Create the FPolicy configuration

ONTAP 9

NetApp
August 12, 2022

Table of Contents

Create the FPolicy configuration

Create the FPolicy external engine

You must create an external engine to start creating an FPolicy configuration. The external engine defines how FPolicy makes and manages connections to external FPolicy servers. If your configuration uses the internal ONTAP engine (the native external engine) for simple file blocking, you do not need to configure a separate FPolicy external engine and do not need to perform this step.

What you'll need

The external engine worksheet should be completed.

About this task

If the external engine is used in a MetroCluster configuration, you should specify the IP addresses of the FPolicy servers at the source site as primary servers. The IP addresses of the FPolicy servers at the destination site should be specified as secondary servers.

Steps

1. Create the FPolicy external engine by using the `vserver fpolicy policy external-engine create` command.

   The following command creates an external engine on storage virtual machine (SVM) vs1.example.com. No authentication is required for external communications with the FPolicy server.

   ```
   vserver fpolicy policy external-engine create -vserver-name vs1.example.com
   -engine-name engine1 -primary-servers 10.1.1.2,10.1.1.3 -port 6789 -ssl-option
   no-auth
   ```

2. Verify the FPolicy external engine configuration by using the `vserver fpolicy policy external-engine show` command.

   The following command display information about all external engines configured on SVM vs1.example.com:

   ```
   vserver fpolicy policy external-engine show -vserver vs1.example.com
   ```

   ```
                                Primary        Secondary
   External
   Vserver            Engine    Servers        Servers       Port Engine
   Type
   ---------------    ----------  -------------- ----------- ------
   ----------
   vs1.example.com    engine1   10.1.1.2,      -             6789
   synchronous
                                10.1.1.3
   ```

   The following command displays detailed information about the external engine named "engine1" on SVM

vs1.example.com:

```
vserver fpolicy policy external-engine show -vserver vs1.example.com -engine
-name engine1
```

```
                            Vserver: vs1.example.com
                             Engine: engine1
              Primary FPolicy Servers: 10.1.1.2, 10.1.1.3
       Port Number of FPolicy Service: 6789
            Secondary FPolicy Servers: -
                 External Engine Type: synchronous
SSL Option for External Communication: no-auth
           FQDN or Custom Common Name: -
         Serial Number of Certificate: -
               Certificate Authority: -
```

Create the FPolicy event

As part of creating an FPolicy policy configuration, you need to create an FPolicy event. You associate the event with the FPolicy policy when it is created. An event defines which protocol to monitor and which file access events to monitor and filter.

Before you begin

You should complete the FPolicy event worksheet.

Steps

1. Create the FPolicy event by using the `vserver fpolicy policy event create` command.

   ```
   vserver fpolicy policy event create -vserver-name vs1.example.com -event-name
   event1 -protocol cifs -file-operations open,close,read,write
   ```

2. Verify the FPolicy event configuration by using the `vserver fpolicy policy event show` command.

   ```
   vserver fpolicy policy event show -vserver vs1.example.com
   ```

   ```
                    Event               File                    Is Volume
   Vserver          Name      Protocols Operations    Filters   Operation
   ---------------  --------- --------- ------------- ---------
   ------------
   vs1.example.com  event1    cifs      open, close,  -         false
                                        read, write
   ```

Create the FPolicy policy

When you create the FPolicy policy, you associate an external engine and one or more events to the policy. The policy also specifies whether mandatory screening is required, whether the FPolicy servers have privileged access to data on the storage virtual machine (SVM), and whether passthrough-read for offline files is enabled.

What you'll need

- The FPolicy policy worksheet should be completed.
- If you plan on configuring the policy to use FPolicy servers, the external engine must exist.
- At least one FPolicy event that you plan on associating with the FPolicy policy must exist.
- If you want to configure privileged data access, a SMB server must exist on the SVM.

Steps

1. Create the FPolicy policy:

   ```
   vserver fpolicy policy create -vserver-name vserver_name -policy-name
   policy_name -engine engine_name -events event_name,… [-is-mandatory
   {true|false}] [-allow-privileged-access {yes|no}] [-privileged-user-name
   domain\user_name] [-is-passthrough-read-enabled {true|false}]
   ```

   - You can add one or more events to the FPolicy policy.
   - By default, mandatory screening is enabled.
   - If you want to allow privileged access by setting the `-allow-privileged-access` parameter to `yes`, you must also configure a privileged user name for privileged access.
   - If you want to configure passthrough-read by setting the `-is-passthrough-read-enabled` parameter to `true`, you must also configure privileged data access.

     The following command creates a policy named "policy1" that has the event named "event1" and the external engine named "engine1" associated with it. This policy uses default values in the policy configuration: `vserver fpolicy policy create -vserver vs1.example.com -policy -name policy1 -events event1 -engine engine1`

     The following command creates a policy named "policy2" that has the event named "event2" and the external engine named "engine2" associated with it. This policy is configured to use privileged access using the specified user name. Passthrough-read is enabled:

     ```
     vserver fpolicy policy create -vserver vs1.example.com -policy-name policy2
     -events event2 -engine engine2 -allow-privileged-access yes -privileged-
     user-name example\archive_acct -is-passthrough-read-enabled true
     ```

     The following command creates a policy named "native1" that has the event named "event3" associated with it. This policy uses the native engine and uses default values in the policy configuration:

     ```
     vserver fpolicy policy create -vserver vs1.example.com -policy-name native1
     -events event3 -engine native
     ```

2. Verify the FPolicy policy configuration by using the `vserver fpolicy policy show` command.

The following command displays information about the three configured FPolicy policies, including the following information:

- The SVM associated with the policy
- The external engine associated with the policy
- The events associated with the policy
- Whether mandatory screening is required
- Whether privileged access is required `vserver fpolicy policy show`

```
   Vserver       Policy     Events    Engine    Is Mandatory  Privileged
                 Name                                       Access
--------------   ---------  --------- --------- ------------
-----------
vs1.example.com  policy1    event1    engine1   true          no
vs1.example.com  policy2    event2    engine2   true          yes
vs1.example.com  native1    event3    native    true          no
```

Create the FPolicy scope

After creating the FPolicy policy, you need to create an FPolicy scope. When creating the scope, you associate the scope with an FPolicy policy. A scope defines the boundaries on which the FPolicy policy applies. Scopes can include or exclude files based on shares, export policies, volumes, and file extensions.

What you'll need

The FPolicy scope worksheet must be completed. The FPolicy policy must exist with an associated external engine (if the policy is configured to use external FPolicy servers) and must have at least one associated FPolicy event.

Steps

1. Create the FPolicy scope by using the `vserver fpolicy policy scope create` command.

   ```
   vserver fpolicy policy scope create -vserver-name vs1.example.com -policy-name
   policy1 -volumes-to-include datavol1,datavol2
   ```

2. Verify the FPolicy scope configuration by using the `vserver fpolicy policy scope show` command.

   ```
   vserver fpolicy policy scope show -vserver vs1.example.com -instance
   ```

```
                    Vserver: vs1.example.com
                     Policy: policy1
          Shares to Include: -
          Shares to Exclude: -
         Volumes to Include: datavol1, datavol2
         Volumes to Exclude: -
Export Policies to Include: -
Export Policies to Exclude: -
File Extensions to Include: -
File Extensions to Exclude: -
```

Enable the FPolicy policy

After you are through configuring an FPolicy policy configuration, you enable the FPolicy policy. Enabling the policy sets its priority and starts file access monitoring for the policy.

What you'll need

The FPolicy policy must exist with an associated external engine (if the policy is configured to use external FPolicy servers) and must have at least one associated FPolicy event. The FPolicy policy scope must exist and must be assigned to the FPolicy policy.

About this task

The priority is used when multiple policies are enabled on the storage virtual machine (SVM) and more than one policy has subscribed to the same file access event. Policies that use the native engine configuration have a higher priority than policies for any other engine, regardless of the sequence number assigned to them when enabling the policy.

A policy cannot be enabled on the admin SVM.

Steps

1. Enable the FPolicy policy by using the `vserver fpolicy enable` command.

   ```
   vserver fpolicy enable -vserver-name vs1.example.com -policy-name policy1
   -sequence-number 1
   ```

2. Verify that the FPolicy policy is enabled by using the `vserver fpolicy show` command.

   ```
   vserver fpolicy show -vserver vs1.example.com
   ```

   ```
                                    Sequence
   Vserver          Policy Name       Number  Status   Engine
   ---------------  ----------------- --------  -------- ---------
   vs1.example.com  policy1                1  on       engine1
   ```