Implementing file access auditing with NetApp FPolicy

In data-intensive industries and fields, data security classification and standards present significant challenges for major companies. This is especially true for those managing millions of file systems, trillions of files, and operations across multiple countries, each with its own set of standards and regulations.

Organizations often struggle with achieving comprehensive visibility into file access activities, which are crucial for maintaining security, compliance, and internal controls. Existing solutions might cover only a fraction of the data, leading to inefficiencies and increased risks of non-compliance. Additionally, the complexity of managing System Access Control Lists (SACLs) can further compound these challenges, requiring extensive effort to maintain.

How to use Splunk software for this use case

The Splunk Add-on for NetApp Data ONTAP is designed to enhance the Splunk platform's capabilities by integrating file event notifications directly from NetApp, helping you to build robust file access auditing (FAA) processes. The integration enables the Splunk platform to monitor and analyze file operations in real time.

File access auditing (FAA)

File access auditing (FAA) is critical for organizations that are concerned with security, regulatory compliance, and internal controls. FAA processes monitor and record actions (who, what, where, when, how) related to files on a file system.

In corporate environments, FAA is critical for several reasons:

- **Security**: By monitoring file access, organizations can detect and respond to unauthorized access attempts, potential data breaches, and other security threats.
- **Regulatory compliance**: Many industries are subject to strict data protection regulations. File Access Auditing helps ensure that companies comply with laws by providing a clear record of who accessed data, what actions were taken, and when these actions occurred.
- **Internal controls**: Auditing file access is also essential for maintaining internal controls and ensuring that employees follow company policies regarding data access and usage.

Monitoring files and folders accessed by clients is typically done through CIFS, NFSv3, or NFSv4.x protocols on various types of file systems, including NTFS and Unix (using NFSv3 permissions or NFSv4 ACLs).

FAA records actions related to files on a file system, including:

- **Creation**: Monitoring who creates new files and folders.
- **Modification**: Tracking changes made to existing files.
- **Deletion**: Recording when and by whom files are deleted.
- **Access**: Logging every instance of file access, including who accessed the file, when, and how.

This comprehensive monitoring process is crucial for maintaining security and compliance with data protection regulations. By having a detailed audit trail, organizations can quickly identify and mitigate potential security issues, ensuring data remains protected and operations stay compliant with industry standards.

FPolicy

FPolicy, a file policy technology from NetApp, can be used with the Splunk platform to create file access auditing (FAA) processes that enhance visibility into previously complex data sources. FPolicy is an infrastructure component of ONTAP that enables partner applications to monitor and set file access permissions.

FPolicy determines how the storage system handles requests from individual client systems for operations like create, open, rename, and delete, and provides visibility on file access.

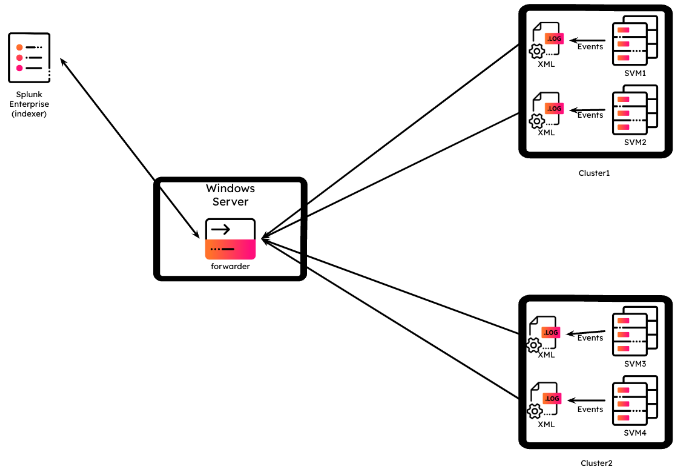
Transitioning to FPolicy eliminates the need for SACL management by allowing you to define file systems to be monitored directly within the FPolicy configuration. This change not only simplifies processes, but also enhances the security of infrastructure by removing the need for a proxy server to read the data.

There are no additional costs if you’re using both NetApp and the Splunk platform.

Implementing file access auditing using FPolicy and the Splunk platform

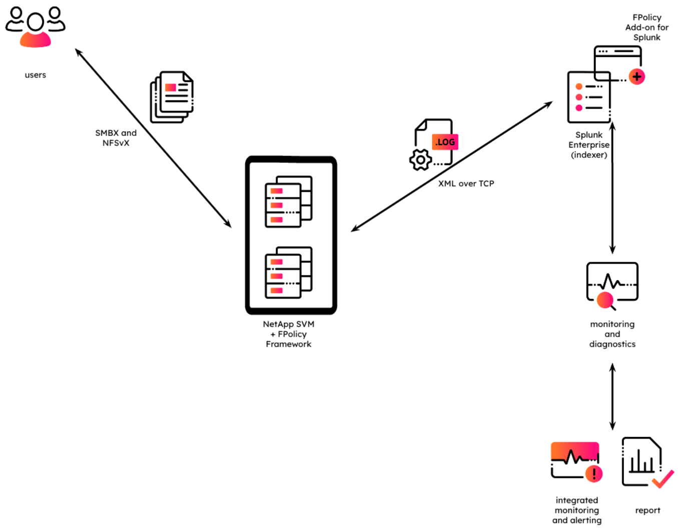
In large-scale corporate environments, millions of I/O operations (I/Ops) occur daily across thousands of file systems. In this example, the current topology for file access auditing involves a centralized logging system where all file access logs are collected and analyzed through the Splunk platform.

Logs are generated locally on the file system in each SVM (Storage Virtual Machine) and ingested into the Splunk platform by an agent (forwarder) that reads XML logs via CIFS.



Overall topology and how the handshake works

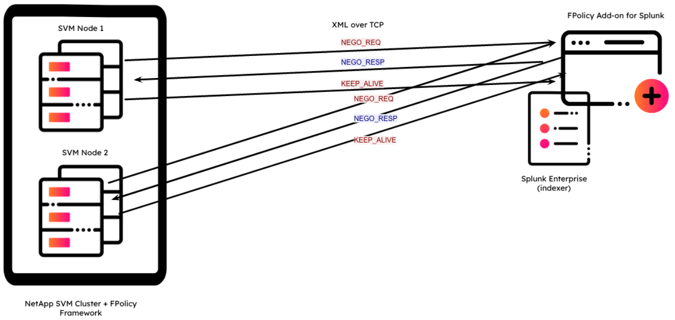
A TCP/IP connection is established between each node and the FPolicy Add-on for Splunk. TCP is used for external on-wire notifications, sending data over a network port.

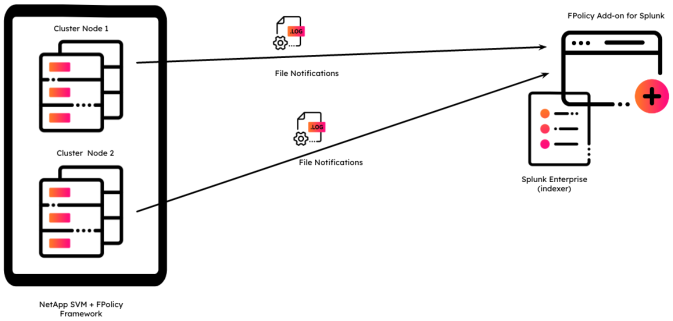

users
SMBX and NFSvX
NetApp SVM + FPolicy Framework
.LOG
XML over TCP
FPolicy Add-on for Splunk
Splunk Enterprise (indexer)
monitoring and diagnostics
integrated monitoring and alerting
report

- TCP/IP connection is established between each node and the FPolicy Add-on for Splunk .
- Two-way communication occurs only during the handshake process.
- After the TCP/IP connection is established between each node and the FPolicy Add-on for Splunk, nodes send file notifications while the Add-on for Splunk listens.
- The add-on only supports ASYNC mode as a communication mode.
- In ASYNC MODE as well TCP layer acknowledgement is there. (TCP Handshake)

Negotiation process

- **Negotiation request**: The node sends supported FPolicy protocol versions to the external server (the Splunk platform). It waits for two seconds for a response, and retries as per max-connection-retries (5).
- **Negotiation response**: The external server (the Splunk platform) responds with the selected protocol version. The node accepts the first response and discards duplicates.
- **Negotiation error**: If an unsupported protocol version is sent, the node sends an error, alerts the external server, and terminates the connection.


SVM Node 1
SVM Node 2
NetApp SVM Cluster + FPolicy Framework
XML over TCP
NEGO_REQ
NEGO_RESP
KEEP_ALIVE
NEGO_REQ
NEGO_RESP
KEEP_ALIVE
FPolicy Add-on for Splunk
Splunk Enterprise (indexer)


Cluster Node 1
Cluster Node 2
NetApp SVM + FPolicy Framework
.LOG
File Notifications
.LOG
File Notifications
FPolicy Add-on for Splunk
Splunk Enterprise (indexer)

Authentication mechanisms

The handshake process is initiated by the NetApp FPolicy policy. Without a user with admin rights, it is not possible to initiate a handshake request.

In the add-on configuration, the user should know the Policy Name set in the NetApp FPolicy policy.

Transport Layer 4 (TCP) ensures secure and reliable data transmission. The use of SSL can be configured to secure the transmission of this data. The policy supports secure transmission using SSL over TCP. This setup helps protect the integrity and confidentiality of file event data as it travels across the network.

FPolicy policy configuration and authentication mechanisms

To configure FPolicy, follow these steps:

1. Create FPolicy event: Determine what you want to audit, including the type of file events (for example, create, delete, or write) and the protocol to use (CIFS, NFSv3, or NFSv4).
2. Create FPolicy external engine: Specify the recipient of the events by providing the IP address and TCP port of the

FPolicy Add-on for Splunk.

3. Create FPolicy policy: Associate the events with the external engine (the Splunk platform).
4. Create FPolicy scope: Define the data you want to audit by specifying the granularity (for example, SVM, share, or export policy).
5. Enable FPolicy: The configuration granularity allows you to send events to different Splunk platform instances.
6. Create policy scope: Specify the data you want to audit.

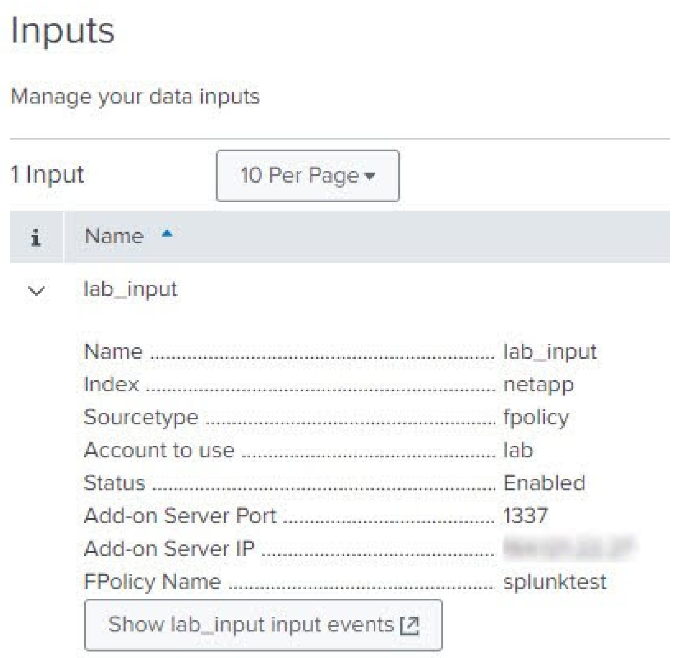
If your environment includes multiple SVM clusters, you can create additional inputs for each cluster. The key details to provide to the add-on include:

- Name: Any identifiable name for the input.
- Index: Specify the index where the data will be stored.
- Account: Use an account with the necessary permissions.
- IP Address: The IP address of the local instance (or use 0.0.0.0 for a generic input).
- Port: Ensure the port is not already in use.

The policy name used in the configuration must exactly match the FPolicy configuration. If they do not match, FPolicy might not consider the handshake response to be proper, potentially causing the policy to fail. Note that only one policy is allowed per protocol.

FPolicy communication mechanisms

FPolicy uses plain text XML over TCP/IP for communication between the Splunk Add-on for NetApp Data ONTAP and the cluster. This method applies to NAS protocols (CIFS, NFSv3, or NFSv4) and is responsible for collecting all file access event notifications, including screening and auditing.



Inputs

Manage your data inputs

1 Input

lab_input

Name	lab_input
Index	netapp
Sourcetype	fpolicy
Account to use	lab
Status	Enabled
Add-on Server Port	1337
Add-on Server IP	[redacted]
FPolicy Name	splunktest

The Splunk Add-on for NetApp Data ONTAP and NetApp clusters use plain text XML over TCP/IP for communication. The add-on supports various notification types, including `KEEP_ALIVE`, `NEGO_REQ`, and `SMB_DEL_DIR`, for protocols like CIFS, NFSv3, and NFSv4x. For CIFS, this add-on collects file access event notifications, including file access screening and auditing.

Example: When a directory is deleted

When a user deletes a directory in a storage virtual machine (SVM), an `SMB_DEL_DIR` notification is generated. For example, if a directory named my_file at the path \\NLABNASC9000\test\my_file is deleted using the SMB protocol version 3.1, several details are captured, such as the client IP address, user context, file ownership, and file attributes. This request is identified by a unique ID, like ReqId: 3591.

Here are the key details included in the request:

- ReqId: 3591, the unique request ID.
- ReqType: SMB_DEL_DIR, indicating a request to delete a directory.
- ClientIp: 192.168.17.245.
- GenerationTime: The timestamp of the request.
- UsrIdType: MAPPED_ID.
- Uid: 65533.
- WinSid: The Windows Security Identifier.
- FileOwner: Owner of the file.
- PathNameType: WIN_NAME for Windows paths.
- PathName: \my_file (Windows).
- FileSize: 4096 bytes.
- FileType: DIR, indicating it is a directory.
- DisplayPath: \NLABNASC9000\test\my_file.
- ProtVer: The protocol version used.

This information gives a detailed view of the directory deletion request.

Integrated monitoring and alerting in the Splunk platform

Using Splunk's props.conf and transforms.conf files, you can extract and filter data from raw logs, or anonymize sensitive information. You can also set up automated alerts in the Splunk platform based on specific triggers from these events. For instance, if a high number of delete operations is detected in a short period, an alert can notify administrators, helping them respond quickly to potential threats.

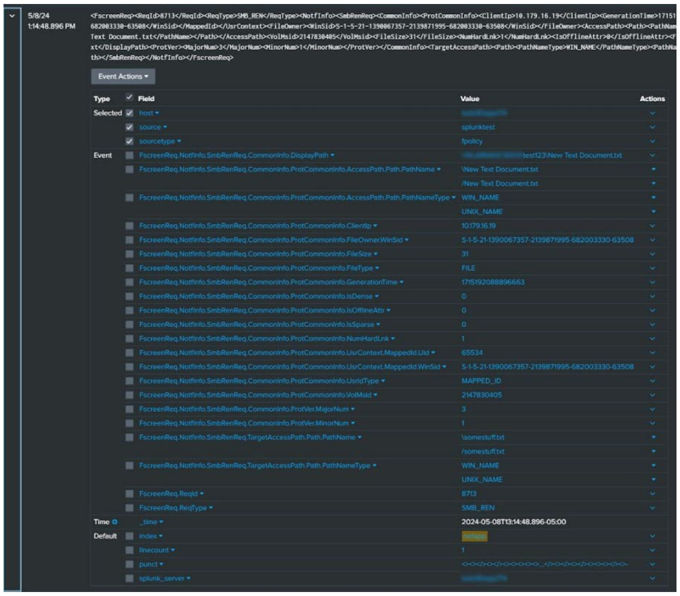
Additionally, you can generate useful metrics that can help in understanding usage patterns and your overall security posture, such as:

- The number of file access events per day

- Types of files most accessed
- Instances of unauthorized access attempts

Here's an example Splunk configuration:

```
[fpolicy]
BREAK_ONLY_BEFORE_DATE =
LINE_BREAKER = ((?:".+"|[\r\n]+)?<\?xml version="1.0"\?>)
TIME_PREFIX = <GenerationTime>
TIME_FORMAT = %s%6Q
NO_BINARY_CHECK = true
SHOULD_LINEMERGE = false
category = Custom
disabled = false
pulldown_type = 1
KV_MODE = xml
```



5/8/24
1:14:48.896 PM

```
<FscreenReq><ReqId>8713</ReqId><ReqType>SMB_REN</ReqType><NotfInfo><SmbRenReq><CommonInfo><ProtCommonInfo><ClientIp>10.179.16.19</ClientIp><GenerationTime>17151
682003330-63508</WinSid></MappedId></UsrContext><FileOwner><WinSid>S-1-5-21-1390067357-2139871995-682003330-63508</WinSid></FileOwner><AccessPath><Path><PathNam
Text Document.txt</PathName></Path></AccessPath><VolMsid>2147830405</VolMsid><FileSize>31</FileSize><NumHardLnk>1</NumHardLnk><IsOfflineAttr>0</IsOfflineAttr><F
xt</DisplayPath><ProtVer><MajorNum>3</MajorNum><MinorNum>1</MinorNum></ProtVer></CommonInfo><TargetAccessPath><Path><PathNameType>WIN_NAME</PathNameType><PathNa
th></SmbRenReq></NotfInfo></FscreenReq>
```

Type	✓	Field	Value	Actions
Selected	☑	host ▾	[redacted]	⌄
	☑	source ▾	splunktest	⌄
	☑	sourcetype ▾	fpolicy	⌄
Event	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.DisplayPath ▾	[redacted]test123\New Text Document.txt	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.AccessPath.Path.PathName ▾	\New Text Document.txt	▾
			/New Text Document.txt	▾
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.AccessPath.Path.PathNameType ▾	WIN_NAME	▾
			UNIX_NAME	▾
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.ClientIp ▾	10.179.16.19	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.FileOwner.WinSid ▾	S-1-5-21-1390067357-2139871995-682003330-63508	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.FileSize ▾	31	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.FileType ▾	FILE	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.GenerationTime ▾	1715192088896663	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.IsDense ▾	0	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.IsOfflineAttr ▾	0	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.IsSparse ▾	0	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.NumHardLnk ▾	1	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.UsrContext.MappedId.Uid ▾	65534	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.UsrContext.MappedId.WinSid ▾	S-1-5-21-1390067357-2139871995-682003330-63508	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.UsrIdType ▾	MAPPED_ID	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtCommonInfo.VolMsid ▾	2147830405	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtVer.MajorNum ▾	3	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.CommonInfo.ProtVer.MinorNum ▾	1	⌄
	□	FscreenReq.NotfInfo.SmbRenReq.TargetAccessPath.Path.PathName ▾	\somestuff.txt	▾
			/somestuff.txt	▾
	□	FscreenReq.NotfInfo.SmbRenReq.TargetAccessPath.Path.PathNameType ▾	WIN_NAME	▾
			UNIX_NAME	▾
	□	FscreenReq.ReqId ▾	8713	⌄
	□	FscreenReq.ReqType ▾	SMB_REN	⌄
Time		_time ▾	2024-05-08T13:14:48.896-05:00	
Default	□	index ▾	netapp	⌄
	□	linecount ▾	1	⌄
	□	punct ▾	<><></><></><><><><><>_.</><></><></><><><></><>-	⌄
	□	splunk_server ▾	[redacted]	⌄

In this article we've primarily focused on CIFS operations, but you can customize the source type to distinguish between CIFS and NFS events.

Next steps

These additional resources might help you understand and implement this guidance:

- Splunkbase: NetApp ONTAP FPolicy Add-on for Splunk
- Splunk GitHub: NetApp ONTAP FPolicy Add-on for Splunk
- .conf24: Enhancing visibility with Splunk and NetApp FPolicy