

Application Note - Management of Data Files in the EpiSensor Gateway



Document Ref: EPI-191-00

Date

25 November 2020

Background

A scenario arose on two NGR-30-5 gateways where the gateway filled up the disk space with data files. In one case, in our office environment, the files were queued for export. In the other case, on a customer site, the files were archived locally.

The problem occurred when an attempt was made to install a newer version of the epi-gateway snap. As part of upgrading a snap to a new revision, the \$SNAP_DATA area is copied over to the newer version, which magnified the problem. The disk space filled during the upgrade and as a result the snapd process was irrevocably damaged. The 'snap install' command for epi-gateway times out, with an error message indicating a problem with connecting to the server. The main writable disk area is full as reported by the linux disk free command:

```
df -Th
dev/mmcblk0p4 ext4      25G   25G    0 100% /writable
```

Deleting the files from \$SNAP_DATA and freeing up disk space does not solve the problem. The only solution is to completely re-image the gateway.

Proposed Solution

- **Storage Location**

- This is only relevant for the NGR-30-5 platform where the epi-gateway snap runs under the snapd architecture.
- Currently files queued for export and the data archive files are stored in \$SNAP_DATA. Here is a description of \$SNAP_DATA from snapcraft.io documentation:

```
$SNAP_DATA
```

```
Directory for system data of a snap.
```

```
This directory is owned and writable by root and is meant to be used by background applications (daemons, services). Unlike SNAP_COMMON this directory is backed up and restored across snap refresh and snap revert operations.
```

Application Note - Management of Data Files in the EpiSensor Gateway



Document Ref: EPI-191-00

Typical value `/var/snap/hello-world/27`

- Propose to instead store files queued for export in `$$SNAP_COMMON`. Here is a description of `$$SNAP_COMMON` from [snapcraft.io](https://snapcraft.io/documentation) documentation:

```
$$SNAP_COMMON
```

```
Directory for system data that is common across revisions of a snap.
```

```
This directory is owned and writable by root and is meant to be used by background applications (daemons, services). Unlike SNAP_DATA this directory is not backed up and restored across snap refresh and revert operations.
```

```
Typical value: /var/snap/hello-world/common
```

- This proposal will mean that the files archived or queued for export are not duplicated when the epi-gateway snap revision is updated.
 - One other issue to consider is that we will need to cater for archived or queued files in `$$SNAP_DATA` during the upgrade to the new version of epi-gateway which contains this fix.
 - Possibly also store gateway log files in `$$SNAP_COMMON` instead of `$$SNAP_DATA` but currently this is not causing a problem as there is a limit of the amount of log files generated. However, log files do not need to be duplicated with revisions of the snap either.
- **Monitoring**
 - Propose to monitor the disk space usage and inode usage periodically with the same frequency as the data export interval.
 - Use the `df` command to check remaining disk space and inode usage
 - The percentage of disk space and inode used will be the indicator that a problem is imminent.

Application Note - Management of Data Files in the EpiSensor Gateway



Document Ref: EPI-191-00

• Corrective Action

- Corrective action will be taken if the disk space or inodes used exceed a threshold. This threshold will be configurable on the Engineering Settings page of the Gateway From End and via the API.
- The default threshold will be 90%. The maximum allowed threshold will be 95% and the minimum will be 20%.
- If the threshold is reached then there are two proposed corrective actions:
 - Option A: Stop writing to disk, that is stop writing data files. In practice, this means the Data Server will stop taking DATB messages from the sensor data queue which will then eventually fill up.
 - Option B: Start deleting files, starting with the oldest
- I propose going with Option A, because this means the user will have to take some corrective action to clear the backlog. With Option B, the system continues with this conveyor belt system of file creation and deletion forever.
- In the case of Option A, the user should delete files using the web interface front end or the API, both of which support deleting files queued for export and archived locally.
- The gateway status via the API and on the homepage of the front end will indicate “File System full”, similar to “ZAP not connected”.

• Deployment

- **Due to the fact that we have only seen this problem manifest itself so far during an upgrade of epi-gateway, we need to be careful of further updates to the epi-gateway snap. It is highly recommended to check the number of files queued for export and the number of data files archived before attempting the upgrade. (Alternatively the disk usage can also be checked).**
- It could also be possible that this scenario will occur without being triggered by an upgrade of epi-gateway snap. We need to determine what happens to snapd if the disk fills during normal execution of the snap. This could be investigated further in our office test set-up.
- Consider also the development of a separate utility which can check the vital statistics of the gateway prior to an installation/upgrade. This utility could also be used by customers performing the upgrade themselves.
- The epi-gateway snap will take care to move existing archived or queued files from \$SNAP_DATA to \$SNAP_COMMON. However it would be a good idea to remove queued

Application Note - Management of Data Files in the EpiSensor Gateway



Document Ref: EPI-191-00

and archived files from older revisions on the epi-gateway snap. By default the current revision and 2 older revisions are retained on the NGR-30-5. These files could be deleted manually in `/var/snap/epi-gateway/x??` where `??` is not the current revision. You would need to be a `root` user to achieve this.