NETAPP TECHNICAL REPORT

FPolicy Safeguards in Data ONTAP

Syed Amin, NetApp
TR-3640

ABSTRACT

FPolicy applications such as quota management, file access control, hierarchical storage management, and so on are handled by NetApp storage systems via integration with external servers called *FPolicy servers.* Because FPolicy servers are in the data path, they need to keep up with the flow of the traffic so that the NetApp storage system can respond to client requests in a timely manner.

An undersized FPolicy server, a network problem between the NetApp system and the FPolicy server, or an error in the FPolicy server hardware or software can cause delays in servicing client requests, and in extreme cases can even cause NFS and CIFS services on the NetApp system to become unavailable.

This document introduces FPolicy safeguards and discusses how to use these safeguards to prevent CIFS and NFS resources on the NetApp system from becoming unavailable as a result of the scenarios just described.

TABLE OF CONTENTS

1 INTRODUCTION TO FPOLICY

FPolicy refers to the NetApp storage system's file policy feature, which provides a flexible way to implement file policies through the use of a specialized server. It is an infrastructure that allows external servers to register for certain file system events (file open, create, remove, rename, directory create, and so on) with the NetApp storage system, and to receive callbacks from the storage system when these events actually occur as a result of a client access.

FPolicy requires CIFS to be licensed and running, even in NFS-exclusive environments. The rationale for this is that an FPolicy server wields a lot of power, and it is thus authenticated by using CIFS security to ensure that the server has Backup-Operator privileges (or more) on the NetApp storage system. This authentication requires that CIFS be licensed on the NetApp storage system. To apply file policies to NFS files, you must also have NFS licensed and running.

There are many possible uses of FPolicy, including various file-access logging products, quota management, hierarchical storage management (HSM), encryption/decryption, compression/decompression, and more.

2 WHY SAFEGUARDS ARE NEEDED IN FPOLICY

FPolicy applications work by sending requests to the external server via the FPolicy interface. The FPolicy server processes the requests and responds to the NetApp system, which then responds to the client's request. Because the FPolicy servers are in the path of the client's request, they are expected to keep up with incoming requests so that the NetApp system can respond to the client's request in a timely manner. An undersized FPolicy server, a bad network connection between the NetApp system and the FPolicy server, or a flaw in the FPolicy server application can cause problems such as delays and timeouts in processing the client's requests.

3 THE SCOPE OF THE SAFEGUARDS BUILT INTO DATA ONTAP

The types of problems that can occur as a result of a slow, undersized, or malfunctioning FPolicy server can be divided into two categories:

1. They can cause the NetApp system to respond slowly or not respond at all to requests from a CIFS or NFS client that is using FPolicy services.

2. They can cause resources on the NetApp system to get indefinitely tied up with pending FPolicy requests, leading to unavailability of CIFS or NFS services to all clients on the NetApp storage system, regardless of whether or not these clients use FPolicy services.

The problems in the first category depend on FPolicy applications running on third-party servers external to the NetApp system, so this category of issues cannot be addressed completely by fixes in the NetApp systems. They have been partially addressed by the FPolicy safeguards in Data ONTAP®. To completely address these issues, the third-party FPolicy applications need to be fixed.

Problems in the second category (such as a complete shutdown of CIFS services), can be completely addressed in the NetApp storage systems, and the safeguards described in the following sections have been designed to address these issues completely.

4 VERSIONS OF DATA ONTAP WITH THE FPOLICY SAFEGUARDS

Starting with the following releases, FPolicy safeguards are built into Data ONTAP, allowing the NetApp system to detect slow, undersized, or malfunctioning FPolicy servers and to take the appropriate actions:

Note: In versions of Data ONTAP prior to 7.3, these safeguards are added as hidden features.

- Data ONTAP 7.2.3P8 (and subsequent releases)

- Data ONTAP 7.2.4P1 (and subsequent releases)
- Data ONTAP 7.2.5 (and subsequent releases)
- Data ONTAP 7.3.0 (and subsequent releases)

5 USING THE FPOLICY SAFEGUARDS

All of these safeguards are disabled by default, and each safeguard must be enabled individually.

5.1 Skip Server Screening on CIFS Requests If the Underlying CIFS Session Is Disconnected

Requests associated with disconnected sessions add unnecessary load on the FPolicy server. This feature makes it possible to detect these requests and handle them appropriately.

When this feature is enabled, if the underlying CIFS session for a CIFS request is disconnected, the request is dispatched without screening (the request is dispatched with status and flags set as if the request were denied by FPolicy).

The feature can be enabled on individual file policies by using the following command:

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

The following is an example of how to view the current `cifs_disconnect_check` setting:

```
NetApp> fpolicy options p1 cifs_disconnect_check
fpolicy options p1 cifs_disconnect_check: off
```

The following is an example of how to set a new value for the `cifs_disconnect_check` setting:

```
NetApp> fpolicy options p1 cifs_disconnect_check on
```

Note: The default value is off.

5.2 Limit on the Number of CIFS Requests Held Up in FPolicy

Each NetApp system has a fixed number of pBLKs that it uses to handle CIFS requests. (This number varies between different NetApp models, depending on the system's cache size and a few other factors.) Every time a CIFS request comes in a pBLK is used, and when the request is completed, the pBLK is freed up and placed back into the pool of available pBLKs. If all of the pBLKs in the pool get used up, due to a large number of CIFS requests, CIFS services become temporarily unavailable to clients.

This feature is being introduced to allow the user to impose a limit on the number of pBLKs (and therefore CIFS requests) that can be held up in FPolicy. When the limit is reached, new CIFS requests are not sent for server screening.

This feature can be enabled by setting the `fp_maxcifsreqs_pblkpercent` flag and thereby setting a limit on the number of pBLKs for FPolicy as a percentage of the total number of pBLKs available on the NetApp system.

The `setflag` command can be used as shown in the following example:

```
NetApp> priv set diag
NetApp*> setflag fp_maxcifsreqs_pblkpercent 30
```

The `printflag` command can be used to view this setting, as shown in the following example:

```
NetApp*> printflag fp_maxcifsreqs_pblkpercent
fp_maxcifsreqs_pblkpercent = 30
```

Note 1: This feature is disabled if it is set to a value of 0, or if the value is outside the range of 1 to 100. The default value is 0 (disabled).

Note 2: To see the maximum number of pBLKs available on the NetApp system, run the command `CIFS stat` and look for the field `Max pBLKs = xxx`.

Note 3: This value is not persistent across reboots and must be set every time the NetApp system boots. Setting this value at boot-up time can be automated by adding these commands to the `/etc/rc` file in the NetApp system's root volume.

5.3 Time Limit on How Long an FPolicy Server Can Be in the "Throttled" State

NetApp systems impose a limit on the number of simultaneous requests that can sent to an FPolicy server at any time (the default setting is 50 simultaneous requests, and this value is configurable). When an FPolicy server reaches this limit, it is considered to be in a "throttled" state and no new requests are sent to it until some of the existing requests are completed, dropping this number to a value below 50.

Until now, there was no limit on how long an FPolicy server could be in the throttled state, so there was a possibility that an FPolicy server might be stuck in the throttled state for an indefinite period of time.

This feature has been introduced to allow users to place a limit on the amount of time that an FPolicy server can remain in the throttled state. If the FPolicy server reaches this time limit, the following actions are taken:

The FPolicy server is disconnected from the NetApp system.

New requests are not sent to the FPolicy server (because the FPolicy server is disconnected from the NetApp system).

Requests being held up by this server are canceled (and the canceled requests are retried with other FPolicy servers or policies, depending on how policies have been set up).

This feature can be enabled for individual file policies by using the following command:

```
fpolicy options <PolicyName> serverprogress_timeout [timeout-in-secs]
```

The following example shows how to view the current `serverprogress_timeout` setting:

```
NetApp> fpolicy options p1 serverprogress_timeout
fpolicy options p1 serverprogress_timeout: 0 secs (disabled)
```

The following example shows how to set a new value for `serverprogress_timeout`:

```
NetApp> fpolicy options p1 serverprogress_timeout 600
```

Note: A value of 0 means that this feature is disabled. The default value is 0.

5.4 Limit on How Long an Individual Request Can Be Held Up in an FPolicy Server

Even with the safeguards discussed in the previous sections, at times individual requests can get stuck in the FPolicy server. Such requests are handled by the CIFS protocol timeout on the client side, but their build up can also have adverse effects on the NetApp system by locking up resources on the system.

This feature has been designed to limit the length of time that an individual request can be held up in an FPolicy server. If this time limit is exceeded, the request is canceled and the canceled request is retried with other FPolicy servers or policies (depending on how the policies have been set up). If no other servers or policies exist, the request is handled according to local file blocking rules. (For details on local file blocking, see the “File Access & Protocols Management Guide“ in the product documentation section of the http://now.netapp.com website).

This feature can be enabled for individual file policies by using the following command:

```
fpolicy options <PolicyName> reqcancel_timeout [timeout-in-secs]
```

The following is an example of how to view the current `reqcancel_timeout` setting:

```
NetApp> fpolicy options p1 reqcancel_timeout
fpolicy options p1 reqcancel_timeout: 0 secs (disabled)
```

The following is an example of how to set a new value for the `reqcancel_timeout` setting:

```
NetApp> fpolicy options p1 reqcancel_timeout 60
```

Note: A value of 0 means that this feature is disabled. The default value is 0.

6 SUGGESTIONS ON USING THE SAFEGUARDS

In most cases, setting “skip server screening on CIFS requests if the underlying CIFS session is disconnected” to on should resolve the issue (section 5.1).

If this doesn’t resolve the problem, set a limit on the number of CIFS requests held up in FPolicy to a value of around 30 (section 5.2).

The use of the features listed in sections 5.3 and 5.4 can have the following adverse effects, and should therefore be used with caution and only after carefully assessing the impact on the FPolicy server application.

6.1 Side Effects of Setting a Limit on How Long an FPolicy Server Can Be in a “Throttled” State

If this option is used to limit the amount of time that an FPolicy server can be in the throttled state, and the FPolicy server reaches that time limit and is disconnected by the NetApp system, FPolicy services are no longer available (unless there is a backup FPolicy server).

6.2 Side Effects of Setting a Limit on the Length of Time an Individual Request Can Be Held Up in an FPolicy Server

If this option is used to limit the amount of time that an individual request can be held up by an FPolicy server, and the FPolicy server takes longer than this time to process the request, the request is canceled by the NetApp system without being completed.

7 APPENDIX A: CHANGES IN THE USER INTERFACES

7.1 Remote Procedure Call (RPC)

The following RPC has been added to the ntapfprq.idl interface. The FPolicy server should return a 0 for this RPC. The return value is ignored on the NetApp system except for logging purposes.

```
DWORD
FP_RequestCancel(
       [in]         handle_t  FilerHandle,
       [in]         DWORD  FilerID,
       [in]         DWORD  RequestID,
       [in]         DWORD  Reason);
```

7.2 Event Management System (EMS)

The following EMSs have been added:

fpolicy.CIFSlimitExceeded: This is raised when a storage system wide limit on the total number of CIFS requests in FPolicy is exceeded.

fpolicy.fscreen.server.disconnect.noProgress: This is raised when an FPolicy server is disconnected due to lack of progress.

fpolicy.fscreen.cancelReqst: This is raised when a screen request sent to an FPolicy server is canceled because it has not been completed within the specified period of time.

8 APPENDIX B: ADDITION OF NEW FPOLICY COUNTERS

The following counters have been added; they are accessible at the priv set diag level.

Use `stats explain counters` for detailed descriptions of these counters:

fpolicy_global:fpolicy:cifs_requests

fpolicy_global:fpolicy:nfs_requests

fpolicy_stats_server:<server_ip>:secs_since_last_completion

fpolicy_stats_server:<server_ip>:secs_since_throttle