SAVEAUDI - Saves SAFEGUARD Audit files
Version 106
27Feb2018

SAVEEMS - Saves EMS log files
Version 100
28Nov2007

When SAFEGUARD is running, it produces audit information, which is stored in files.
The number of files, as well as their location, can be configured.

The usage of the files is as follows:

1. In case all files run full, the oldest file is purged and a new one is created.
   This is a meaningful feature.

2. In case all files run full, the auditing is terminated, but files still can be accessed.
   This does not make much sense.

3. In case all files run full, the auditing is terminated, and all file access is denies.
   This is very secure, but in terms of NonStop, not optimal, and can be used as a Denial of Service attack.

Method 1 is the commonly used, and recommended one.

I urge my customers to have the audit files at least that big, that all audit events of one day/week fit into one file. And depending on the backup frequency, I ask them to have as many files available as they would like to have online for audit reports. This is e.g. for one month, six weeks, what ever.

But there are also customers, not willing to keep track of the audit files through backups, but would like to save them on disk when they run full.

e.g.
- The Audit pool has 2 files defined.
- The actual file names are: A0002345 and A0002346.
- File A0002345 is already full and file A0002346 runs full.
- SAFEGURD purges file A0002345 and creates the new file A0002347.

Up to here everything is normal.
- File A0002346 is duped to a given location.

This is the requested functionality.

SAFEGUARD does not have this feature.

The new FreeWare tool from GreenHouse: SaveAudi does the trick for you. It uses SPI to talk to SAFEGUARD to get the current audit file name, and continues checking for a file change on a configurable regular basis. In case it detects a file change, the just filled audit file is duplicated to a given location.

Two saving methods are supported:
1. The file to save is duplicated to the defined location.

2. The file to save is PAKed and stored in the defined location.

To make this solution as convenient as possible, SAFEAUDI should be defined as a persistent process, controlled by the $ZZKRN process. The TACL Macro named ZZKRN is delivered as an example, how to introduce SAVEAUFI as a persistent process to the system.

SAVEAUDI has to be started from SUPER.SUPER!

Command syntax:

```
  SAVEAUDI [<destination>] [<cycle-time>] [PAKANDSAVE] [PAKPRI nn]
```

where

```
  <destination>   is a subvol, defining the location to which the
                  SAFEGUARD audit files have to be duplicated.
                  In case the destination location is not on the local
                  system, SUPER.SUPER needs to have remote passwords to
                  the remote system.
             When missing, the location of the objectfile of
             SAVEAUDI is used.

  <cycle-time>    defines the cycle time in minutes to be used to check
                  for an audit file switch.
                  Default is 60 minutes.

  PAKANDSAVE       when present causes SAVEAUDI to use PAK to compress,
             and duplicate the file in question.
             When missing, the file is duplicated as it is.

  PAKPRI nn       defines the priority to be used by PAK.
                  Default is 50.
```

SAVEAUDI has to be started from SUPER.SUPER. In case it is NOT - and this is checked in the code - SAVEAUDI does NOT run.

-.--.-..-.--.-..-.--.-..-.--.-..

The EMS system generates log files as well.
On demand by a Tandem user I created a tool that saves the EMS log files.

SAVEEMS has to be started from SUPER.SUPER!

Command syntax:

```
  SAVEEMS <destination> [<cycle-time>]
```

where

destination — is a subvol, defining the location to which the SAFEGUARD audit files have to be duplicated.
In case the destination location is not on the local system, SUPER.SUPER needs to have remote passwords to the remote system.

cycle-time — defines the cycle time in minutes to be used to check for an audit file switch.
Default is 60 minutes.

SAVEEMS has to be started from SUPER.SUPER. In case it is NOT - and this is checked in the code - SAVEEMS does NOT run.

Both products report all activities to the EMS system.
The following sequence of messages is issued when the products are started:

```
07-11-28 13:43:03 \GINKGO.$ZPM      TANDEM.DSC.G06        002048 $SSFG
started
                                    by $ZPM in cpu 0
07-11-28 13:43:03 \GINKGO.$SSFG     GHS.28.103            004748
SAVEAUDI: Save
                                    location: $GHS1.SAVEAUDI
07-11-28 13:43:03 \GINKGO.$SSFG     GHS.28.103            004748
SAVEAUDI: Cycle


                                    time: 1 minute
07-11-28 13:43:03 \GINKGO.$SSFG     GHS.28.103            004748
SAVEAUDI:
                                    Current SAFEGUARD audit file:
                                    $SYSTEM.SAFE.A0001537
07-11-28 13:43:03 \GINKGO.$ZPM      TANDEM.DSC.G06        002048 $SEMS
started
                                    by $ZPM in cpu 0
07-11-28 13:43:03 \GINKGO.$SEMS     GHS.29.100            004748 SAVEEMS:
Save
                                    location: $GHS1.SAVEEMS
07-11-28 13:43:03 \GINKGO.$SEMS     GHS.29.100            004748 SAVEEMS:
Cycle
                                    time: 1 minute
07-11-28 13:43:03 \GINKGO.$SEMS     GHS.29.100            004748 SAVEEMS:
                                    Current EMS log file:
                                    \GINKGO.$SYSTEM.ZLOG01.ZZEV0002
```

27Feb2018
Carl Weber