Use Secure Firewall 7.6 Disk Cleaner Tool FMC and FTD

Contents

Introduction

This document describes how to clean files from the system disk.

Minimum Software and Hardware Platforms

Manager(s) and Version (s): FMC/FDM 7.6.0 CLI executed at the FMC or FTD CLI

Application (ASA/FTD) and Minimum Version of Application: FTD 7.6.0

Supported Platforms: All FMC and FTD devices have this feature.

Migrated from: https://confluence-eng-rtp2.cisco.com/conf/display/IFT/Diskcleaner

Feature Description and Walkthrough

Overview

The diskcleaner and its scripts are installed at installation/upgrade. It is not part of the upgrade scripts. The process diskcleaner.py is started by the Process Manager at system startup. It runs as a daemon process managed by the Process Manager and runs until the system is shut down. The disk cleaner has disk scripts that are invoked when the disk volume= usage percentage exceeds configured thresholds. For example, currently there are disk cleaning scripts that are invoked when the volume usage fullness reaches 85% and others that are invoked when the disk usage fullness reaches 95%. If the disk fullness reaches 95%, it runs the 95% fullness cleanup scripts and if the disk fullness remains greater than 85%, it then runs the 85% fullness cleanup scripts.

FMC Diskcleaner

This is what the diskcleaner is currently configured to run on the FMC:

- 85% disk usage cleanup actions:

- Truncate (to 0 bytes) any deleted files (files that have been deleted but are still open by a running process). These files can be shown with the **lsof +L1** command.

- Force the disk manager to drain all silos.

- Delete all rotated log files (those with file extensions xxxxxxxxx.n.gz such as messages.1.gz)

- 95% disk usage cleanup actions:

- Force the disk manager to do a maximum drain of all silos. This drains the silos to 25% of their normal low-water mark.

FTD Diskcleaner

This is what the diskcleaner is currently configured to run on the FTD:

- 85% disk usage cleanup actions:

- Truncate (to 0 bytes) any deleted files (files that have been deleted but are still open by a running process). These files can be shown with the **lsof +L1** ccommand.

- Force the disk manager to drain all silos.

- Delete all rotated log files (those with file extensions xxxxxxxxx.n.gz such as messages.1.gz)

- 95% disk usage actions:

- Force the disk manager to do a maximum drain of all silos. This drains the silos to 25% of their normal low-water mark.

```
NOTE: At the current time, FXOS files and logs are outside of the scope of the diskcleaner!!!
```

View the Available Diskcleaner Scripts

Use the **system support diskcleaner-show** command to see the disk cleaner scripts that are available to run.

The scripts displayed here can be manually run using the **diskcleaner-run** command to manually free up disk space. Each of these disk cleanup scripts are described in these slides with invocation examples.

<#root>

```
> system support diskcleaner-show
```

```
sfims-file-mgmnt-infra-delete-rotated-logs.sh
sfims-file-mgmnt-infra-diskmanager-partition-drain-max.sh
sfims-file-mgmnt-infra-diskmanager-silo-drain-all.sh
sfims-file-mgmnt-infra-truncate-deleted-files.py
```

Manually Running the Diskcleaner

Enter the **diskcleaner** command. The existing diskmanager and disk pruner methods are insufficient for this new task. The diskmanager was designed to control disk fullness levels in real-time as files are opened and closed. The diskmanager configuration files are selective, complicated, and fragile. The pruner process is designed to trim file space usage for individual feature components without awareness of the overall system disk fullness levels. The new diskcleaner framework has been created to provide the the ability to remove files based upon the system disk fullness levels similar to the diskmanager with the simplicity of using a scripting language similar to that of the disk pruner process.

Use the **diskcleaner-run** command to manually run a diskcleaner script. The command is:

<#root>

```
system support diskcleaner-run <file name>
```

where file name is the name of the diskcleaner script to be run. It can be a full path with directory and file name, a relative path (from /etc/sf/dc), or the name of a script (as it is named in /etc/sf/dc). Standard file globs are accepted. The file name is subject to security requirements that can restrict the permissible set of characters (like no use of backticks `). Remember that the diskcleaner is just blindly running scripts. So, you can create a new script that runs several diskcleaner scripts and then just run that new script (so that all of the diskcleaner scripts included therein are executed).

Delete Rotated Logs

Delete all rotated log files on the specified mount point.

<#root>

```
> system support diskcleaner-run sfims-file-mgmnt-infra-delete-rotated-logs.sh --help
```

Delete all rotated log files on the specified mount point.

<#root>

```
sfims-file-mgmnt-infra-delete-rotated-logs.sh [--debug] [--help] <mount point>
```

```
> system support diskcleaner-run sfims-file-mgmnt-infra-delete-rotated- logs.sh --debug /ngfw/Volume
```

```
Deleting all rotated log files on mount point '/ngfw/Volume'.
Deletion of all rotated log files on mount point '/ngfw/Volume' has
completed - 0 bytes.
```

Disk Manager Partition Drain

Make the disk manager perform a maximum drain on all partitions

<#root>

```
> system support diskcleaner-run sfims-file-mgmnt-infra-diskmanager-partition-drain-max.sh --help
```

Make the disk manager to do a maximum drain of all of its partitions.

```
sfims-file-mgmnt-infra-diskmanager-partition-drain-max.sh [--debug] [--help]
```

<#root>

```
> system support diskcleaner-run sfims-file-mgmnt-infra-diskmanager-partition-drain-max.sh
```

```
Performing a maximum drain on all disk manager partitions - current disk usage:
Partition:Silo Used Minimum Maximum
/ngfw/var:Temporary Files 0 KB 121.704 MB 486.817 MB
...
Performing a maximum drain on disk manager partition '/ngfw/var'.
Partition /ngfw/var has been drained.
Maximum drain on all disk manager partitions has completed - current disk usage:
Partition:Silo Used Minimum Maximum
/ngfw/var:Temporary Files 0 KB 121.704 MB 486.817 MB
...
>
```

Disk Manager Silo Drain

Make the disk manager perform a maximum drain on all the silos

<#root>

```
> system support diskcleaner-run sfims-file-mgmnt-infra-diskmanager-silo-drain-all.sh --help
```

Make the disk manager to do a maximum drain of all of its silos.

<#root>

```
sfims-file-mgmnt-infra-diskmanager-silo-drain-all.sh [--debug] [--help]
```

```
> system support diskcleaner-run sfims-file-mgmnt-infra-diskmanager-silo-drain-all.sh
```

```
Draining all disk manager silos to their low-water mark - current disk usage:
Partition:Silo Used Minimum Maximum
/ngfw/var:Temporary Files 0 KB 121.704 MB 486.817 MB
...
Draining all disk manager silos..
All silos have been drained.
Draining all disk manager silos to their low-water mark has completed - current disk usage:
Partition:Silo Used Minimum Maximum
/ngfw/var:Temporary Files 0 KB 121.704 MB 486.817 MB
>
```

Truncate Deleted Files

Truncate to zero bytes all hidden deleted files that are present due to open file handles.

<#root>

```
> system support diskcleaner-run sfims-file-mgmnt-infra-truncate-deleted-files.py -h
```

<#root>

```
usage: sfims-file-mgmnt-infra-truncate-deleted-files.py [-h] [--debug]
[ignored]
Truncate all deleted (zombie) files to zero-length.
positional arguments:
ignored
optional arguments:
-h, --help show this help message and exit
--debug If specified, the script will output lots of debug information.
```

```
> system support diskcleaner-run sfims-file-mgmnt-infra-truncate-deleted-files.py
```

```
Truncating all deleted (zombie) files.
Deleted file '/run/nscd/dbGG9F8K' is on the deleted files exclude list - NOT truncating.
All deleted (zombie) files have been truncated - total size = 0 bytes.
>
```

Stopping and Starting Diskcleaner

Use the **pmtool** command to stop and start the diskcleaner.

<#root>

```
> pmtool disablebyid diskcleaner

>

> pmtool status

...
diskcleaner (normal) - User Disabled
...

> pmtool enablebyid diskcleaner

>

> pmtool status

...
diskcleaner (normal) – Running 17086
...
```

Software Technology

Diskcleaner.py

This script is the main program for the diskcleaner. This python script is run like this:

```
diskcleaner.py [--debug] [--help] [--interval <interval>]
```

where if debugging is specified, the script outputs debug information, and help is the output usage information and exit.

The interval (in seconds) between disk cleaning cycles. If not specified, the default is 600 seconds. diskcleaner.py is started by the Process Manager at system startup. It runs as a daemon process managed by the Process Manager and runs until the system is shutdown. When the diskcleaner is started, it runs a loop that runs every **interval** seconds. Every cycle, the diskcleaner scans all the mount points (/, /Volume, ...) specified by the **df -a** command and get their current disk usage (as measured by the **df** command).

The diskcleaner walks through each cleaning level starting at level 0 and work its way up until the level number is greater than the disk usage percentage of that mount point. You only need to do cleaning for a level if the usage of the mount point is at or greater than the level number. So, if the usage is 55%, execute each disk cleaning level in order – 0, 1, 2, …, 55.

When cleaning needs to be done, the shell script diskcleaner.sh (see below) is invoked to do the actual cleaning. That script just blindly calls every script found in the diskcleaner directory of the mount point for the specified clean level.

- As an example, let's say the / mount point is currently at 96% usage. When the diskcleaner starts its cycle, it runs the **df** command and find out that the / mount point is at 96% usage. The usage level for level 0 cleaning is set for 0%, so the diskcleaner runs all the level 0 scripts specified for the / mount point. At this point, the diskcleaner reruns the **df** command. Assume that the level 0 cleaning did nothing, so the usage is still 96%. The diskcleaner now runs all the level 1 scripts, then level 2 and so forth.

- Let's say that there are scripts at level 50 that reduce the / mount point usage to 90%. The diskcleaner continues with level 51, 52, ….

- Let's say there are scripts at level 80 that reduce the / mount point usage to 75%. Now the diskcleaner is finished with the / mount point because 75% is less than the next level which is 81.

Diskcleaner.sh

The diskcleaner.sh is a bash script that initiates the disk cleaning event.

```
diskcleaner.sh <mount point> <level>
```

where the mount point is the mount point to be cleaned and the level refers to the cleaning level to be performed. This is the percentage at which the cleaning is done. For example, if 85 were entered, it would run diskcleaner at 85%.

All the disk cleaning scripts present in the directory:

```
[/ngfw]/etc/sf/dc/<mount point>/dc<clean level>
```

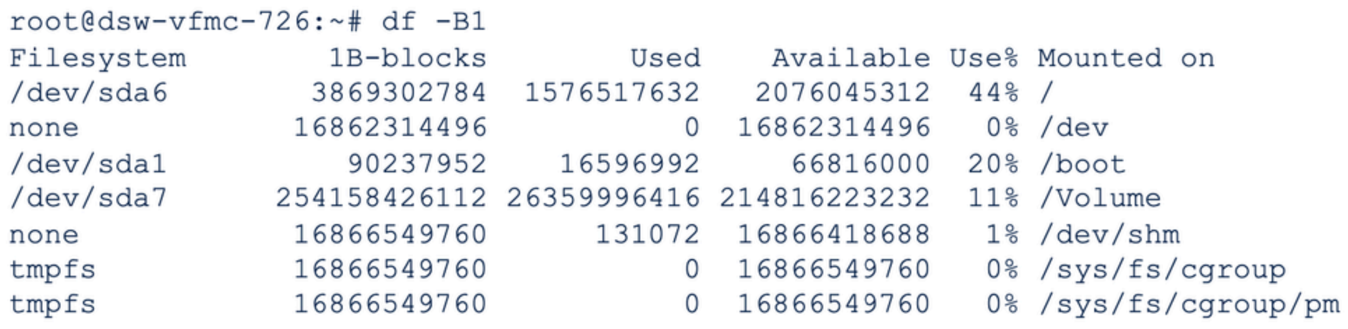
that match the file glob DC* are executed (in alphabetical order).

Note that because the mount point names usually have slashes (/) in them, using the actual mount point name in the directory path is awkward (or illegal). So, change any / to **_.** So, to clean the mount point /dev/shm at clean level 85 (for 85% disk usage), the scripts must match: [/ngfw]/etc/sf/dc/_dev_shm/85/DC*

This paradigm is based off the **rc init** script paradigm whereby all matching scripts in the rcn.d directory are blindly executed. The DC prefix is also a holdover from the rc init paradigm – those scripts all start with either **S** (start) or **K** (kill). I guess it keeps you from picking up crap files that are just left in the directory. So, use a prefix of DC.

Under the Covers

The diskcleaner uses the linux **df** command to check for disk usage:



```
root@dsw-vfmc-726:~# df -B1
Filesystem          1B-blocks        Used    Available Use% Mounted on
/dev/sda6          3869302784  1576517632   2076045312  44% /
none              16862314496           0  16862314496   0% /dev
/dev/sda1            90237952    16596992     66816000  20% /boot
/dev/sda7        254158426112 26359996416 214816223232  11% /Volume
none              16866549760      131072  16866418688   1% /dev/shm
tmpfs             16866549760           0  16866549760   0% /sys/fs/cgroup
tmpfs             16866549760           0  16866549760   0% /sys/fs/cgroup/pm
```

Troubleshooting Diskcleaner

System Files

- Troubleshooting of the diskcleaner can be done by looking at:

<#root>

`/var/log/process_stdout.log and /var/log/process_stderr.log`

- Entries logged by the diskcleaner contain **diskcleaner** as the process name and the diskcleaner PID.
- This is the standard way of looking at logging for a PM-based process.

Steps to Troubleshoot

What the diskcleaner does is logged to process_stdout.log. Usually, there is no output because there is no disk cleaning to be done. But if disk cleaning needs to be done, whatever files are deleted are logged in process_stdout.log. On the next page is an example of what process_stdout.log can contain. **Grep** on diskcleaner.ˇSample Output in process_stdout.log:

```
Feb 9 04:10:23 uhura diskcleaner[2639]: Starting level 85 disk cleaning for mount point '/ngfw/Volume':
Feb 9 04:10:23 uhura diskcleaner[2639]: Starting disk cleaning scripts in directory '/etc/sf/dc/_ngfw_V
Feb 9 04:10:23 uhura diskcleaner[2639]: Filesystem 1K-blocks Used Available Use% Mounted on
Feb 9 04:10:23 uhura diskcleaner[2639]: /dev/sda5 40511148 39040764 1470384 97% /ngfw/Volume
Feb 9 04:10:23 uhura diskcleaner[2639]: Executing '/etc/sf/dc/_ngfw_Volume/dc85/DC660-truncate-deleted-
Feb 9 04:10:23 uhura diskcleaner[2639]: Truncating all deleted (zombie) files.
Feb 9 04:10:23 uhura diskcleaner[2639]: Deleted file '/run/nscd/dbZ5hobS' is on the deleted files exclu
Feb 9 04:10:23 uhura diskcleaner[2639]: All deleted (zombie) files have been truncated - total size = 0
Feb 9 04:10:23 uhura diskcleaner[2639]: Filesystem 1K-blocks Used Available Use% Mounted on
Feb 9 04:10:23 uhura diskcleaner[2639]: /dev/sda5 40511148 39040764 1470384 97% /ngfw/Volume
Feb 9 04:10:23 uhura diskcleaner[2639]: '/etc/sf/dc/_ngfw_Volume/dc85/DC660-truncate-deleted-files.py'
Feb 9 04:10:23 uhura diskcleaner[2639]: Executing '/etc/sf/dc/_ngfw_Volume/dc85/DC663-delete-rotated- l
Feb 9 04:10:23 uhura diskcleaner[2639]: Deleting all rotated log files on mount point '/ngfw/Volume'.
Feb 9 04:10:24 uhura diskcleaner[2639]: Deleted file '/ngfw/Volume/root1/ngfw/var/log/top.log.1.gz' - 2
Feb 9 04:10:24 uhura diskcleaner[2639]: Deleted file '/ngfw/Volume/root1/ngfw/var/log/test/fake-rotated
Feb 9 04:10:24 uhura diskcleaner[2639]: Deleted file '/ngfw/Volume/root1/ngfw/var/log/test/fake-rotated
Feb 9 04:10:24 uhura diskcleaner[2639]: Deleted file '/ngfw/Volume/root1/ngfw/var/log/test/fake-rotated
Feb 9 04:10:24 uhura diskcleaner[2639]: Deletion of all rotated log files on mount point '/ngfw/Volume'
Feb 9 04:10:24 uhura diskcleaner[2639]: Filesystem 1K-blocks Used Available Use% Mounted on
Feb 9 04:10:24 uhura diskcleaner[2639]: /dev/sda5 40511148 13872228 26638920 35% /ngfw/Volume
Feb 9 04:10:24 uhura diskcleaner[2639]: '/etc/sf/dc/_ngfw_Volume/dc85/DC663-delete-rotated-logs.sh' com
Feb 9 04:10:24 uhura diskcleaner[2639]: Completed disk cleaning scripts in directory '/etc/sf/dc/_ngfw_
Feb 9 04:10:24 uhura diskcleaner[2639]: Finished level 85 disk cleaning for mount point '/ngfw/Volume':
```