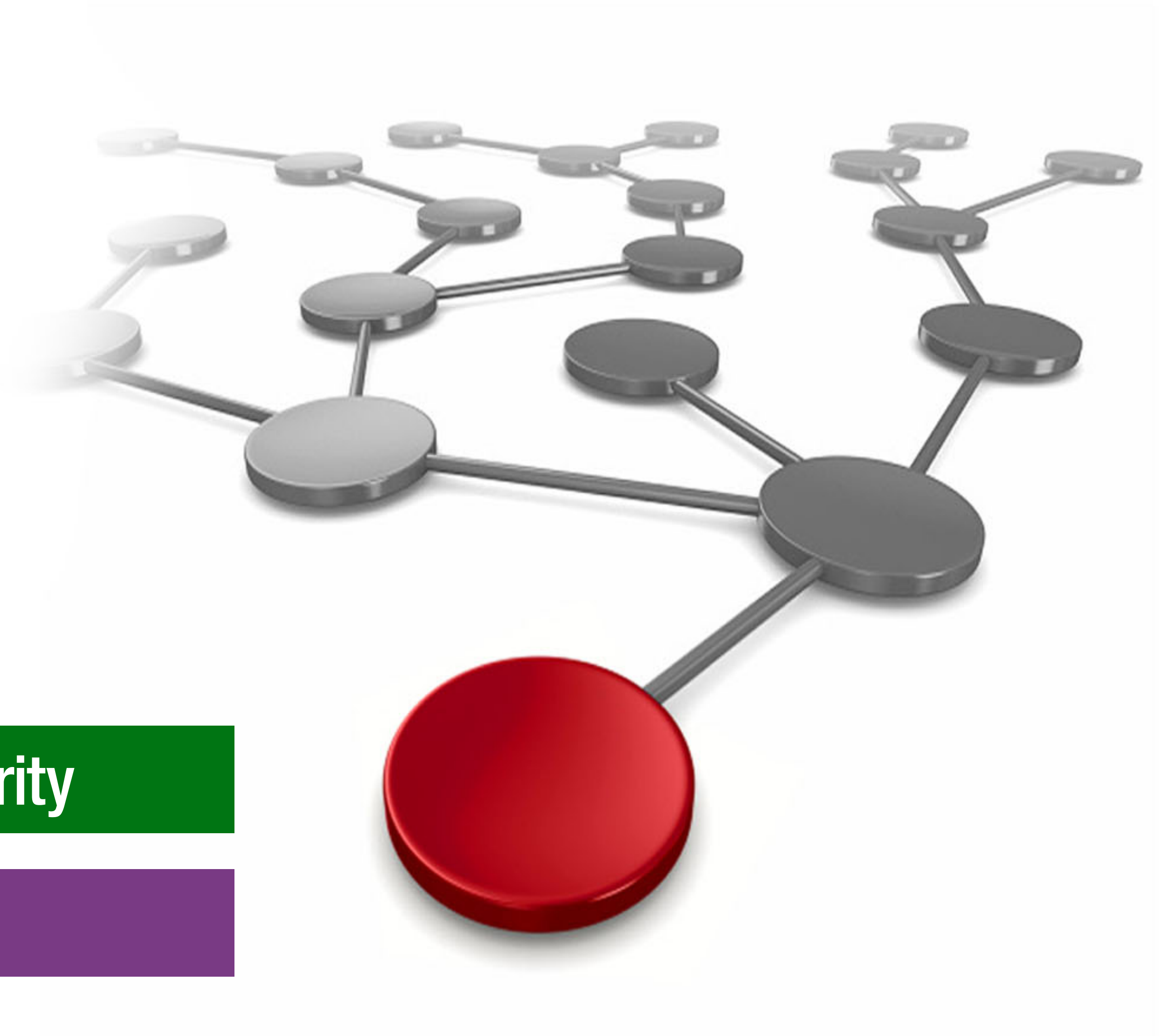
Redbooks
ibm.com/redbooks

IBM Spectrum Scale Immutability Introduction, Configuration Guidance, and Use Cases

Nils Haustein



Security

Storage

IBM

Redpaper

Summary of changes

This section describes the technical changes made in this edition of the paper and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for IBM Spectrum Scale Immutability Introduction, Configuration Guidance, and Use Cases
as created or updated on June 9, 2021.

June 2021, Third Edition

This revision includes the following new and changed information.

New information

- Updated IBM® Spectrum Scale limitations with regard to AFM DR.
- Updated section about “Backup” on page 17 with the recommended order of actions.

Changed information

- Updated for IBM Spectrum® Scale 5.1.

April 2020, Second Edition

This revision includes the following new and changed information.

New information

- Added Compliant-plus mode.

Changed information

- Updated for IBM Spectrum Scale 5.0.

August 2019, First Edition minor updates

This revision includes the following new and changed information.

Changed information

- Removed “Note:” from “PowerShell” on page 7.
- Updated second bullet of list in “NFS exports and SMB shares” on page 12.
- Added “Note:” to the “IBM Spectrum Scale limitations” on page 12.

IBM Spectrum Scale immutability introduction, configuration guidance, and use cases

IBM Spectrum Scale is a scalable parallel file system that can be used for many purposes. IBM Spectrum Scale allows configuring immutable partitions in a file system that are called *immutable filesets*. IBM Spectrum Scale immutable filesets were assessed for compliance in accordance to US regulations (SEC17a-4f) and German and Swiss tax laws and regulations.

From a user perspective, a fileset is a directory within an IBM Spectrum Scale file system. Immutable filesets allow managing immutable and append-only files, similar to the SnapLock method that was invented by NetApp, Inc.

Immutable filesets can also be exported by way of network file systems (NFS) and server message blocks (SMB). This ability makes it easy for applications that support the SnapLock semantic to adopt IBM Spectrum Scale as an immutable file storage.

By using the SnapLock method, files can be set to immutable or append-only for a specific retention time by using standard file system commands. During the retention time, immutable files cannot be deleted or modified. When the retention time expires, immutable files can be deleted but still not modified. With indefinite retention that is provided by IBM Spectrum Scale, deleting and modifying files can be prevented, even if the retention time expired.

With this immutability function, IBM Spectrum Scale can be used for archiving where regulatory requirement demand to prevent changes to and deletion of files during the lifecycle. The same IBM Spectrum Scale cluster also can be used for other purposes. In fact, the file audit logging function that was introduced with IBM Spectrum Scale version 5.0 uses immutable filesets to store file audit logs.

Applications that do not support the SnapLock-like immutability function can still benefit from IBM Spectrum Scale immutability because it allows files to be automatically set as immutable. Such applications can use IBM Spectrum Scale as a file storage by way of NFS, SMB, or as cluster member. IBM Spectrum Scale makes files immutable within minutes after they are stored.

IBM Spectrum Scale immutability offers more value by adding functions that are beneficial for archiving solutions. One key function is the storage tiering, which transparently migrates files to tape and saves cost over time.

The immutability function is also supported by IBM Spectrum Scale encryption, compression. and file audit logging. In addition, IBM Spectrum Scale offers a comprehensive set of techniques to assure high availability, data, and disaster protection.

Introduction to IBM Spectrum Scale immutability

This IBM Redpaper publication introduces the IBM Spectrum Scale immutability function. It shows how to set it up and presents different ways for managing immutable and append-only files.

This publication also provides guidance for implementing IT security aspects in an IBM Spectrum Scale cluster by addressing regulatory requirements. It also describes two typical use cases for managing immutable files. One use case involves applications that manage file immutability; the other use case presents a solution to automatically set files to immutable within a IBM Spectrum Scale immutable fileset.

Immutability means to associate a file with a retention time and prevent any changes to or deletion of the file data during the retention time. Immutable files are write-once-read-many protected (WORM) for a specific period, which can also be unlimited. After the retention time expires, the file can be deleted but not changed.

With the IBM General Parallel File System version 3.4 (IBM GPFS is now named IBM Spectrum Scale) the extended attributes “immutable” and “append-only” were introduced for files and directories in the IBM Spectrum Scale file system. These attributes can be set by the user who has permissions to set GPFS attributes by using the **mmchattr** command.

A file with the attribute “immutable” that is set to “yes” cannot be changed, renamed, or deleted. A file that features the attribute “append-only” is set to “yes”, allows append operations, but no other overwrites, renames, or deletions.

IBM Spectrum Scale allows setting these attributes on any file or directory, regardless if it is in a fileset. The downside of this legacy function is that these attributes can be reset by the user who has permissions to set attributes. In addition, retention times is not a concept that is used.

IBM Spectrum Scale version 4.1.1 enhanced the immutability function to IBM Spectrum Scale filesets. An IBM Spectrum Scale fileset can be configured with an integrated archive manager (IAM) mode by using the **mmchfileset** command. Files that are stored in such an immutable fileset can be set to immutable or append-only over a retention time by using standard POSIX or IBM Spectrum Scale commands.

The process to set a file to immutable is similar to the SnapLock method[1] in which a file can be set to immutable by setting the retention times by way of the file attribute “last access date” and by removing the write permissions. These commands cause the extended IBM Spectrum Scale attributes “immutable” and “append-only” to be set implicitly for the file in the IBM Spectrum Scale file system.

Depending on the IAM mode of the fileset, these attributes cannot be reset. In addition, the concept of retention times disallows modifications and deletions of the files during a defined period (retention or expiration time). When the retention time expires, files can be deleted but not modified.

The IBM Spectrum Scale immutability function in Version 5.1 was assessed for compliance in accordance to Securities and Exchange Commission (SEC) Rule 17a-4(f), Financial Industry Regulatory Authority (FINRA) Rule 4511(c) and the principles-based electronic records requirements of the Commodity Futures Trading Commission (CFTC) in 17 CFR § 1.31(c)-(d).

[1] The SnapLock method was invented by NetApp, Inc., and is state-of-the-art to make files immutable in network-attached storage systems (NAS). NetApp, Inc. and other storage vendors implement this method in their immutable NAS systems. The SnapLock methods that are referred to in this publication acknowledge that it is a trademark by NetApp, Inc.

The IBM Spectrum Scale immutability function in Version 5.0 was assessed for compliance by a globally recognized auditor in accordance to US regulations SEC17a-4f, EU GDPR Article 21 Section 1, German and Swiss trade laws and regulations.

IBM Spectrum Scale IAM modes

The Integrated Archive Manager (IAM) can be configured on an IBM Spectrum Scale fileset to prevent the modification and deletions of files. It essentially restricts some file operations (such as update, append, overwrite, rename, and delete) that are possible for files that are stored in normal (regular) IBM Spectrum Scale filesets.

A fileset can be considered a logical partition within a file system, which allows certain operations that are independent of the rest of the file system. From a user perspective, a fileset is a directory in the file system.

Functions that are available on a fileset include setting quota, taking snapshots, AFM caching, and IAM mode[2]. The IAM mode defines the level of protection that is applied to files in an IBM Spectrum Scale fileset. A fileset in which an IAM mode other than “none” is set is also called an *immutable fileset*; otherwise, the fileset is called a *regular fileset*.

Setting an IAM mode makes the fileset immutable. The IAM mode can be set by using the **`mmchfileset`** command with the parameter **`-iam-mode`**. A fileset can be set to one of the following IAM modes:

- None: No immutability mode is set (default); the fileset is a regular fileset.
- Advisory (ad): Allows setting retention times and immutability, but files can be deleted with the proper file permission.
- Noncompliant (nc): Advisory mode plus files cannot be deleted if the retention time did not expire. However, retention times can be reset and files can be deleted but not changed.
- Compliant (co): Noncompliant mode plus retention time cannot be reset. When the retention time expires, files can be deleted but not changed.
- Compliant-plus (cp): Includes enhancements to IAM mode Compliant. Directories cannot be renamed. Provides better compatibility with the NetApp SnapLock over SMB. Available with IBM Spectrum Scale Version 5.0.2 and above.

IAM modes can be upgraded from “advisory” to “noncompliant” to “compliant”, but not downgraded. When upgrading the IAM mode, intermediate levels can also be skipped; for example, the IAM mode can be upgraded from “none” to “compliant” in one step.

Only the IAM mode “compliant” was assessed for compliance.

It is possible to create nested filesets in which one fileset is within the other fileset. The IAM mode is not inherited. Therefore, when fileset1 is configured in compliant mode and fileset2 within fileset1 is a regular fileset, files that are stored in fileset2 cannot be managed in a compliant manner.

Immutable file operations

Files can be set to “immutable” or “append-only” in a regular fileset, a regular file system, or in an immutable fileset. A regular fileset and file system does not have an IAM mode set (IAM mode is set to none). An immutable fileset includes an IAM mode set other than “none”. The

[2] Although dependent filesets do not have their own inode-space, they do allow quota and IAM modes. Independent filesets have their own inode-space within the IBM Spectrum Scale file system and allow snapshots and AFM caching.

key differences between regular filesets and immutable filesets (assuming the immutable fileset is configured in compliant IAM mode) are listed in Table 1 on page 4.

Table 1 Key differences between regular filesets and immutable filesets

File or directory operation	Regular fileset[a]	Immutable fileset[b]
Reset immutability attribute	Yes	No
Reset append-only attribute	Yes	No
Set immutability by using mmchattr -i	Yes	Yes
Set append only by using mmchattr -a	Yes	Yes
Set retention time by using mmchattr -E	No	Yes
Set immutability by using chmod -w	No	Yes
Set append only by using chmod -l+w	No	Yes

[a] A regular fileset can also be the root fileset of a file system.

[b] Assumes that the fileset is configured in compliant IAM mode.

The fundamental difference between regular filesets and immutable filesets is that files in an immutable fileset can be made immutable or append-only by using standard POSIX commands. In addition, a retention time can be set for files in immutable filesets.

Working with files in immutable filesets

This section describes the steps creating immutable filesets and setting files to immutable or append-only in an immutable fileset. File immutability can be managed by using standard POSIX commands that are available in UNIX systems, by using specific IBM Spectrum Scale commands, or by way of SMB by using Microsoft Windows PowerShell.

Note: When files are stored in an immutable fileset, they do not become immutable automatically. Instead, commands must be used to set files to immutable (or append-only) and apply retention times.

Setting files to immutable or append-only by using POSIX commands is appropriate for applications or users who do not directly interface with IBM Spectrum Scale commands. Applications and users can manage immutable files by using POSIX commands from a cluster node with access to the IBM Spectrum Scale file system or by way of an NFS export. When POSIX commands are used from an NFS export, files cannot be set to append-only.

Setting files to immutable or append-only by using IBM Spectrum Scale commands is appropriate for applications and users that access the immutable fileset directly through the file system provided by IBM Spectrum Scale. Running IBM Spectrum Scale commands is possible from an IBM Spectrum Scale cluster node by a user with sufficient permissions (or privileges) only.

Note: The **mmchattr** command can be run by the file owner user (it does not have to be a administrator). Creating and changing the fileset mmcrfileset and mmchfileset requires administrative privileges.

Setting files to immutable by using PowerShell commands is appropriate for applications or users who accesses the immutable fileset by way of SMB share. Files can be set to immutable but not to append-only by way of SMB. Also, after the file is set to read-only, the retention time cannot be changed.

Creating an immutable fileset

The first step for working with immutable files is to create an immutable fileset. This process requires administrative privileges in the IBM Spectrum Scale cluster. Complete the following steps:

1. Create a fileset (independent in this case), as shown in the following example:

   ```
   # mmcrfileset <filesystem-name> <fileset-name> --inode-space new
   ```

2. Link the fileset to a directory within the IBM Spectrum Scale file system that must not exist now. This directory is the immutable fileset path, as shown in the following example:

   ```
   # mmlinkfileset <filesystem-name> <fileset-name> -J <directory>
   ```

3. Set an IAM mode for the files. In this example, we set the IAM mode “compliance”:

   ```
   # mmchfileset <filesystem> <fileset> --iam-mode compliant
   ```

4. Use the following command to list the IAM mode of a fileset:

   ```
   # mmlsfileset <filesystem> <fileset> --iam-mode
   ```

Note: From a compliance perspective, it is recommended to set the cluster-wide parameter `indefiniteRetentionProtection` to `yes`. Setting this parameter can be done by using the following command (it requires the cluster to be offline):

```
# mmchconfig indefiniteRetentionProtection=yes
```

Managing immutable files

The process to set files to immutable in an immutable fileset includes the following steps. This process requires appropriate access permissions to the file; this is, the user who is completing these steps must be the file owner:

1. Set the retention time.
2. Set the file to immutable or read-only.

This process corresponds the SnapLock method. Files can be set to immutable by using one of the following methods:

- Use of standard POSIX commands
- Use of IBM Spectrum Scale commands

The retention time can be extended but not reduced for immutable files in an immutable fileset that is configured in IAM mode “compliant”. Also, an immutable file in an immutable fileset cannot be renamed, overwritten, appended, or deleted. An immutable file can be deleted if the retention time (relative to the system date and time) expired.

POSIX commands

POSIX commands can be used when accessing the immutable fileset within the IBM Spectrum Scale cluster from a cluster node or by way of an NFS export. Two POSIX command can be used set a file to immutable and assign a retention time.

The retention time is encoded in the last access date of a file and can be set or extended by using the following POSIX command:

```
# touch -at 20300701000000 filename
```

The time stamp that is encoding the retention time features the following format:

```
[[CC]YY]MMDDhhmm[.ss]
```

The time stamp that is given by using the **touch** command in the previous example specifies the following date and time: 01.07.2013 00:00:00.

The file can be set to immutable by removing the write permissions by using the following POSIX command:

```
# chmod a-w filename
```

A file in an immutable fileset is implicitly set to immutable after the **touch** and **chmod** commands are run. Therefore, the IBM Spectrum Scale file attributes “immutable” and “expiration time” are set accordingly. To display the retention setting of the file, the following POSIX command can be used:

```
# stat filename
File: 'filename'
Size: 1              Blocks: 0          IO Block: 1048576
Device: 23h/35d Inode: 353793      Links: 1
Access: (0444/-r--r--r--)
Access: 2030-07-01 00:00:00.000000000 +0200
Modify: 2016-05-03 14:31:59.462278718 +0200
Change: 2016-05-03 14:32:16.915516097 +0200
```

The access pattern shows the permissions of the file that indicates read-only. The access time stamp indicates the expiration time. The retention time can be extended by using the same “touch” command[3].

IBM Spectrum Scale commands

IBM Spectrum Scale allows applications and users to directly set the retention time and the immutability attribute by using IBM Spectrum Scale commands. An IBM Spectrum Scale command is available with different parameters that can be used for this purpose.

The retention time can be set by using the following IBM Spectrum Scale command:

```
# mmchattr -E 2030-07-01@00:00:00 filename
```

The time stamp that is encoding the retention time has the following format: `YYYY-MM-DD@hh:mm:ss`.

The attribute immutable can be set by using the following IBM Spectrum Scale command:

```
# mmchattr -i filename
```

[3] Requires IBM Spectrum Scale version 4.2.0.2 or higher.

A file in an immutable fileset is implicitly set to immutable after the **mmchattr -E** and **mmchattr -i** commands are run. The following IBM Spectrum Scale command can be used to display the file attributes:

```
# mmlsattr -L filename
```

The output is shown in Example 1. The attribute `immutable` is set to `yes` and the expiration time is set to the time that is specified by using the **mmchattr -E** command.

Example 1   Command output

```
#mmlsattr -L filename
file name:            filename
metadata replication: 1 max 2
data replication:     1 max 2
immutable:            yes
appendOnly:           no
indefiniteRetention:  no
expiration Time:      Tue Jul  1 00:00:00 2030
flags:
storage pool name:    system
fileset name:         WORM
snapshot name:
creation time:        Tue Dec 21 21:18:58 2015
Windows attributes:   ARCHIVE
```

The retention time can be extended by using the same **mmchattr -E** command, as shown in Example 1.

PowerShell

File immutability with regard to retention times and read-only settings can be managed by using PowerShell. Setting files to immutable through an SMB share by using PowerShell requires that the immutable fileset or part of it is exported by way of SMB.

The retention time can be set by using the following PowerShell command:

```
# (dir filename).LastAccessTime = "2030-07-01 00:00:00"
```

The time stamp encoding the retention time features the following format: YYYY-MM-DD hh:mm:ss.

To set the file read-only, use the following PowerShell command:

```
# (dir filename).Attributes = "ReadOnly"
```

The file in an immutable fileset is implicitly set to immutable after these commands are run. The following PowerShell command can be used to display the file attributes:

```
# (dir filename | select Name,LastAccessTime,Attributes
```

The output is shown in the following example. The attribute `ReadOnly` is set and the retention time is encoded in the last access date:

```
# dir filename | select Name,LastAccessTime,Attributes
Name           LastAccessTime             Attributes
----           --------------             ----------
filename       01.07.2030 00:00:00        ReadOnly
```

Managing append-only files

The process to set files to append-only in an IBM Spectrum Scale immutable fileset consists of the following steps:

1. Set retention time for the file.
2. Set the file to read-only.
3. Set the file to read/write.

This process corresponds to the SnapLock method. Files can be set to append-only by using one of the following methods:

- Use of standard POSIX commands
- Use of IBM Spectrum Scale commands

Managing append-only files is only possible when the immutable fileset is accessed from a cluster node, not through NFS or SMB.

When a POSIX command is used, the file that is set to append-only must be an empty file. When an IBM Spectrum Scale command is used, the last two steps of setting the file to read-only and read/write can be combined into one step.

The retention time can be extended but not reduced for append-only files in an immutable fileset that is configured in IAM mode `compliant`. Also, data can be appended to an append-only file.

However, an append-only file cannot be renamed, overwritten, or deleted in a fileset that is configured in IAM mode `compliant`. An append-only file can be deleted if the retention time (relative to the system date and time) expires. An append-only file can also be set to immutable, which removes the capability to append any data to it.

POSIX commands

Setting files to append-only mode by using POSIX commands is possible only from a cluster node that accesses the immutable fileset directly, not through NFS or SMB.

The file to be managed in append-only mode must be empty. Therefore, the first step is to create an empty file and set its retention time. This process can be done by using the following POSIX command:

```
# touch -at 20300701000000 filename
```

The time stamp encoding for the retention time features the format `[[CC]YY]MMDDhhmm[.ss]`, which specifies the following date and time as `01.07.2013 00:00:00`.

The empty file can be set to append-only by removing and adding the write permissions by using the following POSIX command:

```
# chmod a-w filename
# chmod +w filename
```

A file in append-only mode cannot be deleted (unless its retention time expires) or overwritten, but data can be appended to it. The retention time can be extended by using the same **`touch`** command[4].

[4] Requires IBM Spectrum Scale version 4.2.0.2 or higher.

A file in an immutable fileset is implicitly set to append-only after the **touch** and two **chmod** commands are run. Therefore, the IBM Spectrum Scale file attributes `append-only` and `expiration time` are set accordingly. The following POSIX command can be used to display the retention setting of the file:

```
# stat filename
File: 'filename'
Size: 1              Blocks: 0         IO Block: 1048576
Device: 23h/35d Inode: 353793      Links: 1
Access: (0644/-rw-r--r--)
Access: 2030-07-01 00:00:00.000000000 +0200
Modify: 2016-05-03 14:31:59.462278718 +0200
Change: 2016-05-03 14:32:16.915516097 +0200
```

The access pattern shows the permissions of the file, which indicate read/write for the user. The access time stamp indicates the expiration time. It is not easy to see that the file is append-only because this state is not a file state in POSIX. The retention time can be extended by using the same **touch** command.

When no other data must be appended to the file, it can be set to immutable by using the following POSIX command:

```
# chmod a-w filename
```

This last step sets the attribute `immutable` to `yes` and the attribute `appendOnly` to `no` and disallow any appends. In IBM Spectrum Scale, the attribute `immutable` is set implicitly. The following POSIX command can be used to display the retention setting of the file:

```
# stat filename
File: 'filename'
Size: 1              Blocks: 0         IO Block: 1048576
Device: 23h/35d Inode: 353793      Links: 1
Access: (0444/-r--r--r--)
Access: 2030-07-01 00:00:00.000000000 +0200
Modify: 2016-05-03 14:31:59.462278718 +0200
Change: 2016-05-03 14:32:16.915516097 +0200
```

This method is similar to the method that was used with SnapLock. However, IBM Spectrum Scale does not support managing files in append-only mode by way of NFS or SMB.

IBM Spectrum Scale commands

One fundamental difference between POSIX and IBM Spectrum Scale commands for managing append-only files is that with IBM Spectrum Scale commands, the file does not need to be empty to make it append-only. Therefore, a normal file can be created and filled with content before it is set to append-only. Before setting a file to append-only, it is recommended to set a retention time by using the following IBM Spectrum Scale command:

```
# mmchattr -E 2030-07-01@00:00:00 filename
```

The time stamp encoding the retention time features the format: YYYY-MM-DD@hh:mm:ss.

Now, the file can be set to append-only by using the following IBM Spectrum Scale command:

```
# mmchattr -a yes filename
```

A file in append-only mode cannot be deleted (unless its retention time expired) or overwritten, but data can be appended to it. The retention time can be extended by using the **mmchattr -E** command.

A file in an immutable fileset is implicitly set to append-only after the **`mmchattr -E`** and **`mmchattr -a`** commands are run. The following IBM Spectrum Scale command can be used to display the file attributes within the IBM Spectrum Scale file system:

```
# mmlsattr -L filename
```

The output is shown in Example 2. The attribute `appendOnly` is set to `yes` and the Expiration time is set to the time that is encoded in the last access date.

Example 2 Output of mmlsattr -L filename command

```
#mmlsattr -L filename
file name:            filename
metadata replication: 1 max 2
data replication:     1 max 2
immutable:            no
appendOnly:           yes
indefiniteRetention:  no
expiration Time:      Tue Jul  1 00:00:00 2030
flags:
storage pool name:    system
fileset name:         WORM
snapshot name:
creation time:        Tue Dec 21 21:18:58 2015
Windows attributes:   ARCHIVE
```

When no other data must be appended to the file, it can be set to immutable by using the following IBM Spectrum Scale command:

```
# mmchattr -i yes filename
```

This last step sets the attribute `immutable` to `yes` and makes the file immutable. The IBM Spectrum Scale command (**`mmlsattr -L filename`**) reflects the flag’s immutable yes flag (see Example 3).

Example 3 Output showing flag’s immutable yes flag

```
#mmlsattr -L filename
file name:            filename
metadata replication: 1 max 2
data replication:     1 max 2
immutable:            yes
appendOnly:           yes
indefiniteRetention:  no
expiration Time:      Tue Jul  1 00:00:00 2030
flags:
storage pool name:    system
fileset name:         WORM
snapshot name:
creation time:        Tue Dec 21 21:18:58 2015
Windows attributes:   ARCHIVE
```

Note: It is normal that at this point that the append-only and immutable attributes are set to yes. It is not possible to reset the append-only attribute before setting the immutable attribute.

Indefinite retention

Immutable files can also be configured with an indefinite retention time. Indefinite retention essentially defines an indefinite retention time for the immutable file. It can be set and reset by the user who has permissions to change file attributes by using an IBM Spectrum Scale command. It cannot be set by using POSIX commands.

One use case for indefinite retention is the implementation of a deletion hold function. A deletion hold does not allow the deletion or expiration of a file, even if the retention time expired. It provides another level of protection above the retention time. For example, if an application or user wants to prevent a file from expiring, the indefinite retention attribute can be set.

To release the deletion hold, the indefinite retention attribute can be reset. If the indefinite retention is reset (set to “no”), the actual retention time of the file takes precedence. If this retention time is expired, the file can be deleted; if it is not expired, the file cannot be deleted.

The following IBM Spectrum Scale command can be used to set indefinite retention for an immutable file:

```
# mmchattr --indefinite-retention yes filename
```

To reset indefinite retention, run the following command:

```
# mmchattr --indefinite-retention no filename
```

Indefinite retention is useful only if the file is set to immutable or append-only. Otherwise, it has no effect.

Managing directories

Directories within an immutable fileset cannot be set to immutable (or append-only) explicitly. Directories within an immutable fileset become immutable automatically whenever files are stored within this directory. Table 2 lists operations on directories within an immutable fileset in accordance with the IAM modes.

Table 2 Directory operations within immutable fileset in accordance to the IAM mode

Directory operation	Advisory, Non-Compliant, Compliant modes	Compliant-plus mode
rename empty directory	Yes	No
rename non-empty directory	No	No
delete empty directory	Yes	Yes
delete non-empty directory	No	No
Set retention time on directory	No	No
Set directory immutable	Directories become immutable when files are stored in the directory. Directories remain immutable along with the immutable files that are stored in the directory	

The key difference between Compliant-plus mode and the other modes is that directories within an immutable fileset cannot be renamed, even no files are stored within the directory.

The fact that directories do not have immutability attributes differs from the SnapLock mode where even empty directories can be set to immutable.

Limitations

In this section, some limitations with IBM Spectrum Scale immutable filesets are explained that also influence the compatibility with SnapLock.

Directories

Most of the POSIX file system commands in an immutable fileset have the same effect as with the standard SnapLock method. One of the key differences from the standard SnapLock method is the concept of immutable directories.

With the standard SnapLock method, directories can be specifically set to immutable, which applies limitations to file operations within this directory. In an IBM Spectrum Scale immutable fileset, a directory becomes immutable implicitly and when a file is stored in the directory. Immutable directories cannot be renamed in Compliant-plus mode. Deletion of such directories is possible only if all files within this directory were deleted.

NFS exports and SMB shares

An immutable fileset can also be exported by way of NFS or SMB. This process includes the following limitations:

- It is not possible to set and manage files in append-only mode.
- With IBM Spectrum Scale version 5.0.3.2 and above, the retention period can be extended and expired files can be deleted over an SMB export. This process requires the IAM mode Compliant-plus of the fileset that is hosting the NFS export or SMB share.
- Extended attributes (POSIX) cannot be set by way of NFS.

Default retention times

An immutable fileset does not support minimum, default, and maximum retention times. This feature is another difference in NetApps, Inc.’s SnapLock function in which an immutable file system has minimum, default, and maximum retention times. However, it can be implemented by using IBM Spectrum Scale policies. For more information, see “Automatically setting files to immutable” on page 23.

IBM Spectrum Scale limitations

The following limitations apply when an immutable fileset is managed within the IBM Spectrum Scale cluster:

- With IBM Spectrum Scale version 5.1.1, Active File Management Disaster Recovery (AFM DR) supports immutable filesets on the primary and the secondary site. When you use immutable filesets with AFM DR, both the primary and the secondary filesets must be immutable filesets that are configured with the same IAM mode.
- Non-empty immutable filesets that are configured in IAM mode “compliant” or “compliant-plus” cannot be deleted by using the **`mmdelfileset`** command.
- File systems cannot be deleted if the cluster-wide parameter `indefiniteRetentionProtection=yes` is set, regardless if the file system contains an immutable fileset.

> **Note:** When a immutable file is copied within the immutable fileset, the destination file is not immutable, even though the immutability flag is set. To make the file immutable, you mist set the file to read-write (chmod +w) and read-only (chmod -w) and assign a retention period.

System configuration guidance

This section provides guidance for addressing IT security aspects. IT security aspects must be considered for the entire solution, including the application and user that are creating and managing immutable files, the IBM Spectrum Scale cluster, file systems that are storing immutable files, the operating system that is running on the cluster nodes, the storage system where data is stored, and the infrastructure environment. An overview of the key components that are considered from an IT security perspective is shown in Figure 1.

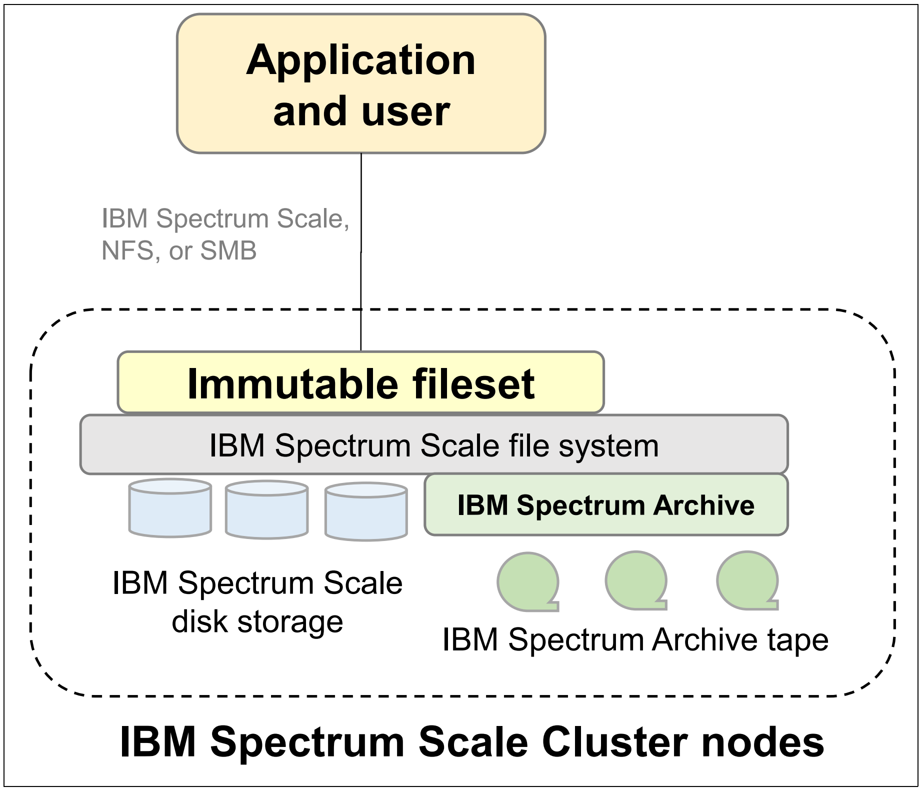

Application and user
IBM Spectrum Scale, NFS, or SMB
Immutable fileset
IBM Spectrum Scale file system
IBM Spectrum Archive
IBM Spectrum Scale disk storage
IBM Spectrum Archive tape
IBM Spectrum Scale Cluster nodes

Figure 1 IBM Spectrum Scale architecture with immutable filesets

The IBM Spectrum Scale cluster is configured with file systems and immutable filesets. The application and user accesses the immutable fileset by way of the IBM Spectrum Scale file system client directly from a cluster node by way of NFS exports or SMB shares (NFS or SMB) to store and retrieves files and manages their immutability.

The files are initially stored on IBM Spectrum Scale disk storage. Optionally, the files that are stored on disk can be migrated to tapes that are formatted in the Linear Tape File system format (LTFS) by the Spectrum Archive Enterprise Edition software.

Configuration guidance that is provided in this section applies to the application and user, the IBM Spectrum Scale cluster node configuration and administration, the IBM Spectrum Scale storage, the Spectrum Archive Enterprise Edition software and WORM tapes, and the operating system of the servers that are used for IBM Spectrum Scale, Spectrum Archive, and by the user and application (subsequently referred as *applications* in this publication).

This guidance is abstract in nature and suggests an appropriate security configuration and setup for IBM Spectrum Scale and Spectrum Archive. The client organization should use security architects to understand the implication of changes to their security policy and review their security architecture from end-to-end. This publication does *not* assure that the system cannot be compromised by using exploits and it is not intended to imply any specific audit guarantees (for more information, see "Disclaimer" on page 30).

For more information, see *IBM Spectrum Scale Security*, REDP-5426.

In addition, the general recommendations that are provided by the external auditor were considered.

User and application

The application uses the immutable filesets to read and write files and make files immutable with appropriate retention times. The application can access immutable filesets by way of the IBM Spectrum Scale file system client, NFS or SMB. From the application perspective, the following guidance should be considered:

- Application server uses the same time source for automated time synchronization as the IBM Spectrum Scale cluster hosting the immutable filesets.
- The application server is hardened to disallow tampering with data and metadata. Consider the guidance that is provided for IBM Spectrum Scale administration and Operating system configuration for the application as well.
- The application sets the file retention time and immutability in a timely manner (immediately after storing a file).
- The application tracks files where retention was not set (for example, because of failures) and set it later.
- The application manages the immutability and retention times of the files. The use of the GPFS command **`mmchattr -i | -E | -a`** is recommended.

  Managing retention by way of NF"S or SMB includes some limitations (for more information, see "Limitations" on page 12).
- The IBM Elastic Storage® Server provides internal check summing for data that is stored in immutable filesets. When IBM Elastic Storage Server is not used as storage for immutable files, consider calculating and storing checksums with immutable files when files are created.

  Checksums can be stored in IBM Spectrum Scale extended attributes (such as `user.cksum=checksum:value`) by using the command **`mmchattr --set-attr`** `user.chksum=checksum:value` filename. Periodically validate the checksums that are stored in extended attributes and create and retain an audit protocols for this validation process. The audit trail is stored in an immutable manner for the immutable files' retention time.
- Implement authentication and authorization according to the organizational standards and the methods that are supported by IBM Spectrum Scale.

IBM Spectrum Scale

The IBM Spectrum Scale file system provides immutable filesets with which files can be retained in an immutable manner. This function was assessed for compliance by independent auditors for different IBM Spectrum Scale versions. The immutability is managed by the application and IBM Spectrum Scale enforces it. Consider the guidelines that are described next for IBM Spectrum Scale configuration and administration.

Configuration

IBM Spectrum Scale provides a comprehensive set of IT security configurations and functions (for more information, see *IBM Spectrum Scale Security*, REDP-5426). In this section, we elaborate on some important setting that should be considered in combination with immutable files:

- Configure immutable fileset in compliant mode (iam-mode=compliant). Filesets that are configured in compliant mode cannot be set to non-compliant or advisory mode.
- Disallow file system deletion by setting the cluster-wide parameter by using the **`mmchconfig indefiniteRetentionProtection=yes`** command. After this parameter is set, it is not possible to delete file systems in the cluster. This parameter also cannot be reset.
- Implement audit logging for administrative IBM Spectrum Scale commands by using the **`mmchconfig commandAudit=yes|syslogOnly`** command. The IBM Spectrum Scale administrative command can be logged in the IBM Spectrum Scale log file or to the syslog file. The audit log file must be kept in an immutable manner during the retention time of the files that are stored in immutable filesets.
- Implement file audit logging by using the audit logging function that is provided with IBM Spectrum Scale version 5.0 and higher. With file audit logging, all major file operations, such as open, close, rename, ACL, and attribute changes and deletions, are logged into audit logs. The audit logs that are created are stored as immutable files that are in immutable filesets of an IBM Spectrum Scale file system. The retention time of audit log files can be configured.

  For prior IBM Spectrum Scale version use the audit logging function for NFS and SMB provided by Varonis.
- When migrating files to an external pool, ensure that these files are stored on a WORM medium, such as WORM tapes.
- Implement business continuity (which uses quorum, replication, backup, and clustering) that is based on SLAs.
- Implement IBM Spectrum Scale file encryption when required by the security policies.

Administration and sudo wrappers

IBM Spectrum Scale does not include its own user management for users who are using the command-line interface (CLI). CLI users are authenticated by the operating system and can run IBM Spectrum Scale specific commands.

By default, the IBM Spectrum Scale cluster requires root privileges to administer the cluster. The IBM Spectrum Scale sudo wrappers allow named users to administer the cluster by using sudo definitions.

> **Note:** The use of IBM Spectrum Scale sudo wrappers does not ensure that named users cannot elevate their privileges. Other operational measures are required according to the security policies that are deployed within the organization.

IBM Spectrum Scale sudo wrappers are not supported by the following components:

- Cluster Export Services, NFS, SMB, and Object
- Installation toolkit (command: **spectrumscale**)
- IBM Spectrum Scale call home
- With Windows nodes in the cluster

Users who are administering a cluster that contains immutable filesets cannot tamper with immutable files. Therefore, it might be preferable to allow only IBM Spectrum Scale and operating system commands that are required for normal operations for these users.

In special cases, the privileges of these users can be temporarily increased by another user. This feature allows for the implementation of the four-eye principle where the administrative user requires the authorization of another user (security administrator) to perform special task.

To implement the four-eye principle, create two user groups and assign named users to each of these groups in the operating system. In this example, one group is named `gpfsadmin` and another group is named `secadmin`. The following principles are applied:

- Users in the `gpfsadmin` group administer IBM Spectrum Scale and Spectrum Archive during normal operations.
- Users in the `secadmin` group manage the privileges of the users in the group `gpfsadmin`; for example, to temporarily allow more commands if problems occur. For this purpose, the user in the group `secadmin` can use sudo definitions.

The four-eye principle allows users of the `gpfsadmin` group to run a limited set of commands that is required for normal operations. If a user of the group `gpfsadmin` requires more privileges (for example, because of a problem), they can contact a user of the `secadmin` group. The user of the `secadmin` group can now grant the user of the `gpfsadmin` group more temporary privileges by changing the sudo definitions.

When configuring IBM Spectrum Scale sudo wrappers, allow users in the `gpfsadmin` group to run commands that are required to administer the IBM Spectrum Scale cluster and file systems. Use the default configuration in the `sudoers.sample` file, and adjust and test this according to their needs. Consider the following points:

- Users in the group `gpfsadmin` should not be able to run all commands; instead, limit this ability to the IBM Spectrum Scale commands that are required.
- Users in the group `gpfsadmin` should not be able to obtain full root permission without being authorized by a user of the group `secadmin`.
- Users in the group `gpfsadmin` should not be able to manage sudo configuration.
- Users in the group `gpfsadmin` should not be able to change the date and time that is related configurations in the operating system.
- Allow users in the group `secadmin` to change the sudo configuration (for example, by allowing the **visudo** command). This ability allows the user of the `secadmin` group to temporary allow certain commands for the user in the `gpfsadmin` group.
- Configure logging of all commands (including operating system commands) that are run by all user groups in the sudo context. These logs can be immediately sent to a remote log server and kept for the retention time of the files.
- Do not alter the set of commands for the group `gpfsadmin` in the sudo definitions that must run without password.

The IBM Spectrum Scale GUI features a separate user management. It is recommended to use the same user names in the GUI and the CLI for the same user. In addition, the roles of the user in GUI should match the roles that the user has in CLI.

When required, disable root logon by using SSH (`PermitRootLogin No in sshd_config`).

Monitoring

Monitoring assures that resource overload and system misbehavior is detected in time. It is highly recommended to configure event notifications by using the IBM Spectrum Scale GUI and send the appropriate events to the administrators by way of email or SNMP. In particular, monitor file system capacities and ensure sufficient capacity and that inodes are available always.

Alternatively, the IBM Spectrum Scale REST API can be used to monitor the cluster.

Access control

Accessing immutable files in the immutable fileset must be controlled by file permissions and access control lists (ACL). This control is typically done by the application or user that creates and works with files.

Configure access control in a way that disallows an IBM Spectrum Scale administrator (users in groups `gpfsadmin` and `secadmin`) accessing immutable files unless the business process requires this access. Do not use the IBM Spectrum Scale GUI to manage access to immutable filesets because this access might allow an administrator to gain data access privileges.

Backup

IBM Spectrum Scale offers the capability to back up immutable file to the IBM Spectrum Protect server by using the **`mmbackup`** command. It is recommended to back up immutable files. The backup process must be monitored and failures must be corrected in a timely manner.

If you backup and also migrate immutable files to tape, for example by using IBM Spectrum Archive Enterprise Edition, the following order of actions is recommended:

1. Set file to immutable with the final expiration date.
2. Backup the file using **`mmbackup.`**
3. Migrate the file using IBM Spectrum Archive Enterprise Edition.

Changing the immutability settings for a file (e.g. expiration time) will update the change time of the file. The subsequent backup process will backup the file again, because the extended attributes of the file have changed.

Immutable files must be restored to immutable filesets; otherwise, the immutability attributes (immutable, append-only, and retention time) are lost for the file that was restored. However, these attributes remain in the backup copy that is stored in the IBM Spectrum Protect server.

Note: Immutable filesets cannot be replicated by using IBM Spectrum Scale Active File Management (AFM) in any AFM mode.

Managing file immutability

The setting of files to immutable is managed by the application or done automatically by using policies with external scripts (for more information, see “Automatically setting files to immutable” on page 23).

It should be periodically assured that files within an immutable fileset are set to immutable with the appropriate retention times. For this purpose, LIST policies can be used to find files that are not set to immutable (for more information, see “List policy example” on page 27).

The administrator must be notified if files are in an immutable fileset that are not set to immutable. The check for non-immutable files should be performed periodically and with an audit trail created that includes the name of the files that are not immutable. This audit trail must be preserved in an immutable manner during the retention time of the immutable files.

When required, implement file-level encryption by using the IBM Spectrum Scale encryption function. Immutable files also can be migrated to WORM tapes by using Spectrum Archive Enterprise Edition.

Operating system configuration

The operating system runs on all IBM Spectrum Scale cluster nodes and on application servers that access the cluster by way of IBM Spectrum Scale, NFS, or SMB. Consider the following guidelines about securing operating systems:

- Configure the NTP or similar time synchronization method.
- Control and restrict access to remote time server and the client.
- Configure password rules (length, complexity, and expiration) according to security standards.
- Implement system log message forwarding (for example, rsyslog). Store the system log messages in an immutable manner during the retention time. The IBM Spectrum Scale commands should be logged to the system log (parameter `commandAudit=yes|syslogOnly`).
- Capture the sudo audit logs and forward them to a remote location (similar to rsyslog). These logs are kept in an immutable manner during the retention time of files.
- Capture operating system commands that are run by the users in the `gpfsadmin` group and `secadmin` group. Forward these logs to a remote location (similar to rsyslog). These logs are kept in an immutable manner or the duration of the retention time of files.
- Periodically gather GUI logs (var/log/cnlog/mgtsrv) and send them to remote location (similar to rsyslog). These logs are kept in an immutable manner during the retention period of files.
- The use of mmap (mapping files in memory) is not supported with files that are stored in immutable filesets.
- Disable unnecessary TCP ports and services by using the firewall.
- Implement patch management to immediately address vulnerabilities.
- Periodically audit the sudo environment (in particular, the sudo configuration). Audit trails must be kept in an immutable manner during the file retention time.
- Harden the operating system according to client standards and test these measures by using IBM Spectrum Scale.

Storage system

The storage is used by IBM Spectrum Scale to store immutable files on disk (hard disk, flash, and solid-state drive). IBM Spectrum Scale can use the following kinds of storage:

- IBM Elastic Storage Server with native RAID functionality (GNR)

The IBM Elastic Storage Server I/O nodes are IBM Spectrum Scale cluster nodes. The underlying disks are managed by the IBM Spectrum Scale GNR software. Protection can be enforced by protecting the IBM Spectrum Scale nodes.

- Internal disk within the NSD server

  The measures outline for the operating system configuration helps to protect the internal storage.

- External storage systems

  The disk system is external to the IBM Spectrum Scale nodes and attached by way of SAN, LAN, or InfiniBand. It might require more configuration work.

Consider the following guidelines for configuring the external storage system:

- Name administrators of the external storage system as users.
- Enforced role-based access control for administrators.
- Differentiate storage system administrators from IBM Spectrum Scale administrators.
- Audit the access to the external storage system and store the audit trails in an immutable manner during the file retention time.
- Encapsulate and isolate the administrative network (typically LAN) from other administrative networks.
- Use secure authentication and messaging (for example, SSH, https, or TLS) to administer the storage system.
- Encapsulate and isolate data networks (SAN, LAN, and InfiniBand) from other data networks, depending on the general security practices.
- Encrypt data over the data network, when required.
- Restrict and audit physical access to the storage system. Also, store audit trails in an immutable manner during the retention time.

IBM Spectrum Archive Enterprise Edition

In deployments where IBM Spectrum Archive Enterprise Edition is used the Spectrum Archive software runs on a subset of IBM Spectrum Scale cluster nodes.

Consider the following guidelines to configure IBM Spectrum Archive Enterprise Edition with IBM Spectrum Scale:

- Enable Logical Block Protection, which validates the data on the flight and at rest by using a CRC or Red Solomon algorithm. LBP includes performance effects. Those effects occur more with Red Solomon (approximately 80%) than with CRC32 (approximately 12%), as measured with TS1150 drives.
- Assure that the administrator for IBM Spectrum Archive Enterprise Edition does not use direct root login. IBM Spectrum Scale sudo wrappers can be used to enable administration of Spectrum Archive Enterprise Edition.

  Prevent access of administrators to the underlying LTFS file system (/ltfs).

- Immutable files should be migrated to WORM tapes. If a non-immutable file is migrated to a WORM tape and the file is changed later, it cannot be migrated to the same WORM tape pool again.
- The use of reclamation and reconciliation with WORM tape pools is not supported.

- Normal export and import of LTFS tapes is not supported with WORM tape pools. Although offline export is supported, it must export a copy tape and never the primary tape. Therefore, offline exporting requires two copies of files.
- Implement two copies of files on two tapes in different libraries and fire zones, when possible.
- Because the dual copy function in Spectrum Archive is not a backup, consider implementing the IBM Spectrum Scale backup function (`mmbackup`) with IBM Spectrum Protect.

When using the IBM Spectrum Scale backup function (`mmbackup`) for immutable files that are migrated by IBM Spectrum Archive Enterprise Edition, the following order of actions is recommended:

1. Set file to immutable with the final expiration date.
2. Backup the file using `mmbackup.`
3. Migrate the file using IBM Spectrum Archive Enterprise Edition.

Changing the immutability setting of a file (e.g. expiration time) will not cause a recall of the file. However, if the IBM Spectrum Scale backup function is used then the next backup cycle will backup the file. This may cause a recall.

For more information about the use of IBM Spectrum Archive Enterprise Edition with WORM tapes, see *IBM Spectrum Archive Enterprise Edition V1.3.1.2 Installation and Configuration Guide*, SG24-8333.

WORM tape

When IBM Spectrum Archive Enterprise Edition is used within the solution, WORM tapes should be used to store migrated files. The IBM TS1100 WORM tape technology also was assessed for compliance. Consider the following guidelines for configuring WORM tapes:

- Restrict and audit the physical access to the tape library. The audit trails should be stored in an immutable manner during the retention time of files.
- Configure Library Managed tape encryption in case the network (SAN) between the server and the WORM tape drive is not secure.

> **Note:** Use WORM tape and not standard read/write tapes with immutable files.

Environment

In this section, some general guidance is provided for the IT environment of the IBM Spectrum Scale solution.

Time server

The time server that is used for time synchronization of the IBM Spectrum Scale nodes and the application nodes must be identical. In addition, this time server must not be configurable by IBM Spectrum Scale or application administrators. Any manipulation of the time server must be prohibited.

In addition, the IBM Spectrum Scale and application administrators cannot manipulate the time service client (for example, ntpd and chronyd) on the IBM Spectrum Scale and application nodes. This requirement can be assured by using the proper sudo configuration.

Networks

The networks between application server and IBM Spectrum Scale nodes and the cluster network between the cluster nodes and the network to disk and tape (SAN) must be secure. If these networks are not secure, consider the use of other network security techniques, such as TLS, VPN, or end-to-end encryption.

Use cases

Managing immutable files often is done for archiving purposes when the regulatory requirements demand storing files in an immutable manner. An archive system is composed of an archive managing system and the archive storage.

The archive management system is an archiving application that drives the archiving process by collecting files to be archived from different sources (email servers, ERP systems, databases, file servers, and so on), indexing the content, and storing files in the archive storage. The archive storage is responsible for storing the files in an immutable manner. The archive storage is represented by IBM Spectrum Scale immutable filesets in the context of these use cases.

In this section, two use cases for IBM Spectrum Scale immutable files are presented. In the first use case, the archiving application supports the immutability function that is provided by IBM Spectrum Scale and manages immutability of files.

In the second use case, the archiving application does not support the immutability function to manage immutable files. Instead, it stores the files in an immutable fileset and IBM Spectrum Scale is configured to automatically set files to immutable in the background. This configuration is based on IBM Spectrum Scale policies with a custom script that periodically identifies files and sets the retention time and immutability attribute for these files. For more information, see “Automatically setting files to immutable” on page 23. To understand the functions of this use case, knowledge about the IBM Spectrum Scale policy engine is required.

Finally, more functions are provided by IBM Spectrum Scale and relevant for archiving are discussed. These functions address requirements for high availability, disaster protection, and tiered storage. For more information, see “Other IBM Spectrum Scale archiving functions” on page 26.

Application supporting immutable filesets

Applications that support the standard SnapLock method by way of the file system interface might support the immutability function in IBM Spectrum Scale because this process is similar to the SnapLock method. However, it is important to test and certify any application with IBM Spectrum Scale immutability before continuing.

As shown in Figure 2, the archiving application accesses the immutable fileset that is configured in an IBM Spectrum Scale file system to store files and manage file-level immutability.

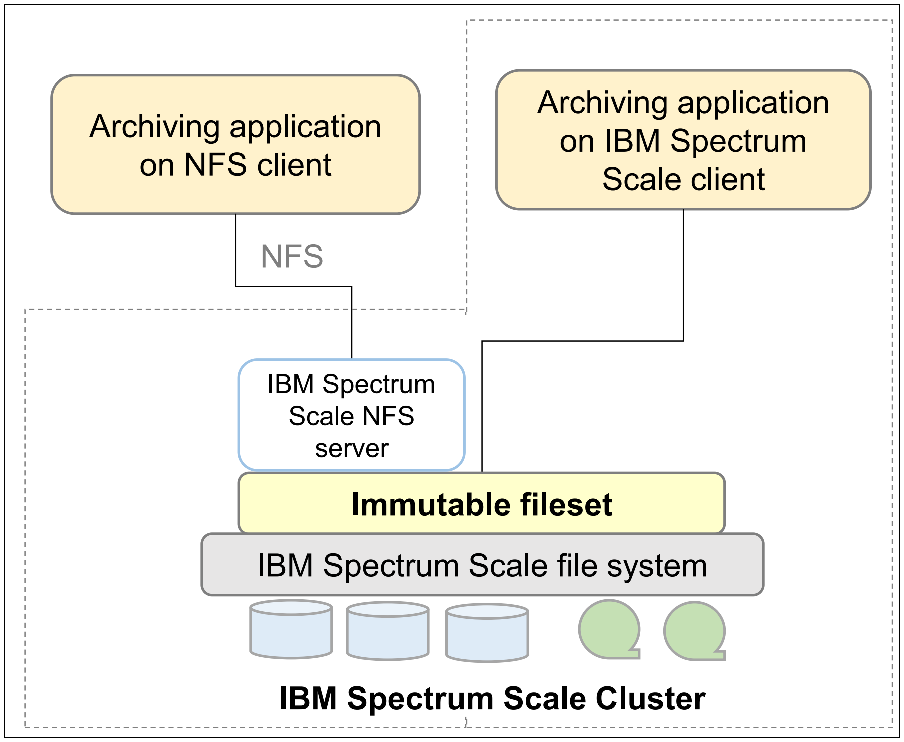

Archiving application on NFS client
NFS
Archiving application on IBM Spectrum Scale client
IBM Spectrum Scale NFS server
Immutable fileset
IBM Spectrum Scale file system
IBM Spectrum Scale Cluster

Figure 2 Architecture with application that uses IBM Spectrum Scale immutable filesets

The application can run on a server that is an IBM Spectrum Scale cluster member or access the immutable fileset through an NFS export. In the first case, the application can directly access the immutable fileset and use POSIX or IBM Spectrum Scale commands to manage immutable files. In the second case, the application can use POSIX commands to manage immutable files.

After the application stores files in the immutable fileset, it makes the file immutable by setting the retention time and immutability by using POSIX or IBM Spectrum Scale commands (for more information, see "Managing immutable files" on page 5). IBM Spectrum Scale assures that immutable files cannot be changed or deleted.

During the lifecycle of the file, the application can extend the retention time. The application typically tracks the retention time of files in its own index database and, if a file expires, the application deletes the file. Alternatively, the application reads the last access data of the file, which encodes the retention time in IBM Spectrum Scale to decide whether the file can be deleted.

In a similar way, applications can manage append-only files (for more information, see"Managing append-only files" on page 8). The only difference is that the application cannot manage append-only files by way of NFS. Therefore, the application must run on an IBM Spectrum Scale client, as shown in Figure 2 on page 22.

The archive application can also use indefinite retention feature. For example, when the application must prevent a file from being expired, it can set the indefinite retention for the file, as with a deletion hold (for more information, see "Indefinite retention" on page 11). This process assures that the file remains immutable, even if the retention time expires. Setting indefinite retention requires the application to run on an IBM Spectrum Scale client because it can be set and reset by using IBM Spectrum Scale commands only.

Although this example shows that the application is connected to the IBM Spectrum Scale cluster as cluster member or by way of NFS, it can also be connected by way of SMB. However, setting retention times and immutability by way of SMB requires different sets of commands or the use of UNIX toolkits because SMB is not POSIX compatible.

Automatically setting files to immutable

In this section, a solution is described that sets files to immutable with a predefined retention time by using the IBM Spectrum Scale policy engine. To understand the functions behind this use case, some basic knowledge about IBM Spectrum Scale policies is required.

Applications that do not support managing immutable files in an IBM Spectrum Scale immutable fileset can still benefit from the immutability function. Similar to the architecture that is shown in Figure 2 on page 22, the archiving application connects to the immutable fileset by way of NFS or SMB, or as IBM Spectrum Scale client directly.

The archive application stores files in the immutable fileset of the IBM Spectrum Scale file system. The IBM Spectrum Scale cluster is configured to automatically set files to immutable by using the IBM Spectrum Scale policy engine. The policy engine can be configured to run periodically (for example, every 30 minutes), possibly by using the cron-daemon.

The policy engine in IBM Spectrum Scale allows files to be identified based on file attributes and processes these files by using custom scripts. The identification criteria for files and the script that is processing these files are defined in a set of rules (also called policy). For this use case, all non-immutable files in a specific immutable fileset must be identified and processed. The processing that is run by a custom script sets the predefined retention time and makes the files immutable.

The following rules are required:

- The first rule defines the name of the custom script to be run for all identified files. In the following example, the script is named `makeimmutable.sh`[5]:

  ```
  RULE EXTERNAL LIST 'makeworm' EXEC 'makeimmutable.sh'
  ```

- The second rule describes the identification criteria of files that must be processed by the custom script `makeimmutable.sh`. The identification criteria is to select all files in the immutable fileset “archive” where the extended attribute “immutable” is not set:

  ```
  RULE 'notworm' LIST 'makeworm' FOR FILESET ('archive')  WHERE NOT
  (MISC_ATTRIBUTES LIKE '%X%')
  ```

For testing purposes, these rules can be written to a file (`policyfile.txt`). Replace the token EXEC `makeimmutable.sh` by EXEC '' and run these rules by using the policy engine in test mode:

```
# mmapplypolicy <filesystem-name> -P <policyfile.txt> -I defer -L 5
```

When running this command, the output shows a list of files that were identified based on the identification criteria in the policy rules.

Now, the custom script must be implemented. This script (`makeimmutable.sh`) is started by the policy engine with the following arguments:

- $1 is the policy operation that can be TEST or LIST.
- $2 for the operation TEST is the file system name for the operation LIST. This name is the name of the policy file, including all selected files.

[5] The script name must be specified with the full qualified path name in the EXTERNAL LIST rule.

The custom script must parse these arguments and act appropriately. For the TEST operation, it should check whether the file system that is specified with the second argument exists. For the LIST operation, a file list is passed that can be processed (see Example 4).

Example 4 Pseudo code that shows the processing a passed file list in an external scrip

```
# pseudo code to for an external script setting files to immutable
# default retention time is 1 year
Def_ret=365
case $1 in
TEST )
  # $2 is the file system path, check if this exists.
  if [ $2 exists ] then
    rc=0
  else
    rc=1
  fi
  ;;
LIST )
  # $2 is the name of the policy file
  # process the file list
  for each line in $2
  do
    extract_the_filename()
    if [ filename exists ] then
      set retentiontime: mmchattr -E $(current + $def_ret) filename
      set immutable: mmchattr -i yes filename
    fi
  done
  rc=0
  ;;
exit $rc
```

The file list that is passed to the custom script by the policy engine consists of multiple lines. Each line features five fields and contains one file name, as shown in the following example:

```
48900 1741777473 0   -- /mnt/filesystem/file1
```

The first three fields are GPFS internal numbers (inodenumber, inodegeneration, and snapid). The fourth field is meaningless for our example purpose. The file name is the fifth field in the file list. The function to extract the file name must parse this file list line-by-line and extract the file name. Be aware that the file name can include blanks.

Assigning different retention times based on file types

The retention time that is set for each non-immutable file can also be different for files with different characteristics; for example, different file types. This configuration requires some minor changes to the policy and custom script.

Assume that the retention time for files that end with `.tiff` should be set to 5 years and the retention time for files that end with `.pdf` should be set to 7 years. to pass the retention time that is based on file type to the custom script, the EXTERNAL LIST rule can be configured with an optional parameter that is denoted by the name: OPTS 'years'. The option `years` is an integer number that specifies the number of years to retain the specific file types. Four rules are required for assigning different retention times based on file types (two for teach file type).

Including macros to make it easier to read rules is shown in Example 5.

Example 5 Including macros

```
/* define some macros */
define( exclude_list, (PATH_NAME LIKE '%/.SpaceMan/%' OR PATH_NAME LIKE
'%/.snapshots/%' OR NAME LIKE '%mmbackup%' ))
define( immutable, MISC_ATTRIBUTES LIKE '%X%')

/* rule to set .tiff files to 5 years retention
RULE EXTERNAL LIST 'settiff' EXEC 'makeimmutable.sh' OPTS '5'
RULE 'mp3' LIST 'settiff' FOR FILESET ('archive') WHERE NOT (exclude_list) and NOT
(immutable) and (NAME LIKE '%.tiff')

/* rule to set .pdf files to 7 years retention
RULE EXTERNAL LIST 'setpdf' EXEC 'makeimmutable.sh' OPTS '7'
RULE 'pdf' LIST 'setpdf' FOR FILESET ('archive') WHERE NOT (exclude_list) and NOT
(immutable) and (NAME LIKE '%.pdf')
```

The custom script obtains a third argument from the policy engine that is the value that is encoded in the OPTS clause. In our case, this value is an integer number that specifies the number of years to retain a file of a certain type. Code that implements the variable retention time by file type is shown in Example 6 on page 25.

Example 6 Code that implements the variable retention time by file type

```
# pseudo code to for an external script setting files to immutable
# default retention time is 1 year
Def_ret=365
case $1 in
TEST )
  # $2 is the file system path, check if this exists.
  if [ $2 exists ] then
    rc=0
  else
    rc=1
  fi
  ;;
LIST )
  # $2 is the name of the policy file
  # $3 is the retention time, if not set give it a default time
  if [ $3 is not set ] then
    retime=$Def_ret
  else
    retime=$3
  fi
  # process the file list
  for each line in $2
  do
    extract_the_filename()
    if [ filename exists ] then
      set retentiontime: mmchattr -E $(current + $retime) filename
      set immutable: mmchattr -i yes filename
    fi
  done
  rc=0
  ;;
exit $rc
```

Example 6 gives an indication about how retention times can be flexibly assigned to files based on their attributes. The assignment of retention times can be based on file attributes, such as file type (extension), path name, size, user ID owning the file, and extended user attributes.

Other IBM Spectrum Scale archiving functions

Archiving is characterized by medium to large volumes of data that must be kept for long periods. During this time, access to the files should always be possible, even if an unwanted situation (such as a disaster) occurs.

Laws and regulations might demand that deleting and changing data during the retention time must be prevented. Because of the long lifetimes of archived files, it is important to manage cost for operations, power, and cooling. The IBM Spectrum Scale cluster can be configured to provide high availability, disaster protection, tiered storage, compression, encryption, and regulatory compliance of archived data.

High availability can be achieved by using IBM Spectrum Scale synchronous replication in combination with intelligent quorum techniques. IBM Spectrum Scale Quorum techniques assure that the cluster remains online, even if a subset of cluster nodes fails.

Synchronous replication copies files to two or three storage systems that are attached to the IBM Spectrum Scale cluster. In combination with multiple file system descriptor disks, the synchronous replication feature in IBM Spectrum Scale allows access to all data, even if one storage system fails.

Disaster protection can be achieved by using the integrated IBM Spectrum Scale backup function with IBM Spectrum Protect (formerly known as IBM Tivoli® Storage Manager). This function uses the **mmbackup** command to quickly create copies for immutable files in the IBM Spectrum Protect server. The backup copies can be stored in a disk pool, a deduplicated disk pool, in the cloud, or directly on WORM tape.

When backing up immutable files, it is important to set the files to immutable (or append-only) with the required retention time before the backup is run. Otherwise, if the files are set to immutable after the backup, these files become backup candidates in the next backup cycle. This status results because changing the file immutability attributes adjusts the CTIME of the file, which is used by mmbackup to identify candidates.

The tiered storage function in IBM Spectrum Scale places files on the most suitable storage technology during the entire lifecycle. This function is important for archiving, especially for large volume of data that must be retained for long periods because it provides optimal total cost of ownership over the lifetime of data. For example, archived files can be stored on a first tier of storage for a first period where access to the files is likely. When access to files diminishes or disappears, files can be transparently migrated to the next tier of storage. The use of tape as a next storage tier helps to reduce cost for maintenance, power consumption, and cooling.

IBM Spectrum Scale offers the hierarchical storage management function with IBM Spectrum Protect for Space Management (Tivoli Storage Manager HSM) or IBM Spectrum Archive Enterprise Edition to identify and migrate files to WORM tape. It is important to set the files to immutable (or append-only) with the required retention time before migrating these files to tape. For files that are backed up (by mmbackup) and migrated to tape (by IBM Spectrum

Protect for Space Management or IBM Spectrum Archive Enterprise Edition), it is important to set the files to immutable before backup and migrate the files after the backup is complete.

File identification is based on file attributes, such as file types, access time, and file size. Files that are identified are migrated to the next storage tier where they can be stored on WORM tape. Access to file in the IBM Spectrum Scale immutable fileset remains transparent, which means the application or user still sees the migrated file and, upon access, the file is fetched from tape by the WORM tape.

File encryption allows only authorized users to read the content of files and is one of the important data security function in IBM Spectrum Scale. Data encryption supports stringent requirements for data security that are demanded; for example, by the Payment Card Industry.

File compression provides storage capacity optimization and is another value that adds functions that are provided by IBM Spectrum Scale. File compression can be controlled on an individual file basis. In the context of archiving, compression is useful for data that does not change and must be kept for long periods to archive cost savings.

File and command audit logging creates comprehensive audit trails. File audit logging was introduced with IBM Spectrum Scale Version 5.0 audits file operations and stores the resulting audit logs in an immutable fileset.

Command audit logging sends all commands that are causing changes to the IBM Spectrum Scale cluster configuration to the syslog and from there, to a remote log server. These audit trails can be used to determine changes to files and the IBM Spectrum Scale cluster configuration.

The managing of immutable files in an immutable fileset, which is provided by an IBM Spectrum Scale cluster, was assessed for compliance by different independent auditors for different IBM Spectrum Scale versions in accordance to US regulations (SEC17a-4f) and German and Swiss tax and trade laws.

Examples

In this section, examples are provided for the use of list policies to identify immutable files. Also included are example for a sudo group configuration that can be used with IBM Spectrum Scale sudo wrappers.

List policy example

How to identify immutable files is shown in Example 7. It might need further adjustments for particular environments.

Example 7 List policy example

```
/* define macros */
define( exclude_list, (PATH_NAME LIKE '%/.SpaceMan/%' OR PATH_NAME LIKE
'%/.snapshots/%' OR NAME LIKE '%mmbackup%' OR PATH_NAME like
'%working-directory/%' ))
define( immutable, MISC_ATTRIBUTES LIKE '%X%')

/* external list rule */
RULE EXTERNAL LIST 'immut' EXEC ''
RULE 'listimmut' LIST 'immut' FOR FILESET ('worm') WHERE NOT (exclude_list) and
NOT (immutable)

/* to run this policy: mmapplypolicy fsname -P policy -f ./gpfs -I defer */
/* result is written to ./gpfs.list.immut */
```

The tag '`%working-directory/` stands for the directory in the file system that might be used as the working directory for the policy engine (parameters **`-s`** and **`-g`** of the **`mmapplypolicy`** command).

Sudo group example

The sudo configuration for the `gpfsadmin` and `secadmin` groups is shown in Example 8 on page 28. Users in the `gpfsadmin` group can run most of the GPFS and LTFS commands. They cannot run certain commands. These commands can be temporarily enabled by `secadmin` users. Users in the `secadmin` group can change the sudo configuration.

Note: This example of the group definitions was not fully tested and might require more adjustments in the context of a particular implementation.

Example 8 Sudo configuration for the gpfsadmin and secadmin groups

```
#### Allow members of gpfsadmin to run all mm-commands
%gpfsadmin ALL=(ALL) PASSWD: LOG_INPUT: LOG_OUTPUT: /usr/lpp/mmfs/bin/mm*
NOPASSWD: LOG_INPUT: LOG_OUTPUT:
/usr/lpp/mmfs/bin/mmremote, /usr/bin/scp, /bin/echo,
/usr/lpp/mmfs/bin/mmsdrrestore

#### disallow certain mm-commands for members of the GPFS admin group
%gpfsadmin ALL=(ALL) LOG_INPUT: LOG_OUTPUT: !/usr/lpp/mmfs/bin/mmfsadm,
!/usr/lpp/mmfs/bin/mmaddcallback, !/usr/lpp/mmfs/bin/mmchcluster,
!/usr/lpp/mmfs/bin/mmauth

#### allow members of gpfsadmin the following commands temporarily
%gpfsadmin ALL=(ALL) PASSWD: LOG_INPUT: LOG_OUTPUT: /usr/lpp/mmfs/bin/mmchconfig,

#### Allow members of the security admin group to change the sudo configuration
%secadmin ALL=(ALL) PASSWD: LOG_INPUT: LOG_OUTPUT: /usr/sbin/visudo

#### Add logging
Defaults iolog_dir=/var/log/sudo-io/%{user}
```

Note: The use of GPFS sudo wrappers does not ensure that users in the `gpfsadmin` group cannot elevate their privileges. In particular, the password-less use of `/usr/bin/scp` and `/usr/bin/echo` in the definition for the `gpfsadmin` group makes the solution vulnerable. More operational measures are required.

References

The following references were used to create this IBM Redpaper publication:

- IBM Spectrum Scale Knowledge Center: Immutability and append-only:

  https://ibm.biz/BdfSgT
- End-to-End checksums with IBM Spectrum Scale Native RAID:

  https://ibm.biz/BdYRgS
- Configuring sudo wrappers in an IBM Spectrum Scale cluster:

  https://ibm.biz/BdfSgw
- IBM Spectrum Scale Information Lifecycle Management Policies - Quick Start Guide

  https://www.ibm.com/support/pages/node/6260749
- IBM Spectrum Scale Information Lifecycle Management Policies:

  https://ibm.biz/BdfSgk
- IBM Spectrum Scale backup function using the mmbackup command:

  https://ibm.biz/BdfSgt
- Cohasset Associates assessment report for IBM Spectrum Scale V 5.1:

  https://www.ibm.com/docs/en/STXKQY/pdf/scale_assessment_report.pdf
- KPMG software certificate for IBM Spectrum Scale V 5.0 immutable filesets:

  https://ibm.biz/Bdqf47
- KPMG assessment report for IBM Spectrum Scale V 5.0 immutable filesets:

  https://ibm.biz/Bdqf4W
- IBM Spectrum Scale audit logging for file system activity:

  https://ibm.biz/BdfSgU
- IBM Spectrum Scale encryption:

  https://ibm.biz/BdfSg5
- IBM Spectrum Scale compression:

  https://ibm.biz/BdfSgN
- *IBM Spectrum Scale Security*, REDP-5426:

  https://www.redbooks.ibm.com/redbooks.nsf/RedpieceAbstracts/redp5426.html
- Assessment report for IBM TS1100 WORM tapes:

  https://ibm.biz/BdYRge
- IBM Spectrum Scale Authentication and Authorization:

  https://ibm.biz/BdfSg7
- IBM Spectrum Scale command audit logging:

  https://ibm.biz/BdfSgW
- Logical Block Protection with IBM Spectrum Archive:

  https://ibm.biz/BdYRgg
- IBM Spectrum Scale Firewall considerations:

  https://ibm.biz/BdfShc

- *IBM Spectrum Archive Enterprise Edition V1.3.1.2 Installation and Configuration Guide*, SG24-8333:

  http://www.redbooks.ibm.com/redpieces/abstracts/sg248333.html?Open
- IBM Spectrum Scale Monitoring overview:

  https://ibm.biz/BdfShB
- IBM Spectrum Scale management REST API overview:

  https://ibm.biz/BdfShd

Disclaimer

This document reflects the understanding of the author in regard to questions asked about archiving solutions with IBM hardware and software. This document is presented "as-is" and IBM does not assume responsibility for the statements expressed herein. It reflects the opinions of the author. These opinions are based on several years of joint work with the IBM Systems group. Direct any questions about the contents of this document to the author: nils_haustein@de.ibm.com.

The Techdocs information, tools, and documentation ("Materials") are provided to IBM Business Partners to assist them with customer installations. Such Materials are provided by IBM on an "as-is" basis. IBM makes no representations or warranties regarding these Materials and does not provide any guarantee or assurance that the use of such Materials will result in a successful customer installation. These Materials can only be used by authorized IBM Business Partners for installation of IBM products and otherwise in compliance with the IBM Business Partner Agreement.

This document provides guidance for certain configuration and operational aspects. IBM does not guarantee that this guidance complies with laws or regulation. To obtain a compliance assessment, an independent auditor must be engaged by the client. IBM cannot be made liable for any findings or violations of laws and regulations.

The software nature of the solution may allow malicious hackers to exploit the system and elevate privileges of users. The use of GPFS sudo wrappers does not completely assure that there is no way to escape the sudo environment and elevate privileges of users.

In addition, it might be possible to use certain undocumented GPFS commands to exploit the system and elevate privileges of users. IBM cannot be made liable for any damage resulting of the use of exploits.

The guidance given herein does not imply warranty that the commands given or their intention satisfies the purpose. IBM cannot be made liable upon damage caused by any of the commands or guidance.

Author

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Tucson Center.

Nils Haustein is a Senior Technical Staff Member at IBM Systems group and is responsible for designing and implementing backup, archiving, file, and object storage solutions in EMEA. He co-authored the book *Storage Networks Explained*. As a leading IBM Master Inventor, he has created more than 200 patents for IBM and is a respected mentor for the technical community worldwide.

Thanks to the following people for their contributions to this project:

Prashant Sodhiya (IBM Spectrum Scale development) for his help with the certification.

Kuei-Yu Wang-Knop and Haizhu Liu (IBM Spectrum Scale development) for the thorough review of this paper and great collaboration to make this function available.

Ulf Troppens (IBM Spectrum Scale architect) for valuable feedback after reviewing this document.

Mark Roberts of AWE, UK for the great feedback regarding system security configuration guidance.

Felipe Knop (IBM Spectrum Scale development) for the guidance about security, and especially sudo wrappers.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new IBM Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

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

Trademarks

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

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

IBM®
IBM Elastic Storage®
IBM Spectrum®
Redbooks®
Redbooks (logo) ®
Tivoli®

Other company, product, or service names may be trademarks or service marks of others.

REDP-5507-02

ISBN 0738459771

Printed in U.S.A.