Protegrity FPVE-Core User Guide 9.0.0.0

Created on: Nov 19, 2024

Copyright

Table of Contents

Chapter 1

Overview of the File Protector Volume Encryption (FPVE)-Core

This section provides information about the concepts, architecture, supported platforms, and features of the FPVE-Core on Windows.

1.1 Concept of the FPVE-Core

The FPVE-Core provides full disk encryption solution at the volume, or disk, or partition level. The data written to the volume is encrypted and decrypted upon read.

The FPVE-Core based volume protection encrypts volume partitions which contains sensitive information. The volume Encryption offered by FPVE Core is a data at rest type of protection.

The FPVE-Core is recommended in setups where access to volumes is managed by file system level permissions and only encrypted data needs to be stored on the volume.

1.2 Architecture of the FPVE-Core

The following diagram illustrates the architecture of the FPVE-Core.

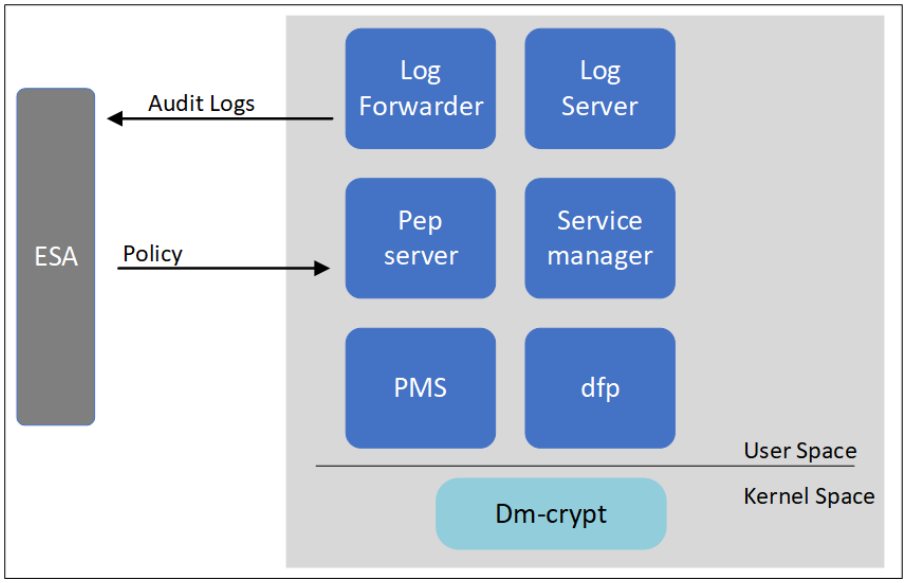

Figure 1-1: Architecture of the FPVE-Core

The key components of the FPVE-Core are:

- **Policy Management Server (PMS)**

  The PMS manages the policy on the FPVE node and communicates with the *dfp* utility to execute the commands triggered by dfp utility.

- **Log Server**

  The Log Server connects to the ESA and sends the audit logs to the ESA.

- **Service Manger**

  The Service Managers manages the start and stop services of the FPVE-Core.

- **PEP server**

  The PEP server is the policy enforcement point which connects to the ESA to receive the policy, key, and certificate related information.

- **Dm-crypt**

  The *Dm-crypt* is a transparent disk encryption subsystem available in the Linux. It is implemented as a device mapper target and is stacked on top of the other device mapper transformations. It can encrypt the block devices, partitions, software RAID (Redundant array of Independent Disks) volumes, and logical volumes.

1.3 How an Application Accesses an Encrypted Volume

This section describes the underlying operation that the FPVE- core performs to execute the volume encryption and key management commands. It also describes the workflow that an application must perform to access an encrypted volume.

The following diagram illustrates how an application accesses the encrypted volumes.

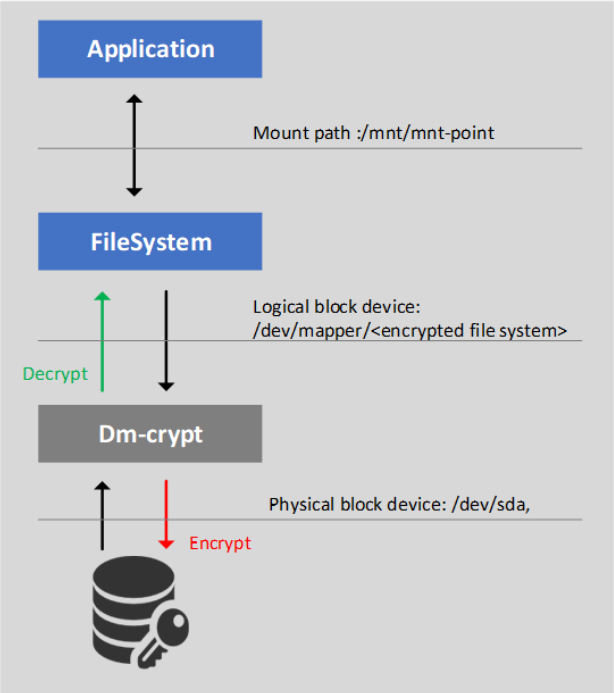

Figure 1-2: How an Application Accesses the Encrypted Volumes

The FPVE-Core maintains the key to mount a protected volume on a host system. After you mount the volume, all the data on the protected volume is available in clear form. If the data is written on the protected volume, then the written data is encrypted.

Workflow for Volume Protection

You must run the **`dfp volume`** commands to protect a volume. The **`dfp volume protect`** command internally perform the following operations:

1. Format the disk for *Linux Unified Key Setup (LUKS)* setup. After formating, all the data stored on the disk is lost. You must backup the data stored on the disk. Run the **`dfp volume protect`** command with the *–backup* option to copy all the disk data to a backup disk. If you run the **`dfp volume protect`** command with the *–erase* option, then no data backup is initiated, and the data volume is formated for *LUKS* setup.
2. Encrypt the volume with the generated key and protect the key using the data element key provided during the **`dfp volume protect`** command.
3. If the automount option is used, then the protected volume is mounted after the system restart without any manual effort.

Volume Encryption and Key Management

The FPVE-Core generates keys to encrypt the volumes. These keys are created when you run the protection command. The key is protected using the data element key provided with the protection command. The key information remains static and associated with the disk unless the volume is re-protected or unprotected.

1.4 Supported Platforms

The FPVE-Core supports the following platform:

- RHEL versions 6 and 7

1.5 Features of the FPVE-Core

The FPVE-Core provides the following features:

- Volume encryption and decryption
- Audit log

Chapter 2

Installing, Upgrading, and Uninstalling the FPVE-Core

This section describes the procedures to install, uninstall, and configure the FPVE-Core.

Perform the following tasks to install the FPVE-Core:

Task Order	Description	Reference
1.	Create a datastore in ESA.	For more information about creating a data store, refer to *Protegrity Policy Management Guide 9.1.0.5*.
2.	Run the LogForwarder	For more information about installing the LogForwarder, refer to *Installing the Log-Forwarder*.
3.	Run the PEP server installer.	For more information about installing the PEP server and extracting the certificate files, refer to *Installing the PEP Server*.
4.	Run the pre-install check.	For more information about the pre-install check, refer to *Running the FPVE-Core Pre-installation Check Script*.
5.	Run the FPVE-Core installation script.	For more information about installing the FPVE-Core, refer to *Running the FPVE-Core Installation Script*.

2.1 Installing the Log-Forwarder

This section describes the installation of the Log-Forwarder component.

2.1.1 Installing the Log-Forwarder

➤ To install the Log-Forwarder:

1. Run the following command.

```
LogforwarderSetup_Linux_x64_1.1.0+69.ga6521.1.1.sh
```

2. Enter the IP address and the port number for the audit store endpoint.
3. If you want add another audit store end point, enter *y*.
4. After successfully adding all the audit store endpoints, enter *y* to accept the installation process.
   The installation process begins.

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.
-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_1.1.0+69.ga6521.1.1.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_1.1.0+69.ga6521.1.1.sh -endpoint1 <ip address and port number> [-
endpoint2 <ip
address and port number>] [-endpoint3 <ip address and port number>] -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.

To install the PEP server:

1. Download the *FileProtector_RHEL-ALL-64_x86-64_FPVE-Core_x.x.x.x.tgz* installer on the system.
2. Extract the *FileProtector_RHEL-ALL-64_x86-64_FPVE-Core_x.x.x.x.tgz* file using the following command.
   ```
   tar -xvf FileProtector_RHEL-ALL-64_x86-64_FPVE-Core_x.x.x.x.tgz
   ```
   The following files are extracted:
   - *PepServerSetup_Linux_x64_x.x.x.x.sh*
   - *FileProtector_Linux_x64_FPVE-Core_x.x.x.x.sh*
   - *FileProtector_Linux_x64_PreInstallCheck_x.x.x.x.sh*
   - *INSTALL.txt*
3. Run the PEP server installation script using the following command.
   ```
   ./PepServerSetup_Linux_x64_x.x.x.x.sh
   ```
   A prompt for the ESA host name or IP address appears.
4. Enter the ESA host name or IP address.
5. Press **ENTER**.
   A prompt for the ESA user name appears.
6. Enter the ESA user name.
7. Press **ENTER**.
   A prompt for the ESA password appears to download the certificates.
8. Enter the ESA administrator password.
9. 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  29108      0  0:00:01  0:00:01 --:--:-- 29173

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

Protegrity PepServer installed in /opt/protegrity/defiance_dps.
```

2.3 Running the FPVE-Core Pre-installation Check Script

Before installing the FPVE-Core, you must run the pre-installation check script to verify the prerequisites are present.

To run the pre-installation check script:

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

```
./FileProtector_Linux_x64_PreInstallCheck_x.x.x.x.sh
```

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

```
[root@labrh7 ~]# ./FileProtector_Linux_x64_FPVE-Core_PreInstallCheck_9.0.0.1.sh

Enter ('Logforwarder') installation directory.
It should have the 'fluent-bit' sub directory
[/opt/protegrity]

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

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

2.4 Running the FPVE-Core Installation Script

This section describes the procedure to install the FPVE-Core.

To install the FPVE-Core:

1. Run the following script on the node.
   ./FileProtector_Linux_x64_FPVE-Core_x.x.x.x.sh
   A prompt to continue with the installation of the FPVE-Core appears.
2. If you want to install the FPVE-Core, then enter *yes*.
3. Alternatively, if you do not want to the FPVE-Core, then enter *no*.
4. Press **ENTER**.
   A prompt for the LogForwarder installation directory appears.
5. Enter the installation directory for the LogForwarder.
   The LogForwarder is installed in the `/opt/protegrity` directory by default.
6. Press **ENTER**.
   A prompt for the PEP server installation directory appears.
7. Enter the installation directory for the PEP server.
   The PEP server is installed in the `/opt/protegrity` directory by default.
8. Press **ENTER**.
   A prompt for the FPVE-Core installation directory appears.
9. Enter the installation directory for the FPVE-Core.
   A new directory is created in the `/opt/protegrity` installation directory by default.
10. Press **ENTER**.
    The installation of the FPVE-Core starts.
    The following files are extracted in the `/opt/protegrity/fileprotector/bin` directory:
    - *authldap.plm*
    - *dfp*
    - *dfpadmin*
    - *dfp_changepath*
    - *dfp_conf*
    - *dfp_convert_db.sh*

- *dfp_enforce_cleanup*
- *dfp_get_env.sh*
- *dfp_log*
- *dfp_log_server*
- *dfp_policy_management_server*
- *dfp_service_manager*
- *dfpshell*
- *dfp_uninstall*
- *dfp_uninstall_vec*
- *dfp_volume*

11. Enter a new *dfpshell* password.

> **Note:** The *dfpshell* is the system administrator shell for the FPVE-Core. 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.

12. Press **ENTER**.
    A prompt to verify the *dfpshell* password appears.
13. Re-enter the *dfpshell* password to verify.
14. Press **ENTER**.
    The following message appears.

```
Create dfpshell password successfully!

File Protector(FPVE-Core) installed in /opt/protegrity/fileprotector .
```

2.5 Introducing the *dfpshell*

The *dfpshell* is the system administrator shell for the FPVE-Core. It is a privileged mode of operation for the FPVE-Core management that requires a users to login using a *dfpshell* password.

You can create the *dfpshell* password when you install the FPVE-Core for the first time.

If you run the FPVE-Core 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 *FPVE-Core Commands Overview*.

The *dfpshell* command uses the following syntax:

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

Table 2-2: dfpshell Commands

Commands	Description
`dfpshell -t`	Checks if the current process has the *dfpshell* privilege. If the current process has the required privilege, then the following message appears:

Commands	Description

	`Has privilege!`
	
	Else, it displays the following error message:
	`INFO: No privilege!`

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

2.5.1 dfpshell Password Management

You can create a *dfpshell* password, reset the password, and activate or deactivate the *dfpshell* mode using the FPVE-Core.

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

2.5.2 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.

2.5.3 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`

2.6 Licensing for the FPVE-Core

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

The FPVE-Core license can have one of the following three states:

- Valid

- Expired
- Invalid

If the license is valid, then you can encrypt and decrypt a volume using the FPVE-Core.

If the license is expired, then you can only decrypt a encrypted volume using the FPVE-Core.

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 *Licensing Guide v9.0.0.0*.

2.6.1 Checking License Validity

The FPVE-Core provides the **`dfpadmin`** commands to check the license validity.

➤ To check validity of the FPVE-Core license:

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

```
dfpadmin license check
```

- If the license of the FPVE-Core is valid, then the following message appears.
  ```
  File Protector License is OK!
  ```
- If the license of the FPVE-Core is invalid, then the following message appears.
  ```
  File Protector license is invalid
  ```
- If the license of the FPVE-Core has expired, then the following message appears.
  ```
  File Protector license is expired
  ```

2.6.2 Checking License Status

This section describes how to check the license status of the FPVE-Core using the **`dfpadmin license status`** command.

➤ To view the status details of the FPVE-Core 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
```

2.6.3 Opeartions Denied for Invalid or Expired FPVE-Core License

If the license of FPVE-Core is invalid or expired, then you cannot encrypt a new volume.

2.6.4 Operations Allowed for Invalid or Expired FPVE-Core License

If the FPVE-Core license is invalid or expired, then due to policy rights, no errors are generated when you perform certain voliume protection operations.

If the license of FPVE-Core is expired or is invalid, then the following operations are allowed:

- Access encrypted volume
- Unprotect volumes

2.7 Uninstalling the FPVE-Core

This section describes the steps for uninstalling the FPVE-Core from a system.

To uninstall the FPVE-Core:

1. Run the following command to uninstall the FPVE-Core from a system.
   dfp_uninstall
2. Enter the *dfpshell* password.
3. Press **ENTER**.
   After the uninstallation of FPVE-Core is successful, then the following message appears.

```
Removing File-Protector(FPVE-Core) package...

--Stoping FP services, please wait ...

Shutting down              log :               [ OK ]
Shutting down              pms :               [ OK ]
File-Protector(FPVE-Core has been successfully removed.)
```

2.8 Upgrading the FPVE-Core to v9.0.0.0

This section describes the steps to upgrade the FPVE-Core from v7.1 or above to v9.0.0.0.

Depending on the FPVE-Core build version from which you are upgrading, you can upgrade the FPVE-Core in the following ways:

- Upgrading from a non-UUID version to v9.0.0.0: If you are upgrading the FPVE-Core from a non-UUID version (all versions below v6.6.5.22) to v9.0.0.0, then use this method.

For more information on upgrading the FPVE-Core from a non-UUID version, refer to the section *Appendix: Upgrading from a non-UUID version to v9.0.0.0*.

- Upgrading from a UUID version to v9.0.0.0: If you are upgrading the FPVE-Core from a UUID version (v6.6.5.22 and above) to v9.0.0.0, then use this method.

2.8.1 Upgrading from v7.x version to v9.0.0.0

This section describes the steps to upgrade from v7.1 or above to v9.0.0.0.

Before you begin
Before upgrading, ensure that the protected volume names are consistent between the system name and *VolEnc.db* file using the following command.

```
dfp volume stat /<protected volume name>
```

If the protected volumes name varies, then you need to run the **`dfp volume sync`** command to update the protected volume names in *VolEnc.db* file.

After the protected volume names are updated, verify the protected volume names are consistent between the system name and *VolEnc.db* file using the following command.

```
dfp volume stat /<protected volume name>
```

➤ To upgrade the FPVE-Core:

1. Extract the FPVE-Core package using the following command.
   ```
   tar -xvf FileProtector_RHEL-ALL-64_x86-64_FPVE-Core_x.x.x.x.tgz
   ```
2. Install the LogForwarder.
   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.

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

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

4. Run the pre-installation check script.
   For more information about running the pre-installation check script, refer to *Running the FPVE-Core Pre-installation Check Script*.

5. Run the following FPVE-Core installation script.
   ```
   ./FileProtector_Linux_x64_FPVE-Core_x.x.x.x.sh
   ```
   For more information about installing the FPVE-Core, refer to *Running the FPVE-Core Installation Script*.

6. If you need to upgrade the FPVE-Core, then type *yes*.
   Enter a *dfpshell* password.
7. Press **ENTER**.
8. Check the entries in the *VolumeEnc.db* and *automount.conf* files to verify the UUIDs using the following commands.

```
cat <File protector install path>/data/VolumeEnc.db
cat <File protector install path>/data/automount.conf
```

9. Restart the PMS server to update the protected volumes names in the *automount.conf* file using the following commands.

```
dfpadmin service pms off

dfpadmin service pms on
```

Chapter 3

FPVE-Core Commands Overview

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

3.1 dfp Commands

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

- Display the version of the FPVE-Core product
- Protect and unprotect existing volumes
- Display the status of encrypted volumes

The following snippet lists all the *dfp* commands of FPVE-Core.

```
[root@labrh7 ~]# dfp help
Usage:
    dfp start -p <role>@<policy>
    dfp info
    dfp proc [-l] [ <PID> ]
    dfp volume protect [-y] -p <policy> <-|passwd> -d <data element>
          <--erase-data | --backup <temp backup dir> >
          [--prev_exec <prev script>]
          [--post_exec <post script>]
          <-l <devices list file> | <device> [device2 ...] >

    dfp volume unprotect [-y] [-p <policy> <-|passwd>]
          <--erase-data | --backup <temp backup dir> >
          [--prev_exec <prev script>]
          [--post_exec <post script>]
          <-l <device list file> | <device> [device2 ...] >

    dfp volume init -p <policy> [passwd] -d <data element> <device>
    dfp volume open [-p <policy> [passwd]] <device> <name>
    dfp volume close <name>
    dfp volume automount add [-f] <device> <auto mount> <volume type> <name> <mount point>
    dfp volume automount del <device> [<device2> ... ]
    dfp volume stat <device> [<device2> ...]
    dfp volume status
    dfp volume sync
    dfp volume version
    dfp volume help
    dfp version
    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 3-1: FPVE-Core 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 data elements of the policy in the process (bash shell).
`dfp info`	No	Displays product information, such as product version, and the available policies on the PEP server, along with data elements of the current process. **Note:** If the policy is loaded, then it displays the data elements in the policy.

`dfp proc [-l] [ <PID> ]`	No	Displays information about the data elements of a specified process.
		The following list specifies the command options and their meaning.
		• *-l* - Displays the following 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 process, whose data elements you want to view.

`dfp volume protect [-y] -p <policy> <-|passwd> -d <data element> <--erase-data | --backup <temp backup dir> > [--prev_exec <prev script>] [--post_exec <post script>] <-l <devices list file> | <device> [device2 ...] >`	Yes	Encrypts the existing volumes. **Note:** Ensure that you must run the `dfp volume protect` command in the following situations: • When the file system is ready in use and mounted • When you want to backup or erase data
`dfp volume unprotect [-y] [-p <policy> <-|passwd>] <--erase-data | --backup <temp backup dir> > [--prev_exec <prev script>] [--post_exec <post script>] <-l <device list file> | <device> [device2 ...] >`	Yes	Decrypts the protected volumes, When you want to backup or erase data.
`dfp volume init -p <policy> [passwd] -d <data element> <device>`	Yes	Formats and adds the encryption header to a device or volume for encryption after defining policy and data element. **Note:** Ensure that you must run the `dfp volume init` command in the following situations:

Commands	*dfpshell* Privilege	Description
		• When the file system is not in place
`dfp volume open [-p <policy> [passwd]] <device> <name>`	No	Accesses the device and maps to a logical device.
`dfp volume close <name>`	No	Removes the encrypted device mapping temporarily.
`dfp volume automount add [-f] <device> <auto mount> <volume type> <name> <mount point>`	Yes	Configures auto-mounting for the encrypted volumes.
`dfp volume automount del <device> [<device2> ... ]`	Yes	Deletes the auto-mount for the specified encrypted volumes.
`dfp volume stat <device> [<device2> ...]`	Yes	Shows the encrypted key status of the specified volumes.
`dfp volume status`	Yes	Shows the following information about the encrypted volume. • Device path • Encrypted status and data element • Mapper path • Mount point
`dfp volume sync`	Yes	Synchronize the system defined UUID (Universally Unique Identifier) and the protected volume names in the *VolumeEnc.db* file. **Note:** If the protected volume name changes, then you must run the **`dfp volume sync`** command to update the volume names in *VolumeEnc.db* file.
`dfp volume version`	No	Displays the FPVE-Core product version and copyright information.
`dfp volume help`	No	Displays all volume encryption commands and the options, and explanations for the commands.
`dfp version`	No	Displays the FPVE-Core *dfp* version.
`dfp help`	No	Displays the *dfp* commands and options of FPVE-Core.

3.2 dfpadmin Commands

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

The following snippet lists all the *dfpadmin* commands of FPVE-Core.

```
[root@labrh7 ~]# dfpadmin help
Usage:
    dfpadmin status
    dfpadmin update
    dfpadmin service <service name> [ on | off | status ]
    dfpadmin service all [ on | off | status ]
    dfpadmin database -o updatedb-policy-passwd -p <policy> [passwd]
    dfpadmin license check
    dfpadmin license status
    dfpadmin help
```

The following table describes the FPVE-Core admin commands and the *dfpshell* privileges required.

Table 3-2: FPVE-Core 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: • Information about the components in the product • Service information • Available policies • License status
`dfpadmin service <service name> [ on | off | status ]`	Yes	Starts or stops the specific service, or displays its status. *<service name>* - Includes pms and log. **Note:** The Process Management Service (PMS) monitors the processes that are created and terminated on the system. It also maintains the data structure for the delegated process and their related policies. **Note:** The Log service handles the audit logs and updates the log files.
`dfpadmin service all [ on | off | status ]`	Yes	• Starts or stops all the FPVE-Core services. • Displays the status of all the FPVE-Core services.
`dfpadmin database -o updatedb-policy-passwd -p <policy>`	Yes	Updates the *VolumeEnc.db* file if you change the policy password and deploy the policy on ESA.
`dfpadmin license check`	No	Checks the validity of the FPVE-Core 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 FPVE-Core *dfpadmin* commands.

3.3 Scenarios to Use dfp volume sync Command

The following section describes scenarios where you must run the `dfp volume sync` command to resolve an inconsistency between the protected volume name and UUID in the *VolumeEnc.db* file.

Table 3-3: Scenarios to Use the ***dfp volume sync*** *Command*

Current Command	Errors	Solution

dfp volume protect [-y] -p <policy> <-|passwd> -d <data element> <--erase-data | --backup <temp backup dir> > [--prev_exec <prev script>] [--post_exec <post script>] <-l <devices list file> | <device> [device2 ...] >	If a specified device name and the UUID in the *VolumeEnc.db* file does not match, then the following sample error appears.
	`[root@node2 ~]# dfp volume protect -y -p policy_fe --backup /home/joel /dev/sdg2`
	`ERROR: Inconsistent device name and uuid found, UUID corresponding to passed device name </dev/sdg2> exists in database.`
	`Database record: [/dev/sde2] [178a73ea-5d09-4542-a974-al2347bdl869]`
	`Please run: <dfp volume sync> to update the database.`
	`ERROR: failed to do volume protection on device </dev/sdg2>!`
		Run the ***dfp volume sync*** command to update the device name in the *VolumeEnc.db* file.
		For example,
		`[root@node2 ~]# dfp volume sync`
		`Enter dfpshell Pass Phrase: *`
		`Inconsistent UUID:<afbac3d7-6660-41d6-97d2-25a61702a3e7>, current device:</dev/sdf1>, database device:</dev/sddl>`
		`Inconsistent UUID:<178a73ea-5d09-4542-a974-al2347bdl869>, current device:</dev/sdg2>, database device:</dev/sde2>`
		`Inconsistent UUID:<078b792c-c8ed-4e88-9a09-f3c8e2271db4>, current device:</dev/sdc>, database device:</dev/sdh>.`
		`Do you want to continue? [yes/no] yes`
		`Database updated successfully ! !`

dfp volume unprotect [-y] [-p <policy> <-|passwd>] <--erase-data | --backup <temp backup dir> > [--prev_exec <prev script>] [--post_exec <post script>] <-l <device list file> | <device> [device2 ...] >	If a specified device name and the UUID in the *VolumeEnc.db* file does not match, then the following sample error appears.
	`[root@node2 ~]# dfp volume unprotect -y -p policy_fe --backup /home/joel /dev/sdg2`
	`ERROR: Inconsistent device name and uuid found, UUID corresponding to passed device name </dev/sdg2> exists in database.`
	`Database record: [/dev/sde2] [178a73ea-5d09-4542-a974-al2347bdl869]`
	`Please run: <dfp volume sync> to update the database.`
	`ERROR: failed to do volume unprotection on device </dev/sdg2>!`
		Run the ***dfp volume sync*** command to update the device name in the *VolumeEnc.db* file.
		For example,
		`[root@node2 ~]# dfp volume sync`
		`Enter dfpshell Pass Phrase: *`
		`Inconsistent UUID:<afbac3d7-6660-41d6-97d2-25a61702a3e7>, current device:</dev/sdf1>, database device:</dev/sddl>`
		`Inconsistent UUID:<178a73ea-5d09-4542-a974-al2347bdl869>, current device:</dev/sdg2>, database device:</dev/sde2>`
		`Inconsistent UUID:<078b792c-c8ed-4e88-9a09-f3c8e2271db4>, current device:</dev/sdc>, database device:</dev/sdh>.`
		`Do you want to continue? [yes/no] yes`
		`Database updated successfully ! !`

Current Command	Errors	Solution

dfp volume init -p <policy> [passwd] -d <data element> <device>	If a specified device name and the UUID in the *VolumeEnc.db* file does not match, then the following sample error appears.
	`[root@labrh73base-0 ~]# dfp volume init -p policy_fe -d aesl-rcwd /dev/sdhl`
	`Enter dfpshell Pass Phrase: *`
	`ERROR: </dev/sdhl> already active by </dev/mapper/sdhl-6242> which mounted on </emountl> Command Failed!`
		Run the ***dfp volume sync*** command to update the device name in the *VolumeEnc.db* file.
		For example,
		`[root@node2 ~]# dfp volume sync`
		`Enter dfpshell Pass Phrase: *`
		`Inconsistent UUID:<afbac3d7-6660-41d6-97d2-25a61702a3e7>, current device:</dev/sdf1>, database device:</dev/sddl>`
		`Inconsistent UUID:<178a73ea-5d09-4542-a974-al2347bdl869>, current device:</dev/sdg2>, database device:</dev/sde2>`
		`Inconsistent UUID:<078b792c-c8ed-4e88-9a09-f3c8e2271db4>, current device:</dev/sdc>, database device:</dev/sdh>.`
		`Do you want to continue? [yes/no] yes`
		`Database updated successfully ! !`

dfp volume automount add [-f] <device> <auto mount> <volume type> <name> <mount point>	If a specified device name and the UUID in the *VolumeEnc.db* file does not match, then the following sample error appears.
	`[root@labrh73base-0 ~]# dfp volume automount add -f /dev/sdhl yes volume gvol /gmount`
	`ERROR: Inconsistent device name and uuid found, UUID corresponding to passed device name </dev/sdhl> exists in database. Database record: [/dev/sdel] [24fc9967-401e-4012-bd74-f938d08f4elf]`
	`Please run: <dfp volume sync> to update the database.`
		Run the ***dfp volume sync*** command to update the device name in the *VolumeEnc.db* file.
		For example,
		`[root@node2 ~]# dfp volume sync`
		`Enter dfpshell Pass Phrase: *`
		`Inconsistent UUID:<afbac3d7-6660-41d6-97d2-25a61702a3e7>, current device:</dev/sdf1>, database device:</dev/sddl>`
		`Inconsistent UUID:<178a73ea-5d09-4542-a974-al2347bdl869>, current device:</dev/sdg2>, database device:</dev/sde2>`
		`Inconsistent UUID:<078b792c-c8ed-4e88-9a09-f3c8e2271db4>, current device:</dev/sdc>, database device:</dev/sdh>.`
		`Do you want to continue? [yes/no] yes`
		`Database updated successfully ! !`

dfp volume automount del <device> [<device2> ...]	If a specified device name and the UUID in the *VolumeEnc.db* file does not match, then the following sample error appears. `[root@labrh73base-0 ~]# dfp volume automount del /dev/sdhl`	Run the ***dfp volume sync*** command to update the device name in the *VolumeEnc.db* file.

Current Command	Errors	Solution
	ERROR: Inconsistent device name and uuid found, UUID corresponding to passed device name </dev/sdhl> exists in database. Database record: [/dev/sdel] [24fc9967-401e-4012-bd74-f938d08f4elf] Please run: <dfp volume sync> to update the database.	For example, [root@node2 ~]# dfp volume sync Enter dfpshell Pass Phrase: * Inconsistent UUID:<afbac3d7-6660-41d6-97d2-25a61702a3e7>, current device:</dev/sdf1>, database device:</dev/sddl> Inconsistent UUID:<178a73ea-5d09-4542-a974-al2347bdl869>, current device:</dev/sdg2>, database device:</dev/sde2> Inconsistent UUID:<078b792c-c8ed-4e88-9a09-f3c8e2271db4>, current device:</dev/sdc>, database device:</dev/sdh>. Do you want to continue? [yes/no] yes Database updated successfully ! !
dfp volume stat <device> [<device2> ...]	If a specified device name and the UUID in the *VolumeEnc.db* file does not match, then the following sample error appears. [root@node2 ~]# dfp volume stat /dev/sdg2 ERROR: Inconsistent device name and uuid found, UUID corresponding to passed device name </dev/sdg2> exists in database. Database record: [/dev/sde2] [178a73ea-5d09-4542-a974-al2347bdl869] Please run: <dfp volume sync> to update the database.	Run the ***dfp volume sync*** command to update the device name in the *VolumeEnc.db* file. For example, [root@node2 ~]# dfp volume sync Enter dfpshell Pass Phrase: * Inconsistent UUID:<afbac3d7-6660-41d6-97d2-25a61702a3e7>, current device:</dev/sdf1>, database device:</dev/sddl> Inconsistent UUID:<178a73ea-5d09-4542-a974-al2347bdl869>, current device:</dev/sdg2>, database device:</dev/sde2> Inconsistent UUID:<078b792c-c8ed-4e88-9a09-f3c8e2271db4>, current device:</dev/sdc>, database device:</dev/sdh>. Do you want to continue? [yes/no] yes Database updated successfully ! !

Chapter 4

Features of the FPVE-Core

Supported Features of FPVE-Core

The following table describes the list of features supported by FPVE-Core:

Table 4-1: List of features

Features	Refer to
Protecting a volume	*Encrypting a Volume*
Unprotecting a Volume	*Decrypting a Volume*
FPVE-Core Audit Logging	*FPVE-Core Audit Logging*

4.1 Encrypting Volumes

The FPVE-Core provides transparent volume encryption to protect volumes.

The FPVE-Core utilizes open source Dm-crypt tools and provides policy based volume level encryption.

4.1.1 Encrypting a Volume

You can encrypt an existing volume using the FPVE-Core. The *dfpshell* privilege is necessary to run the volume protect command.

➤ To protect an existing volume:

Run the following command.

```
dfp volume protect [-y] -p <policy> <-|passwd> -d <data element> <--erase-data | --backup <temp backup dir> > [--prev_exec <prev script>] [--post_exec <post script>] <-l <devices list file> | <device> [device2 ...] >
```

Where,

- *-y* - Specifies yes for all interactive questions that occur during the volume protect command execution, and is optional.
- *-p <policy>* - Specifies the policy name.
- *-d <data element>* - Provides the <data element name>.

- *<--erase-data | --backup <temp backup dir> >* - Provides option to erase your data in the target volume by *--erase-data*, or backup your data to a specified temp directory by *--backup <temp backup dir>* before the volume protect.
- *[--prev_exec <prev script>]* - Specifies a script to be executed before executing the volume protect command and is optional.
- *[--post_exec <post script>]* - Specifies a script to be executed after executing the volume protect command and is optional.
- *< -l <devices list file> | <device> [device2 ...] >* - Provides option to specify a file included in a list of devices names. In the file, every device name is included as an absolute path in a separate line.

  For example, add the following content to a file.

  /dev/sda1

  /dev/sdb2

  /dev/sdc3

  Or select to use <device> [device2 ...] to manually specify the devices names for volume protection.

4.1.2 Decrypting a Volume

You must run *volume unprotect* command to decrypt an encrypted volume.

To unprotect a protected volume:

Run the following command.

```
dfp volume unprotect [-y] [-p <policy> <-|passwd>] <--erase-data | --backup <temp backup dir> > [--prev_exec <prev script>] [--post_exec <post script>] <-l <device list file> | <device> [device2 ...] >
```

Where,

- *-y* - Specifies yes for all interactive questions that occur during the volume unprotect command execution, and is optional.
- *[-p <policy> <-|passwd>]* - Provides option to use the policy and policy password, if no data element of the policy is loaded in the current session. This is an optional parameter.
- *<--erase-data | --backup <temp backup dir> >* - Provides option to erase your data in the target volume by *--erase-data*, or backup your data to a specified temp directory by *--backup <temp backup dir>* before the volume unprotect.
- *[--prev_exec <prev script>]* - Specifies a script to be executed before running the *volume unprotect* command.
- *[--post_exec <post script>]* - Specifies a script to be executed after running the *volume unprotect* command.
- *< -l <devices list file> | <device> [device2 ...] >* - Provides option to specify a file included in a list of devices names. In the file, every device name is included as an absolute path in a separate line.

  For example, add the following content to a file.

  /dev/sda1

  /dev/sdb2

  /dev/sdc3

Or select to use <device> [device2 ...] to manually specify the devices names for volume unprotection.

> **Note:** Do not use the `dfp unprotect` command to unprotect the Logical Volume Group.

4.2 FPVE-Core Audit Logging

The FPVE-Core monitors the security operations and logs in an audit log.

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

- Install and uninstall the FPVE-Core
- Load a policy
- Enter a privileged shell
- Mount and unmount a volume
- Protect a volume
- Unprotect a volume
- Initialise a volume

The following events are generated for the audit configuration:

Events	Description
INSTALL	Install the FPVE-Core
UNINSTALL	Uninstall the FPVE-Core
DFPSHELL	Login the *dfpshell* and change the *dfpshell* password operation
LOAD_POLICY	Load the policy for the user
VE_PROTECT	Protect a volume
VE_UNPROTECT	Unprotect a volume
VOLUM_MNT	Add a volume in automount.conf file
VOLUM_UNMNT	Remove a volume from automount.conf file
VOLUM_INI	Initialise a volume for volume encryption.

4.2.1 Log Message Format

The following examples show the local logging (ESA) message format:

```
<Login ID>: <Time stamp> : <UserID> : <Operation> :(operation result) : <File Type> :
<Process ID> : <opeartion>
```

For example:

```
[596516649] Tue Jul 10 03:41:55.219 2018 [EDT] root:VE_PROTECT:
(Success):ISDEV:3764:volume:Volume Protect, device(/dev/sdb) with data element(rcwd)
successfully
```

```
[260401255] Mon Feb 4 15:56:00.618 2013 [CST] root: DFPSHELL: (Success): ISREG: 12916:
dfpshell: Load privilege
```

The following table describes various log message parameters:

Table 4-2: Log Message Parameters

Log Parameters	Explanation
<Time Stamp>	Time zone, date, and time
<LoginID or userID>	User name or user ID. If kernel cannot get the user name, then use the *UID[ID].*
<Operation>	Operation type such as VE_UNPROTECT etc.
<Operation result>	Operation result, access, or operation success or failure
<File Type>	File type such as ISREG, ISDEV etc.
<process ID>	Process ID
<opeartion>	Operation performed on a volume

4.3 Limitation of the FPVE-Core

This section describes limitation of the FPVE-Core.

1. The GFS2 file system is not supported.

Chapter 5

Metering for the FPVE-Core

The Metering feature counts the number of successful protect and unprotect operations on file basis.

The ESA, which is connected to the protectors in the production environment of customers collates the total count of successful protect and unprotect operations per file, as reported by the FPVE-Core. As part of Protegrity Prime, these counts need to be shared with Protegrity by generating the Metering report, from the ESA Web UI.

The pricing model for Protegrity Prime customers is derived from these reported counts containing the number of successful protect and unprotect operations performed on each and directory.

The following table describes how the metering count is estimated for various scenarios of the FPVE-Core.

Table 5-1: Metering Count Estimation

Protect Count	Unprotect Count	Note
The protect count is incremented based on the total number of files and directories are created on a protected volume. All system files, which are generated by an operating system, are also counted.	The unprotect count is incremented based on the total number of files and directories are unprotected on a protected volume.	The reprotect count is not included for the FPVE-Core.

5.1 Generating the Metering Report

The Metering report, which is available on the ESA, can be generated using the Web UI.

> **Note:** Ensure that you are assigned the *Custom Business Manager* role to generate a metering report.

➤ To generate the Metering Report:

1. On the ESA Web UI, navigate to **Analytics** > **Reporting** > **Reports**.
2. Click **Download Report**.

   The Metering report is created by collating all successful protect and unprotect operations on each file that are reported by the FPVE-Core.

   The Protegrity Metering report includes the information as per the following table.

Attribute	Description
Description	The description provided when generating the report
Hostid	The ESA host ID assigned as a part of the licensing requirement
Created	The timestamp for the report specifying the date and time of report creation
Metadata	The metadata information includes the following attributes: **hostname**: ESA host name **ip**: ESA IP address **platform**: ESA platform **version**: ESA platform version
Integrity	A check to determine if any modifications are done to the ESA repository where counts are stored and report is generated
Date	The date and month for which the collective counts are recorded
Node information	The metadata information includes the following attributes: **uid**: Unique identifier for the node **hostname**: Node host name **ip**: Node IP **platform**: Node platform **version**: Node version (PEP version)
Protect	The details that are sent to ESA for all successful protect operations per file **metering**: The total count of protected files till date **delta**: The delta indicating the count of protected files for the node in a month
Unprotect	The details that are sent to ESA for all unprotect operations per file **metering**: The total count of unprotected files till date **delta**: The delta indicating the count of unprotected files for the node in a month
Reprotect	Not Applicable
Signature	The signature for the Metering report that can help validate if the report has been tampered

A sample of the Metering report is provided in the following snippet:

```
{
  "description" : "",
  "hostid" : "",
  "created" : "",
  "metadata" : {
    "hostname" : "",
    "ip" : "",
    "platform" : "",
    "version" : ""
  },
  "integrity" : "ok",
  "dates" : [ {
    "date" : "2018-03"
  }, {
    "nodes" : [ {
      "uid" : "",
      "metadata" : {
        "hostname" : "",
        "ip" : "",
        "platform" : "",
        "version" : ""
      },
      "protect" : {
        "metering" :,
        "delta" :
      },
      "unprotect" : {
        "metering" :,
        "delta" :
      },
      "reprotect" : {
        "metering" :,
        "delta" :
      }
    } ]
  } ],
  "signature" : ""
}
```

Chapter 6

Migrating a FPVE Volume to the FPVE-Core

This section describes the procedure to migrate the kernel-based FPVE to FPVE-Core.

Before you begin

Before starting migration, ensure that the volumes are added in the *automount.conf* and *VolumeEnc.db* files to mount the protected volumes after restart.

> **Note:** If the volumes are not added in the *automount.conf* file, then the volumes will not be mounted after restart.

> **Note:** It is recommended not to upgrade from FP to FPVE-Core. You must uninstall the existing File Protector and then install the FPVE-Core.

> **Note:**
>
> Before migrating from FPVE to FPVE-Core, ensure that you consider the following points:
>
> - **Access Control**
>
> The access control feature is available in the kernel-based FPVE. In case of FPVE-core, the access control feature is not supported as the product is designed to encrypt volume only. If the volumes are encrypted by the FPVE-Core, then it can be accessed by any user who have the required file system permissions.
>
> - **Encryption compatibility**
>
> The volumes, which were encrypted by the kernel-based FPVE, are fully compatible with the FPVE-Core. The user is not required to re-protect the protected volumes after migration.

To migrate from FPVE to FPVE-Core:

1. Uninstall the File Protector v6.6.5.
2. Restart the system.
3. Ensure that the *VolumeEnc.db* file is available in the `/opt/protegrity/fileprotector/data/` directory.
4. Install the FPVE-Core v7.1 on the system.

   After the successful installation, the FPVE-Core migrates the required database and configuration files from the FPVE.

   > **Note:** For more information about installation of the FPVE-Core, refer to *Installing, Upgrading, and Uninstalling the FPVE-Core*.

5. Run the following command to ensure that all services are running.

   ```
   dfpadmin service all status
   ```

   The following services should be running:

- dfp_service_manager
- dfp_policy_management_server
- dfp_log_server

6. Restart the PMS server to update the protected volumes names in the *automount.conf* file using the following commands.

```
dfpadmin service pms off

dfpadmin service pms on
```

After successful migration, all the encrypted disks are automatically mounted.

Chapter 7

Use Cases for the FPVE-Core

This section describes how you can protect a physical or logical volume using the FPVE-Core.

Depending upon the mount status of a volume, you can protect the volume in the following ways:

- If a volume is mounted, then it is recommended to run the **`dfp volume protect`** command to protect the physical or logical volume.
- If a volume is not mounted, then it is recommended to run the **`dfp volume init`** and **`dfp volume open`** command to protect the physical or logical volume.

7.1 Protecting a Physical Volume Using the *dfp volume protect* Command

The following section describes how to protect a physical volume, which can include partitions, using the **`dfp volume protect`** command.

Before you begin
Before running the *dfp volume protect* Command ensure that the following prerequisites are met:

- Ensure that the disk or partition is mounted. If the disk or partition is not mounted, then you must run the following command to mount.
  `mount<disk name><directory name>`
- Ensure that you must backup the existing data available on the disk or partition.

To protect a physical partition or volume:

1. Protect the disk or partition using the following command.
   `dfp volume protect -p <policy name> -d <data element name> --<backup path> <disk or partition name>`

   For example, **`dfp volume protect -p policy_fe -d aes1-rcwd --backup /home/admin /dev/sdc1`**

   Where,

/home/admin is the backup path used to backup the existing data available in the */dev/sdc1* partition. In this case, */dev/sdc1* is the newly created partition.

The ***dfp volume protect*** command automatically adds the protected volumes in the *automount.conf* file. The protected volumes are available after system restart.

> **Note:** The *dfpshell* privilege is required to execute the ***dfp volume protect*** command.

> **Note:** The ***dfp volume protect*** command provides a backup or erase option to backup and erase data from the disk or partition before encryption.

> **Note:** Ensure that the backup disk has the same capacity or additional capacity than the data available on the existing drive or partition.

2. Verify the encrypted volume status using the following command.
   dfp volume status
   The following result appears.

```
# dfp volume status
Enter dfpshell Pass Phrase: *
DevicePath       EncryptedStatus(DE)              MapperPath            MountPoint
/dev/sdc1        Encrypted-Mounted(aes1-rcwd)    /dev/mapper/sdc1-5862              /mnt
```

   The volume </dev/sdc1> is encrypted and mounted to */mnt* directory.

> **Note:** The *dfpshell* privilege is required to execute the ***dfp volume status*** command.

After a successful volume encryption, you can read the protected data in clear format. If you try to write on the protected disk, then it appears as encrypted.

7.2 Protecting a Physical Volume Using the *dfp volume init* and *dfp volume open* Command

The following section describes how to protect a physical volume, which can include partitions, using the *dfp volume init* and *dfp volume open* Commands.

Before you begin
Before running the *dfp volume init* and *dfp volume open* Commands ensure that the following prerequisites are met:

- Ensure that the physical volume is not mounted.
- Ensure that you do not have any existing data on the physical volume for backup.

  > **Note:** The ***dfp volume open*** command provides the option to create your own mapper device.

- Ensure that the protected volume is not automatically mounted after system restart. If you want the protected data available after system restart, you must run the following command to add the protected volume into automount configuration list.
  dfp volume automount add -f <device absolute path> yes volume <mapper device name> <mount point>

To protect a physical partition or volume:

1. Check the list of physical volumes for volume encryption using the following command.
   fdisk -l
2. Check the mount status of the physical volumes using the following command.
   df -Th
3. Initialize the volume for protection using the following command.
   dfp volume init -p <policy name> -d <data element name> <Device abosolute path>
   This command formats the disk for encryption format.

   For example,

   ```
   dfp volume init -p policy_fpveCore holeinone1 -d aes1-rcwd /dev/sdd1
   It will destory all data on the device </dev/sdd1>.
   ```
4. Enter the *dfpshell* password.
5. Press **ENTER**.
6. If you want to Initialize the volume, then enter *yes*.
   The following message appears.
   Command successfully!
7. Alternatively, if you do not want to Initialize the volume, then enter *no*.
8. Protect the volume using the following command.
   dfp volume open -p policy_fpveCore holeinone1 /dev/sdd1 dvol
   This command maps the *LUKS* disk device under the new disk device */dev/mapper* directory.

   For example,

   ```
   dfp volume open -p policy_fpveCore holeinone1 /dev/sdd1 dvol
   device-mapper: remove ioctl failed: Device or resource busy
   Key slot 0 unlocked.
   The opened volume is: /dev/mapper/dvol !
   Command successfully!
   ```

   > **Note:** The *dfpshell* privilege is required to execute the ***dfp volume status*** command.
9. Format the mapper device to access the protected volume using the following command.
   mkfs.ext3 /dev/mapper/dvol
   Where,

   /dev/mapper/dvol is a mapper device
10. Mount the volume for automount configuration setup using the following command.
    mount /dev/mapper/dvol /lmount
    Where,

    /dev/mapper/dvol is a mapper device

    /lmount is a mount path

11. Add the protected volume in the automount configuration list using the following command.
    dfp volume automount add -f <Protected volume> yes volume <Mapper device name> <Mount path>
    dfp volume automount add -f /dev/sdd1 yes volume dvol /dmount

    Where,

    /dev/sdd1 directory is a protected volume

    dvol is a mapper device name

    /dmount is a mount path

12. Verify the status of the encrypted volume using the following command.
    dfp volume status

    > **Note:** The *dfpshell* privilege is required to execute the ***dfp volume protect*** command.

13. Enter the *dfpshell* password.
14. Press **ENTER**.
    The following message appears.

```
DevicePath       EncryptedStatus(DE)              MapperPath            MountPoint
-------------------------------------------------------------------------------------
-
/dev/sdd1    Encrypted-Mounted(aes1-rcwd)          /dev/mapper/sdd1-5862          /
dmount
```

After a successful volume encryption, you can read the protected data in clear format. If you try to write on the protected disk, then it appears as encrypted.

7.3 Protecting a Logical Volume Using the *dfp volume protect* Command

The following section describes how to protect a logical volume using the ***dfp volume protect*** command.

Before you begin
Before running the *dfp volume protect* Command ensure that the following prerequisites are met:

- Ensure that the logical volume is mounted. If the logical volume is not mounted, then you must run the following command to mount.
  mount<disk name><directory name>

- Ensure that you must backup the existing data available on the disk or partition.

➤ To protect a logical volume:

1. Check the list of logical volumes using the following command.
   lsblk <Physical volume name>

For example, the following snippet describes the result of the ***lsblk*** command.

```
[root@labrh7 ~]# lsblk /dev/sdc
NAME                MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sdc                   8:32   0  16G  0 disk
└sdc1                 8:33   0  16G  0 part
  └LVMVolGroup-lvm 253:2    0   5G  0 lvm
```

2. Format the logical volume with the required file system using the following command.

   mkfs.<File system type> <Absolute path of the logical device>

   For example, the following snippet describes the result of the ***mkfs*** command.

```
[root@labrh7 ~]#mkfs.ext4 /dev/LVMVolGroup/lvm
The following message appears.
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
327680 inodes, 1310720 blocks
65536 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1342177280
40 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

3. Mount the logical volume using the following command.

   mount /dev/LVMVolGroup/lvm /mnt

4. Protect the logical volume using the following command.

   dfp volume protect -p policy_fe holeinone1 -d aes1-rcwd --backup <Backup path> <Absolute path of the logical device>

```
[root@labrh7 ~]# dfp volume protect -p policy_fe holeinone1 -d aes1-rcwd  --backup /home/
admin/ /dev/LVMVolGroup/lvm
```

   Where, the */home/admin* directory is the backup path and the */dev/LVMVolGroup/lvm* is the LVM volume.

5. Enter the *dfpshell* password.
6. Press **ENTER**.

   The following message appears.

```
[Protect] Protect volume </dev/LVMVolGroup/lvm> with data element <aes1-rcwd> starting ...
[Protect] Doing Prev-Check
    |__ [OK] check if backup directory under device </dev/LVMVolGroup/lvm>!
    |__ [OK] check if device </dev/LVMVolGroup/lvm> key file exist!
    |__ [OK] check if device </dev/LVMVolGroup/lvm> in use!
    |__ [OK] check file system on device </dev/LVMVolGroup/lvm>!
    |__ [OK] check backup size of directory </home/admin>!
    |__ [OK] check backup protection of directory </home/admin>!
    |__ [OK] check the integrity of backup archive file </home/admin/LVMVolGroup/
lvm_fpve_protect_backup_archive_2018-08-22-08-32-31.cpio>!
[Protect] *NOTE*: volume </dev/LVMVolGroup/lvm> data backup in archive file
          </home/admin/LVMVolGroup/
lvm_fpve_protect_backup_archive_2018-08-22-08-32-31.cpio> successfully!
[Protect] Doing Prev-Check ... [OK]
```

```
[Protect] Doing Volume-Encryption
It will destory all data on the device </dev/LVMVolGroup/lvm>.
```

7. If you want to encrypt the volume, then enter *yes*.

   The following message appears.

```
[Protect] Doing Volume-Encryption ... [OK]
[Protect] Doing Post-Others
Key slot 0 unlocked.
    |__ [OK] post recovery data to device </dev/LVMVolGroup/lvm>!
Key slot 0 unlocked.
[Protect] Doing Post-Others ... [OK]
[Protect] Add </dev/LVMVolGroup/lvm> to </opt/protegrity/fileprotector/data/
automount.conf> successfully!
The volume </dev/LVMVolGroup/lvm> protected by data element <aes1-rcwd> successfully!
```

8. Alternatively, if you do not want to encrypt the volume, then enter *no*.
9. Verify the status of the encrypted volume using the following command.

   dfp volume status

   > **Note:** The *dfpshell* privilege is required to execute the ***dfp volume protect*** command.

10. Enter the *dfpshell* password.
11. Press **ENTER**.

    The following message appears.

```
DevicePath       EncryptedStatus(DE)             MapperPath            MountPoint
-------------------------------------------------------------------------------------
-
/dev/LVMVolGroup/lvm    Encrypted-Mounted(aes1-rcwd)     /dev/mapper/
lvm-963   /mnt
```

After a successful volume encryption, you can read the protected data in clear format. If you try to write on the protected disk, then it appears as encrypted.

7.4 Protecting a Logical Volume Using the *dfp volume init* and *dfp volume open* Command

The following section describes how to protect a logical volume using the *dfp volume init* and *dfp volume open* commands.

Before you begin

Before running the *dfp volume init* and *dfp volume open* Commands ensure that the following prerequisites are met:

- Ensure that the logical volume is not mounted.
- Ensure that you do not have any existing data on the logical volume for backup.

  > **Note:** The ***dfp volume open*** command provides the option to create your own mapper device.

- Ensure that the protected volume is not automatically mounted after system restart. If you want the protected data available after system restart, you must run the following command to add the protected volume into automount configuration list.

  dfp volume automount add -f <device absolute path> yes volume <mapper device name> <mount point>

➤ To protect a logical volume:

1. Check the list of logical volumes using the following command.

   lsblk <Physical volume name>

   For example, the following snippet describes the result of the ***lsblk*** command.

   ```
   [root@labrh7 ~]# lsblk /dev/sdc
   NAME                MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   sdc                   8:32   0  16G  0 disk
   ↳sdc1                 8:33   0  16G  0 part
     ↳LVMVolGroup-lvm 253:2    0   5G  0 lvm
   ```

2. Format the LVM volume for volume encryption using the following command.

   mkfs.ext3 /dev/LVMVolGroup/lvm

3. Initialize the volume for protection using the following command.

   dfp volume init -p <policy name> -d <data element name> <Absolute path of the logical volume>

   This command formats the disk and creates the *LUKS* signature on the disk.

   For example,

   ```
   dfp volume init -p policy_fe -d rcwd /dev/LVMVolGroup/lvm
   It will destory all data on the device </dev/LVMVolGroup/lvm>.
   ```

4. If you want to Initialize the volume, then enter *yes*.

   The following message appears.

   Command successfully!

5. Alternatively, if you do not want to Initialize the volume, then enter *no*.
6. Protect the volume using the following command.

   dfp volume open -p <policy name> <Absolute path of the logical device> <mapper device name>

   This command maps the *LUKS* disk device under the new disk device */dev/mapper* directory.

   For example, the following snippet describes the result of the ***dfp volume open*** command.

   ```
   dfp volume open -p policy_fe /dev/LVMVolGroup/lvm lv1
   Open it successfully
   Key slot 0 unlocked.
   The opened volume is: /dev/mapper/lv1!
   Command successfully!
   ```

7. Format the mapper device to access the protected volume using the following command.

   mkfs.ext3 /dev/mapper/lv1

   Where,

   /dev/mapper/lv1 is a mapper device

8. Mount the volume for automount configuration setup using the following command.

   mount /dev/mapper/lv1 /lmount

   Where,

   /dev/mapper/lv1 is a mapper device

/lmount is a mount path

9. Add the protected volume in the automount configuration list using the following command.

   dfp volume automount add -f <Protected volume> yes volume <Mapper device name> <Mount path>

   dfp volume automount add -f /dev/LVMVolGroup/lvm yes volume lv1 /lmount

   Where,

   /dev/LVMVolGroup/lvm directory is a protected volume

   lv1 is a mapper device name

   /lmount is a mount path

10. Verify the status of the encrypted volume using the following command.

    dfp volume status

    > **Note:** The *dfpshell* privilege is required to execute the ***dfp volume protect*** command.

11. Enter the *dfpshell* password.
12. Press **ENTER**.

    The following message appears.

```
DevicePath       EncryptedStatus(DE)              MapperPath            MountPoint
-------------------------------------------------------------------------------------
-
/dev/LVMVolGroup/lvm    Encrypted-Mounted(aes1-rcwd)    /dev/mapper/lv1           /
lmount       -                      -
```

After a successful volume encryption, you can read the protected data in clear format. If you try to write on the protected disk, then it appears as encrypted.

Chapter 8

Troubleshooting

This section describes the various problem or error that the user may encounters while working with FPVE-Core.

8.1 Resolving the FPVE-Core Pre-Installation Check Script Fail Error During Upgradation

If the protected volume names are inconsistent for the device names available on the *VolumeEnc.db* file and current protected volume names, then the pre-installation check script fails.

For example, the following snippet describes the error message that appears after running the pre-installation check script.

```
[root@labrh6u8Base FPVE_core_new]# ./FileProtector_Linux_x64_PreInstallCheck_x.x.x.x.sh
Enter ('PepServer') installation directory.
It should have the 'defiance dps' sub directory
[ /opt/protegrity]

Found: /opt/protegrlty/flleprotector/data/VolumeEnc.db
Executing :Sanity checks for protected volumes present in /opt/protegrity/fileprotector/data/
VoluaeEnc.db

Sanity checks for protected volumes Failed.
Please backup the VoluaeEnc.db and automount.conf, update the <PATH Value> in both
VolumeEnc.db and automount.conf to the appropriate device path.

Following devices fron VolumeEnc.db does not exist
 /dev/sdf3
Following devices exists in VoluaeEnc.db but are not protected.
/dcv/sdi
/dev/sdj
Possible Reason: Device name changed due to changes in system configuration OR old stale
entry.
Unfortunately, File Protector Pre-Installation Checking Failed!
```

➤ To resolve the FPVE-Core pre-installation check script fail error:

1. Backup the *VolumeEnc.db* and *automout.conf* files to restore at a later point in time.
2. After backup, update the the modified entries in the *VolumeEnc.db* and *automout.conf* files manually.
3. After modifying the entries, run the pre-installation check script to ensure that no inconsistencies exist in the device names between the *VolumeEnc.db* file and current volume names.
4. Run the following FPVE-Core installation script.

```
./FileProtector_Linux_x64_FPVE-Core_x.x.x.sh
```

For more information about installing the FPVE-Core, refer to *Running the FPVE-Core Installation Script*.

5. Restart the PMS server to update the protected volumes names in the *automount.conf* file using the following commands.

```
dfpadmin service pms off

dfpadmin service pms on
```

Chapter 9

Appendix: Upgrading from a non-UUID version to v9.0.0.0

This section describes the steps to upgrade the FPVE-Core from a non-UUID version to v9.0.0.0.

> **Caution:**
> There are few scenarios where the volume name may change due to system change. If the name of the protected volume is changed, then you need to manually update the entries for the volume name in both the *volumeEnc.db* and *automount.conf* files.

To upgrade the FPVE-Core:

1. Extract the FPVE-Core package using the following command.
   ```
   tar -xvf FileProtector_RHEL-ALL-64_x86-64_FPVE-Core_x.x.x.tgz
   ```
2. Install the LogForwarder.
   For more information about installing the LofForwarder, refer to *Installing the Log-Forwarder*.
   > **Note:** If the PEP server is already installed and running, then skip this step.
3. Install the PEP Server.
   For more information about installing the PEP Server, refer to *Installing the PEP Server*.
   > **Note:** If the PEP server is already installed and running, then skip this step.
4. Run the pre-installation check script.
   If no inconsistency is found for the device names available on the *VolumeEnc.db* file and current protected volume names, then the pre-installation check script is successful.
   If the protected volume names are inconsistent for the device names available on the VolumeEnc.db file and current protected volume names, then refer to *Resolving the FPVE-Core Pre-Installation Check Script Fail Error During Upgradation*.

   For more information about running the pre-installation check script, refer to *Running the FPVE-Core Pre-installation Check Script*.
5. Run the following FPVE-Core installation script.
   ```
   ./FileProtector_Linux_x64_FPVE-Core_x.x.x.sh
   ```

For more information about installing the FPVE-Core, refer to *Running the FPVE-Core Installation Script*.

> **Note:** Before upgrading, ensure that the pre-installation check is successful. If you upgrade the FPVE-Core without fixing the pre-installation check errors, then the upgrade fails, and the following message appears.

```
[root@labrh6u8base FPVE_core_new]#./FileProtector_Linux_x64_FPVE-Core_x.x.x.x.sh
PrelnstallCheck completed with errors. Please fix errors and again run PrelnstallCheck
```

6. If you need to upgrade the FPVE-Core, then type *yes*.

   Enter a new *dfpshell* password.

7. Press **ENTER**.

   The following message appears.

```
-- Stopping the services, please wait ...
Shutting down                      log :             [ OK ]
Shutting down                      pms :             [ OK ]
Unpacking.. .
Extracting files...
--Installed /opt/protegrity/fileprotector/bin/authldap.plm
--Installed /opt/protegrity/fileprotector/bin/dfp
--Installed /opt/protegrity/fileprotector/bin/dfpadmin
--Installed /opt/protegrity/fileprotector/bin/dfp_changepath
--Installed /opt/protegrity/fileprotector/bin/dfp_conf
--Installed /opt/protegrity/fileprotetor/bin/dfp_convert_db.sh
--Installed /opt/protegrity/fileprotector/bin/dfp_enforce_cleanup
--Installed /opt/protegrlty/fileprotector/bin/dfp_get_env.sh
--Installed /opt/protegrity/fileprotector/bin/dfp_log
--Installed /opt/protegrity/fileprotector/bin/dfp_log_server
--Installed /opt/protegrity/fileprotector/bin/dfp_policy_management_server
--Installed /opt/protegrity/fileprotector/bin/dfp_service_manager
--Installed /opt/protegrity/fileprotector/bin/dfpshell
--Installed /opt/protegrity/fileprotector/bin/dfp uninstall
--Installed /opt/protegrity/fileprotector/bin/dfp_uninstall_vec
--Installed /opt/protegrity/fileprotector/bin/dfp_volume
File Protector(FPVE-Core) Installed In /opt/protegrlty/fileprotector.
```

8. Check the entries in the *VolumeEnc.db* and *automount.conf* files to verify the UUIDs using the following commands.

```
cat <File protector install path>/data/VolumeEnc.db
cat <File protector install path>/data/automount.conf
```

9. Restart the PMS server to update the protected volumes names in the *automount.conf* file using the following commands.

```
dfpadmin service pms off

dfpadmin service pms on
```

Chapter 10

Glossary

Mapper Device

The Mapper Device is a Linux kernel framework to map a physical block device on a higher-level virtual block device.

LUKS

The Linux Unified Key Setup or LUKS is a disk encryption specification for Linux.