

# Manage NTFS file security, NTFS audit policies, and Storage-Level Access Guard on SVMs using the CLI

**ONTAP 9** 

NetApp July 14, 2022

This PDF was generated from https://docs.netapp.com/us-en/ontap/smb-admin/manage-ntfs-security-audit-policies-slag-concept.html on July 14, 2022. Always check docs.netapp.com for the latest.

# **Table of Contents**

Ma	anage NTFS file security, NTFS audit policies, and Storage-Level Access Guard on SVMs using the CLI	1
	Manage NTFS file security, NTFS audit policies, and Storage-Level Access Guard on SVMs using the	
	CLI overview	1
	Use cases for using the CLI to set file and folder security	2
	Limits when using the CLI to set file and folder security	2
	How security descriptors are used to apply file and folder security	3
	Guidelines for applying file-directory policies that use local users or groups on the SVM disaster recovery	
	destination	4
	Configure and apply file security on NTFS files and folders using the CLI	6
	Configure and apply audit policies to NTFS files and folders using the CLI overview	. 14
	Considerations when managing security policy jobs	. 21
	Commands for managing NTFS security descriptors	. 22
	Commands for managing NTFS DACL access control entries	. 22
	Commands for managing NTFS SACL access control entries	. 23
	Commands for managing security policies	. 23
	Commands for managing security policy tasks	. 24
	Commands for managing security policy jobs	. 24

# Manage NTFS file security, NTFS audit policies, and Storage-Level Access Guard on SVMs using the CLI

# Manage NTFS file security, NTFS audit policies, and Storage-Level Access Guard on SVMs using the CLI overview

You can manage NTFS file security, NTFS audit policies, and Storage-Level Access Guard on storage virtual machines (SVMs) by using the CLI.

You can manage NTFS file security and audit policies from SMB clients or by using the CLI. However, using the CLI to configure file security and audit policies removes the need to use a remote client to manage file security. Using the CLI can significantly reduce the time it takes to apply security on many files and folders using a single command.

You can configure Storage-Level Access Guard, which is another layer of security applied by ONTAP to SVM volumes. Storage-Level Access Guard applies to accesses from all NAS protocols to the storage object to which Storage-Level Access Guard is applied.

Storage-Level Access Guard can be configured and managed only from the ONTAP CLI. You cannot manage Storage-Level Access Guard settings from SMB clients. Moreover, if you view the security settings on a file or directory from an NFS or SMB client, you will not see the Storage-Level Access Guard security. Storage-Level Access Guard security cannot be revoked from a client, even by a system (Windows or UNIX) administrator. Therefore, Storage-Level Access Guard provides an extra layer of security for data access that is independently set and managed by the storage administrator.



Even though only NTFS access permissions are supported for Storage-Level Access Guard, ONTAP can perform security checks for access over NFS to data on volumes where Storage-Level Access Guard is applied if the UNIX user maps to a Windows user on the SVM that owns the volume.

# NTFS security-style volumes

All files and folders contained within NTFS security-style volumes and qtrees have NTFS effective security. You can use the vserver security file-directory command family to implement the following types of security on NTFS security-style volumes:

- · File permissions and audit policies to files and folders contained in the volume
- · Storage-Level Access Guard security on volumes

# Mixed security-style volumes

Mixed security-style volumes and qtrees can contain some files and folders that have UNIX effective security and use UNIX file permissions, either mode bits or NFSv4.x ACLs and NFSv4.x audit policies, and some files and folders that have NTFS effective security and use NTFS file permissions and audit policies. You can use the vserver security file-directory command family to apply the following types of security to mixed security-style data:

- File permissions and audit policies to files and folders with NTFS effective security-style in the mixed volume or gtree
- Storage-Level Access Guard to volumes with either NTFS and UNIX effective security-style

# **UNIX security-style volumes**

UNIX security-style volumes and qtrees contain files and folders that have UNIX effective security (either mode bits or NFSv4.x ACLs). You must keep the following in mind if you want to use the vserver security file-directory command family to implement security on UNIX security-style volumes:

- The vserver security file-directory command family cannot be used to manage UNIX file security and audit policies on UNIX security-style volumes and qtrees.
- You can use the vserver security file-directory command family to configure Storage-Level
   Access Guard on UNIX security-style volumes, provided the SVM with the target volume contains a CIFS
   server.

#### **Related information**

Display information about file security and audit policies

Configure and apply file security on NTFS files and folders using the CLI

Configure and apply audit policies to NTFS files and folders using the CLI

Secure file access by using Storage-Level Access Guard

# Use cases for using the CLI to set file and folder security

Because you can apply and manage file and folder security locally without involvement from a remote client, you can significantly reduce the time it takes to set bulk security on a large number of files or folders.

You can benefit from using the CLI to set file and folder security in the following use cases:

- · Storage of files in large enterprise environments, such as file storage in home directories
- · Migration of data
- · Change of Windows domain
- Standardization of file security and audit policies across NTFS file systems

# Limits when using the CLI to set file and folder security

You need to be aware of certain limits when using the CLI to set file and folder security.

• The vserver security file-directory command family does not support setting NFSv4 ACLs.

You can only apply NTFS security descriptors to NTFS files and folders.

# How security descriptors are used to apply file and folder security

Security descriptors contain the access control lists that determine what actions a user can perform on files and folders, and what is audited when a user accesses files and folders.

#### Permissions

Permissions are allowed or denied by an object's owner and determine what actions an object (users, groups, or computer objects) can perform on specified files or folders.

# Security descriptors

Security descriptors are data structures that contain security information that define permissions associated with a file or folder.

# Access control lists (ACLs)

Access control lists are the lists contained within a security descriptor that contain information on what actions users, groups, or computer objects can perform on the file or folder to which the security descriptor is applied. The security descriptor can contain the following two types of ACLs:

- Discretionary access control lists (DACLs)
- System access control lists (SACLs)

# Discretionary access control lists (DACLs)

DACLs contain the list of SIDS for the users, groups, and computer objects who are allowed or denied access to perform actions on files or folders. DACLs contain zero or more access control entries (ACEs).

# System access control lists (SACLs)

SACLs contain the list of SIDS for the users, groups, and computer objects for which successful or failed auditing events are logged. SACLs contain zero or more access control entries (ACEs).

# Access Control Entries (ACEs)

ACEs are individual entries in either DACLs or SACLs:

- A DACL access control entry specifies the access rights that are allowed or denied for particular users, groups, or computer objects.
- A SACL access control entry specifies the success or failure events to log when auditing specified actions performed by particular users, groups, or computer objects.

## · Permission inheritance

Permission inheritance describes how permissions defined in security descriptors are propagated to an object from a parent object. Only inheritable permissions are inherited by child objects. When setting permissions on the parent object, you can decide whether folders, sub-folders, and files can inherit them with "Apply to this-folder, sub-folders, and files".

# **Related information**

SMB and NFS auditing and security tracing

Configuring and applying audit policies to NTFS files and folders using the CLI

# Guidelines for applying file-directory policies that use local users or groups on the SVM disaster recovery destination

There are certain guidelines that you must keep in mind before applying file-directory policies on the storage virtual machine (SVM) disaster recovery destination in an ID discard configuration if your file-directory policy configuration uses local users or groups in either the security descriptor or the DACL or SACL entries.

You can configure a disaster recovery configuration for an SVM where the source SVM on the source cluster replicates the data and configuration from the source SVM to a destination SVM on a destination cluster.

You can set up one of two types of SVM disaster recovery:

· Identity preserved

With this configuration, the identity of the SVM and the CIFS server is preserved.

· Identity discarded

With this configuration, the identity of the SVM and the CIFS server is not preserved. In this scenario, the name of the SVM and the CIFS server on the destination SVM is different from the SVM and the CIFS server name on the source SVM.

# **Guidelines for identity discarded configurations**

In an identity discarded configuration, for an SVM source that contains local user, group, and privilege configurations, the name of the local domain (local CIFS server name) must be changed to match the CIFS server name on the SVM destination. For example, if the source SVM name is "vs1" and CIFS server name is "CIFS1", and the destination SVM name is "vs1\_dst" and the CIFS server name is "CIFS1\_DST", then the local domain name for a local user named "CIFS1\user1" is automatically changed to "CIFS1\_DST\user1" on the destination SVM:

cluster1::> vserver cifs users-and-groups local-user show -vserver vs1 dst Vserver User Name Full Name Description CIFS1\Administrator Built-in administrator account vs1 CIFS1\user1 cluster1dst::> vserver cifs users-and-groups local-user show -vserver vs1 dst Vserver User Name Full Name Description \_\_\_\_\_\_ \_\_\_\_ vs1 dst CIFS1 DST\Administrator Built-in administrator account vs1\_dst CIFS1\_DST\user1

Even though local user and group names are automatically changed in the local user and group databases, local users or group names are not automatically changed in file-directory policy configurations (policies configured on the CLI using the vserver security file-directory command family).

For example, for "vs1", if you have configured a DACL entry where the -account parameter is set to "CIFS1\user1", the setting is not automatically changed on the destination SVM to reflect the destination's CIFS server name.

```
cluster1::> vserver security file-directory ntfs dacl show -vserver vs1
Vserver: vs1
 NTFS Security Descriptor Name: sd1
   Account Name Access Access
                                Apply To
                Type Rights
                _____
   CIFS1\user1 allow full-control this-folder
cluster1::> vserver security file-directory ntfs dacl show -vserver
vs1 dst
Vserver: vs1 dst
 NTFS Security Descriptor Name: sdl
   Account Name
               Access Access
                                      Apply To
                Type Rights
   -----
   **CIFS1**\user1 allow full-control this-folder
```

You must use the vserver security file-directory modify commands to manually change the CIFS server name to the destination CIFS server name.

# File-directory policy configuration components that contain account parameters

There are three file-directory policy configuration components that can use parameter settings that can contain local users or groups:

# Security descriptor

You can optionally specify the owner of the security descriptor and the primary group of the owner of the security descriptor. If the security descriptor uses a local user or group for the owner and primary group entries, you must modify the security descriptor to use the destination SVM in the account name. You can use the vserver security file-directory ntfs modify command to make any necessary changes to the account names.

#### DACL entries

Each DACL entry must be associated with an account. You must modify any DACLs that use local user or group accounts to use the destination SVM name. Because you cannot modify the account name for existing DACL entries, you must remove any DACL entries with local users or groups from the security descriptors, create new DACL entries with the corrected destination account names, and associate these new DACL entries with the appropriate security descriptors.

# SACL entries

Each SACL entry must be associated with an account. You must modify any SACLs that use local user or group accounts to use the destination SVM name. Because you cannot modify the account name for existing SACL entries, you must remove any SACL entries with local users or groups from the security descriptors, create new SACL entries with the corrected destination account names, and associate these new SACL entries with the appropriate security descriptors.

You must make any necessary changes to local users or groups used in the file-directory policy configuration before applying the policy; otherwise, the apply job fails.

# Configure and apply file security on NTFS files and folders using the CLI

# Create an NTFS security descriptor

Creating an NTFS security descriptor (file security policy) is the first step in configuring and applying NTFS access control lists (ACLs) to files and folders residing within storage virtual machines (SVMs). You can associate the security descriptor to the file or folder path in a policy task.

#### About this task

You can create NTFS security descriptors for files and folders residing within NTFS security-style volumes, or for files and folders residing on mixed security-style volumes.

By default, when a security descriptor is created, four discretionary access control list (DACL) access control entries (ACEs) are added to that security descriptor. The four default ACEs are as follows:

Object	Access type	Access rights	Where to apply the permissions
BUILTIN\Administrators	Allow	Full Control	this-folder, sub-folders, files
BUILTIN\Users	Allow	Full Control	this-folder, sub-folders, files
CREATOR OWNER	Allow	Full Control	this-folder, sub-folders, files
NT AUTHORITY\SYSTEM	Allow	Full Control	this-folder, sub-folders, files

You can customize the security descriptor configuration by using the following optional parameters:

- · Owner of the security descriptor
- · Primary group of the owner
- Raw control flags

The value for any optional parameter is ignored for Storage-Level Access Guard. See the man pages for more information.

# Add NTFS DACL access control entries to the NTFS security descriptor

Adding DACL (discretionary access control list) access control entries (ACEs) to the NTFS security descriptor is the second step in configuring and applying NTFS ACLs to a file or folder. Each entry identifies which object is allowed or denied access, and defines what the object can or cannot do to the files or folders defined in the ACE.

#### About this task

You can add one or more ACEs to the security descriptor's DACL.

If the security descriptor contains a DACL that has existing ACEs, the command adds the new ACE to the DACL. If the security descriptor does not contain a DACL, the command creates the DACL and adds the new ACE to it.

You can optionally customize DACL entries by specifying what rights you want to allow or deny for the account specified in the -account parameter. There are three mutually exclusive methods for specifying rights:

- Rights
- · Advanced rights
- Raw rights (advanced-privilege)



If you do not specify rights for the DACL entry, the default is to set the rights to Full Control.

You can optionally customize DACL entries by specifying how to apply inheritance.

The value for any optional parameter is ignored for Storage-Level Access Guard. See the man pages for more information.

# **Steps**

 Add a DACL entry to a security descriptor: vserver security file-directory ntfs dacl add -vserver vserver\_name -ntfs-sd SD\_name -access-type {allow|deny} -account name\_or\_SIDoptional\_parameters

```
vserver security file-directory ntfs dacl add -ntfs-sd sd1 -access-type deny
-account domain\joe -rights full-control -apply-to this-folder -vserver vs1
```

2. Verify that the DACL entry is correct: vserver security file-directory ntfs dacl show -vserver vserver\_name -ntfs-sd SD\_name -access-type {allow|deny} -account name or SID

vserver security file-directory ntfs dacl show -vserver vs1 -ntfs-sd sd1
-access-type deny -account domain\joe

```
Vserver: vs1

Security Descriptor Name: sd1

Allow or Deny: deny

Account Name or SID: DOMAIN\joe

Access Rights: full-control

Advanced Access Rights: -

Apply To: this-folder

Access Rights: full-control
```

# Create security policies

Creating a file security policy for SVMs is the third step in configuring and applying ACLs to a file or folder. A policy acts as a container for various tasks, where each task is a single entry that can be applied to files or folders. You can add tasks to the security policy later.

## About this task

The tasks that you add to a security policy contain associations between the NTFS security descriptor and the file or folder paths. Therefore, you should associate the security policy with each SVM (containing NTFS security-style volumes or mixed security-style volumes).

#### **Steps**

1. Create a security policy: vserver security file-directory policy create -vserver vserver name -policy-name policy name

```
vserver security file-directory policy create -policy-name policy1 -vserver vs1
```

2. Verify the security policy: vserver security file-directory policy show



# Add a task to the security policy

Creating and adding a policy task to a security policy is the fourth step in configuring and applying ACLs to files or folders in SVMs. When you create the policy task, you associate the task with a security policy. You can add one or more task entries to a security policy.

#### About this task

The security policy is a container for a task. A task refers to a single operation that can be done by a security policy to files or folders with NTFS or mixed security (or to a volume object if configuring Storage-Level Access Guard).

There are two types of tasks:

· File and directory tasks

Used to specify tasks that apply security descriptors to specified files and folders. ACLs applied through file and directory tasks can be managed with SMB clients or the ONTAP CLI.

Storage-Level Access Guard tasks

Used to specify tasks that apply Storage-Level Access Guard security descriptors to a specified volume. ACLs applied through Storage-Level Access Guard tasks can be managed only through the ONTAP CLI.

A task contains definitions for the security configuration of a file (or folder) or set of files (or folders). Every task in a policy is uniquely identified by the path. There can be only one task per path within a single policy. A policy cannot have duplicate task entries.

Guidelines for adding a task to a policy:

- There can be a maximum of 10,000 tasks entries per policy.
- · A policy can contain one or more tasks.

Even though a policy can contain more than one task, you cannot configure a policy to contain both filedirectory and Storage-Level Access Guard tasks. A policy must contain either all Storage-Level Access Guard tasks or all file-directory tasks.

• Storage-Level Access Guard is used to restrict permissions.

It will never give extra access permissions.

When adding tasks to security policies, you must specify the following four required parameters:

- SVM name
- · Policy name

- Path
- · Security descriptor to associate with the path

You can customize the security descriptor configuration by using the following optional parameters:

- · Security type
- · Propagation mode
- · Index position
- · Access control type

The value for any optional parameter is ignored for Storage-Level Access Guard. See the man pages for more information.

# **Steps**

1. Add a task with an associated security descriptor to the security policy: vserver security file-directory policy task add -vserver vserver\_name -policy-name policy\_name -path path -ntfs-sd SD nameoptional parameters

file-directory is the default value for the -access-control parameter. Specifying the access control type when configuring file and directory access tasks is optional.

```
vserver security file-directory policy task add -vserver vs1 -policy-name policy1 -path /home/dir1 -security-type ntfs -ntfs-mode propagate -ntfs-sd sd2 -index-num 1 -access-control file-directory
```

2. Verify the policy task configuration: vserver security file-directory policy task show -vserver vserver\_name -policy-name policy\_name -path path

vserver security file-directory policy task show

# Apply security policies

Applying a file security policy to SVMs is the last step in creating and applying NTFS ACLs to files or folders.

#### About this task

You can apply security settings defined in the security policy to NTFS files and folders residing within FlexVol volumes (NTFS or mixed security style).

# Step

1. Apply a security policy: vserver security file-directory apply -vserver vserver\_name -policy-name policy name

```
vserver security file-directory apply -vserver vs1 -policy-name policy1
```

The policy apply job is scheduled and the Job ID is returned.

```
[Job 53322] Job is queued: Fsecurity Apply. Use the "Job show 53322 -id 53322" command to view the status of the operation
```

# Monitor the security policy job

When applying the security policy to storage virtual machines (SVMs), you can monitor the progress of the task by monitoring the security policy job. This is helpful if you want to ascertain that the application of the security policy succeeded. This is also helpful if you have a long-running job where you are applying bulk security to a large number of files and folders.

#### About this task

To display detailed information about a security policy job, you should use the -instance parameter.

#### Step

1. Monitor the security policy job: vserver security file-directory job show -vserver vserver name

vserver security file-directory job show -vserver vs1

```
Job ID Name Vserver Node State

53322 Fsecurity Apply vs1 node1 Success
Description: File Directory Security Apply Job
```

# Verify the applied file security

You can verify the file security settings to confirm that the files or folders on the storage virtual machine (SVM) to which you applied the security policy have the desired settings.

## About this task

You must supply the name of the SVM that contains the data and the path to the file and folders on which you want to verify security settings. You can use the optional <code>-expand-mask</code> parameter to display detailed information about the security settings.

## Step

1. Display file and folder security settings: vserver security file-directory show -vserver vserver name -path path [-expand-mask true]

vserver security file-directory show -vserver vs1 -path /data/engineering -expand-mask true

```
Vserver: vs1
           File Path: /data/engineering
    File Inode Number: 5544
       Security Style: ntfs
      Effective Style: ntfs
       DOS Attributes: 10
DOS Attributes in Text: ----D---
Expanded Dos Attributes: 0x10
    ...0 .... = Offline
    .... = Sparse
    \dots 0\dots = Normal
    .... = Archive
    .... = Directory
    .... .... .0.. = System
    .... .... .... ..0. = Hidden
    \dots 0 = Read Only
        Unix User Id: 0
        Unix Group Id: 0
       Unix Mode Bits: 777
Unix Mode Bits in Text: rwxrwxrwx
               ACLs: NTFS Security Descriptor
                   Control:0x8004
                       1... - Self Relative
                       .0.. .... = RM Control Valid
                       ..0. .... = SACL Protected
                       ...0 .... = DACL Protected
                       .... 0... ... = SACL Inherited
                       .... .0.. .... = DACL Inherited
                       .... ..0. .... = SACL Inherit Required
                       .... = DACL Inherit Required
                       .... = SACL Defaulted
                       .... = SACL Present
                       .... 0... = DACL Defaulted
                       .... .... .1.. = DACL Present
                       \dots 0 = Owner Defaulted
                    Owner: BUILTIN\Administrators
```

	Group:BUILTIN\Administrators
	DACL - ACEs
	ALLOW-Everyone-0x1f01ff
	0 =
Generic Read	
	.0 =
Generic Write	
	0 =
Generic Execute	
Generic Execute	0 =
0 111	0
Generic All	
	=
System Security	
	=
Synchronize	
	=
Write Owner	
Write DAC	
WIICC DAC	
Read Control	
	=
Delete	
	1 =
Write Attributes	
	1 =
Read Attributes	
	=
Delete Child	
Delete onlin	=
D	
Execute	
	1 =
Write EA	
	1 =
Read EA	
	1 =
Append	
Write	
Dood	
Read	
	ALLOW-Everyone-0x10000000-01 C1 IO
	0 =
Generic Read	
	.0 =

Generic Write	
Generic Execute	0 =
Generic All	1 =
	=
System Security	=
Synchronize	
Write Owner	=
Write DAC	=
David Cantural	=
Read Control	=
Delete	=
Write Attributes	
Read Attributes	0 =
Delete Child	=
The cut o	=
Execute	=
Write EA	0 =
Read EA	0
Append	0 =
Write	
	=
Read	

# Configure and apply audit policies to NTFS files and folders using the CLI overview

There are several steps you must perform to apply audit policies to NTFS files and folders when using the ONTAP CLI. First, you create an NTFS security descriptor and add SACLs to the security descriptor. Next you create a security policy and add policy tasks. You then apply the security policy to a storage virtual machine (SVM).

# About this task

After applying the security policy, you can monitor the security policy job and then verify the settings for the applied audit policy.



When an audit policy and associated SACLs are applied, any existing DACLs are overwritten. You should review existing security policies before creating and applying new ones.

#### Related information

Securing file access by using Storage-Level Access Guard

Limits when using the CLI to set file and folder security

How security descriptors are used to apply file and folder security

SMB and NFS auditing and security tracing

Configure and apply file security on NTFS files and folders using the CLI

# Create an NTFS security descriptor

Creating an NTFS security descriptor audit policy is the first step in configuring and applying NTFS access control lists (ACLs) to files and folders residing within SVMs. You will associate the security descriptor to the file or folder path in a policy task.

#### About this task

You can create NTFS security descriptors for files and folders residing within NTFS security-style volumes, or for files and folders residing on mixed security-style volumes.

By default, when a security descriptor is created, four discretionary access control list (DACL) access control entries (ACEs) are added to that security descriptor. The four default ACEs are as follows:

Object	Access type	Access rights	Where to apply the permissions
BUILTIN\Administrators	Allow	Full Control	this-folder, sub-folders, files
BUILTIN\Users	Allow	Full Control	this-folder, sub-folders, files
CREATOR OWNER	Allow	Full Control	this-folder, sub-folders, files
NT AUTHORITY\SYSTEM	Allow	Full Control	this-folder, sub-folders, files

You can customize the security descriptor configuration by using the following optional parameters:

- Owner of the security descriptor
- · Primary group of the owner
- Raw control flags

The value for any optional parameter is ignored for Storage-Level Access Guard. See the man pages for more information.

# Steps

- 1. If you want to use the advanced parameters, set the privilege level to advanced: set -privilege advanced
- 2. Create a security descriptor: vserver security file-directory ntfs create -vserver vserver name -ntfs-sd SD nameoptional parameters

 $\hbox{\tt vserver security file-directory ntfs create -\tt ntfs-sd sdl -\tt vserver vsl -\tt owner DOMAIN \verb|\joe| \\$ 

3. Verify that the security descriptor configuration is correct: vserver security file-directory ntfs show -vserver vserver\_name -ntfs-sd SD\_name

vserver security file-directory ntfs show -vserver vsl -ntfs-sd sdl

```
Vserver: vs1
Security Descriptor Name: sd1
Owner of the Security Descriptor: DOMAIN\joe
```

4. If you are in the advanced privilege level, return to the admin privilege level: set -privilege admin

# Add NTFS SACL access control entries to the NTFS security descriptor

Adding SACL (system access control list) access control entries (ACEs) to the NTFS security descriptor is the second step in creating NTFS audit policies for files or folders in SVMs. Each entry identifies the user or group that you want to audit. The SACL entry defines whether you want to audit successful or failed access attempts.

## **About this task**

You can add one or more ACEs to the security descriptor's SACL.

If the security descriptor contains a SACL that has existing ACEs, the command adds the new ACE to the SACL. If the security descriptor does not contain a SACL, the command creates the SACL and adds the new ACE to it.

You can configure SACL entries by specifying what rights you want to audit for success or failure events for the account specified in the -account parameter. There are three mutually exclusive methods for specifying rights:

- Rights
- · Advanced rights
- Raw rights (advanced-privilege)



If you do not specify rights for the SACL entry, the default setting is Full Control.

You can optionally customize SACL entries by specifying how to apply inheritance with the apply to parameter. If you do not specify this parameter, the default is to apply this SACL entry to this folder, subfolders, and files.

# Steps

1. Add a SACL entry to a security descriptor: vserver security file-directory ntfs sacl add -vserver vserver\_name -ntfs-sd SD\_name -access-type {failure|success} -account name or SIDoptional parameters

```
vserver security file-directory ntfs sacl add -ntfs-sd sdl -access-type failure -account domain\joe -rights full-control -apply-to this-folder -vserver vsl
```

2. Verify that the SACL entry is correct: vserver security file-directory ntfs sacl show -vserver vserver\_name -ntfs-sd SD\_name -access-type {failure|success} -account name\_or\_SID

vserver security file-directory ntfs sacl show -vserver vs1 -ntfs-sd sd1
-access-type deny -account domain\joe

```
Vserver: vs1

Security Descriptor Name: sd1

Access type for Specified Access Rights: failure

Account Name or SID: DOMAIN\joe

Access Rights: full-control

Advanced Access Rights: -

Apply To: this-folder

Access Rights: full-control
```

# **Create security policies**

Creating an audit policy for storage virtual machines (SVMs) is the third step in configuring and applying ACLs to a file or folder. A policy acts as a container for various tasks, where each task is a single entry that can be applied to files or folders. You can add tasks to the security policy later.

#### About this task

The tasks that you add to a security policy contain associations between the NTFS security descriptor and the file or folder paths. Therefore, you should associate the security policy with each storage virtual machine (SVM) (containing NTFS security-style volumes or mixed security-style volumes).

#### Steps

 Create a security policy: vserver security file-directory policy create -vserver vserver\_name -policy-name policy\_name

vserver security file-directory policy create -policy-name policy1 -vserver vs1

2. Verify the security policy: vserver security file-directory policy show

```
vserver security file-directory policy show

Vserver Policy Name

-----
vs1 policy1
```

# Add a task to the security policy

Creating and adding a policy task to a security policy is the fourth step in configuring and applying ACLs to files or folders in SVMs. When you create the policy task, you associate the task with a security policy. You can add one or more task entries to a security policy.

#### About this task

The security policy is a container for a task. A task refers to a single operation that can be done by a security policy to files or folders with NTFS or mixed security (or to a volume object if configuring Storage-Level Access Guard).

There are two types of tasks:

• File and directory tasks

Used to specify tasks that apply security descriptors to specified files and folders. ACLs applied through file and directory tasks can be managed with SMB clients or the ONTAP CLI.

Storage-Level Access Guard tasks

Used to specify tasks that apply Storage-Level Access Guard security descriptors to a specified volume. ACLs applied through Storage-Level Access Guard tasks can be managed only through the ONTAP CLI.

A task contains definitions for the security configuration of a file (or folder) or set of files (or folders). Every task in a policy is uniquely identified by the path. There can be only one task per path within a single policy. A policy cannot have duplicate task entries.

Guidelines for adding a task to a policy:

- There can be a maximum of 10,000 tasks entries per policy.
- · A policy can contain one or more tasks.

Even though a policy can contain more than one task, you cannot configure a policy to contain both filedirectory and Storage-Level Access Guard tasks. A policy must contain either all Storage-Level Access Guard tasks or all file-directory tasks.

• Storage-Level Access Guard is used to restrict permissions.

It will never give extra access permissions.

You can customize the security descriptor configuration by using the following optional parameters:

- · Security type
- · Propagation mode
- · Index position
- · Access control type

The value for any optional parameter is ignored for Storage-Level Access Guard. See the man pages for more information.

# **Steps**

1. Add a task with an associated security descriptor to the security policy: vserver security file-directory policy task add -vserver vserver\_name -policy-name policy\_name -path path -ntfs-sd SD nameoptional parameters

file-directory is the default value for the -access-control parameter. Specifying the access control type when configuring file and directory access tasks is optional.

```
vserver security file-directory policy task add -vserver vs1 -policy-name
policy1 -path /home/dir1 -security-type ntfs -ntfs-mode propagate -ntfs-sd sd2
-index-num 1 -access-control file-directory
```

2. Verify the policy task configuration: vserver security file-directory policy task show -vserver vserver name -policy-name policy name -path path

vserver security file-directory policy task show

```
Vserver: vs1
Policy: policy1
Index
       File/Folder
                     Access
                                   Security
                                             NTFS
                                                       NTFS
Security
       Path
                     Control
                                   Type
                                             Mode
Descriptor Name
       _____
_____
       /home/dir1
                   file-directory ntfs
                                             propagate sd2
```

# Apply security policies

Applying an audit policy to SVMsis the last step in creating and applying NTFS ACLs to files or folders.

#### About this task

You can apply security settings defined in the security policy to NTFS files and folders residing within FlexVol volumes (NTFS or mixed security style).

## Step

1. Apply a security policy: vserver security file-directory apply -vserver vserver\_name

```
-policy-name policy_name
vserver security file-directory apply -vserver vs1 -policy-name policy1
```

The policy apply job is scheduled and the Job ID is returned.

```
[Job 53322] Job is queued: Fsecurity Apply. Use the "Job show 53322 -id 53322" command to view the status of the operation
```

# Monitor the security policy job

When applying the security policy to storage virtual machines (SVMs), you can monitor the progress of the task by monitoring the security policy job. This is helpful if you want to ascertain that the application of the security policy succeeded. This is also helpful if you have a long-running job where you are applying bulk security to a large number of files and folders.

#### About this task

To display detailed information about a security policy job, you should use the -instance parameter.

## Step

1. Monitor the security policy job: vserver security file-directory job show -vserver vserver name

vserver security file-directory job show -vserver vs1

```
Job ID Name Vserver Node State

53322 Fsecurity Apply vs1 node1 Success
Description: File Directory Security Apply Job
```

# Verify the applied audit policy

You can verify the audit policy to confirm that the files or folders on the storage virtual machine (SVM) to which you applied the security policy have the desired audit security settings.

#### About this task

You use the vserver security file-directory show command to display audit policy information. You must supply the name of the SVM that contains the data and the path to the data whose file or folder audit policy information you want to display.

#### Step

1. Display audit policy settings: vserver security file-directory show -vserver vserver name -path path

## **Example**

The following command displays the audit policy information applied to the path "/corp" in SVM vs1. The path has both a SUCCESS and a SUCCESS/FAIL SACL entry applied to it:

```
cluster::> vserver security file-directory show -vserver vs1 -path /corp
                Vserver: vs1
              File Path: /corp
         Security Style: ntfs
        Effective Style: ntfs
         DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
           Unix User Id: 0
          Unix Group Id: 0
         Unix Mode Bits: 777
 Unix Mode Bits in Text: rwxrwxrwx
                   ACLs: NTFS Security Descriptor
                         Control:0x8014
                         Owner: DOMAIN\Administrator
                         Group:BUILTIN\Administrators
                         SACL - ACEs
                           ALL-DOMAIN\Administrator-0x100081-OI|CI|SA|FA
                           SUCCESSFUL-DOMAIN\user1-0x100116-OI|CI|SA
                         DACL - ACEs
                           ALLOW-BUILTIN\Administrators-0x1f01ff-OI|CI
                           ALLOW-BUILTIN\Users-0x1f01ff-OI|CI
                           ALLOW-CREATOR OWNER-0x1f01ff-OI|CI
                           ALLOW-NT AUTHORITY\SYSTEM-0x1f01ff-OI|CI
```

# Considerations when managing security policy jobs

If a security policy job exists, under certain circumstances, you cannot modify that security policy or the tasks assigned to that policy. You should understand under what conditions you can or cannot modify security policies so that any attempts that you make to modify the policy are successful. Modifications to the policy include adding, removing, or modifying tasks assigned to the policy and deleting or modifying the policy.

You cannot modify a security policy or a task assigned to that policy if a job exists for that policy and that job is in the following states:

- The job is running or in progress.
- The job is paused.
- The job is resumed and is in the running state.
- If the job is waiting to failover to another node.

Under the following circumstances, if a job exists for a security policy, you can successfully modify that security policy or a task assigned to that policy:

- The policy job is stopped.
- The policy job has successfully finished.

# **Commands for managing NTFS security descriptors**

There are specific ONTAP commands for managing security descriptors. You can create, modify, delete, and display information about security descriptors.

If you want to	Use this command
Create NTFS security descriptors	vserver security file-directory ntfs create
Modify existing NTFS security descriptors	vserver security file-directory ntfs modify
Display information about existing NTFS security descriptors	vserver security file-directory ntfs show
Delete NTFS security descriptors	vserver security file-directory ntfs delete

See the man pages for the vserver security file-directory ntfs commands for more information.

# Commands for managing NTFS DACL access control entries

There are specific ONTAP commands for managing DACL access control entries (ACEs). You can add ACEs to NTFS DACLs at any time. You can also manage existing NTFS DACLs by modifying, deleting, and displaying information about ACEs in DACLs.

If you want to	Use this command
Create ACEs and add them to NTFS DACLs	vserver security file-directory ntfs dacl add
Modify existing ACEs in NTFS DACLs	vserver security file-directory ntfs dacl modify
Display information about existing ACEs in NTFS DACLs	vserver security file-directory ntfs dacl show

If you want to	Use this command
Remove existing ACEs from NTFS DACLs	vserver security file-directory ntfs dacl remove

See the man pages for the vserver security file-directory ntfs dacl commands for more information.

# Commands for managing NTFS SACL access control entries

There are specific ONTAP commands for managing SACL access control entries (ACEs). You can add ACEs to NTFS SACLs at any time. You can also manage existing NTFS SACLs by modifying, deleting, and displaying information about ACEs in SACLs.

If you want to	Use this command
Create ACEs and add them to NTFS SACLs	vserver security file-directory ntfs sacl add
Modify existing ACEs in NTFS SACLs	vserver security file-directory ntfs sacl modify
Display information about existing ACEs in NTFS SACLs	vserver security file-directory ntfs sacl show
Remove existing ACEs from NTFS SACLs	vserver security file-directory ntfs sacl remove

See the man pages for the vserver security file-directory ntfs sacl commands for more information.

# **Commands for managing security policies**

There are specific ONTAP commands for managing security policies. You can display information about policies and you can delete policies. You cannot modify a security policy.

If you want to	Use this command
Create security policies	vserver security file-directory policy create
Display information about security policies	vserver security file-directory policy show

If you want to	Use this command
Delete security policies	vserver security file-directory policy delete

See the man pages for the vserver security file-directory policy commands for more information.

# Commands for managing security policy tasks

There are ONTAP commands for adding, modifying, removing, and displaying information about security policy tasks.

If you want to	Use this command
Add security policy tasks	vserver security file-directory policy task add
Modify security policy tasks	vserver security file-directory policy task modify
Display information about security policy tasks	vserver security file-directory policy task show
Remove security policy tasks	vserver security file-directory policy task remove

See the man pages for the vserver security file-directory policy task commands for more information.

# Commands for managing security policy jobs

There are ONTAP commands for pausing, resuming, stopping, and displaying information about security policy jobs.

If you want to	Use this command
Pause security policy jobs	vserver security file-directory job pause -vserver vserver_name -id integer
Resume security policy jobs	<pre>vserver security file-directory job resume -vserver vserver_name -id integer</pre>
Display information about security policy jobs	vserver security file-directory job show -vserver vserver_name You can determine the job ID of a job using this command.

If you want to	Use this command
Stop security policy jobs	<pre>vserver security file-directory job stop -vserver vserver_name -id integer</pre>

See the man pages for the  ${\tt vserver}$   ${\tt security}$  file-directory job commands for more information.

# **Copyright Information**

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system- without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## **Trademark Information**

NETAPP, the NETAPP logo, and the marks listed at <a href="http://www.netapp.com/TM">http://www.netapp.com/TM</a> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.