

SFACL

Version 200

30. July 2000


GreenHouse

Software & Consulting

Karl-Heinz Weber

Heinrichstraße 12

D-45711 Datteln/Horneburg

SAFEGUARD does not allow duplicate ACL entries, but allows 'super fluous' ACLs and access rights.

Assume the following ACL for a disk file:

```
=info diskfile save
$GHS1.CLEANACL
SAVE          15FEB99,  9:55          100,5          THAWED
  100,005      R,W,E,P,C,O
  100,006 DENY      E
  100,255      R,W,E,P,  O
  100,*        R
  100,* DENY      W,E,P,  O
\*.*,*        R,W,E,P,  O
=
```

When you have a closer look, then you easily find out, that there are duplicate entries. In the following list, super fluous ACLs are marked with an asterisk, and super fluous access rights are shown in lower case letters:

```
  100,005      r,w,e,p,C,o
  100,006 deny      e          *
  100,255      r,w,e,p,    o *
  100,*        r          *
  100,* DENY      W,E,P,  O
\*.*,*        R,W,E,P,  O
```

Wouldn't it be nice to have a tool, that automatically adjusts the ACLs and access right entries to the minimum of necessary entries, e.g. to the following list:

```
=info diskfile save
$GHS1.CLEANACL
SAVE          15FEB99, 12:10          100,5          THAWED

  100,005      C
  100,* DENY      W,E,P,  O
\*.*,*        R,W,E,P,  O
=
```

The freeware tool SFACL (Super fluous ACL) from *GreenHouse* does this.

SFACL is a TAL program, which has an interface to SAFEGUARD/SAFECON to get all SAFEGUARD settings, and to perform necessary changes.

SFACL can be used to:

- Display all super fluous ACLs and access entries, and optionally show the intermediate clean-up steps (EXPAMIN option)
- Display AND clean-up super fluous ACLs and access entries.

Command syntax:

```
[run] SFACL [/OUT [<file>]/] <type> <item> [,CLEANUP] [,EXPLAIN]
```

where

file	OUT device to which the evaluation and access results have to be reported. In case the file does not exists, it is creates a an EDIT type file. In case the file exists, the output is appended to the EOF.
type	Defines the object type of interest. Is one of: <ul style="list-style-type: none">• DISKFILE• SUBVOLUME• VOLUME• PROCESS• SUBPROCESS• DEVICE• SUBDEVICE• OBJECTTYPE Types can be abbreviated according to the SAFEGUARD supported rules.
item	Name of item, e.g. a file name; Depending on <type>, wild cards are supported.
CLEANUP	When present directs SFACL to clean up all super fluous ACLs and access entries.
EXPLAIN	When present directs SFACL to displays the intermediate clean-up results. Does work only for fully qualified file names (no wildcard support).

Running SFACL without any parameter, or with an unexpected one, causes SFACL to display the following help screen:

Required parameter(s) missing

Command syntax is:

```
[run] SFACL [/OUT[<file>]/] <object-type> <template> [,CLEANUP][,EXPLAIN]
```

where

<file>	when present defines the file to which the evaluation results and actions are reported.
<object-type>	defines the SAFEGUARD object of interest. Valid object types are: VOL[UME] SUBVOL[UME] [DISK]FILE PROC[ESS] SUBPROC[ESS] DEV[ICE] SUBDEV[ICE] OBJECT[TYPE]
<template>	is an object name template; wild cards are supported when object-type does.
CLEANUP	causes SFACL to purge all superfluous ACL and access entries.
EXPLAIN	displays the intermediate results of the clean-up steps.

e.g.

```
SFACL DISKFILE $GHS1.*.*
```

A typical clean-up session looks like this:

```
$GHS1 CLEANACL 5> object diskfile save,CLEANUP
SFACL (100) - T9999G03 - (11Feb1999) System \BEECH, running NSK G02
Copyright (c) GreenHouse Software & Consulting 1999
>>> Collecting all ACLs
>>> Relevant ACLs

$GHS1.CLEANACL.SAVE          100,005      r,w,e,p,C,o
                             100,006 DENY      e      *
                             100,255      r,w,e,p,  o  *
                             100,*        r          *
                             100,*        DENY    W,E,P,  O
                             \*.*,*      R,W,E,P,  O

>>> Adjusting super fluous ACLs

ALTER DISKFILE $GHS1.CLEANACL.SAVE ,ACCESS 100,005 - (R,W,E,P,O)
ALTER DISKFILE $GHS1.CLEANACL.SAVE ,ACCESS 100,006 - DENY (E)
ALTER DISKFILE $GHS1.CLEANACL.SAVE ,ACCESS 100,255 - (R,W,E,P,O)
ALTER DISKFILE $GHS1.CLEANACL.SAVE ,ACCESS 100,* - (R)
Done
$GHS1 CLEANACL 6>
```

The output can be directed to an OUT file, e.g. a SPOOLER location. In case OUT does not exist, it becomes created as EDIT type file.

To list the super fluous ACLs and access entries of all disk files, the command is:

```
SFACL DISKFILE $*.*.*
```

To clean-up all super fluous ACLs and access entries of all processes, the command is:

```
SFACL PROCESS $*,CLEANUP
```

In case the object types:

- DISKFILE
- SUBVOL
- PROCESS
- SUBPROCESS
- DEVICE
- SUBDEVICE

are set to CHECK OFF, all access entries of these types are marked super fluous.

```
SFACL (101) - T9999G03 - (16Feb1999)   System \BEECH, running NSK G02
Copyright (c) GreenHouse Software & Consulting 1999
>>> Collecting all ACLs
>>> Relevant ACLs

$GHS1.SFACL.SAVE                        100,005      r,w,e,p,c,o   Check Off
                                         100,006 DENY  r,w,  p      Check Off
                                         100,255      r,w,e,p,  o * Check Off
                                         200,005      r              * Check Off
                                         200,005 DENY      e          Check Off
                                         100,*        r              * Check Off
                                         *,*          r,w,e,p,  o * Check Off
\*.*,*                                r,w,e,p,  o   Check Off

Done
$GHS1 SFACL 39>
```

In case the object types

- VOLUME

is set to CHECK OFF, all access entries, EXCEPT Create (C), are marked super fluous:

```
$GHS1 SFACL 33> sfacl volume $ghs1
SFACL (101) - T9999G03 - (16Feb1999)   System \BEECH, running NSK G02
Copyright (c) GreenHouse Software & Consulting 1999
>>> Collecting all ACLs
>>> Relevant ACLs

$GHS1                                100,005      r,w,e,p,c,o * Check Off
                                         100,*        r,w,e,p,C,o   Check Off

Done
$GHS1 SFACL 34>
```

ACL Evaluation Logic

SFACL uses the following steps to find, and delete super fluous ACL entries:

1. In a first step, all ACLs, matching the given type and item are read from the system.
2. In the second step all ACL entries are processed in the following order:
 - Access rights are marked super fluous when there is a matching DENY entry.
 - Access rights are marked super fluous when there is a 'higher' matching access right.
 - DENY access rights are checked for 'higher' DENY access rights.
 - We finally check for 'lonely' DENY entries.
 - In case we evaluate a VOLUME ACL, we take the optional CHECK VOLUME OFF setting into account
 - else
 - In case we evaluate a SUBVOL, DISKFILE, PROCESS, SUBPROCESS, DEVICE or SUBDEVICE, we take the optional CHECK <item> OFF setting into account.
3. In the final clean-up step, the found super fluous ACL entries are passed to SAFECOM in an ALTER command, causing a deletion of them.

Please report any error and/or request for enhancements to:

GreenHouse Software & Consulting

Karl-Heinz Weber

Heinrichstraße 12

D-45711 Datteln/Horneburg

Germany

Phone +49 (0)2363 72566

Fax +49 (0)2363 66106

AOL Fax +49 (0)40 3603 044706

Mobile +49 (0)172 23 18248

E-Mail: Info@GreenHouse.de

Home page: <http://www.GreenHouse.de>

PGP fingerprint: 3A 32 D9 0A D1 25 54 18
11 50 24 84 66 29 2D D2

