Technical Report

File Access Auditing on NetApp Controller

Sharyathi Nagesh, Reena Gupta, NetApp
Version 2.0
July 2011 | TR-3595

CIFS AND NFS AUDITING IN DATA ONTAP

This guide explores various features available in Data ONTAP® to monitor file access on NFS exports and CIFS shares. This document explains how you can configure the NetApp® storage box to CIFS and NFS auditing using either FPolicy® or native auditing frameworks. This guide is intended to serve as a quick reference only. All of the commands and options referred to in this document are based on Data ONTAP version 7.2.2 and higher. NetApp recommends using this version for the NFS auditing feature.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Using Data ONTAP to enable auditing solutions at the storage level provides multiple benefits, such as the ability to:

- Monitor file access in NFS exports and CIFS shares.
- Manage persistent audit logs across system reboots.
- Provide real-time notifications, on wire, to give instant alerts to storage administrators.

The NetApp native auditing and FPolicy framework, along with the NetApp partner solution, provides end-to-end solutions to NetApp customers. Solutions can be related to regulatory compliancy, data protection, access monitoring, and storage analysis.

2 NATIVE AUDITING

NetApp native auditing, the auditing framework provided by Data ONTAP, is similar to the auditing performed on Windows® servers. This feature has been present from the early versions of Data ONTAP, but NetApp recommends using Data ONTAP version 7.2.2 or higher. After the native audit options are configured in the NetApp controller, system access control lists (SACLs) must be set on folders and files in order to show auditing in action. Native auditing can be enabled on either CIFS shares or NFS exports, as explained in the following sections.

2.1 CIFS AUDITING

CIFS auditing refers to auditing access events from Windows clients, which access data on the storage system using the CIFS protocol.

PREREQUISITES FOR CIFS AUDITING

- CIFS must be licensed and enabled on the storage system before auditing can be enabled.
- The file or directory to be audited must be in a mixed or New Technology File System (NTFS) volume or qtree. You cannot audit CIFS events for a file or directory in a UNIX® volume or qtree unless the Storage-Level Access Guard is enabled. For more information, refer to NetApp TR-3596: Storage-Level Access Guard Quick Start Guide.
- Event auditing is turned off by default. To audit specific events, you must enable auditing and individual options for the events.

CONFIGURATION

Configure the following options in Data ONTAP for CIFS auditing:

- Enable CIFS audit

  `cifs audit start | stop`

  `options cifs.audit.enable on | off` (alternate for CIFS audit start/stop)
- Audit file access events

  `options cifs.audit.file_access_events.enable on | off`
- Audit log-on/log-off events

  `options cifs.audit.logon_events.enable on | off`
- Audit account management events

  `options cifs.audit.account_mgmt_events.enable on | off`

Once you have enabled auditing options on the controller, you must specify more granular operations to be audited using SACLs, as described in section 2.3 under “Specifying Auditable File Access Events (SACLs).”

2.2 NFS AUDITING

NFS auditing refers to auditing access events from UNIX/Linux® clients that access data on the storage system using the NFS protocol. Since NFS auditing is also based on NTFS ACLs, effective mapping of UNIX users to Windows users is required. Underneath, both CIFS and NFS auditing use the native auditing framework.

PREREQUISITES FOR NFS AUDITING

- CIFS must be licensed and enabled on the storage system to enable NFS auditing.
- NFS event auditing is turned off by default. To enable NFS auditing, enable NFS auditing options after enabling CIFS auditing.

CONFIGURATION

Configure the following options in Data ONTAP for NFS auditing:

- Enable NFS audit
  `options cifs.audit.nfs.enable on`
- Audit file access events
  `options cifs.audit.file_access_events.enable on`
- Audit log-on/log-off events
  `options cifs.audit.logon_events.enable on | off`

Once you have enabled auditing options on the controller, you must specify more granular operations to be audited using SACLs, as explained here:

1. **NFS access to NTFS/mixed volume or qtree.** For an NTFS or mixed security style volume or qtree, you must set SACLs on the files and directories using the Windows Explorer GUI or fsecurity tool, as described in section 2.3 under “Specifying Auditable File Access Events (SACLs).”
2. **NFS access to UNIX volume or qtree.** For a UNIX security style volume or qtree and files without ACLs, it is necessary to configure an NFS log filter file. There are two steps:
   a. Create an empty log filter file (usually called `/etc/log/nfs-audit`) on the storage system. This file identifies which file events are included in the audit log by default.
   b. Set the **`cifs.audit.nfs.filter.filename`** option to identify the filter file:
      `options cifs.audit.nfs.filter.filename /etc/log/nfs-audit`

   Note:

   - You must create the NFS log filter file in an NTFS or mixed style volume or qtree. If you do not, you will not be able to set an SACL on the filter file, which is required for auditing.
   - SACLs set on individual files and directories take precedence over the SACLs set on the filter file.
   - Set the filter file's SACL using the Windows Explorer GUI or fsecurity as described in section 2.3 under “Auditing Access Events on Individual Files and Directories.”

NFS AUDITING FOR MULTIPLE QTREES

The NFS log filter file is a global file, and there can be only one per storage system. SACLs on this file would apply to all the UNIX file access events; that is, the same auditing setting would be applied to all of the UNIX qtrees. Therefore, to audit the NFS file access events in multiple UNIX security-style qtrees that have different auditing requirements, NetApp recommends configuring the Storage-Level Access Guard for each qtree as detailed in NetApp TR-3596: Storage-Level Access Guard Quick Start Guide.

2.3 HANDLING AUDIT EVENTS

SPECIFYING AUDITABLE FILE ACCESS EVENTS (SACLS)

Setting Auditing on Individual Files and Directories

There are two ways to set SACLs to audit access events on individual files and directories:

- Using the Windows Explorer GUI:
  1. Select the file or directory for which you want to enable auditing access.
  2. Right-click the file or directory and select Properties.
  3. Select the Security tab and click Advanced.
  4. Select the Auditing tab and add, edit, or remove the auditing options you want.
- Using the fsecurity command, as explained in NetApp TR-3597: Bulk Security Quick Start Guide

Note: Be sure to select only the events that must be audited because selecting too many audit options might affect system performance.

Setting Auditing on Volumes and Qtrees

To audit access events on all files and directories within a volume or a qtree, NetApp recommends that you set SACLs by applying Storage-Level Access Guard security. For more information, see NetApp TR-3596: Storage-Level Access Guard Quick Start Guide.

Note: SACLs can also be set on the volume or the qtree directly by using the Windows Explorer GUI, similar to an individual file or directory. The only caution would be that if there are SACLs applied to the child objects of that volume or qtree, then any user who has the privilege to modify the SACLs at those levels can unset the settings, and that specific subfolder or file will be skipped for auditing. SACLs applied through the Storage-Level Access Guard on the volume or qtree cannot be changed by the users at the child object level.

AUDITABLE EVENTS

- Log-on and log-off events (available only with CIFS-based access)
- Local user and group account management (available only with CIFS-based access)
- File access events at the file and directory level
- File access events at the qtree or volume level (using Storage-Level Access Guard)

For a list of all audit events that Data ONTAP captures, see Events That Data ONTAP Can Audit.

EVENT FORMAT

The Data ONTAP event log follows Windows `.evt` log format. This is the default Windows auditing file format in the Windows 2000 and Windows 2003 server environment. The `.evt` log schema can be understood in more detail using the Windows Event Viewer application. This article provides a general idea about the `.evt` file format.

PERSISTENT LOG DISPLAY

Log information can be saved in persistent logs using the static log feature in native auditing. By default, the event log file is `/etc/log/adtlog.evt`. This can be changed, however, using the **`cifs.audit.saveas`** option. You can configure to save log files using autosave configuration options.

Automatic Saving Based on Size of the Internal Log File

The default size threshold for the internal log file is 75%, so whenever the internal log file is 75% full, the contents are automatically saved to the external event file. You can specify the size threshold as a percentage (%), kilobytes (k), megabytes (m), or gigabytes (g).

```
options cifs.audit.autosave.onsize.enable on | off
options cifs.audit.autosave.onsize.threshold Nsuffix
```

`N` is the value of the size threshold, and `suffix` is the unit of measure.

Automatic Saving Based on a Time Interval

The default time interval is one day. You can specify the time interval as seconds (s), minutes (m), hours (h), or days (d).

```
options cifs.audit.autosave.ontime.enable on | off
options cifs.audit.autosave.ontime.interval Nsuffix
```

`N` is the value of the time interval, and `suffix` is the unit of measure.

Automatically Saved Event File Extensions

Each time the internal log file is automatically saved to the external event file, an extension is added to the base name of the event file.

- Counter-based extensions:

  ```
  options cifs.audit.autosave.file.extension counter
  ```

  Examples: `eventlog.evt`, `eventlog1.evt`, `eventlog2.evt`, and so on
- Timestamp extension:

  ```
  options cifs.audit.autosave.file.extension timestamp
  ```

  Format: base name of event file `YYYYMMDDHHMMSS.evt`

Maximum Number of Automatically Saved Event Files

This option specifies the number of simultaneous logs kept by the storage system:

```
options cifs.audit.autosave.file.limit value
```

The `value` takes a number between 0 and 999. The default value is 4. Once the limit is reached, the oldest file is purged or overwritten based on the extension option.

Note: Audit events can be saved manually by using the **`cifs audit save`** command. Executing the command replaces the latest log file. Old log files are rotated based on the file limit value, as explained in the preceding paragraph.

REAL-TIME DISPLAY

Log events generated in the NetApp controller can be viewed in close to real time using the Live View feature. When this feature is turned on, it provides updated audit information when any of these circumstances happen:

- A minute is completed.
- The internal buffers are 75% full.
- The user explicitly calls `CIFS audit clear`.

Characteristics of Live View

- When the Live View feature is enabled, it takes over from the autosave feature. In that case, on-size and on-time options are controlled by Live View.
- At any time, you can access information that is no older than a minute.
- Records are accessible through a circular-indexed file, and indexes are reused when 5,000 records are reached.
- The Live View feature allows access to the latest records—up to 5,000—on the controller.

Note: When events are overwritten, they are saved in a backed-up `.evt` file. Each overwrite causes the previous record to be backed up. The records can be accessed in the conventional CIFS way.

Configuring Live View

For Live View auditing, configure the option **`Enable Live View`** in Data ONTAP:

```
options cifs.audit.liveview.enable on | off
```

ACCESSING THE EVENT LOGS

With either the Live View or the static log feature enabled, you can access the logs on the NetApp controller using any of these methods:

- Static log files on storage can be accessed in the conventional CIFS way. You can specify the location for saving log files and protect that directory with appropriate rights.
- Static logs can be transferred through `ftp/sftp/scp` services, supported in the NetApp controller, to remote server boxes. Currently, however, there is no push strategy from the controller.
- Live View and static logs can be accessed through the event-logging API interface

Note: Currently, NetApp supports only the Unicode RPC calls. The event-logging APIs are supported in Windows 2000 and 2003. From Windows 2008 onward, Windows supports the new Windows event-logging API.

3 FPOLICY AUDITING

File policy, or FPolicy, is an event-notification framework within Data ONTAP that enables partner applications to connect to a NetApp storage system to monitor or control file access. FPolicy supports both NFS and CIFS protocols.

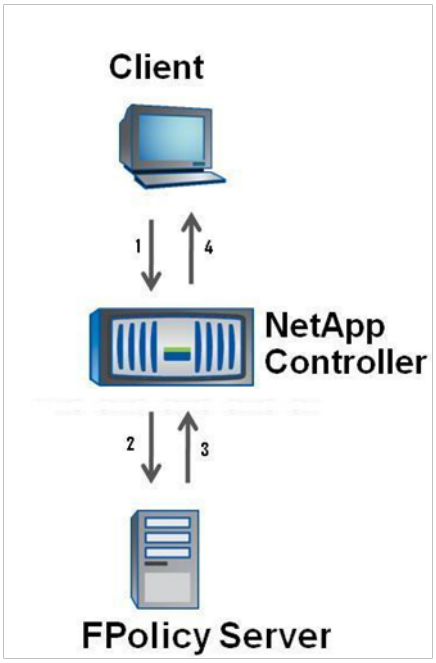
FPolicy notifications can be classified into two types: synchronous notifications and asynchronous notifications. For auditing, FPolicy must be set in asynchronous notifications mode. In this mode, the controller does not wait for a response from the FPolicy server, which enhances overall throughput of the system.

Figure 1 shows the interactions of the NetApp controller with the client and the FPolicy server. The numbered arrows indicate the order in which the interactions happen:

1. The client sends an NFS/CIFS request for file access.

2. The FPolicy engine in the NetApp controller sends a request notification to the external FPolicy server. This notification can be either synchronous or asynchronous.
3. The FPolicy server sends a response to the FPolicy engine.
4. The FPolicy engine sends the client a response, allowing access.

Figure 1) FPolicy framework.



3.1 PREREQUISITES

- CIFS must be licensed and enabled on the storage system before FPolicy auditing can be enabled.
- Configure an external server that listens to FPolicy in-wire notifications.
- Enable FPolicy and create a resident policy on the controller.
- The resident policy can be set to monitor:
  - Specific file operations
  - Specific volumes
  - Specific file extensions
- NetApp recommends using Data ONTAP version 7.3.3 or higher.

3.2 CONFIGURATION

FPolicy configuration involves configuring the NetApp controller on one hand and configuring the FPolicy server to receive FPolicy notification on the other. If a partner solution is installed, most of these configurations are done by the partner solution. It is still good to know what must be configured in order to get FPolicy to work in your environment. FPolicy configuration for auditing primarily involves the following points.

On the FPolicy Server

FPolicy auditing is not a standalone solution; it requires installing a partner solution on a server box to take complete advantage of this feature. In configuring the FPolicy server for auditing, keep these points in mind:

- High-bandwidth network connectivity is needed between the FPolicy server and the NetApp controller to be audited.
- The FPolicy solution must register with the NetApp controller using the async mode of communication.
- Configuring a secondary FPolicy server increases reliability.
- Network disruption between the controller and the FPolicy server will cause client outage for a few seconds.

On the NetApp Controller

With the following NetApp controller commands, you can:

- Enable or disable FPolicy:

  `options fpolicy.enable on | off`

  `(fpolicy` is the primary command used to manage file policy on the controller.)
- Create or destroy a file policy:

  `fpolicy create PolicyName PolicyType`

  `fpolicy destroy PolicyName`

  (`PolicyName` is the name of the policy being created. For `Policy Type`, the only value currently supported is `screen`.)
- Enable or disable specific policies:

  `fpolicy enable | disable PolicyName`
- Display information about existing file policies:
  - **`fpolicy`**

    (This command shows information about all the policies.)
  - **`fpolicy show PolicyName`**

    (This command shows information about the specific policy `PolicyName`.)
- Specify file extensions to include or exclude from screening:
  - **`fpolicy ext[ensions] inc[lude] add PolicyName ext-list`**

    Example: **`fpolicy ext inc add imagescreen jpg,gif,bmp`**

    (This command includes file extensions `jpg`, `gif`, and `bmp` in the list of files to be screened.)
  - **`fpolicy ext[ensions] exc[lude] remove PolicyName ext-list`**

    Example: **`fpolicy ext exc remove audioscreen wav`**

    (This command excludes the file extension `.wav` from the list of files to be screened.)
- Specify volumes to include or exclude from screening:
  - **`fpolicy vol[ume] inc[lude] add PolicyName vol[,vol]...`**

    Example: **`fpolicy vol inc add imagescreen vol1,vol2,vol3`**

    (Only files in the volumes `vol1`, `vol2`, and `vol3` will be screened by the file screening server.)
  - **`fpolicy vol[ume] exc[lude] add PolicyName vol[,vol]...`**

    Example: **`fpolicy vol exc add imagescreen vol4,vol5,vol6`**

    (Files in the volumes `vol4`, `vol5`, and `vol6` will not be screened by the file screening server.)

NetApp recommends the following settings:

- When FPolicy is configured for auditing, NetApp recommends setting `FPolicy required` to `off`.

  `fpolicy options <PolicyName> required [on|off]`

  (This will not block client operation in case the FPolicy server fails.)

- When FPolicy is configured for auditing, NetApp recommends setting `secondary_servers` to identify a secondary FPolicy server.

  ```
  fpolicy options <PolicyName> secondary_servers [<IP-address>
  ```

  (This will enhance product reliability in case of FPolicy/network failures.)

3.3 HANDLING EVENTS

AUDITABLE EVENTS

FPolicy supports most of the file and directory operations. It supports specific events based on the file access protocol. For example, **`file open`** is supported only in CIFS and NFS v4 and is not supported in NFS v3. Similarly, **`lookup`** and **`symlink`** are NFS-related file operations, so they are not supported in CIFS.

- FPolicy supports these generic events:

  `File open/create/rename/close/delete/read/write`

  `Directory delete/rename/create`

  `Setattr`
- FPolicy supports the following events only under NFS:

  `Getattr`

  `Create hard link/Sym link`

  `Lookup`
- FPolicy supports these notification-of-permission changes only under CIFS:
  - Change of owner/group
  - Change of SACL/DACL
- The NetApp controller can be configured to monitor specific events using the **`fpolicy mon`** command:

  `fpolicy mon[itor] {add|remove|set} <PolicyName> [-p {nfs|cifs|cifs,nfs}] [-f] <op_spec>[,<op_spec>]*`

  Using this command, you can specify a list of file operations for specific file access protocols such as NFS or CIFS or both.

  Note: FPolicy does not currently support shared create/delete operations.

EVENT FORMAT

Notifications are generated in a NetApp proprietary format. This is explained in more detail in the FPolicy SDK.

ACCESSING THE EVENT LOGS

The NetApp FPolicy framework provides two interfaces to manage FPolicy on the controller:

- ONTAPI® calls, which are used to set policy-related options; all operations that can be performed on the NetApp console can be performed through ONTAPIs
- DC-RPC calls, which are used by the FPolicy server to perform advanced operations such as registering to the policy, receiving notifications, and responding to notifications

3.4 FPOLICY AUDITING PARTNERS

NetApp auditing partners provide end-to-end solutions with reporting/alerting/compliancy features. These are some of the partner solutions:

- STEALTHbits offers StealthAUDIT. You can find more information here.
- NTP Software® offers NTP File Auditor™. You can find more information here.
- Varonis® offers Data Governance. You can find more information here.
- Symantec™ has Data Insight. You can find more information here.

4 COMPARISON BETWEEN AUDITING FRAMEWORKS

Table 1 compares native auditing with FPolicy auditing as offered by NetApp Data ONTAP. Note that this is not a comparison between partner solutions.

Table 1) Comparison of native auditing and FPolicy auditing.

	Native Auditing	FPolicy Auditing
Licensing requirement:	CIFS licensing	CIFS licensing
NFS auditing:	Simulation using NTFS ACLs	Supported completely*
Event formats:	Windows-like `.evt` format	NetApp proprietary format
Monitoring events:	Using SACLs or SLAG	Using **`fpolicy mon`** command
Real-time notification:	Close to real time	Real-time and on-wire notification
Saving events as persistent logs:	Supported on storage	Not supported
MultiStore® compliancy:	Completely supported	Completely supported
Snapshot™ file auditing:	Completely supported	Partner solution must support

* No mapping is required between UNIX users and Windows users.

5 REFERENCES

- Events That Data ONTAP Can Audit
  http://now.netapp.com/NOW/knowledge/docs/ontap/rel722/html/ontap/filesag/7multi49.htm
- EVT
  http://www.forensicswiki.org/wiki/EVT
- Governing Data: A Comprehensive Approach
  http://www.netapp.com/us/library/datasheets/ds-3106.html
- NetApp TR-3596: Storage-Level Access Guard Quick Start Guide, by Nagesh Sharyathi and Reena Gupta
  http://www.netapp.com/us/library/technical-reports/tr-3596.html
- NetApp TR-3597: Bulk Security Quick Start Guide, by Nagesh Sharyathi and Reena Gupta
  http://www.netapp.com/us/library/technical-reports/tr-3597.html
- NTP File Auditor
  http://www.ntpsoftware.com/products/FileAuditor.aspx

6 REVISION HISTORY

Date	Version	Comments
July 2011	2.0	Added sections on FPolicy-based auditing and Live View
June 2008	1.1	Revised content
July 2007	1.0	Initial release

Go further, faster®

www.netapp.com