Protegrity FUSE File Protector Guide 9.1.0.0

Created on: Nov 19, 2024

Copyright

Table of Contents

Chapter 1

Introduction to the FUSE File Protector (FUSE FP)

The Filesystem in Userspace (FUSE) is an open source framework that enables you to develop a file system in the user space. The FUSE File Protector (FUSE FP) is based on the open source FUSE-based file system.

The FUSE FP provides the following distinct advantages over the Kernel File Protector (Kernel FP):

- Ease of deployment and upgrade
- Free from conflicts with other kernel-based third-party software

For more information about improvement over kernel FP, refer to *Improvements over the Kernel File Protector (Kernel FP)*.

The FUSE FP uses the FUSE framework for protecting files and directories. The FUSE FP provides support for encryption, decryption, and other file protection features.

1.1 Architecture

The following diagram illustrates the architecture of FUSE FP.

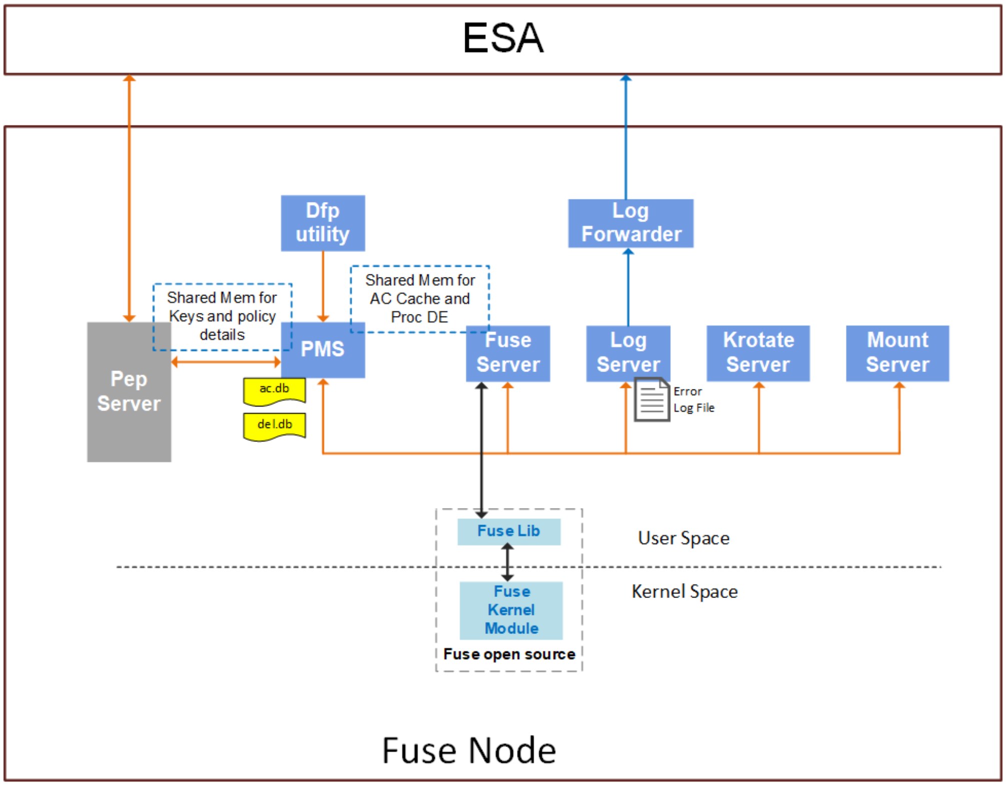

Figure 1-1: Architecture of FUSE FP

The key components of FUSE FP are:

- **FUSE FP service**

  The FUSE-based file system is mounted on the mount point. The FUSE mount utility creates the mount point.

  > **Note:** For more information about the mount point, refer to *Concept of FUSE Mount Point*.

- **PEP server**

  The PEP server is installed on the FUSE FP node and it provides the data element key for the file protection.

- **Process Management Service (PMS)**

  The PMS is responsible for executing `dfp` commands. It also maintains the data structure for the delegated process and their related policies.

- **Utilities**

  The Utilities consist of the *dfpadmin*, *dfp*, and *dfpshell* utilities.

- **Log service**

  The Log service handles the audit logs and updates the log files.

- **Key rotate service**

  The Key rotate service monitors the key rotation to be performed on the files specified in the *keyrotate.db* file.

- **Mount service**

The Mount service is added to handle multiple mount points. This service keeps an updated information about the currently active mount points.

On the Enterprise Security Administrator (ESA), an unstructured policy is created and deployed to data store. To protect, mount the required path using the FUSE FP `mount` command. After creating the mount point, you can protect the files or directories using the `dfp ac or file` commands. An authorized user, process, or program with the required policy loaded, can see the files in cleartext format through the mount point. The files appear as protected outside the mount point.

> **Note:** For more information about the policy, refer to *Creating and Deploying Policies*.

> **Note:** For more information about the data store, refer to *Data Stores*.

Perform the following steps for protecting files and directories:

1. Add the data stores on the ESA.
2. Create policies with the required data elements.
3. Identify files and directories that need to be protected.
4. Create the FUSE FP mount point (protection point) for the path that needs protection.
5. Protect files using the `dfp ac or file` commands.
6. Delegate the programs, users, or process to access the mount point.

1.2 Concept of FUSE FP

The following diagram illustrates the FUSE FP workflow.

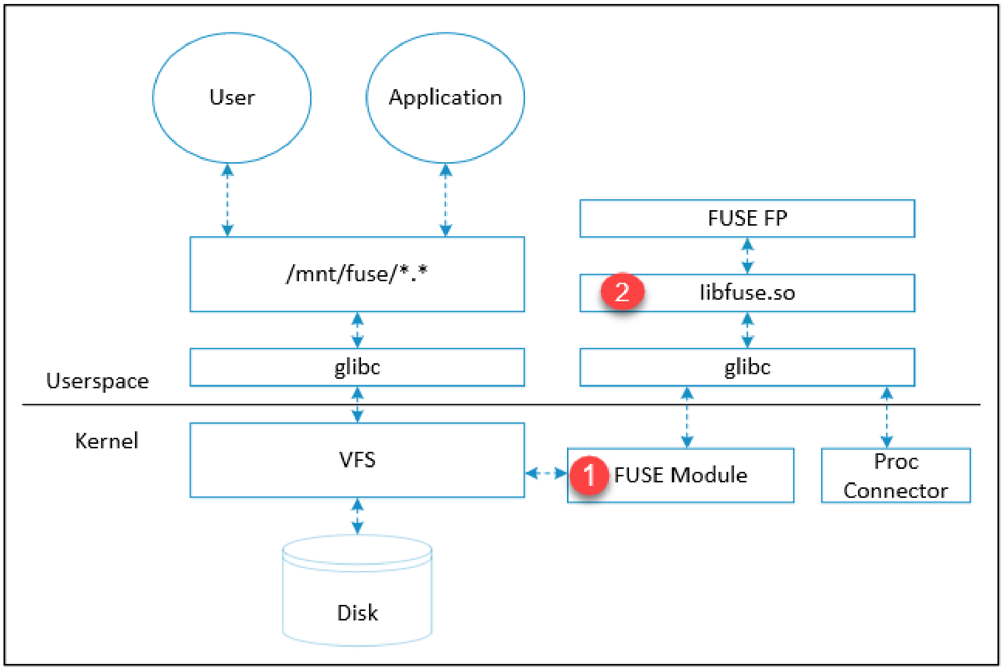

Figure 1-2: The FUSE FP Workflow

The FUSE FP uses the following two components of the FUSE framework:

1. Open source FUSE module
2. Open source *libfuse.so* library

The FUSE module is in the kernel space and the *libfuse.so* library is in the user space. The FUSE module registers a FUSE-based file system with the Virtual File System (VFS).

You can access the FUSE-based file system through the FUSE FP mount point, which is created by the FUSE FP mount utility.

After you mount the FUSE-based file system, all the file system calls targeting the mount point are forwarded to the FUSE module.

To communicate with the user space library, the FUSE module uses the character device `/dev/fuse` file. The character device `/dev/fuse` file is created after the FUSE driver module is registered with the VFS.

The VFS layer performs the following tasks:

- Registers the FUSE FP calls that are targeting the mount point
- Directs all the file operations that are targeting the FUSE FP mount to the user space FUSE FP calls

Note: A mount point is the directory where the FUSE-based file system is applied by the FUSE FP mount utility.

1.3 Concept of FUSE Mount Point

The FUSE FP is implemented using the FUSE-based file system. The FUSE FP enables support for the file system operations to provide the protection functionality. To enforce the FUSE-based file system, you must create a FUSE FP mount point for the required path. The `dfp mount` command of the FUSE FP is used to create the FUSE FP mount point.

The underlying file systems are transparent to the application. The standard *glibc* layer makes the file system calls to the underlying file system. These calls are passed to the kernel VFS layer in the Linux kernel. The VFS layer detects the call for the FUSE FP and directs the call to the FUSE FP.

The following diagram illustrates the mount concept of the FUSE FP.

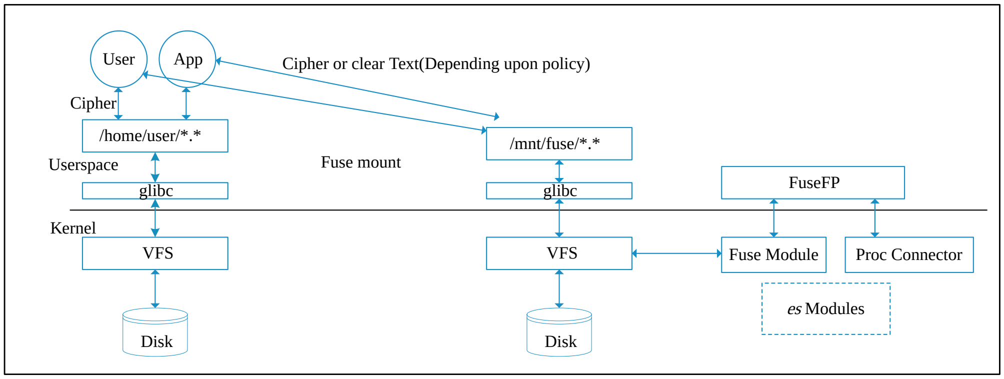

Figure 1-3: FUSE FP Mount Concept

Note: The Proc Connector is the linux kernel interface that provides notification for process events.

Note: The Enhanced Security (*es*) modules are now delivered as a separate package. For more information about the packages required for specific platforms, contact Protegrity Support.

1.4 Supported Platforms, File Systems, and FUSE Library Versions

The FUSE FP supports the following:

Platforms and library versions

- RHEL versions 7 and 8, and Centos version 7
- FUSE user space library (*libfuse.so*), version 2.9.4

File Systems

- Extent File System (XFS)
- Extended File System (EXT4)
- Network Shared File System (NFS), version 3 and 4
- SAMBA
- Global File System 2 (GFS2)

1.5 Features

The following are the salient features of FUSE FP:

- Enable encryption and decryption of files and directories.
- Provide access control feature to protect files and directories.
- Support user, process, and program delegation.
- Support key rotation by rotating the data encryption key periodically and protects the data with the new generated key.
- Provide seamless usage of files and directories that are protected by Kernel FP, provided both the File Protector and FUSE FP are using the same policy.
- Provide audit logging for the generated events for any security operation is performed using the FUSE FP.

Chapter 2

Installing and Uninstalling

This section describes the installation and configuration of the FUSE FP.

Perform the following tasks to install FUSE FP:

Task Order	Description	Reference
1.	Run the Log Forwarder Installer.	For more information about installing the Log Forwarder, refer to *Installing the Log-Forwarder*
2.	Run the PEP server installer.	For more information about installing the PEP server, refer to *Installing the PEP Server*.
3.	Run the pre-install check.	For more information about running the pre-install check, refer to *Running the FUSE FP Pre-installation Check Script*.
4.	Run the FUSE FP installation script.	For more information about installing the FUSE FP, refer to *Installing the FUSE FP*.

> **Note:** The *root* user privileges are required for the installing the FUSE FP.

2.1 Installing the Log-Forwarder

To install the Log-Forwarder:

1. Log on to the node as a user having *root* privilege and download the *FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.tgz* installer.
2. Extract the *FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.tgz* file using the following command.

   ```
   tar -xvf FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.tgz
   ```

   The following files are extracted:

   - *PepServerSetup_Linux_x64_1.2.*.sh*

- *LogforwarderSetup_Linux_x64_1.2.*.sh*
- *FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.sh*
- *FileProtector_LINUX-ALL-64_x86-64_FUSEPreInstallCheck_9.1.0.0.x.sh*
- *FileProtector_LINUX-ALL-64_x86-64_FUSEClusterDeploy_9.1.0.0.x.sh*
- *FuseFileProtector_ClusterDeploy_hosts*
- *INSTALL.txt*

3. Run the following command.

   ```
   ./LogforwarderSetup_Linux_x64_1.2.*.sh
   ```

   A promt to enter the Audit Store end point appears.
4. Enter the IP address and the port number for the Audit Store endpoint.

   > **Note:** Enter the ESA IP address as the IP address for the Audit Store endpoint.

   > **Note:** The default port number is 9200.

   A prompt appears to add another audit store end point.
5. To add another audit store end point, enter *y*.
   A prompt appears to accept the installation process.
6. After successfully adding all the audit store endpoints, enter *y* to accept the installation process.
   The installation process for the Log Forwarder begins.
7. After a successful installation, run the following command to start the Log Forwarder.

   ```
   /opt/protegrity/fluent-bit/bin/logforwarderctrl start
   ```

   > **Note:** Before you install the FUSE FP, ensure that the Log Forwarder is running.

2.1.1 Silent Mode of Installation

You can also execute the Log Forwarder installer without any manual intervention, which is also known as the Silent mode of installation. The following parameters must be provided to execute the installer in the Silent mode.

Table 2-1: Parameter List for Silent Installation

Parameter	Description
-endpoint1, -endpoint2, -endpoint3	Audit Sore IP address and the Port number where the Log forwarder listens for logs **Note:** The default port number is *9200.* **Note:** The parameters *-endpoint2* and *-endpoint3* are optional.
-dir	Installation directory of the Log Forwarder, which is optional. If the installation directory is not specified, then the installation path is the default directory, which is the *../opt/protegrity* directory.

Parameter	Description
-pepdir	Installation directory of the PEP server, which is optional. If the installation directory is not specified, then the installation path is the default directory, which is the *../opt/protegrity* directory.

At the command prompt, type the following command from the installer directory.

```
LogforwarderSetup_Linux_x64_1.2.*.sh
        <ip address and port number> [-endpoint2 <ip address and port
        number>] [-endpoint3 <ip address and port number>]
```

If you want to install the Log Forwarder and the PEP server in a directory other than the default directory, then you can add the*-dir*parameter to the command to specify the Log Forwarder installation directory and the *-pepdir* parameter to the command to specify the PEP server installation directory. The following snippet displays a sample command.

```
LogforwarderSetup_Linux_x64_1.2.*.sh
        [-endpoint1 <ip address and port number>]
        [-endpoint2 <ip address and port number>]
        [-endpoint3 <ip address and port>]
        -dir <Log Forwarder installation directory>
        -pepdir <PEP server installation directory>
```

2.2 Installing the PEP Server

This section describes the steps to install the PEP server.

Before you begin

Verify the following prerequisites.

- Ensure that the *root* user performs installation.
- Ensure that a minimum of 100 MB disk space is available.
- Ensure that the ESA is running to download the certificates.

To install PEP server:

1. Run the PEP server installation script using the following command.

   ```
   ./PepServerSetup_Linux_x64_1.2.*.sh
   ```

   A prompt for the ESA host name or IP address appears.

   > **Note:** The *./PepServerSetup_Linux_x64_1.2.*.sh --help* provides the following options for installation.
   >
   > - *-esa host*
   > - *-esaport port*
   > - *-certuser name*
   > - *-certpw password*
   > - *-dir installation_directory*

> The following snippet describes the options of `./PepServerSetup_Linux_x64_1.2.*.sh --help` command.
>
> ```
> ./PepServerSetup_Linux_x64_1.2.*.sh --help
> Usage: ./PepServerSetup_Linux_x64_1.2.*.sh (-esa host) (-esaport port) (-certuser
> name) (-certpw password) (-dir installation_directory)
> ```

2. Enter the ESA host name or IP address.

   A prompt to configure the host as ESA proxy appears.
3. Press **ENTER**.

   A prompt for the ESA user name appears.
4. Enter the ESA user name.
5. Press **ENTER**.

   A prompt for the ESA password appears to download the certificates.
6. Enter the ESA administrator password.
7. Press **ENTER**.

   After the successful installation of PEP server, the following message appears.

   ```
   Unpacking...
   Extracting files...
   Downloading certificates from 10.10.140.165:8443...
     % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                    Dload  Upload   Total   Spent    Left  Speed
   100 30720  100 30720    0     0   9065      0  0:00:03  0:00:03 --:--:--  9067

   Extracting certificates...
   Certificates successfully downloaded and stored in /opt/protegrity/defiance_dps/data.

   Protegrity PepServer installed in /opt/protegrity/defiance_dps.
   ```
8. Run the following command to start the PEP server.

   ```
   /opt/Protegrity/defiance_dfp/bin/pepsrvctrl start
   ```

2.3 Running the FUSE FP Pre-installation Check Script

Before installing the FUSE FP, you must run the pre-installation check script to verify the prerequisite modules are present and install them separately.

Before you begin

Before running the pre-installation check script ensure that the following prerequisites are met.

1. Ensure that the FUSE framework is installed in the kernel. If the FUSE framework is not installed, then run the following command to install the FUSE framework.

   `yum install fuse`

2. Ensure that the FUSE module is installed in the kernel. If the FUSE module is not installed, then run the following command to install the FUSE module.

`modprobe fuse`

Caution: Ensure that the FUSE framework is installed in the kernel before running the `modprobe fuse` command to avoid the failure of the pre-installation check script.

Note: If any prerequisite is missing, then the following message appears:

```
Unfortunately, Fuse-File Protector Pre-Installation Checking Failed!
```

To run the pre-installation check script:

Run the pre-installation check script using the following command.

`./FileProtector_LINUX-ALL-64_x86-64_FUSEPreInstallCheck_9.1.0.0.x.sh`

If the pre-installation requirements are met, then the following message appears:

```
[root@rhel74base fuse_install]# ./FileProtector_LINUX-
ALL-64_x86-64_FUSEPreInstallCheck_9.1.0.0.x.sh

Enter ('PepServer') installation directory.
It should have the 'defiance_dps' sub directory
[/opt/protegrity]

Congratulations, Fuse-File Protector Pre-Installation Checking Passed!
```

2.4 Installing the FUSE FP

This section describes the prerequisites and procedures required to install the FUSE FP.

Before you begin

Ensure the following criteria is met:

- Ensure that the *ed editor* is installed for the FUSE FP installer. If the *ed editor* is not installed in the system, then run the following command to install the *ed editor*.

  `yum install ed`

To install the FUSE FP:

1. Run the following script on the node.

   `./FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.tgz`

   A prompt to continue with the installation of the FUSE FP appears.

2. If you need to use the FUSE FP, then enter *yes*.
3. Alternatively, if you want to abort the installation of FUSE FP, then enter *no*.
4. Enter the installation directory for the PEP server.

   The PEP server is installed in the `/opt/protegrity/defiance_dps` directory by default.

5. Press **ENTER**.

   A prompt for the installation of the FUSE FP directory appears.
6. Enter the installation directory for the FUSE FP.

   A new directory is created in the `/opt/protegrity` installation directory by default.
7. Press **ENTER**.

   The installation of the FUSE FP starts.

   The following files are extracted in the `/opt/protegrity/fileprotector/fuse/bin` directory:

   - *ac_migrate_v1tov2.sh*
   - *dfp*
   - *dfpadmin*
   - *dfp_clone*
   - *dfp_fusefs_server*
   - *dfp_get_env.sh*
   - *dfp_krotate_server*
   - *dfp_log*
   - *dfp_log_server*
   - *dfp_mnt_srv*
   - *dfp_process_management_server*
   - *dfp_remote*
   - *dfp_remote_management_server*
   - *dfp_service_manager*
   - *dfpshell*
   - *dfp_uninstall_fusefp*
   - *dfp_wrapper*
   - *fp_to_fuse.sh*
   - *libfpfuse.so.2.9.4*

   The following note appears during installation.

   ```
   Note: File protector config files not found in current install directory.
   If you have installed File protector before in some other path and are willing to restore
   the File protector configutration
   Please run the /opt/protegrity/fileprotector/fuse/bin/fp_to_fuse.sh after installation.
   ```

   > **Note:**
   >
   > If you have installed kernel based File Protector before in another path and want to migrate the FUSE FP, then run the following script after installation.
   >
   > `/opt/protegrity/fileprotector/fuse/bin/fp_to_fuse.sh`

8. Enter a new *dfpshell* password.

   > **Note:** The *dfpshell* is the system administrator shell for the File-Protector. The *dfpshell* password must contain a minimum of 8 characters and a maximum of 129 characters in length. It should contain a mix of numeric, alphabetic, and printable characters.

9. Press **ENTER**.

A prompt to verify the *dfpshell* password appears.

10. Re-enter the *dfpshell* password to verify.
11. Press **ENTER**.

    The following message appears.

```
Create dfpshell password successfully!

WARNING: Enhanced security kernel modules used for applying protecting Installation path
and source path protection for AC protected files on fuseFP mount are DISABLED.
If you wish to enable the kernel modules after successfull installation, Please run
dfpadmin help command.

Initializing the services...

File Protector(FUSE-ALL) installed in /opt/protegrity/fileprotector/fuse .

INFO: Please create the mount point directory and use <dfp mount> command with options
for starting dfp_fusefs_server.
 It is highly recommended to restart All the bash login sessions
 in order to update configuration settings.
```

> **Caution:** It is recommended to restart all the *bash* login sessions after installation to update the configuration settings of the FUSE FP.

After the successful installation of the FUSE FP, you must update the FUSE FP configuration files.

For more information about configuration files, refer to *Setting the Configuration Files*.

2.5 Installing the FUSE FP on a Linux Cluster

The FUSE FP is provided with a *FUSE FP_ClusterDeploy* installation script to enable installation in a cluster environment. The clone installer script helps to automate the installation and configuration of the PEP server and the FUSE FP on each node of the cluster by cloning the installed binaries, configuration files, and database files from the host node.

Before you begin
Prerequisites:

- All the nodes in the cluster must be configured with the same *dfpshell* password.
- The nodes in the cluster on which the cluster deploy script is executed, must have an SSH public key authentication setup.
- The same major kernel version must be installed on all nodes in the cluster.
- Ensure that all the FUSE FP prerequisites are met on each node.

  For more information about the prerequisites of FUSE FP, refer to *Installing the PEP Server*.

➤ To install the FUSE FP on a Linux cluster:

1. Ensure that the FUSE FP is installed on the host node and all the FUSE FP services are running successfully.

> **Note:** The host node can be any node in the cluster.

2. Modify the *FuseFileProtector_ClusterDeploy_hosts* file in the FUSE FP installer, and add all the host names or IP addresses for all the nodes in the cluster.

```
[root@rhclus-n4 Fuse_latest]# cat FuseFileProtector_ClusterDeploy_hosts
rhclus-n5
rhclus-n6
[ root@rhclus-n4 Fuse_latest]#
```

3. Run the *FUSE FP_ClusterDeploy* installation script from the host node using the following command.

   ./FileProtector_LINUX-ALL-64_x86-64_FUSEClusterDeploy_9.1.0.0.x.sh --with-pepserver --DFPSHELL-PASSWORD=<pswd>

   Note: <pswd> is the *dfpshell* password that you have set.

```
[root@rhel74base fuse_install]# ./FileProtector_LINUX-
ALL-64_x86-64_FUSEClusterDeploy_9.1.0.0.x.sh --help
Options:
    --yes                            # answer 'yes' for all questions
    --with-pepserver                 # installation including pepserver if specified
    --DFPSHELL-PASSWORD=<pswd>       # specify dfpshell passwd <Mandatory>
    --help                           # print this message
```

4. Ensure that the target nodes have PEP server and Log Forwarder installed.

   For more information about FUSE FP commands, refer to *Commands Overview*.

5. Verify if all the FUSE FP services and commands, such as Process Management Service (PMS), Log, Krotate, and Mounts service are running successfully on all the target nodes.

   Caution: It is recommended to restart all the bash login sessions after installation to update the configuration settings of the FUSE FP.

   After the FUSE FP is installed on the Linux cluster, the following message appears.

```
[root@rhclus-n4 Fuse_latest]# ./FileProtector_LINUX-
ALL-64_x86-64_FUSEClusterDeploy_9.1.0.0.x.sh --yes --with-pepserver --DFPSHELL-
PASSWORD=omnisecureipol
********************************************
Fuse File Protector 9.1 for FUSEClusterDeploy
********************************************
Unpacking...
Extracting files...

Installing Fuse File protector on all nodes, Please wait...
Done.
It is highly recommended to restart All the bash login sessions of target nodes in order
to update configuration settings.
```

2.6 Uninstalling

This section describes the steps for uninstalling the FUSE FP from a node.

Before you begin

- Ensure that no mount point is active before running the ***dfp_uninstall_fusefp*** command. Run the following command to check the active mount points.

  dfp mount list

  Note: The *dfpshell* privilege is required for executing this command.

- If any mount point is active, then run the following command to unmount all mount points.

```
dfp umount all
```

➤ To uninstall FUSE FP:

1. Run the following command to uninstall FUSE FP from a node.

   ```
   dfp_uninstall_fusefp
   ```

2. Enter *yes* to continue with the uninstallation.

   The FUSE FP uninstallation starts. The FUSE FP uninstallation script checks for active mount points.

3. Enter the *dfpshell* password.

   If no mount points is active, then the FUSE FP uninstalled successfully and the following message appears.

   ```
   Removing Fuse File-Protector(FuseFp) package...
   --Stoping FuseFP services, please wait ...
   Shutting down              pms :              [ OK ]
   Shutting down              log :              [ OK ]
   Shutting down          krotate :              [ OK ]
   Shutting down             mnts :              [ OK ]
   Fuse File-Protector(FuseFp) has been successfully removed.
   ```

Chapter 3

Setting the Configuration Files

The Configuration files contain configuration settings that are required for the FUSE FP services.

The following table describes the configuration files and their purposes for FUSE FP.

Table 3-1: FUSE FP Configuration Files

Configuration files	Description
disallow. conf	Contains the list of file paths that are not allowed to be FE encrypted.
ac_disallow.conf	Contains the list of file paths that are not allowed to be AC protected.
fusefs.conf	Contains details about the FUSE services. Each created mount point must have a related FUSE service. This configuration is common for all the FUSE FP mount points.
mnt_srv.conf	Contains details about the configuration of mount services.
misc. conf	Contains the configuration of Process Management Services (PMS).
audit_log.conf	Contains the default settings configuration for audits.
key_rotation.conf	Contains configuration related to the key rotation functionality of the encrypted files and specifies the time interval of key rotation.
automount.conf	Contains details about the permanent mount points of FUSE FP. This file entries are auto-generated and should not be manually edited.
debug.conf	Contains the configuration file used to set the parameters for debugging the system in real-time.

> **Note:**
>
> After updating the FUSE FP configuration files except *disallow. conf* and *ac_disallow.conf* files, ensure that you run the following command to update the configuration changes.
>
> ```
> dfpadmin update
> ```

3.1 *disallow.conf* File

In the configuration file, you can list the file and directory source paths that must not be protected by the `dfp file protect` command. By default, the system file source paths are added in this configuration file to prevent protection of system file.

To add a new file path in the *disallow.conf* file, update and save the new file path manually.

The following snippet describes the contents of the *disallow.conf* file.

```
#
# File protection disallow configuration file.
# The following paths are disallowed for Encyption.
# dfp file protect -noac -d <dename> <pathname>
# The  above command will fail when the below specifed path is tried to be encrypted
# Only absolute path(sourcepath) must be configured in the disallow.conf.
# It wont work with mount point path
#
# e.g.
#    /root/my_dir
#    /root/my_dir/my_file.txt
#
/etc
/etc/inittab
/var/adm
/var/adm/syslog
/usr/bin
/bin
/sbin
/boot
```

> **Caution:** It is recommended not to delete the system file paths. If you delete the system file paths, then the system files are available for protection by the FUSE FP commands. If you protect the system files, then this may disrupt the system operation.

3.2 *ac_disallow.conf* File

In the configuration file, you can list the file and directory source paths that must not be protected by the `dfp ac protect` and `dfp file protect` commands. By default, the system file source paths are added in this configuration file to prevent protection of system file.

To add a new file path in the *ac_disallow.conf* file, update and save the new file path manually.

The following snippet describes the configuration settings in the *ac_disallow.conf* file.

```
#
# File Protector access control disallow configuration file.
# The following paths are disallowed to protect by 'dfp ac protect'.
#
# e.g.
#    /root/my_dir
#    /root/my_dir/my_file.txt
#
# Only absolute path(sourcepath) must be configured in the ac_disallow.conf.
# It wont work with mount point path
#
#
/etc
/etc/inittab
/var/adm
/var/adm/syslog
/usr/bin
/bin
```

```
/sbin
/boot
/
```

> **Caution:** It is recommended not to delete the system file paths. If you delete the system file paths, then the system files are available for protection by the FUSE FP commands. If you protect the system files, then this may disrupt the system operation.

3.3 *fusefs.conf* File

The configuration file *fusefs.conf* provides details about the configuration of FUSE-based File System (FuseFS) services.

This file contains the configuration settings:

- FuseFS service connection type
- Client connection type
- Local log configuration
- LOGSET_LEVEL
- Mode to open local log file
- MAX FILE SIZE
- FuseFS mount default options

> **Note:**
>
> On a single node, only 50 FUSE mount points can be configured by FUSE FP.

> **Note:** If you need to enable any default configuration parameter in the *fusefs.conf* file, then add and uncomment the required configuration parameter under the respective heading. For example, if you want to enable the *Allow other option* in *FUSEFS default option*, then you must add and uncomment under the heading *FUSEFS default option*.

The following table describes the configuration settings that can be modified in the *fusefs.conf* file.

Table 3-2: Default Configuration Settings

Configuration Settings	Options	Default Value	Notes
FuseFS server connection type	SERVER_CONNECT_TYPE	ssl	
Client connection type	CLIENT_CONNECT_TYPE	TLSv1.2	
Port Configuration	STARTING_PORT	15000	
Local Log Configuration	LOGSET_LEVEL	3	
MODE: Mode to open local log file	MODE	0	
MAXFILESIZE	MAXFILESIZE	1	
FuseFS Default Option	ALLOW_OTHER UID=N GID=N BIG_WRITES AUTO_CACHE	BIG_WRITES AUTO_CACHE DEFAULT_PERMISSIONS HARD_REMOVE	If you enable the *ALLOW_OTHER* option, then it allows the authorized users for the source path to access the FUSE FP mount point. The *GID* option enables all users in the Linux system group.

Configuration Settings	Options	Default Value	Notes
	DEFAULT_PERMISSIONS		

Note: The typical FUSE-based use cases can be addressed with the default configuration settings value.

Note: The *use_ino* and *nonempty* mount option are not supported.

The following snippet describes the configuration settings in the *fusefs.conf* file.

```
#
# FuseFS Server Configuration
#
[Server Configuration]
# FuseFS server IP address
#
# FuseFS server connection type
#  Options:
#    ssl -- use SSL connection.
#
SERVER_CONNECT_TYPE=ssl
[Client Configuration]
#
#client connection type
#  (choose one option of the following values)
#           - TLSv1.2
#           - TLSv1.1
#           - TLSv1.0
#           - SSLv3
#
CLIENT_CONNECT_TYPE=TLSv1.2
[Port Configuration]
#
# Port Configuration for FuseFS server listening
#   Default starting port = 15000 (port range from 15001 - 15050)
# Port Limits:
#   Minimum Limit = 1023  (System reserved ports)
#   Maximum Limit = 65502 [65535(Total ports) - 50(Max mount points) - 1(Upper limit) = 65484]
#
# Starting Port should lie between : 1023 <= STARTING_PORT <= 65484
#
STARTING_PORT=15000
[Local Log Configuration]
#
# LOGSET_LEVEL: Level of logs to be acquired in local log file
# Following are the log levels,
# SEVERITY_CRITICAL 1
# SEVERITY_WARNING 2
# SEVERITY_INFO 3
# SEVERITY_DEBUG  4
# If LOGSET_LEVEL=3 , logs with level less than or equal to 3 is being logged.
#
LOGSET_LEVEL=3
#
# MODE: Mode to open local log file
# 2 modes are supported TRUNCATE and APPEND
# TRUNCATE 0
# APPEND 1
MODE=0
#
# MAXFILESIZE: Maximum file size of local log file, after reaching max file will reset to
size 0.
#              Unit for size is MB.
#              Maximum value can be 35.
# example:
#     MAXFILESIZE=2 , will set a maximum size to 2MB.
MAXFILESIZE=1
#FuseFS mount with default options
```

```
#
#
[FuseFs Default Option]
#
#ALLOW_OTHER            all user (including root) can access the files
#ALLOW_ROOT               allow access to root
#DEFAULT_PERMISSIONS        enable permission checking by kernel
#FSNAME=NAME             set filesystem name
#SUBTYPE=NAME            set filesystem type
#MAX_READ=N            set maximum size of read requests
#HARD_REMOVE            immediate removal (don't hide files)
#READDIR_INO            try to fill in d_ino in readdir
#DIRECT_IO            use direct I/O
#KERNEL_CACHE            cache files in kernel
#AUTO_CACHE            enable caching based on modification times (off)
#NOAUTO_CACHE            set file permissions (octal)
#UMASK=M            set file owner
#UID=N                set file group
#GID=N                cache timeout for names (1.0s)
#ENTRY_TIMEOUT=T        cache timeout for deleted names (0.0s)
#NEGATIVE_TIMEOUT=T        cache timeout for attributes (1.0s)
#ATTR_TIMEOUT=T            auto cache timeout for attributes (attr_timeout)
#AC_ATTR_TIMEOUT=T        auto cache timeout for attributes (attr_timeout)
#INTR                allow requests to be interrupted
#INTR_SIGNAL=NUM        signal to send on interrupt (10)
#MAX_WRITE=N            set maximum size of write requests
#MAX_READAHEAD=N        set maximum readahead
#ASYNC_READ            perform reads asynchronously (default)
#SYNC_READ            perform reads synchronously
#ATOMIC_O_TRUNC            enable atomic open+truncate support
#BIG_WRITES            enable larger than 4kB writes
#NO_REMOTE_LOCK            disable remote file locking
#
#
#
#Most of the generic mount options described in mount are supported
#
#
#RO
#RW
#SUID
#NOSUID
#DEV
#NODEV
#EXEC
#ATIME
#NOATIME
#SYNCASYNC
#
#
# Set default option for FuseFS
#
# Some options needs values,
#   if its value is invalid or not set then mounting will be failed
#
#
BIG_WRITES
AUTO_CACHE
DEFAULT_PERMISSIONS
HARD_REMOVE
```

3.4 *mnt_srv.conf* File

The configuration file *mnt_srv.conf* provides details about the mount server configuration.

This file contains the following details:

- Mount server IP address
- Mount server listening port

- Mount server listening threads
- Mount server connection type
- Client connecting server IP address
- Client connecting server port
- Client connection type

The following table describes the configuration settings for the *mnt_srv.conf* file.

Table 3-3: Default Configuration Settings

Configuration Settings	Options	Default Value
Mount server IP address	SERVER_IP_ADDR	127.0.0.1
Mount server listening port	SERVER_LISTEN_PORT	15316
Mount server listening threads	SERVER_LISTEN_THREADS	1
Mount server connection type	SERVER_CONNECT_TYPE	ssl
Client connecting server IP address	CLIENT_CONNECT_IP	127.0.0.1
Client connecting server port	CLIENT_CONNECT_PORT	15316
Client connection type	CLIENT_CONNECT_TYPE	TLSv1.2

> **Note:** The typical FUSE-based use cases can be addressed with the default configuration settings value.

The following snippet describes the configuration settings in the *mnt_srv.conf* file.

```
#
# Mount Server Configuration
#

[Server Configuration]
#
# Mount server IP address
#
SERVER_IP_ADDR=127.0.0.1

#
# Mount server listening port
#
SERVER_LISTEN_PORT=15316

#
# Mount server listening threads
#
SERVER_LISTEN_THREADS=1

#
# Mount server connection type
#  Options:
#    ssl -- use SSL connection.
#
SERVER_CONNECT_TYPE=ssl

[Client Configuration]
#
# client connecting server IP address
#
CLIENT_CONNECT_IP=127.0.0.1
#
# client connecting server port
#
CLIENT_CONNECT_PORT=15316
#
#client connection type
#  (choose one option of the following values)
#           - TLSv1.2
#              - TLSv1.1
#              - TLSv1.0
#              - SSLv3
#
CLIENT_CONNECT_TYPE=TLSv1.2
```

3.5 *misc.conf* File

The *misc.conf* file contains the following details:

- Policy Management server IP address
- Policy Management server listening port
- Policy Management server listening threads
- Policy Management server connection type
- PEP server installation directory
- Database encryption key directory location and it contains the *kekup.bin*, *master.key*, and *repository.key* files.
- Client connecting server IP address
- Client connecting server port
- Client connection type
- *dfpshell* privilege timeout interval
- Default access control switch position
- Default runtime binary validation

The following table describes the configuration settings that can be modified in the *misc.conf* file.

Table 3-4: Default Configuration Settings

Configuration Settings	Options	Default Value	Notes
Policy Management server IP address	SERVER_IP_ADDR	127.0.0.1	
Policy Management server listening port	SERVER_LISTEN_PORT	15312	
Policy Management server listening threads	SERVER_LISTEN_THREADS	1	
Policy Management server connection type	SERVER_CONNECT_TYPE	ssl	
The parent directory where PEP server installed	PEP_INSTALL_DIR	/opt/protegrity	The user needs to update whether the default path is used.
The directory where DataBase Encryption Key located	DFP_DB_ENC_KEY_DIR	15316	This path is updated by installer and should not to be updated manually
Client connecting server IP address	CLIENT_CONNECT_IP	127.0.0.1	
Client connecting server port	CLIENT_CONNECT_PORT	15312	
Client connection type	CLIENT_CONNECT_TYPE	TLSv1.2	
dfpshell privilege timeout interval in minutes	PRIVILEGE_TIMEOUT_INTERVAL	20	The user can configure this setting.

Note: The typical FUSE-based use cases can be addressed with the default value of configuration settings.

The following snippet describes the configuration settings in the *misc.conf* file.

```
#
# Policy Management Server Configuration
#

[Server Configuration]
#
# policy management server IP address
#
SERVER_IP_ADDR=127.0.0.1

#
# policy management server listening port
#
SERVER_LISTEN_PORT=15312

#
# policy management server listening threads
#
SERVER_LISTEN_THREADS=1

#
# policy management server connection type
#  Options:
#    ssl -- use SSL connection.
#
SERVER_CONNECT_TYPE=ssl

#
# the parent directory where pepserver installed
# it should have "defiance_dps" sub directory
# for example:
#     PEP_INSTALL_DIR=/opt/protegrity
#
PEP_INSTALL_DIR=/opt/protegrity

#
# the directory where DataBase Encryption Key localed
# it should have "kekup.bin", "master.key", "repository.key" sub files
# for example:
#     DFP_DB_ENC_KEY_DIR=/opt/protegrity/defiance_dps/data
#
DFP_DB_ENC_KEY_DIR=


[Client Configuration]
#
# client connecting server IP address
#
CLIENT_CONNECT_IP=127.0.0.1
#
# client connecting server port
#
CLIENT_CONNECT_PORT=15312
#
#client connection type
#  (choose one option of the following values)
#           - TLSv1.2
#           - TLSv1.1
#           - TLSv1.0
#           - SSLv3
#
CLIENT_CONNECT_TYPE=TLSv1.2

#
##  dfpshell privilege timeout interval in minutes
#
##  0 means disable timeout check
# Default will be set to 20 minutes.
#
PRIVILEGE_TIMEOUT_INTERVAL=20
```

```
#
# By Default the Binary validation during program delegation with md5sum is ON.
# During Runtime delegated binary will be validated for security reason.
#
```

3.6 *automount.conf* File

You can view the FUSE FP permanent mount points in the *automount.conf* file.

Caution: This file is auto-generated and should not be edited.

Caution: If you unmount the FUSE FP mount point using the system **`umount`** command, then the entry for the mount point is maintained in the *automount.conf* file and can be deleted using the **`dfp umount [--del-entry] <mount point>`** command.

The following snippet describes the configuration settings in the *automount.conf* file.

```
#
#FUSE File System Daemon Automount Configuration
#

#
# Mount Point Directory
#


[AUTOMOUNTLIST]

#
#FUSE File System Daemon Automount Configuration
#

##

##  1. Original Directory to be mapped to the Mount Point Default is  /

##  2. Mount Point Indicates the Empyty Directory Where the FUSE FS Will be Mounted

##  3.Operation can be rw (Read Write), ro(Read Only),, nodev(Do not set character or special
devices access on this partition),nosuid(do no set suid/sgid access on this partition)

##The Table Will Be Updated If the User Add the Mount Point Using the dfp utility

##

# Original Directory            MountPoint                     Option
/localsrc        /localmnt        modules=subdir,subdir=/localsrc        rw_lock
```

3.7 *audit_log.conf* File

The following table describes the configuration parameters that can be modified in the *audit_log.conf* file.

Table 3-5: Default Configuration Settings

Configuration Parameters	Options	Default Value	Notes
Audit Configuration Items	LOAD_POLICY	{ALL}/SF	The configuration of audit items is not supported in this release.
	DFPSHELL	{ALL}/SF	
	UPDATE	{ALL}/SF	
	UNINSTALL	{ALL}/SF	

Configuration Parameters	Options	Default Value	Notes
	KEY_ROTATION_ADD	{ALL}/SF	
	KEY_ROTATION_DEL	{ALL}/SF	

The following snippet describes the configuration settings in the *audit_log.conf* file, which is used to configure the user audit attributes.

```
[Audit]
#
# Audit Configuration Items
#
#  Some audit settings cannot be controlled by Data Element,
#  so audit settings for these operations could be controlled here.
#
# Overview:
# LOAD_POLICY: [<user/FS>, ...]               Log load policy for user.
# DFPSHELL: [<user/FS>, ...]                  Log run dfpshell, chage dfpshell passwd 
operation.
#
# Format Description:
# <event>: [<user/condition> ...]
# Where,
# 'event':       one of LOAD_POLICY, DFPSHELL
#               See section above for the meaning of each manifest symbols.
# 'user':       login-id or '{ALL}' for all users in the system.
# 'conditionsk': Any combinations of letter F, S.
#               F stands for Failure,
#               S stands for Success,
#
# Examples:
# LOAD_POLICY: guest/F                 For user 'guest', log failed load policy operation
# VOLUM_INI: tom/SF                    For tom log 'volume init' success or failure.
# UNINSTALL: {ALL}/SF                  For all users  log 'uninstall' success or failure.
# DFPSHELL: Mary/S, herry/S            For Mary and herry log success  run dfpshell.
#
# Any characters after '#' is comments.
# This is audit default setting configuration
LOAD_POLICY: {ALL}/SF
DFPSHELL: {ALL}/SF
UPDATE: {ALL}/SF
UNINSTALL: {ALL}/SF
KEY_ROTATION_ADD: {ALL}/SF
KEY_ROTATION_DEL: {ALL}/SF
```

Note: After you have updated the *audit_log.conf* file, ensure that you run the following command to update the changes in the file.

```
dfpadmin update
```

Important: The audit logs for the install, uninstall, or upgrade operations are generated irrespective of the configuration of *audit_log.conf* file.

3.8 *key_rotation.conf* File

The *key_rotation.conf* configuration file is used to set the parameters for configuring the key rotation.

The following snippet describes the configuration settings for the *key_rotation.conf* file.

```
#
#  File Encryption Key Rotation Configuration File
#
###############################################################
```

```
##
## Notes:
## 1. TIME:  The time to do the key rotation
##       - Format
##             [Minute]  [Hour]  [Day]  [Month]  [Week]
##               *         *       *       *        *
##               -         -       -       -        -
##               |         |       |       |        |
##               |         |       |       |        +----- day of week (0 - 6) (Sunday=0)
##               |         |       |       +------- month (1 - 12)
##               |         |       +--------- day of month (1 - 31)
##               |         +----------- hour (0 - 23)
##               +------------- min (0 - 59)
##
##       - options:
##           -Month: 1~12
##           -   Day: 1~31
##           - Week: 0~6   (0 means Sunday)
##           -  Hour: 0~23  (0 means 12 a.m.)
##           -Minute: 0~59
##
##     e.g.
##     30  21  *  *  *   ( means 21:30 every night )
##     0  0  *  *  1-5   ( means 00:00 pm from Monday to Friday )
##     0  6  *  *  6,0   ( means 6:00 am from Saturday to Sunday )
##
##############################################################
##
##  [Minute]  [Hour]  [Day]  [Month]  [Week]
0  0  *  *  *

##
## Times of retrying key rotation when the expired file was busy.
## NOTE: '0' means ignore and skip
##
KEY_ROTATE_RETRY_TIMES=3

##
## The interval (in seconds) between key rotation retries.
##
KEY_ROTATE_RETRY_INTERVAL=60
```

In this configuration file, you can enable the key rotation for encrypted files and specify the time interval for key rotation. The specified encrypted files are verified and the corresponding key is rotated.

After you configure the key rotation, if the data element key is changed and deployed again to the FUSE FP using the ESA, then the current active key replaces the previous encryption key. The new encryption key forces a re-encryption of the files encrypted with the previous encryption key.

Chapter 4

Managing the *dfpshell*

The *dfpshell* is the system administrator shell for the FUSE FP. It is a privileged mode of operation for the FUSE FP management that requires users to log on using a special *dfpshell* password.

You can set the *dfpshell* password, when you install the FUSE FP for the first time.

The *dfpshell* password must have the following characteristics:

- Minimum 8 characters in length.
- Contain a mix of numeric, alphabetic, and printable characters.

If you run the FUSE FP commands without the *dfpshell* privilege, then the following error message appears.

```
ERROR: file protector privilege is needed!
```

Note: For more information about the commands that require *dfpshell* privilege, refer to *Commands Overview*.

The *dfpshell* command uses the following syntax:

- `dfpshell`
- `dfpshell -t`
- `dfpshell -c`

Table 4-1: dfpshell Commands

Commands	Description

`dfpshell -t`	Checks if the current session has the *dfpshell* privilege. If the current process has the required privilege, then the following message appears:
	`Has privilege!`
	Else, it displays the following error message:
	`INFO: No privilege!`

`dfpshell -c`	Changes the *dfpshell* password.

4.1 dfpshell Password Management

You can create a *dfpshell* password, reset the password, and activate or deactivate the *dfpshell* mode using the FUSE FP.

The *-c* option changes both the key and the password. The command verifies the current key and prompts for the new key.

4.2 Changing the dfpshell Password

Before you begin

➤ To change the dfpshell password:

1. Run the following command.
   ```
   dfpshell -c
   ```
2. Enter the current *dfpshell* password.
3. Enter the new *dfpshell* password.
4. Verify the new *dfpshell* password.

4.3 Activating the dfpshell Mode

➤ To activate the dfpshell mode:

1. Run the following command in the shell.
   ```
   dfpshell
   ```
2. Enter the *dfpshell* password.
3. Run the following command to exit the *dfpshell* privileges.
   ```
   exit
   ```

Chapter 5

Licensing

The FUSE FP features and functionalities are determined by the status of *Protegrity Data Security Platform License* and the terms of the license agreement with Protegrity.

The FUSE FP license can have the following three states:

- Valid
- Expired
- Invalid

If the license is valid, then you have all read or write permissions for all the FUSE FP operations.

If the license is expired or invalid, then your permissions are determined by the following points.

- The license agreement with Protegrity
- The policy enforcement and management are enabled or disabled after the license has expired

For more information about licensing, refer to *Protegrity Data Security Platform Licensing Guide 9.1.0.0*.

5.1 Checking License Validity

The FUSE FP provides the **`dfpadmin`** commands to check the license validity.

➤ To check validity of the FUSE FP license:

Run the following command to verify whether the license is valid.

```
dfpadmin license check
```

- If the license of the FUSE FP is valid, then the following message appears.

```
File Protector License is OK!
```

- If the license of the FUSE FP is invalid, then the following message appears.

`File Protector license is invalid`

- If the license of the FUSE FP has expired, then the following message appears.

  `File Protector license is expired`

5.2 Checking License Status

This section describes how to check the license status of the FUSE FP using the ***`dfpadmin license status`*** command.

➤ To view the status details of the FUSE FP license:

Run the following command to view the license status details.

`dfpadmin license status`

The following license details appear:

- License State
- Valid Date
- Last Valid Date

```
[root@labcos64-64 ~]# dfpadmin license status
======================================================
   LICENSE STATUS
------------------------------------------------------
      License State : OK
         Valid Date : from 2017-11-03 05:24:32 to 2017-12-03 04:24:32
    Last Valid Date : 2017-12-01 01:36:28
```

5.3 Permissions Allowed for Invalid or Expired FUSE FP License

If the FUSE FP license is invalid or expired, then due to policy rights, no errors are generated when you perform certain file protection operations.

If the license of FUSE FP is expired or is invalid, then the following operations are allowed:

- Access encrypted data
- Run delegated programs
- Delegate processes that retain delegated policy permissions
- Decrypt encrypted data
- Undelegate programs or processes

5.4 Permissions Denied for Invalid or Expired FUSE FP License

If the license of FUSE FP is invalid or expired, then you cannot perform the following tasks:

- Encrypt a new file or directory
- Delegate a new program, process, or user

Chapter 6

Upgrading to v9.1.0.0

This section describes the steps to upgrade the FUSE FP from version v7.1.0 to v 9.1.0.0.

Before you upgrade to FUSE FP v9.1.0.0, ensure that the following products are upgraded to v9.1.0.0:

- ESA
- PEP Server

6.1 Upgrading ESA from v7.1 to v9.1.0.0

For more information about upgrading the ESA to v9.1.0.0, refer to the *Protegrity Upgrade Guide 9.1.0.0*.

6.2 Upgrading PEP Server from v7.1.0 to v9.1.0.0

This section describes the task to upgrade PEP Server from v7.1.0 to v9.1.0.0.

➤ To install the PEP Server v9.1.0.0 patch:

1. Login to the system where FUSE FP is installed.
2. View the available mount points using the following command.
   ```
   dfp mount list
   ```
3. Unmount all the active FUSE FP mount points using the following command.
   ```
   dfp umount all
   ```
4. Stop the PEP Server v7.1 using the following command.
   ```
   /opt/protegrity/defiance_dps/bin/pepsrvctrl stop
   ```
5. Check the status of the PEP Server using the following command.
   ```
   ps -ef |grep pep |grep -v grep
   ```
6. Extract the *FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.tgz* file using the following command.
   ```
   tar -xvf FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.tgz
   ```

   The following files are extracted:

- *PepServerSetup_Linux_x64_1.2.*.sh*
- *LogforwarderSetup_Linux_x64_1.2.*.sh*
- *FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.sh*
- *FileProtector_LINUX-ALL-64_x86-64_FUSEPreInstallCheck_9.1.0.0.x.sh*
- *FileProtector_LINUX-ALL-64_x86-64_FUSEClusterDeploy_9.1.0.0.x.sh*
- *FuseFileProtector_ClusterDeploy_hosts*
- *INSTALL.txt*

7. Install the Log Forwarder.

   For more information about installing the Log Forwarder, refer to *Installing the Log-Forwarder*.

   > **Note:** If the Log Forwarder is already installed and running, then skip this step.

8. Install the PEP Server v9.1.0.0 using the following command.

   ```
   ./PepServerSetup_Linux_x64_1.2.*.sh
   ```

   > **Note:** Ensure that you remove the PEP Server v7.1 installation directory files.

   For more information about installing the PEP Server, refer to *Installing the PEP Server*.

9. Start the PEP Server v9.1.0.0 using the following command.

   ```
   /opt/protegrity/defiance_dps/bin/pepsrvctrl start
   ```

10. Ensure that the PEP Server is running using the following command.

    ```
    ps -ef| grep pep| grep -v grep
    ```

6.3 Upgrading FUSE FP from v7.1 to v9.1.0.0

This section describes the steps to upgrade the FUSE FP from version v7.1 to v9.1.0.0. If you are upgrading the FUSE FP to v9.1.0.0, then ensure that the ESA is v9.1.0.0.

➤ To upgrade the FUSE FP from v7.1 to v9.1.0.0:

1. Upgrade the FUSE FP from version 7.1 to version 9.1.0.0 using the following command.

   ```
   ./FileProtector_LINUX-ALL-64_x86-64_FUSE-ALL_9.1.0.0.x.sh
   ```

2. Ensure that the required policies are available from ESA v9.1.0.0 using the following command.

   ```
   dfp info
   ```

After successful upgradation of FUSE FP from v7.1 to v9.1.0.0, ensure that all the services are running and all the FUSE FP commands are working accurately with the ESA v9.1.0.0.

Chapter 7

Using the Policy Management

You can create and manage the policies of the FUSE FP using ESA. A FUSE FP policy stores one or multiple data elements. Each FUSE FP policy is protected by a password that is defined at the ESA.

> **Note:** For more details about a policy, refer to *Protegrity Enterprise Security Administrator Guide 9.1.0.0*.

7.1 Open and Close Multiple Policies

You can load a policy in the terminal using `dfp` commands of the FUSE FP. The loaded policy becomes unavailable when the terminal is closed.

The following table explains how to access and close policies:

Table 7-1: How to Access and Close Policies

Operations	Required Steps
Open multiple policies	Run the following command for each policy that you want to access. `dfp start –p <policy name>`
Close a policy	At the shell prompt, run the following command to close the policies in the reverse order of opening them. `exit`
Stop the current shell session by starting a new shell session or a new program with no loaded policy in the process (command prompt)	On the shell prompt, run the following command. `dfp start -n [<program>]`
Verify any policy is loaded in the process (command prompt)	Run the following command. `dfp info`

> **Note:**
>
> You cannot use **ENTER** which ends the password entry. You cannot use **BACKSPACE** to delete any incorrect keystrokes.

> If you entered a wrong password, you must press **ENTER**, and then press **ENTER** again at the password confirmation prompt.
>
> The mismatched password entries can cause confirmation to fail. You must enter your password again in case of the mismatched password entries.

7.2 Removing the loaded Policies for a Program

This section describes the steps you must perform to remove the loaded policies for a program.

➤ To remove the loaded policies for a program:

Run the following command.

```
dfp start -n
```

To get the loaded policy status of a running process, such as view the names and access permissions, or the audit masks of the data elements on a loaded policy, run the following command.

```
dfp proc [–l] [<pid>]
```

Chapter 8

Commands Overview

This section describes the FUSE FP commands and their usage. You can run the FUSE FP commands using the command line interface, which is used for configuring protection and management of the FUSE FP.

8.1 dfp Commands

Using the *dfp* commands, you can perform the following tasks:

- Load policies and its data elements
- Protect and unprotect files and directories
- Encrypt and decrypt files
- Delegate and undelegate programs, processes, and users
- Add, delete the key rotation configuration for the encrypted files and directories
- Remove the inactive key rotation configuration for the encrypted files and directories
- Display the status of AC, delegation, and key rotation for files and directories

The following snippet displays a list of all the *dfp* commands of FUSE FP.

Figure 8-1: FUSE FP dfp Commands

```
[root@labrh73base ~]# dfp info


*************************************************
FUSE File Protector 9.1.0.0.0
*************************************************



Available Policy List:
-------------------------
1: fusepolicy



Data Elements of Process [15634]: empty



[root@labrh73base ~]# dfp
```

```
dfp version 9.1.0.0.0 (Jan 26 2023)
Copyright © 2023 Protegrity Corporation. All Rights Reserved.



Usage:
dfp start -p <role>@<policy>
dfp start -p <policy>
dfp info
dfp proc [-l] [ <PID> ]
dfp proc
dfp delegate [-f] [-s] [-o <options>] -e <program> <role>@<policy>
dfp delegate [-f] [-r] -p <PID> <role>@<policy>
dfp delegate [-f] -u <username> <role>@<policy>
dfp delegate import [-f] <imported delegate.db file>
dfp delegate export <exported delegate.db file>
dfp undelegate -e <program>
dfp undelegate -u <username>
dfp undelegate [-r] -p <PID>
dfp delegate cleanup
dfp delegate status
dfp delegate help
dfp file krotate add [-f] [-r] [-p <policy> <-|passwd>] <path ...>
dfp file krotate del [-r] <path ...>
dfp file krotate cleanup [-y] [path-wildcard]
dfp file krotate status
dfp file protect [-noac] [-f] -d <data element> <file>
dfp file protect [-noac] [-f] [-r] -d <data element> <folder>
dfp file unprotect <file>
dfp file unprotect [-r] <folder>
dfp file stat <file>
dfp file stat [-r] <folder>
dfp file dump <file>
dfp ac stat <file or folder>
dfp ac protect [-f] -d <data element> <file>
dfp ac protect [-f] [-r] -d <data element> <folder>
dfp ac protect [-f] [-I] -d <data element> <folder>
dfp ac unprotect <file>
dfp ac unprotect [-r] <folder>
dfp ac cleanup
dfp ac status
dfp ac update
dfp ac import [-f] <imported ac.db file>
dfp ac export <exported ac.db file>
dfp version
dfp mount [--add-entry] [--rc] [--gfs] [< -o [fuseoption1] -o [fuseoption2]..>] <mount point>
dfp mount -V
dfp mount list
dfp mount all
dfp umount [--del-entry] <mount point>
dfp umount all
dfp help
```

The following table describes the *dfp* commands and their *dfpShell* privileges.

> **Note:** The following *dfpShell* privilege column denotes whether you should provide the *dfpShell* password before running the respective command.

Table 8-1: FUSE FP dfp Commands

Commands	*dfpshell* Privilege	Description
dfp start -p <role>@<policy>	No	Loads the data elements of the policy role in the process (bash shell).
dfp start -p <policy>	No	Loads the policy data elements in the process (bash shell).

Commands	*dfpshell* Privilege	Description
`dfp info`	No	Displays product information, such as product version, and available policy list on the PEP server, along with data elements of the current process. **Note:** Display the policy data elements, if the policy is loaded.
`dfp proc [-l] [ <PID> ]`	No	Displays information related to data elements of a specified process. The following list specifies the command options and their meaning. • *-l* - Displays the following detailed information about the data elements. • Access mask • Success audit mask • Failure audit mask • No access operation, which includes EXPT, NULL, and CIPH status • Data element name with the corresponding policy • *<PID>* - Process ID for a particular process whose data elements you want to view.
`dfp delegate [-f] [-o <options>] -e <program> <role>@<policy>`	Yes	Delegates or authorizes a program with policies, which are enforced when the program starts. *-f* - Forces delegation.
`dfp delegate [-f] [-r] -p <PID> <role>@<policy>`	Yes	Delegates a process with policies. The following list specifies the command options and their meaning. • *-f* - Forces delegation. • *-r* - Forces recursive delegation on the child processes.
`dfp delegate [-f] -u <username> <role>@<policy>`	Yes	Delegates or authorizes a user with the policies, which become available when the user creates a process. For example, if a newly-delegated user logs into a machine, then the policies are automatically loaded for the user. *-f* - Forces delegation.
`dfp delegate [-f] [-s] [-o <options>] -e <program> <role>@<policy>`	Yes	Delegates scripts (bash, python, perl) using the wrapper mechanism.
`dfp undelegate –e <program>`	Yes	Disables program and script delegation.
`dfp undelegate -u <username>`	Yes	Disables user delegation.
`dfp undelegate [-r] -p <PID>`	Yes	Disables process delegation referred by PID. *-r*- Forces recursive delegation on the child processes.
`dfp delegate cleanup`	Yes	Removes the inactive delegations in the delegation list. **Note:** The term inactive denotes the delegated binaries that are removed from the system.

Commands	***dfpshell* Privilege**	**Description**
`dfp delegate status`	Yes	Displays the list of delegated program and user.
`dfp delegate help`	No	Displays all the delegation commands.
`dfp file krotate add [-f] [-r] [-p <policy> <-|passwd>] <source directory path>`	Yes	Adds the configuration to rotate keys for the encrypted files and directories.
`dfp file krotate del [-r] <source directory path>`	Yes	Deletes the key rotation for the specified encrypted files.
`dfp file krotate cleanup [-y] [path]`	Yes	Removes the inactive key rotation for the specified FE encrypted files.
`dfp file krotate status`	Yes	Displays the key rotation status for the specified encrypted files.
`dfp file protect -noac -d <data element> <file>`	Yes	Encrypts the specified files.
`dfp file protect -noac [-f] [-r] -d <data element> <folder>`	Yes	Encrypt the specified directories. The following list specifies the command options and their meaning. • *-f* - Forces encryption on the specified protected directory. • *-r* - Enables you encrypt directories recursively. It entails recursive encrypting the directory, all existing and new sub-directories or sub-files in the directory. **Note:** The new sub-directories or sub-files must be created with the required policy loaded in the process. **Note:** If new files and directories are created in delegated environment, then they will be encrypted with the respective policy.
`dfp file protect -d <data element> <file>`	Yes	Protects the specified files.
`dfp file protect [-f] [-r] -d <data element> <folder>`	Yes	Protects the specified directories. The following list specifies the command options and their meaning. • *-f* - Forces access control and encryption on the specified protected directory. • *-r* - Enables you to encrypt and AC protect directories recursively. It involves recursive encrypting the directory, all existing and new sub-directories or sub-files in the directory. **Note:** The new sub-directories or sub-files must be created with the required policy loaded in the process. **Note:** If new files and directories are created in delegated environment, then they will be encrypted with the respective policy.

Commands	*dfpshell* Privilege	Description
`dfp file unprotect <file>`	Yes	Unprotects files. **Note:** You must load the required policy in the process before running this command.
`dfp file unprotect [-r] <folder>`	Yes	Unprotects directories. **Note:** You must load the required policy before running this command. • *-r* - Option lets you unprotect directories and files recursively.
`dfp file stat <path ...>`	Yes	Displays the protection status of files.
`dfp file stat [-r] <folder>`	Yes	Displays the protection status of directories. *-r* - Option shows protected status of the directory and all sub-directories or sub-files recursively.
`dfp file status`	Yes	Displays the active access control list of files and directories and the related data elements for protection.
`dfp file dump <file...>`	Yes	Displays the encryption details of files.
`dfp ac protect[-f] –d <data element> <file>`	Yes	Protects files. *-f* - Option forces protection on the AC protected files.
`dfp ac protect [-f][-r] [-I]–d <data element> <folder>`	Yes	AC protect the directory. The following list specifies the command options and their meaning. • *Without -r and -I*- AC Protect the directory and its files but not sub-directory. • *-r* - AC protect the directory and existing sub-directories recursively, but not newly created sub-directory. • *-I* - Apply inheritance AC protection on the newly created sub-directories as well as the aforementioned scenarios. • *-f* - Forces protection on the specified AC protected directories.
`dfp ac unprotect <file>`	Yes	Unprotect files. **Note:** Ensure that the required data element is loaded to the process.
`dfp ac unprotect [-r] <folder>`	Yes	Unprotect directories. *-r* - Enables you to unprotect directories and existing sub-directories recursively in the directory. **Note:** Ensure that the required policies are added to the process.
`dfp ac cleanup`	Yes	Removes the inactive AC protections from AC protections list.
`dfp ac status`	Yes	Displays information of all the AC protected files and directories.
`dfp ac update`	Yes	Update the access control setting in the *ac.db* file.

Commands	*dfpshell* Privilege	Description
dfp ac import [-f] <imported ac.db file>	Yes	Restores the backed up *ac.db* file.
dfp ac export <exported ac.db file>	Yes	Back up the *ac.db* file.
dfp delegate import [-f] <imported delegate.db file>	Yes	Restores the backed up *delegate.db* file.
dfp delegate export <exported delegate.db file>	Yes	Back up the *delegate.db* file.
dfp ac version	No	Displays the AC version and copyright information about FUSE FP.
dfp ac help	No	Displays the *dfp* AC commands and options for the FUSE FP.
dfp help	No	Displays the *dfp* commands and options of FUSE FP.

8.2 dfp mount Commands

The *dfp mount* command helps to create a FUSE FP mount point, which is required to enforce the FUSE FP file system calls. After the mount point is created, all the contents in the source path are available in the FUSE FP mount point. You can map the source path using the FUSE mount point. You can protect contents of the FUSE FP mount point using the *dfp* file protect commands.

The following figure displays a list of all the *dfp mount* commands of the FUSE FP.

```
[root@labrh71 -]# dfp mount
Usage :
dfp mount <mount point>
dfp mount [--add-entry] [<-o [fuseoptionl] -o [fuseoption2]..>] <mount point>
dfp mount -V
dfp mount all
dfp mount list
dfp mount help
[fuse option]:
    -o allow_other          all user (including root) can access the files
    -o allow_root           allow access to root
    -o default_permissions enable permission checking by kernel
    -o fsname=NAME          set filesystem name
    -o subtype=NAME         set filesystem type
    -o large_read           issue large read requests (2.4 only)
    -o max read=N           set maximum size of read requests

    -o hard_remove          immediate removal (don't hide files)
    -o readdir_ino          try to fill in d_ino in readdir
    -o direct_io            use direct I/O
    -o kernel_cache         cache files in kernel
    -o [no]auto_cache       enable caching based    on modification times (off)
    -o umask=M              set file permissions (octal)
    -o uid=N                set file owner
    -o gid=N                set file group
    -o entry_timeout=T      cache timeout for names (1.0s)
    -o negative_timeout=T   cache timeout for deleted names (0.0s)
    -o attr_timeout=T       cache timeout for attributes (1.0s)
    -o ac_attr_timeout=T    auto cache timeout for attributes (attr_timeout)
    -o intr                 allow requests to be interrupted
    -o intr_signal=NUM      signal to send on interrupt (10)
    -o modules=Ml[:M2...]   names of modules to push onto filesystem stack

    -o max_write=N          set maximum size of write requests
    -o max_readahead=N      set maximum readahead
    -o async_read           perform reads asynchronously (default)
    -o sync_read            perform reads synchronously
    -o atomic_o_trunc       enable atomic open+truncate support
    -o big_writes           enable larger than 4kB writes
    -o no remote lock      disable remote file locking

Module options:
[iconv]
-o from_code=CHARSET       original encoding of file names (default: UTF-8)
-o to_code=CHARSET         new encoding of the file names (default: UTF-8)

[subdir]
-o subdir=DIR              prepend this directory to all paths (mandatory)
-o [no]rellinks            transform absolute symlinks to relative
```

The following table describes all the FUSE FP *dfp* mount commands, their usage, and the *dfpshell* privilege requirement.

> **Note:** Before mounting the FUSE FP mount point, ensure that the FUSE mount point is empty.

> **Note:** Ensure to mount the FUSE FP mount point using the absolute path. If you mount the FUSE FP mount point using the relative path, you will be required to perform the mount operation again after the service restarts.

Table 8-2: FUSE FP dfp mount Commands

Commands	dfpshell Privilege	Description
`dfp mount [--add-entry] [< -o [fuseoption1] -o [fuseoption2]..>] <mount point>`	Yes	Mount the FUSE-based File System (FUSE FS) on the user specified directory. • *<-o>* - option for applying the FUSE option. If there is no option given the FUSE mount with default option. • *[--add-entry]* - option to make the mount point persistent after restarting your system. **Note:** On a single node, only 50 FUSE mount points can be configured by FUSE FP.
`dfp mount -V`	No	Displays the FUSE library version, the FUSE mount version, and the FUSE kernel interface version.
`dfp mount list`	Yes	Displays the status of all the FUSE mount points. **Note:** There are following two states of FUSE FP mount points. • Active • Inactive
`dfp mount all`	Yes	Mounts all the FUSE mount points that are available in the *automount.conf* file.
`dfp umount [--del-entry] <mount point>`	Yes	Remove specified FUSE FP mount point entries from the *automount.conf* file. The removed entries are not mounted after restart.
`dfp umount all`	Yes	Unmount all the FUSE FS mount points.
`dfp mount help`	No	Displays all the FUSE FP *dfp mount* commands and options.

8.3 dfpadmin Commands

The *dfpadmin* commands are used for administrative purposes like database updates and service management.

The following figure displays a list of all the *dfpadmin* commands of FUSE FP.

```
[root@labrh73base ~]# dfpadmin
dfpadmin version 9.1.0.0.0 (Jan 26 2023)
Copyright © 2023 Protegrity Corporation. All Rights Reserved.

Usage:
dfpadmin update
dfpadmin status
dfpadmin service <service name> [ on | off | status ]
dfpadmin service all [ on | off | status ]
dfpadmin modules plug
dfpadmin modules unplug
dfpadmin modules status
dfpadmin database -o updatedb-policy-passwd -p <policy>
dfpadmin license check
dfpadmin license status
dfpadmin help
```

The following table describes the FUSE FP admin commands and the *dfpshell* privilege requirement.

Table 8-3: FUSE FP dfpadmin Commands

Commands	dfpshell Privilege	Description
`dfpadmin update`	Yes	Updates the configuration files if new settings are applied.
`dfpadmin status`	No	Displays the following information: • Product components information • Available policies • FUSE FP license status
`dfpshell dfpadmin status`	Yes	Displays the following information: • Product components information • Available policies • FUSE FP license status • Service information • The file encryption krotate list
`dfpadmin service <service name> [ on | off | status ]`	Yes	Starts or stops the specific service, or displays its status. *<service name>* - Includes pms, log, krotate, rms, and mnts.
`dfpadmin service all [ on | off | status ]`	Yes	• Starts or stops all the FUSE FP services. • Displays the status of all the FUSE FP services. **Note:** `dfpadmin service all off` command stops all the FUSE FP related services. It checks for the active FUSE mount and prompts the user to unmount using the `dfp umount all` command. Run the `dfpadmin`

Commands	dfpshell Privilege	Description
		`service all off` command again to stop all the FUSE FP related services. **Note:** Only the root user can run the `dfpadmin service all` command. If a non-root user run this command, then the following error message appears. `ERROR: permission denied!`
`dfpadmin modules plug`	Yes	Loads the *es_export* and *es* modules into the kernel. **Note:** The *es_export* and *es* modules provide source path protection. **Important:** This command will work only if the *es* modules are enabled. The Enhanced Security (*es*) modules are now delivered as a separate package. For more information about the packages required for specific platforms, contact Protegrity Support.
`dfpadmin modules unplug`	Yes	Unloads the *es_export* and *es* modules from the kernel. **Important:** This command will work only if the *es* modules are enabled. The Enhanced Security (*es*) modules are now delivered as a separate package. For more information about the packages required for specific platforms, contact Protegrity Support.
`dfpadmin modules status`	Yes	Displays the following status information regarding the *es* modules: • Module Name • State • Description • Version • Copyright • Author

Commands	dfpshell Privilege	Description
		- Build Date **Important:** This command will work only if the *es* modules are enabled. The Enhanced Security (*es*) modules are now delivered as a separate package. For more information about the packages required for specific platforms, contact Protegrity Support.
`dfpadmin database -o updatedb-policy-passwd -p <policy>`	Yes	Updates the *delegate.db* file, if you change the policy password and deploy the policy from the ESA.
`dfpadmin license check`	No	Checks the validity of the FUSE FP license.
`dfpadmin license status`	No	Displays the following license status information. - License State - Valid Date - Last Valid Date
`dfpadmin help`	No	Displays the help for FUSE FP *dfpadmin* commands.

Chapter 9

Using the FUSE FP

Supported Features of FUSE FP

The following table describes the list of features supported by FUSE FP:

Table 9-1: List of features

Features	Refer to
File Encryption	*Using File Encryption*.
Delegation	*Using Delegation*.
NFS	*Using Network Shared File System (NFS)*.
CIFS	*Using Common Internet File System (CIFS)*.
Key Rotation	*Using Key Rotation*.
FUSE FP Audit Logging	*Using Audit Logging*.

9.1 Using File Encryption

The file encryption function provides file-level transparent encryption on the sensitive files and directories.

The FUSE FP encrypt files with AES (128-bit and 256-bit) and 3DES encryption algorithms. The file encryption operates on top of the underlying file system to encrypt or decrypt files transparently using the standard ciphers.

The user, program, and process that have the required policy loaded, can access the encrypted files. If the policy is not loaded, then the user, program, and process get a `Permission denied` error message while accessing the encrypted files.

For more information about loading policy, refer to *Deploying Policy*.

The *dfpshell* privilege is required to execute the encryption and decryption commands.

9.1.1 Encryption and Decryption Commands

The FUSE FP encrypts and decrypts the files and directories with the specified data element contained in the policy.

You can run the following command to encrypt your sensitive files.

```
dfp file protect -noac -d <data element> <file>
```

Where, *-d <data element>* provides the data element name.

> **Note:**
> Before you run the file protection command, ensure that you mount the files in the FUSE FP mount point.
>
> For more information about the mount point, refer to *Concept of FUSE Mount Point*.
>
> For more information about mounting the files inside the mount point, refer to *Encrypting a File*.

The FUSE FP can decrypts the file and makes it available in cleartext format after you run the file unprotect command.

Run the following command to decrypt the files.

```
dfp file unprotect <file>
```

The encryption or decryption operations are supported only for regular files and must not be used to encrypt or decrypt system files, block devices, character device file, and soft link files.

File encryption provides options for encrypting and decrypting your sensitive directories and their child files and child directories in real time. The `file protect` command encrypts a directory and its child files and the `file unprotect` command decrypts the directory contents.

To encrypt the directories, run the following command.

```
dfp file protect -noac [-r] -d <data element> <folder>
```

Where:

-d <data element> provides the data element name.

-r used for recursively FE encrypting directory and all the sub-directories or sub-files in the directory.

To decrypt the directories, run the following command.

```
dfp file unprotect [-r] <folder>
```

Where:

-r used for recursively FE decrypting directory and all the sub-directories or sub-files in the directory.

9.1.2 Reviewing Encryption Status

The FUSE FP encrypts files in the file system, and makes them accessible for users who have the loaded policy.

➤ To view the encryption status of files or directories:

Run the following command.

```
dfp file stat [-r] <path ...>
```

Where:

-r used recursively for FE encrypting directory and all the sub-directories or sub-files in the directory.

If you run this command on a file or directory that is not encrypted, then the following message appears.

```
INFO: <path ...>: not protected.
```

If you run this command on a file or directory that is encrypted, then the following message appears.

```
INFO: <path ...>: protected by <dataelement name> (encryption).
```

9.1.3 Reviewing Encrypted File Status

The FUSE FP provides the *file dump* command to get detailed information about the encrypted files and directories.

➤ To view the detailed information about the encrypted files and directories:

1. Run the following command.

   ```
   dfp file dump <file ...>
   ```

2. Enter the *dfpshell* password.

   The following figure displays the result of the file dump command.

```
[root@labcos64-64 fuseprotect]# dfp file dump x.txt
Enter dfpshell Pass Phrase: *
               File Name: /root/fuseprotect/x.txt
               File Type: File Protector Encrypted File Format
            File Version: 10
               File Size: 9217
             Header Size: 720
             Ext Address: 0
                Ext Size: 0
            Data Address: 8192
               Data Size: 6
            Padding Size: 1019
         Encryption Time: 2017-09-22 00:59:33
       Cipher Block Size: 1024
                   Flags: 1
       Data Element Name: rcwd Data Element
                  Key ID: 2
```

9.1.4 Encrypting a File

You use the `file protect` command to encrypt a sensitive file.

To encrypt files:

1. Run the following command to load your policy.
   ```
   dfp start -p <policy>
   ```
2. Create a sample text file using a text editor.
3. Save the file and exit from the editor.
4. Run the following command to mount the files inside the mount point.
   ```
   dfp mount -o modules=subdir -o subdir=<source path> <mount point>
   ```
5. Run the following command to protect files.
   ```
   dfp file protect -noac -d <data element> <file>
   ```
6. Run the following command to verify the protected file status.
   ```
   dfp file stat <file>
   ```
7. View the *dfptest.txt* file in any editor.

   If you view the file inside the mount point, then you can see the file in cleartext format, provided you have the necessary policy loaded. As the mount point is protected, any files or directories inside the protected path must be encrypted.

   > **Note:**
   > The following message occurs if an undelegated program, process, or user accesses the protected path
   >
   > ```
   > Permission denied
   > ```

8. Run the following command to access a new shell without any loaded policy.
   ```
   dfp start -n $SHELL
   ```
9. Run the following command to ensure that no data elements are loaded.
   ```
   dfp proc -l
   ```
10. View the file in any editor.

    If the policy is not loaded or if files are accessed from outside the mount point, the `Permission denied` error appears.

9.1.5 Decrypting a File

To decrypt files:

Run the following command.

```
dfp file unprotect <file>
```

After the files are decrypted, you can view the file content in cleartext format.

9.1.6 FUSE FP Encryption Output Settings

The output settings for the FUSE FP encryption are available for FE encrypted or AC and FE encrypted files or directories. The output settings are used to specify how you can access the content of files that are FE encrypted or AC and FE encrypted, according to the access mask of the data element. You can specify the output setting of a data element in ESA by selecting either Exception or Protected Value.

Specify the following output setting:

1. If you select the data element permission as *Read*, then the output setting must be set to *NULL* value.
2. If you select the data element permission as *Write*, then the output setting must be set to *Exception* value.
3. If you select the data element permission as *Create* or *Delete*, then you can select either the *Exception* or *Cipher* value in the output setting.

The following table describes different scenarios of return values based on the output settings.

Table 9-2: Different Scenarios of Output Returns based on Output Settings

Data Element Permission				**Specified Output Setting**	**Desired Output on Read When the Policy is Loaded**
Read	**Write**	**Create**	**Delete**		
Yes	Any	Any	Any	NULL	Cleartext
No	Yes	Any	Any	Exception	Permission Denied
No	No	Yes	Any	Cipher	Ciphertext
No	No	Yes	Any	Exception	Permission Denied
No	No	Any	Yes	Cipher	Ciphertext
No	No	Any	Yes	Exception	Permission Denied

9.2 Using Access Control

The FUSE FP provides the access control feature to protect files and directories. This feature prevents unauthorized access of sensitive files and directories. The file content is visible to authorized users who load the proper data element of the policy in a process (bash shell), program, and user.

Using the FUSE FP, you can use the access control features on the files and directories in the following ways:

- If you want to apply the AC protection on files and directories, then you must run the `dfp ac protect [-f] [-r] -d <data element> <folder>` command.
- If you want to remove the AC protection from files and directories, then you must run the `dfp ac unprotect [-r] <folder>` command.
- If you want to apply the AC and FE protection (file protection) on files and directories, then you must run the `dfp file protect [-noac] [-f] [-r] -d <data element> <folder>` command.
- If you want to remove the AC and FE protection (file protection) from files and directories, then you must run the `dfp file unprotect [-r] <folder>` command.

> **Note:** The *dfpshell* privilege is necessary to execute the AC and file protection commands.

9.2.1 AC Behavior

If the access control feature is implemented, then it can be enforced on the two points, such as FUSE mount path and source path.

The following behavior of AC is observed for FUSE mount path and source path:

Table 9-3: AC Behavior on the Source and Mount Path

ESLinux Module Status	**AC Behavior on the Source Path**	**AC Behavior on the FUSE Mount Path**
Enable	Any delegated or undelegated program, user, or process tries to access the AC protected	Any delegated program, user, or process can access the AC protected files and directories on the FUSE mount path based on the access mask of data elements.

ESLinux Module Status	AC Behavior on the Source Path	AC Behavior on the FUSE Mount Path
	files and directories on the source path, then the following error message appears. `Permission Denied`	
Disable	Any delegated or undelegated program, user, or process can access the AC protected files and directories on the source path.	• Any delegated program, user, or process can access the AC protected files and directories on the FUSE mount path. • If any undelegated program, user, or process tries to access the AC protected files and directories on the mount path, then the following error message appears. `Permission Denied`

> **Note:** The AC behavior on the source path does not depend upon the access mask of data elements and file system permissions, when the *es* module is enabled.

> **Note:** The AC behavior on the FUSE mount path varies depending upon the access mask of data elements and file system permissions.

The following diagram illustrates the AC behavior with ESLinux module enabled and policy loaded.

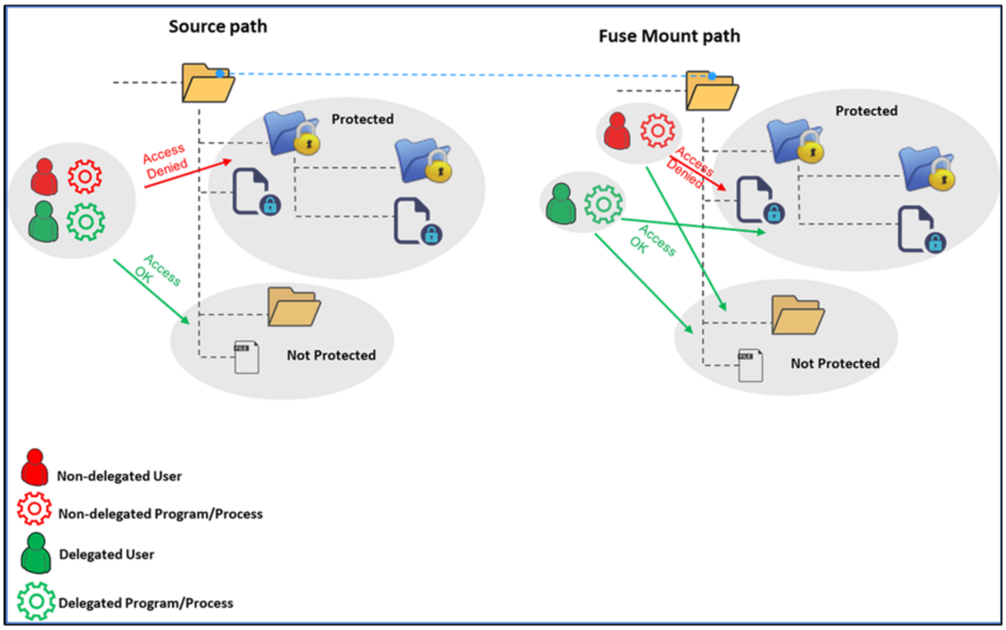

Figure 9-1: AC behavior with ESLinux Module Enabled and Policy Loaded

9.2.2 Loading a Policy in the FUSE FP

After you have selected a policy, you can load the policy in the process (bash shell).

Before you begin
You must first start the *dfpshell* to protect and unprotect files and directories, and then select a data element from the loaded policy list in the process.

➤ To load a policy in process:

1. Run the following command to access your policy.

   ```
   dfp start –p <policy>
   ```

   You must enter the configured policy password. The bash shell displays your FUSE FP privileges, and the loaded data elements. You can protect the files and directories with the required data element.

2. Run the following command to view the list of policies loaded in the process.

   ```
   dfp info
   ```

   This command provides the following FUSE FP details:

   - Product information
   - Available policy list and loaded data elements in the process

9.2.3 Protecting a File

You use the `ac protect` command to protect a sensitive file.

> **Note:** You should create a text file as a sample file to protect.

To protect a file:

1. Run the following command to access your policy.

   ```
   dfp start –p policy
   ```

2. Create a sample text file using a text editor.
3. Save the file and exit from the editor.
4. Run the following command to protect the file.

   ```
   dfp ac protect –d <data element> <file name>
   ```

5. View the protected file in any editor.

   You can access the protected files based upon usage of the data element of an access mask .

6. Run the following command to access a new shell with no data elements.

   ```
   dfp start -n $SHELL
   ```

7. Run the following command to ensure that no data elements are loaded in the process.

   ```
   dfp info
   ```

8. View the file in any editor.

   As no data elements are loaded in the process, the following error message appears.

   ```
   Permission Denied
   ```

9.2.4 Unprotecting a File

You can use the `ac unprotect` command to unprotect a sensitive file.

To unprotect a file:

Run the following command to unprotect a protected file.

```
dfp ac unprotect <file name>
```

After unprotecting a file, you can access that file based on the *Operating System(OS)* permissions.

9.2.5 Protecting a Directory

You can use the `ac protect` command to protect a sensitive directory.

To protect a directory:

Run the following command to protect a directory.

```
dfp ac protect [-f] [-I] [-r] -d <data element> <folder>
```

Table 9-4: Command options and their meaning

Command Options	Meaning
-d <data element>	Provides the <data element name>.
<folder>	Provides the directory name.
-I	Applies inheritance protection on the existing and newly created files, directories, and sub-directories.
-r	Applies the AC protect operation recursively to all the existing files, sub-directories, and newly created files inside the protected directories. **Note:** The protection rule does not protect newly created directories.
-f	Applies the force protect operation on the existing and new AC protected directories and files depending on the usage of the *-f* command option with the *-r* or *-I* command option.

> **Note:** The new child files and directories must be created with the required policy key loaded in the process.

> **Note:**
>
> If a directory is AC protected, then any file inside that protected directory can not be specifically reprotected with a different DE.
>
> To protect the parent and child directories with different DE, perform the following steps:
>
> 1. Protect child file1 with DE1, file2 with DE2 etc.
> 2. Protect the parent directory with DE0.
>
> If you protect the child file and parent directory in the above mentioned way, then the entries of the files as well as the directory are created in the *ac.db* file and any file that was protected earlier can be reprotected with a different data element.
>
> The access permissions of the protected files depend upon the individual DE with which they are protected.

> **Note:**
>
> If you unprotect any of the protected files inside the protected directory, then it inherits the protection from its parent directory. Once the child file is unprotected, it cannot be protected again as it will now inherit the protection from its parent directory.

9.2.6 Unprotecting a Directory

➤ To unprotect a directory:

Run the following command.

```
dfp ac unprotect [-r] <folder>
```

The FUSE FP unprotects the directory, subdirectories, files and enables the user with appropriate OS permissions to access the contents of the directory.

The unprotect operation without the *-r* option will only unprotects the specified directory and its child files. To apply the unprotect operation recursively to all the child directories within the specified directory, use the *-r* option.

> **Caution:**
> It is recommended that you should not unprotect a file or subdirectory in an inherited protected directory.
>
> If you still want to perform the above unprotection, then refer to *Unprotecting a File or Subdirectory in an Inherited Protected Directory*.
>
> If you try to unprotect a subdirectory in an inherited protected directory, then the following error message appears.
>
> ```
> INFO: the path <inherited protected directory path> been inherited protected and couldn't be unprotected!
> ```
>
> - The inheritable protected file and subdirectory cannot be accessed without a data element.
> - Each file or subdirectory in an inherited protected directory gets the inherent protection.
> - Each file or subdirectory in an inherited protected directory cannot be reprotected with a different data element.

9.2.6.1 Unprotecting a File or Subdirectory in an Inherited Protected Directory

The following section describes how the user can unprotect a file or subdirectory in an inherited protected directory.

➤ To unprotect a file or subdirectory in an inherited AC protected directory:

1. Add the **source path** of a file or subdirectory in the *ac_disallow.conf* file.

   For example:

   - **Source Path:** `/home/source`
   - **Mount Path:** `/home/mount`
   - **Directory Structure:** `parent/child/sub-child/file`
   - **Inherit protect:** *parent*
   - **Exclude:** *parent/child/*

   ```
   echo "/home/source/parent/child/" >> /opt/protegrity/fileprotector/fuse/data/ac_disallow.conf
   ```
2. Apply inherit protection on the parent directory using the following command.

`dfp ac protect -I -d <data element> parent`

```
INFO: Recursively protecting path </home/source/parent> by data element <rcwd> is
successful!
ERROR: path </home/source/parent/child> is disallowed to ac protect!
ERROR: path </home/source/parent/child/sub-child> is disallowed to ac protect!
```

> **Note:**
>
> Alternatively, you can first apply the inherit protection on the parent directory, then add a subdirectory in the *ac_disallow.conf* file, and run the `dfp ac protect [-f] [-I] -d <data element> <folder>` command.

> **Note:**
>
> If you remove a subdirectory path from the *ac_disallow.conf* file, then the subdirectory path remains unprotected. But, the status of the protected file or directory will appear as protected using `dfp ac stat <file or folder>` command.
>
> To resolve this issue, run the `dfp ac protect [-f] [-I] -d <data element> <folder>` command.

9.2.7 Reviewing Protection Status

You can check the protection status of any file and directory.

➤ To review the protection status:

1. Run the following command to view the protection status of files and directories.
   `dfp ac stat <filename>`
   - If a file or directory has not been protected, then the following message appears.

     ```
     Not Protected
     ```

   - If a file or directory is protected, then the following message appears.

     ```
     Protected by <data element>
     ```

2. Run the following command to view the current protection status.
   `dfp ac status`
   This command displays the following results:
   - Current protection list
   - Protection status
   - Related data element

   The following snippet describes the sample result of the `dfp ac status` command.

   ```
   Access Control List:
   ACTIVE  1: /tmp/ls <protected by d1>
   ACTIVE  2: /tmp/111 <protected by d1>
   ```

```
ACTIVE  3: /mnt/dmloop/1 <protected by d1>
ACTIVE  4: /lib/1 <protected by d1>
```

> **Note:** The access control list is based on the source path.

9.2.8 Cleaning up Inactive AC Protection List

If you want to clean the inactive AC protections in the status list, then you can execute the *clean up* command.

To clean up invalid AC protections:

Run the following command.

```
dfp ac cleanup
```

This command only cleans up the invalid AC protections, such as *INACTIVE* AC protections list and the files or directories that have non-existent AC protections.

The following snippet describes the sample result of the `dfp ac cleanup` command.

```
[root@labrh7 mount]# dfp ac cleanup
Enter dfpshell Pass Phrase: *
INFO: Acess Control Cache Cleaned Up Successfully !!
```

9.2.9 Protecting and Encrypting a File

Before you begin
Prerequisite:

The protection and encryption data element must be same.

To encrypt and protect a file:

1. Run the following command to encrypt the file.

   ```
   dfp file protect -d <data element> <file>
   ```

2. View the protected file in a text editor.

   If the data elements are loaded in the process and you have all the read, write, and delete permissions, then you can read, write, and delete the protected file.

3. Run the following command to access a new shell with no data elements.

   ```
   dfp start -n $SHELL
   ```

4. Open the file in any editor and the following message appears.

   ```
   Permission denied
   ```

9.2.10 Protecting and Encrypting a Directory

Before you begin
Prerequisite

The protection and encryption data element must be same.

➤ To encrypt and protect a directory:

1. Run the following command to protect and encrypt a directory.
   ```
   dfp file protect [-f] [-r] -d <data element> <folder>
   ```
2. Create files and directories within the protected directory.
3. You can edit and delete the files and directories as per the access mask permission of the data element.
4. Run the following command to access a new shell with no data elements.
   ```
   dfp start -n $SHELL
   ```
5. If you try to review, create, and write operation for the protected directory, then the following message appears.
   ```
   Permission denied
   ```

9.3 Using Delegation

The ability to authorize programs for a given policy is known as delegation. Using FUSE FP, you can grant access to the protected files by loading the required policy. When a process or program is delegated, it can perform security operations on the data without loading the policy, as it loads the policy automatically.

9.3.1 Program Delegation

When a delegated program starts, the policy becomes available for the executable process. Any child process, which is created by a delegated process, inherits policies from the parent process.

After delegation, whenever the program starts, it automatically accesses the data elements. The only access restrictions are the standard system permissions.

> **Note:** The program delegation uses absolute path of the binaries. If you change the absolute path for any delegated program, then the program delegation is lost and you must configure the new path.

9.3.2 Process Delegation

The running processes can be delegated with data elements from a specific policy, so that it can access protected and encrypted directories or files. This type of arrangement is useful for running applications, or services.

> **Note:** The delegated running process ID (PID) loses its access to the data elements when the processes end.

9.3.3 User Delegation

The *OS* users can be delegated with data elements from a specific policy, so that the newly created processes of the *OS* users can access protected and encrypted directories or files. This type of arrangement is useful for the specific *OS* user to get permissions such as when the *OS* user logs into the system, start applications or services, which the *OS* user is configured to run.

When you are delegating the *OS* user, a warning message appears that prompts you to continue or abort the delegation. The warning message states the risk that after you delegate the *OS* user, for example, the administrator users such as root or administrator can run a program or process via the delegated user to get the delegated policy permissions.

The following snippet displays a warning message related to the delegation of the *OS* user.

```
[root@rhel74base ~]# dfp delegate -f -u root fuse_fe
Enter dfpshell Pass Phrase: *
WARNING: once you delegate the user, Administrator users like administrator or root would
be able to execute a program/process via the delegated user to get the delegated policy
permissions.
INFO: Aware of the risk, do you still want to continue? [Y|N] (Y)
y
Enter the policy password : *
INFO: Delegate user <root> by policy <fuse_fe> successfully!
```

If the OS user switches to another user’s session using the Linux command `su`, then the following scenarios can occur:

Condition	Scenario	Result
User1 is delegated	The User1 switches to the User2’s session.	The policy of the User1 will not be transferred to the User2 regardless of delegation status of User2.
User2 is delegated	The User1 switches to the User2’s session.	If User2 is delegated, then any user, regardless of its delegation status, can access the policy of User2 after logging in to it .
User2 is delegated	The User1 either load *dfpshell* or policy using the `dfp start -p <policy>` command and switches to the User2’s session.	The policy of both User1 and User2 will be available in the session. The policy of the User2 will be only available for system commands, but not for the *dfp* commands. To drop policy of the User1 completely, the User1 should unload the current policy and exit the *dfpshell* if loaded, and then switch to the User2.

Note:

After switching to a new user, the *dfpshell* privilege will not be transferred to the next user irrespective of the delegation status of the user.

9.3.4 Delegating a Process

You can delegate a process using the <PID> values of the running process and associating a policy with it for enforcing a delegation.

To delegate a process:

Run the following command to enforce delegation for the process referred by *<PID>*.

```
dfp delegate [-f] [-r] -p <PID> <role>@<policy>
```

where:

- *-f* - Provides force delegation of the policy to the running PID, if the process has already been delegated.
- *-r* - Provides recursive delegation of child processes.
- *-p* - Specifies the PID that need to be delegated.

9.3.5 Undelegating a Process

You can undelegate a process using <PID> values of the running process.

➤ To remove the policy from the process:

Run the following command.

```
dfp undelegate [-r] -p <PID>
```

9.3.6 Delegating a User

You can delegate the system users using the `dfp delegate` command. Policy is loaded automatically for delegated users upon login.

➤ To delegate a user:

Run the following command.

```
dfp delegate [-f] -u <username> <role>@<policy>
```

-f - Forces delegation of the policy to the user if the user is already delegated. After you delegate the user, the newly created processes for the user include the specified policy.

-u - Specifies the username that needs to be delegated.

> **Note:** The user needs to re-login after executing the `dfp delegate` command, as data elements are loaded only upon re-login.

> **Note:** If you are delegating the system user, then the user must exist.

9.3.7 Undelegating a User

You can undelegate a previously delegated user using the *dfp undelegate* command.

➤ To undelegate a user:

Run the following command.

```
dfp undelegate -u <username>
```

After undelegating the user, if you do not terminate or close the previous process, then all the previous processes linked to the user will continue to retain the delegated data elements. For example, a delegated user logs on to a machine where the current processes have the delegated data element. If you undelegate the user without terminating or closing the current login, then the current login session continues to remain delegated.

9.3.8 Delegating Program

Application binaries or executables can be delegated using the `dfp delegate` command. Policy is loaded automatically for delegated programs upon execution.

➤ To delegate a program:

Run the following command to enforce delegation.

```
dfp delegate [-f] -e <program> <policy>
```

where:

- *-f* provides force delegation of the policy to the running process ID if the process has already been delegated.
- *-e* specifies the program that needs to be delegated.

> **Note:** Programs need to be restarted after executing the `dfp delegate` command.

9.3.9 Undelegating Program

You can undelegate a previously delegated program using the *dfp undelegate* command.

➤ To undelegate a program:

Run the following command.

```
dfp undelegate -e <program>
```

9.3.10 Reviewing the Delegation Status

The *delegation status* command lists the active delegated program, user, and process and specifies the policy in the delegated list.

➤ To view the delegation status:

1. Run the following command.

   ```
   dfp delegate status
   ```

2. Enter the *dfpshell* password.

The following snippet displays the result of the *delegation status* command.

```
[root@labrh7 -]# dfp delegate status
Enter dfpshell Pass Phrase: *
Delegated Program List:
ACTIVE 1: /usr/bin/cat <policyFuse>
Delegated User List: empty
```

> **Note:**
> If any program, user, or process is not delegated, then the `empty` status message is displayed.
>
> The *dfpshell* privilege is required for executing this command.
>
> ```
> [root@labcos64-64 fuseprotect]# dfp delegate status
> Enter dfpshell Pass Phrase: *
> Delegated Program List: empty
> Delegated User List: empty
> ```

9.3.11 Removing Invalid Delegation

The `dfp delegate cleanup` command removes only the invalid delegations, such as the inactive delegations list and the files or directories with delegations that are non-existent.

To clean up the invalid delegation:

Run the following command.

`dfp delegate cleanup`

The following figure displays the result of the *delegation cleanup* command.

```
[root@labcos64-64 -]# dfp delegate cleanup
Enter dfpshell Pass Phrase: *
INFO: Delegation Cache Cleaned Up Successfully !!
```

9.3.12 Script Delegation

The script delegation is a feature that offers delegation of scripts (bash, python, perl) using the wrapper mechanism.

> **Caution:** Before delegating any script, ensure that the script has execution permission.

Run the following command to delegate any script.

`dfp delegate [-f] [-s] -e <program> <policy>`

After delegation, a wrapper for the script is created in the FUSE FP Installation path. This wrapper acts as a link or pointer to the script. Whenever the script is triggered, the wrapper is executed and it transfers the control to the script.

> **Note:**

> If the execution permission of the script is removed after delegation, whenever the user try to run the wrapper, then the following message appears:
>
> ```
> [root@labrh7u2bas-0 -]# dfp delegate -s -e temp.sh policy_fe
> Enter dfpshell Pass Phrase: *
> INFO: Delegate program </root/temp.sh> by policy <policy_fe> successfully!
> [root@labrh7u2bas-0 ~]# chmod 644 temp.sh
> [root@labrh7u2bas-0 ~]# Is -lai temp.sh
> 80960246 -rw-r--r-- l root root 130 Dec 18 02:13 temp.sh
> [root@labrh7u2bas-0 ~]# /opt/protegrity/fileprotector/fuse/wrappers/temp.sh
> ERROR: Please make sure you have execute permissions on script Operation not permitted
> ```
>
> However, if the user runs the **`dfp delegate status`** command, then the delegated script status still appears as `ACTIVE`.

9.3.12.1 Delegating Script

➤ To delegate a script:

1. Identify the script that needs to be delegated.
   For example, `/usr/sbin/my_script.sh`
2. Provide execute permission to the script.

   ```
   chmod 111 my_script.sh
   ```
3. Run the **`dfp delegate`** command to delegate the script.
   `dfp delegate [-f] [-s] -e <program> <policy> [<password>]`
4. Enter the *dfpshell* and policy password.

   After successful execution of the **`dfp delegate`** command, the following wrapper is created in the FUSE installer path.

   ```
   INFO: Delegate program </root/fuseprotect/my_script.sh> by policy <fuse_fe> successfully
   [root@labcos64-64 fuseprotect]# cd /opt/protegrity/fileprotector/fuse/wrappers/
   [root@labcos64-64 wrappers]# ls
   ```
5. Run the following command to check the delegation status.
   `dfp delegate status`

   ```
   [root@labcos64-64 fuseprotect]# dfp delegate -s -e my_script.sh fuse_fe
   Enter dfpshell Pass Phrase: *
   Enter the policy password : *
   INFO: Delegate program </root/fuseprotect/my_script.sh> by policy <fuse_fe> successfully
   [root@labcos64-64 fuseprotect]# dfp delegate status
   Enter dfpshell Pass Phrase: *
   ```
6. Enter the *dfpshell* password.

   The following active delegated program list appears.

   ```
   Delegated Program List:
   ACTIVE 1: /root/fuseprotect/my_script.sh <fuse_fe>
   Delegated User List: empty
   ```

Whenever the user triggers the script *my_script.sh*, the script is delegated and the user can access the protected path.

> **Note:**

> The script delegation relies on the Linux *file* command to identify whether the filename provided to the *dfp* command is a script.
>
> If the filename provided is not identified as a script, then the script is delegated successfully and the wrapper execution of that particular script provides the following error:
>
> *`ERROR:Please make sure script has a valid shebang/interpreter specified !!`*
>
> To add the *#!<interpreter>*, perform the following steps:
>
> i. Add the *#!<interpreter>* as the first line of the script.
>
>   For example,
>
>   - For bash scripts - *#!/usr/bin/bash*
>   - For shell scripts - *#!/usr/bin/sh*
>   - For perl scripts - *#!/usr/bin/perl*
>   - For python scripts - *#!/usr/local/bin/python*
>
> ii. Check if the executable permissions are assigned to the script.

9.3.12.2 Undelegating Script

To undelegate a script:

Run the following command to undelegate a previously delegated script.

`dfp undelegate -e <program>`

The following snippet describes the example of the **`undelegate`** command.

```
[root@labcos64-64 testing]# dfp undelegate -e my_script.sh
INFO: Undelegate program </root/testing/my_script.sh> successfully!
```

After successful execution of the **`dfp undelegate`** command, the following wrapper is removed from the FUSE installer path.

`/opt/protegrity/fileprotector/fuse/wrappers`

```
[ root@labcos64-64 testing]# dfp undelegate -e my_script.sh
Enter dfpshell Pass Phrase: *
INFO: Undelegate program </root/testing/my_script.sh> successfully!
[ root@labcos64-64 testing]#
[ root@labcos64-64 testing]# cd /opt/protegrity/fileprotector/fuse/wrappers/
[ root@labcos64-64 wrappers]# ls
[ root@labcos64-64 wrappers]#
[ root@labcos64-64 wrappers]#
```

9.3.12.3 Script Management Cases

As the original script and the wrapper binary are located in separate paths, the following actions are required to keep the wrapper binary updated. The following table describes various script management cases for updating the wrapper binary.

Table 9-5: Script Management Cases

Action on Original Script	**Wrapper Status** **< dfp delegate status>**	**User Action Required**	**Delegated Binary Status**
Replace	INACTIVE	Run the `dfp delegate -f` command.	Previous instance of the delegated script remains delegated. If no user action is taken, then the following message appears for the failure of new instance of the delegated script. `Integrity Check Fail`.
Rename	INACTIVE	Run `dfp delegate cleanup` and `dfp delegate -s -e <rename script name>` command.	Previous instance of delegated script remains delegated. If no user action is taken, then the following message appears for the failure of new instance of the delegated script. `No such File or Directory`.
Copy	ACTIVE		If the user runs the copied script, then it is not being delegated because the wrapper points to the source script. If the user runs the wrapper, then it runs the original script.
Delete	INACTIVE	Run the `dfp delegate cleanup` command.	
Move to different path	INACTIVE	Run the `dfp delegate cleanup` and `dfp delegate -s -e <move script name>` commands.	Previous instance of delegated script remains delegated. If no user action is taken, then the following message appears for the failure of new instance of the delegated script. `No such File or Directory`.

9.4 Using Network Shared File System (NFS)

The FUSE FP system can be configured to protect the shared files. The mechanism to configure NFS requires a combination of file encryption and delegation setup.

For more information about NFS setup, refer to *Use Cases for the FUSE FP*.

9.5 Using Common Internet File System (CIFS)

The FUSE FP system can be configured to protect the Common Internet File System (CIFS). The mechanism to configure CIFS requires a combination of file encryption and delegation setup.

For more information about CIFS setup, refer to *FUSE FP for the Common Internet File System (CIFS)*.

9.6 Using Key Rotation

The FUSE FP encryption provides the key rotation functionality, which automatically replaces the encryption key for the specified encrypted files. Key rotation re-encrypts the encrypted files with the new active key in the same data element.

You can specify the encrypted files or directories for key rotation and configure the time interval after which the key must be rotated.

9.6.1 Key Rotation Status

The FUSE FP file encryption function provides the `dfp file krotate` commands to add, delete, and display configuration to rotate the key. The *dfpshell* privilege is required to run these commands.

The following table describes different statuses of Key Rotation:

Status	Description
Valid	The encryption key is valid and rotation is not required.
Invalid	The status lists Invalid for the encrypted file or directory in the following cases: • The user specifies the key rotation configuration for an encrypted file or directory, and then tries to force encryption on the encrypted file or directory with another data element. • The user decrypts the encrypted file or directory, and then these encrypted files or directories were deleted.
Expired	The encryption key has expired and rotation is required.
Rotating	The file is processing key rotation.

9.6.2 Adding the Key Rotation Configuration

You can run the `dfp file krotate add` command to add the key rotation configuration for the new and already specified encrypted files or directories.

➤ To add key rotation configuration for encrypted files:

Run the following command.

```
dfp file krotate add [-f] [-r] [-p <policy> <-|passwd>] <path ...>
```

Where:

- *-f* - Adds new key rotation configuration for the new and already specified encrypted files or directories (optional).
- *-r* - Adds key rotation configuration recursively for the specified encrypted directories (optional).
- *-p <policy> <-|passwd>* - Specifies the policy name, policy password (optional), and use *"-"* to prompt for the policy password.

- *<path ...>* - Specifies the encrypted files or directories path. You can only specify the key rotation for encrypted files or directories.

You can either specify the policy and password or only policy (password is prompted). If you run the command without specifying the policy, then it prompts for the policy and password.

9.6.3 Deleting the Key Rotation Configuration

You can run the `dfp file krotate del` command to delete the key rotation configuration for the specified encrypted files.

➤ To delete the key rotation configuration for the specified encrypted files:

Run the following command.

```
dfp file krotate del [-r] <path ...>
```

This command deletes the key rotation configuration and displays the following information about the key rotation configuration for the specified encrypted files or directories:

- Status
- Index
- Path of the specified encrypted files or directories
- Encrypted data element (policy)
- Encrypted key ID
- Last rotate date

Where:

- *-r* - recursively deletes key rotation configuration for the specified encrypted directories (optional).

9.6.4 Removing Invalid Key Rotations

The user must run the `dfp file krotate cleanup` command to remove all the invalid key rotations, such as invalid key rotations list and the FE encrypted files or directories with key rotations that are not present.

➤ To remove invalid key rotation:

Run the following command.

```
dfp file krotate cleanup [-y]
```

Where:

-y - Applies *yes* for all interactive questions during cleaning up the key rotations.

> **Note:** If any command option is not provided, then this command provides a prompt to remove all the invalid key rotation.

9.6.5 Displaying the Key Rotation Status

The user must run the `dfp file krotate status` command to display the key rotation status for the encrypted files and directories.

To display the key rotation status for the specified encrypted files:

Run the following command.

```
dfp file krotate status
```

This command displays the following key rotation status information for the encrypted files or directories:

- Status
- Index
- Path
- DE (Policy)
- Encrypted KeyID
- Last Rotate Date

The following snippet describes the FUSE FP file key rotate Status command result.

```
   Status  Index     Path        DE(Policy)    Encrypted-KeyID
                                      [LastRotateDate]
    -----------------------------------------------------------------------------------
    [Valid]  1. /home/enc_folder/subfile   de1(policy1)   <1>   [-]
    [Valid]  2. /home/enc_folder/subfolder  de1(policy1)   <1>   [-]
    [Valid]  3. /home/rchen/krotate/fe_file2 de1(policy1)   <1>  [2013-01-16 14:51:19]
    [Expired] 4. /home/enc_folder       de3(policy2)   <16>   [-]
    [Rotating]6. /home/enc_bigfile      de3(policy2)   <0>   [-]
    [Invalid] 7. /home/enc_folder/subfile2  de1(policy1)   <0>   [-]
```

9.6.6 Configuring Key Rotation

This section enables you to configure key rotation. You can select to rotate the keys at a specific time interval.

Before you begin
For example, consider that you have a mount point encrypted by data element *de1* in the policy1. For this mount, you want to configure the key rotation to occur at 9 am every day. If the encrypted file is busy, then you can attempt the key rotation four times every 50 seconds.

To configure Key Rotation:

1. Run the following command to add key rotation for the encrypted files or directories.

   ```
   dfp file krotate add [-f] [-r] -p policy1 - /<source path or fuse mount point>
   ```

2. Edit the *key_rotation.conf* file to add the following configuration. This configuration file follows the Linux crontab format:

   ```
   00 09 * * *
   key_rotate_retry_times=4
   key_rotate_retry_interval=50
   ```

3. Run the following command to ensure that the key rotation service is running.

   ```
   dfpadmin service all status
   ```

4. Run the following command to verify the configuration status for the key rotation.

   ```
   dfp file krotate status
   ```

5. Change the key of `de1` in ESA and again deploy the policy to the FUSE FP.
6. At 9:00 am, run the following command to view the changed key rotation status of the encrypted files or directories.

   ```
   dfp file krotate status
   ```

9.7 Using Audit Logging

The FUSE FP monitors the security operations and logs in an audit log.

An audit log is triggered when you perform the following tasks:

- Encrypt and decrypt files and directories
- Read or write FE encrypted files
- Delegate or undelegate a program
- Delete an encrypted files or directories
- Load a policy
- Enter a privileged shell
- Add or Delete key rotation configuration
- Install, uninstall, or upgrade FUSE FP
- Add or Delete key rotation configuration

An audit logging operation includes success, failure, auditing, mount, and unmount operations.

The following events are generated for the audit configuration:

Events	Description
OPEN_W	Open FE encrypted and AC protected file for writing.
OPEN_R	Open FE encrypted and AC protected file for reading.
FE_PROTECT	FE encrypt file or directory.
FE_UNPROTECT	FE decrypt file or directory.
AC_PROTECT	AC protect file or directory.
AC_UNPROTECT	AC unprotect file or directory.
CDEL	Delegate a program.
UDEL	Undelegate a program.
RMVF	Remove FE encrypted file or directory.
USDEL	Delegate the user.
USUDEL	Undelegate the user.
MOUNT	Mount the FUSE FP mount point.
UMOUNT	Unmount the FUSE FP mount point.
LOAD_POLICY	Load the policy for the user.
DFPSHELL	Login the *dfpshell* and change the *dfpshell* password operation.
KEY_ROTATION_ADD	Add the key rotation.
KEY_ROTATION_DEL	Delete the key rotation.
UPDATE	Update or upgrade FUSE FP.
UNINSTALL	Uninstall FUSE FP.

9.7.1 Error and Drop Mode Configuration

The logging section of the *pepserver.cfg* file is used to configure the Log Server service-related settings.

The following snippet displays a sample of the logging configuration settings. The usage of the *pepserver.cfg* file is described in the comments within the *pepserver.cfg* file.

```
# ---------------------------------
# Logging configuration
# ---------------------------------
# In case that connection to the fluentbit is lost, set how logs must be handled.
# This setting is only for the protector logs and not application logs, sent from pepserver#
drop = (default) Protector throws logs away if connection to the fluentbit is lost
# error = Protector returns error without protecting/unprotecting data if connection to the
fluentbit is lost
#mode = drop
```

Here, the *mode* denotes how the logs will be handled under error conditions.

> **Note:**
>
> After setting the mode, to update the configuration settings, execute the **`dfpadmin update`** command.

- Drop Mode: Drop the logs if connection between Log Forwarder and Log Server breaks.
- Error Mode: Save logs in "/var/protegrity/dfperrorlogd/errorlog" when connection between Log Forwarder and Log Server breaks and resend when connection re-establishes.

> **Note:**
>
> These settings are reflected in the File Protector by doing **`dfpadmin update`** after changing the configuration.

> **Note:**
>
> Ensure the Log Forwarder is working before restarting the PEP Server. If you restart the PEP Server in the error mode and the Log Forwarder is stopped, then the policy will not be available.

> **Note:** Ensure that the Logforwarder is in running state before restarting the PEP Server. If you restart the PEP Server in the error mode and the Logforwarder is stopped, then the policy will be unavailable.

Chapter 10

Backing up and Restoring the Protected Data

This section describes how to back up and restore the File Encryption (FE) encrypted data, Access Conrol (AC) protected data, or both using the FUSE FP.

The following snippet describes the **`dfp delegate`** commands with the *share-with-es* option.

```
[root@rhel74base ~]# dfp delegate -o share-with-es -e <Absolute path of the backup binary> <role
name>@<policy name>
```

Important:
The *share-with-es* option will work only if the *es* modules are enabled.

The Enhanced Security (*es*) modules are now delivered as a separate package. For more information about the packages required for specific platforms, contact Protegrity Support.

10.1 Working with the FE Encrypted Data

This section describes the various scenarios to back up and restore the FE encrypted data.

10.1.1 Backing up the Data

This section describes steps for backing up the FE encrypted data.

Before you begin
Prior to a back up, ensure that you perform the following tasks on the ESA:

1. Create a data element with the following access masks and output settings.

Table 10-1: Access Masks and Output Settings of a Data Element

Access Masks					Output Settings	
Read	Write	Create	Delete	Protect	Cipher	Exception
-	-	-	-	-	√	-

2. Create a role for backup and restore that includes the data element created in step 1.
3. From the ESA, deploy a policy for the backup and restore role that includes no access masks and the cipher output setting.

To back up the FE encrypted data:

1. Delegate the required binary with the backup and restore role using the following command.

   ```
   dfp delegate -e <Absolute path of the binary> <role name>@<policy name>
   ```

   For example, `dfp delegate -e /usr/bin/cp role1@policy1`

   Where,

   - */usr/bin/cp* is the absolute path of the binary
   - *role1@policy1* are the backup and restore role and policy

   For example, `dfp delegate -e /usr/bin/scp role1@policy1`

   Where,

   - */usr/bin/scp* is the absolute path of the binary for remote backup
   - *role1@policy1* are the backup and restore role and policy
2. Copy the FE encrypted data from the source path to a location outside the FUSE mount point using the delegated binary.

   ```
   cp -r <source path>/feDir <backup path>/
   ```

   Where,

   - *cp* is the delegated binary
   - *-r* is the recursive option
   - *feDir* is the FE protected directory

   > **Note:** The source path is the location from where you are taking back up of the protected data and the backup path is the location that is used for backing up of the protected data.
3. Back up the *delegate.db* file using the following command.

   ```
   dfp delegate export <backup path/delegate.db file name>
   ```

10.1.2 Restoring the Data

You can restore the FE encrypted data from the backup path to the following locations.

- The same path from where the backup was created.
- To a different path

> **Note:** If you restore the FE encrypted data to a different path, then you must create a FUSE mount path to access the encrypted data.

Before you begin

Prerequisites

1. Ensure that the required binary is delegated using the backup and restore role using the following command.

   ```
   dfp delegate status
   ```

2. If the required binaries are not delegated, then delegate the required binaries with the backup and restore role using the following command.

   ```
   dfp delegate -e <Absolute path of the binary> <role name>@<policy name>
   ```

   For example, `dfp delegate -e /usr/bin/cp role1@policy1`

   - */usr/bin/cp* is the absolute path of the binary
   - *role1@policy1* are the backup and restore role and policy

   For example, `dfp delegate -e /usr/bin/scp role1@policy1`

   Where,

   - */usr/bin/scp* is the absolute path of the binary for remote backup
   - *role1@policy1* are the backup and restore role and policy

To restore the FE encrypted data:

1. Restore the *delegate.db* file using the following command.

   ```
   dfp delegate import [-f] <path of exported delegate.db file>
   ```

   Where,

   [-f] - option overwrites the existing *delegate.db* file.

   > **Note:** This command restores the delegation settings contained in the exported *delegate.db* file.

2. Copy the files from the backup path to the FUSE source path using the delegated binary.

   For example, run the following command to restore the only FE encrypted data from the backup path of the local backup system.

   ```
   cp -r /<backup path> /<restore path>
   ```

   Where,

   - *cp* is the delegated binary for local backup.
   - *-r* is the recursive option.

   For example, run the following command to restore the only FE encrypted data from the backup path of the remote backup system.

   ```
   scp -r <User>@2.10.1.3:/<Backup path> /<restore path>
   ```

   Where,

   - *scp* is the delegated binary
   - *-r* is the recursive option

- *2.10.1.3* is the remote system IP

> **Note:** It is recommended to restore the FE encrypted data from the backup path to the source path.

10.2 Working with the AC or AC and FE Protected Data

This section describes the various scenarios to back up and restore the AC or AC and FE protected data.

10.2.1 Backing up the Data

This section describes steps for backing up the AC or AC and FE protected data.

Before you begin

Prior to a back up, ensure that you perform the following tasks on the ESA:

1. Create a data element with the following access masks and output settings.

Table 10-2: Access Masks and Output Settings of a Data Element

Access Masks					**Output Settings**	
Read	Write	Create	Delete	Protect	Cipher	Exception
-	-	-	-	-	√	-

2. Create a role for backup and restore that includes the data element created in step 1.
3. From the ESA, deploy a policy for the backup and restore role that includes no access masks and the cipher output setting.

➤ To back up the AC and FE protected data:

1. Delegate the required binary with backup and restore role using the following command.

```
dfp delegate -e <Absolute path of the binary> <role name>@<policy name>
```

For example, `dfp delegate -e /usr/bin/cp role1@policy1`

Where,

- */usr/bin/cp* is the absolute path of the binary
- *role1@policy1* are the backup and restore role and policy

For example, `dfp delegate -e /usr/bin/scp role1@policy1`

Where,

- */usr/bin/scp* is the absolute path of the binary for remote backup
- *role1@policy1* are the backup and restore role and policy

2. Copy the AC or AC and FE protected data to a location outside the FUSE mount point using the delegated binary.

```
cp -r <source path>/acfeDir <backup path>/
```

Where,

- *cp* is the delegated binary

- *-r* is the recursive option
- *acfeDir* is the AC and FE protected directory

> **Note:** The source path is the location from where you have taken back up and the backup path is the location that is used for backing up the protected data.

> **Note:** It is recommended to back up the AC or AC and FE protected directories from the source path.

3. Back up the *ac.db* file using the following command.

```
dfp ac export <backup path/ac.db file name>
```

4. Back up the *delegate.db* file using the following command.

```
dfp delegate export <backup path/delegate.db file name>
```

10.2.2 Restoring the Data

You can restore the only AC or both AC and FE protected data from the backup path to the following locations.

- The same path from where the backup was created.
- To a different path

> **Note:** If you restore the protected data to a different path, then you must create a FUSE mount path to access the protected data.

Before you begin

Prerequisites

1. Ensure that the required binary is delegated using the backup and restore role using the following command.

```
dfp delegate status
```

2. If the required binaries are not delegated, then delegate the required binaries with the backup and restore role using the following command.

```
dfp delegate -e <Absolute path of the binary> <role name>@<policy name>
```

For example, `dfp delegate -e /usr/bin/cp role1@policy1`

- */usr/bin/cp* is the absolute path of the binary
- *role1@policy1* are the backup and restore role and policy

For example, `dfp delegate -e /usr/bin/scp role1@policy1`

Where,

- */usr/bin/scp* is the absolute path of the binary for remote backup
- *role1@policy1* are the backup and restore role and policy

➤ To restore the AC or AC and FE protected data:

1. Restore the *delegate.db* file using the following command.

```
dfp delegate import [-f] <path of exported delegate.db file>
```

Where,

[-f] - option overwrites the existing *delegate.db* file.

> **Note:** This command restores the delegation settings contained in the exported *delegate.db* file.

2. Restore the *ac.db* file using the following command.

```
dfp ac import [-f] <path of exported ac.db file>
```

[-f] - option overwrites the existing *ac.db* file.

> **Note:** This command restores the delegation settings contained in the exported *ac.db* file.

3. Copy the files from the backup path to the FUSE source path using the delegated binary.

For example, run the following command to restore the AC or AC and FE protected data from the backup path of the local backup system.

```
cp -r /<backup path> /<restore path>
```

Where,

- *cp* is the delegated binary for local backup.
- *-r* is the recursive option.

For example, run the following command to restore the AC or AC and FE protected data from the backup path of the remote backup system.

```
scp -r <User>@2.10.1.3:/<Backup path> /<restore path>
```

Where,

- *scp* is the delegated binary
- *-r* is the recursive option
- *2.10.1.3* is the remote system IP

> **Note:** It is recommended to restore the AC or AC and FE protected directory from the backup path to the source path.

> **Note:**
>
> If you are restoring the AC protected data on a different path, then you must protect the files using access control.
>
> For more information about AC protection of the files, refer to *dfp Commands*.

Chapter 11

Use Cases for the FUSE FP

11.1 FUSE FP for the Network File System (NFS)

This section describes the following use cases for the Network File System (NFS).

- Use Case: FUSE FP is installed on the NFS Client
- Use Case: FUSE FP is installed on the NFS Server

11.1.1 Use Case: FUSE FP is Installed on the NFS Client

In this scenario, you install the FUSE FP on the NFS Client to protect the files located in the NFS shared path on the server. Here, multiple clients connect to the Network Attached Storage (NAS) server through the NFS.

Perform the following steps.

1. Configure the NFS client in the ESA as the data store.
2. Create and deploy the policies on the NFS client.
3. Mount the NFS shared path on the NFS client.
4. Configure the FUSE FP mount point for the NFS mount.
5. Encrypt file and directories in the FUSE FP mount point.

The following figure displays the encrypted data flow within the NFS server and clients.

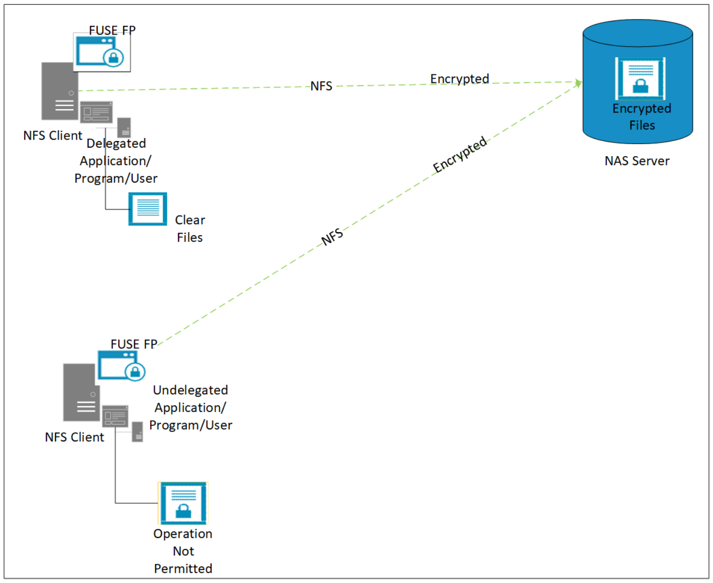

Figure 11-1: Use Case 1: FUSE FP Installed on the NFS Client

11.1.1.1 Encrypting Files and Directories on a NFS Client

➤ To encrypt files and directories on the NFS client:

1. Install FUSE FP on the NFS client.

   For more information about installing FUSE FP, refer to *Installing the FUSE FP*.
2. Configure the NFS client in the ESA as the data store.

   For more information about configuring data stores, refer to *Data Stores*.
3. Create policies on the ESA.

   For more information about creating policies, refer to *Creating Policy*.
4. Deploy the policy on the NFS client.

   For more information about deploying policies, refer to *Deploying Policy*.
5. Run the following NFS mount command to mount the NFS share on the NFS client.

   ```
   mount -t nfs -o vers=3/4 <ServerIP>:<NFS shared path>/ nfs_client_mount
   ```

   NFS mount point should be made persistent after restart. Check mtab entry to verify the NFS mount.
6. Create an entry in the `/etc/fstab` file to make the NFS mount point persistent.

   ```
   <serverIp>:/ <absolute nfs client path> nfs
   ac,rsize=1048576,wsize=1048576,nfsvers=3/4 0 0
   ```
7. Create the FUSE FP mount point for the NFS mount on the NFS client.

   ```
   dfp mount --add-entry -o modules=subdir,allow_other -o subdir=/<nfs_client_mount>
   <absolute path for FUSE mount>
   ```

   > **Note:** The `subdir` denotes the absolute path for FUSE FP mount point.
8. Run the following command to encrypt files in the NFS shared path.

   ```
   dfp file protect -noac -d <data element> <file>
   ```
9. Run the following command to verify the status of the protected files.

```
dfp file stat <file>
```

10. Run the following command to encrypt directories in the NFS shared path.

    ```
    dfp file protect -noac [-r] -d <data element> <folder>
    ```

    > **Note:** The *-r* option allows you to protect the directories recursively.

11. Run the following command to check the status of the protected directories.

    ```
    dfp file stat [-r] <folder>
    ```

12. On the FUSE FP mount point, only a delegated user, process, or program can run file operations on the encrypted files. The files are stored in an encrypted format on the server and always remain encrypted on the network.

    > **Note:**
    > If an undelegated program, process, or user tries to access the protected files and directories, then the following error message appears.
    >
    > ```
    > Permission denied
    > ```
    >
    > To access a file operation, the program, process, or user must be delegated.
    >
    > Run the following command to delegate programs.
    >
    > ```
    > dfp delegate -e <program> <policyName>
    > ```
    >
    > After performing the given steps, the program must be delegated and can access the encrypted files or directories.
    >
    > For more information about delegation, refer to *Using Delegation*.

11.1.2 Use Case: FUSE FP is Installed on the NFS Server

In this scenario, the FUSE FP is installed on the NFS server.

Perform the following steps.

1. Create the FUSE FP mount point for the NFS mount in the NFS server.
2. Mount the files and directories that need to be protected in the FUSE FP mount point.
3. Update the FUSE FP mount point and the client IP address in the `/etc/exports` directory.
4. Protect the files and directories in the FUSE FP mount point and delegate the NFS startup script with the required policy.
5. Mount the FUSE FP server mount point on the NFS client.

The following figure displays the encrypted data flow within the NFS server and clients.

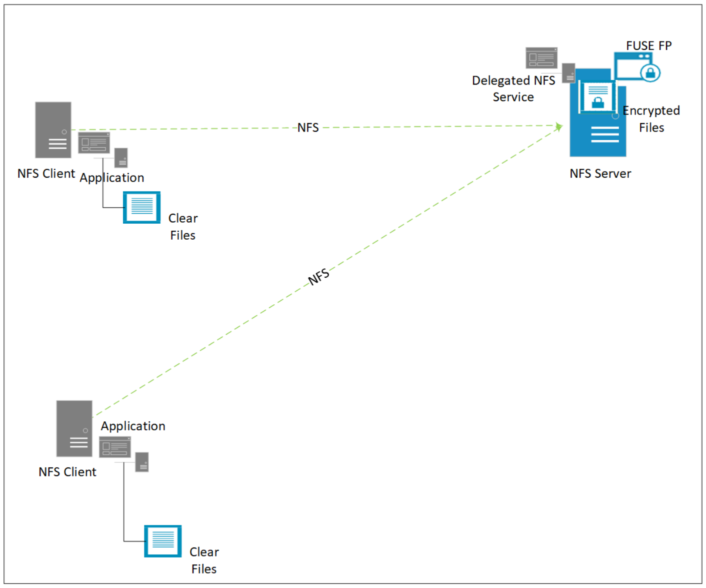

Figure 11-2: Use Case 2: FUSE FP installed on the NFS Server

11.1.2.1 Encrypting Files and Directories on a NFS Server

Before you begin

Before configuring the FUSE FP for the NFS share, ensure that the following prerequisites are met.

- The FUSE FP is installed on the NFS server.
- The FUSE FP is not installed on the NFS client.

➤ To encrypt files and directories on the NFS file system:

1. Perform the following steps on the NFS server.
   a. Install the FUSE FP on the NFS server.

      For more information about installing the FUSE FP, refer to *Installing the FUSE FP*.

   b. Configure the NFS server as the data store.

      For more information about configuring data stores, refer to *Data Stores*.

   c. Create policies on the ESA.

      For more information about creating policies, refer to *Creating Policy*.

   d. Deploy the policies on the NFS server.

      For more information about deploying policies, refer to *Deploying Policy*.

   e. On the NFS server, run the following command to create the FUSE FP mount point for the NFS mount.

      ```
      dfp mount --add-entry -o modules=subdir,allow_other -o subdir=<absolute path of nfs shared path> <absolute FUSE FP mount point>
      ```

      For example,

      ```
      dfp mount --add-entry -o modules=subdir,allow_other -o subdir=/testData/test_nfs /testData/test_mnt
      ```

      where, `/testData/test_nfs` is the nfs shared path and `/testData/test_mnt` is the FUSE mount point.

      subdir denotes the absolute path for the FUSE FP mount point.

allow_other enables the users of the NFS client to have access to the FUSE mount point.

f. Update the `/etc/exports` directory to add client entries.

   Ensure that only FUSE FP mount point is exported.

   ```
   <absolute path for fuse mount point> <client ip> (rw,,sync,no_root_squash,fsid=0)
   ```

   Where, these options *(rw,,sync,no_root_squash,fsid=0)* are must for FUSE FP on NFS. The user can add additional options as per their configurations.

   For example,

   ```
   /testdata/test_mnt <client ip>(rw,,sync,no_root_squash,fsid=0)
   ```

g. Run the following command to export the updated entries.

   ```
   export fs -a
   ```

h. Using the NFS server in the FUSE FP mount point, perform the following steps.

   i. Run the following command to protect files.

      ```
      dfp file protect -noac -d <data element> <file>
      ```

   ii. Run the following command to check the protected status of files.

      ```
      dfp file stat <file>
      ```

   iii. Run the following command to protect directories.

      ```
      dfp file protect -noac [-r] -d <data element> <folder>
      ```

   iv. Run the following command to check the protected status of directories.

      ```
      dfp file stat [-r] <folder>
      ```

i. Delegate the NFS start up script or daemon with the required policy.

   ```
   dfp delegate -e <NFS startup script or daemon> <policy name>
   ```

   For example,

   ```
   dfp delegate -e /usr/sbin/rpc.nfsd <policy name>
   ```

   where, */usr/sbin/rpc.nfsd* is NFS startup script or daemon, *<policy name>* is the policy name.

j. Restart the NFS service to enforce delegation.

2. Perform the following steps on the NFS Client.

   a. Run the following NFS mount command to mount the NFS share (Fuse FP mount point on server) on the NFS client.

      ```
      mount -t nfs -o vers=3/4 <ServerIP>:<NFS shared path> /nfs_client_mount
      ```

   b. Run `mount` command to verify that the NFS server is mounted.

After performing the given steps, the NFS client can access the encrypted directories and files in clear format. The files are stored in encrypted format on the server and remain in clear format on the network.

11.2 FUSE FP for the Common Internet File System (CIFS)

The following section is useful for those users who want to install FUSE FP in Common Internet File System (CIFS) for protecting files and directories.

This section describes the following use cases for CIFS.

- Use Case: FUSE FP is installed on the CIFS Client
- Use Case: FUSE FP is installed on the CIFS Server

11.2.1 Use Case: FUSE FP is Installed on the CIFS Client

In this scenario, the FUSE FP is installed on the CIFS Client to protect files located in the CIFS shared path on the server. In this case multiple clients connect to the CIFS server through CIFS.

Perform the following steps.

1. Configure the CIFS client node in the ESA as the data store.
2. Create and deploy the policies on the CIFS client.
3. Mount the CIFS shared path on the CIFS client.
4. Setup the FUSE FP mount point for the CIFS client.
5. Encrypt the file and directories in the FUSE FP mount point.

The following figure displays the encrypted data flow within the CIFS server and clients.

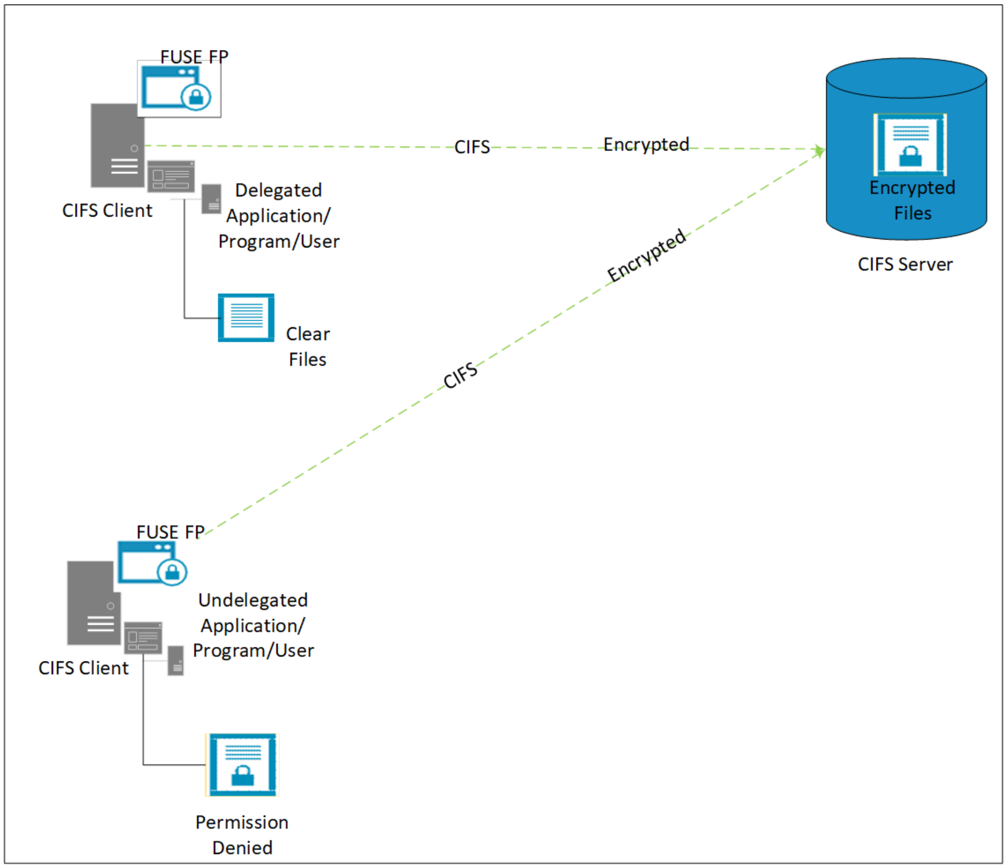

Figure 11-3: Use Case 1: FUSE FP Installed on the CIFS Client

11.2.1.1 Encrypting Files and Directories on a CIFS Client

To encrypt files and directories on the CIFS client:

1. Install the FUSE FP on the CIFS client.

   For more information about installing the FUSE FP, refer to *Installing the FUSE FP*.

2. Configure the CIFS client in the ESA as the data store.

   For more information about configuring data stores, refer to *Data Stores*.
3. Create policies on the ESA.

   For more information about creating policies, refer to *Creating Policy*.
4. Deploy the policy on the CIFS client.

   For more information about deploying policies, refer to *Deploying Policy*.
5. Mount the CIFS shared path on the CIFS client using the following command.

   ```
   mount -t cifs //<Server IP>/<CIFS shared path> /cifs_client_mount
   ```

   For example, `mount -t cifs //<Server IP>/smbshare -o username=root /cifs_client`

   Where, */smbshare* is the configuration setup for the CIFS shared directory.

6. Create an entry in the `/etc/fstab` file to make a persistent CIFS mount point, which remains active even in case of a system restart, using the following command.

   ```
   //rhclus-n4/sharedrepo /mnt cifs username=root,password=protegrity,defaults 0 0
   ```

   where, *//rhclus-n4/sharedrepo* directory is the server shared path and */mnt* directory is the CIFS client mount.

7. On the CIFS client, setup the FUSE FP mount point for the CIFS mount using the following command.

   ```
   dfp mount --add-entry -o allow_other -o modules=subdir -o subdir=<absolute path of CIFS mount path> <absolute FUSE FP mount point>
   ```

   > **Note:** The *subdir* parameter denotes the absolute path for the FUSE FP mount point.

   For example, `dfp mount --add-entry -o allow_other -o modules=subdir -o subdir=/mnt /Fuse_client_mnt`

   where, the */mnt* directory is the CIFS shared path and the */Fuse_client_mnt* directory is the FUSE mount path.

8. Using the FUSE FP with the required policy, perform the following steps.

   a. Run the following command to encrypt files in the CIFS shared path.

   ```
   dfp file protect -noac -d <data element> <file>
   ```

   b. Run the following command to check the protected file status.

   ```
   dfp file stat <file>
   ```

   c. Run the following command to encrypt directories in the CIFS shared path.

   ```
   dfp file protect -noac [-r] -d <data element> <folder>
   ```

   > **Note:** The *-r* option allows you to protect directories recursively.

   d. Run the following command to check the status of the protected directories.

   ```
   dfp file stat [-r] <folder>
   ```

9. On the FUSE FP mount point, only a delegated user, process, or program can run file operations on the encrypted files. The files are stored in an encrypted format on the server and always remain encrypted on the network.

   > **Note:**
   >
   > If any undelegated program, process, or user tries to access the protected file and directory path, then the following error message appears.
   >
   > ```
   > Permission denied
   > ```

> To access a file operation, the program, process, or user must be delegated.
>
> Run the following command to delegate programs.
>
> ```
> dfp delegate -e <program> <policyName>
> ```
>
> After performing the given steps, the program will be delegated and can access the encrypted files or directories.
>
> For more information about delegation, refer to *Using Delegation*.

11.2.2 Use Case: FUSE FP is Installed on the CIFS Server

In this scenario, the FUSE FP is installed on the CIFS server.

Perform the following steps.

1. Create the FUSE FP mount point for the CIFS shared path on the CIFS server.
2. Mount the files and directories that need to be protected in the FUSE FP mount point.
3. Create the CIFS configuration setup on the server side for sharing protected data with the CIFS clients.
4. Protect the files and directories in the FUSE FP mount point and delegate the CIFS binary with the required policy.
5. Mount the FUSE FP server mount point on the CIFS client.

The following figure displays the encrypted data flow within the CIFS server and clients.

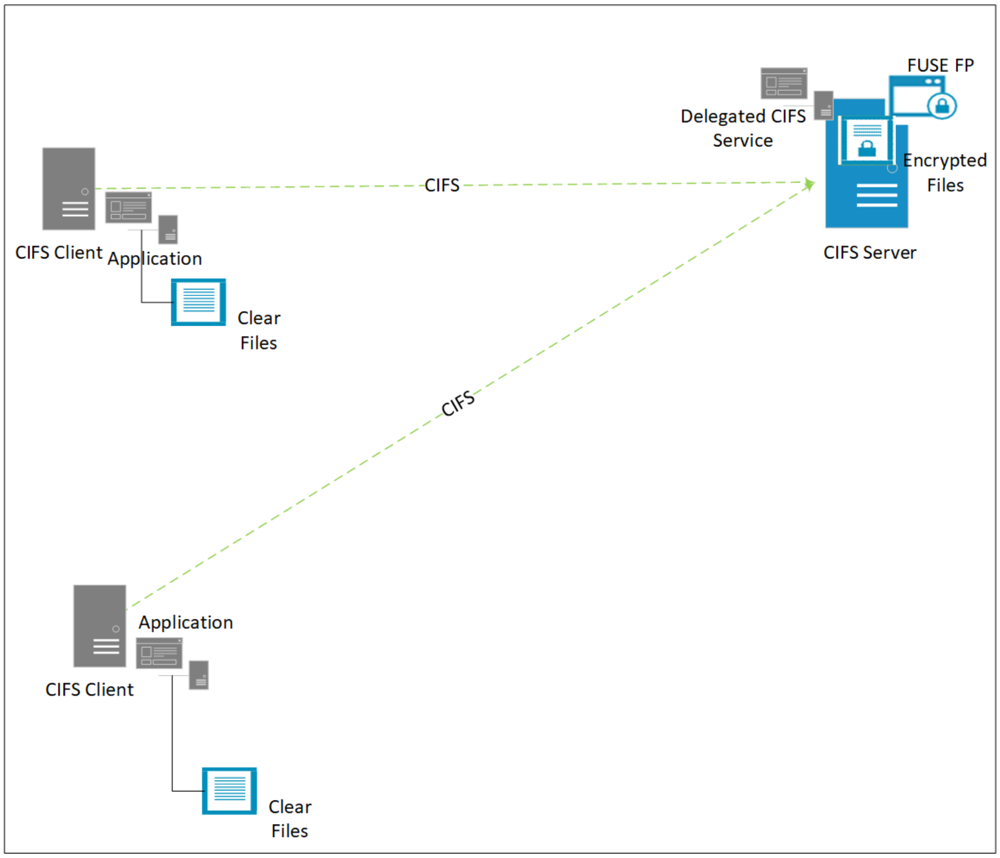

Figure 11-4: Use Case 2: FUSE FP installed on the CIFS Server

11.2.2.1 Encrypting Files and Directories on a CIFS Server

Before you begin

Before configuring the FUSE FP for the CIFS share, you must verify the following prerequisites:

1. The FUSE FP is installed on the CIFS server.
2. The FUSE FP is not installed on the CIFS client.

➤ To encrypt files and directories on the CIFS file system:

1. Perform the following steps on the CIFS server.
   a. Install the FUSE FP on the CIFS server.

      For more information about installing the FUSE FP, refer to *Installing the FUSE FP*.
   b. Configure the CIFS server as the data store.

      For more information about configuring data stores, refer to *Data Stores*.
   c. Create policies on the ESA.

      For more information about creating policies, refer to *Creating Policy*.
   d. Deploy the policies on the CIFS server.

      For more information about deploying policies, refer to *Deploying Policy*.
   e. On the CIFS server, run the following command to create the FUSE FP mount point for the CIFS shared path.

      ```
      dfp mount --add-entry -o allow_other -o modules=subdir -o subdir=<absolute path of CIFS source path for FUSE> <absolute FUSE FP mount point>
      ```

      For example,

      ```
      dfp mount --add-entry -o allow_other -o modules=subdir -o subdir=/testData/test_cifs /testData/test_mnt
      ```

      where, `/testData/test_cifs` is the CIFS source path for FUSE and `/testData/test_mnt` is the FUSE mount point. The FUSE mount point must be shared with the CIFS clients.

      The *subdir* parameter denotes the absolute path for the FUSE FP mount point.

      The *allow_other* parameter enables the users of the CIFS client to have access to the FUSE mount point.
   f. Create the CIFS configuration setup on the server side and modify the *smb.conf* file in the `/etc/samba/` directory.

      ```
      [smbshare]
      path = /testData/test_mnt
      browseable = yes
      writable = yes
      public = yes
      ```

      where, `/testData/test_mnt` is the FUSE mount created for the CIFS shared directory.
   g. Using the CIFS server in the FUSE FP mount point, perform the following steps.
      i. Run the following command to protect files.

         ```
         dfp file protect -noac -d <data element> <file>
         ```
      ii. Run the following command to check the protected status of files.

         ```
         dfp file stat <file>
         ```
      iii. Run the following command to protect directories.

```
dfp file protect -noac [-r] -d <data element> <folder>
```

iv. Run the following command to check the protected status of directories.

```
dfp file stat [-r] <folder>
```

h. Delegate the CIFS binary with the required policy using the following command.

```
dfp delegate -e <CIFS binary> <policy name>
```

For example,

```
dfp delegate -e /usr/sbin/smbd <policy name>
```

where, */usr/sbin/smbd* directory is the CIFS binary and *<policy name>* is the policy name.

i. Restart the CIFS or Samba services to enforce delegation.

2. Perform the following steps on the CIFS client.

a. Run the following CIFS mount command to mount the CIFS share (FUSE FP mount point on the server) on the CIFS client.

```
mount -t cifs //<Server IP><CIFS shared path> <cifs_client_mount>
```

For example, `mount -t cifs //<Server IP>/testData/test_mnt /fuse/mnt`

where, `/testData/test_mnt` is the FUSE mount created for CIFS shared directory and `/fuse/mnt` is the CIFS client mount.

b. Run the following command and ensure that the mount list contains the CIFS entry.

```
mount -v | grep cifs
```

After performing the given steps, the CIFS client can access the encrypted directories and files in clear format. The files are stored in encrypted format on the server and the files are always remain in clear format on the network.

11.3 FUSE FP for the File Transfer Protocol (FTP) Server Node

In this scenario, the FUSE FP is installed on the File Transfer Protocol (FTP) server node to protect files located in the FTP shared path on the server. This is the case of FTP setup, wherein multiple FTP clients access the FTP server node for the protected files and directories.

> **Note:** Before configuring the FUSE FP for the FTP share path, ensure that the FUSE FP is installed on the FTP server node.

Ensure that you perform the following steps to run this use case:

1. Setup the FUSE FP mount point for the FTP share path in the FTP server node.
2. Mount the files and directories that need to be protected in the FUSE FP mount point.
3. Protect files and directories in the FUSE FP mount point and delegate the FTP daemon with the required policy.

> **Note:** The FUSE FP mount point must be shared path.

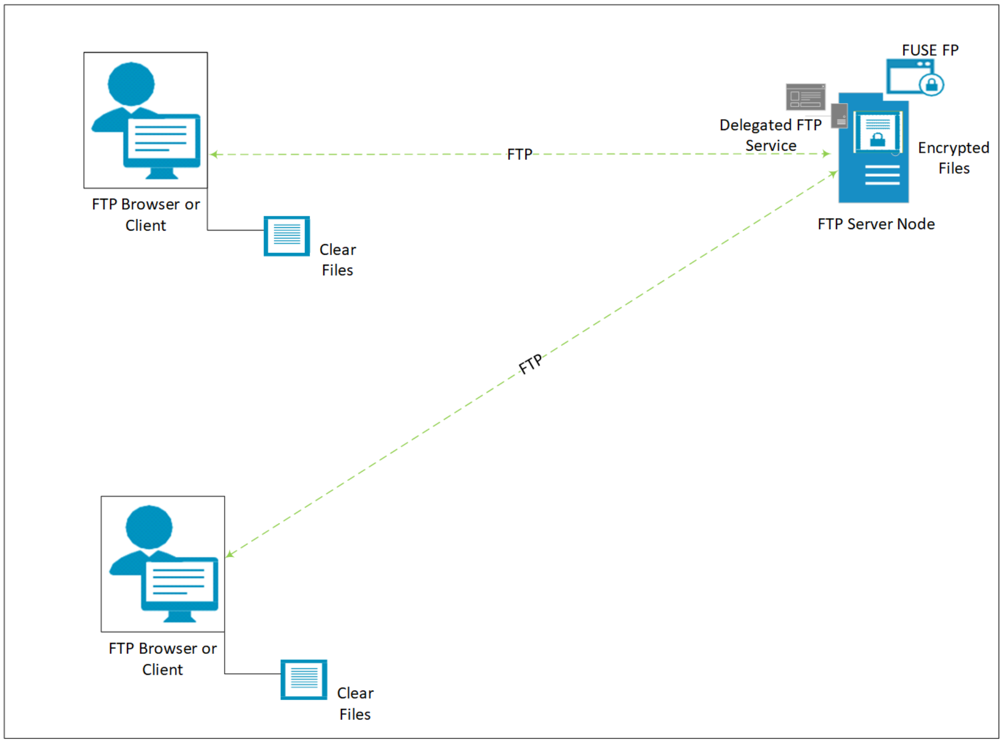

Figure 11-5: FUSE FP installed on the FTP Server

11.3.1 Encrypting Files and Directories on a FTP Server

➤ To encrypt files and directories on the FTP server node:

1. Install the FUSE FP on the FTP server node.

   For more information about installing the FUSE FP, refer to *Installing the FUSE FP*.

2. To create policy on the ESA:

   a. Configure the FTP server node as data store.

      For more information about configuring data stores, refer to *Data Stores*.

   b. Create policies.

      For more information about creating policies, refer to *Creating Policy*.

3. Deploy policies on the FTP server node.

   For more information about deploying policies, refer to *Deploying Policy*.

4. On the FTP server node, run the following dfp mount command to setup the FUSE FP mount point for the FTP mount.

   ```
   dfp mount --add-entry -o allow_other -o modules=subdir -o subdir=<absolute path of FTP shared path> <absolute FUSE FP mount point>
   ```

   For example,

   ```
   dfp mount --add-entry -o allow_other -o modules=subdir -o subdir=/srcData/src_ftp /testFTP/test_mnt
   ```

Filepath and options	Denotes
`/srcData/src_ftp` directory	Protection point
`/testFTP/test_mnt` directory	FTP shared path

Filepath and options	Denotes
subdir option	Absolute path for the FUSE FP mount point
allow_other option	The users of the FTP client to have access on the FUSE mount point

5. On the FTP server node in the FUSE FP mount point,
   a. Run the following command to protect files.
   ```
   dfp file protect -noac -d <data element> <file>
   ```
   b. Run the following command to check the protected status of files.
   ```
   dfp file stat <file>
   ```
   c. Run the following command to protect directories.
   ```
   dfp file protect -noac [-r] -d <data element> <folder>
   ```
   d. Run the following command to check the protected status of directories.
   ```
   dfp file stat [-r] <folder>
   ```
6. Delegate the FTP daemon with the required policy.
   For example,
   ```
   dfp delegate -e /usr/sbin/rpc.vsftpd <policy name>
   ```

   where, */usr/sbin/rpc.vsftpd* is FTP daemon, *<policy name>* is the policy name.

After performing the given steps, the FTP client can access the encrypted path in clear format. The files are stored in encrypted format on the server node and always remain clear on the network.

11.4 FUSE FP for the Local File System

The following section is useful for those users who want to install FUSE FP in Local File System for protecting files and directories.

11.4.1 Encrypting Files and Directories on a Local File System

➤ To encrypt files and directories in local file system:

1. Install FUSE FP on a system.
   For more information about installing FUSE FP, refer to *Installing the FUSE FP*.
2. Configure the system or virtual machine (VM) as data stores.
   For more information about configuring data stores, refer to *Data Stores*.
3. Create policies on the ESA.
   For more information about creating policies, refer to *Creating Policy*.
4. Deploy policies on the node.
   For more information about deploying policies, refer to *Deploying Policy*.
5. Create a FUSE mount for the path that must be protected.
   ```
   dfp mount --add-entry -o max_read=131072 -o modules=subdir,allow_other -o
   subdir=<absolute local source dir> <absolute path for fuse mount>
   ```
6. Using the FUSE FP with required policy, perform the following steps.
   a. Run the following command to encrypt files inside the FUSE mount point.

```
dfp file protect -noac -d <data element> <file>
```

To check the protected file status, run the following command:

```
dfp file stat <file>
```

b. Run the following command to encrypt directories inside the mount point:

```
dfp file protect -noac [-r] -d <data element> <folder>
```

To check the protected status of directories, run the following command:

```
dfp file stat [-r] <folder>
```

For more information about delegating the program, user, or process, refer to *Using Delegation*. The delegation is required if any user, program, or process are required to access the encrypted path.

> **Note:**
>
> Within the FUSE FP mount point,
>
> - If the policy is not loaded and the `ls` command is run to list the encrypted files, then it displays `permission denied` message.
> - If the policy is not loaded and `ls -l` command is run to list the encrypted files, then it displays the file attributes.

Chapter 12

Improvements over the Kernel File Protector (Kernel FP)

This section describes the improvements of the FUSE FP over the Kernel File Protector (Kernel FP).

- **No restart required**

  The Kernel FP requires a system restart to apply the FP hooks for system calls. The upgrade to the FUSE FP replaces packages in the user space and no server restart is required for upgradation.

- **No conflicts with third party kernel software**

  The FUSE FP implements file system in the user space. Hence, it does not interfere with other kernel space software such as antivirus.

- **Simpler deployment than the Kernel FP**

  To deploy a Kernel FP, kernel development, and other kernel-related patches are required. The FUSE FP deployment does not have any such dependency.

- **Behavior for non-delegated Entities**

  In case of the FUSE FP, a `Permission denied` error appears, when a non-delegated program, user, or process access the encrypted paths. In case of the Kernel FP (without Access Control), the non-delegated entities can access and read the protected or encrypted path as ciphertext.

- **System impact only limited to the FUSE FP mount point**

  As the FP syscalls are triggered for every system call including the non-protected path, the Kernel FP components have a system-wide impact. In the FUSE FP, the calls are limited to the protected path.

Chapter 13

Limitations

This section describes various limitations of the FUSE FP.

- **Protection Limitation**

  The FUSE FP enables protection based on the inode number of any file or directory paths. The FUSE FP does not protect any non-existent paths, as non-existent paths can not provide the inode number.

  > **Note:** An inode is an entry in inode table and it contains information (the metadata) about a regular file and directory.

- **Mount Point Limitation**
  - It is not recommended to create any mount points inside the FUSE FP mount point, as the FUSE FP mount point is used to enforce the FUSE-based file system. If the mount points are created inside the FUSE FP mount point, then it causes cyclic dependencies at the file system level.
  - In the mount point, the maximum length of directory path name supported by the FUSE FP is 4088 characters.
- **Delegation Limitation**

  The FUSE FP does not support program delegation of binaries located or accessed through the FUSE mount point.

- **Interoperability Limitation**

  The FUSE FP and kernel FP should not be installed on the same machine.

Chapter 14

Troubleshooting

This section describes the various problems or errors that the user may encounters while working with the FUSE FP.

14.1 Recovering the Unknown FUSE FP Mount Path

This section describes how to recover the *unknown* FUSE FP mount path.

In case of *unknown* FUSE FP mount path, you can not unmount the FUSE FP mount path executing the `dfp umount` command. You must unmount the *unknown* FUSE FP Mount Path and subsequently execute the `dfp umount` command to completely remove the *unknown* FUSE FP Mount Path from the *automount.conf* file.

➤ To recover the *unknown* FUSE FP mount path:

1. Verify the status of the FUSE FP mount path using the following command.

   ```
   dfp mount list
   ```

   The following status message appears for the *unknown* FUSE FP mount path.

   ```
   UNKNOWN
   ```
2. Run the following command to unmount the *unknown* FUSE FP mount path.

   ```
   umount <absolute FUSE FP mount path>
   ```

   > **Note:** In case of NFS, if the *unknown* FUSE FP mount path is accessed by NFS services, then you must stop the NFS services before running the `umount` command.
3. Run the following command to remove the FUSE FP mount path from the *automount.conf* file.

   ```
   dfp umount --del-entry <absolute FUSE FP mount path>
   ```
4. Run the following command to mount again the same FUSE FP mount path.

```
dfp mount --add-entry -o modules=subdir,allow_other -o subdir=<absolute FUSE FP mount path> <absolute path for new FUSE FP mount>
```

14.2 Recovering the *dfpshell* Active Password

➤ To recover the dfpshell active password:

1. Remove the *.syslock* file from the `/etc/protegrity/fileprotector/fuse` directory.
2. Run the following command to create the new *dfpshell* password.
   ```
   dfpshell
   ```

14.3 Resolving Terminal Issue

This section describes the steps to resolve unresponsive terminal.

This issue may happen in the following scenarios:

- After the fresh installation, if the user mounts the *fusefs* server, then run the `dfp file stat <TAB>` command.
- If the ESA is changed and then run the `dfp file stat <TAB>` command on the existing bash session.

> **Note:** If you start a new bash session, then the error message do not appears.

➤ To resolve this issue:

Start a new bash session.

> **Note:** It is not recommended to use *<tab>* completion for *dfp* commands.

14.4 Restarting the PMS with the Mounted FUSE File System

This section describes the steps to restart the PMS if there is a FUSE mount server running.

➤ To have consistent mount points after PMS exit, perform the following steps:

1. Unmount all the FUSE file system mount path using the following command.
   ```
   umount -a
   ```
2. Restart the PMS using the following command.
   ```
   dfpadmin service PMS on
   ```

> **Note:**
> If you get this error message:

> ```
> Error : failed to connect process management server!
> ```
>
> Then perform the following steps to resolve the issue:
>
> 1. Verify the validity of the certificates to check if they are expired.
> 2. Check if there are any error messages within the files using the path`/var/log/message` or `/var/protegrity/dfplogd/log`

14.5 Resolving the *dfp* command errors

If a non-root user gets the following error message while executing *dfp* or *dfp mount* commands, then check the permission of *.dfp_private.env* and *.syslock* files.

For example, while executing the **`dfp info`** command, the following error message appears.

```
ERROR: failed to open file </etc/protegrity/fileprotector/.dfp_private.env>!
ERROR: client initialize fail!.Please validate the parameter in configuration file!
```

For example, while executing the **`dfp mount`** command, the following error message appears.

```
ERROR: Permission denied!Fusermount
```

The *.dfp_private.env* file is located in `/etc/protegrity/fileprotector` directory and the *.syslock* file is located in `/etc/protegrity/fileprotector/fuse` directory.

To resolve this issue, ensure that the `/etc/protegrity/fileprotector` directory have read permission for all users.

14.6 Unmounting the FuseFS Mount Point from the NFS Server

This section describes the steps to unmount the FuseFS mount point from the NFS server in a NFS server-client setup.

Before you begin

Ensure that the following per-requisites are met:

- the FUSE FP is installed on the server
- the FUSE mount path is shared to each client
- the nfsd is delegated

For more information about the NFS server-client setup, refer to *Use Case: FUSE FP is Installed on the NFS Server*.

➤ To unmount the FuseFS mount point from the NFS server:

1. List all the FUSE FP mount points using the following command.
   `lsof <actual path of the mount point>`
2. If any application is using the mount point locally, then close that application.
3. Login to each client which is accessing the shared mount point.
4. Unmount the mount point from each client.
5. On the server, restart the NFS service.
6. Unmount the FUSE mount point from the server using the following command.

```
dfp umount <Mount Point>
```

Chapter 15

Appendix A: Enhanced Security (*es*) modules

This section describes the *es* modules, the process of installing the FUSE FP with the *es* modules, and the process of backing up and restoring the protected data using the *es* modules.

> **Note:**
>
> The working of the FUSE FP *es* modules is based on the compatibility of the installed kernel version. If the installed kernel version is not compatible with the FUSE FP *es* modules, then shows an error.
>
> To avoid this error, the *es* modules are now delivered as a separate package. Please contact Protegrity for platform specific package requirements.

15.1 Installing the FUSE FP

This section describes how you can install the FUSE FP.

The FUSE FP provides options to the user to prevent access to the source and installation path.

Depending on the requirement of protecting the installation and source path of Access Control(AC) protected files, you can install the FUSE FP in the following ways:

- Install the FUSE FP without the *es* module: Use this method if you do not need to protect the installation and source path of AC protected files.
- Install the FUSE FP with the *es* module: Use this method if you need to protect the installation and source path of AC protected files.

> **Note:** Ensure that the *ed editor* is installed for the FUSE FP installer. If the *ed editor* is not installed in the system, then run the following command to install the *ed editor*.
>
> ```
> yum install ed
> ```

15.1.1 Installing FUSE FP with the *es* Modules

This section describes the steps for installing the FUSE FP with the *es* modules.

➤ To install the FUSE FP with the *es* modules:

1. If the kernel version is supported with the FUSE FP, then run the following script on the node.
   `./FileProtector_RHEL-x-64_x86-64_FUSE-ALL_x.x.x.x.sh --loadenhancedsecurity`
2. If you need to use the FUSE FP, then enter *yes*.
3. Alternatively, if you want to abort the installation of FUSE FP, then enter *no*.
4. Enter the installation directory for the PEP server.

   The PEP server is installed in the `/opt/protegrity/defiance_dps` directory by default.
5. Press **ENTER**.

   A prompt for the installation of the FUSE FP directory appears.
6. Enter the installation directory for the FUSE FP.

   A new directory is created in the `/opt/protegrity` installation directory by default.
7. If the installed kernel version is not compatible with the FUSE FP *es* modules, then the following message appears.

   ```
   =================================================================
    ERROR:
      The kernel 3.10.0-957.el7.x86_64 is not compatible with the module es!
      please contact Protegrity, Inc. for help!
   =================================================================

   Abort the installation ...

   INFO: Fuse File protector ES modules not compatible with running kernel.

   Do you wish to proceed ? [yes or no]
   ```

   > **Note:** If the installed kernel version is compatible with the FUSE FP *es* modules, then no error message appears.

   a. If you want to abort the installation of FUSE FP, then enter *no*.

   b. Alternatively, if you want to install the FUSE FP without the *es* modules, then enter *yes*.
8. Press **ENTER**.

   The installation of the FUSE FP starts.

   The following files are extracted in the `/opt/protegrity/fileprotector/fuse/bin` directory:

   - *ac_migrate_v1tov2.sh*
   - *dfp*
   - *dfpadmin*
   - *dfp_clone*
   - *dfp_fusefs_server*
   - *dfp_get_env.sh*
   - *dfp_krotate_server*
   - *dfp_log*
   - *dfp_log_server*
   - *dfp_mnt_srv*
   - *dfp_process_management_server*
   - *dfp_remote*
   - *dfp_remote_management_server*
   - *dfp_service_manager*
   - *dfpshell*
   - *dfp_uninstall_fusefp*

- *dfp_wrapper*
- *es_export_install.sh*
- *es_export_uninstall.sh*
- *es_install.sh*
- *es_uninstall.sh*
- *fp_to_fuse.sh*
- *libfpfuse.so.2.9.4*

The following note appears during installation.

```
Note: File protector config files not found in current install directory.
If you have installed File protector before in some other path and are willing to restore
the File protector configutration
Please run the /opt/protegrity/fileprotector/fuse/bin/fp_to_fuse.sh after installation.
```

> **Note:**
>
> If you have installed kernel based File Protector before in another path and want to migrate the FUSE FP, then run the following script after installation.
>
> ```
> /opt/protegrity/fileprotector/fuse/bin/fp_to_fuse.sh
> ```

9. Enter a new *dfpshell* password.

> **Note:** The *dfpshell* is the system administrator shell for the File-Protector. The *dfpshell* password must contain a minimum of 8 characters and a maximum of 129 characters in length. It should contain a mix of numeric, alphabetic, and printable characters.

10. Press **ENTER**.

    A prompt to verify the *dfpshell* password appears.

11. Re-enter the *dfpshell* password to verify.
12. Press **ENTER**.

    The following message appears.

```
Create dfpshell password successfully!
Module es_export loaded successfully !!
Module es loaded successfully !!
Initializing the services...

File Protector(FUSE-ALL) installed in /opt/protegrity/fileprotector/fuse .

INFO: Please create the mount point directory and use <dfp mount> command with options
for starting dfp_fusefs_server.
 It is highly recommended to restart All the bash login sessions
 in order to update configuration settings.
```

> **Caution:** It is recommended to restart all the *bash* login sessions after installation to update the configuration settings of the FUSE FP.

After the successful installation of the FUSE FP, you must update the configuration files.

For more information about configuration files, refer to *Setting the Configuration Files*.

15.2 Backing up and Restoring the Protected Data

This section describes how to back up and restore the File Encryption (FE) encrypted data, Access Conrol (AC) protected data, or both using the FUSE FP.

Depending upon the states of the FUSE FP *es* modules, you can back up the data in the following ways:

- Backing up with the *es* modules: Use this method to back up the protected files when the *es* modules are enabled.
- Backing up without the *es* modules: Use this method to back up the protected files when the *es* modules are not enabled.

The following snippet describes the `dfp delegate` commands with the *share-with-es* option.

```
[root@rhel74base ~]# dfp delegate -o share-with-es -e <Absolute path of the backup binary>
<role name>@<policy name>
```

Important:

The *share-with-es* option will work only if the *es* modules are enabled.

The Enhanced Security (*es*) modules are now delivered as a separate package. For more information about the packages required for specific platforms, contact Protegrity Support.

15.2.1 Working with the FE Encrypted Data

This section describes the various scenarios to back up and restore the FE encrypted data.

Depending upon the states of the FUSE FP *es* modules, you can back up and restore the FE encrypted data in the following ways:

- Backing up and restoring the data with the *es* modules: Use this method, if the *es* modules are loaded in your system.
- Backing up and restoring the data without the *es* modules: Use this method, if the *es* modules are not loaded in your system.

15.2.1.1 Backing up the Data with the *es* Modules

This section describes steps for backing up the FE encrypted data with the *es* module loaded.

Before you begin

Prior to a back up, ensure that you perform the following tasks on the ESA:

1. Create a data element with the following access masks and output settings.

Table 15-1: Access Masks and Output Settings of a Data Element

Access Masks					**Output Settings**	
Read	Write	Create	Delete	Protect	Cipher	Exception
-	-	-	-	-	√	-

2. Create a role for backup and restore that includes the data element created in step 1.
3. From the ESA, deploy a policy for the backup and restore role that includes no access masks and the cipher output setting.

➤ To back up the FE encrypted data:

1. Delegate the required binary with the backup and restore role and the *share-with-es* option using the following command.

```
dfp delegate -o share-with-es -e <Absolute path of the binary> <role name>@<policy name>
```

For example, `dfp delegate -o share-with-es -e /usr/bin/cp role1@policy1`

Where,

- */usr/bin/cp* is the absolute path of the binary for local backup
- *role1@policy1* are the backup and restore role and policy

For example, `dfp delegate -o share-with-es -e /usr/bin/scp role1@policy1`

Where,

- */usr/bin/scp* is the absolute path of the binary for remote backup
- *role1@policy1* are the backup and restore role and policy

2. Copy the FE encrypted data from the source path to a location outside the FUSE mount point using the delegated binary.

```
cp -r <source path>/feDir <backup path>/
```

Where,

- *cp* is the delegated binary
- *-r* is the recursive option
- *feDir* is the FE protected directory

> **Note:** The source path is the location from where you are taking back up of the protected data and the backup path is the location that is used for backing up of the protected data.

3. Back up the *delegate.db* file using the following command.

```
dfp delegate export <backup path/delegate db file name>
```

15.2.1.2 Restoring the Data with the *es* Modules

You can restore the FE encrypted data from the backup path to the following locations.

- The same path from where the backup was created.
- To a different path

> **Note:** If you restore the FE encrypted data to a different path, then you must create a FUSE mount path to access the encrypted data.

Before you begin

Prerequisites

1. Ensure that the required binary is delegated using the *share-with-es* option and backup and restore role using the following command.

```
dfp delegate status
```

2. If the required binaries are not delegated, then delegate the required binaries with the backup and restore role and the *share-with-es* option using the following command.

```
dfp delegate -o share-with-es -e <Absolute path of the binary> <role name>@<policy name>
```

For example, `dfp delegate -o share-with-es -e /usr/bin/cp role1@policy1`

- */usr/bin/cp* is the absolute path of the binary
- *role1@policy1* are the backup and restore role and policy

For example, `dfp delegate -o share-with-es -e /usr/bin/scp role1@policy1`

Where,

- */usr/bin/scp* is the absolute path of the binary for remote backup
- *role1@policy1* are the backup and restore role and policy

➤ To restore the FE encrypted data:

1. Restore the *delegate.db* file using the following command.

   ```
   dfp delegate import [-f] <path of exported delegate.db file>
   ```

   Where,

   [-f] - option overwrites the existing *delegate.db* file.

   > **Note:** This command restores the delegation settings contained in the exported *delegate.db* file.

2. Copy the FE encrypted data from the backup path to the FUSE source path using the delegated binary.

   For example, run the following command to restore the only FE encrypted data from the backup path of the local backup system.

   ```
   cp -r /<backup path> /<restore path>
   ```

   Where,

   - *cp* is the delegated binary for local backup.
   - *-r* is the recursive option.

   For example, run the following command to restore the only FE encrypted data from the backup path of the remote backup system.

   ```
   scp -r <User>@2.10.1.3:/<Backup path> /<restore path>
   ```

   Where,

   - *scp* is the delegated binary
   - *-r* is the recursive option
   - *2.10.1.3* is the remote system IP

   > **Note:** It is recommended to restore the FE encrypted data from the backup path to the source path.

15.2.2 Working with the AC or AC and FE Protected Data

This section describes the various scenarios to back up and restore the AC or AC and FE protected data.

Depending upon the states of the FUSE FP *es* modules, you can back up and restore the AC or AC and FE protected data in the following ways:

- Backing up and restoring the data with the *es* modules: Use this method, if the *es* modules are loaded in your system.
- Backing up and restoring the data without the *es* modules: Use this method, if the *es* modules are not loaded in your system.

15.2.2.1 Backing up the Data with the *es* Modules

This section describes steps for backing up the AC or AC and FE protected data using the FUSE FP.

Before you begin

Before backing up the data, ensure that you perform the following tasks on the ESA:

1. Create a data element with the following access masks and output settings.

Table 15-2: Access Masks and Output Settings of a Data Element

Access Masks					Output Settings	
Read	Write	Create	Delete	Protect	Cipher	Exception
-	-	-	-	-	√	-

2. Create a role for backup and restore that includes the data element created in step 1.
3. From the ESA, deploy a policy for the backup and restore role that includes no access masks and the cipher output setting.

➤ To back up the AC and FE protected data:

1. Delegate the required binary with the backup and restore role using the following command.

   ```
   dfp delegate -o share-with-es -e <Absolute path of the binary> <role name>@<policy name>
   ```

   For example, `dfp delegate -o share-with-es -e /usr/bin/cp role1@policy1`

   Where,

   - */usr/bin/cp* is the absolute path of the binary for local backup
   - *role1@policy1* are the backup and restore role and policy

   For example, `dfp delegate -o share-with-es -e /usr/bin/scp role1@policy1`

   Where,

   - */usr/bin/scp* is the absolute path of the binary for remote backup
   - *role1@policy1* are the backup and restore role and policy

2. Copy the AC or AC and FE protected data to a location outside the FUSE mount point using the delegated binary.

   For example, `cp -r <source path>/acfeDir <backup path>/`

   Where,

   - *cp* is the delegated binary
   - *-r* is the recursive option

- *acfeDir* is the AC and FE protected directory

> **Note:** The source path is the location from where you are taking back up of the protected data and the backup path is the location that is used for backing up of the protected data.

> **Note:** It is recommended to back up the AC or AC and FE protected directories from the source path.

3. Back up the *ac.db* file using the following command.

   ```
   dfp ac export <backup path/ac.db file name>
   ```

4. Back up the *delegate.db* file using the following command.

   ```
   dfp delegate export <backup path/delegate.db file name>
   ```

15.2.2.2 Restoring the Data with the *es* modules

You can restore the AC or AC and FE protected data from the backup path to the following locations.

- The same path from where the backup was created.
- To a different path

> **Note:** If you restore the protected data to a different path, then you must create a FUSE mount path to access the protected data.

Before you begin

Prerequisites

1. Ensure that the required binary is delegated using the *share-with-es* option and backup and restore role using the following command.

   ```
   dfp delegate status
   ```

2. If the required binaries are not delegated, then delegate the required binaries with the backup and restore role and the *share-with-es* option using the following command.

   ```
   dfp delegate -o share-with-es -e <Absolute path of the binary> <role name>@<policy
   name>
   ```

   For example, `dfp delegate -o share-with-es -e /usr/bin/cp role1@policy1`

   - */usr/bin/cp* is the absolute path of the binary
   - *role1@policy1* are the backup and restore role and policy

   For example, `dfp delegate -o share-with-es -e /usr/bin/scp role1@policy1`

   Where,

   - */usr/bin/scp* is the absolute path of the binary for remote backup
   - *role1@policy1* are the backup and restore role and policy

➤ To restore the AC or AC and FE protected data:

1. Restore the *delegate.db* file using the following command.

```
dfp delegate import [-f] <path of exported delegate.db file>
```

Where,

[-f] - option overwrites the existing *delegate.db* file.

> **Note:** This command restores the delegation settings contained in the exported *delegate.db* file.

2. Restore the *ac.db* file using the following command.

```
dfp ac import [-f] <path of exported ac.db file>
```

[-f] - option overwrites the existing *ac.db* file.

> **Note:** This command restores the delegation settings contained in the exported *ac.db* file.

3. Copy the AC or AC and FE protected data from the backup path to the FUSE source path using the delegated binary.

For example, run the following command to restore the AC or AC and FE protected data from the backup path of the local backup system.

```
cp -r /<backup path> /<restore path>
```

Where,

- *cp* is the delegated binary for local backup.
- *-r* is the recursive option.

For example, run the following command to restore the AC or AC and FE protected data from the backup path of the remote backup system.

```
scp -r <User>@2.10.1.3:/<Backup path> /<restore path>
```

Where,

- *scp* is the delegated binary
- *-r* is the recursive option
- *2.10.1.3* is the remote system IP

> **Note:** It is recommended to restore the AC or AC and FE protected directory from the backup path to the source path.

> **Note:**
>
> If you are restoring the AC protected data on a different path, then you must protect the files using access control.
>
> For more information about AC protection of the files, refer to *dfp Commands*.

Chapter 16

Glossary

AES

Advanced Encryption Standard

DES

Data Encryption Standard

DFPSHELL

Defiance File Protector Shell

ESA

Enterprise Security Administrator

FUSE

Filesystem in Userspace

FP

File Protector

FE

File Encryption

KEK

Key Encryption Key

PMS

Process Management Service

RMS

Remote Management Service

NFS

Network File System

NAS

Network Attached Storage