
2. Set with the following parameters:

ID	uname	osDomains
1	SYSTEM	NT AUTHORITY

3. Create a Process Set named: **AD_Process_Set**.

4. Set with the following parameters:

ID	Directory	Base Name
1	C:\Program Files\Vormetric\DataSecurityExpert\agent\secfs\sec\bin	secfsd.exe
3	c:\Windows\System32\	ntoskrnl.exe
4	c:\Windows\System32\	vds.exe
5	c:\Windows\System32\	vssvc.exe
6	c:\Windows\System32\	wbengine.exe
7	c:\Windows\System32\	lsass.exe

5. Create a Security rule.
6. Set with the following parameters:

Order	User	Process	Action	Effect	Browsing
1	AD_Minimum_User_Set	AD_Process_Set	all_ops	Audit, Permit, Apply key	Yes
2				Audit, Deny	Yes

Creating a Restricted Policy in DSRM Mode

Create the following policy for the initial transformation of an AD database in DSRM mode. The policy allows access to the local administrator.

In DSRM mode, you use the `NTDSUTIL` utility to perform maintenance for an Active Directory.

To create a restricted policy:

1. Create a User Set named: **AD_Minimum_User_Set**.
2. Set with the following parameters:

ID	uname	osDomains
1	SYSTEM	NT AUTHORITY
2	Administrator	localhost

3. Create a Process Set named: **AD_Process_Set**.
4. Set with the following parameters:

ID	Directory	Base Name
1	C:\Program Files\Vormetric\DataSecurityExpert\agent\secfs\sec\bin	secfsd.exe
2	c:\Windows\System32\	ntdsutil.exe

ID	Directory	Base Name
3	c:\Windows\System32\	ntoskrnl.exe
4	c:\Windows\System32\	vds.exe
5	c:\Windows\System32\	vssvc.exe
6	c:\Windows\System32\	wbengine.exe
7	c:\Windows\System32\	lsass.exe

5. Create a Security rule.
6. Set with the following parameters:

Order	User	Process	Action	Effect	Browsing
1	AD_Minimum_User_Set	AD_Process_Set	all_ops	Audit, Permit, Apply key	Yes
2				Audit, Deny	Yes

Guard Directories

The best practice for guarding a directory with a Secure Start GuardPoint is to:

1. Create a directory.
2. Guard that directory with a standard production or LDT policy. Follow the steps in “Apply Secure Start GuardPoints to a Directory” on page 142.
3. Move the AD service into that directory.

Perform Subsequent System State Backups

After you move an AD service into a guarded directory, or out of a guarded directory:

1. Perform another system state backup.
2. Save this subsequent backup to a different location.

Chapter 13: Enhanced Encryption Mode

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

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 AGT-		
Local FS (Linux)	EXT3/EXT4/XFS	EXT3/EXT4/XFS
Remote FS (Linux)	NFS3/NFS4/ CIFS	NFS3/NFS4
Local FS (Windows)	NTFS/ReFS	NFS3/NFS4/CIFS
Remote FS (Windows)	CIFS	CIFS (if the backend storage for the CIFS share is Windows-based storage).
Block Device Support (secvm)	Fully supported	No. When a policy contains a key with CBC-CS1 encryption mode, the guarding fails on the DSM, and an error message displays.

Compatibility

VTE version 6.3.1 and later is backward compatible with and fully supports the existing AES-CBC mode, both for new and existing datasets, after the Agent is upgraded to VTE v6.1.0 or later.

CAUTION
AES-CBC-CS1 encryption is only supported with VTE 6.1.0 and later versions. A pre-v6.1.0 protected host is incapable of supporting AES-CBC-CS1. On these earlier versions, attempting to guard using a policy containing an AES-CBC-CS1 key will fail.

- AES-CBC-CS1 encryption is supported for LDT and offline dataxform on VTE Linux and Windows environments.
- Protected hosts supporting AES-CBC-CS1 encryption can be added to host groups

	AES-CBC	AES-CBC-CS1
Offline data transformation	Supported	Supported
Live transformation	Supported	Supported

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.

Note: 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 DSM, 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.”

Note: 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 FS (Linux)	No change to file size. No extended attribute allocation	Internal use of extended attribute per file. Extra 4KB increase in file size.
Remote FS (Linux)	No change to file size. No extended attribute allocation	Extra 4KB allocation in the form of an embedded header per file. With VTE guarding enabled, file size expansion is hidden.
Local FS (Windows)	No change to file size. No ADS allocation.	Extra 4KB allocation (at minimum) in the form of an embedded header per file. With VTE guarding enabled, file size expansion is hidden.

Encryption Migration

You can use both 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

File Systems Compatibility

On Linux and Windows,, 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, VTE masks the file header -- to applications and system utilities. The expanded file is only apparent when VTE 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 on Linux 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 VTE guarding. Similarly, AES-CBC-CS1 encrypted files on local file systems cannot be copied over to remote file systems in the absence of VTE 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 VTE guarding.

	AES-CBC	AES-CBC-CS1
EXT3/EXT4 on RHEL6	None	EXT3/EXT4 must be mounted with *user_xattr* 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 in either the extended attributes or in the file metadata. On local FS (EXT3/EXT4/XFS), it saves it as an extended attribute associated with the file. It saves the base IV of a file on remote FS (NFS and CIFS) in the embedded header of the file.
- AES-CBC-CS1 encrypted files on Windows stores the IV in a Windows ADS (Alternate Data Streams) associated with the file. The size required for saving the CS1 key depends on the allocation size of the file system. If the allocation size is set to 4k, then the new IV will require 4K of extra space on the disk. User can run `fsutil fsinfo` tool to find out the allocation size of the file system.
  - On Windows, CS1 key is supported on following file systems:
    - **NTFS**: Disks formatted with NTFS file system on all platforms
    - **REFS**: File system on Windows 2012 R2 and later
    - **CIFS**: If the backend storage for the CIFS share is Windows-based storage.

Note: Some network storage servers do not support multiple ADS associated with a 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 encryption for VTE Linux local file systems like XFS and EXT file systems store the file IV as a part of the file metadata. The underlying file system requires that you mount it with the extended attribute mount option.
- AES-CBC-CS1 depends on the physical file system's support for extended attributes in a manner similar to the LDT feature.

Missing IV file

If you are using the CBC-CS1 key, then each file has a unique IV associated with the file. This IV is stored in ADS on Windows.

If the IV for a file is missing, or VTE is unable to read the IV, then VTE 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 FS (Linux)	No change	Internal extended attribute for each file
Remote FS (Linux)	No change	4KB embedded header for each file
Local FS (Windows)	No change	Alternate Data Streams

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.

VTE 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
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

FileTable Support on Windows

The CBC-CS1 key does not support FileTables. This is because FileTables do not support alternate data streams. The CS1 key requires the ability to write the per-file IV into an alternate data stream on each file.

Container Compatibility

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

Using the new Encryption mode

Deploy the new encryption mode (AES-CBC-CS1) by using the new symmetric agent key type created in DSM 6.1.x or later:

1. In the DSM, click **Keys > Agent Keys > Keys.**
2. Click **Add**.
3. In the Encryption Mode dropdown, select **CBC-CS1**.
4. In the Algorithm dropdown, select **AES128** or **AES256** to create an AES-CBC-CS1 key.
5. Add the key to your policy.

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 transformationor 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

The following are the recommended practices for deploying host groups with AES-CBC CS1 keys:

- In a host group, do not deploy policies associated with AES-CBC and AES-CBC CS1 keys unless all hosts are intended to run VTE 6.1.0 or later versions.
- If VTE 6.1.0 and older VTE versions are intended to be a part of the same host group, Thales recommends that you use policies without AES-CBC CS1 keys.

Chapter 14: Exchange DAG

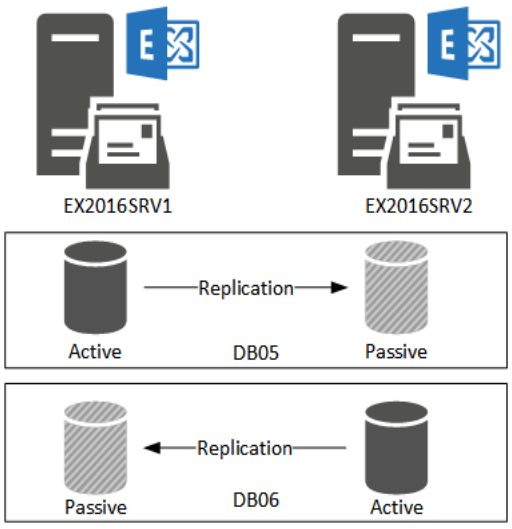
This chapter describes encrypting email databases using Microsoft Exchange database availability group (DAG). It contains the following sections:

Exchange DAG Overview

A DAG is a high-availability (HA) and data-recovery feature of the Microsoft Exchange Server. A DAG, which can consist of up to 16 Exchange mailbox servers, automates recovery at the database-level after a database, server or network failure. You can now use VTE to encrypt Exchange DAG mailboxes.

Note: Microsoft Exchange DAG is for Windows only. It is not compatible with Linux.

Figure 14-1: High Level Overview of Exchange DAG functions



You can encrypt the Exchange databases with a standard (offline) policy or an LDT policy. In an offline policy, users cannot access the database during initial data encryption. LDT is Live Data Transformation. It encrypts the data while users and applications are accessing files within it. LDT is used for Initial data transformation as well as transparent encryption/decryption.

Use Case tested and supported by Exchange DAG with VTE

Use VTE to encrypt Exchange DAGs in the following scenarios:

- Initial Data Transformation of Exchange Databases using:
  - Live Data Transformation
  - Standard Data Transformation

Note: For more information about LDT and Data Transform, request the guides for the software from Technical Support.

- Transparent Encryption/Decryption of the Exchange Database on DAG nodes
- Rotate the key using the Live Data Transformation policy

The following Exchange DAG operations were tested using the following use cases:

- Failover/Failback of databases from one node to another node and making both databases active on each node.
- Add new Databases to the existing nodes

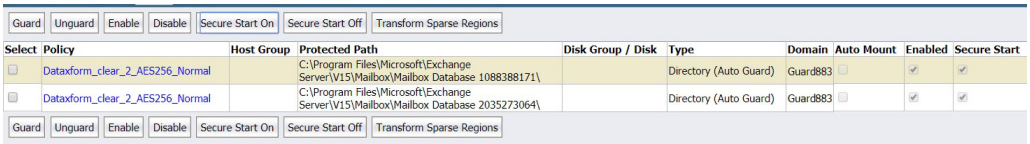
Recommendations

To use Microsoft DAG Exchange:

1. Disable all antivirus software as suggested by Microsoft.
   https://technet.microsoft.com/en-us/library/bb332342(v=exchg.160).aspx
2. Disable Windows defender:
   a. Type:

      ```
      > gpedit.msc
      ```

   b. Go to **Computer Configuration > Administrative Templates > Windows Components > Windows Defender**.
   c. Double-click and then, in the context menu, click **Turn Off Windows Defender**.
      For more information see https://www.windowscentral.com/how-permanently-disable-windows-defender-windows-10.
3. Only guard the Mailbox Database. Do not guard at a higher or lower directory.



4. Guard one host at a time. Do not apply GuardPoints to hostgroups.
5. Make sure that communication port 7024 is not blocked so that VTE agent can register with the host.

Requirements

The following is required for encrypting an Exchange DAG database with VTE :

- Windows Exchange Server 2016 Cumulative Patch 5
- Windows Server 2016 for Exchange Server nodes
- Windows 2016 Server as a file share witness
- Two nodes in the Exchange DAG configuration
- Both nodes on the same subnet

Note: Thales has not tested an environment with nodes on different subnets, but VTE supports this configuration. If you encounter any issues with different subnets, please contact Thales customer support.

- Use the same VTE Policy and keys on both nodes.
- You must enable Secure Start on the GuardPoints.
- All users must be offline when applying the initial GuardPoint, (even for LDT).

Note: Alternatively, you can choose to make all of the DBs active on one node and perform data transformation on the other node.

- Prepare your environment for suspending databases:
  a. Suspend the Exchange Database to stop replication and access to data.
  b. Disable replication between the nodes.

Note: No file access can happen within the target directory

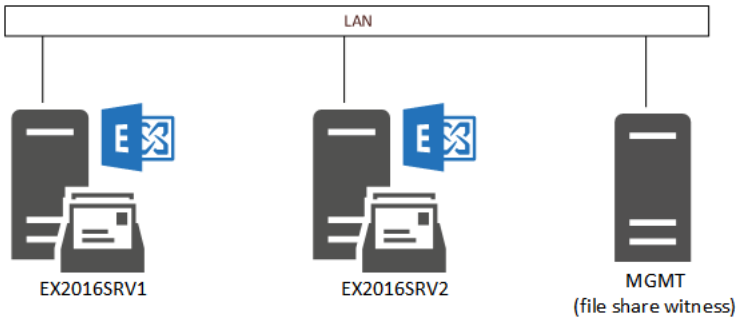


Figure 14-2: Proper Configuration for Exchange DAG

- Create Keys and Policies for the system

Note: You must use the same Keys and Policies on each node. For more information on Keys and Polices, see the *LDT User Guide* and the *VDS DSM Administrators* guide.

- Install VTE agent on the systems
- Register the agent with the DSM and make sure LDT is enabled, (if they are planning to use LDT. For more information on LDT, see the *LDT Users Guide*.)

Preparing your Exchange DAG Environment for Encrypting/Decrypting with VTE

The following describes how to prepare your environment for encrypting with Exchange.

To use Microsoft DAG Exchange with LDT to encrypt data:

1. In the **Exchange Admin Center**, make Exchange node 1 the primary node.

   Make node 1 the active node and move all of the databases to that node.

WARNING

Make sure that all of the Exchange services in node 2 are down and not accessing the Exchange databases. All Exchange Services must be stopped. All databases must be suspended.

2. Make all of the databases active on node 1.
3. Suspend all databases on node 2.

Note: Wait for 2-3 minutes for the database to finish with replication so the database will be suspended.

Encrypting with LDT in the Exchange DAG environment

You can use LDT for initial data transformation as well as transparent encryption/decryption.

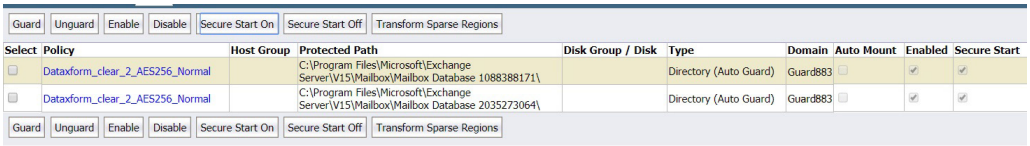
WARNING

You must guard the databases with the same Key/Policy on both nodes.

1. Make sure that you have prepared your environment properly. See “Preparing your Exchange DAG Environment for Encrypting/Decrypting with VTE” on page 162
2. Make sure that the GuardPoint is active on the host. Live Data Transformation starts on the server as soon as the GuardPoint is established.
3. Guard the Mailbox Database directory and apply the Live Data Transformation policy to the directory on node 2.
4. Select the GuardPoint and click **Secure Start On**.

WARNING

Only guard the Mailbox Database. Do not guard at a higher or lower directory.



WARNING

Guard one host at a time. Do not apply GuardPoints to both nodes using hostgroups.

5. In the **Exchange Admin Center**, go to the Exchange Database tab and **Resume** all Passive database copy on node 2.

Note: It may take a few minutes for the Exchange Service to resync. Monitor the Exchange logs on the system and make sure that replication is working. Make sure that database replication finishes and databases are in a healthy state before proceeding.

6. Wait for 10-15 minutes more for the server to move to the healthy state. If not, wait for some more time for the Content Index state to change to Healthy.
7. In the **Exchange Admin Center**, move all of the databases from node 1 to node 2.

   Now the databases on node 1 are mounted as passive. All databases on node 2 are mounted as active.
8. Repeat these steps on node 1.

Decrypting with LDT in the Exchange DAG environment

To use an LDT policy:

1. Make sure that the LDT state is set to REKEYED before unguarding.
2. Make sure that all of the files inside the GuardPoint are at the same version of the key.
   a. Run the LDT report to find the version:

      `> voradmin ldt report <GuardPoint path> [<logfile>]`

   b. Run the Key map report to find the version:

      `> voradmin ldt key [report|map] <key_name, version> <GuardPoint path>`

To use Microsoft DAG Exchange with LDT to decrypt data:

1. Make sure that all of the Exchange services in node 2 are down and not accessing the Exchange databases.

Note: Suspension can take 2-3 Minutes.

2. In the **Exchange Admin Center**, make Exchange node 1 the primary node.
   This means that node 1 is mounted as the active node and node 2 is mounted as the passive node.
3. Make all of the databases active on Exchange node 1.
4. Go to the Exchange Database tab and suspend all databases on node 2.
5. Unguard the database folders that you previously guarded on node 2.
6. Delete all of the metadata on all of the database folders on node 2, type:

   ```
   > voradmin ldt attr delete [<file name path> | <guard path>]
   ```

7. Guard with an LDT policy set for Encryption to Clear on node 2.

Note: You must clone the current version of the encryption key to use as the current key in the new LDT policy and clear key as the transformation key.

8. Go to the Exchange Database tab and resume all databases on node 2.

Note: After a few minutes, the databases should become healthy automatically. If not wait for the LDT process to decrypt the data. Make sure that all of the data is transformed back to clear and that the LDT state is set to **REKEYED**.

9. Move the database from node 1 to node 2.
10. Repeat Step 2. - Step 8. for node 1.

After both nodes are rekeyed and transformed from encryption to clear, you can unguard them.

1. In the **Exchange Admin Center**, make Exchange node 1 the primary node.
   This means that node 1 is mounted as the active node and node 2 is mounted as the passive node.
2. Make all of the databases active on Exchange node 1.
3. Go to the Exchange Database tab and suspend all databases on node 2.
4. Unguard the database folders that you previously guarded on node 2.

WARNING
Always ensure that you are unguarding a passive node.

5. Repeat Step 1. - Step 4. for Node 1.

Encrypting with a Standard VTE Policy in the Exchange DAG Environment

Refer to the following sections to prepare for encrypting in the Exchange environment.

To use Microsoft DAG Exchange with VTE:

1. In the **Exchange Admin Center**, make Exchange node 1 the primary node.
   This means that node 1 is mounted as the active node and node 2 is mounted as the passive node.

2. Make all of the databases active on Exchange node 1.
3. Go to the Exchange Database tab and suspend all database on node 2.

   Make sure that all of the exchange database services in node 2 are down and not accessing the Exchange databases.

Note: Suspension can take 2-3 Minutes.

4. Guard the mailbox database directory with a data transform policy.
5. Run dataxform.exe, type:

   ```
   dataxform --rekey --print_stat --gp <directory>
   ```

6. After the data transformation is finished, unguard the directory and guard with a Production policy. Apply a standard producton data transformation policy on only the node 2 database folders.

Note: Use the same Key/Policy on both nodes.

   See “Remove the dataxform policy and apply production policy” in the *VTE Agent DataXform* guide for more information.
7. In the **Exchange Admin Center**, go to the Exchange Database tab and resume all databases on node 2.

   After a few minutes, all nodes should become Healthy.
8. In the **Exchange Admin Center**, try to move a database from node 1 to node 2. If the data move is successful: this means that node 2 is mounted as the active node and node 1 is mounted as the passive node.
9. Repeat the previous steps for node 1.

Unsupported Use Cases

The following scenarios are not supported:

- Using different encryption keys on Exchange DAG nodes; both nodes must use the same encryption key
- Adding a new node to the Exchange DAG Environment
- Using the new CBC-CS1 key (not tested for this release)
- Encryption of Exchange Binaries
- Using nodes in a different subnet, data center or site, (Thales is not testing this scenario, but we do not believe it will cause any issues)

Chapter 15: Storage Spaces Direct

This chapter describes how VTE integrates with Windows Storage Spaces Direct (S2D) hyper-converged clusters.

S2D Overview

S2D uses industry-standard servers with local-attached drives to create high-availability (HA) software-defined storage. SD2 is included in Windows Server 2019 Datacenter and Windows Server 2016 Datacenter, both of which are supported by VTE.

S2D extends the stack of usable storage devices to storage devices such as SATA and SAS HDD's, SSD's and NVMe (Non-Volatile Memory Express) disks to create shared disk volumes. S2D supports clusters of a minimum of two nodes, and a maximum of 16 nodes and 400 drives. S2D aggregates the available storage into a Storage Pool.

The hyper-converged deployment option runs virtual machines on the servers providing the storage.

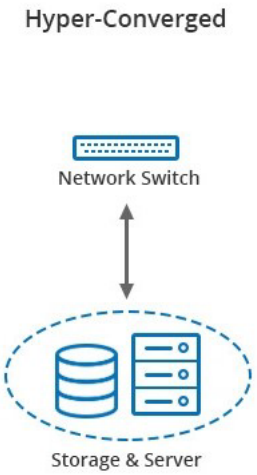
A complete description of the S2D product, and instructions on how to set up the S2D environment is available on the Microsoft website at https://docs.microsoft.com/en-us/windows-server/storage/storage-spaces/storage-spaces-direct-overview

Note: S2D is for Windows only. It is not compatible with Linux.

Deployment Options

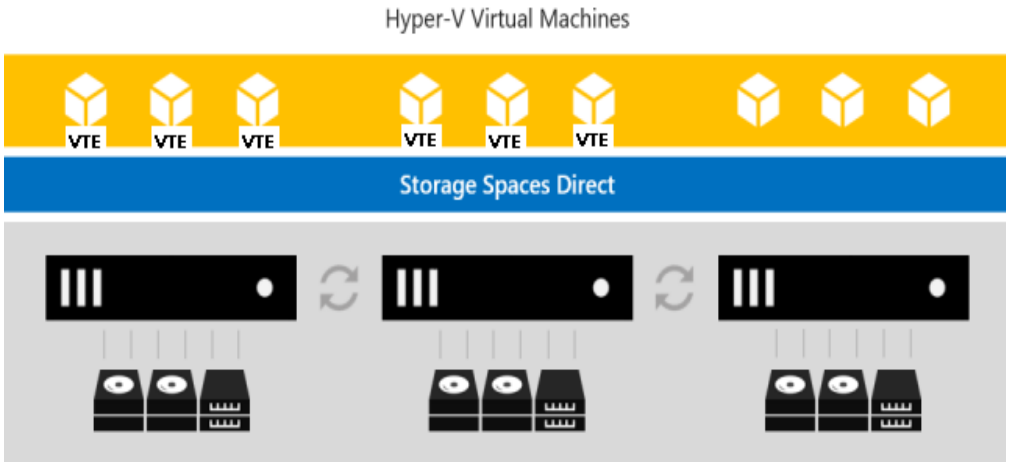
VTE supports S2D in a hyper-converged infrastructure where computing and storage components are in a single cluster as shown in the following figure.

Figure 15-1: Hyper-converged infrastructure



Hyper-converged with S2D and VTE virtual machines run on the servers providing the storage. In the following figure, VTE is installed inside 6 of the VMs to protect the data.

Figure 15-2: High-Level view of S2D



You can use all the capabilities of VTE to protect the data in the VMs in a S2D hyper-converged deployment. These capabilities are describe in “Supported Use Cases” on page 168.

Supported Use Cases

Thales tested only Hyper-converged deployments in the following scenarios.

- Initial Data Transformation of data using:
  - Live Data Transformation
  - Offline Data Transformation
- Transparent Encryption/Decryption of structure and unstructured data
- Key rotation using a Live Data Transformation policy

Chapter 16: Using VTE with Quantum StorNext

This chapter describes how to configure VTE and Quantum StorNext devices to interoperate to allow VTE policies to apply to storage managed by Quantum StorNext. This chapter contains the following sections:

Overview of using VTE 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 DSM policies by installing VTE agents on Linux or Windows 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).

VTE and Quantum StorNext Compatibility

The following sections list the supported operating systems and VTE settings supported for use with Quantum StorNext file systems. Important unsupported configuration parameters are also listed.

Supported StorNext Server and Client Configurations

The VTE integration with SNFS file systems works only with certain SNFS versions, SNFS storage policies, and client operating systems.

Configuration parameter	Linux	Windows
StorNext (SNFS) operating system version	6.x	6.x
StorNext metadata controller (MDC) server OS type	Linux MDC supported	Windows MDC supported
StorNext replication policy	Not supported	Not supported
StorNext deduplication policy	Not supported	Not supported
StorNext truncation	Supported	Supported
StorNext full and partial backup	Supported	Supported
StorNext expand filesystem	Supported	Supported
StorNext data migration	Supported	Supported
StorNext read-ahead cache	Disable for use with VTE	Disable for use with VTE
Client operating systems	Red Hat Enterprise Linux (RHEL) 7.x	Windows Server 2012 R2, Windows Server 2016

Configuration parameter	Linux	Windows
StorNext LAN client	DLC	DLC
StorNext mount method: locally mounted directory	Supported	Supported
StorNext mount method: Windows drive letter	Not applicable	Supported
StorNext mount method: CIFS	Not supported	Not supported
StorNext mount method: NFS	Not supported	Not supported

Supported GuardPoint and Key Settings for SNFS File Systems

When configuring VTE 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 will be by both Windows and Linux LAN clients.

Configuration element	Linux	Windows
Offline data transformation	Supported	Supported
Live Data Transformation (LDT)	Not supported	Not supported
DSM compatibility	See the *Vormetric Transparent Encryption Agent Compatibility Matrix* for your VTE version	See the *Vormetric Transparent Encryption Agent Compatibility Matrix* for your VTE version
Guard unstructured data	Supported	Supported
Guard structured data	Not supported	Not supported
GuardPoint type: Directory (including entire SNFS volume)	Supported	Supported
GuardPoint type: Raw device	Not supported	Not supported
GuardPoint type: Block device	Not supported	Not supported
GuardPoint mount option: manual guard	Supported	Not applicable
GuardPoint mount option: auto guard	Supported	Supported
GuardPoint mount option: automount	Not supported	Not applicable
AES-CBC key type	Supported	Supported
AES-CBC-CS1 key type	Supported	Not supported

Supported Concurrent Access Read/Write Scenarios

If you want to allow access by multiple clients (users) to VTE-protected SNFS files under the same GuardPoint, just read-only access is supported. StorNext file locking is not implemented in VTE, so there is currently no way to prevent concurrent conflicting writes to the same file. As a result, Thales eSecurity 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	Windows
Read/write access from a single LAN client to a GuardPoint	Supported	Supported
Read/write access from two or more LAN clients to the same GuardPoint	Not supported	Not supported
Read-only access from one, two, or more LAN clients to the same GuardPoint	Supported	Supported

Setting up VTE and Quantum StorNext Integration

For the most part, VTE 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 VTE work with SNFS. Later sections provide more information about the steps that are new or differ significantly from a typical VTE setup. Some steps are already documented in this guide or in the DSM Administration Guide.

Integration Task Overview

The table below provides an overview of the steps involved in setting up SNFS and VTE to work together. As noted in the table, some of these tasks are described in the Vormetric Data Security Manager (DSM) Administrators Guide. 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 Vormetric and you don't have access to the DSM.

Task	Key configuration notes	For more information
Install and configure a Quantum StorNext MDC server for use with VTE	Disable the StorNext read-ahead cache. Only certain StorNext policies, features, and mount types are supported. See "Supported StorNext Server and Client Configurations" on page 169.	In this chapter, see "Installing and Configuring a Quantum StorNext MDC Server for Use with VTE" on page 172.
Install and configure Quantum StorNext clients for use with VTE	On Linux, ensure that SNFS starts before `secfs`. See "Ensuring that the StorNext SNFS File System Starts Before secfs (Linux)" on page 173. Only certain operating systems are supported. See "Supported StorNext Server and Client Configurations" on page 169.	In this chapter, see "Installing and configuring Quantum StorNext DLC Clients for Use with VTE" on page 172.

Task	Key configuration notes	For more information
Create a domain for one or more SNFS hosts, or add them to an existing domain	No difference from standard VTE agent configuration.	See “Domain Management” in the DSM Administration Guide.
Add the host to the DSM	No difference from standard VTE agent configuration.	See “Configuring Hosts and Host Groups” in the DSM Administration Guide.
Install and register the VTE Agent on the host system	No difference in installation.	In this chapter, see “Installing the VTE Agent on Each StorNext LAN client” on page 173. See the following chapters in this guide: Chapter 2, “Installing VTE for Windows.” Chapter 3, “Installing VTE for Linux.”
Create encryption keys (optional)	AES-CBC-CS1 keys are not supported on Windows. See the note in “Supported GuardPoint and Key Settings for SNFS File Systems” on page 170.	See “Managing Keys” in the DSM Administration Guide. For creating AES-CBC-CS1 keys on Linux, see “Enhanced Encryption Mode” on page 151 in this guide.
Configure host groups containing one or more StorNext LAN clients (optional)	No difference from standard VTE agent configuration.	See “Configuring Hosts and Host Groups” in the DSM Administration Guide.
Configure policies (including user, process, and resource sets) to control access or enable encryption	No difference from standard VTE agent configuration.	See “Configuring Policies” in the DSM Administration Guide.
Configure one or more GuardPoints	Some GuardPoint settings are not supported. See “Supported GuardPoint and Key Settings for SNFS File Systems” on page 170.	See “Managing GuardPoints” in the DSM Administration Guide.

Installing and Configuring a Quantum StorNext MDC Server for Use with VTE

Install and configure a Quantum StorNext metadata controller (MDC) server using the Quantum StorNext documentation as a guide. The VTE integration works with both Linux and Windows StorNext MDCs. Ensure that you configure the StorNext server to work with the settings supported by VTE as listed in “Supported StorNext Server and Client Configurations” on page 169. 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 VTE

Install and configure Quantum StorNext DLC clients using the Quantum StorNext documentation as a guide. The VTE integration works with both Linux and Windows StorNext DLCs.

Ensure that you configure DLC clients to work with the settings supported by VTE as listed in “Supported StorNext Server and Client Configurations” on page 169. 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 171.

Ensuring that the StorNext SNFS File System Starts Before secfs (Linux)

For VTE to function properly for Linux SNFS clients, the SNFS service must start before the VTE `secfs` service. Add an entry for the SNFS file system to `/etc/fstab` on each Linux client that has a VTE 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 documentation for more details.

Choosing a Mounting Method (Windows)

There are two methods for mounting a StorNext file system on Windows in the StorNext Client Configuration application:

- Map to Drive Letter
- Map to Directory

Both methods are supported in VTE. If you mount the StorNext file system using the Map to Directory method, you must create the directory on the Windows computer before assigning that directory in the Client Configuration application. For example, the default Map to Directory folder is `C:\Mount\snfs1`. If you use that default, you must create `C:\Mount\snfs1` before mounting the StorNext file system in the Client Configuration application.

If you change mounting methods (drive letter to directory or vice versa), you may need to close and reopen Windows Explorer or reboot the computer for the change to take effect.

Installing the VTE Agent on Each StorNext LAN client

Install a VTE agent on each computer that is set up as a StorNext LAN client and for which you want to set DSM policies. For supported operating systems, refer to the table in “Supported StorNext Server and Client Configurations” on page 169.

Use any installation method supported for your operating system. See one of the following chapters for installation instructions specific to your operating system.

- See “Installing VTE for Windows” on page 23.
- See “Installing VTE for Linux” on page 41.

Stop secfs Before Upgrading StorNext LAN Clients (Linux)

Before you upgrade StorNext Linux LAN clients, disable or stop the VTE `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 17: Using VTE 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 VTE to work together.

This chapter contains the following sections:

Supported McAfee Versions and Linux Operating Systems

In general, Thales has verified that VTE is compatible with McAfee version 10.6.5 and later on Red Hat Enterprise Linux (RHEL) 7 and RHEL 8. See the most recent *Vormetric Transparent Encryption Agent Compatibility Matrix* for details about the versions of McAfee Endpoint Security for Linux Threat Prevention that have been verified to work with VTE. You can find that document on the Thales eSecurity Support website (login account required).

Ensuring Correct McAfee Service Startup Order

VTE services and McAfee services must be started and stopped in the correct order to prevent problems with your data that is guarded by VTE. 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 VTE.
- Before performing a manual upgrade of VTE.
- As needed for maintenance or troubleshooting.

Ensuring Correct McAfee Service Startup Order in systemd

Configuring the proper startup and shutdown order of VTE 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 VTE.

The following McAfee services must be configured to start after VTE services:

- `isecespd.service`
- `isectpd.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. See "Location of Application Unit Configuration Files" on page 318 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 319 for information about how to add services to the `secfs-fs-barrier.service` file.

Ensuring Correct McAfee Service Startup Order Manually

Perform the following commands in this order if you need to stop McAfee and VTE services manually:

1. Stop McAfee services:

```
# systemctl stop isecespd.service isectpd.service
```

2. Stop VTE:

Linux distributions that support systemd	`/etc/vormetric/secfs stop`
Linux distributions that do not support systemd	`service secfs stop`

Perform the following commands in this order if you need to start McAfee and VTE services manually:

1. Start VTE:

Linux distributions that support systemd	`/etc/vormetric/secfs start`
Linux distributions that do not support systemd	`service secfs start`

2. Start McAfee services:

```
# systemctl start isecespd.service isectpd.service
```

Updating McAfee

It is not necessary to shut down VTE 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 VTE as described in “Supported McAfee Versions and Linux Operating Systems” on page 175.

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 18: Using VTE 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 VTE on Red Hat Enterprise Linux (RHEL) 7 and RHEL 8.

This chapter contains the following sections:

Supported Deep Security Versions and Linux Operating Systems

VTE and Deep Security can be used with RHEL 7 and RHEL 8. See the most recent "Vormetric Transparent Encryption Agent Compatibility Matrix" for the current versions of Trend Micro Deep Security that have been verified to work with VTE. You can find that document on the Thales Support website (login account required).

Ensuring Correct Deep Security Service Startup Order

VTE services and Deep Security services must be started and stopped in the correct order to prevent problems with your data that is guarded by VTE. 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 VTE.
- Before performing a manual upgrade of VTE.
- As needed for maintenance or troubleshooting.

Ensuring Correct Deep Security Service Startup Order in systemd

Configuring the proper startup and shutdown order of VTE 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 VTE.

The following Deep Security service must be configured to start after VTE 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 318 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 319 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 VTE services manually:

1. Stop Deep Security services:

```
# systemctl stop ds_agent.service
```

2. Stop VTE:

Linux distributions that support systemd	`/etc/vormetric/secfs stop`
Linux distributions that do not support systemd	`service secfs stop`

Perform the following commands in this order if you need to start Deep Security and VTE services manually:

1. Start VTE:

Linux distributions that support systemd	`/etc/vormetric/secfs start`
Linux distributions that do not support systemd	`service secfs start`

2. Start Deep Security services:

```
# systemctl start ds_agent.service
```

Updating Deep Security

It is not necessary to shut down VTE 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 VTE as described in “Supported Deep Security Versions and Linux Operating Systems” on page 177.

Chapter 19: VTE for Amazon S3

This chapter discusses how to configure VTE for Amazon S3 buckets. It contains the following sections:

Overview

VTE for Cloud Object Storage (VTE-COS) is an object storage service that customers can use it 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. VTE-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.

VTE-COS for Amazon Simple Storage Service (VTE-COS S3) is an extension to VTE for inline encryption and local host access controls involving applications performing REST-API-based operations to S3 object stores.

VTE-COS S3 is not a replacement for AWS IAM access controls which are enforced independently at the AWS server end.

Supported operations

- VTE-COS S3 will process only AWS S3 REST https calls issued by applications. The interception is done by a bundled Squid TLS proxy with additional VTE services.
- VTE'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

- VTE-COS S3 is supported only on RHEL 7.
- Only S3 protocol-aware applications are supported.
- Only locally generated self-signed TLS Proxy CA Certificate are supported.
- AWS S3 URL path validations are not currently implemented. VTE-COS S3 requires that the user must specify the correct URLs for bucket paths.
- VTE-COS S3 permits only AES CBC-CS1 encryption. Encrypted files in protected buckets therefore will prepend the 4K embedded header used in VTE's AES CBC-CS1 encryption. The DSM will enforce usage of AES CBC-CS1 keys within S3 bucket policies.

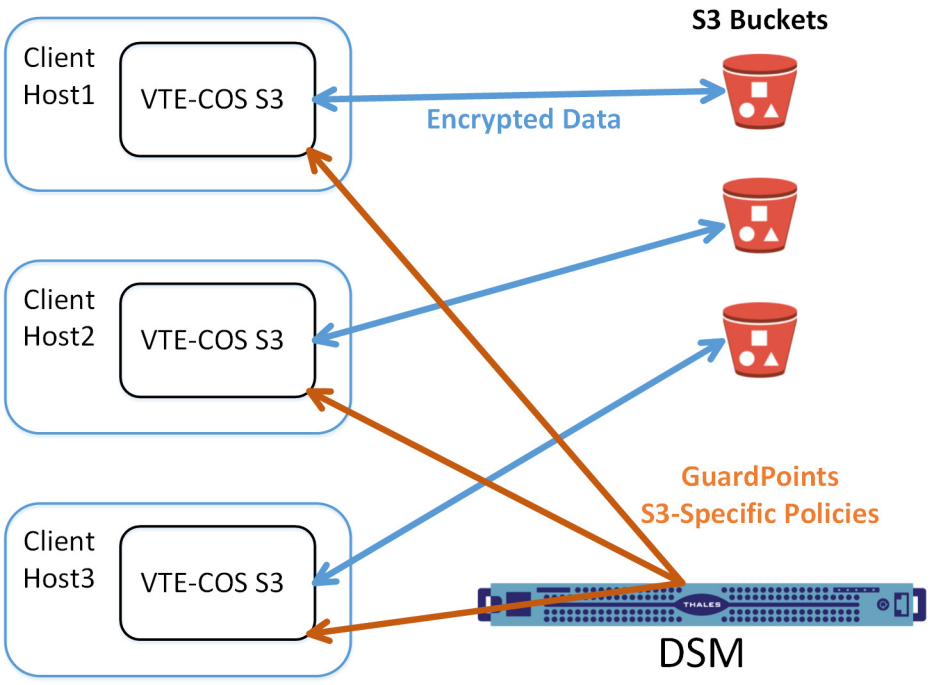
Multi-part Upload Restrictions

- All upload operations 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 VTE S3's AWS credential file must match the part size specified within the application.
- Content-MD5 header should be present in the request message.

System Components

The VTE-COS S3 system includes the following components:

- COS TLS proxy
- VTE-COS S3 adapter
- Custom Hardened Open SSL bundle



System and Software Requirements

- The RHEL 7 Linux host must meet the standard VTE requirements.
- You must have an AWS account
- We recommend that you guard the bucket by restricting it with a VTE-COS S3 Role to prevent accidental data corruption from connections outside the control of VTE-COS S3. For details, see “Optionally Configure a VTE-COS S3 Role for Guarded Buckets” on page 184.
- You must download and install the pre-requisite rpm packages before you install VTE-COS S3. For details, see “Install Required Linux Packages” on page 181.
- The VTE Agent must be installed in the default directory on the host. You cannot change the default path if you also enable VTE-COS.

Client Software Requirements

- Clients must exist on the same host as the VTE-COS Service. External Client connections will be rejected by the VTE-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 network Proxy Gateway (default is localhost:3128 or 127.0.0.1:3128).
- Clients must use the TLS 1.2 / TLS 1.3 network encryption protocol over TCP/IP to establish connections to the Proxy Gateway.
- Clients must be configured to use the COS CA Root Certificate to verify and authenticate TLS connections.

VTE-COS S3 Installation Overview

In order to configure VTE-COS S3, you must complete the following tasks:

1. Install the required libraries for VTE-COS S3. For details, see “Install Required Linux Packages” on page 181.
2. Install VTE and generate the local COS Proxy CA Certificate. For details, see “Install VTE” on page 182.
3. Optionally configure the client to use the COS Proxy CA Certificate. For details, see “Optionally Configure the AWS CLI to use the COS Root CA Certificate” on page 183.
4. Optionally configure the client to use the COS Network Proxy port. For details, see “Optionally Configure the AWS CLI Network Proxy” on page 183.
5. Configure COS Service with AWS Credentials. For details, see “Configure the VTE-COS S3 Service” on page 183.
6. Configure Role & IAM policies for COS Service for guarded buckets. For details, see “Optionally Configure a VTE-COS S3 Role for Guarded Buckets” on page 184.
7. Configure the guarded buckets. For details, see “Guard an AWS Bucket” on page 187.

Install Required Linux Packages

VTE-COS S3 requires the following pre-requisite packages:

- `boost-regex`.
- `lib-curl`.
- `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 libcurl epel-release
$ sudo yum install cryptopp log4cpp
```

VTE-COS S3 supports both Python2 and Python3. If both versions of Python are available, VTE-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 VTE

When you install VTE-COS S3 COS service for use with S3, use the instructions in or “Installing VTE for Linux” on page 41. In addition to those instructions, answer the following prompts as described in this section.

1. If you are using Fingerprint registration, make sure you know the fingerprint for this agent on the host. This Fingerprint can be seen in the Host Information shown in the DSM Management Console.
2. When the installer asks about Cloud Object Storage, type **`Y`** and follow the prompts as shown.

   Do *not* change the default VTE installation directory. VTE 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
VTE 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 Dashboardpage 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 Vormetric Encryption Expert File System Agent with the
primaryVormetric Data Security Server on dsm64-64-97.qa.com.
```

```
Starting VTE Cloud Service

Installation success.
```

Optionally Configure the AWS CLI to use the COS Root CA Certificate

If you want to configure the AWS CLI to use the COS root CA certificate, 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
```

Optionally Configure the AWS CLI Network Proxy

If you want to configure the AWS CLI to use a network proxy, set the environment variable `HTTPS_PROXY` or `https_proxy`. If both variables are defined, then the AWS CLI will use `https_proxy`.

```
Export HTTPS_PROXY=localhost:3128
```

Configure the VTE-COS S3 Service

Use the following command to add your AWS credentials to the VTE-COS S3 service:

`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.

If you do not specify the secret key and the key ID, the `voradmin` command prompts for this information.

For example:

```
voradmin cos s3 cred add AKIA****P KQSm****D
```

Generating a New Secret Key and Secret Key ID

If you need to generate a new secret key and secret key ID, do the following:

1. Log into the AWS Management Console.
2. In the top right hand corner, under the login id, click **My Security Credentials**.
3. Under the AWS IAM Credentials tab, click **Create Access Key** to generate an Access Key ID and Secret Access Key.

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 VTE-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 VTE-COS S3 Role for Guarded Buckets

In order to prevent access to a bucket outside the VTE-COS S3 protection, you can create a special IAM user for VTE-COS S3 that is assigned a custom IAM role and policy. The policy restricts access to the bucket, thereby ensuring that all access to the bucket must go through VTE. This is not required but it is highly recommended.

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 VTE protection. Make sure you name the policy something that you will remember.

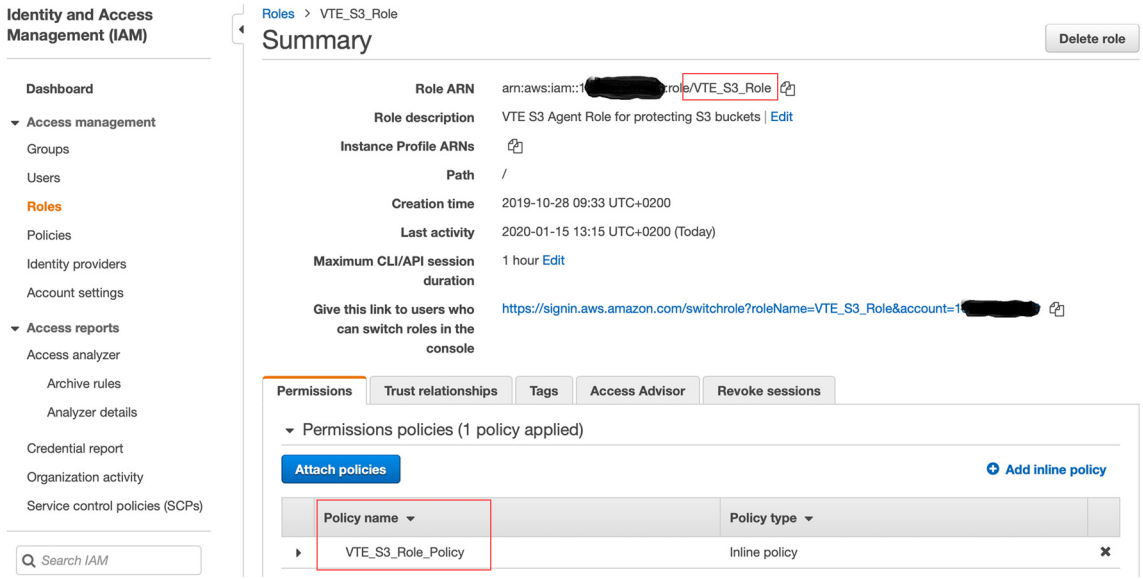
   For example, you can create a policy called VTE_S3_Role_Policy that allows access to the single S3 bucket vte-cos-s3-rtb. To verify that the policy restricts access to that bucket, you can look at the Resource allocation in the Policy summary.

   The full JSON for the the VTE_S3_Role_Policy is:

```
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "VisualEditor0",
"Effect": "Allow",
"Action": [
"s3:PutAccountPublicAccessBlock",
"s3:GetAccountPublicAccessBlock",
"s3:ListAllMyBuckets",
"s3:ListJobs",
"s3:CreateJob",
"s3:HeadBucket"
],
"Resource": "*"
},
{
"Sid": "VisualEditor1",
"Effect": "Allow",
"Action": "s3:*",
```

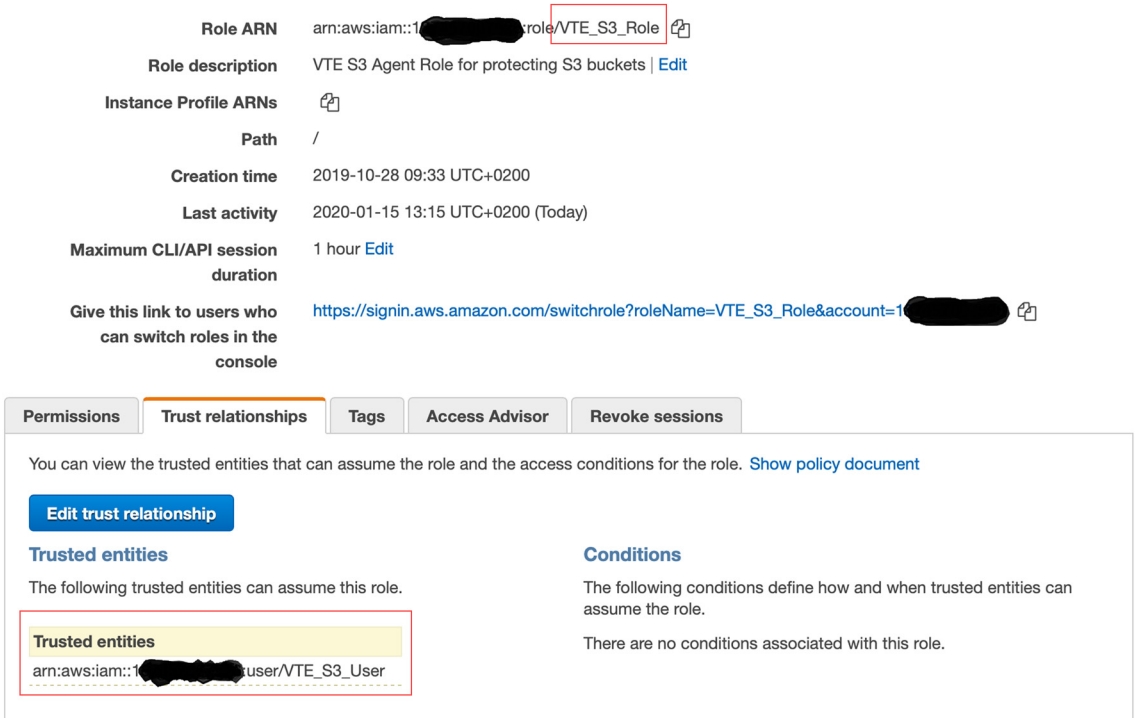
```
        "Resource": [
        "arn:aws:s3:::vte-cos-s3-rtb",
        "arn:aws:s3:::vte-cos-s3-rtb/*"
        ]
        }
        ]
        }
```

2. Create a new role that you can use for the VTE-COS S3 Role. For example, you could call the role VTE_S3_Role.
3. Assign the VTE-COS S3 policy you created to the role. For example:



4. Create an IAM user for the VTE-COS S3 Role. The role does not require any privileges because its only job is to assume the VTE-COS S3 Role. The 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 and no credentials.

5. Create a trust relationship between the VTE-COS S3 Role and the VTE-COS S3 user on the Trust relationships tab. For example:



6. Configure the VTE-COS S3 Role with the credentials of the VTE-COS S3 IAM user you created earlier by entering the following command:

   `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 VTE-COS S3 user that you created.
   - `<aws_secret_key>` is the AWS secret key or the VTE-COS S3 user that you created.
   - `<user_arn>` is the Amazon Resource Name for the VTE-COS S3 user that you created.
   - `<role_arn>` is the Amazon Resource Name for the VTE-COS S3 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 VTE-COS S3 user is 1XXXXXXXXXXX, the user name is VTE_S3_User, and the VTE-COS S3 Role is VTE_S3_Role, you would enter:

```
voradmin cos s3 role config AKIA****P KQSm****D arn:aws:iam::1XXXXXXXXXXX:user/
VTE_S3_User arn:aws:iam::1XXXXXXXXXXX:role/VTE_S3_Role
```

   After you configure the user and assign the VTE-COS S3 Role, VTE will access the S3 bucket through the VTE-COS S3 user account using temporary credentials that VTE regenerates periodically. These credentials are maintained entirely by VTE and are never exposed to end users.

Secure an S3 Bucket with the VTE-COS S3 Role

When you enable the VTE-COS S3 Role for a bucket, the associated policy prevents unauthorized users from accessing the contents of the bucket. To enable the VTE-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 VTE-COS S3 user that you created.
- `<aws_secret_key>` is the AWS secret key or the VTE-COS S3 user that you created.

- `<bucket_name>` is the name of the S3 bucket on which you want to enable the VTE-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 VTE-COS S3 Role for an S3 Bucket

To remove the VTE-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 VTE-COS S3 user that you created.
- `<aws_secret_key>` is the AWS secret key or the VTE-COS S3 user that you created.
- `<bucket_name>` is the name of the S3 bucket on which you want to disable the VTE-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 details, see “Create a CBC_CS1 Key” on page 187.
2. Create a Cloud Object Storage (COS) policy. For details, see “Create the Cloud Object Storage (COS) policy” on page 188.
3. Apply the policy to the host. For details, see “Creating GuardPoints in a Host” on page 189.

Create a CBC_CS1 Key

1. Log on to the Management Console as an administrator of type: All or Security Administrator with Key role permissions.
2. In the menu bar select **Keys > Agent Keys > Keys**.

   The Agent Keys page displays.
3. Click **Add**. The Add Agent Key window opens.
4. Select the Symmetric tab.
5. Complete the fields in this window by using the following information.

   Name: Provide a name for the key.

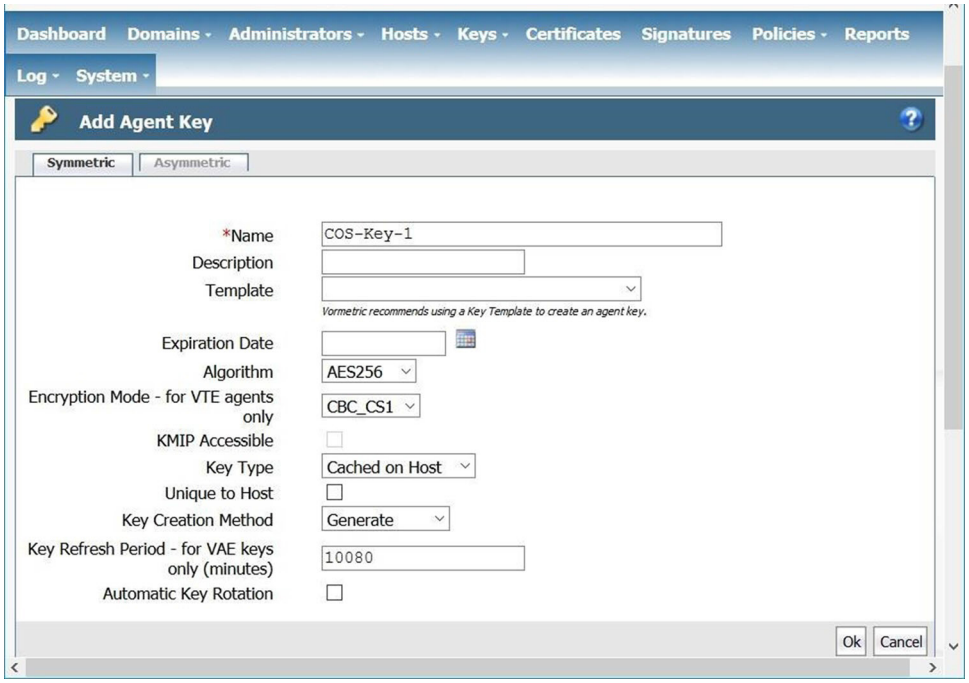
   Algorithm: Select AES256.

   Encryption Mode for VTE agents only: DSM allows only CBC-CS1 encryption mode for COS policy type.

   Key Type: Cached on Host

   Key Creation Method: Generate

For example:



6. Click OK.

Create the Cloud Object Storage (COS) policy

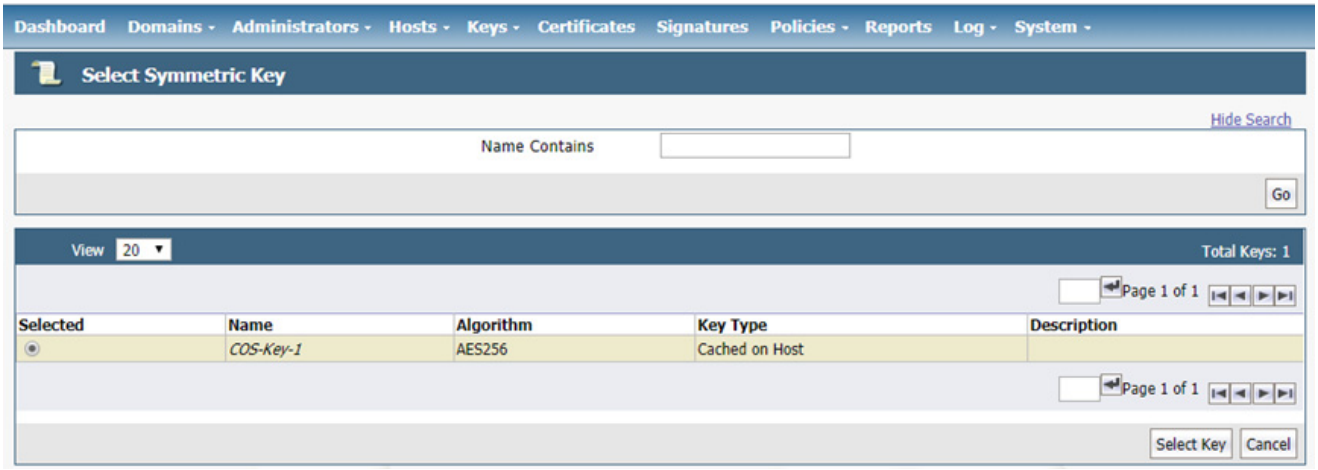
Creating a COS policy differs from creating standard policies. Some notes about COS policies:

- Allow Browsing is not supported for Cloud Object Storage polices.
- Policies for Cloud Object Storage do not contain a resource set. The resource set is automatically the cloud object storage.
- Policies for Cloud Object Storage do not contain a When/Time set.
- Key Rules for Cloud Object Storage policies only contain one Key.
- Policies for Cloud Object Storage do not support Exclusion rules.

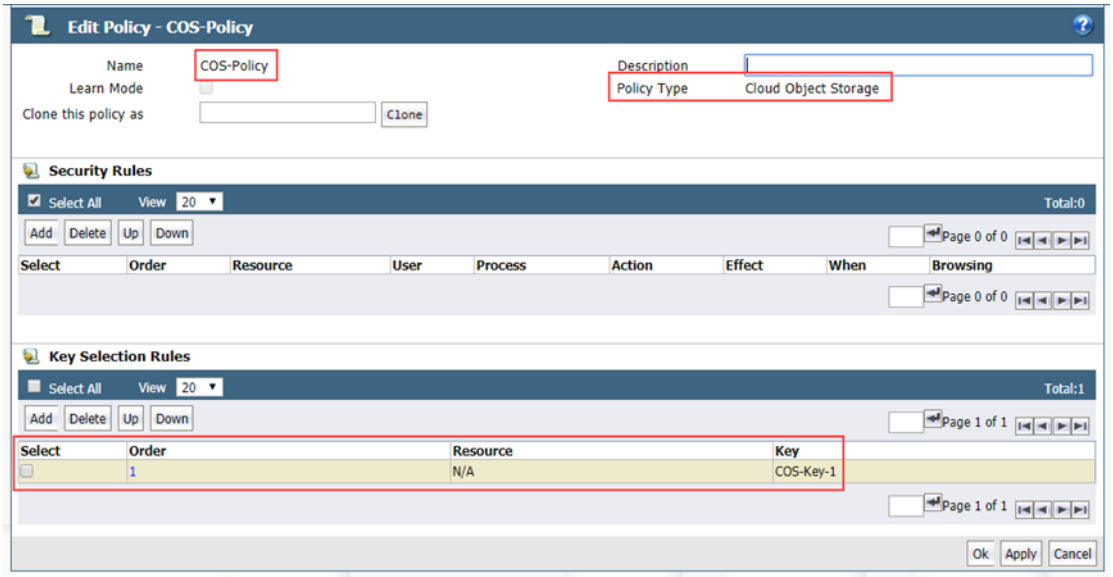
To create the policy:

1. Click **Policies > Manage Policies** to list the policies available to this domain.
2. Click **Add**. The **Add Policy** page appears.
3. For **Policy Type**, select **Cloud Object Storage**.
4. In the **Name** field, enter a name for the policy. For example
5. In the **Key Selection Rules** section, click **Add**. The Add Key Rule page appears.

6. Click **Select** and choose the CBC_CS1 key you created in "Create a CBC_CS1 Key" on page 187.



7. Click **Select Key**, then click **OK** to confirm that you want to use the CBC_CS1 key.
8. Click **OK** to save the Policy with the CBC_CS1 key rule.



Creating GuardPoints in a Host

Security/All administrators can create new GuardPoints or edit existing host GuardPoints.

Note: DSM allows **exactly one** COS GuardPoint per S3 bucket.

The following features are NOT supported in COS GuardPoints:

- Existing data transformation, either offline or with Live Data Transformation (LDT). If you want to encrypt existing data, you must move it out of the S3 bucket, guard the bucket, then move the existing data back into the guarded bucket. When the data is moved back into the S3 bucket, it will be encrypted by VTE before it becomes accessible to users.
- Secure Start.

- Browsing to a bucket to be guarded.
- Host to Browse.
- Auto Mount.
- Efficient Storage.
- Buckets inside a container or Docker.

Note: In the Edit Host window, you will see that the **Guard FS** tab has been renamed to **GuardPoints**.

To create a COS GuardPoint:

1. In the Hosts window, click on the host for which you want to set GuardPoints.
2. In the Edit Host window, click **GuardPoints**.
3. In the GuardPoints tab, click **Guard**.
4. In the Policy dropdown menu, select an appropriate policy.
5. For type, click Cloud Object Storage (Auto Guard or Manual Guard).
6. In the Path field, enter the path for the GuardPoint.
7. Click **OK**. COS GuardPoints display on the GuardPoints tab of the Host Detail page.

Additional COS Proxy Root CA Certificate Information

The VTE COS CA Certificate, not to be confused with the Kernel and VMD Kernel Certificates, is used with the COS Service 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.

Chapter 20: Efficient Storage

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. Vormetric Transparent Encryption (VTE) offers a much higher level of security. However, in deployments where storage arrays receive encrypted data streams from hosts protected by a VTE instance (protected hosts), the encrypted data streams make the efficiency of storage array systems ineffective. This is because every block written by VTE is different resulting in zero deduplication and compression on these blocks.

VTE offers Efficient Storage as the solution to the storage efficiency challenge in storage arrays on Linux and Windows.

This chapter contains the following sections:

Introduction to Efficient Storage

Efficient Storage for Linux is a licensed feature available with VTE 6.2 and DSM 6.3 or later and Efficient Storage for Windows is a licensed feature available with VTE 6.3 and DSM 6.3 or later.

Note: All references to 'storage array' in this document assume storage array systems capable of supporting Efficient Storage functionality.

With Efficient Storage, the VTE 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 the VDS Platform is essential for achieving storage efficiency with encrypted data.

Note: Efficient Storage requires XTS-AES mode of the AES algorithm for encryption.
VTE only supports Efficient Storage on servers with microprocessors integrated with Advanced Encryption Standard instruction set (AES-NI).

In the context of this solution, a LUN exported from a storage array system to a VTE-managed host, is referred to as an *Efficient Storage GuardPoint (ES GuardPoint)*. An ES GuardPoint is a guarded device, configured with the Efficient Storage capability.

Efficient Storage Enhanced Storage Arrays

The VDS Platform 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 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 VTE Efficient Storage

You can use VTE Efficient Storage with:

- FlashArray from Pure Storage
- PowerMaxOS 5978 Q3 2020 SR from EMC

FlashArray

FlashArray from Pure Storage is enhanced with Efficient Storage capabilities and inter-operates with VDS Platform to provide VTE Efficient Storage on Linux and Windows. The Pure Storage system is a client of the Vormetric Data Security Manager (DSM) and shares the encryption keys protecting the LUNs exported from the storage system to the protected hosts registered with the same DSM.

Note: See the *EncryptReduce Installation Guide* from Pure Storage for information on setting up interoperability with the VDS Platform.

PowerMax

PowerMaxOS 5978 Q3 2020 SR introduces the availability of end-to-end efficient encryption which increases security by encrypting data at the host level while also looking for maximum data reduction on the PowerMax array. The encryption functionality is provided by the integration of PowerMax with VTE and the DSM.

End-to-end efficient encryption can be added to pre-existing PowerMax arrays that are D@RE enabled and have a free front-end I/O slot per director to accommodate the addition of the dedicated I/O module.

Sharing Encryption Keys

The VDS Platform shares the encryption key for a LUN with the storage system using the KMIP protocol. In this solution, the DSM is the KMIP server, and the storage system is a KMIP client registered with the DSM. Any host accessing and protecting the LUNs from the storage system is a VTE managed host registered with the same DSM. The protected hosts register with the DSM using the `register_host` script executed on the protected hosts.

With the host and the storage system registered with the same DSM, 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 DSM (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 DSM 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 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 DSM administrator. The DSM administrator produces a corresponding certificate specific to the array which is given to the storage array administrator. The DSM 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 DSM's KMIP server.

Efficient Storage Header and Private Region

The key sharing aspect of an ES GuardPoint requires a small amount of disk space in the storage device reserved for VTE private use. The reserved space is where VTE 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.

VTE allocates a small amount of storage space on each device configured as Efficient Storage. This region is reserved for exclusive use by VTE and is referred to as the VTE private region. On Linux, the VTE private region is 63 megabytes. On Windows, the VTE private region is 64 megabytes.

VTE stores the ES header and other metadata information to allow VTE and the storage array to exchange information. The ES header occupies the first sector on the device. The method that VTE uses to claim the private region on a device depends on whether the device is new (holds no data) or has existing data that you want to preserve. VTE 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 VTE 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 VTE 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 VTE private region. To view the ES Header on Linux, 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 VTE. 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 AES256/XTS key applied to the device. The reported UUID matches the UUID of the key as shown on the DSM key property and KMIP key object.

Note: You cannot view the ES Header on Windows in the current VTE release.

DSM Domain Enabled for KMIP

The KMIP protocol is the interface between the DSM and the storage arrays registered with the DSM. The Domain in the DSM for serving security requirements of storage arrays and protected hosts must be KMIP enabled. For deployment of ESG, you must create a new domain with the Enable KMIP option checked or modify an existing domain so that the Enable KMIP box is checked. This allows the DSM to make the keys for guarding Efficient Storage devices within the domain available to KMIP clients (storage arrays). Existing keys in a domain before checking KMIP Enable checkbox will not be accessible through KMIP.

A DSM domain can serve any number of FlashArrays as a KMIP client, but a FlashArray registered with a DSM domain *cannot* register with other domains in the DSM or with another DSM. This restriction forces the FlashArrays registered with a DSM domain to serve LUNs *only* to the VTE hosts registered with that specific DSM domain in that specific DSM.

Note: See the *EncryptReduce Installation Guide* from Pure Storage for detailed instructions on registering the FlashArray as KMIP client with the with the DSM's KMIP server.

ES GuardPoint Encryption Keys

ES GuardPoints must be encrypted using XTS-AES256 keys. An XTS-AES256 type key is a 512-bit key composed of two components:

- The first 256 bits of the key is the AES256 encryption key component
- The second 256 bits is the tweak component

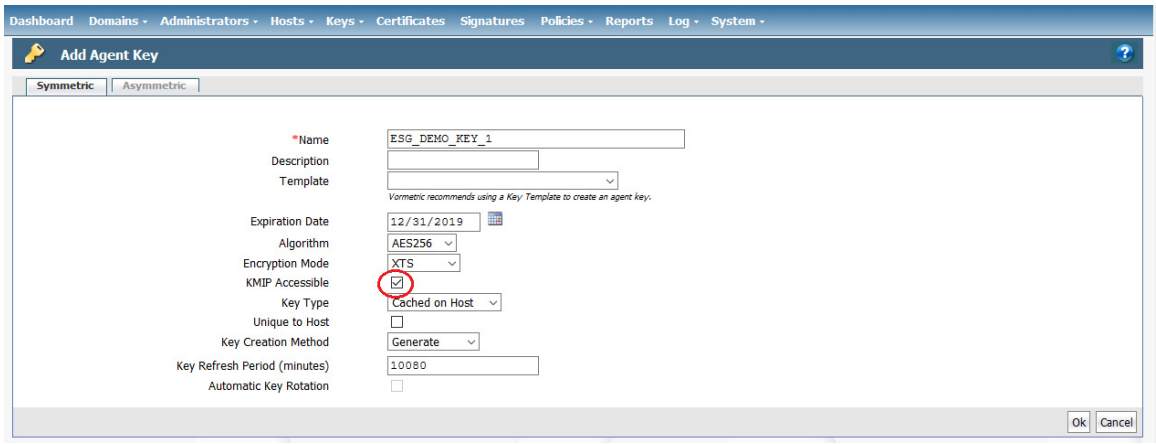
You create XTS keys on the DSM using the “Add Key” function, similar to non-XTS keys. You must check the “KMIP Accessible” box on the “Add Agent Key” page to make the key available to KMIP clients through KMIP. If you do not check KMIP Accessible, storage arrays cannot get keys for Efficient Storage devices from the DSM (KMIP server) for sharing those keys with protected hosts.

Note: Be sure to check the “KMIP Accessible” checkbox when creating keys for ES GuardPoints.

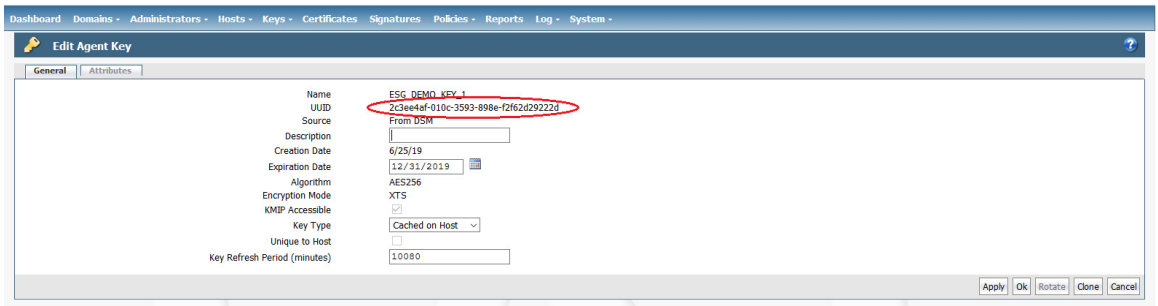
The DSM also generates a UUID for a newly added key. The DSM provides the key and its attributes to the protected hosts for guarding Efficient Storage device. The protected host protecting the device writes the ES header to the device, including the UUID of the key, before the initial attempt to enable 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 AES256 component of the XTS key from the DSM using KMIP.

Key Attributes - Example

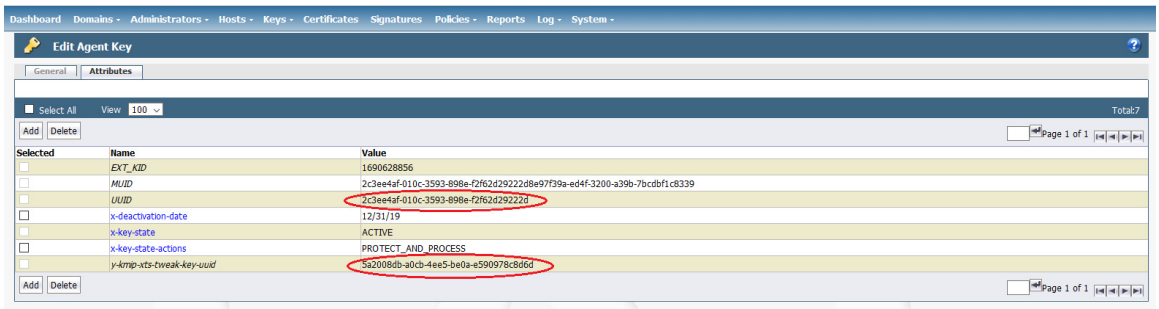
The following figure illustrates the parameters you should specify to add a new XTS-AES256 key named ESG_DEMO_KEY_1. Note the key algorithm and encryption mode specified for the key and the checked “KMIP Accessible” option.



The following figure illustrates the general properties of ESG_DEMO_KEY_1 after adding the key. Note the UUID of the key.



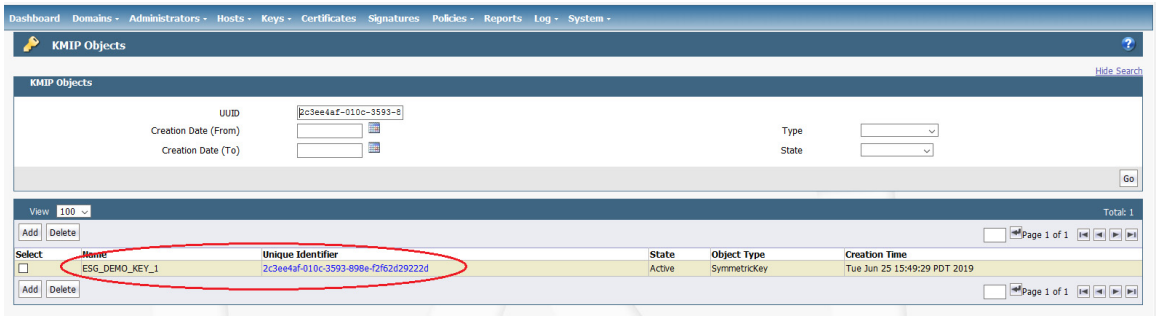
The following figure illustrates the attributes of ESG_DEMO_KEY_1 after adding the key. Note that this tab displays both the UUID of the key and the UUID of the tweak component.



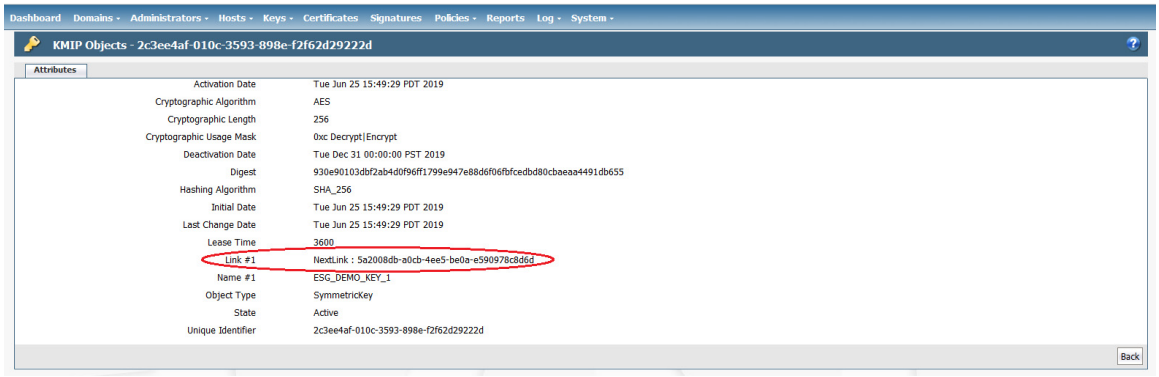
The attribute “x-deactivation-date” represents the date on which ESG_DEMO_KEY_1 expires. When the expiration date is reached, the key is no longer valid. It is important to choose an expiration date that exceeds any data retention period imposed on the data stored in the protected LUNs where the key is used. Note that any snapshot of a protected LUN inherits the same key protecting the primary LUN, therefore, if the key expires before reaching the end of retention period, the LUN and its snapshots will not be accessible. More importantly, do not delete the key before the data retention period expires on the LUN. The data on the LUN and its snapshots cannot be decrypted if the key is deleted prior to the data retention period.

KMIP Key Object

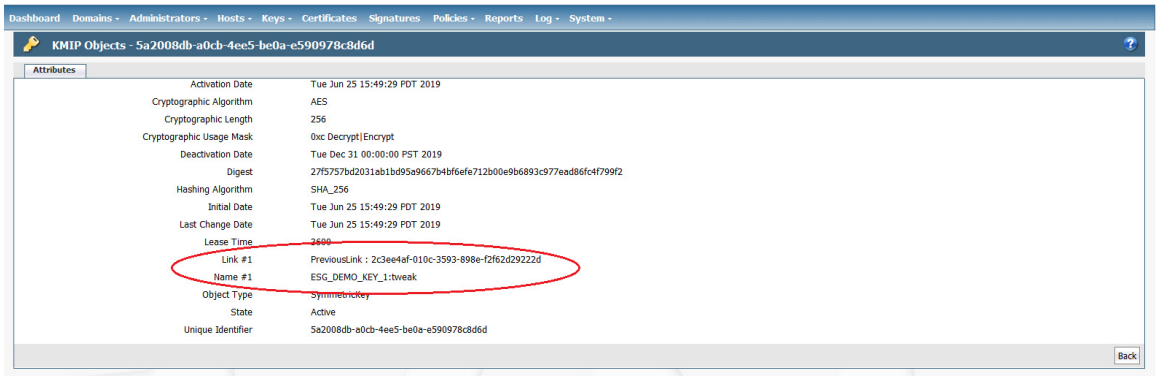
The following figure shows the attribute of the KMIP object for the specified UUID. The UUID specified in the example is the UUID of ESG_DEMO_KEY_1.



The following figure shows the KMIP attributes of the KMIP objects associated with ESG_DEMO_KEY_1.



The following figure shows the KMIP attributes of the KMIP object associated with the tweak component of ESG_DEMO_KEY_1.

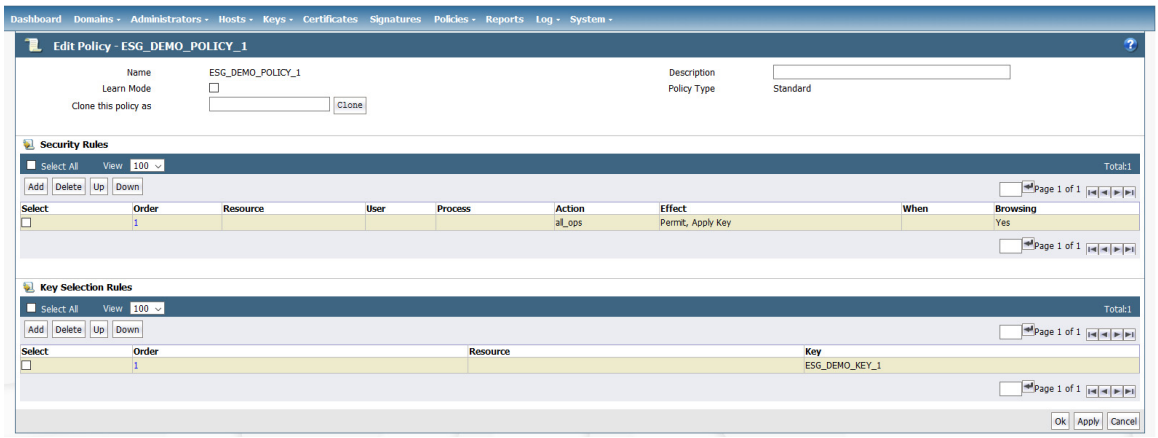


Policy Requirements for ES GuardPoints

Efficient Storage requires a standard policy with an XTS-AES256 key rule as described in “ES GuardPoint Encryption Keys” on page 196.

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 225.

The following figure shows an example of a simple policy that uses ESG_DEMO_KEY_1 key to guard devices.



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 “Guarding an Efficient Storage Device on Windows” on page 213.

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 Efficient Storage GuardPoints. For details, see “Requirements for Efficient Storage GuardPoints on Linux” on page 200.
2. Register the protected host with the DSM with Efficient Storage enabled. For details, see “Register the Host with DSM” on page 200.
3. Initialize the storage device to create a Private Region for the Efficient Storage Header. For details, see “Initialize a New Linux Device” on page 202.

4. Log on to the DSM to apply the ES GuardPoint to the storage device. For details, see "Guard the Linux Device with an Efficient Storage GuardPoint" on page 204.

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 Efficient Storage capability.
- The storage array exporting the LUN to the protected host must be a KMIP client registered with the same DSM 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 Efficient Storage GuardPoint cannot currently be initialized/added as physical volumes for use by LVM. When LVM support is added, it will be announced in the VTE Release Notes.
- Existing devices divided into one or more logical partitions cannot be guarded as Efficient Storage 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 Efficient Storage GuardPoint. Using `/dev/secvm/dev/sda1` is invalid as `/dev/secvm/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 Efficient Storage devices, see the available documentation from the Storage Array system vendor.

Register the Host with DSM

If the host is already registered with a DSM, you can initialize the devices you want to protect as Efficient Storage GuardPoints as described in "Initialize Linux Efficient Storage Devices" on page 201 as long as:

- The host is part of a domain that has KMIP enabled.
- A license for the Efficient Storage feature has been uploaded to the DSM.

To verify this, have the DSM Administrator check the host settings in the DSM.

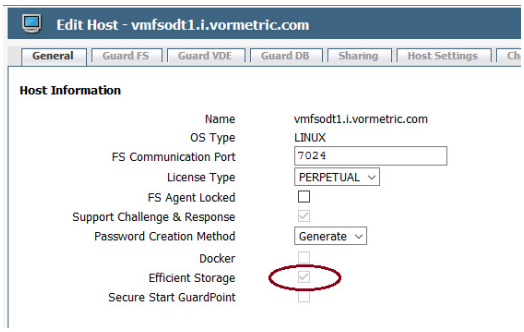
If the host is not already registered with a DSM, you need to register it before you can protect the devices on that host. When you register the host with the DSM, you must have AES-NI available on the host, a current license for registering host, and a current license for Efficient Storage.

1. If you are planning to use the fingerprint registration method, make sure the DSM Security Administrator has added the host to the appropriate KMIP-enabled domain.
2. If necessary, install VTE on the host. Follow the installation prompts as until you get to the registration portion of the installation. If VTE is already installed, you can start the registration process by running `/opt/vormetric/DataSecurityExpert/agent/vmd/bin/register_host`.

3. Follow the prompts to register the host with the DSM. During the registration process, make sure that you enable Efficient Storage support when prompted.

```
Enter a number, or type a different host name or IP address in manually:
What is the name of this machine? [1]:
You selected "vmfsodt1.i.vormetric.com".

Would you like to register to the Security Server using
a registration shared secret (S) or using fingerprints (F)? (S/F) [S]: F

It is possible to associate this installation with the hardware of this
machine.  If selected, the agent will not contact the DSM 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]: n

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

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

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

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.
```

4. Complete the registration process as normal.
5. To verify that the host is properly registered with the Efficient Storage option enabled, have the DSM Administrator verify that Efficient Storage is checked for the host as shown below.



6. After the host is registered with the DSM, initialize each device that you want to guard as described in the next section.

Initialize Linux Efficient Storage Devices

When you initialize a Linux efficient storage device, the process creates a private region on the device for VTE to write the Efficient Storage header along with metadata that identifies the ES device as a guarded device. The VTE 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” on page 202
- “Initialize and Resize an Existing Linux Device” on page 203

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 VTE can reserve the first 63MB of storage on the device for the VTE private region. The remaining storage space is available for new user data. The device size reported to applications is the actual device size minus VTE 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 an Efficient Storage Device with Multiple IO Paths on Linux" on page 208.

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). VTE will create the VTE 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, VTE 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 VTE 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 DSM Administrator can protect the device as an ES GuardPoint through the DSM Management Console. For details, see "Guard the Linux Device with an Efficient Storage GuardPoint" on page 204.

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 DSM 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 VTE. This command examines the current disk size and computes the size required to hold the existing data plus the VTE private region at the beginning of the device. After the VTE 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 VTE. 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. VTE 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. VTE 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 unaccessible 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, VTE 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 VTE 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 DSM Administrator can protect the device as an ES GuardPoint through the DSM Management Console as described in "Guard the Linux Device with an Efficient Storage GuardPoint" on page 204.

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 DSM 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 Efficient Storage GuardPoint

Note: For details about how to create a GuardPoint, see the chapter, “Managing GuardPoints”, in the *DSM Administration Guide*.

After the device has been initialized, you can guard the device as an ES GuardPoint from the DSM Management Console. For existing devices, as soon as the GuardPoint configuration has been pushed to the host and the status changes to guarded, VTE begins transforming the data on the disk using the encryption key associated with the GuardPoint Policy.

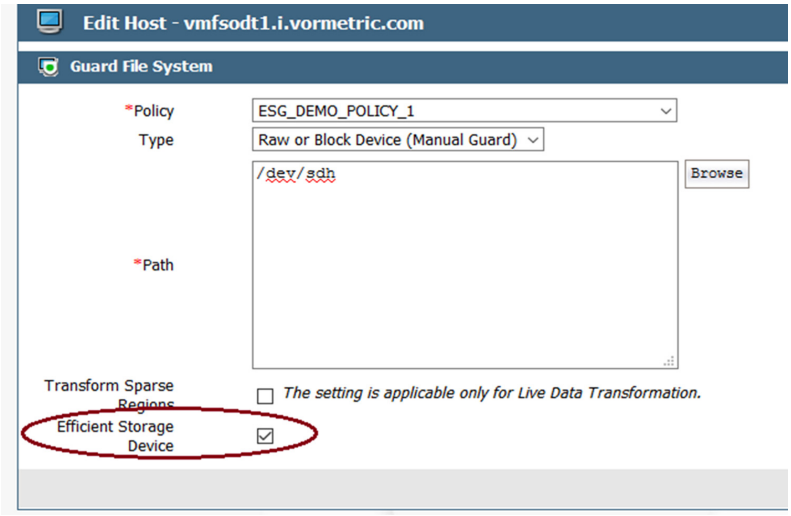
1. Log on to the DSM Management Console as an administrator of type Security with Host role permissions, type Domain and Security, or type All.
2. Make sure that you know what Policy you want to associate with the GuardPoint or create a new standard policy if needed. The policy you use for Efficient Storage must use an XTS-AES256 key as the key rule. For details, see “ES GuardPoint Encryption Keys” on page 196.
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. VTE will use the XTS-AES256 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**, VTE 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 210.

   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 VTE 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 DSM pushes the policy and the GuardPoint configuration to the host and the VTE agent on the host writes the ES header into the VTE 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, VTE 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 page 205 and "Data Transformation on Existing Linux Devices" on page 206.

    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 209.

After the device is initialized and guarded, the protected device must be accessed through the VTE 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.

Note: Be sure to use the secvm device name when using file system management tools such as `mkfs` and `fsck`.

Note: 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 a GuardPoint to a device that has been initialized with the `xform` option, VTE first relocates existing data in the region of the device designated as VTE 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 GuardPoint. VTE 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.

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 (IDT).

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 private region on the device to manage the entire transformation process. 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 private region of the device, and then transforms the data in-place. IDT also maintains the segments undergoing transformation in the 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.

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, 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 IDT over legacy offline data transformation is that IDT does not require a separate policy for data transformation. With the same production policy applied to the device, IDT determines whether the device is in need of data transformation, per specification of `xform` option when device was initialized, and starts the IDT process when transformation is required. During the IDT process, access to the device is blocked until 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

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 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
```

Figure 20-1: Device before resizing on Linux

Now you must resize the device on the storage array to the specified size or larger, and then guard the device on the DSM as ES GuardPoint using Manual Guard. For details about guarding the device on the DSM, see “Guard the Linux Device with an Efficient Storage GuardPoint” on page 204.

After resizing the device and configuring the GuardPoint in the DSM:

```
# 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
```

Figure 20-2: Device after resizing on Linux

At this point VTE 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 xform /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 an 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 ESG, 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: ESG is the only feature of VTE 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 DSM, guard `/dev/mapper/mpathA` as Device GuardPoint using the policy created above. Be sure to check the **Efficient Storage** checkbox.
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 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, VTE shows the status of any data transformation processes happening on the device. If you do not specify this option, VTE 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 4797
4798 4799
   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.

Linux System and ES GuardPoint Administration

voradmin ESG Commands on Linux

The `voradmin` command is a command line utility for management of VTE specific configuration and status reporting. The `voradmin` command also supports configuration management related ES GuardPoints (ESG).

For details about the Linux `voradmin esg` 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 ES GuardPoint. VTE 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” on page 229.

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-db1e-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 a GuardPoint configured for Auto Guard and Manual Guard. For more information about Auto and Manual Guard options, see “Guard the Linux Device with an Efficient Storage GuardPoint” on page 204.

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 `x-systemd.requires=secvm-barrier.service` and `nofail` settings in the `fstab` entries listing ES GuardPoints, as shown in the following table.

Option	Description
`x-systemd.requires=secvm-barrier.service`	Ensure that the ESG GuardPoint is enabled before the device is mounted at startup and disabled after the device is unmounted at shutdown. The `secvm-barrier.service` service is a proxy for all the services that make up VTE.
`nofail`	The system boot will proceed without waiting for the ESG device if it can’t be mounted successfully.

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,nofail 0 0
```

For information about configuring systemd for VTE, see “VTE and systemd” on page 315.

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		

Changing the Encryption Key on Linux ESG Devices

To meet various compliance requirements, you may want to change the key that VTE has used to encrypt ES 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 ES GuardPoint, the device must be taken offline. The data on the device will be inaccessible during the key rotation process.

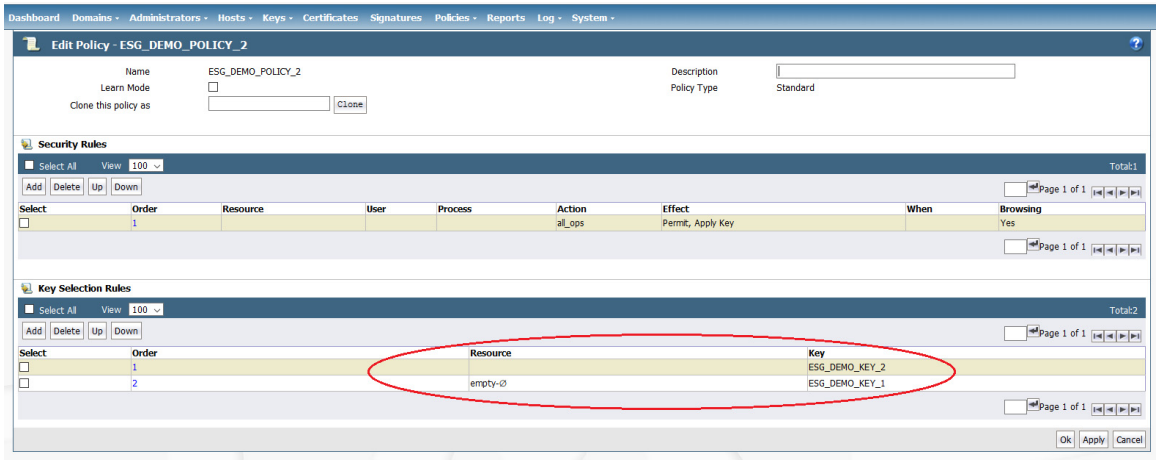
The key rotation process involves the following:

- Creating a new policy for key rotation
- Preparing the ESG device for key rotation
- Applying the new policy to the ESG device on the DSM

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 DSM Policy for Key Rotation

As part of rekeying the data on an ESG 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 VTE 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 DSM not to push the key ESG_DEMO_KEY_1 to the protected host.

Notice the resource set empty-Ø 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 Efficient Storage GuardPoint for Rekey

Before you apply the new policy, you must shutdown any applications accessing the GuardPoint. This also includes unmounting file system if the 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 DSM. Unguarding the device on the DSM deletes the GuardPoint.

The following screenshot enumerates the administrative steps for rekeying ES GuardPoints. The screen shot 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: 3317 ... 20189, 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 DSM and then run the `voradmin` command to prepare the device for rekey. Be sure the device is unguarded on the DSM 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 DSM as an Efficient Storage device using the policy ESG_DEMO_POLICY_2. After the DSM pushes the configuration on the GuardPoint, proceed with guarding the device on the protected host. The guard operation detects the policy/key change on the device and begins the IDT process to transform data from ESG_DEMO_KEY_1 to ESG_DEMO_KEY_2. During the IDT process, user access to the 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 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
```

Guarding an Efficient Storage Device on Windows

The following sections discuss how to guard an efficient storage device on Windows. If you want to guard an efficient storage Linux device, see “Guarding an Efficient Storage Device on Linux” on page 199.

In order to guard an efficient storage device, you need to:

1. Make sure the devices you intend to guard meet the requirements for Efficient Storage GuardPoints. For details, see “Requirements for Efficient Storage GuardPoints on Windows” on page 214.

2. Register the protected host with the DSM with Efficient Storage enabled. For details, see “Register the Windows Host with DSM” on page 215.
3. Initialize the storage device to create a Private Region for the Efficient Storage Header. For details, see “Initialize Windows Efficient Storage Devices” on page 216.
4. Log on to the DSM to apply the ES GuardPoint to the storage device. For details, see “Guard the Windows Device with an Efficient Storage GuardPoint” on page 220.

Requirements for Efficient Storage GuardPoints on Windows

Windows-specific requirements:

- The Windows host must be running one of the following:
  - Windows Server 2012 R2
  - Windows Server 2016
  - Windows Server 2019
- You can *only* protect data volumes with an ES GuardPoint. Protecting the boot volume with an ES GuardPoint is *not* supported.
- You must enable Secure Start on the protected host and in the ES GuardPoint in the DSM.
- Existing data LUNs must be increased in size by at least 128MB (64MB for the VTE private region plus an additional 64MB to allow enough space for the data to be shifted after the VTE private region has been created). For details, see “Data Relocation on Existing Windows Devices” on page 221.

Note: If you create a new Windows device to be protected, do *not* initialize that device with the Windows Disk Manager until *after* you have created and assigned the ES GuardPoint. After the GuardPoint has been assigned through the DSM, you can manage the disk as normal using the standard Windows disk management tools.

A LUN must meet the following requirements before it can be protected as an ES GuardPoint:

- The storage array exporting the LUN to the protected host must be enhanced with the Efficient Storage capability. For details, see “Storage Arrays Compatible with VTE Efficient Storage” on page 194.
- The storage array exporting the LUN to the protected host must be a KMIP client registered with the same DSM 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.
- In a HyperV environment, the LUN *cannot* be a virtual disk.

Limitations for Efficient Storage GuardPoints on Windows

The current implementation of ES GuardPoints on Windows has the following limitations:

- VTE does not support dynamic disks.

- VTE does not support clusters, DFS/DFSR, or shared disk scenarios.
- All applications that access the Pure Storage LUN directly must be shut down while the devices are being initialized, guarded, and encrypted. Once you begin this process, devices must not be accessed by any other applications until all data has been transformed. If other applications do access the device, VTE may not be able to successfully apply the GuardPoint and the user may have to reboot the device.
- Once the process has started, Administrators cannot use any Disk Management tools to manage the devices. All disk administration must wait until after the process is complete.
- Once a device has been guarded, VTE does not support rekeying that device with a new encryption key. This functionality will be added in a later release of VTE.

Register the Windows Host with DSM

If the host is already registered with a DSM, you can initialize the devices you want to protect as Efficient Storage GuardPoints as described in “Initialize Windows Efficient Storage Devices” on page 216 as long as:

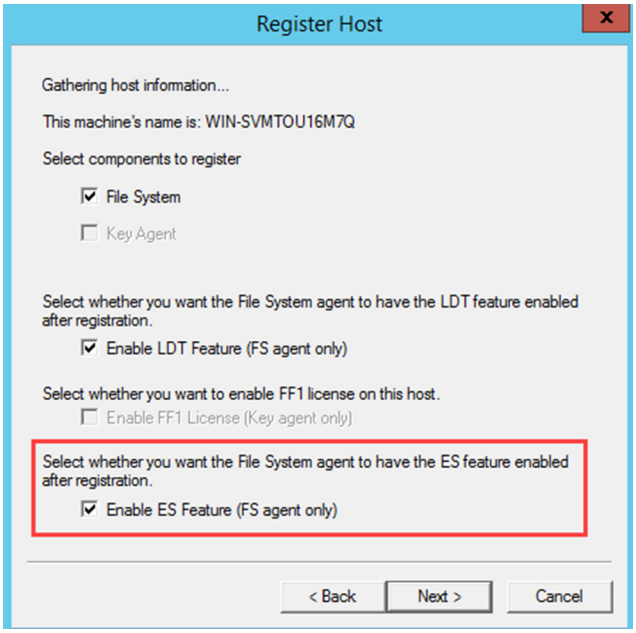
- The host is part of a DSM domain that has KMIP enabled.
- The DSM allows the host to register the File System component.
- A license for the Efficient Storage feature has been uploaded to the DSM.

To verify this, have the DSM Administrator check the host settings in the DSM.

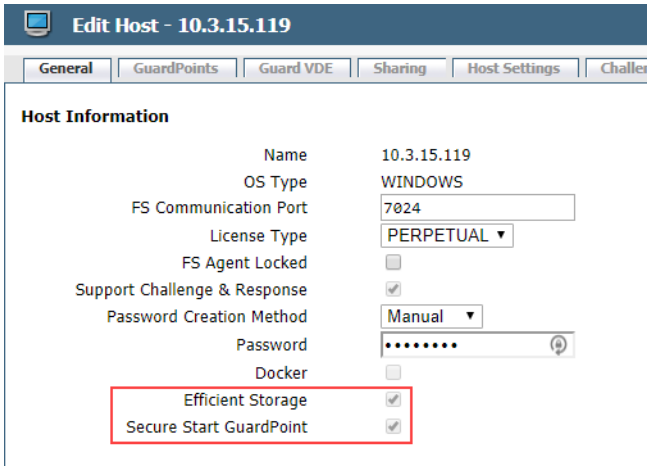
If the host is not already registered with a DSM, you need to register it before you can protect the devices on that host. When you register the host with the DSM, you must have AES-NI available on the host, a current license for registering host, and a current license for Efficient Storage. The following procedure describes how to register the host using the GUI. If you want to register the host silently, see “Windows Silent Install Using MSI or Self-Extracting .exe” on page 30.

1. If you are planning to use the fingerprint registration method, make sure the DSM Security Administrator has added the host to the appropriate domain and enabled at least File system and KMIP as registration options.
2. If necessary, install VTE on the host. Follow the installation prompts as until you get to the registration portion of the installation. If VTE is already installed, you can start the registration process by running `C:\Program Files\Vormetric\DataSecurityExpert\agent\shared\bin\register_host.exe`.

3. Follow the prompts to register the host with the DSM. During the registration process, make sure that you enable Efficient Storage support when you are selecting the features you want to register.



4. Complete the registration process as normal.
5. Reboot the machine after the registration process is complete.
6. To verify that the host is properly registered with the Efficient Storage option enabled, have the DSM Administrator verify that **Efficient Storage** and **Secure Start GuardPoint** are checked for the host as shown below.



7. After the host is registered with the DSM, initialize each device that you want to guard as described in the next section.

Initialize Windows Efficient Storage Devices

When you initialize a Windows efficient storage device, the process creates a private region on the device for VTE to write the Efficient Storage header along with metadata that identifies the ES device as a guarded device. The VTE 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. The initialization process also adds a user-defined label for the storage device that the DSM Administrator will use when referring to the device in the DSM.

Note: In Windows, this user-defined label is maintained across system reboots, allowing VTE to always find the device regardless of any device name changes that may happen within Windows.

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 New Windows Devices" on page 217
- "Initialize and Resize Existing Windows Devices" on page 218

Initialize New Windows Devices

For each new device you want to initialize, run the `voradmin esg config new` command. The `new` option specifies that the device does not hold user data, and that VTE can reserve the first 64MB of storage on the device for the VTE private region. The remaining storage space is available for new user data. The device size reported to applications is the actual device size minus VTE private region size.

WARNING

Do not use the `voradmin esg config new` command if the Windows disk has existing data that you want to keep. After you guard a device that has been initialized with this command, you will need to reformat the device and all existing data will be lost. To initialize a disk with existing data, see "Initialize and Resize Existing Windows Devices" on page 218.

1. Log into the device as an Administrator and open PowerShell or Cmd (command prompt).
2. Close all applications, including any Windows disk management tools, that are using or mounting the device.
3. Make sure you know the Device Names of the devices that you want to protect.

   To get a list of the Device Names for the available devices, use the `voradmin esg list disk` command and look in the **Device Name** column. Any new disk that is not a boot disk and that does not contain any data can be initialized by VTE as a new disk. For example:

```
C:\>voradmin esg list disk

Disk###  Device Name                    Boot Disk  Size       Status     Partition  Read Only  SERIAL NUMBER
-------  -----------------------------  ---------  ---------  ---------  ---------  ---------  ---------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0  Yes        127.0 GB   Online     MBR        No         6000c29d241599...
Disk1    \Device\00000032               No         49.9 GB    Online     MBR        No         6000c29b1d5a4c...
Disk2    \Device\00000033               No         50.9 GB    Online     MBR        Yes        6000c290582227...
Disk3    \Device\00000034               No         50.9 GB    Online     MBR        No         6000c290fd627b...
```

   In the example above, the available Device Names are `\Device\00000032`, `\Device\00000033`, and `\Device\00000034`.

4. Run the `voradmin esg config new <device-name>=<label>` command, where:
   - `new` (required) indicates that the device contains no data (it is a new disk). VTE will create the VTE private region at the beginning of the disk and the rest of the disk will be available for user data.
   - `<device-name>=<label>` (required) is the device name and a user-defined label for the device. This label will be the path the DSM Administrator uses to specify to the device in the DSM. (For example, `\Device\00000033=NewESDisk`.) The label can be 1 to 32 ASCII characters. Do not use spaces or special characters in the label.

     Make sure that the device you select does *not* contain any existing data. When VTE applies the GuardPoint to a new device, it removes the existing file system information from the device. That

means the device will need to be reformatted and all existing data will be unrecoverable as soon as the GuardPoint is applied.

For example, if you want to initialize a new disk Windows device named `00000033` with the label "NewESDisk", you would specify:

```
PS C:> voradmin esg config new \Device\00000033=NewESDisk
Disk is initialized successfully with VTE ESG protection.
```

5. To verify that the disk has been initialized, run the `voradmin esg status` command.

   This command shows that the device label has been set and the Xform Status has been set to NA (not applicable). For example:

```
C:\>voradmin esg status

Disk###  Device Name                        Boot Disk  ESG Device label                 Guard Status   Xform Status
-------  ---------------------------------  ---------  ------------------------------  -------------  -------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0      Yes        NA                              unguarded
Disk1    \Device\00000032                   No                                         unguarded
Disk2    \Device\00000033                   No         NewESDisk                       unguarded      NA
Disk3    \Device\00000034                   No                                         unguarded
```

6. At this point the DSM Administrator can protect the device as an ES GuardPoint through the DSM Management Console as described in "Guard the Windows Device with an Efficient Storage GuardPoint" on page 220.

Note: The initialization process prepares a device to be guarded but does not actually guard it. You need to assign an ES GuardPoint to the devices in the DSM before the devices are actually protected. In addition, the initialization process is only kept in memory until the device is guarded or rebooted. If a device is rebooted before you guard it, you will need to perform the initialization procedure again.

Initialize and Resize Existing Windows Devices

If a Windows device has existing data, you need to use the `voradmin esg config xform` command to initialize the disk for VTE. This command tells VTE that the data on the device needs to be encrypted after an ES GuardPoint is assigned to the device through the DSM. After the VTE 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 existing devices for VTE. 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. VTE 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 an Administrator and open PowerShell or Cmd (command prompt).
2. Close all applications, including any Windows disk management tools, that are using or mounting the device.
3. Make sure you know the Device Names of the devices that you want to protect.

   To get a list of available devices on Windows, use the `voradmin esg list disk` command. The Disk Name column shows the names of the available disks. In the list, existing disks must *not* be boot disks and they must *not* be Read Only. In the following example, `\Device\00000032` and `\Device\00000034` show **No** in the **Boot Disk** and **Read Only** columns:

```
C:\>voradmin esg list disk

Disk###  Device Name                    Boot Disk  Size       Status     Partition  Read Only  SERIAL NUMBER
-------  -----------------------------  ---------  ---------  ---------  ---------  ---------  ---------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0  Yes        127.0 GB   Online     MBR        No         6000c29d241599...
Disk1    \Device\00000032               No         49.9 GB    Online     MBR        No         6000c29b1d5a4c...
Disk2    \Device\00000033               No         50.9 GB    Online     MBR        Yes        6000c290582227...
Disk3    \Device\00000034               No         50.9 GB    Online     MBR        No         6000c290fd627b...
```

4. If you want to make sure the disk has not yet been initialized, used the `voradmin esg status` command. If the disk already has an ESG Device Label, then the disk has already been initialized. In the following example, Disk2 has already been initialized, but Disk1 and Disk3 have not:

```
C:\>voradmin esg status

Disk###  Device Name                        Boot Disk  ESG Device label                 Guard Status   Xform Status
-------  ---------------------------------  ---------  ------------------------------  -------------  -------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0      Yes        NA                              unguarded
Disk1    \Device\00000032                   No                                         unguarded
Disk2    \Device\00000033                   No         NewESDisk                       unguarded      NA
Disk3    \Device\00000034                   No                                         unguarded
```

5. For each existing device you want to initialize, run the `voradmin esg config xform <device-name>=<label>` command, where:

   - `xform` (required) indicates that the device contains existing data. VTE 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 unaccessible 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.
   - `<device-name>=<label>` (required) is the device name and a user-defined label for the device. This label will be the path the DSM Administrator uses to specify to the device in the DSM. (For example, `\Device\00000032=ExistWinDisk1`.) The label can be 1 to 32 ASCII characters. Do not use spaces or special characters in the label.

   For example, if you want to initialize a new disk Windows device named `00000032` with the label `ExistWinDisk1` and the device `00000034` with the label `ExistWinDisk2`, you would specify:

```
C:> voradmin esg config xform \Device\00000032=ExistWinDisk1
Disk is initialized successfully with VTE ESG protection. Disk must be Resized to at
least 128MB before guarding as Efficient Storage GuardPoint
C:> voradmin esg config xform \Device\00000034=ExistWinDisk2
Disk is initialized successfully with VTE ESG protection. Disk must be Resized to at
least 128MB before guarding as Efficient Storage GuardPoint
```

   With Windows, you always need to increase the disk size on each device by at least 128MB, which provides enough space for the VTE private region as well as room to relocate the existing data. After you guard the disk, you can expand it again later but you cannot shrink it unless you remove the GuardPoint. For details about the data relocation, see "Data Relocation on Existing Windows Devices" on page 221.

6. To verify that the disks have been initialized, run the `voradmin esg status` command.

   This command shows that the device labels have been set and the Xform Status has been set to Not Started. For example:

```
C:\>voradmin esg status

Disk###  Device Name                        Boot Disk  ESG Device label                 Guard Status   Xform Status
-------  ---------------------------------  ---------  ------------------------------  -------------  -------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0      Yes        NA                              unguarded
Disk1    \Device\00000032                   No         ExistWinDisk1                   unguarded      Not Started
Disk2    \Device\00000033                   No         NewESDisk                       unguarded      NA
Disk3    \Device\00000034                   No         ExistWinDisk2                   unguarded      Not Started
```

7. At this point, you need to resize all initialized existing devices by increasing their volume size through the Pure Storage management interface. Make sure you increase the device size on each device by at least 128 MB. For details, see your Pure Storage documentation.

   To verify that the disk size has been increased, use the `voradmin esg list disk` command.

```
C:\>voradmin esg list disk

Disk###  Device Name                    Boot Disk  Size       Status     Partition  Read Only  SERIAL NUMBER
-------  -----------------------------  ---------  ---------  ---------  ---------  ---------  ---------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0  Yes        127.0 GB   Online     MBR        No         6000c29d241599...
Disk1    \Device\00000032               No         50.1 GB    Online     MBR        No         6000c29b1d5a4c...
Disk2    \Device\00000033               No         50.9 GB    Online     MBR        Yes        6000c290582227...
Disk3    \Device\00000034               No         51.1 GB    Online     MBR        No         6000c290fd627b...
```

   You cannot assign an ES GuardPoint to the devices until it they have been resized. If you do not resize the devices, the GuardPoint assignment will fail.

8. After the devices have been resized, the DSM Administrator can protect the devices as an ES GuardPoint through the DSM Management Console as described in “Guard the Windows Device with an Efficient Storage GuardPoint” on page 220.

Note: The initialization process prepares the devices to be guarded but does not actually guard them. You need to assign an ES GuardPoint to each device through the DSM before the devices are actually protected. In addition, the initialization process is only kept in memory until the devices are guarded or rebooted. If a device is rebooted before you guard it, you will need to perform the initialization procedure again.

Guard the Windows Device with an Efficient Storage GuardPoint

Note: For details about how to create a GuardPoint, see the chapter, “Managing GuardPoints”, in the *DSM Administration Guide*.

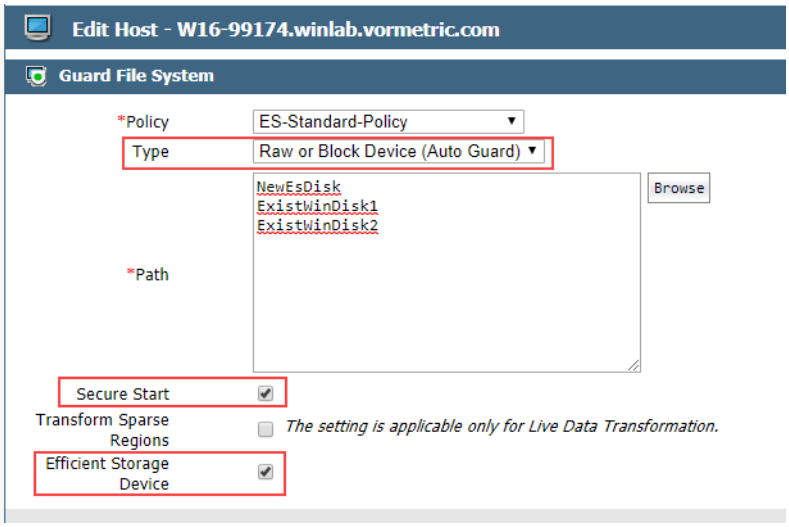
After the device has been initialized, you can guard the device as an ES GuardPoint from the DSM Management Console. For existing devices, as soon as the GuardPoint has been pushed to the host and the status changes to guarded, VTE begins transforming the data on the disk using the encryption key associated with the GuardPoint Policy.

1. Log on to the 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 standard policy if needed. The policy you use for Efficient Storage must use an XTS-AES256 key as the key rule. For details, see “ES GuardPoint Encryption Keys” on page 196.
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 File System window opens.
6. In the **Policy** field, select the Policy you identified or created earlier in this procedure. VTE will use the XTS-AES256 key associated with this policy to encrypt the data on the device.
7. In the **Type** field, select **Raw or Block Device (Auto Guard)**.

   When you select **Auto Guard**, VTE 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.
8. In the **Path** field, add the device label you assigned when you initialized the disk. For example, `ExistWinDisk1`.

   If you specify multiple device labels in this field, all specified devices will be guarded and all will be encrypted with the encryption key specified in the selected policy.
9. Make sure the **Secure Start** check box is checked.

10. Make sure the **Efficient Storage Device** check box is checked. If this option is not selected, the host will *not* enable the device as an ES GuardPoint.



11. Click **OK**.

    The DSM pushes the policy and the GuardPoint configuration to the host and the VTE agent on the host writes the ES header into the VTE 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. At this point you can use the Windows Disk Manager to perform any required disk management tasks and all data that gets written to the disk will be protected by VTE.

    If there is existing data on the device, VTE 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 size of the disk. For details, see "Data Transformation on Existing Windows Devices" on page 221.

Data Relocation on Existing Windows Devices

When you add a GuardPoint to a device that has been initialized with the `xform` option, VTE shifts the existing data by 64MB, then it creates the VTE private region in the first 64MB on the device. This relocation occurs only once when the device is guarded for the first time.

Data Transformation on Existing Windows 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 referred to In-Place Data Transformation, or IDT for short.

IDT is not the same as the legacy offline data transformation. IDT is a block level data transformation with built-in resiliency to recovery from system crashes during the data transformation process. IDT uses the private region on the device to manage the entire transformation process. IDT partitions the data on a device in segments of 512KB 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 private region of the device, and then transforms the data in-place. IDT also maintains the segments undergoing transformation in the 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. With the same production policy applied to the device, IDT determines whether the device is in need of data transformation, per specification of `xform` option when device was initialized, and starts the IDT process when transformation is required. During the IDT process, access to the device is blocked until IDT process completes.

To view the data transformation status, use the `voradmin esg status` command and look in the **Xform Status** column. In the following example:

- Disk1 has been guarded and the data transformation process has completed, so the device is guarded and ready to use.
- Disk2 was initialized as a new device, so no data transformation was required. The device is guarded and ready to use.
- Disk3 has been guarded but the data transformation process is still in progress. This device cannot be accessed until the data transformation process has completed.

```
C:\>voradmin esg status

Disk###  Device Name                          Boot Disk  ESG Device label                 Guard Status   Xform Status
-------  ---------------------------------  ---------  -------------------------------  -------------  -------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0      Yes        NA                               unguarded
Disk1    \Device\00000032                   No         ExistWinDisk1                    guarded        Completed
Disk2    \Device\00000033                   No         NewESDisk                        guarded        NA
Disk3    \Device\00000034                   No         ExistWinDisk2                    guarded        In-Progress (18%)
```

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.

Windows System and ES GuardPoint Administration

The `voradmin` command is a command line utility for management of VTE specific configuration and status reporting. The `voradmin` command also supports configuration management related ES GuardPoints (ESG).

Windows supports the following `voradmin esg` commands.

voradmin esg list disk

Command: `voradmin esg list disk`

Lists the disks available on the Windows host. The Disk### column matches the disk numbers in the Windows Disk Manager. The Device Name column shows the name of the disk that you need to use with other `voradmin` commands. In the following example, the Device Name for Disk1 is \Device\00000036.

Example output:

```
C:\>voradmin esg list disk

Disk###  Device Name                    Boot Disk  Size       Status     Partition  Read Only  SERIAL NUMBER
-------  ------------------------------ ---------  ---------  ---------  ---------  ---------  ---------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0  Yes        127.0 GB   Online     MBR        No         6000c29d241599...
Disk1    \Device\00000032               No         49.9 GB    Online     MBR        No         6000c29b1d5a4c...
Disk2    \Device\00000033               No         50.9 GB    Online     MBR        Yes        6000c290582227...
Disk3    \Device\00000034               No         50.9 GB    Online     MBR        No         6000c290fd627b...
```

voradmin esg config

Command: `voradmin esg config [new|xform] <device-name>=<device-label>`

Initializes a new or existing Windows device so that it can be protected as an ES GuardPoint through the DSM. For details on using this command, see "Initialize New Windows Devices" on page 217 and "Initialize and Resize Existing Windows Devices" on page 218.

voradmin esg status

Command: `voradmin esg status`

Displays the status of the disks on the host. The **Disk###** column matches the disk numbers in the Windows Disk Manager. The **Device Name** column shows the name of the disk that you need to use with other `voradmin` commands. In the following example, the Device Name for Disk1 is \Device\00000032.

If the device has been initialized, the user-defined disk label appears in the **ESG Device Label** column. If the device has been protected with an ES GuardPoint through the DSM, the **Guard Status** column displays "guarded". The **Xform Status** column displays the status of any data transformation processes run on the disk. NA means it was a new disk, so no data transformation was needed. For existing disks, the Xform Status can be Not Started, In Progress, or Completed.

Example output:

```
C:\>voradmin esg status

Disk###  Device Name                        Boot Disk  ESG Device label                 Guard Status   Xform Status
-------  ---------------------------------  ---------  -------------------------------  -------------  -------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0      Yes        NA                               guarded
Disk1    \Device\00000032                   No         ExistWinDisk1                    guarded        Completed
Disk2    \Device\00000033                   No         NewESDisk                        guarded        NA
Disk3    \Device\00000034                   No         ExistWinDisk2                    guarded        Completed
```

voradmin esg delete

Command: `voradmin esg delete <disk-label>`, where:

`<disk-label>` is the user-defined label that was specified when the device was initialized. To view a list of disk labels, use the `voradmin esg status` command.

Removes the ES header from the specified device. The device *cannot* be protected as an ES GuardPoint or this command will fail.

Example output:

```
# voradmin esg delete disk NewESDisk
ES disk header deleted successfully.
```

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` on Linux, or the native disk management tools on Windows.

WARNING

Do not shrink ES GuardPoints. Due to relocation of user data from VTE 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 is enabled on the DSM, or manual-guard if 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. On Windows you can use the `voradmin esg list disk` 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 in Figure 20-3: “Original size of storage array”, resized to 43,395,072 sectors in Figure 20-4: “Resized storage array”.

```
# 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:                       b1c449bd-db1e-3a8f-b829-5893dd2fd0b0
        Mount Path:                     N/A (GuardPoint not enabled)
# umount /xfs
# secfsd -unguard /dev/sdh
secfsd: Path is not guarded
```

Figure 20-3: Original size of storage array

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:                       b1644bbd-db1e-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:                       b1644bbd-db1e-3a8f-b829-5893dd2fd0b0
        Mount Path:                     /xfs
```

Figure 20-4: Resized storage array

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
```

Figure 20-5: XFS file system to including the extended space

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

This use case applies to both Linux and Windows.

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 GuardPoint(s) specifies one key rule for encryption, and, on Linux only, 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.

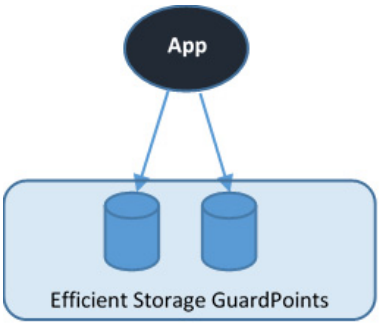


Figure 20-6: Protect structured data stored in LUNs guarded as Efficient Storage

Use Case 2

This use case applies to both Linux and Windows.

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 on Linux and NTFS or ReFS on Windows, 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.

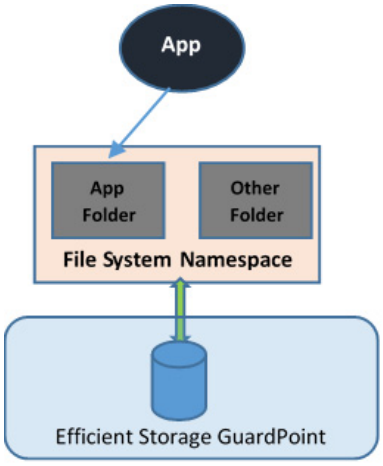


Figure 20-7: File system resides in device guarded as ES GuardPoint

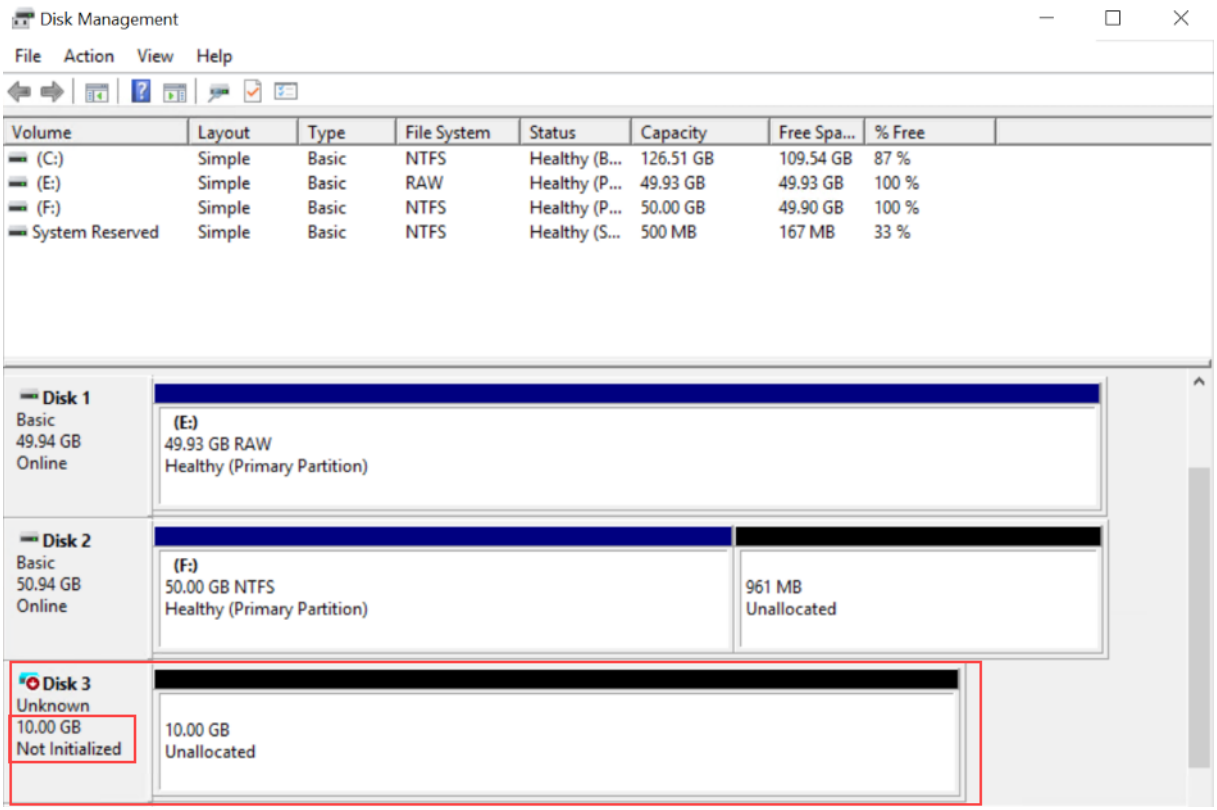
Linux Example

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-db1e-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                   extsz=4096   blocks=0, rtextents=0
# mount -t xfs /dev/secvm/dev/sdh /xfs
# mount | grep xfs
/dev/secvm/dev/sdh on /xfs type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

Windows Example

In the following example, the Windows Administrator has created a new 10 MG VHD in the Windows Disk Management tool. This VHD is called Disk 3, and it has not yet been initialized.



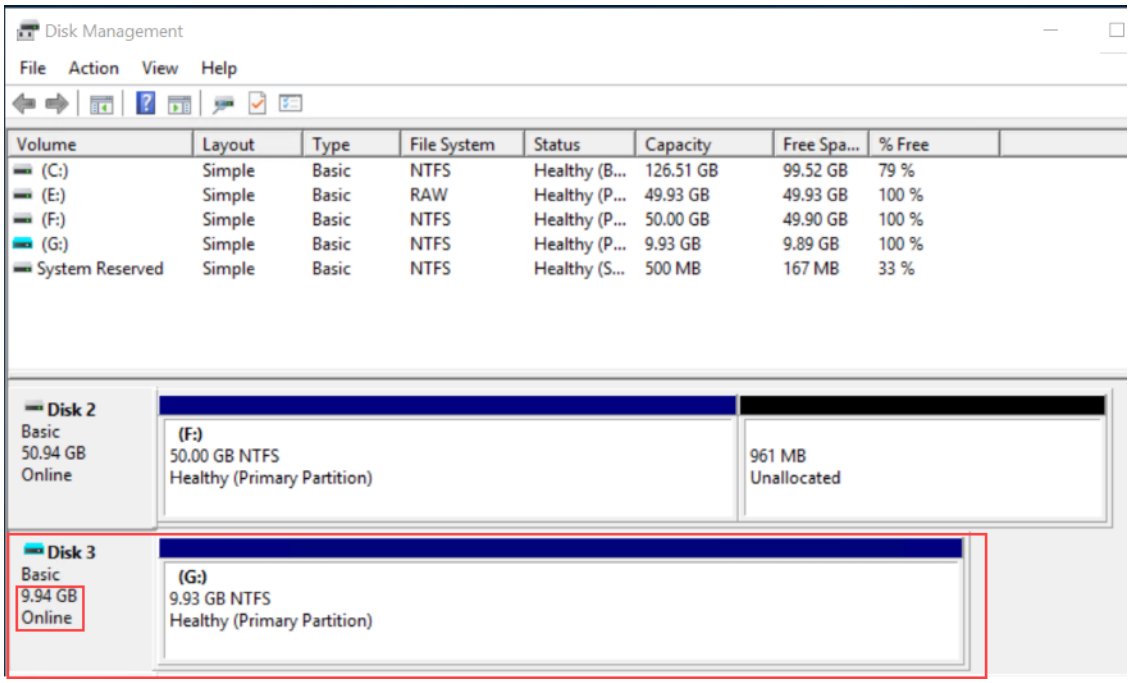
The Windows Administrator then uses `voradmin esg list disk` to get the VTE device name for the new disk and initializes it using the `voradmin esg config new` command, as shown:

```
C:\>voradmin esg list disk

Disk###  Device Name                          Boot Disk  Size       Status     Partition  Read Only  SERIAL NUMBER
-------  -----------------------------------  ---------  ---------  ---------  ---------  ---------  ------------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0        Yes        127.0 GB   Online     MBR        No
Disk1    \Device\00000032                     No         49.9 GB    Online     MBR        No
Disk2    \Device\00000033                     No         50.9 GB    Online     MBR        No
Disk3    \Device\00000051                     No         10.0 GB    Online     MBR        No

C:\>voradmin esg config new \Device\00000051=NewDisk3
Disk is initialized successfully with VTE ESG protection.
```

The Administrator guards the new disk through the DSM, and uses the `voradmin esg status` command to make sure the new disk has been successfully guarded.

```
C:\>voradmin esg status

Disk###  Device Name                          Boot Disk  ESG Device label                 Guard Status   Xform Status
-------  -----------------------------------  ---------  -------------------------------  -------------  -------------
Disk0    \Device\Ide\IdeDeviceP0T0L0-0        Yes        NA                               unguarded
Disk1    \Device\00000032                     No         esg-disk1-demo                   guarded         Completed
Disk2    \Device\00000033                     No         esg-disk2-demo                   guarded         Completed
Disk3    \Device\00000051                     No         NewDisk3                         guarded         NA
```

After the device has been guarded, the Administrator returns to the Windows Disk Manager and selects **Action > Rescan Disks** to make sure the Windows Disk Manager is synchronized with VTE. They then initialize the disk, create a new volume for it, and format it. Notice that the new volume size is slightly smaller than the original 10 GB because VTE has reserved room for the VTE Private Region.



Use Case 3

The basic use case applies to both Linux and Windows, but the challenges are only applicable to Linux.

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 on Linux and NTFS or ReFS on Windows, 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. Like the previous use case, the policy applied to the ES GuardPoint specifies a key rule, and, for Linux only, potentially the second key rule with empty resource set for rekey, and no access rules.

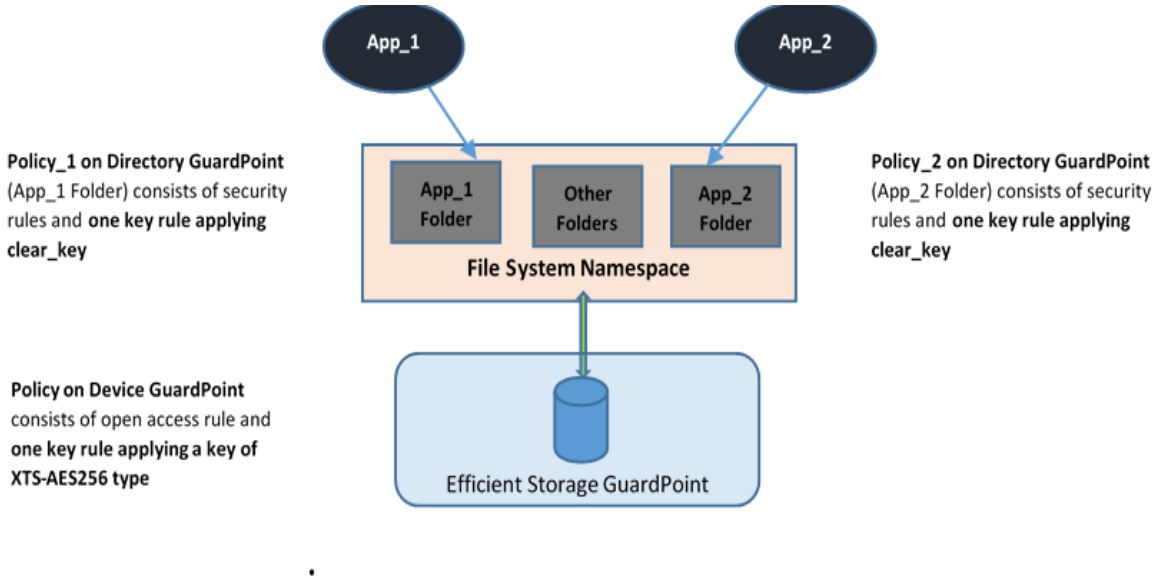
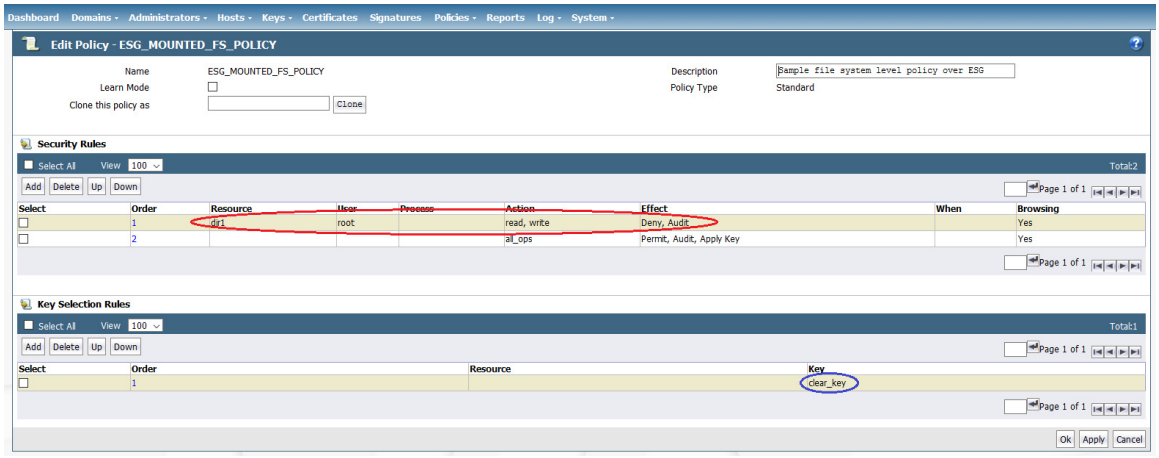


Figure 20-8: All Data in file system Device Encrypted through an ES GuardPoint

In this use case, two policies are enforced to protect the data in the device. As stated, one policy protects the entire file system namespace guarded under the Directory GuardPoint, and the guarded directory is the mount point pathname of the mounted file system. The policy on the mount point directory must specify *clear_key* in the key rule. Specifying a key other than *clear_key* must be avoided because all the data in the file system device is encrypted through the ES GuardPoint corresponding to the file system device. The screen shot below represents a sample Linux policy over the file system mount point where the policy blocks root access to read/write files under the subdirectory `dir1` in the mounted file system.



The second Linux policy protecting the device is the same policy as use case 2.

Linux Example

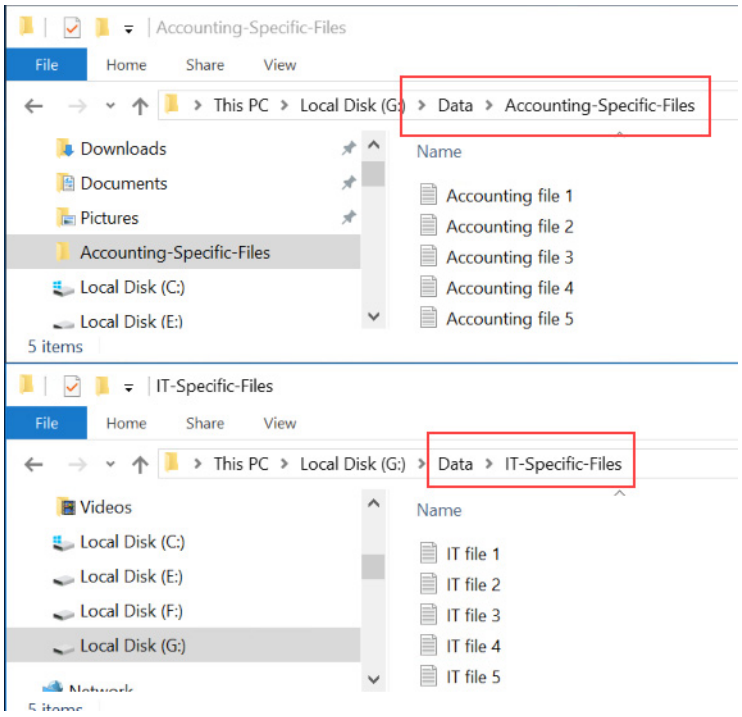
Below is an example of this use case where a file system created in a guarded device and mounted on `/xfs` is protected under a policy that denies root access to the files under `/xfs/dir1`:

```
# 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.
```

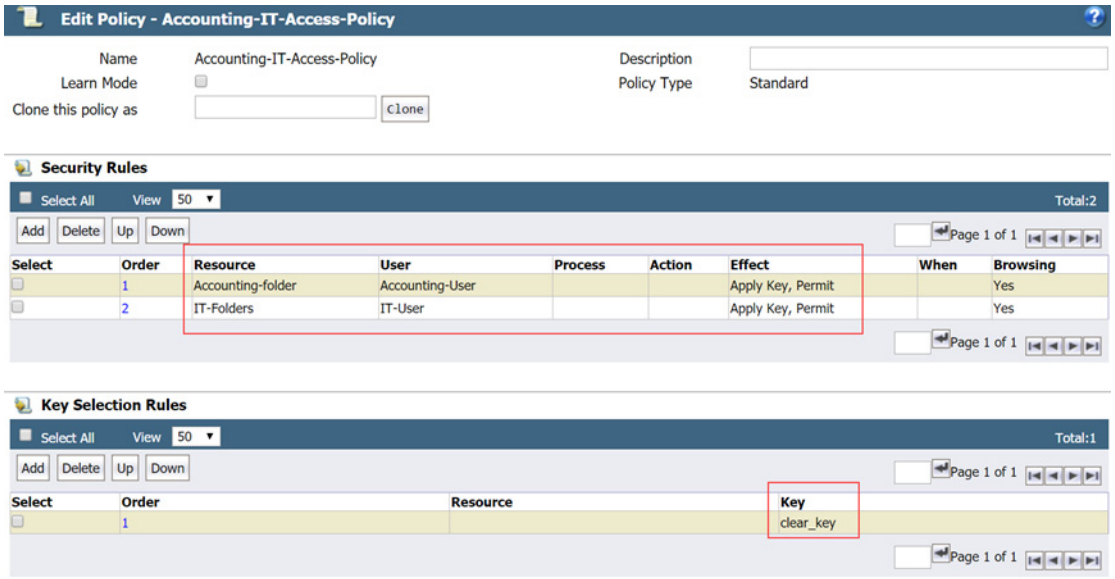
As depicted above, the root user is denied access to read/write the files associated with the resource set representing files under `/xfs/dir1` subdirectory.

Windows Example

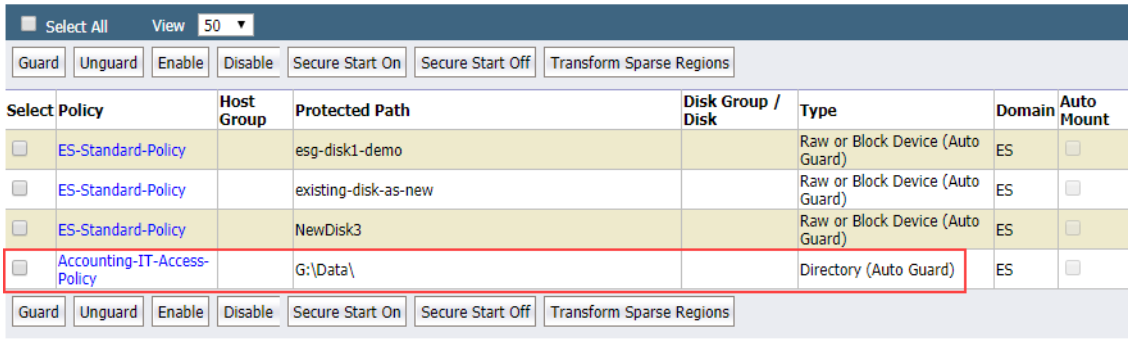
In this example, Jane Doe is a member of the Accounting team and John Fredricks is a member of the IT team. There are two folders on the guarded disk called `G:\Data\Accounting-Specific-Files` and `G:\Data\IT-Specific-Files`. Even though the disk is protected by an Efficient Storage GuardPoint, both Jane and John can see the files in either folder. For example:



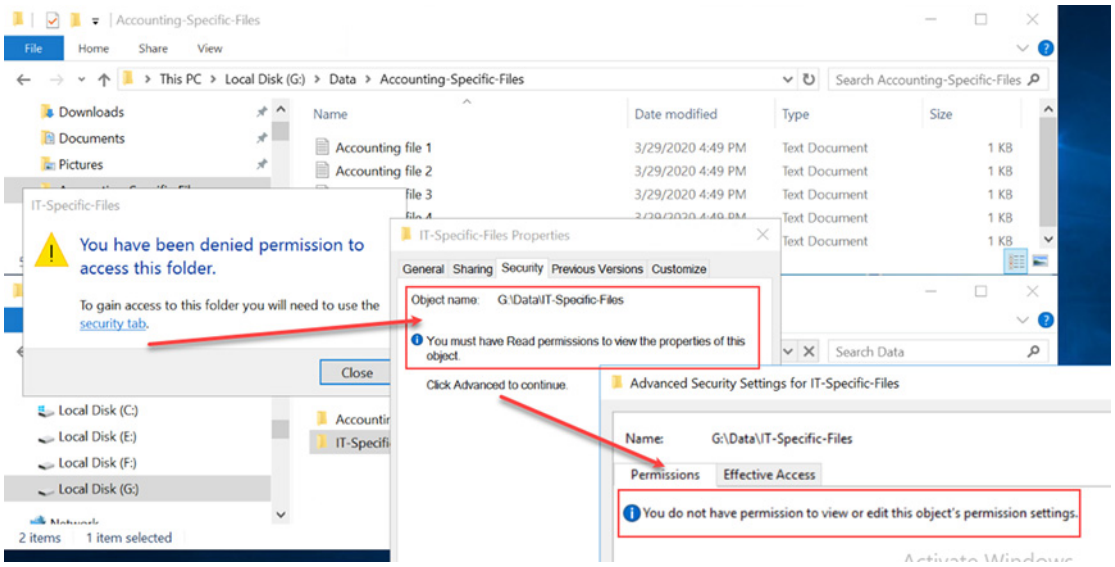
The DSM Administrator goes into the DSM and creates a standard policy that grants access to Jane Doe in the Accounting folder and grants access to John Fredricks in the IT folder. For the key rule, the policy must specify `clear_key`. For example:



Then the DSM Administrator applies the policy to `G:\Data`. For example:



Now when Jane Doe logs into the server, she can see the files in the `Accounting-Specific-Files` directory but she cannot access the files in the `IT-Specific-Files` directory, even if her account has Administrator-level access. For example:



Similarly, when John Fredricks logs in he will be able to access the files in the `IT-Specific-Files` directory but he will be unable to access the `Accounting-Specific-Files` directory.

Challenges with Root Access on Linux

This challenge is only applicable to 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

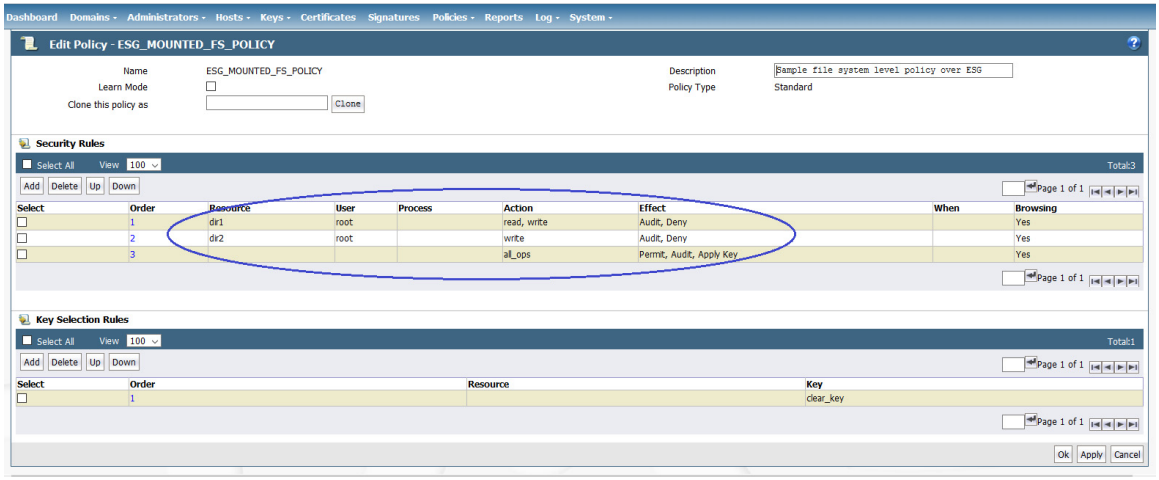
As the above example shows, the policy ESG_MOUNTED_FS_POLICY denies the root user access to the files associated with the resource set `dir1`. Enforcement of the rule become ineffective as soon as the GuardPoint on the file system mount point is disabled. Since data is in clear-text at the file system 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 DSM. With this option, VTE immediately guards the mount point directory as soon as the file system mounts, and similarly, VTE 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.



Challenge 2 - Deny root access to view sensitive data in protected Linux files

Another challenge with root user privilege is that root can still view sensitive information stored in the ES GuardPoint device. As explained, the policy ESG_MOUNTED_FS_POLICY denies root access to the read/writes files under `/xfs/dir1`. Although this policy enforces the rule on the Directory GuardPoint `/xfs`, the rule is not enforced if the root user dumps the content of the ES GuardPoint. As the example below shows, the root user can view sensitive information in the protected files under `/xfs/dir1`.

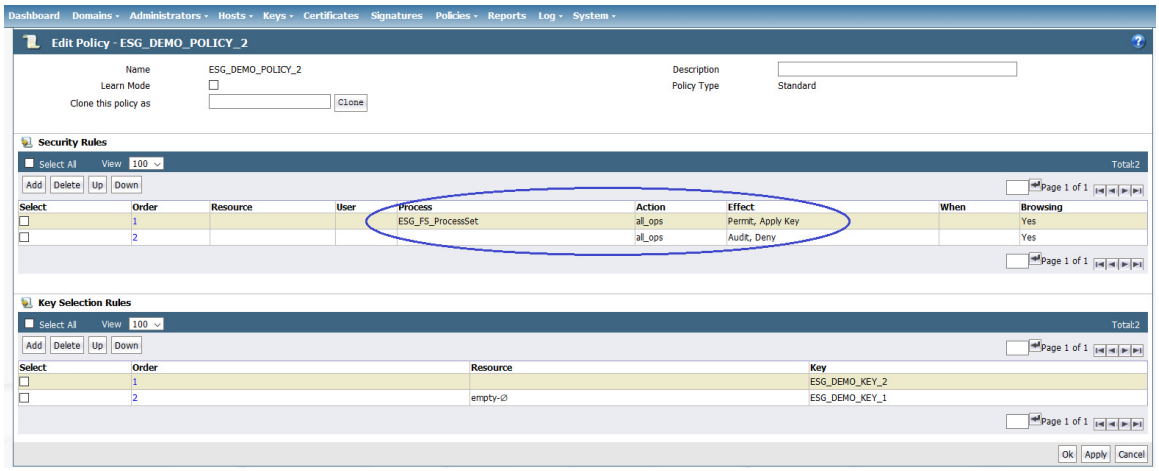
```
# secfsd -status guard | grep ESG
/dev/sdh                    ESG_DEMO_POLICY_2      manualrawdevice  guarded      guarded      N/A
/xfs                        ESG_MOUNTED_FS_POLICY  automount        guarded      guarded      N/A

# cat /xfs/dir1/secret
cat: /xfs/dir1/secret: No such file or directory

# dd if=/dev/secvm/dev/sdh bs=1048576 | grep --binary-files=text "file holds highly sensitive data"
1▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒[▒-m▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒K▒Dn▒▒▒P▒Km▒▒▒▒IN▒▒▒▒9D▒co▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒x▒▒p▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒D}▒▒q▒▒▒P▒Km▒▒▒▒IN
▒▒▒mr▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒bxs▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒M▒Lt▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒`B▒ku▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒;R▒v▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒]▒%w▒▒▒P▒Km▒▒▒▒
▒y▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒e▒z▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒OJ▒{▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒Pq▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒JV}▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒Z?~▒▒▒P▒Km▒▒▒▒IN▒▒▒▒▒P
▒▒P▒Km▒▒▒▒This file holds highly sensitive data.
This file holds highly sensitive data.
ad▒▒▒▒▒▒▒This file holds highly sensitive data.This file holds highly sensitive data.
4578+0 records in
4577+0 records out
4799332352 bytes (4.8 GB) copied, 32.3096 s, 149 MB/s
```

To block the root user from dumping context of the device, the policy on the ES GuardPoint must enforce a security rule that denies root access to the GuardPoint device. However, denying root access to GuardPoint devices is not feasible because file system utilities require root access for file system administration. Rather than enforcing complete denial of root access, you can impose a restriction that allows only a limited set of system utilities to access ES GuardPoints, such as `mount`, `fsck`, `mkfs`, `dmsetup`, etc., and deny access to other utilities, such as `dd`.

The solution is to restrict root to execute a limited set of utilities, which do not expose content of the file system devices, on Efficient Storage Devices. If the root user attempts to execute other utilities on the device, the root user will be denied access. You can impose this restriction by selecting the administrative utilities that root must execute on ES GuardPoints. See “Linux System Utilities for Signing” on page 211 for the list of utilities that root must be able to execute.

To implement this solution, you can create a signature set on your DSM and add the system utilities that root is permitted to execute on ES GuardPoints. Those utilities can be added to the signature set for signing. After signing the binary files of those system utilities, you can add a security rule to the policy on the ES GuardPoint that grants root the right to execute the system utilities in the signature set to access the ES GuardPoint. In the above example, since `dd` is not in the signature set, the `dd` command is denied access to read the file system device guarded as an ES GuardPoint.

The following screenshot depicts the policy that restricts root access to the ES GuardPoint limited to the system utilities listed in ESG_FS_ResourceSet.



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. On the DSM, click on **Signature** in the top menu bar to get to the **Signature Sets** page, and then 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.

In the example below, the signature set named ESG_FS_Utilities includes sample of file system utilities whose binary files have been signed on the protected host. You may add other system utilities to the set as necessary.

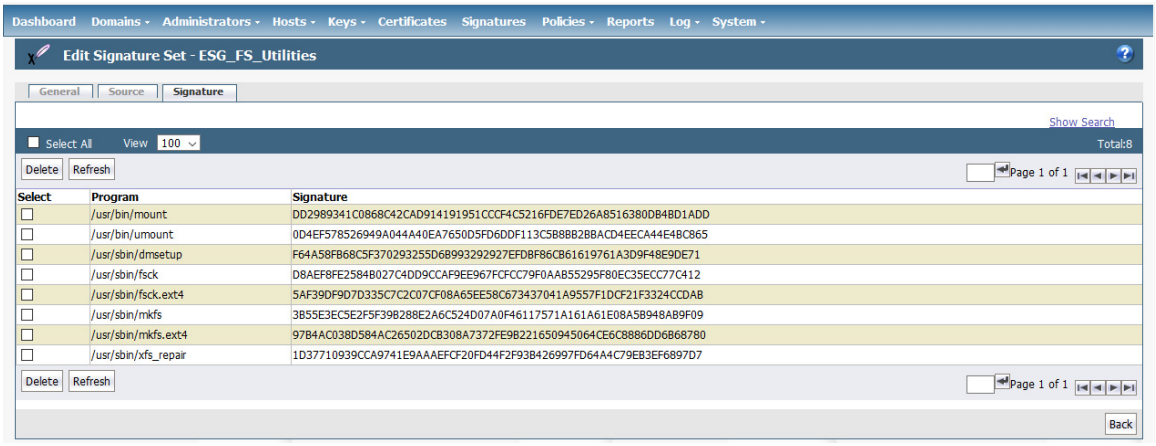
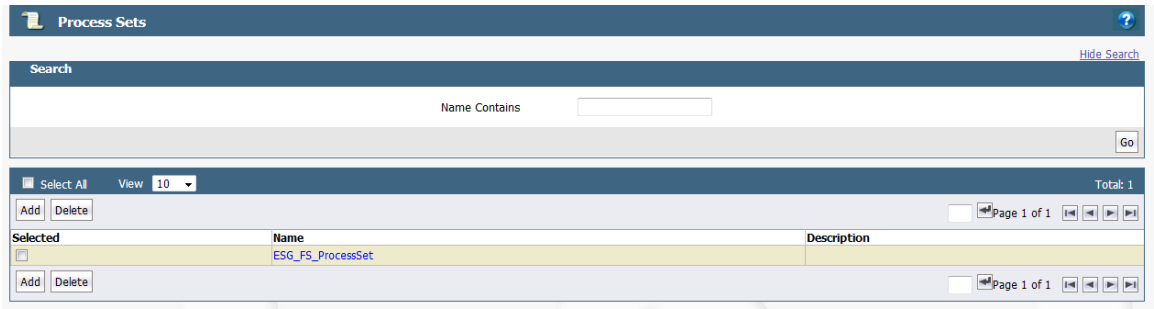


Figure 20-9: Signature Set ESG_FS_Utilities includes file system utilities

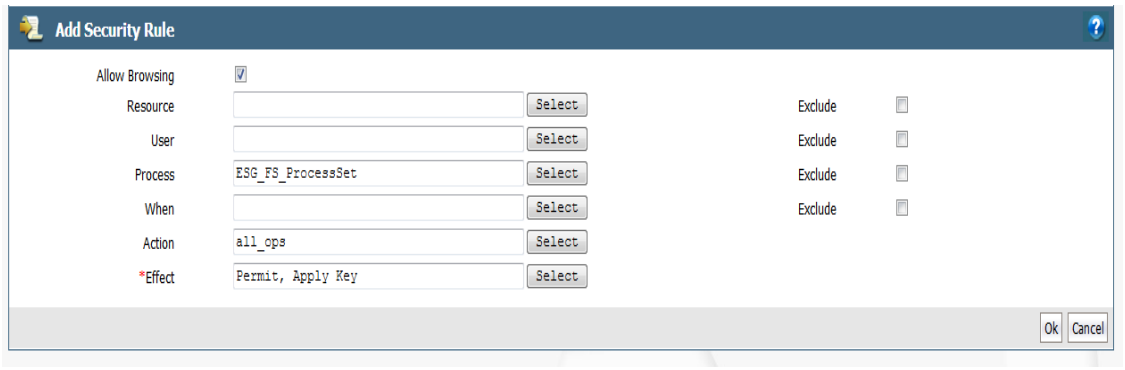
At this point, you have created a signature set consisting of the system utilities that root is allowed to execute on Efficient Storage devices. Next, create a process set from the signature set. The process set will be included in a security rule in the policy protecting the ES GuardPoint. The security 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_Utilities 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. 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**.
7. On the **Add Process Set** page, click OK to create the process set associated with the selected signature set and the protected host.

   In the following figure, the process set named ESG_FS_ProcessSet has been created using the ESG_FS_Utilities signature set which includes system utilities whose binary files have been signed on the protected host.



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.

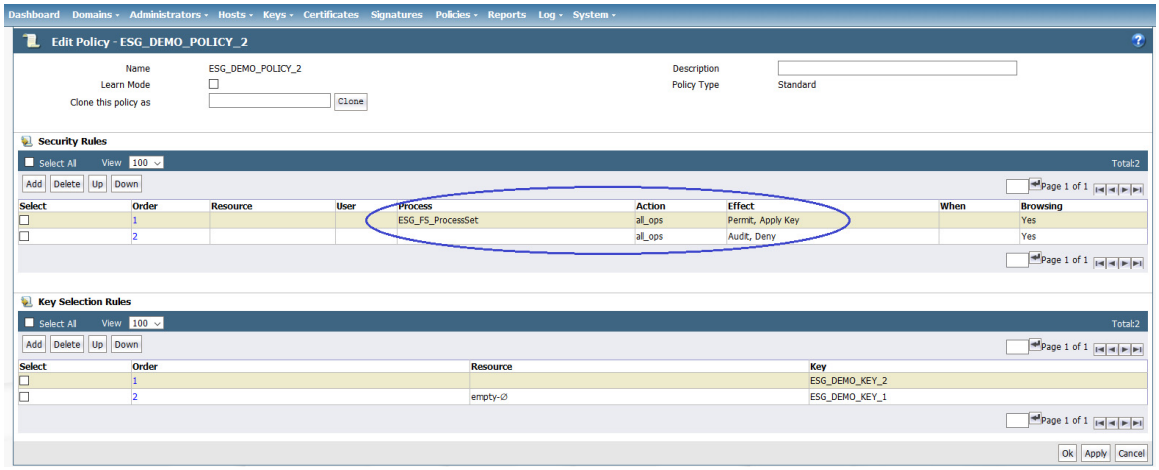


Figure 20-10: Policy protecting ES GuardPoints after adding security rule

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/dir1/secret
cat: /xfs/dir1/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
```

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 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 DSM 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 VTE private region.

Solution: Unguard the GuardPoint from the DSM, resize the LUN and verify that the host sees the expanded size, and then guard the GuardPoint from the DSM.

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 Vormetric 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 DSM

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 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 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 VTE shutdown when there is a busy VTE 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 VTE 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 VTE startup when system fails to mount a file system. This error message is typically accompanied by a long timeout during the VTE 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 DSM. It occurs when VTE 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 Vormetric 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 DSM. It occurs when protected host encounters an error transforming the data on device during IDT process.

Solution: Please contact Vormetric Support for troubleshooting and recovery.

ESG-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 ES GuardPoint.

ESG-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 ES header on the device.

Solution: Please contact Vormetric 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 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. Refer to “Linux System Utilities for Signing” on page 211.

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.

Alerts and Errors on Windows

ESG-ALERT: Data transformation failure on [GuardPoint]

This is an alert message to the DSM. It occurs when a protected host encounters an error transforming the data on the device during the IDT process.

Solution: Please contact Vormetric Support for troubleshooting and recovery.

ESG-INFO: Data transformation complete on [GuardPoint]

This message notifies the DSM administrator that the protected host has completed the data transformation on the specified ES GuardPoint.

Disk label validation failed. Check your disk label and run command again.

This error occurs when you run the `voradmin esg config` command on an ESG device and you specify a disk label that is too long or that contains unsupported characters.

Solution: Check disk label and make sure that it meets the label name requirements.

Failed to get disk information

This error occurs when ESG found the device but cannot open it.

Solution: Please contact Vormetric Support for troubleshooting and recovery.

Boot partition is present on the disk. Disk or LUN can not be protected using VTE agent.

This message occurs when you run the `voradmin esg config` command on a disk that has a boot partition. ESG only supports protecting the data disks. You cannot use ESG to protect the system or boot disks.

The disk is dynamic disk. This disk or LUN can not be protected using VTE agent.

This message occurs when you run the `voradmin esg config` command on a dynamic disk. ESG does not support guarding dynamic disks.

Failed to initialize disk

This error message occurs when the `voradmin esg config` command fails to initialize the disk.

Solution: Please contact Vormetric Support for troubleshooting and recovery.

Disk is already initialized/guarded with VTE ESG protection

This message occurs when you run the `voradmin esg config` command on a disk is already protected by an ES GuardPoint or has already been initialized using the `voradmin esg config` command. The error message shows the disk label that was assigned to the device when it was initialized.

To determine the ESG status of all disks on the host, use the `voradmin esg status` command.

Failed to initialize disk with VTE ESG protection. Size must be greater than %xMB, Current size: %yMB

This error occurs when there is not enough free disk space to store the ES header in the VTE private region on the disk.

Solution: Increase the disk size to at least the size shown in the error message.

Disk is initialized successfully with VTE ESG protection.

This notification indicates that VTE ESG protection has been successfully applied to the disk.

Disk is initialized successfully with VTE ESG protection. Disk must be Resized to at least 128MB before guarding as Efficient Storage GuardPoint

This message occurs after you use the `voradmin esg config xform` command on an existing disk. It indicates that the initialization was successful but that you must now increase the disk size by at least 128MB before you can apply the ES GuardPoint through the DSM.

Failed to initialize disk with VTE ESG protection. The specified disk does not exist or is not online.

This error message occurs when VTE cannot find the disk specified on the `voradmin esg config` command.

Solution: Check in the disk management utility to make sure the disk is available on the system. If it has gone offline, bring it back online and re-submit the `voradmin esg config` command. If the disk appears to be online and can be accessed by other applications but ESG still cannot find it, please contact Vormetric Support for troubleshooting and recovery.

Disk with specified label does not exist. Please select another disk.

This error occurs when the disk label specified on a `voradmin esg` command does not exist.

Solution: Check the disk label and resubmit the command. To see all available disk labels, use the `voradmin esg status` command.

Header deletion failed with error code

This error occurs when the Administrator enters the `voradmin esg delete <disk-label>` command for a valid ES disk but VTE cannot delete the ES header from the specified disk.

Solution: Please contact Vormetric Support for troubleshooting and recovery.

Disk is protected with VTE ESG. Please unguard the disk before deleting ESG header.

This error occurs when the Administrator enters the `voradmin esg delete <disk-label>` command but there is still an ES GuardPoint assigned to the disk in the DSM.

Solution: Remove the ES GuardPoint through the DSM and then re-submit the `voradmin esg delete` command.

VTE ESG header deleted successfully.

This message indicates that VTE has successfully deleted the ESG header on the disk specified in the `voradmin esg delete <disk-label>` command.

VTE ESG header does not exist on the selected disk. Please select another disk

This message occurs when the Administrator enters the `voradmin esg delete <disk-label>` command but VTE cannot find an ES header on the specified disk.

Solution: Check the disk label and resubmit the request. If the disk label is correct and the problem persists, please contact Vormetric Support for troubleshooting and recovery.

Chapter 21: InPlace Data Transformation for Linux

This chapter contains the following sections:

Introduction to InPlace Data Transformation (IDT)

VTE offers InPlace 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 InPlace 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 VTE Private Region to manage the entire transformation process (For details, see "The VTE Private Region and IDT Device Header" on page 246).

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 VTE Private Region, and then transforms the data in-place. IDT also maintains the segments undergoing transformation in the VTE 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

- IDT-Capable GuardPoints are available for Linux with VTE 6.3.1 or later. They also require DSM version 6.4.2 or later.
- The host server must use the Advanced Encryption Standard instruction set (AES-NI).
- The DSM policy assigned to the IDT-Capable GuardPoint must be of type “In-place Data Transformation - Device” 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 VTE 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/secvm/dev/sda1` is invalid as `/dev/secvm/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.

The VTE Private Region and IDT Device Header

IDT-Capable GuardPoints require a small amount of disk space in the standard VTE Private Region. The reserved space is where VTE 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 VTE Private Region is known as the IDT Device Header. By default, when you initialize a device as an IDT-Capable GuardPoint, VTE reserves 63 MB of space starting at the first sector on the device for the VTE Private Region.

VTE writes the IDT Device Header into the VTE 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 VTE creates the VTE Private Region starting at the first sector. For details, see “Initialize a Linux Device with Existing Data” on page 252.

VTE Private Region Location

Normally, VTE requires that the VTE Private Region be embedded at the beginning of the device. IDT, however, allows you to specify that the VTE Private Region for an IDT-Capable GuardPoint should be located in a central VTE metadata directory on the host called `<vte-install-dir>/vte/vte-metadata-dir` (default: `/opt/vte/vte-metadata-dir`). If you use this option, VTE stores the VTE Private Region and IDT Device Header for the device in this directory. The location of the VTE 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 251.

WARNING

Access to the VTE metadata directory is local to the VTE protected host. Devices whose access is shared across multiple VTE protected hosts in a cluster *must* be configured with the VTE Private Region embedded in those devices. Using a centralized metadata directory for shared devices will lead data corruption.

The location of the VTE Private Region does not affect VTE's functionality, but there are some considerations if you choose to use the centralized metadata directory `<vte-install-dir>/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 VTE Private Region.
- The centralized VTE metadata directory must be guarded by the DSM administrator to prevent accidental modification or deletion of the VTE metadata. If the VTE 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-install-dir>/vte/vte-metadata-dir`.
- Devices with existing data do not need to be resized to accommodate the VTE 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 VTE Private Region is not embedded on the device.

Device Size

If you embed the VTE 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 VTE 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 VTE 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 VTE 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 VTE. 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.

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.

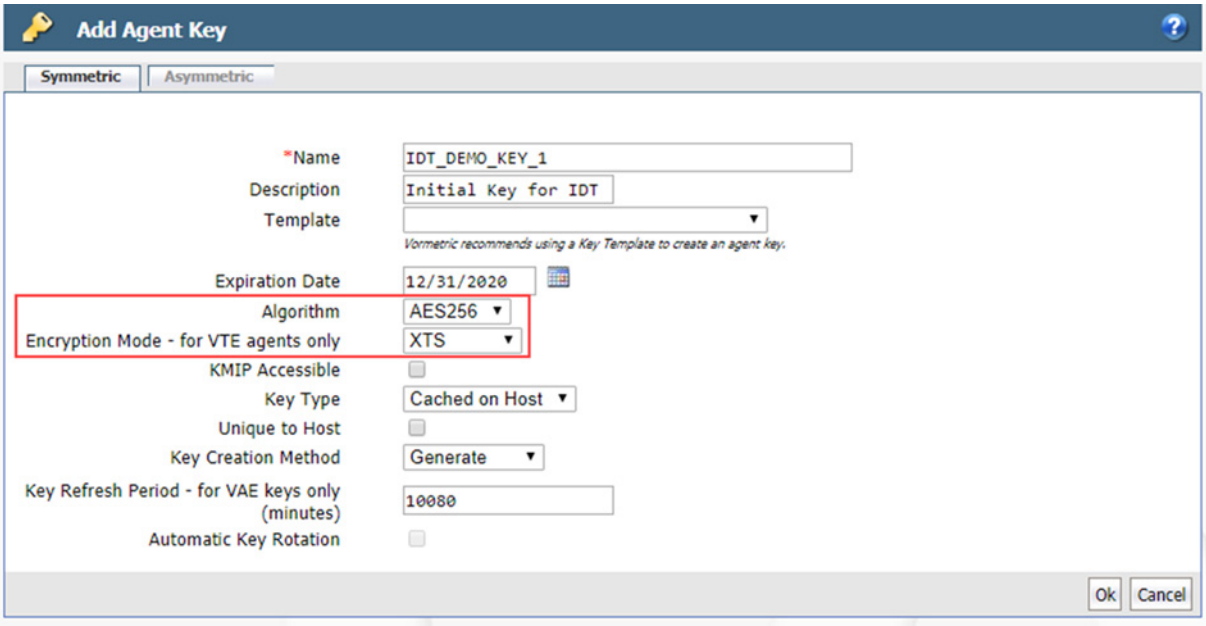
You create XTS keys on the DSM using the "Add Agent Key" function.

Note: When you create agent keys for IDT-Capable GuardPoints, you do not need to check the "KMIP Accessible" box on the "Add Agent Key" page. If you do check the KMIP Accessible option, the IDT-Capable GuardPoint ignores this setting.

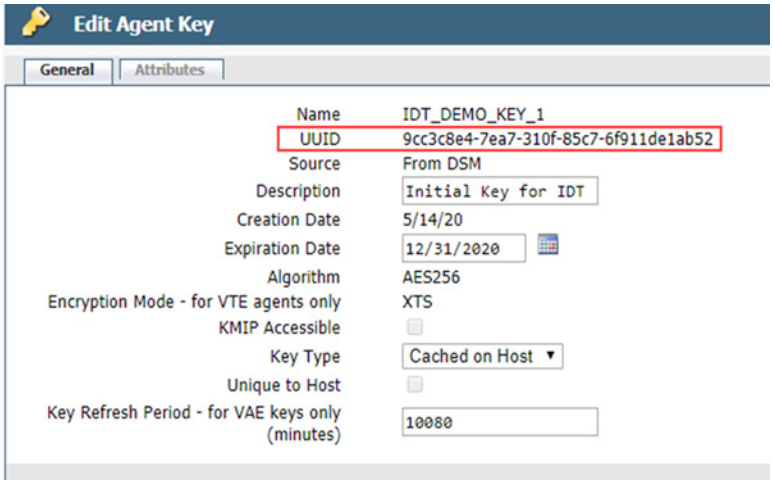
The DSM generates a UUID, along with other relevant attributes, for each newly-added key. The DSM then provides the key and its attributes to the VTE protected host when the policy containing the key is pushed to the host device. VTE stores the key and its attributes, including the key's UUID, in the IDT Device Header. The first time a device is guarded as IDT-Capable GuardPoint, VTE writes the IDT Header on the device before data transformation takes place, if data transformation is required.

Key Attributes - Example

The following figure illustrates 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.



The following figure illustrates the general properties of IDT_DEMO_KEY_1 after adding the key. 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

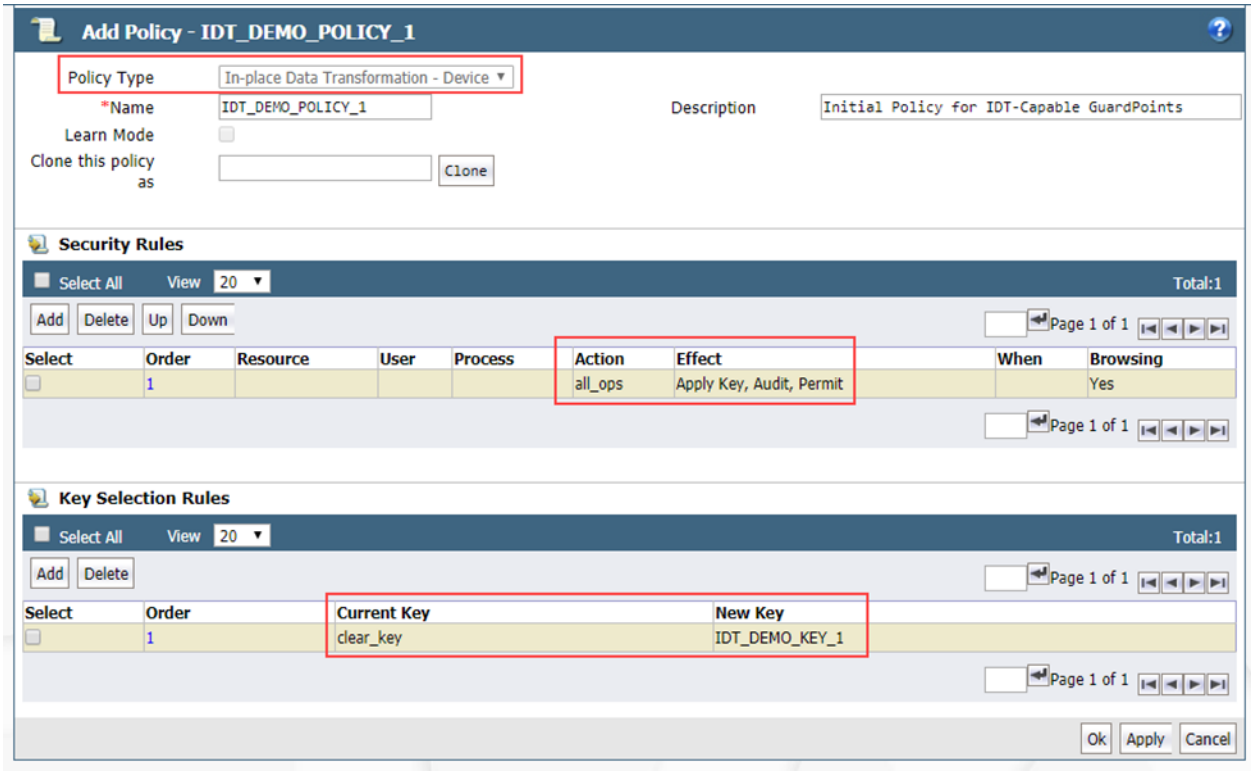
IDT-Capable GuardPoints require a policy of type **In-place Data Transformation - Device** with a single key rule specifying the key names for **Current Key** and **New Key**. The **Current Key** name is `clear_key` or an XTS-AES 256 key name, depending on whether the data on the device has already been 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, VTE 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 - Device policy, VTE 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 265.

The following figure shows an example of a simple policy named IDT_DEMO_POLICY_1 with policy type “In-place Data Transformation - Device”, a simple security rule that permits access to all users and programs, and that uses `clear_key` as the current key and IDT_DEMO_KEY_1 as the key to encrypt the data on any device associated with this policy.



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 246.
2. Install the VTE Agent and register the host with the DSM if it is not already registered. IDT does not require any special registration options or licenses. For details, see “Installing VTE for Linux” on page 41.
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 VTE Private Region. For details, see “Initializing an IDT-Capable Device” on page 251.
4. Log on to the DSM to apply an IDT-Capable GuardPoint to the device. For details, see “Guard the Linux Device with an IDT-Capable GuardPoint” on page 254.

WARNING

For devices with shared access across multiple VTE 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 VTE 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 VTE 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 VTE Private Region on the device itself or in the central VTE metadata directory on the host. (For details, see "The VTE Private Region and IDT Device Header" on page 246.)

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" on page 251
- "Initialize a Linux Device with Existing Data" on page 252

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 258.

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 VTE you want to use the centralized VTE metadata directory instead of embedding the VTE Private Region on the device itself. If you use this option, you must have configured and guarded the centralized VTE metadata directory as described in "VTE Private Region Location" on page 246.
   - `new` (required) indicates that the device contains no data (it is a new disk). As soon as you push the IDT policy from the DSM, the device will be available as a guarded IDT-Capable GuardPoint.
   - `-c <n>` (optional). If you use this option, VTE 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 VTE 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 VTE 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 VTE Private Region in the centralized VTE 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 DSM Administrator can protect the device as an IDT-Capable GuardPoint through the DSM Management Console. For details, see "Guard the Linux Device with an IDT-Capable GuardPoint" on page 254.

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 DSM 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 VTE. Unless you are using the centralized VTE metadata directory, this command examines the current disk size and computes the size required to hold the existing data plus the VTE Private Region at the beginning of the device. After the VTE 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 VTE Protected hosts in a cluster, be sure to initialize the device on one and only one of the VTE hosts.

The following procedure describes how to initialize the device for VTE. 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. VTE 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 VTE you want to use the centralized VTE metadata directory instead of embedding the VTE 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 VTE metadata directory as described in "VTE Private Region Location" on page 246.
   - `xform` (required) indicates that the device contains existing data. VTE 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 unaccessible 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, VTE 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 VTE 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 VTE 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 VTE 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 VTE 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 VTE 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 VTE metadata directory has been configured and guarded, the DSM Administrator can protect the device as an IDT-Capable GuardPoint through the DSM Management Console as described in “Guard the Linux Device with an IDT-Capable GuardPoint” on page 254.

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 DSM 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, see the chapter, “Managing GuardPoints”, in the *DSM Administration Guide*.

After the device has been initialized, you can guard the device as an IDT-Capable GuardPoint from the DSM Management Console. For existing devices, as soon as the GuardPoint configuration has been pushed to the host and the status changes to guarded, VTE 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 VTE Protected hosts in a cluster, be sure to guard the device on one and only one of the VTE hosts.

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 - Device** policy if needed. The policy you use for IDT must use an XTS-AES 256 key as the key rule. For details, see “IDT-Capable GuardPoint Encryption Keys” on page 248.
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 - Device policy you identified or created earlier in this procedure. VTE 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)**.

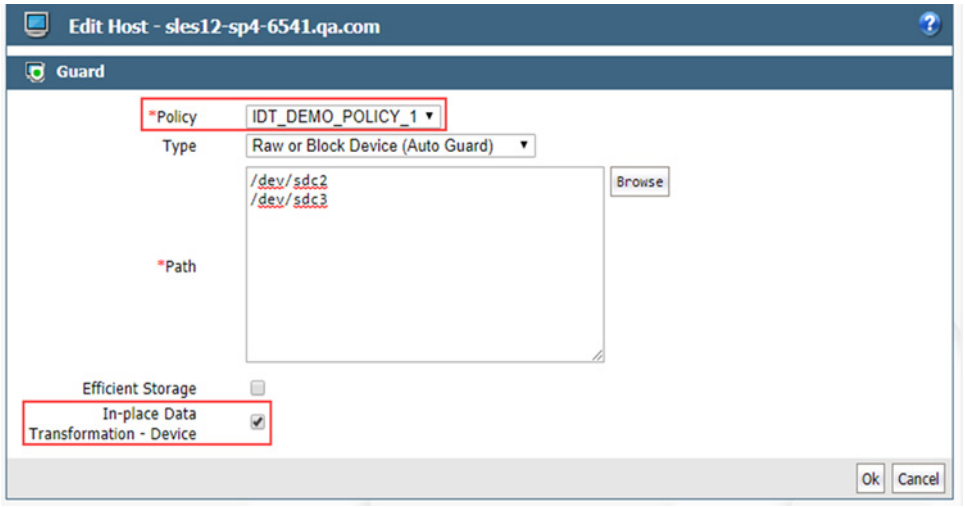
   If you select **Auto Guard**, VTE starts the guard process as soon as the policy is pushed to the host. You enable, disable, guard, and unguard the GuardPoint in the DSM. If you want to have the device automatically guarded and mounted at system start up, add the device to `/etc/fstab`. For details, see “Auto Mount Options for File System Devices on Linux” on page 263.

   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 VTE 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 251. They can be new devices, devices with existing data, or a mix of both.

9. Make sure the **In-place Data Transformation - Device** 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 VTE Agent on the host writes the IDT Device Header into the VTE 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, VTE 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 page 255.

    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 261.

After the device is initialized and guarded, the protected device must be accessed through the VTE 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.

Note: Be sure to use the secvm device name when using file system management tools such as `mkfs` and `fsck`.

Note: 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 embedded the VTE Private Region on the device, VTE first relocates existing data in the region of the device designated as VTE 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. VTE 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, VTE 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 DSM Administrator can guard the device with the desired In-place Data Transformation - Device policy. If the DSM Administrator chooses Auto Guard, data transformation begins as soon as the policy is pushed to the host. If the DSM 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 VTE 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.

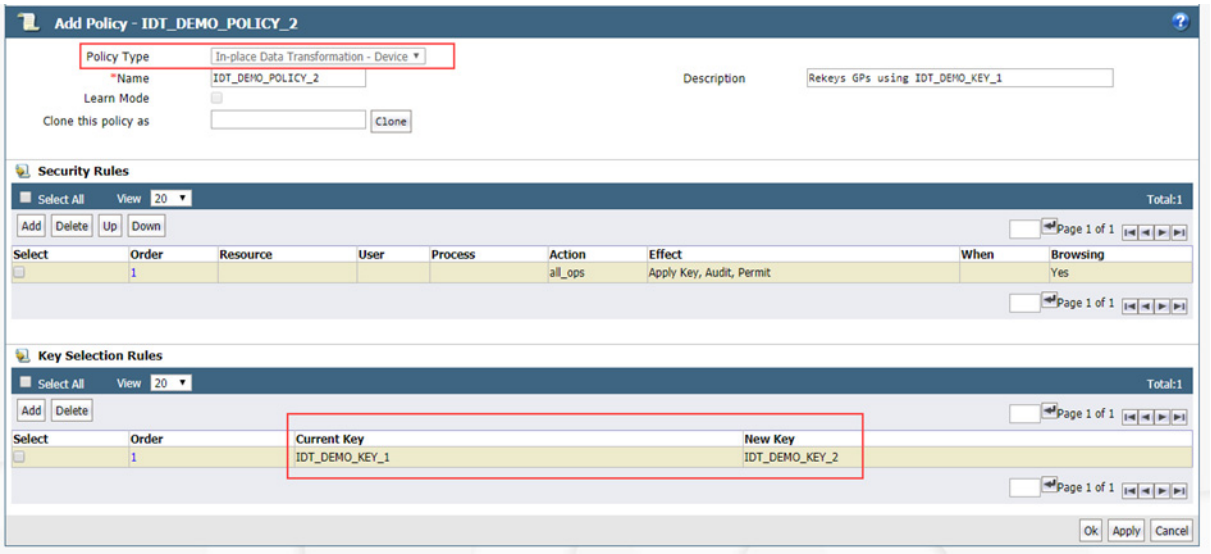
WARNING

For devices with shared access across multiple VTE 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 VTE 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 - Device** 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 - Device policy or you can change the keys assigned to an existing In-place Data Transformation - Device 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 VTE 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 VTE 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 Guard FS 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 - Device** 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 feature of VTE 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 - Device** 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. On the DSM, guard /`dev/mapper/mpathA` as Device GuardPoint using the policy created above. Be sure to check the **In-place Data Transformation - Device** 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, VTE shows the status of any data transformation processes happening on the device. If you do not specify this option, VTE 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

voradmin IDT Commands on Linux

The `voradmin` command is a command line utility for management of VTE 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. VTE 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
# unmount /xfs
# makedir /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 Auto Guard and Manual Guard. For more information about Auto and Manual Guard options, see "Guard the Linux Device with an IDT-Capable GuardPoint" on page 254.

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 `x-systemd.requires=secvm-barrier.service` and `nofail` settings in the `fstab` entries listing IDT-Capable GuardPoints, as shown in the following table.

Option	Description
`x-systemd.requires=secvm-barrier.service`	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 `secvm-barrier.service` service is a proxy for all the services that make up VTE.
`nofail`	The system boot will proceed without waiting for the IDT-Capable device if it can't be mounted successfully.

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,nofail 0 0
```

For information about configuring systemd for VTE, see "VTE and systemd" on page 315.

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		

EXT Utilities	Deny/ Allow	XFS	Deny/ Allow	Generic Utilities	Deny/ 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 VTE 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 on the DSM 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 VTE 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.

For an example of resizing a guarded Linux device, see “Example: Resizing a Linux Device” on page 224.

Use Cases Involving IDT-Capable GuardPoints

IDT-Capable GuardPoints support the same three use cases as Efficient Storage GuardPoints. For details, see “Use Cases involving Efficient Storage GuardPoints” on page 225.

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 251.

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 VTE 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 Vormetric 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 251.

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 VTE shutdown when there is a busy VTE 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 VTE 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 VTE startup when system fails to mount a file system. This error message is typically accompanied by a long timeout during the VTE 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 VTE 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 Vormetric 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: Please contact Vormetric 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: Please contact Vormetric 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 264.

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 22: VTE with Teradata Database Appliances

VTE 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 VTE's IDT-Capable GuardPoints can also be applied to protect Teradata Database Appliances. (For details about IDT, see Chapter 21: InPlace Data Transformation for Linux.)

This document contains the following sections:

IDT-Capable GuardPoints and Teradata Database Appliances

The Teradata Database Appliance must be protected by an IDT-Capable GuardPoint. The IDT (InPlace Data Transformation) 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 VTE and DSM support for IDT-Capable GuardPoints before you can configure VTE to work with Teradata. For details about IDT, see Chapter 21: InPlace Data Transformation for Linux.

Requirements and Considerations

Location of the VTE Private Region

The VTE Private Region contains the metadata VTE requires in order to support initial transformation of data on the device and subsequent data transformation to other encryption keys. By default, VTE creates the VTE Private Region at the beginning of the guarded device. If data already exists on the device, VTE requires that the device to be expanded by 63 MB to make room for the VTE 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 VTE metadata. This data relocation is totally transparent to applications and users.

With a Teradata Database Appliance, however, VTE cannot create the VTE Private Region at the beginning of the Teradata pdisk devices because the disks in the Appliance cannot be expanded. Therefore, for Teradata Databases, VTE stores the metadata in special directory called the VTE 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 VTE.

While this does not affect the functionality of VTE, 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 VTE 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. VTE 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 286.

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 VTE. Refer to the release notes document for VTE releases for compatibility requirements between the kernel releases from Teradata and VTE releases.

CAUTION

Be sure the version of the Teradata Database is fully compatible with VTE.

The Parallel Upgrade Tool (PUT) component of Teradata Database has been enhanced to discover VTE protected devices. The Parallel Upgrade Tool (PUT) component of Teradata (TDput) must be version `TDput-03.09.06.09` or higher. This capability in the PUT component must be at 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 capability in the PDE component must be at available in your Teradata Database.

The Teradata I/O Scheduler (tdsched) component must be version `01.04.02.02-1` or higher. This capability in the I/O Scheduler component must be at 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 VTE 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 VTE. For example:

```
# ./vee-fs-6.3.1-70-sles12-x86_64.bin -d /opt/teradata/vormetric
```

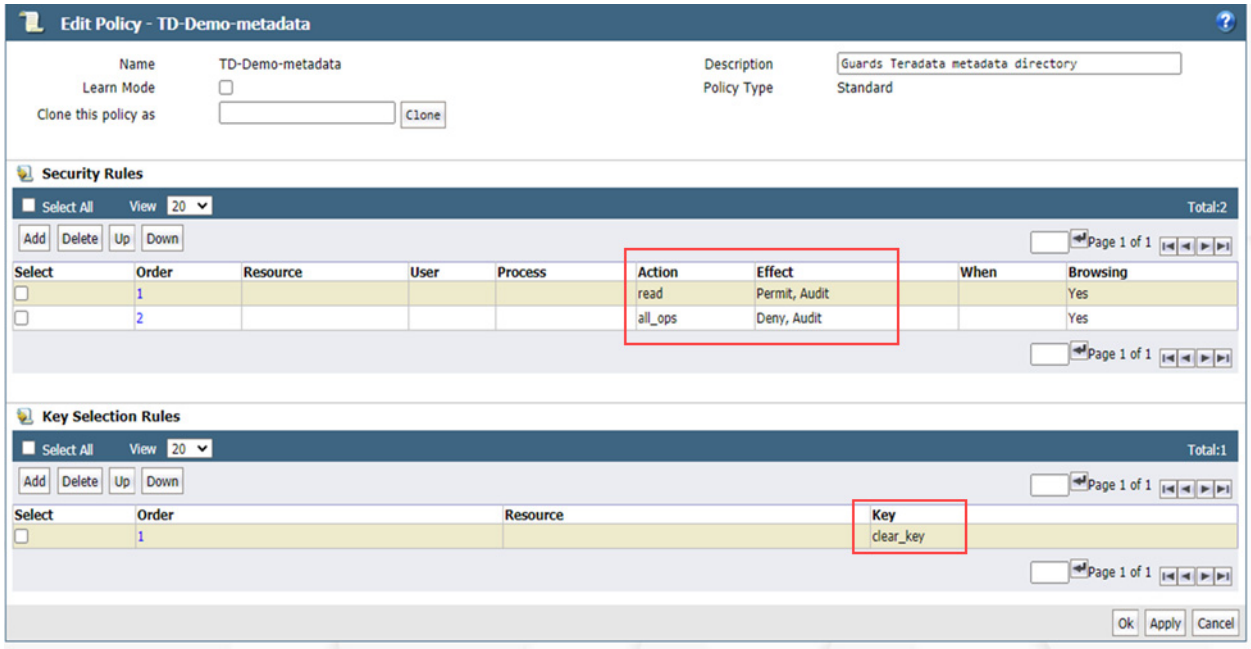
Note: The VTE Agent can be installed without stopping the Teradata Database service.

- The VTE metadata directory `/var/opt/teradata/vormetric/vte-metadata-dir` must be guarded by the DSM administrator to prevent accidental modification or deletion of the VTE metadata files. If the VTE metadata directory is not guarded, any attempt to configure or enable an IDT-Capable GuardPoint on the Teradata appliance will be rejected.

  The standard VTE 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 figure shows a sample policy for guarding the VTE metadata directory:



- The Teradata Database service must be stopped in order to upgrade an existing VTE Agent installation.

  To stop the Teradata Database service, use the `tpareset -x Stopping Database` command.
- The Teradata Database service must be stopped before VTE can rekey any guarded devices, and the service must remain stopped until VTE has finished rekeying *all* guarded devices on the appliance.

Guarding a Teradata Database Device

In order to guard a Teradata Database device, you need to:

1. Install VTE on every node in your Teradata Database Appliance. During this procedure you will register each node (also called the 'protected host') with the DSM. For details, see “Install VTE on the Teradata Database Appliance” on page 274.
2. Identify the devices that you want to guard as IDT-Capable GuardPoints. For details, see “Identify the Devices to Be Guarded” on page 274.
3. Select the initial configuration method that you want to use. For details, see “Select the Initial Configuration Method” on page 275.
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 275 or “Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method” on page 279.

Note: You must use the “In-place Data Transformation - Device” policy type when guarding those devices.

Install VTE on the Teradata Database Appliance

VTE must be installed on every node in the Teradata cluster, including any Hot Standby Nodes. When you install VTE on every node in your Teradata cluster and register each node with the DSM, keep the following things in mind:

- You must have AES-NI 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 VTE. Refer to VTE release notes for compatibility between VTE and Teradata kernel releases.
- Your version of Teradata Database must be fully compatible with VTE for supporting the pdisk devices protected as IDT-Capable GuardPoints.
- The VTE Agent must be installed in `/opt/teradata/vormetric`. In order to specify this location, use the `-d` option when installing VTE. For example:

```
# ./vee-fs-6.3.1-70-sles12-x86_64.bin -d /opt/teradata/vormetric
```

WARNING

Before you install or upgrade VTE, you must stop Teradata Database and any other applications that access the database devices directly or through the database service. Failure to stop the application will result in a failure to install or upgrade VTE.

CAUTION

Be sure to install the VTE 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 VTE Agent and registering it with the DSM, see Chapter 3: Installing VTE for Linux.

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
```

VTE 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`. 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 VTE 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 DSM 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" on page 275.
- **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 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 279.

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 VTE. 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 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 so that the devices you want to protect can be configured. You cannot initialize an online database device.
2. For each device, designate one of the nodes in the cluster as the node 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 and run the `voradmin idt config [-external] [new|xform] [-c <n>] <device-name>` command, where:
   - `[new|xform]` (required) indicates whether data already exists on the device. If the device contains no data (it is a new disk), 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`, VTE 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 data has been transformed.

- `-external` (required). You *must* use this option when initializing any Teradata device. With this option, VTE writes the VTE Private Region to a metadata file located in the VTE metadata directory. For details, see "Location of the VTE Private Region" on page 271.
- `-c <n>` (optional). If you use this option, VTE 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 VTE 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.

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 you want to initialize. If you want to distribute the initial data transformation or subsequent rekey load on all 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 an IDT-Capable GuardPoints and assign the In-place Data Transformation - Device policy (or policies) you want to use on those devices as described below.

Guard the Devices as IDT-Capable GuardPoints

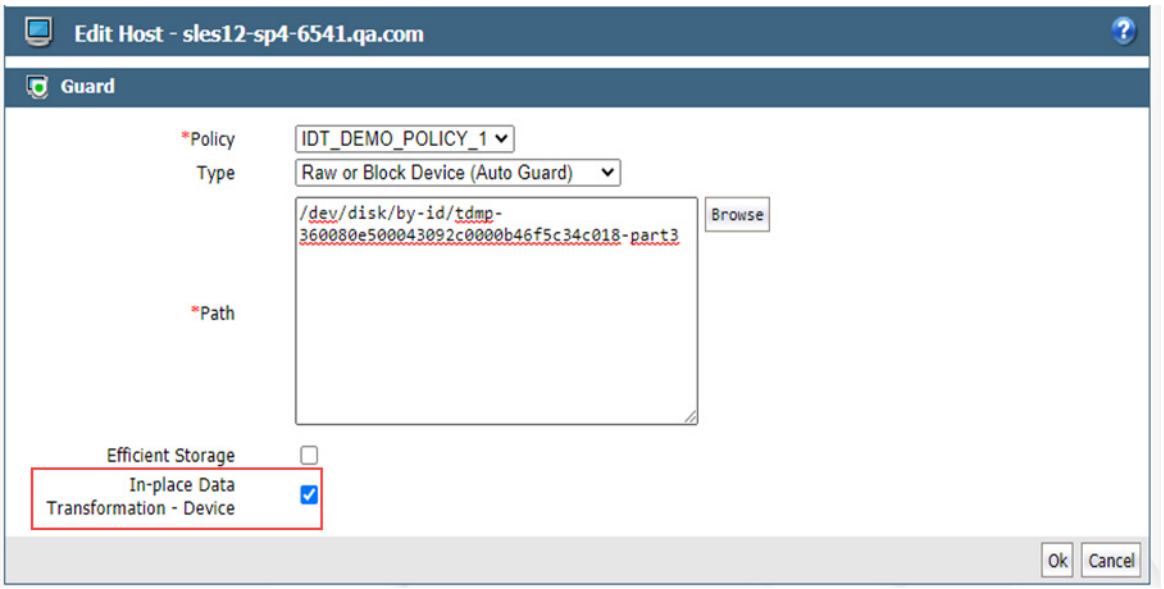
WARNING

For each device, 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 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 an administrator of type Security 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 - Device policy from the list of available policies. The policy must meet the requirements described in "Policy Requirements for IDT-Capable GuardPoints" on page 249.

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. An example of this is shown below.



8. Click **OK**.

   The DSM pushes the policy and the GuardPoint configuration to the node in the cluster and the VTE Agent on the node writes the IDT Device Header into the VTE Private Region in the local VTE 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, VTE 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 page 278.

   Devices with existing data are transformed from clear-text to cipher-text using the encryption key specified in the selected policy through the InPlace Data Transformation (IDT) process. For details on how VTE does data transformation on IDT-Capable GuardPoints, see Chapter 21: InPlace Data Transformation for Linux.

WARNING

The initial data transformation of each device must be executed on *one and only one* of the nodes in the cluster. DO NOT initialize or guard the device on multiple nodes in the cluster before the initial data transformation has completed, because doing so initiates the data transformation process on the device on multiple nodes. The simultaneous attempts to transform the data for the first time can cause data corruption of all data on the entire device.

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.
10. 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 VTE `secvm` 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 secvm 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
```

11. 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:

```
/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, VTE shows the status of any data transformation processes happening on the device. If you do not specify this option, VTE 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 timeframe.

To address this issue on an existing Teradata database, you can configure VTE on Teradata Appliances using the Backup and Restore method to retain existing data. The total time required to configure your Teradata Appliances with VTE 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 VTE 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 VTE protected devices. VTE automatically encrypts the data as it is restored to each protected device.

Prerequisites

Before starting this process, the following prerequisites must be completed:

- Make sure you have met the requirements described in “Requirements and Considerations” on page 271.
- Create or identify the XTS-AES 256 key that you will use for the In-place Data Transformation - Device policy you will apply to the IDT-Capable GuardPoints.
- Create a **Standard** policy that will be used to guard the VTE metadata directory (`/var/opt/teradata/vormetric/vte-metadata-dir`).
- Create an **In-place Data Transformation - Device** 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 274.

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 requirements on a full backup.
2. After confirming that your backup of the Teradata database completed and successful, shutdown the Teradata database.

```
# tpareset -x -f "shutting down for VTE installation"
```

3. Install VTE in `/opt/teradata/vormetric` on all nodes in the Teradata cluster and register them to a DSM server as described in “Install VTE on the Teradata Database Appliance” on page 274.

```
# ./vee-fs-6.3.1-70-sles12-x86_64.bin -d /opt/teradata/vormetric
```

4. Create the VTE 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, do the following. If you are using the Teradata Inteliflex model, proceed to Step 6.
   a. Log on to the DSM, click the **Hosts** tab, and set the GuardPoint to the VTE 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 274.
   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 VTE to guard the device without transforming data.

   d. Log on the DSM, click the **Hosts** 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 - Device policy that you created.

      Repeat this step for all other hosts until all initialized devices have been guarded with an In-place Data Transformation - Device policy.
6. If you are using the Teradata Inteliflex model, do the following. 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 DSM, click the **Host 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 Groups** tab, click the name of one of the host groups you just created in the Host Group **Name** column. Click on the **Members** tab and add the designated node from the clique as a member of this host group.

      For example, if you clicked on the host group clust-clique-1, you would add the designated node from clique 1 to this host group.

      Repeat this step until all designated nodes have been added to the Host Group that matches their clique. In the previous example, you would have one designated node in each of the 4 host groups that you created.
   d. On the **Host Groups** tab, click the name of one of the host groups you just created in the Host Group **Name** column. Click on the **GuardPoints** tab and add a GuardPoint for the VTE 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 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 274.
   f. Configure each disk device 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 tells VTE 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.

   Repeat this configuration process until all devices on all designated nodes that you want to guard have been configured as new IDT-Capable devices.

h. On the **Host Groups** tab, click the name of one of the host groups you just created in the Host Group **Name** column. 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 - Device policy.

   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 host 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 286. 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.

i. After metadata replication is completed on all cliques, log on the DSM and click the **Hosts Group** tab, then click the name of one of the host groups you just created in the Host Group **Name** column. 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.

   Repeat this step for each one of the Host Groups that you created for your cliques.

7. Verify that all nodes in a clique shows all the devices as guarded.

```
# pcl -shell "secfsd -status guard"
```

   If your Teradata database has more than one clique, repeat this step for each clique.

8. Work with a 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 VTE has 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 258.
- Use the Backup/Restore method as described in “Initialize and Guard the Teradata Database Devices Using the Backup/Restore Method” on page 279. 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 VTE Metadata Directory.
  - You need to create a new InPlace 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 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 you use the new policy so that VTE knows it needs to re-encrypt the data with the new key after you restore the devices from the backup.

Note: Make sure you stop the Teradata database before you rekey the devices.

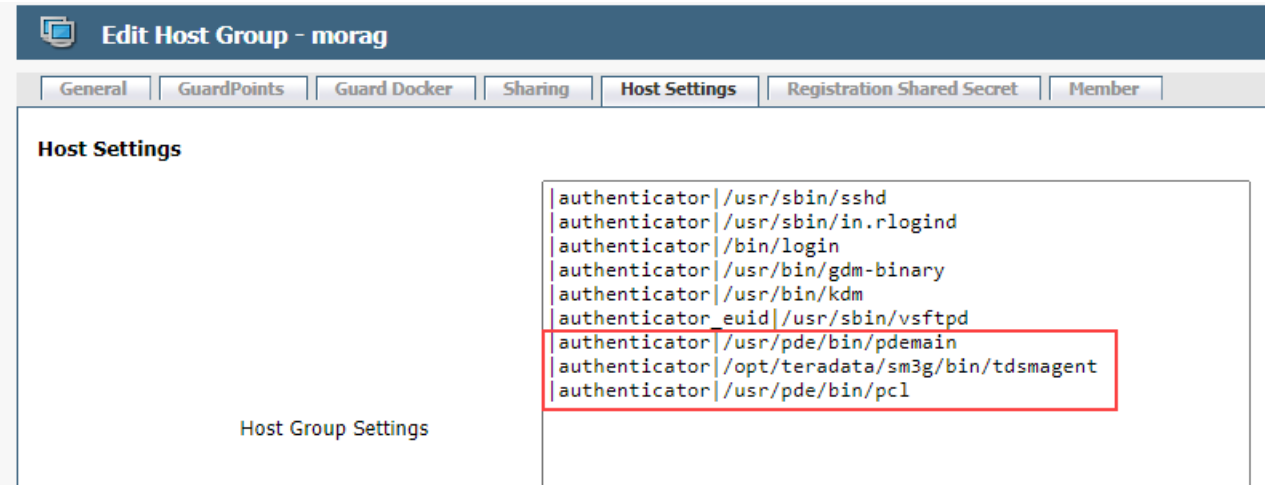
WARNING

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 data on the entire device.

Access Rules to Apply on the Teradata Database Appliance

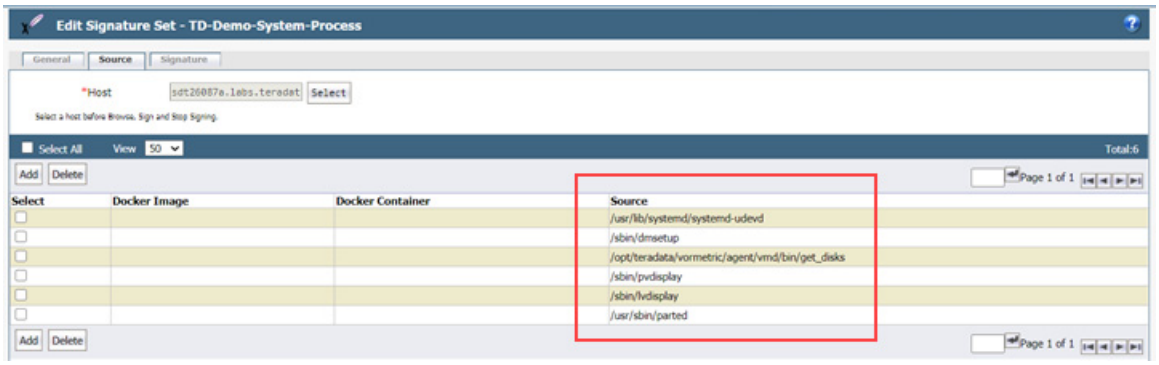
This section provides the instructions for creating a sample policy and signature sets specific to a Teradata Database Appliance to deny unprivileged users access to clear-text data on guarded devices. Such a policy requires the inclusion of specific security rules to restrict access to the Teradata Device GuardPoints for specific set of users, groups, and/or processes.

1. In the DSM Management Console, edit the host entry for the Teradata Database Appliance.
2. Go to the **Host Settings** tab and add the following entries to the list of binaries:
   - `|authenticator|/usr/pde/bin/pdemain`
   - `|authenticator|/opt/teradata/sm3g/bin/tdsmagent`
   - `|authenticator|/usr/pde/bin/pcl`



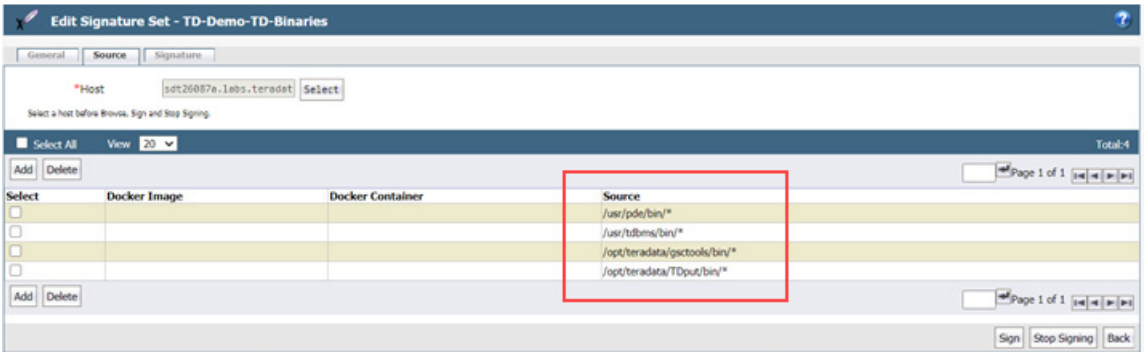
3. Create a signature set for system processes that require access to guarded devices. The following system processes on the Teradata Database Appliance *must* be permitted access:
   - `/usr/lib/systemd/systemd-udevd`
   - `/sbin/dmsetup`
   - `/opt/teradata/vormetric/agent/vmd/bin/get_disks`
   - `/sbin/pvdisplay`
   - `/sbin/lvdisplay`
   - `/usr/sbin/parted`

   You can include additional processes as needed. In the following example, this signature set is called TD-Demo-System-Process.

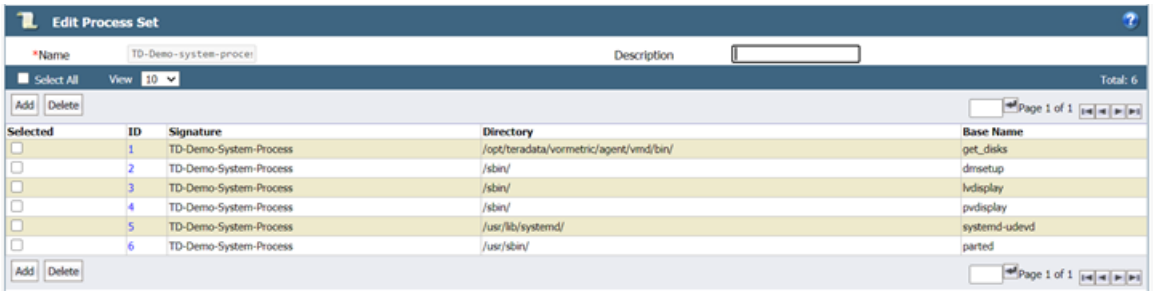


4. Create a signature set for Teradata Database processes that require access to guarded devices. The following directories contain the Teradata binaries that require access:
   - `/usr/pde/bin/*`
   - `/usr/tdbms/bin/*`
   - `/opt/teradata/gsctools/bin/*`
   - `/opt/teradata/TDput/bin/*`

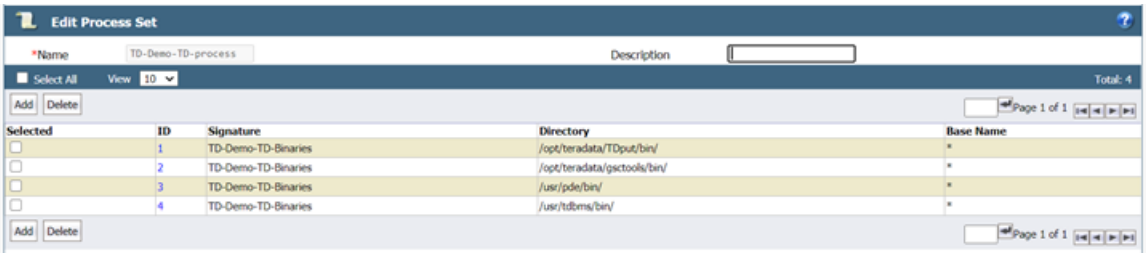
You can include additional processes as needed. In the following example, this signature set is called TD-Demo-TD-Binaries.



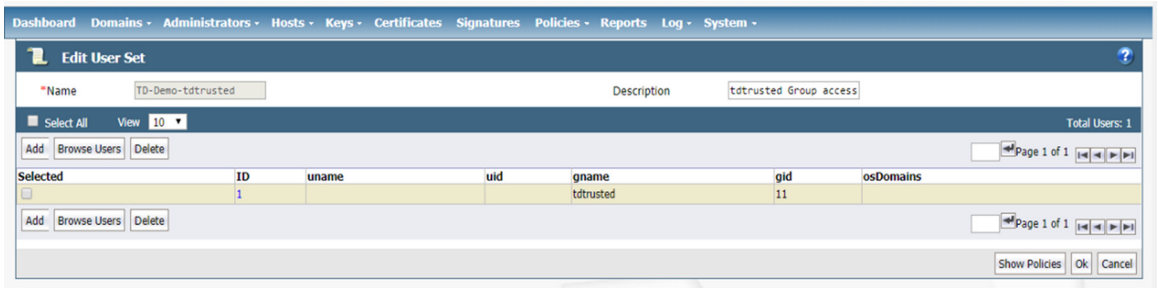
5. Create a system-level process set to associate the system-level processes listed in step 3 with the signature set you created in step 3. In the following example, this process set is called TD-Demo-system-process.



6. Create a Teradata Database process set to associate the signature set for the Teradata Database binaries created in step 4. In the following example, this process set is called TD-Demo-TD-process.

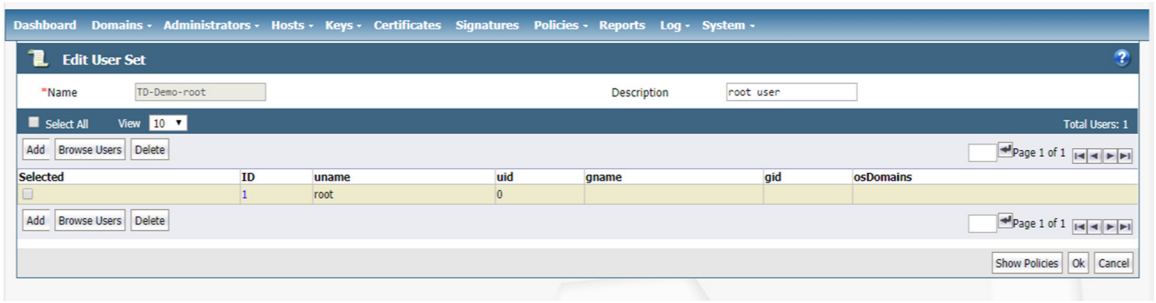


7. Create the user set for the Teradata trusted group on your appliance. In the example below, the trusted group name on the Teradata Appliance is `tdtrusted`.

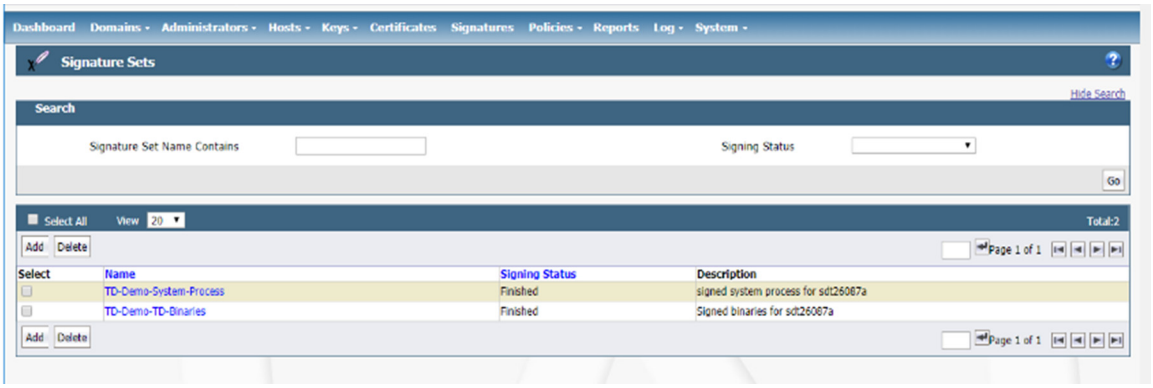


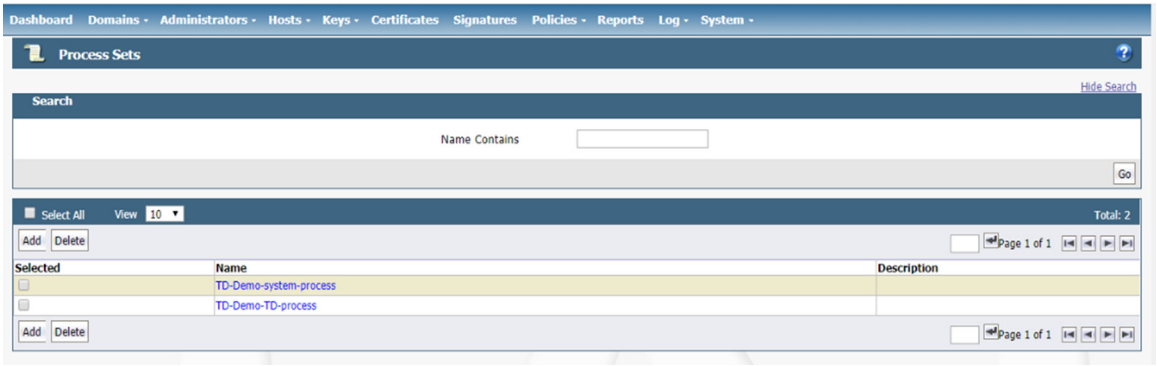
8. If you have any other trusted groups you want to include in your security rule, add a user set for each one of those trusted groups. You can add as many user sets to the policy as you need.

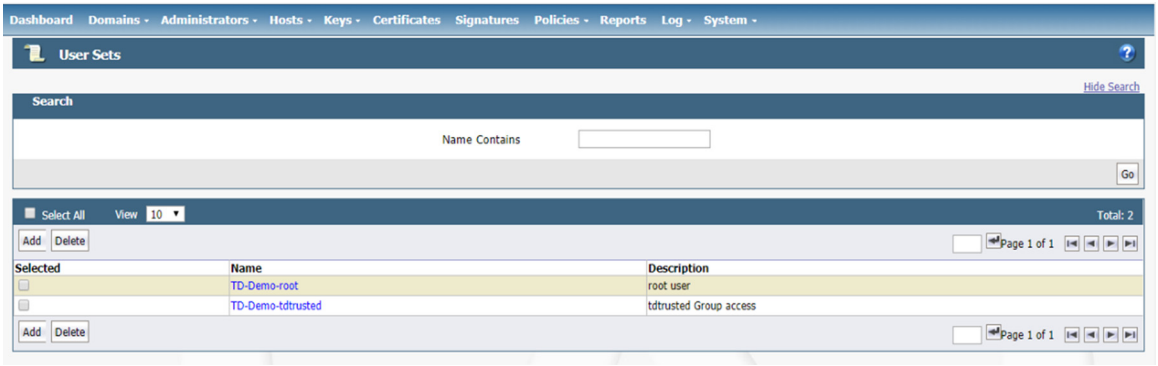
9. Create a user set for the root user and any other privileged users you want to add. In the example below, only the root user is trusted to access the guarded Teradata devices.



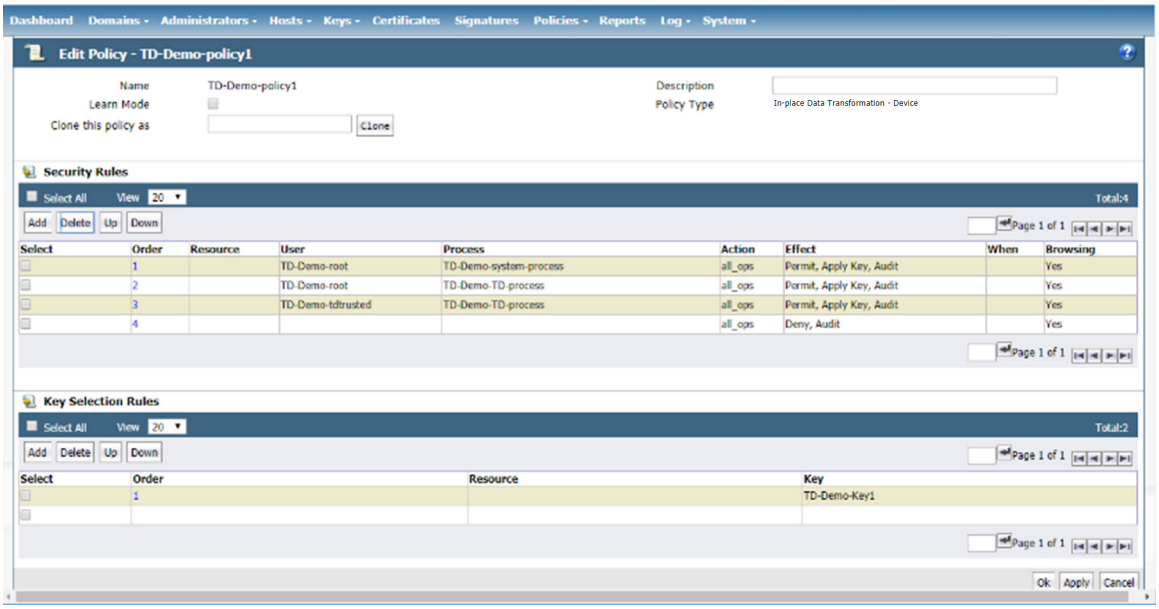
10. Verify that you have the complete signature sets, process sets, and user sets as shown in the examples below.







11. Apply the policy to the Teradata Device GuardPoints. For example:



Replication of IDT Metadata Files Across Members of a Clique

For Teradata, metadata for IDT-Capable GuardPoints is stored in external files in the VTE metadata directory. Because the Teradata appliance is a cluster of multiple hosts (or nodes) 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 VTE Agent on that device automatically replicates the metadata file for that device to the other nodes in the clique using the Teradata **`pcl`** command. VTE 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 VTE Agent. This replacement is achieved by sending the updated metadata file to all 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 required to recover from the reported issues. For troubleshooting and recovery steps, see “Alerts and Errors” on page 291.

General PCL Error

If the `pcl` command fails during the replication process, a message indicating the error will be logged on the DSM. The messages on DSM 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 and so 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, VTE will log a message on the DSM 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 host 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 intialized and guarded on one and only one of the nodes in the cluster. That means the IDT-Capable GuardPoint cannot be part of a Host Group when an IDT-Capable GuardPoint is created or rekeyed because membership in a Host Group means that any data transformation on any member of the Host Group is initiated for that member on all nodes in the Host Group 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 host with the Host Group.

To configure a new device whose host is part of a host group:

1. In the DSM, make sure that there is no GuardPoint for the device at the Host Group level.
2. Designate one of the nodes in the cluster as the node 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 275.
4. In the DSM, guard the device on the designated node using an In-place Data Transformation - Device policy. For details, see “Guard the Devices as IDT-Capable GuardPoints” on page 276.
5. Wait for the data transformation to complete on the host and for VTE to replicate the metadata to the other members of the clique.

   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.
6. In the DSM, remove the IDT-Capable GuardPoint you added in step 4.

7. Guard the device through the Host Group to make sure that all nodes in the cluster recognize it as guarded.

To rekey a new device whose host is part of a host group:

1. In the DSM, unguard the IDT-Capable GuardPoint through the Host 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 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. In the DSM, guard the device on the designated node with the In-place Data Transformation - Device 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. In the DSM, remove the IDT-Capable GuardPoint you created in step 6.
8. Guard the device through the Host Group to make sure that all nodes in the cluster recognize it as guarded.

Best Practices

Using a of Host Group to Guard Metadata Directories

Thales recommends that customers use a host group exclusively for guarding and management of the VTE external metadata directories across all nodes in Teradata cluster. (For details about the VTE metadata directory, see “Location of the VTE Private Region” on page 271.)

By using a Host Group, the same DSM policy can be applied to the external metadata directory (`/var/opt/teradata/vormetric/vte-metadata-dir`) on all of the VTE protected nodes in the cluster.

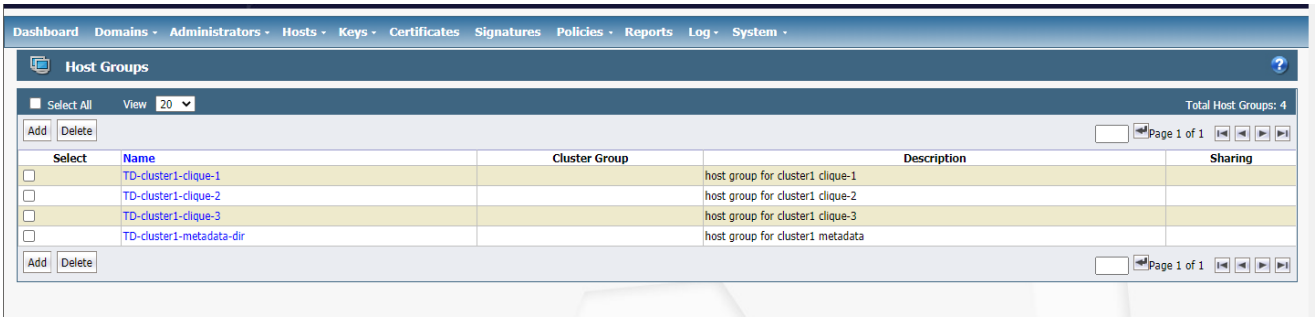
Using a Host Group for Guarding Teradata Devices in a Clique

Thales recommend that customers use a Host Group on the DSM to guard and manage the devices shared by all VTE protected nodes in a clique. Each clique in the cluster should be associated with a separate Host Group. When configuring the Host Group, the members for that host 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 Groups, one for each clique. In each Host 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 host 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:



Best Practice for Preparation for Initial Data Transformation or Rekey

With large amount 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 entire 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 Group associated with the clique and re-enabling each IDT-Capable GuardPoint only 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 Group level. Enabling the guarded device at Host Group level enables the guarded devices on all the nodes within the clique to share access to the device.

After transforming data and guarding each device on all the nodes, you can then restart the database.

As described in “Interoperability with Host Groups” on page 287, 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 VTE from the Teradata Cluster

The following procedure describes how to remove VTE from your Teradata cluster. Note that uninstalling VTE will unprotect the pdisk devices that VTE is protecting/encrypting. Because the data on the pdisks remain encrypted after uninstalling VTE, 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 VTE. 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 VTE operations for protecting Teradata clusters and IDT-Capable GuardPoints.

To uninstall VTE:

1. Make sure you have access to a full database backup so that you can restore your Teradata database after uninstalling VTE.
2. Shut down your Teradata database.
3. Use the `secfsd -status guard` command to view the full list of device GuardPoints guarded by VTE 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 DSM:
   a. Click on the host group name associated with your Teradata cluster and then click the **Guard FS** tab.
   b. Select the device names listed in Guard FS table. The devices listed in Guard FS table are the same devices listed in the consolidated list of devices compiled in step 3.
   c. Click the **Disable** button to 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 the **Unguard** button 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 Guard FS 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 Guard FS 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 VTE service on each node

```
# /etc/vormetric/secfs stop
```

   b. Run the following command to uninstall the VTE Agent on each node:

```
# /opt/teradata/vormetric/agent/vmd/bin/uninstall
```

To restore to the full backup of the database to your cluster, contact your Teradata Customer Support Representative (CSR) to perform System Initializer and do a full restore of the database from your backup.

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 VTE system operations for Teradata clusters and will always be paired with an operation error which indicates which VTE 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>

VTE 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 DSM for remote nodes or VTE failed to ascertain the GuardPoint status for remote nodes.

Solution: Verify that the GuardPoint for the device has been removed from the DSM 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 VTE 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

VTE 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

VTE 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 correct 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

VTE was unable to complete the rekey operation on remotes 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 23: VTE for Windows Utilities

This chapter describes utilities you can run on Windows. For information on Linux utilities, see Chapter "VTE for Linux Utilities" on page 299.

Vormetric provides a variety of utilities that administrators can use to help manage VTE. This chapter describes the following utilities:

voradmin secfs Commands

The `voradmin secfs list` and `voradmin secfs status` commands display GuardPoint and policy information on the host.

voradmin secfs List Commands

The `voradmin secfs list` command has the following options:

Table 23-1: `voradmin secfs list` Options

`guardpoints`	Displays all the GuardPoints on the host.
`policy`	Displays all the policies used on the host.
`logger`	Displays the logging details on the host.
`status`	Displays the authentication settings on the host.

For example, to view all the GuardPoints on the host, you would enter:

```
C:\>voradmin secfs list guardpoints
Guard Point:     1
Policy ID:          16553
Policy name:     ES-Standard-Policy
Directory:       esg-disk1-demo
Type:            rawdevice
Status:          guarded

Guard Point:     2
Policy ID:          18857
Policy name:     Accounting-IT-Access-Policy
Directory:       G:\Data
Type:            local
Status:          guarded

Guard Point:     3
Policy ID:          18985
Policy name:     LDT-Policy
Directory:       C:\LDT-Folder
```

```
Type:            local
Status:          guarded
```

To view just the policies in use on the host, you would enter:

```
C:\>voradmin secfs list policy
Policy:          1
Policy name:     LDT-Policy
Type:            LDT

Policy:          2
Policy name:     ES-Standard-Policy
Type:            ONLINE

Policy:          3
Policy name:     Accounting-IT-Access-Policy
Type:            ONLINE
```

voradmin secfs status Commands

The `voradmin secfs status` command has the following options:

Table 23-2: `voradmin secfs status` Options

`keys`	Displays the current status of the keys on the host.
`lock`	Displays the status of any system or agent locks on the host.

For example:

```
C:\>voradmin secfs status keys
Encryption keys are available

C:\>voradmin secfs status lock
FS Agent Lock: Disabled
System Lock: Disabled
```

vmsec Utility

The `vmsec` utility allows you to manage the security aspect of the VTE agents on the host. The Windows `vmsec` utility is located in `C:\Program Files\Vormetric\DataSecurityExpert\agent\vmd\bin\vmsec.exe`

Syntax

Table 23-3: `vmsec` Syntax [options]

`check_install`	Verifies that the kernel component is running. This command checks VTE services and reports if any of the services are not running.
`challenge`	Initiates challenge-response on the host. This command displays a File System Agent password challenge string and enter the response string when the DSM is not network accessible.
`status`	Displays kernel configuration.

`vmdconfig`	Displays the vmd configuration.
`check_hwenc`	Determines whether this system supports hardware crypto.
`hwok`	Reports status of hardware signature.
`passwd [-p passwd]`	Enters the host password when the DSM is not network accessible. User can unlock the GuardPoints with this password.
`version`	Displays VTE version.

agenthealth Utility

The `agenthealth.ps` utility validates:

- Super-user privilege
- VTE Agent installation
- VTE registration to DSM Server
- VTE processes/modules that are running
- Available disk resources
- Current GuardPoints

  Tests if the agent can reach the GuardPoints
- VTE log directory resource status

  This directory contains pending VTE 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

The Agent health check script (`agenthealth.ps1)` is located in `C:\Program Files\Vormetric\DataSecurityExpert\agent\shared\bin\`

To run the `Agenthealth` check script:

1. Run the power shell command to enable self-signing for the system.

   Before running agent-health script make sure power shell command has enough privileges to execute the Powershell script. Some Windows operating systems have default execution policy set as `restricted`.

```
Use the Powershell command "Set- Execution-policy Remote Signed" to change the
execution policy if needed.
```

2. Open the `Powershell` prompt as administrator.
3. Type:

```
.\agenthealth.ps1
```

Example

```
.\agenthealth.ps1
```

System Response:

```
Log file is at
C:\ProgramData\Vormetric\DataSecurityExpert\agent\log\agent_health.log
```

```
Checking super user privilege.............................. OK
Vormetric Agent installation............................... OK
Vormetric policy directory................................. OK
Registration to server..................................... OK
Kernel drivers are loaded.................................. OK
VMD is running............................................. OK
SECFSD is running.......................................... OK
rhat26130.qa1.com is resolvable............................ OK
rhat26130.qa1.com port 8446 is reachable................... OK
rhat26130.qa1.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 server2016-12-01         14:39:49.038
Checking available disk space.............................. OK
Checking logging space ........................... OK
        Log directory is "/var/log/vormetric"
        File system for log data is "/", 32G free (17% full)
        Log directory contains 2 of maximum 200 files (1% full)
        Log directory contains 1 of maximum 100 Mbytes used (1% full)
Testing access to C:\GP2................................... OK
```

agentinfo Utility

The `agentinfo` utility collects system logs, VTE agent logs, VTE agent trace information, and system information for diagnostic purposes. All this information is saved in the destination path and compressed into a zip file. The `agentinfo` utility is available as an `agentinfo.js` Java command and as an `agentinfo.ps1` PowerShell command.

agentinfo Utility (Java version)

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

The `agentinfo.js` support collection scripts reside in the following path on systems where the VTE agent is installed.

To run the `agentinfo` script on Windows, navigate to one of the following folders:

`C:\program files\vormetric\DataSecurityExpert\agent\vmd\bin`

`or`

`C:\program files\vormetric\DataSecurityExpert\agent\shared\bin`

then run the following script:

`agentinfo.js`

agentinfo Utility (PowerShell version)

The PowerShell version of `agentinfo` supports several parameters.

PowerShell version agentinfo parameters

`Directory` - Specify the directory where all the collection information is saved. By default, this information is saved in the current directory.

`ZipFile` - Specify the name of the compressed file, where all the collected information will be archived. By default, this information is saved in the current directory.

`Logfile` - Specify the name of the files where verbose logs will be saved. By default, this information is saved in the current directory.

Examples for using agentinfo utility (PowerShell version)

To save all the collection information in "`c:\AgentLogs`" folder, run the following command:

```
.\agentinfo.ps1 -Directory 'C:\AgentLogs'
```

To save all the collected information in "`c:\AgentLogs`" folder and verbose logs in "`c:\temp\AgentInfo.log`", run the following command:

```
.\agentinfo.ps1 -Directory 'C:\AgentLogs' -LogFile 'c:\temp\AgentInfo.log'
```

To save all the collected information in "`c:\AgentLogs`" folder, verbose logs in "`c:\temp\AgentInfo.log`", and create the "AgentInfo.zip" archive file, run the following command:

```
.\agentinfo.ps1 -Directory 'c:\AgentLogs' -LogFile 'c:\temp\AgentInfo.log' -
ZipFile 'C:\temp\AgentInfo.zip'
```

Note: PowerShell 5.1 or later is required. Use the `$PSVersionTable.PSVersion` command to confirm which PowerShell version you are using.

Chapter 24: VTE for Linux Utilities

This chapter describes VTE for Linux utilities. The Windows utilities are described in "VTE for Windows Utilities" on page 293.

Thales provides a variety of utilities that an administrator can use to help manage VTE. These utilities reside in storage until summoned by the administrator.

The following utilities are described in this chapter:

secfsd Utility

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

- GuardPoints defined in the *Guard FS* 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 **Guard FS** 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, VTE 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

Table 24-1: `secfsd` Syntax

Command	Description
`-help`	display `secfsd` options
Status Options	
`-status guard [-v | -tree]`	list all GuardPoints
`-status keys`	show current encryption key state

Command	Description
`-status auth`	list authentication settings
`-status lockstat`	show VTE lock status
`-status logger`	list logging details
`-status policy`	list configured policies
`-status pslist`	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 `secfs2`

secfsd Examples

Display GuardPoint information

To display the GuardPoint paths, applied policies, policy type, and guard status, type:

```
# secfsd -status guard
```

System Response

```
# 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 **Guard FS** tab.
ConfigState	Guard status of the GuardPoint, as recognized by the DSM. It can be guarded or unguarded.
Status	Current guard status, as recognized by VTE. State can vary.

Note: 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.

Display GuardPoint information in a different format

To display the same information in a different format, include the `-v` argument, type:

```
# secfsd -status guard -v
```

System Response:

```
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
GuardPoint: 3
        Policy:          allowAllOps-winusers1_fs
        Directory:       /opt/apps/apps1/doc
        Type:            local
        ConfigState:     guarded
        Status:          guarded
        Reason:          N/A
```

Display GuardPoints in a tree view

Use the `secfsd status guard -tree` to list GuardPoints in a tree view, type:

```
# secfsd -status guard -tree
```

Display host settings

Use the `auth` argument to display the SHA2 hash signature for each protected host setting, type:

```
# secfsd -status auth
```

System Response:

```
/bin/su 3E765375897E04C39AB17D4C755F50A35195535B6747DBA28DF9BD4AA672DFF9
|authenticator|/usr/sbin/sshd
98FC599D459EDEA52A60AB394B394803B5DAB96B53148DC608732DDA6777FA1A
/usr/sbin/in.rlogind
5C9A0EDD8BF54AE513F039476D21B3032507CF957AA0CB28C368EB8AB6E684FB
/bin/login 0D2EE0B995A30AE382B4B1CA5104715FC8902F457D283BDABAAD857B09259956
/usr/bin/gdm-binary
363780522E3CCF9ABF559F059E437743F9F97BBBB0EE85769007A464AD696BD1
/usr/bin/kdm BAD41BBCDD2787C7A33B5144F12ACF7ABC8AAA15DA9FDC09ECF9353BFCE614B5
```

To display the status of VTE keys, type:

```
# secfsd -status keys
```

System Response:

```
Encryption keys are available
```

Display Lock Status

To display the status of VTE locks, type:

```
# secfsd -status lockstat
```

System Response:

```
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 VTE software, or change operating system files, the status of FS Agent Lock and System Lock must be false.

Display VTE Log Status

To display the status of VTE log service, type:

```
# secfsd -status logger
```

System Response:

```
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 VTE certificate. VTE uses the certificate to authenticate VTE when it uploads the log data to the DSM.

Display Applied Policies

To display the policies that are applied to VTE, type:

```
# secfsd -status policy
```

System Response:

```
Policy: allowAllOps_fs
Type: regular
Policy: allowAllRootUsers_fs
Type: regular
Policy: allowAllOps-winusers1_fs
Type: regular
```

Display VTE processes

To display VTE processes, type:

```
# secfsd -status pslist
```

System Response:

```
Protected pid list:     739   731
```

Display Detail about VTE processes

The example displays the process PID numbers for the `vmd` and `secfsd` processes. The `ps` commands show the processes for those PIDs.

```
# ps -fp <process #>
```

Example:

```
# ps -fp 739
```

System Response:

```
     UID      PID     PPID   C    STIME    TTY  TIME CMD
    root  7012404        1   0 11:04:56      -  0:00 /opt/vormetric/
DataSecurityExpert/agent/vmd/bin/vmd
```

Display VTE Version Information

To display VTE version information, type:

```
# secfsd -version
```

System Response:

```
version: vee-fs-6.3.1-45-sels12-x86_64.bin
```

Manually Enable a GuardPoint

To manually enable a GuardPoint on a Linux host:

1. Click **Hosts > Hosts >** <*hostName*> **Guard FS**.
2. Click **Guard**.
3. In the Policy field, select a policy.
4. Set Type to **Directory (Manual Guard)**.
5. Click **Browse** and enter the GuardPoint path.
6. Click **OK**.
7. Log onto the system hosting VTE as the root user.
8. Verify the change, type:

```
# secfsd -status guard
```

System Response:

```
GuardPoint    Policy         Type   ConfigState Status     Reason
----------    ------         ----   ----------- ------     ------
/opt/apps/etc allowAllOps_fs manual unguarded  not guarded Inactive
```

Verifying a GuardPath

Verify that a GuardPath is guarded, type:

```
# secfsd -guard <path>
```

For example:

```
# secfsd -guard /opt/apps/etc
```

System Response:

```
secfsd: Path is Guarded
```

secfsd and Raw Devices

VTE for Linux only creates block devices.

To display them, type:

```
# ls -l /dev/secvm/dev
```

System Response:

```
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 VTE on the host. On Linux-hosts, the `vmsec` utility is located in:

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

vmsec syntax

Table 24-2: vmsec Syntax [options]

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

vmsec Examples

Display VTE Challenge String

To display a VTE password challenge string and enter the response string when the DSM is not network accessible, type:

```
# vmsec challenge
```

System Response:

```
Contact the help desk at 1-800-555-1212 for response generation.
Your host name is "Host120" Your challenge is: HPTQ-ZYLK
Response -> IHFY-W7WG-PDAO-QKKQ
```

The contact information is configured in the DSM Management Console (Domains > Manage Domains) *Add Domain* window. Contact the DSM administrator and give them the challenge string. The DSM 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.

The Windows equivalent of this command is right-click the tray icon and select **Challenge...-> Response**.... The *VTE Challenge/Response* window opens. Note the Challenge string displayed. If no string is displayed, the host password is static. If a challenge string displays, contact a DSM administrator for the response string.

Display VTE Status

This utility shows you if VTE is configured and running. If it is not running, you might need to start it manually.

To display VTE status, type:

```
# vmsec checkinstall
```

System Response:

```
The kernel component is installed and running.
```

Entering a Password

To enter VTE static host password, type:

```
# vmsec passwd
```

System Response:

```
Please enter password:
OK passwd
```

To enter VTE static host password on the command line so you can specify it in a batch script, type:

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

System Response:

```
OK passwd
```

Display Kernel Status

To display the kernel status, type:

```
# vmsec status
```

System Response:

```
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=6.1.0.73
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 VTE Build Information

For Linux, type:

```
# vmsec version
version 5, Service Pack 2
2019-08-17 20:41:51 ()
Copyright (c) 2009-2019, Vormetric. All rights reserved.
```

System Response:

```
Version 6
6.3.1
2020-03-17 15:15:23 (PDT)
Copyright (c) 2009-2020, Thales. All rights reserved.
```

Display Contents of Conf files

To display the contents of the `agent.conf` and `.agent.conf.defaults` files, type:

```
# vmsec vmdconfig
```

System Response:

```
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=6.3.1
VMSDK_AGENT_BUILD_ID=28
PREV_URLS=https://srv.my.vormetric.com:8443
syslog_appender_myhost name=dev.my.vormetric.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.vormetric.com:8443
STRONG_ENTROPY=false
```

Binary Resigning

Any executable that is part of either a host setting or Signature set, and resides in a GuardPoint that uses an LDT policy, will use different signatures for an LDT key rotation. The result is that the host settings binaries will no longer be authenticated, or the Signature Set policy rules will no longer trigger for those binaries. To prevent these issues, the security administrator must manually resign each affected binary after each key rotation.

VTE for Linux version 6.1.2 introduced binaries that are signed with a signature that does not change with a key rotation. The security administrator must do only one manual resigning after a key rotation. After that, there is no longer a need to resign after each key rotation.

If upgrading or installing a new machine using the same signature sets that you used previously, do the following:

1. Install the latest version of the VTE agent. (Starting with version 6.1.2, you can generate unencrypted signatures of binaries inside GuardPoints). The previous signatures will be used until the next key rotation.
2. Before the next key rotation, the administrator resigns the binaries.
3. Do not remove the old signatures on the DSM until all agents have been upgraded to at least VTE version 6.1.2 (which introduced the ability to generate unencrypted signatures on binaries inside GuardPoints). Refer to the DSM Installation and Configuration Guide for information on how to do a manual resign.
4. When all agents have been upgraded, remove the old signatures.

If you are installing the VTE agents for the first time, there are no special steps, if no signatures have been defined. The agent will sign using the new method.

Note: In previous versions, if the binary was in a GuardPoint protected directory, but was the same as an unguarded binary, the administrator could restrict to only the guarded binary. With this change, the unguarded binary is now unrestricted. This means that if a user uses the unguarded binary and its SHA matches the guarded binary, it will now match as if it was the guarded binary.

Enable Automatic Signing for Host Settings

As of version 6.1.2, VTE blocks automatic re-signing of the host settings. Some users may have established procedures for updating system software based on the assumption that restarting the `vmd` will generate new signatures when signed software is updated. This process will not work with VTE unless you disable automatic signing.

To disable automatic signing:

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:

   ```
   RE_SIGN_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
   ```

WARNING

Enabling the automatic regeneration of signatures exposes a potential security vulnerability for agents. When enabled, host setting binaries are resigned when it receives a push from the DSM. If an attacker were to replace a binary with a Trojan, and then force a push from the DSM by, for example, restarting the agent, VTE could generate a signature for the malicious binary and pass it to the kernel.

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. You can instruct VTE to not trust any authentication attempt performed by certain identities by assigning restricted users to a user shell that VTE 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.

VTE 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:

1. In the DSM Management Console, click **Hosts > Hosts**.
2. Click on an existing Host name to edit the host.
3. Click **Host Settings** tab.
4. Add the following to the host 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 **OK**.

Using Advanced Encryption Set New Instructions (AES-NI)

To verify AES-NI hardware support, type:

```
# vmsec check_hwenc
```

Unlike the -c algo command above, VTE 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 VTE 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 VTE version
`-f`	runs `vmd` in the foreground

Display the Installed Version

To display the installed VTE version, type:

`# vmd -v`

System Response:

```
Version 6
6.3.1.38
2019-08-12 20:41:51 ()
Copyright (c) 2009-2019, Thales. All rights reserved.
```

agenthealth Utility

The `agenthealth` utility validates:

- Super-user privilege
- VTE Agent installation
- VTE registration to DSM Server
- VTE processes/modules that are running

- Available disk resources
- Current GuardPoints

  Tests if the agent can reach the GuardPoints
- VTE log directory resource status

  This directory contains pending VTE 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

To run the `Agenthealth` check script, type:

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

System Response:

```
Checking for super-user privilege ..... OK
Vormetric Agent installation ..... OK
Vormetric policy directory ..... OK
Registration to server ..... OK
Kernel modules are loaded ..... OK
VMD is running ..... OK
SECFSD is running ..... OK
dsm602-33-101.qa.com is resolvable ..... OK
dsm602-33-101.qa.com port 8446 is reachable ..... OK
dsm602-33-101.qa.com port 8447 is reachable ..... OK
dsm602-63-183.qa.com is resolvable ..... OK
dsm602-63-183.qa.com port 8446 is reachable ..... OK
dsm602-63-183.qa.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..... 2018-02-13 20:25:37.446
Checking available disk space..... OK
Checking logging space ..... OK
      Log directory is "/var/log/vormetric"
      File system for log data is "/", 32G free (17% full)
      Log directory contains 2 of maximum 200 files (1% full)
      Log directory contains 1 of maximum 100 Mbytes used (1% full)
Testing access to /ofx-fs1 ..... OK
Testing access to /gp1 ..... Access denied as per policy
```

agentinfo Utility (Java version)

The `agentinfo` utility collects system and VTE 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
```

Note: The Java version is supported on VTE/Linux.

check_host Utility

If a VTE 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 DSM during registration.

Run the `check_host` utility on a system that is hosting VTE 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 [[-a | -h | -i ] [-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 DSM and the host, and to register VTE on the DSM. After the host is registered, you can configure VTE, apply GuardPoints, or perform database backups. Run the Linux 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 DSM 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 VTE.

Platform Restrictions

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

- VTE 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).

Note: The existing behavior of using `fsfreeze|xfs_freeze` in the underlying XFS file system works for XFS only. We do not support `fsfreeze|xfs_freeze` on GuardPoints located on VxFS file systems.

`fsfreeze|xfs_freeze` is supported with Redhat 6 (2.6 kernel) only with XFS file system.

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 VTE 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 VTE 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 VTE 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 VTE 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.

Chapter 25: VTE and systemd

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

This chapter contains the following sections:

Overview of VTE and systemd

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

`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 VTE and systemd" on page 315). 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 VTE and systemd

The entries in the following table are Linux distributions that are compatible with VTE. The table shows which distributions implement `systemd`, and those that do not. See the *Vormetric Transparent Encryption Agent Compatibility Matrix* for the Linux distributions and kernels supported with your VTE version.

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

VTE Agent Control Changes on systemd

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

Command	Command syntax for distributions that support systemd
Start	`/etc/vormetric/secfs start`

Command	Command syntax for distributions that support systemd
Restart	`/etc/vormetric/secfs restart`
Stop	`/etc/vormetric/secfs stop`
Check status	`/etc/vormetric/secfs status`

The normal states of VTE 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 VTE, type:

```
# /etc/vormetric/secfs status
```

System Response

```
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
```

VTE 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 VTE during system startup and shutdown.

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

CAUTION
Configuring the dependencies as recommended here mean that whenever you stop VTE or if VTE stops unexpectedly, the dependent applications will automatically stop shortly before VTE 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 318

Task	For more information

Add the following lines to the `[Unit]` section of the `systemd` unit configuration file for each application:
`After=secfs-fs-barrier.service`
`BindsTo=secfs-fs-barrier.service`
	"Adding Dependencies to systemd Unit Configuration Files" on page 318

To the `Before=` line in the `[Unit]` section of the `secfs-fs-barrier.service` file, add the following: • `saslauthd.service` • An entry for each application you added to the `systemd` unit configuration file.	"Adding Applications to the secfs-fs-barrier.service File" on page 319

Add the following lines to the `[Unit]` section of the `saslauthd.service` configuration file:
`BindsTo=secfs-fs-barrier.service`
`After=secfs-fs-barrier.service`
	"Adding Dependencies to the saslauthd.service File" on page 319

Force the system to read the changed systemd configuration files by typing `systemctl daemon-reload`.	"Adding Dependencies to the saslauthd.service File" on page 319
Restart the applications that you shut down to make these changes.	"Adding Dependencies to the saslauthd.service File" on page 319

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 the `After=` and `BindsTo=` unit configuration options.

CAUTION

In some cases, the unit configuration file for an application other than VTE 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 VTE.

The `secfs-fs-barrier.service` service ensures that VTE services and dependent applications start after VTE services and stop before VTE 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 VTE agent:

File and Change	Purpose
File: Application-specific unit configuration file Add this line to `[Unit]` section: `After=secfs-fs-barrier.service`	In conjunction with the `BindsTo=` option, this option ensures that the application starts after the VTE agent on system startup.
File: Application-specific unit configuration file Add this line to `[Unit]` section: `BindsTo=secfs-fs-barrier.service`	If the VTE agent shuts down, the application also shuts down.

File and Change	Purpose
File: `secfs-fs-barrier.service` Add each application to the existing `Before=` line in the `[Unit]` section. Add `saslauthd.service` to the existing `Before=` line in the `[Unit]` section.	Ensure that the `secfs-fs-barrier.service` starts before the application services mentioned in the `Before=` line.

Location of Application Unit Configuration Files

To configure the proper startup order of services, modify the unit configuration file for applications that require VTE 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	/usr/lib/systemd/system/<*application_name*>.service
Ubuntu 16.04 Ubuntu 18.04	/lib/systemd/system/<*application_name*>.service

Adding Dependencies to systemd Unit Configuration Files

If your system supports `systemd` (see "Linux Distributions that Support VTE and systemd" on page 315), 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 VTE 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 the next section 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 VTE. The `secfs-fs-barrier.service` file ensures that VTE starts before dependent applications during boot and VTE 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 mongodb.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 mongodb.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, VTE will start before each application listed in the `Before=` line and those applications will stop before VTE 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

VTE 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 318.

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

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

```
Stopping the Vormetric 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.
isecespd.service
isectpd.service
mysql.service
These services are required to start manually.
Successfully stopped the Encryption Expert File System Agent.
```

WARNING

The 3rd party application services using the `BindsTo` clause that automatically stop must be restarted manually after an upgrade.

WARNING

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 26: Ubuntu Upstart Service Support

Vormetric Transparent Encryption (VTE) supports the Ubuntu upstart service framework. VTE services ensure that dependent Upstart application services start only after the VTE process starts. In addition, applications started by traditional scripts can also be synchronized to start after VTE processes have started.

These changes are specific to Ubuntu 14.04, or earlier versions. They do not apply to Ubuntu 16.04, Red Hat and SLES. This chapter contains the following sections:

Administering Vormetric Services

This section describes the startup and shutdown of the VTE services.

Starting the VTE Services

Run the following commands to start the VTE services in the order shown:

```
# start secfs-init

# start secfs-fs
```

Stopping the VTE Services

Run the following commands to stop the VTE services in the order shown:

```
# stop secfs-fs

# stop secfs-init
```

You cannot stop `secfs-fs` if guarded directories associated with mysql or other third-party applications are in use. Stop the application before stopping the VTE process. The Upstart framework on Ubuntu does not display error messages. Check for error messages in the following location:

```
/var/log/upstart/secfs-fs.log
```

Querying VTE Status

Use the following commands to obtain the VTE status:

Command:

```
secfs-fs start/running, # status secfs-fs
```

System Response:

```
secfs-fs start/running, process 1501
```

Command:

```
# status secfs-init
```

System Response:

```
secfs-init start/running
```

Vormetric Upstart Service Management Logs

You can find upstart services management log in the following locations:

```
/var/log/upstart/secfs-init.log
/var/log/upstart/secfs-fs.log
```

Upgrading VTE

There are no restrictions to upgrade VTE.

Administering Third-Party Services

VTE ensures that GuardPoints are available before mysql and all other dependent services are guarded.

Guarding mysql Folders (mysql already installed)

- Ensure that the latest patch is installed
- Confirm that secfs-init and secfs-fs Vormetric services are running
- Create and guard mysql folders
- Install and start mysql

Adding New Upstart Dependencies

VTE supports the following Upstart services on the following applications:

- mysql
- mongodb
- apache2
- postgres

If your application is not listed above, then modify the `secfs-fs-barrier.conf` as follows:

At the end of the `start on` section located at:

```
/etc/init/secfs-fs-barrier.conf
```

Add the following:

```
or starting <your_service_name>
```

For example, for service foo:

```
start on starting mysql or starting mongodb or starting apache2 or starting
postgres or starting foo
```

Configuring rc/sysvinit Services

You can synchronize applications started by traditional `sysvinit` scripts to start after VTE starts.

Enabling the barrier for rc services

Run the following command after guarding data to insure all rc (sysvinit) based services start after the VTE starts, type:

```
# update-rc.d secfs-fs-barrier defaults 00 99
```

Disabling the barrier for rc services

Run the following command after unguarding data to allow independent starting of rc (sysvinit)-based services, type:

```
# update-rc.d –f secfs-fs-barrier remove
```

Appendix A: Troubleshooting and Best Practices

Windows Systems

VTE will not register with the DSM

- If there is a firewall between the DSM and VTE, configure `vmd.exe` as a firewall exception on VTE for Windows. Otherwise, the DSM is unable to browse VTE.
- If using a Windows XP or Windows 2003 system Firewall, select **Control Panel > Windows Firewall > Exceptions > Add Program...** and browse for `vmd.exe`. The default location is

```
\Program Files\Vormetric\DataSecurityExpert\agent\vmd\bin\vmd.exe
```

- If using a Windows 7 system Firewall, select **Control Panel > System and Security > Windows Firewall > Allowed Programs...**click **Change settings**, click **Allow another program** and browse for `vmd.exe`. The default location is

```
\Program Files\Vormetric\DataSecurityExpert\agent\vmd\bin\vmd.exe
```

VTE with NFSv4: Permission denied error

You man see a permission denied error if you try to access files on an NFSv4 file system that is guarded by VTE.

VTE imposes a restriction on NFSv4 file systems to prevent write-only permissions from being set on individual files. You can work around this restriction by configuring read and write permissions on the same files.

You can also add a policy that allows write permissions.

This restriction applies only in the case of files resident in guarded NFSv4 file systems.

Americas
Arboretum Plaza II, 9442 Capital of Texas Highway North,
Suite 100, Austin, TX 78759 USA
Tel: +1 888 343 5773 or +1 512 257 3900
Fax: +1 954 888 6211 | E-mail: sales@thalesesec.com

Asia Pacific – Thales Transport & Security (HK) Ltd
Unit 4101-3, 41/F, Sunlight Tower, 248 Queen's Road East
Wanchai, Hong Kong | Tel: +852 2815 8633
Fax: +852 2815 8141 | E-mail: asia.sales@thales-esecurity.com

Europe, Middle East, Africa
Meadow View House, Long Crendon,
Aylesbury, Buckinghamshire HP18 9EQ
Tel: +44 (0)1844 201800 | Fax: +44 (0)1844 208550
E-mail: emea.sales@thales-esecurity.com

> thalescpl.com <