Encrypting virtual pattern data with IBM Encryption Pattern for Security First SPxBitFiler-IPA

Note

Before using this information and the product it supports, read the information in “Notices” on page 35.

Contents

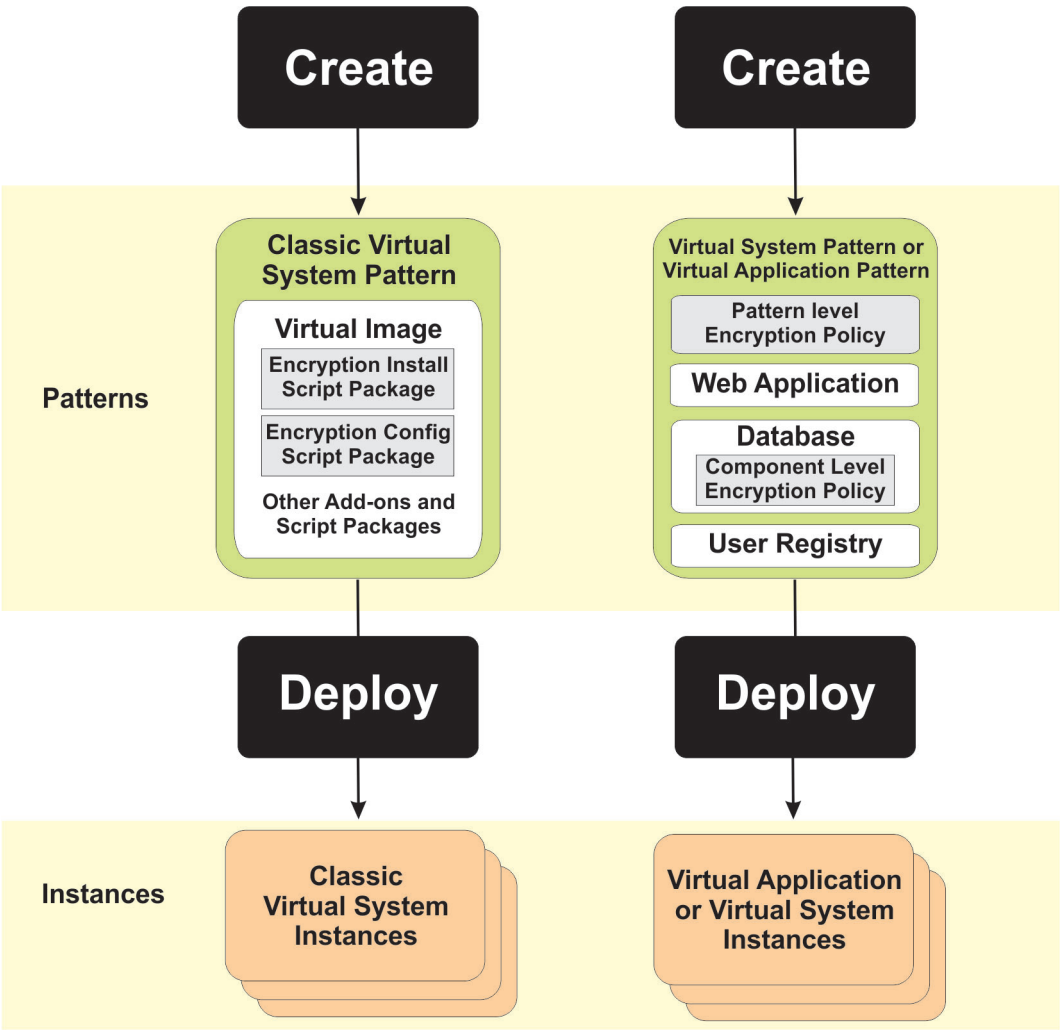
Encrypting data for virtual patterns

You can encrypt data that is stored on disk for your virtual patterns by creating encryption policies, plug-ins, and script packages. You then include these artifacts in your virtual application patterns, virtual system patterns, and classic virtual system patterns. When you deploy these patterns, the encryption software is installed and configured to encrypt the data that is located in the directory paths that you specify.

Overview

Security First SPxBitFiler-IPA is an application that is installed through the provided encryption package. You can use this application in your IBM® PureApplication® System environment to encrypt data-at-rest (on-disk data) associated with deployed classic virtual systems. The Security First SPxBitFiler-IPA software is added to your virtual application patterns, virtual system patterns, and classic virtual system patterns as part of the encryption pattern deployment. When these patterns are deployed into your IBM PureApplication System environment, the encryption software is installed and configured to encrypt the data-at-rest that is used by your virtual machines.

Figure 1. Creating and deploying virtual patterns with encryption capability



This figure illustrates where encryption installation and configuration script packages are used in classic virtual system patterns to provide encryption capability for data associated with deployed classic virtual system instances. For virtual system patterns and virtual application patterns, encryption policies can be added at the pattern level to apply to all components in the pattern that do not already have an encryption policy added at the local component level.

Supported platforms

Security First SPxBitFiler-IPA encryption software is available for the following supported operating system platforms:

Table 1. Supported operating system platforms

Operating System	Version
Red Hat Enterprise Linux (RHEL) (select kernel versions)	6.2
	6.3
	6.4
AIX®	6.1 (running Technology Level 4)
	7.1 (running Technology Level 1 or later)

Table 1. Supported operating system platforms (continued)

Operating System	Version
Microsoft Windows Server	2008-R2
	2012

To access all of the kernel versions of supported Red Hat Enterprise Linux platforms, and to install the latest supported versions, the RedHat OS Update Service must be running in your IBM PureApplication System environment. The RedHat OS Update Service is a shared service provided with IBM PureApplication System that is based on the Red Hat Update Infrastructure (RHUI) model.

To verify that this shared service is running, complete the following steps:

1. From the IBM PureApplication System Workload Console, click **Instances** > **Shared Services**.
2. Verify that at least one RedHat OS Update Shared Service instance is deployed and running.

   If there are no deployed and running instances of this shared service, refer to the IBM PureApplication System information center for details on deploying the shared service.

In addition, even though the RedHat OS Update Shared Service is deployed and running, you also need to verify that Security First packages are being replicated. Complete the following steps:

1. From the Workload Console, click **Instances** > **Shared Services**.
2. Select an instance of the RedHat OS Update Shared Service.
3. Click **Manage** to open the Virtual Application Console.
4. Click **Operations**.
5. From the list of operations, select the **RHUSService-RHUA.RHUA** operation.**Operations**.
6. In the **Operation Execution Results** pane, verify if there is a recent entry, *Replicate Security First packages (Optional)* and verify that the results indicate that the replication completed successfully. If so, then Security First replication should already be enabled.
7. If there is no indication that replication of Security First packages is occurring, then enable replication at this time. Expand the **Replicate Security First (Optional)** node and click **Submit** to perform the replication operation.

For more information about the RedHat OS Update Service, see the IBM PureApplication System information center section about working with shared services.

For more information about specific kernel versions that are supported for these operating system versions, see the Security First web site: http://www.securityfirstcorp.com/SPxBitFiler-IPA-Kernels.php

Downloading the IBM Encryption Pattern for Security First SPxBitFiler-IPA files

You must download the compressed files that are needed to import the pattern type and create the encryption script packages.

Procedure

1. Search for the file downloads on IBM PureSystems® Centre, using the following link:
   https://www.ibm.com/software/brandcatalog/puresystems/centre/browse#rc=PureApplication&page=1
2. Click **PureApplication System**, under the **IBM PureSystems** section.
3. Click **IBM**, under the **Providers** section.
4. Find **IBM Encryption Pattern for Security First SPxBitFiler**.
5. Click the link to download the package containing the compressed files.

Results

The package contains the following files:

securityfirst-1.0.0.1.tgz
Use this file to add the Security First pattern type to the PureApplication System catalog.

securityfirst-install-aix-1.0.0.1.tgz
Use this file to upload into an installation script package for AIX that you will create and add to the catalog.

securityfirst-config-aix-1.0.0.1.tgz
Use this file to upload into a configuration script package for AIX that you will create and add to the catalog.

securityfirst-install-linux-1.0.0.1.tgz
Use this file to upload into an installation script package for Linux that you will create and add to the catalog.

securityfirst-config-linux-1.0.0.1.tgz
Use this file to upload into a configuration script package for Linux that you will create and add to the catalog.

securityfirst-install-windows-1.0.0.1.zip
Use this file to upload into an installation script package for Windows that you will create and add to the catalog.

securityfirst-config-windows-1.0.0.1.zip
Use this file to upload into a configuration script package for Windows that you will create and add to the catalog.

Adding the pattern type to the catalog

Before you can deploy virtual patterns with encryption, you must add the Security First pattern type to the PureApplication System catalog.

Before you begin

You must be a PureApplication System administrator with the Workload resources administration role with Full permission to add a new pattern type to the catalog.

Procedure

1. Open the **Workload Console**.
2. Select **Cloud** > **Pattern Types**.
3. Click the **+** icon to create a new pattern type.
4. Click **Browse** to locate and select the `securityfirst-1.0.0.1.tgz` compressed file that you downloaded previously, and click **OK**.
5. Select **SecurityFirst Pattern Type 1.0.0.1** from the pattern type list.
6. In the **License Agreement** field, click **License** to read the licensing agreement. Click **Accept** if you agree with the licensing agreement.
7. In the **Status** field, click **Enable** to enable the **SecurityFirst Pattern Type 1.0.0.1** pattern type.
8. In the **System Plug-ins** field, click **Show me all plug-ins in this pattern type**.
9. In the **System Plug-ins** page, select the **securityfirst 1.0.0.1** plug-in from the system plug-ins list.
10. Click **Configure**, and configure the **Key Manager Type** parameter to specify the key manager that will be used with virtual system pattern and virtual application pattern data encryption.

    Initially, this parameter is set to *default* signifying that the default key manager provided by PureApplication System will be used for virtual system pattern and virtual application pattern data encryption.

    If you want to use a different key manager for encrypting data, you can leave this page, and click **System** > **Key Manager Adapters** to display the list of available key managers that have been registered with PureApplication System, or you can register your own key manager. Ensure that the key manager adapter that you choose is already installed. Return to this page and enter the name of the selected key manager in the **Key Manager Type** field to configure that parameter for the plug-in, and click **Update** to save your changes.

    This selection of the key manager type is not used when encrypting data for classic virtual system patterns. However, you must still configure this plug-in because other parts of it are used by the encryption script packages that are added to classic virtual system patterns. For classic virtual system patterns, you specify the key manager later, as part of the encryption installation script package when you add it to your classic virtual system patterns, or at deployment time.

Results

The Security First pattern type is now available for use with virtual application patterns, virtual system patterns, and also referenced by classic virtual system patterns.

Adding the encryption policy to virtual application patterns and virtual system patterns

You must add the **SecurityFirst Encryption Policy** to your virtual application patterns and virtual system patterns to encrypt data-at-rest associated with the pattern when it is deployed.

Procedure

1. Select **Patterns** > **Virtual Applications** or **Patterns** > **Virtual Systems**.
2. Select the pattern to which the encryption policy is to be added.
3. Click **Open** to open the **Pattern Builder**.
4. Add the **SecurityFirst Encryption Policy** to your pattern. You can add the policy in two ways:
   - Add the policy globally to the entire pattern by clicking **Add policy to pattern** and selecting **SecurityFirst Encryption Policy**.
   - Add the policy locally to a component in the pattern by clicking the **+** icon on a selected component in the topology, and selecting **SecurityFirst Encryption Policy** from the list of available policies.

   If encryption policies exist both locally at the component level and globally at the pattern level, components with a locally added policy use that policy, while other components that do not have a locally added policy use the global policy.

   When you select either the globally added encryption policy or a locally added encryption policy, the **SecurityFirst Encryption Policy** section is displayed to the right of the topology canvas, showing the following fields for you to configure one or more directory paths for encryption:
   - Paths to encrypt:
   - Paths to decrypt:
   - Paths to blacklist:
   - Paths to remove from blacklist:

   Important: When you create a virtual system pattern, you will see the option to use either script packages or the policy to enable the encryption. You must use the policy to enable the encryption. The script packages do not work for virtual system patterns.

   If you promote a classic virtual system pattern that uses the script packages, the system generates a virtual system pattern that uses script packages. Encryption for the promoted virtual system pattern will not work unless you modify the pattern to use the policy after it is promoted.
5. For each directory path whose files are to be encrypted, type the absolute directory path in the **Paths to encrypt** field. Examples:
   - For Linux or AIX: `/tmp/DirA/myFiles/`
   - For Windows: `C:\temp\DirA\myFiles\`

   If entering more than one directory path, separate each absolute directory path with a comma (do not insert a space before each comma, and do not end a directory path with a forward slash (/) character). For example:
   - For Linux or AIX: `/tmp/DirA/myFiles, /tmp/DirB/yourFiles`
   - For Windows: `C:\temp\DirA\myFiles, C:\temp\DirB\yourFiles`

   Note: As you define your directory paths for encryption, keep the following considerations in mind:
   - You must specify absolute directory paths for encryption using the syntax of the underlying operating system. For example:
     - For Linux or AIX: `/tmp/DirA`
     - For Windows: `C:\temp\DirA`

For simplicity, most examples that follow use Linux and AIX (forward slash) formatting. You should apply appropriate syntax for Windows as needed.

- Security First SPxBitFiler-IPA is intended to cryptographically split data files, and is not intended to be applied to program executable program files. For example, if you are applying encryption to DB2® version 9.7 files:
  - The default directory for database files is `/db/db2ins/db2ins/NODE0000`. Files in this directory are safe for encryption.
  - The default directory for executable files is `/opt/ibm/db2/V9.7/bin`. Files in this directory should not be encrypted, because they are executable programs.

  Using an executable that is cryptographically split by SPxBitFiler-IPA is not supported and might result in incorrect operation of the running virtual machine.

  As another example for virtual system DB2 virtual machines, if you add the appropriate `/bin` directory path (such as `/opt/ibm/db2/V10.1/bin`) to the blacklist, you can encrypt files in `/opt/ibm` and `/opt/IBM` directory paths.

  You should avoid encrypting `/bin` directories because they usually include executable files, not just data files.

Important: Avoid encrypting directory paths that are used during system startup (boot and login processes). For example, the following directory paths should not be encrypted:

- `/root`
- `/home/virtuser`
- `/opt/IBM/WebSphere`

  Note, most subdirectories can be safely encrypted, such as `/opt/IBM/WebSphere/Profiles/Default/AppSrv01/installedApps`.

  For virtual machines created from deployed classic virtual system patterns, directory paths such as the following examples are safe to encrypt:
  - `/opt/ibm`
  - `/opt/ibm/ae`

Similarly, avoid encrypting temporary directories. For example, do not encrypt files in the following WebSphere® temporary directory: `/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/wstemp`

6. Within the directory paths to be encrypted, if there are subdirectories containing files that should not be encrypted, type the absolute subdirectory path in the **Paths to blacklist** field to exclude that directory from encryption .

   For example, suppose the following directory path is designated for encryption: `/tmp/DirA/myFiles`, and within the `/myFiles` directory there exist two subdirectories, `/myFiles/myDir1` and `/myFiles/myDir2`.

   Exclude the files in `/myDir2` from being encrypted while allowing the files in `/myDir1` to be encrypted by specifying the directory paths as follows:

   - **Paths to encrypt**: `/tmp/DirA/myFiles`
   - **Paths to blacklist**: `/tmp/DirA/myFiles/myDir2`

   Note: Separate multiple directory paths in the **Paths to blacklist** field with a comma. Also, do not end directory paths with a forward slash (/) character.

   The first time that you are specifying directory paths to encrypt or blacklist, the two remaining fields, **Paths to decrypt** and **Paths to remove from blacklist** do not apply and can be left blank. Later, when you change the encryption configuration, you can specify directory paths in these fields to decrypt, or remove directory paths from the blacklist so that the files in those directory paths are encrypted.
7. Save your changes and close the **Pattern Builder**.

Results

Your pattern is now configured with an encryption policy, and is ready for deployment.

Deploying your virtual application pattern

Deploy your virtual application pattern to the cloud.

Procedure

1. Click **Patterns** > **Virtual Applications**.
2. Click **Deploy** in the **Actions** column for the pattern that you want to deploy, or select the pattern that you want to deploy and click **Deploy** near the top of the details page.

On the **Configure** pane:

3. Edit the name for the deployment, if needed. This name displays on the Instances page after the pattern deploys.
4. Select the environment profile that you want to use for the deployment.
   - If the network for the selected environment profile is set to **Internally Managed**:
     - Select a **Cloud Group** and an **IP Group**.

       Note: You can also set the IP group for each virtual machine on the **Distribute** pane. You cannot change the **Cloud Group** for the deployment after you configure it on this pane.
     - The deployment is limited to a single cloud group.
   - If the network for the selected environment profile is set to **Externally Managed**, you select the cloud group and IP group for the deployment later, on the **Distribute** pane.
5. Set the priority for the deployment.

   Note: For more information about deployment priorities, see the **Related tasks**.
6. Optional: To set up SSH access, use one of the following options in the **SSH Key** section to set the public key:
   - To generate a key automatically, click **Generate**. Click **Download** to save the private key file to a secure location. The default name is `id_rsa.txt`.

     The system does not keep a copy of the private key. If you do not download the private key, you cannot access the virtual machine, unless you generate a new key pair. You can also copy and paste the public key into a text file to save the key. Then, you can reuse the same key pair for another deployment. When you have the private key, make sure that it has the correct permissions (`chmod 0400 id_rsa.txt`). By default, the SSH client does not use a private key file that provides open permission for all users.
   - To use an existing SSH public key, open the public key file in a text editor and copy and paste it into the **SSH Key** field.

     Important: Do not use **`cat`**, **`less`**, or **`more`** to copy and paste from a command shell. The copy and paste operation adds spaces to the key that prevent you from accessing the virtual machine.
7. Modify the deployment schedule as needed:
   - Choose **Start now**, or choose **Start later** and select a date and time for the deployment to start.
   - Choose **Run indefinitely**, or choose **Run until** and select a date and time for the deployment to end.
8. Modify the component attributes as needed.

   The attributes that display in the pattern configuration column are attributes from the components in the pattern that are not locked from editing. You can modify existing values or set values that were not specified during pattern creation. Be sure that all required fields have values. Components that have a blue dot next to the name contain required attributes that must be set before the pattern is deployed.
9. When you are finished configuring all of the fields on the **Configure** tab,

- if you chose an environment profile that does not have the **IP addresses provided by** field set to **Pattern Deployer** and you do not want to modify the placement, click **Quick Deploy**. If you choose this option, the deployment process starts. You do not need to complete any of the subsequent steps.
- if you chose an environment profile that has the **IP addresses provided by** field set to **Pattern Deployer**, or you want to modify the placement, click **Continue to distribute**.

On the **Distribute** pane:

The virtual machines in the deployment are placed in cloud groups by the system.

10. Optional: To modify the placement of the virtual machines, drag the virtual machines to different cloud groups.
    - If you drag a virtual machine cell that contains more than one virtual machine, you are prompted to select the number of virtual machines that you want to move. You must select the number from the list in the dialog. After you move a virtual machine to a different cell, the IP group assignments are set to default values. If needed, you can edit the virtual machine network settings in the next step to modify the IP group.
    - If you modify the placement of the virtual machines, the new placement is validated to ensure that the necessary resources and artifacts are available in the selected cloud group.
    - If there is a problem with the placement, a message is displayed. Resolve the issue with the placement before you continue.

      For example, if this message displays when you modify the placement: `CWZKS7002E Insufficient memory to place the pattern`, move the virtual machine to a different cloud group with sufficient memory resources for the pattern.

      If you see the error: `Unable to assign to cloud group`, there is an error with the location, cloud group, NIC or IP groups for the cell where the error is displayed. If this error message occurs, you must resolve the issue with that cell before you are allowed to drag a virtual machine to that cell for placement there. Hover your mouse over the error to display more details about the problem in a pop-up window.
11. Optional: To edit the network or storage volume settings for a virtual machine, hover your mouse over the virtual machine icon and click the pencil icon.
    a. On the **IP Groups** tab, you can modify the IP group for each of the NICs in the virtual machine. The IP groups that are listed are the IP groups that are associated with the environment profile that you chose for the deployment.
    b. If you are deploying the IBM General Parallel File System (GPFS) Pattern, you can modify the storage volumes for the virtual machine on the **Storage Volumes** tab.
    c. Click **OK** when you are finished updating the settings.
12. When you are finished modifying the settings, click **Deploy**.

What to do next

When the virtual application is deployed, the resulting virtual application instance is listed under the **Instances** section of the workload console. To view the virtual instance, select **Instances** > **Virtual Applications**.

After deployment, you can modify the encryption configuration settings as needed. See “Modifying the encryption configuration for a deployed virtual application or virtual system” on page 14 for more information.

Deploying your virtual system pattern

Deploy your virtual system pattern to the cloud.

Procedure

1.
2. Click **Deploy** in the **Actions** column for the pattern that you want to deploy, or select the pattern that you want to deploy and click **Deploy** on the toolbar.

On the **Configure** pane:

3. Edit the name for the deployment, if needed. This name displays on the Instances page after the pattern deploys.
4. Select the environment profile that you want to use for the deployment.
   - If the network for the selected environment profile is set to **Internally Managed**:
     – Select a **Cloud Group** and an **IP Group**.

       Note: You can also set the IP group for each virtual machine on the **Distribute** pane. You cannot change the **Cloud Group** for the deployment after you configure it on this pane.
     – The deployment is limited to a single cloud group.
   - If the network for the selected environment profile is set to **Externally Managed**, you select the cloud group and IP group for the deployment later, on the **Distribute** pane.
5. Set the priority for the deployment.

   Note: For more information about deployment priorities, see the **Related tasks**.
6. Optional: To set up SSH access, use one of the following options in the **SSH Key** section to set the public key:
   - To generate a key automatically, click **Generate**. Click **Download** to save the private key file to a secure location. The default name is `id_rsa.txt`.

     The system does not keep a copy of the private key. If you do not download the private key, you cannot access the virtual machine, unless you generate a new key pair. You can also copy and paste the public key into a text file to save the key. Then, you can reuse the same key pair for another deployment. When you have the private key, make sure that it has the correct permissions (`chmod 0400 id_rsa.txt`). By default, the SSH client does not use a private key file that provides open permission for all users.
   - To use an existing SSH public key, open the public key file in a text editor and copy and paste it into the **SSH Key** field.

     Important: Do not use **`cat`**, **`less`**, or **`more`** to copy and paste from a command shell. The copy and paste operation adds spaces to the key that prevent you from accessing the virtual machine.

   The SSH key provides access to the virtual machines in the cloud group for troubleshooting and maintenance purposes. See the topic, "Configuring SSH key-based access", for details about SSH key-based access to virtual machines.
7. Modify the deployment schedule as needed:
   - Choose **Start now**, or choose **Start later** and select a date and time for the deployment to start.
   - Choose **Run indefinitely**, or choose **Run until** and select a date and time for the deployment to end.
8. Modify the pattern and component attributes as needed.

   The attributes that display in the pattern configuration column are attributes from the pattern and components in the pattern that are not locked from editing. You can modify existing values or set values that were not specified during pattern creation. Be sure that all required fields have values. Components that have a blue dot next to the name contain required attributes that must be set before the pattern is deployed.

9. When you are finished configuring all of the fields on the **Configure** tab,
   - if you chose an environment profile that does not have the **IP addresses provided by** field set to **Pattern Deployer** and you do not want to modify the placement, click **Quick Deploy**. If you choose this option, the deployment process starts. You do not need to complete any of the subsequent steps.
   - if you chose an environment profile that has the **IP addresses provided by** field set to **Pattern Deployer**, or you want to modify the placement, click **Continue to distribute**.

On the **Distribute** pane:

The virtual machines in the deployment are placed in cloud groups by the system.

10. Optional: To modify the placement of the virtual machines, drag the virtual machines to different cloud groups.
    - If you drag a virtual machine cell that contains more than one virtual machine, you are prompted to select the number of virtual machines that you want to move. You must select the number from the list in the dialog. After you move a virtual machine to a different cell, the IP group assignments are set to default values. If needed, you can edit the virtual machine network settings in the next step to modify the IP group.
    - If you modify the placement of the virtual machines, the new placement is validated to ensure that the necessary resources and artifacts are available in the selected cloud group.
    - If there is a problem with the placement, a message is displayed. Resolve the issue with the placement before you continue.

      For example, if this message displays when you modify the placement: `CWZKS7002E Insufficient memory to place the pattern`, move the virtual machine to a different cloud group with sufficient memory resources for the pattern.

      If you see the error: `Unable to assign to cloud group`, there is an error with the location, cloud group, NIC or IP groups for the cell where the error is displayed. If this error message occurs, you must resolve the issue with that cell before you are allowed to drag a virtual machine to that cell for placement there. Hover your mouse over the error to display more details about the problem in a pop-up window.
11. To edit the network or storage volume settings for a virtual machine, hover your mouse over the virtual machine icon and click the pencil icon.
    a. On the **IP Groups** tab, you can modify IP group for each of the NICs in the virtual machine. The IP groups that are listed are associated with the environment profile that you chose for the deployment. If the **IP address provided by** field in the environment profile that you chose for the deployment is set to **Pattern Deployer**, you must set the IP address for each NIC in the deployment.
    b. If there is a **Default attach block disk** add-on in the pattern, you can modify the storage volumes for the virtual machine on the **Storage Volumes** tab. You can use an existing storage volume or create one to attach to the component during deployment. If you choose to create a new storage volume, configure these settings:

       Name Set the name for the storage volume.

       Description
       Optional. Set a description for the storage volume.

       Size (GB)
       Set the size for the storage volume, in GB.

       Volume Groups
       Select a volume group for the storage volume. A storage volume group is a logical grouping of volumes that can span workloads and cloud groups.
    c. Click **OK** when you are finished updating the settings.
12. When you are finished modifying the settings, click **Deploy**.

When the virtual system is deployed, the virtual system instance is listed under the **Instances** section of the IBM PureApplication System 8283. To view the virtual system instance, click **Instances** > **Virtual Systems**.

The virtual memory and virtual processor settings that are configured for the virtual images in the virtual system pattern must be met by the requirements for the software components in the pattern. If these requirements are not met, the deployment fails and an error message that lists the memory and processor requirements is displayed. If this error occurs, modify the processor and memory settings in the pattern so that the requirements are met, and deploy the pattern again.

What to do next

When the virtual system is deployed, the resulting virtual system instance is listed under the **Instances** section of the workload console. To view the virtual instance, select **Instances** > **Virtual Systems**.

After deployment, you can modify the encryption configuration settings as needed. See "Modifying the encryption configuration for a deployed virtual application or virtual system" on page 14 for more information.

Modifying the encryption configuration for a deployed virtual application or virtual system

You can modify the encryption settings for a deployed virtual application or virtual system.

Before you begin

Use these procedures to update the encryption configuration settings for a deployed virtual application or virtual system.

Attention: The Security First SPxBitFiler software relies on data in several installed directories that must be kept available:

- For AIX:

  The Security First SPxBitFiler software uses a driver file stored in the following directory:

  `/usr/lib/drivers/bfipa`

  The cryptographic keys by default are stored in the driver file.
- For Linux:

  The Security First SPxBitFiler software uses an XML configuration file stored in the following directory:

  `/etc/spxbitfiler-ipa/config/bitfiler-ipa.xml`

  The cryptographic keys for workgroups by default are stored in the following directory:

  `/etc/spxbitfiler-ipa`
- For Windows:

  The SPxBitFiler-IPA files are stored in the following directory:

  `C:\Program Files (x86)\Bitfiler-IPA`

If these directories are lost or deleted, ALL of the data stored in the share files will be inaccessible and considered useless. Take care to ensure that the data in these directories is available.

About this task

When you initially apply the data at rest encryption policy to your pattern prior to deployment, you specify the directory paths to encrypt and blacklist (see "Adding the encryption policy to virtual application patterns and virtual system patterns" on page 6 for specifying initial encryption configuration settings prior to deployment).

This initial configuration is applied to the virtual machines that are created at deployment time, as well as any new virtual machines that are created later due to scaling operations, workload balancing, or node recovery situations.

After deployment, you can modify your encryption configuration settings in several ways:

- You can modify the original encryption policy in the pattern and deploy the pattern again.
- You can modify the encryption configuration settings for virtual machines that are currently running in the instance by changing the **Update Encryption** section of the **Operations** tab. Note, however, that this type of change does not apply to any new virtual machines that are created due to scaling, workload balancing, or recovery operations, only to the virtual machines that are currently running.
- You can modify the encryption settings by using the **Fundamental > Configuration** section of the **Operations** tab. These settings are applied to any new virtual machines that are created due to scaling, workload balancing, and recovery operations. Note that if these configuration settings do not cancel out or reverse the initial encryption settings at deployment time, then the initial configuration settings are also applied to the new virtual machines.

Modifying the Update Encryption settings

Modify the encryption settings for a deployed virtual application or virtual system in the Update Encryption section.

About this task

Use this procedure to modify the encryption configuration settings for virtual machines that are currently running in the virtual application or virtual system instance.

Procedure

1. Click **Instances** > **Virtual Applications** or **Instances** > **Virtual Systems**.
2. From the list of available instances, select the virtual application instance or virtual system instance to modify.
3. Click **Manage**.
4. Click the **Operations** tab in the console.
5. Select the role that you want to modify. Typically the name has an extension such as `.ENCRYPTION-SECURITYFIRST`.
6. In the **Update Encryption** section, update the following fields as needed:
   - **Paths to blacklist**
   - **Paths to remove from blacklist**
   - **Paths to encrypt**
   - **Paths to decrypt**
7. Click **Submit**.

Modifying the Fundamental Configuration settings

Modify the encryption settings for a deployed virtual application or virtual system in the Fundamental Configuration section.

About this task

Use this procedure to modify the encryption configuration settings for virtual machines that are currently running in the virtual application or virtual system instance, as well as new virtual machines that are created after initial deployment, due to scaling, workload balancing, and recovery operations. These settings are applied along with the initial settings specified at pattern deployment time.

For example, if you initially encrypted files in `/opt/ibm/myFiles` at deployment time, and then later you changed the encryption configuration settings in the Fundamental Configuration fields to also encrypt files in `/opt/ibm/DirA`, the currently running virtual machines and future virtual machines that are created due to a scaling operation have files in both `/myFiles` and `/DirA` directories encrypted.

However, these configuration settings can also override the initial settings that were applied at deployment. For example, if you initially encrypted files in `/opt/ibm/myFiles`, but then changed the configuration by specifying to encrypt files in `/opt/ibm/DirA` and to decrypt files in `/opt/ibm/myFiles` in the Fundamental Configuration settings, the initial encryption settings for files in `/opt/ibm/myFiles` are effectively canceled out, and only files in `/opt/ibm/DirA` are encrypted in subsequent encryption operations.

Keep in mind the effect that your configuration changes can have on previous settings as you configure encryption in these fields.

Procedure

1. Click **Instances** > **Virtual Applications** or **Instances** > **Virtual Systems**.
2. From the list of available instances, select the virtual application instance or virtual system instance to modify.
3. Click **Manage**.
4. Click the **Operations** tab in the console.
5. Select the role that you want to modify. Typically the name has an extension such as `.ENCRYPTION-SECURITYFIRST`.
6. Click **Fundamental** > **Configuration**.
7. In the **Modify Encryption** section, update the following fields as needed:
   - **Paths to encrypt**
   - **Paths to decrypt**
   - **Paths to blacklist**
   - **Paths to remove from blacklist**

   Important: Configure these fields with the complete set of directory paths that you want to decrypt and encrypt, as you did when you configured these fields during deployment. This update is not a differential operation. You must specify the complete set of directory paths in these fields.
8. Click **Submit**.

Creating encryption script packages for classic virtual system patterns

You must create new encryption script packages for Security First installation and configuration to make them available to be added to your classic virtual system patterns.

Before you begin

You should have already completed the following tasks:

- You should have installed the virtual application pattern type and created your Security First encryption policy.
- Depending on your operating system, you should have downloaded the two compressed package files for Security First installation and configuration, as shown in the following table:

Table 2. Operating system and script packages

Operating System	Installation Files	Configuration Files
AIX	`securityfirst-install-aix-1.0.0.1.tgz`	`securityfirst-config-aix-1.0.0.1.tgz`
Red Hat Enterprise Linux	`securityfirst-install-linux-1.0.0.1.tgz`	`securityfirst-config-linux-1.0.0.1.tgz`
Windows Server	`securityfirst-install-windows-1.0.0.1.zip`	`securityfirst-config-windows-1.0.0.1.zip`

Creating the encryption installation script package for AIX

You must create a new encryption script package for Security First installation on AIX to make it available to be added to your classic virtual system patterns.

Procedure

1. Open the **Workload Console**.
2. Click **Catalog** > **Script Packages**.
3. Click the **+** icon to create a new encryption installation script package to add to the catalog.
4. Type a unique name for the new encryption installation script package in the **Script Name** field, and click **OK** to create your script package.

   Note: Provide a name that clearly describes the script package, such as `securityfirst-install-aix-1.0.0.1`.
5. In the **Script package file** field, click **Browse** and locate the `securityfirst-install-aix-1.0.0.1.tgz` file.
6. Click **Upload** to upload the `securityfirst-install-aix-1.0.0.1.tgz` file to your script package.
7. Click **Refresh** to refresh the display with the script package parameter values that are imported from the upload operation.

   Note: The **License agreement** field displays *Not accepted*. After refreshing the display, the **accept** link is displayed.
8. In the **License agreement** field, click **accept** to display the license agreement link for this script package.
9. Click the link to the license agreement and review the details and conditions of the agreement. If you agree, click **Accept** to accept the license agreement for this script package, and click **OK**. The **License agreement** field should change to *Accepted*. You can click the **view** link at any time to display the license agreement again.
10. Verify the contents of the following fields, which are populated from information in the `securityfirst-install-aix-1.0.0.1.tgz` file:
    - **Product IDs**: shows the product IDs, such as *5725-L53 (PVU license)*
    - **Environment**: shows a single environment variable, `SF_KEY_MANAGER`, which is set to an initial value of *default*. You can configure this value later if needed when you add this script package to your classic virtual system patterns.
    - **Working directory**: shows the default working directory, */tmp/securityfirst/install*.
    - **Logging directory**: shows the default directory where logs related to the script package execution are stored, */tmp/securityfirst/install/logs*.
    - **Executable**: shows the command to run the main script in the script package, and the directory path where the script is located, *python /tmp/securityfirst/install/install.py*.
    - **Executes**: shows the value, *at virtual system creation*.
    - **Operating system**: shows *Linux/Unix*.

    This completes the creation of the encryption installation script package.

Results

The `securityfirst-install-aix-1.0.0.1` script package is added to the catalog, and is available to be added to appropriate parts in your classic virtual system patterns.

Creating the encryption configuration script package for AIX

You must create a new encryption configuration script package for Security First configuration on AIX to make it available to be added to your classic virtual system patterns.

Procedure

1. Open the **Workload Console**.
2. Click **Catalog** > **Script Packages**.
3. Click the **+** icon to create a new encryption configuration script package to add to the catalog.
4. Type a unique name for the new encryption configuration script package in the **Script Name** field, and click **OK** to create your script package.

   Note: Provide a name that clearly describes the script package, such as `securityfirst-config-aix-1.0.0.1`.
5. In the **Script package file** field, click **Browse** and locate the `securityfirst-config-aix-1.0.0.1.tgz` file.
6. Click **Upload** to upload the `securityfirst-config-aix-1.0.0.1.tgz` file to your script package.
7. Click **Refresh** to refresh the display with the script package parameter values that are imported from the upload operation.

   Note: The **License agreement** field displays *Not accepted*. After refreshing the display, the **accept** link is displayed.
8. In the **License agreement** field, click **accept** to display the license agreement link for this script package.
9. Click the link to the license agreement and review the details and conditions of the agreement. If you agree, click **Accept** to accept the license agreement for this script package, and click **OK**. The **License agreement** field should change to *Accepted*. You can click the **view** link at any time to display the license agreement again.
10. Verify the contents of the following fields, which are populated from information in the `securityfirst-config-aix-1.0.0.1.tgz` file:
    - **Product IDs**: is left blank for this script package.
    - **Environment**: shows the following parameters that you will configure later to define the directory paths to be encrypted when you add this script package to your classic virtual system patterns:
      - `SF_PATH_TO_ENCRYPT`
      - `SF_PATH_TO_DECRYPT`
      - `SF_PATH_TO_BLACKLIST`
      - `SF_PATH_TO_REMOVE_BLACKLIST`
    - **Working directory**: shows the default working directory, */tmp/securityfirst/configure*.
    - **Logging directory**: shows the default directory where logs related to the script package execution are stored, */tmp/securityfirst/configure/logs*.
    - **Executable**: shows the command to run the main script in the script package, and the directory path where the script is located, *python /tmp/securityfirst/configure/configure.py*.
    - **Executes**: shows the value, *at virtual system creation and when I initiate it*.
    - **Save environment variables after post-deployment executions**: shows the value, *Yes*.
    - **Operating system**: shows *Linux/Unix*.

    This completes the creation of the encryption configuration script package.

Results

The `securityfirst-config-aix-1.0.0.1` script package is added to the catalog, and is available to be added to appropriate parts in your classic virtual system patterns.

Creating the encryption installation script package for Linux

You must create a new encryption script package for Security First installation on Linux to make it available to be added to your classic virtual system patterns.

Procedure

1. Open the **Workload Console**.
2. Click **Catalog** > **Script Packages**.
3. Click the **+** icon to create a new encryption installation script package to add to the catalog.
4. Type a unique name for the new encryption installation script package in the **Script Name** field, and click **OK** to create your script package.

   Note: Provide a name that clearly describes the script package, such as `securityfirst-install-linux-1.0.0.1`.
5. In the **Script package file** field, click **Browse** and locate the `securityfirst-install-linux-1.0.0.1.tgz` file.
6. Click **Upload** to upload the `securityfirst-install-linux-1.0.0.1.tgz` file to your script package.
7. Click **Refresh** to refresh the display with the script package parameter values that are imported from the upload operation.

   Note: The **License agreement** field displays *Not accepted*. After refreshing the display, the **accept** link is displayed.
8. In the **License agreement** field, click **accept** to display the license agreement link for this script package.
9. Click the link to the license agreement and review the details and conditions of the agreement. If you agree, click **Accept** to accept the license agreement for this script package, and click **OK**. The **License agreement** field should change to *Accepted*. You can click the **view** link at any time to display the license agreement again.
10. Verify the contents of the following fields, which are populated from information in the `securityfirst-install-linux-1.0.0.1.tgz` file:
    - **Product IDs**: shows the product IDs, such as *5725-L53 (PVU license)*
    - **Environment**: shows a single environment variable, `SF_KEY_MANAGER`, which is set to an initial value of *default*. You can configure this value later if needed when you add this script package to your classic virtual system patterns.
    - **Working directory**: shows the default working directory, */tmp/securityfirst/install*.
    - **Logging directory**: shows the default directory where logs related to the script package execution are stored, */tmp/securityfirst/install/logs*.
    - **Executable**: shows the command to run the main script in the script package, and the directory path where the script is located, *python /tmp/securityfirst/install/install.py*.
    - **Executes**: shows the value, *at virtual system creation*.
    - **Operating system**: shows *Linux/Unix*.

    This completes the creation of the encryption installation script package.

Results

The `securityfirst-install-linux-1.0.0.1` script package is added to the catalog, and is available to be added to appropriate parts in your classic virtual system patterns.

Creating the encryption configuration script package for Linux

You must create a new encryption configuration script package for Security First configuration on Linux to make it available to be added to your classic virtual system patterns.

Procedure

1. Open the **Workload Console**.
2. Click **Catalog** > **Script Packages**.
3. Click the **+** icon to create a new encryption configuration script package to add to the catalog.
4. Type a unique name for the new encryption configuration script package in the **Script Name** field, and click **OK** to create your script package.

   Note: Provide a name that clearly describes the script package, such as `securityfirst-config-linux-1.0.0.1`.
5. In the **Script package file** field, click **Browse** and locate the `securityfirst-config-linux-1.0.0.1.tgz` file.
6. Click **Upload** to upload the `securityfirst-config-linux-1.0.0.1.tgz` file to your script package.
7. Click **Refresh** to refresh the display with the script package parameter values that are imported from the upload operation.

   Note: The **License agreement** field displays *Not accepted*. After refreshing the display, the **accept** link is displayed.
8. In the **License agreement** field, click **accept** to display the license agreement link for this script package.
9. Click the link to the license agreement and review the details and conditions of the agreement. If you agree, click **Accept** to accept the license agreement for this script package, and click **OK**. The **License agreement** field should change to *Accepted*. You can click the **view** link at any time to display the license agreement again.
10. Verify the contents of the following fields, which are populated from information in the `securityfirst-config-linux-1.0.0.1.tgz` file:
    - **Product IDs**: is left blank for this script package.
    - **Environment**: shows the following parameters that you will configure later to define the directory paths to be encrypted when you add this script package to your classic virtual system patterns:
      - `SF_PATH_TO_ENCRYPT`
      - `SF_PATH_TO_DECRYPT`
      - `SF_PATH_TO_BLACKLIST`
      - `SF_PATH_TO_REMOVE_BLACKLIST`
    - **Working directory**: shows the default working directory, */tmp/securityfirst/configure*.
    - **Logging directory**: shows the default directory where logs related to the script package execution are stored, */tmp/securityfirst/configure/logs*.
    - **Executable**: shows the command to run the main script in the script package, and the directory path where the script is located, *python /tmp/securityfirst/configure/configure.py*.
    - **Executes**: shows the value, *at virtual system creation and when I initiate it*.
    - **Save environment variables after post-deployment executions**: shows the value, *Yes*.
    - **Operating system**: shows *Linux/Unix*.

    This completes the creation of the encryption configuration script package.

Results

The `securityfirst-config-linux-1.0.0.1` script package is added to the catalog, and is available to be added to appropriate parts in your classic virtual system patterns.

Creating the encryption installation script package for Windows

You must create a new encryption script package for Security First installation on Windows to make it available to be added to your classic virtual system patterns.

Procedure

1. Open the **Workload Console**.
2. Click **Catalog** > **Script Packages**.
3. Click the **+** icon to create a new encryption installation script package to add to the catalog.
4. Type a unique name for the new encryption installation script package in the **Script Name** field, and click **OK** to create your script package.

   Note: Provide a name that clearly describes the script package, such as `securityfirst-install-windows-1.0.0.1`.
5. In the **Script package file** field, click **Browse** and locate the `securityfirst-install-windows-1.0.0.1.zip` file.
6. Click **Upload** to upload the `securityfirst-install-windows-1.0.0.1.zip` file to your script package.
7. Click **Refresh** to refresh the display with the script package parameter values that are imported from the upload operation.

   Note: The **License agreement** field displays *Not accepted*. After refreshing the display, the **accept** link is displayed.
8. In the **License agreement** field, click **accept** to display the license agreement link for this script package.
9. Click the link to the license agreement and review the details and conditions of the agreement. If you agree, click **Accept** to accept the license agreement for this script package, and click **OK**. The **License agreement** field should change to *Accepted*. You can click the **view** link at any time to display the license agreement again.
10. Verify the contents of the following fields, which are populated from information in the `securityfirst-install-windows-1.0.0.1.zip` file:
    - **Product IDs**: shows the product IDs, such as *5725-L53 (PVU license)*
    - **Environment**: shows a single environment variable, `SF_KEY_MANAGER`, which is set to an initial value of *default*. You can configure this value later if needed when you add this script package to your classic virtual system patterns.
    - **Working directory**: shows the default working directory, *c:\temp\securityfirst\install*.
    - **Logging directory**: shows the default directory where logs related to the script package execution are stored, *c:\temp\securityfirst\install\logs*.
    - **Executable**: shows the command to run the main script in the script package, and the directory path where the script is located, *python c:\temp\securityfirst\install\install.py*.
    - **Executes**: shows the value, *at virtual system creation*.
    - **Operating system**: shows *Windows*.

    This completes the creation of the encryption installation script package.

Results

The `securityfirst-install-windows-1.0.0.1` script package is added to the catalog, and is available to be added to appropriate parts in your classic virtual system patterns.

Creating the encryption configuration script package for Windows

You must create a new encryption configuration script package for Security First configuration on Windows to make it available to be added to your classic virtual system patterns.

Procedure

1. Open the **Workload Console**.
2. Click **Catalog** > **Script Packages**.
3. Click the **+** icon to create a new encryption configuration script package to add to the catalog.
4. Type a unique name for the new encryption configuration script package in the **Script Name** field, and click **OK** to create your script package.

   Note: Provide a name that clearly describes the script package, such as `securityfirst-config-windows-1.0.0.1`.
5. In the **Script package file** field, click **Browse** and locate the `securityfirst-config-windows-1.0.0.1.zip` file.
6. Click **Upload** to upload the `securityfirst-config-windows-1.0.0.1.zip` file to your script package.
7. Click **Refresh** to refresh the display with the script package parameter values that are imported from the upload operation.

   Note: The **License agreement** field displays *Not accepted*. After refreshing the display, the **accept** link is displayed.
8. In the **License agreement** field, click **accept** to display the license agreement link for this script package.
9. Click the link to the license agreement and review the details and conditions of the agreement. If you agree, click **Accept** to accept the license agreement for this script package, and click **OK**. The **License agreement** field should change to *Accepted*. You can click the **view** link at any time to display the license agreement again.
10. Verify the contents of the following fields, which are populated from information in the `securityfirst-config-windows-1.0.0.1.zip` file:
    - **Product IDs**: is left blank for this script package.
    - **Environment**: shows the following parameters that you will configure later to define the directory paths to be encrypted when you add this script package to your classic virtual system patterns:
      - `SF_PATH_TO_ENCRYPT`
      - `SF_PATH_TO_DECRYPT`
      - `SF_PATH_TO_BLACKLIST`
      - `SF_PATH_TO_REMOVE_BLACKLIST`
    - **Working directory**: shows the default working directory, *c:\temp\securityfirst\configure*.
    - **Logging directory**: shows the default directory where logs related to the script package execution are stored, *c:\temp\securityfirst\configure\logs*.
    - **Executable**: shows the command to run the main script in the script package, and the directory path where the script is located, *python c:\temp\securityfirst\configure\configure.py*.
    - **Executes**: shows the value, *at virtual system creation and when I initiate it*.
    - **Save environment variables after post-deployment executions**: shows the value, *Yes*.
    - **Operating system**: shows *Windows*.

    This completes the creation of the encryption configuration script package.

Results

The `securityfirst-config-windows-1.0.0.1` script package is added to the catalog, and is available to be added to appropriate parts in your classic virtual system patterns.

Adding encryption script packages to classic virtual system patterns

The installation and configuration script packages for your chosen operating system must be added to the classic virtual system pattern to encrypt data associated with the classic virtual system pattern when it is deployed.

Before you begin

You should have already created two encryption script packages, one for installation and one for configuration on your chosen operating system. In this procedure you add these two script packages to your classic virtual system patterns. Note that you must have already accepted the license agreement for each script package.

Procedure

1. Open the **Workload Console**.
2. Select **Patterns** > **Virtual Systems (Classic)**.
3. From the list of available classic virtual system patterns, select the pattern to which you will add the encryption script packages, or create a new classic virtual system pattern if needed.
4. Click **Edit** to open the Pattern Editor.
5. If you are working with a newly created virtual system pattern, select one or more virtual image parts from the **Parts** list and drag them onto the editing canvas.
6. Click the **Scripts** section of the Pattern Editor to expand the list of available script packages.
7. Select the encryption installation script package for your operating system from the list (for example, for Linux, select `securityfirst-install-linux-1.0.0.1`), and drag it onto one or more targeted virtual image parts, on the Pattern Editor canvas.
8. Select the encryption configuration script package for your operating system from the list (for example, for Linux, select `securityfirst-config–linux-1.0.0.1`), and drag it onto the same targeted virtual image parts, on the Pattern Editor canvas.

   Important: Keep in mind the following considerations as you add script packages to your image parts:
   - Be sure to select the installation and configuration script packages for the appropriate operating system.
   - Be sure to add the installation script package before you add the configuration script package to each part. When deployed, script packages are run in the order that they were added, and the installation script package must run before the configuration script package.
   - You must add both installation and configuration script packages to each virtual image part whose data is to be encrypted. Do not, for example, add the installation script package to one part and the configuration script package to a different part.
9. For each part to which you have added installation and configuration script packages, verify the field contents and edit the parameters of your script packages, as needed. You can also lock these parameters to prevent further changes at deployment time, or leave them unlocked and allow users to modify the settings at deployment time.

   For each encryption installation script package, you can configure the **SF_KEY_MANAGER** parameter, which is initially set to a value of *default*, indicating that the default key manager provided with PureApplication System is used. You can also lock this parameter setting by clicking the lock icon next to the field. This prevents unauthorized changes to this parameter setting at deployment time.

   To use a different key manager for encrypting classic virtual system pattern data, you can leave this page and click **System** > **Key Manager Adapters** to display the list of available key managers that are registered with PureApplication System, or you can register your own key manager. Ensure that

the key manager adapter that you choose is already installed. Return to the Pattern Editor and enter the name of the selected key manager in the **SF_KEY_MANAGER** field to configure that parameter for the installation script package.

For each encryption configuration script package, you can configure the following parameters:

SF_PATH_TO_ENCRYPT
For each directory path to be encrypted, type the absolute directory path in this field. If you enter more than one directory path, separate each directory path string with a comma.

SF_PATH_TO_DECRYPT
When you are configuring encryption files paths for the first time, there are no directory paths to decrypt, so you can leave this field blank. After deployment, when you want to remove encryption from one or more directory paths, you can specify the directory paths in this field when you run the configuration script again. If you enter more than one directory path, separate each directory path string with a comma.

SF_PATH_TO_BLACKLIST
Within the directory path to be encrypted, if you have subdirectories that you do not want to encrypt, type the absolute directory paths to exclude from encryption in this field. If you enter more than one directory path, separate each directory path string with a comma.

For example, suppose the following directory path is designated for encryption: `/tmp/DirA/myFiles`, and within the `/myFiles` directory there exist two subdirectories, `/myFiles/myDir1` and `/myFiles/myDir2`.

Exclude the files in `/myDir2` from being encrypted while allowing the files in `/myDir1` to be encrypted by specifying the directory paths as follows:

- **SF_PATH_TO_ENCRYPT**: `/tmp/DirA/myFiles`
- **SF_PATH_TO_BLACKLIST**: `/tmp/DirA/myFiles/myDir2`

SF_PATH_TO_REMOVE_BLACKLIST
When you are configuring encryption files paths for the first time, there are no directory paths defined in the blacklist, so you can leave this field blank. After deployment, when you want to remove one or more directory paths from the blacklist and thus enable it for encryption, type the absolute directory paths to remove from the blacklist in this field. If you enter more than one directory path, separate each directory path string with a comma.

As you define your directory paths for encryption, keep the following considerations in mind:

- Security First SPxBitFiler-IPA is intended to encrypt data, not executable programs.
- You must specify absolute directory paths for encryption using the syntax of the underlying operating system. For example:
  - For Linux or AIX: `/tmp/DirToEncrypt`
  - For Windows: `C:\temp\DirToEncrypt`

  For simplicity, most examples that follow use Linux and AIX (forward slash) formatting. You should apply appropriate syntax for Windows as needed.
- If entering more than one directory path, separate each absolute directory path with a comma (and do not insert a space before the comma, and do not end the directory path with a forward slash (`/`) character). For example:
  - For Linux or AIX: `/tmp/DirA/myFiles, /tmp/DirB/yourFiles`
  - For Windows: `C:\temp\DirA\myFiles, C:\temp\DirB\yourFiles`
- You can also lock these parameter settings by clicking the lock icon next to each field. This prevents unauthorized changes to this parameter setting at deployment time. However, you typically will leave these fields unlocked so that you can change encryption settings later after deployment, if needed.
- Be sure to complete this configuration for each pair of installation and configuration script packages you add to each virtual image part in your topology.

Important: Avoid encrypting directory paths that are used during system startup (boot and login processes). For example, the following directory paths should not be encrypted:

- `/root`
- `/home/virtuser`
- `/opt/IBM/WebSphere`

  Note, most subdirectories can be safely encrypted, such as `/opt/IBM/WebSphere/Profiles/Default/AppSrv01/installedApps`.

  For virtual machines created from deployed classic virtual system patterns, directory paths such as the following examples are safe to encrypt:

  – `/opt/ibm`
  – `/opt/ibm/ae`
- As another example for virtual system DB2 virtual machines, if you add the appropriate `/bin` directory path (such as `/opt/ibm/db2/V10.1/bin`) to the blacklist, you can encrypt files in `/opt/ibm` and `/opt/IBM` directory paths.

You should avoid encrypting `/bin` directories because they usually include executable files, not just data files.

Similarly, avoid encrypting files in temporary directories. For example, do not encrypt files in the following WebSphere temporary directory: `/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/wstemp`

10. Continue to add other parts, script packages, and add-ons to your classic virtual system pattern as needed.
11. When you have completed the definition and configuration of the script packages your classic virtual system pattern, click **Done editing**.
12. Verify that the **License status in this pattern** section indicates that all license agreements related to the virtual images and script packages for this pattern are marked as accepted. You must accept all license agreements before you can deploy your pattern.

Results

Your classic virtual system pattern is now configured for encryption and is ready for deployment.

Important: If you promote a classic virtual system pattern that uses the script packages, the system generates a virtual system pattern that uses script packages. Encryption for the promoted virtual system pattern will not work unless you modify the pattern to use the policy after it is promoted. For more information, see “Adding the encryption policy to virtual application patterns and virtual system patterns” on page 6.

Encrypting files in nonexistent directory paths

Directory paths that have not yet been created can still be included in your encryption configuration, and files in those directories are encrypted later after the directories are created.

Adding a nonexistent directory path to the encryption definition

You can add a directory path that has not yet been created to the list of directory paths to be encrypted. The Security First SPxBitFiler-IPA encryption software will watch indefinitely for the directory path to be created. When the directory path is created, there might be a delay of up to two minutes for the Security First SPxBitFiler-IPA software to detect the presence of the directory path and begin to encrypt the associated files.

For example, suppose that you add the directory path `/tmp/DirA` to the **Paths to encrypt** configuration setting (`SF_PATH_TO_ENCRYPT`), but `/DirA` does not yet exist. When this encryption configuration setting is run, Security First SPxBitFiler-IPA begins to monitor for the presence of this directory path.

When you later create this directory and add files to it, the encryption software takes approximately two minutes to detect the presence of this directory and begin the encryption process on the associated files.

Renaming an existing encrypted directory and creating a new directory with the original name

Suppose you have a directory path `/tmp/DirA`, and it is currently included in the encrypted directory paths in your deployed environment.

Now, suppose you rename directory `/tmp/DirA` to `/tmp/DirB`. Security First SPxBitFiler-IPA encryption software continues to keep the renamed directory `/tmp/DirB` encrypted, and begins to monitor for the presence of nonexistent directory `/tmp/DirA`.

At a later time, when you create a new directory `/tmp/DirA` and add files, it might take Security First SPxBitFiler-IPA approximately two minutes to detect the directory and begin to encrypt the associated files.

Deploying your classic virtual system pattern

After adding encryption script packages to your classic virtual system pattern, you can deploy the pattern into your cloud environment to encrypt associated data-at-rest.

Procedure

1. Click **Patterns** > **Virtual Systems (Classic)**.
2. Select the classic virtual system pattern to deploy.
3. Click **Deploy** on the toolbar. If your classic virtual system pattern contains script packages that contain license agreements that have not yet been accepted, the **Deploy** icon is not available. Ensure that all license agreements in script packages are accepted before attempting to deploy a pattern.
4. Provide any additional information necessary to deploy the classic virtual system pattern. The parameters that are required differ depending on the defined configuration and any associated script packages, but usually include the following information:
   - Virtual system name
   - Choose environment
   - Schedule deployment
   - Configure virtual parts

     For the encryption installation script packages, if the `SF_KEY_MANAGER` parameter is not already defined or is not locked, you can specify the name of the key manager adapter to use.

     For the encryption configuration script packages, if any of the following parameters are not already defined or are not locked, you can modify them as needed at this time:
     - `SF_PATH_TO_ENCRYPT`
     - `SF_PATH_TO_DECRYPT` (leave blank for initial deployment, since there should not be any encrypted paths to decrypt)
     - `SF_PATH_TO_BLACKLIST`
     - `SF_PATH_TO_REMOVE_FROM_BLACKLIST` (leave blank for initial deployment, since there should not be any directory paths to remove from the blacklist)

   See the PureApplication System Information Center for more information about deploying classic virtual system patterns.
5. Click **OK** to deploy the pattern.

What to do next

When the virtual application is deployed, the resulting virtual system instance is displayed in the **Instances** section of the workload console. To view the virtual instance, select **Instances** > **Virtual Systems (Classic)**.

Modifying the classic virtual system pattern encryption configuration after deployment

After deploying your classic virtual system pattern and initially encrypting associated data-at-rest, you can run the encryption configuration script package again to modify the encryption configuration as needed.

Before you begin

Use these procedures to update the encryption configuration settings for a deployed classic virtual system.

Attention: The Security First SPxBitFiler software relies on data in several installed directories that must be kept available:

- For AIX:

  The Security First SPxBitFiler software uses a driver file stored in the following directory:

  `/usr/lib/drivers/bfipa`

  The cryptographic keys by default are stored in the driver file.
- For Linux:

  The Security First SPxBitFiler software uses an XML configuration file stored in the following directory:

  `/etc/spxbitfiler-ipa/config/bitfiler-ipa.xml`

  The cryptographic keys for workgroups by default are stored in the following directory:

  `/etc/spxbitfiler-ipa`
- For Windows:

  The SPxBitFiler-IPA files are stored in the following directory:

  `C:\Program Files (x86)\Bitfiler-IPA`

If these directories are lost or deleted, ALL of the data stored in the share files will be inaccessible and considered useless. Take care to ensure that the data in these directories is available.

Procedure

1. Click **Instances** > **Virtual Systems (Classic)**.
2. Select a virtual system instance from the list of available instances.
3. Expand the **Virtual machines** node.
4. Select a virtual machine from the list by expanding the node associated with the virtual machine.
5. Scroll down through the details for the virtual machine to the **Script Packages** section. There you should find the encryption installation and encryption configuration script packages as well as other script packages deployed with the classic virtual system pattern.
6. For the encryption configuration script package, click **Execute now**.
7. Enter the authorized product administrator user name and password and click **OK**.
8. The encryption configuration parameters are displayed. For any of the following parameters, if they are not locked, modify or enter new directory paths to be encrypted or blacklisted. Type over or erase any previously specified directory paths as needed:
   - `SF_PATH_TO_ENCRYPT`
   - `SF_PATH_TO_DECRYPT`
   - `SF_PATH_TO_BLACKLIST`
   - `SF_PATH_TO_REMOVE_BLACKLIST`

   The script package is executed, and the files in the directories are encrypted or decrypted, added or removed from the blacklist as you specified. The results are displayed in the standard output and error log files.

See the PureApplication System Information Center for more information about running script packages for deployed classic virtual system patterns.

What to do next

You can run the encryption configuration script package on demand to continue to modify the encryption configuration as needed.

Updating the IBM Encryption Pattern for Security First SPxBitFiler-IPA

If a new version of the Encryption Pattern for Security First SPxBitFiler-IPA is available, you can update the pattern type.

For instructions on updating pattern types, see these IBM PureApplication System documentation topics:

- Updating virtual application pattern types by using the workload console
- Updating virtual application pattern types by using the command-line interface
- Updating deployed pattern types

Troubleshooting your encryption environment

Some common areas are discussed to help you identify and resolve problems related to your virtual pattern encryption environment.

Checking logs after running script packages

In general you should check your `remote_std_out.log` after each execution of your encryption script packages, in particular the configuration script package. The `remote_std_err.log` usually includes additional information to help you identify and resolve problems.

When you initially deploy your virtual patterns with encryption capability, be sure to examine the log files for the plug-in to see all of the information about the Security First SPxBitFiler-IPA installation and data encryption processes. You should look at `console.log` and `trace.log` files, located in the log directory for the deployed virtual machine, similar to the following example:

`/opt/IBM/maestro/agent/usr/servers/<part_id>/logs/<part_id>.ENCRYPTION-SECURITYFIRST`

In this directory path, *<part_id>* is a variable comprising the part name and an identifier. For example, `Core_OS.11371732048989`.

Running an encryption configuration script package results in errors

When you attempt to run an encryption configuration script package to change the encryption path settings, you might encounter errors similar to the following examples (for Linux and AIX systems):

```
Processing directory /root/.ssh for blacklisting
found a directory.../root/.ssh
Result from spxenc:
ERROR: Unable to get message queue handle err[2]

spxinfo -l : show the full list of directories that are protected
ERROR: Unable to get message queue handle err[2]
Failed to open list output file - err[2]

spxinfo -e : show the full list of directories that are on the blacklist
ERROR: Unable to get message queue handle err[2]
Failed to open list output file - err[2]

spxinfo -s : show the status of the transformation of existing data
ERROR: Unable to get message queue handle err[2]
Failed to open stats file - err[2]
```

This problem can occur if the Security First SPxBitFiler-IPA service is not running. You can verify this condition on Linux or AIX systems by running the following command:

```
ps -ef | grep spx
```

If no results are displayed, Security First SPxBitFiler-IPA is probably not running. You can attempt to start the service manually on Linux or AIX systems by running the following command:

```
/etc/init.d/spxbitfiler-ipa start
```

Then run the following command again and verify if the service starts running:

```
ps -ef | grep spx
```

If this action does not resolve the problem, you can try restarting the virtual machine. If this still does not resolve the problem you might need to redeploy the classic virtual system pattern.

Encrypting data associated with cloned virtual machines

There is a known limitation illustrated by the following scenario:

- Suppose that you create and deploy a classic virtual system pattern that contains encryption installation and configuration script packages.
- Virtual machines are created and data associated with these virtual machines are encrypted according to the initial encryption definitions specified in the encryption configuration script package.
- After deployment, you run the encryption configuration script package again, this time specifying different encryption configuration settings (adding or removing directory paths from encryption, etc).
- You then clone a virtual machine that has associated encrypted files.
- When you examine the encryption configuration of the cloned virtual machine, you find that only the most recent encryption configuration settings, that had been applied to files for the original virtual machine, are applied to the files associated with the cloned virtual machine.

This limitation occurs because a cloned virtual machine is actually a new deployment, so only the most recent configuration parameters are recognized as part of the deployment of this new cloned virtual machine.

You must run the encryption configuration script package again manually for the cloned virtual machine to define the encryption settings needed to match the original virtual machine encryption configuration.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other product and service names may be trademarks of IBM or other companies.

Printed in USA