SECURE

Version 303

Users Manual

05. June 2007

GreenHouse
Software & Consulting
Heinrichstrasse 12
D-45711 Datteln/Horneburg

Trademarks or Service Marks

The following are trademarks or service marks of Tandem Computers Incorporated:
Atalla, Challenge/Response, Enform, Expand, Guardian, Guardian90, Inspect, Multilan, NonStop, TACL, Tandem.
All brand names and product names are trademarks or registered trademarks of their respective companies.

The following are trademarks or service marks of *GreenHouse Software & Consulting*:
$ARROW, $AS, CRYSTAL, CURIOUS, FTPSERV-E, FUNCTRAC, MPWD, MPWD-L, PASSYNC, SECMAN, SECOM, SECOM-L, GSTK, SSTK.

The following are trademarks or service marks of Jelinek EDV:
SECMAIN

Copyright

Please Comment

If you have questions or problems concerning the content of this document, please let me know! Send your comments to:

GreenHouse Software & Consulting
Karl-Heinz Weber
Heinrichstraße 12
D-45711 Datteln/Horneburg
Germany
Phone +49 (0)2363 72566
Fax +49 (0)2363 66106
Mobile +49 (0)172 23 18248
E-Mail: Info@GreenHouse.de
Internet: www.GreenHouse.de
PGP fingerprint: 3A32 D90A D125 5418 1150 2484 6629 2DD2

Introduction

SECURE is a new tool, allowing to set the GUARDIAN security vectors based on

- wildcarding of the file-set to secure
- wildcarding of the security string to set
- file attributes of the files to secure
  = file OWNER
  = file CODE
  = file SECURITY
  = file TYPE

Beside all this, it also supports two hard coded security settings:

- NETDUP
- MYSEC

NETDUP sets the target security string to **"N***"** for all objects, and to **"N*-*"** for all other files. This secures the file in file-set ready to NETDUP[1]!

MYSEC sets the target security string to **"UUOO"** (= my favorite setting).

The security string can be provided with quotation marks (old format), e.g. "OONO", or without the the quotation makrs, e.g. OONO.
The security characters are NOT case sensitive!

A SHOW command is implemented. When given causes SECURE to display all activities as well as some statistics at the session end.

SECURE is optimized: In case the new security setting is equal to the current one, SECURE does NOT re-secure the file.

SECURE comes in two flavors:
1. FreeWare
2. PayWare

Freeware is free software, which can be used for free, without the need to pay for it.
PayWare is exactly the sysme software as FreeWare, but you need a license and have to pay for it when you use it.
Please refer to the PayWare documentation which can be found in the PayWare folder.

[1] NETDUP is a Tandem internal tool to duplicate files accross the network without having remote passwords

Examples

Here are some example where the mentioned features make sense:

1. To secure all OBJECTS in $SYSTEM.SYSTEM and $SYSTEM.SYSnn for LOCAL EXECUTE access only, normally you need some high sophisticated logic (called OBEY files). SECURE does it with the following command:

```
SECURE $SYSTEM.SYS*.* "**L*" WHERE CODE = 100  or
SECURE $SYSTEM.SYS*.*  **L*  WHERE CODE = 100
```

2. To secure all Key sequenced files, owned by SA.CARL on all volumes to "OOU-", run SECURE:

```
SECURE $*.*.* "OOU-" WHERE OWNER = SA.CARL
```

3. To secure all EDIT type files on $DATA?.MPWD?.MPWD*, owned by SA.CARL, having an actual security attribute of "NN-U", to "NN-O", do it with:

```
SECURE $DATA?.MPWD?.MPWD* "NN-O" WHERE OWNER = SA.CARL&
AND SECURITY = "NN-U" AND CODE = 101
```

Installation

SECURE comes along with the following files:

```
README          101    a brief description
SECURE          100    the SECURE program
SAVESEC         100    Tool to save security settings
LISTSEC         100    Tool to list saved security settings
RESEC           100    Tool to re-set saved security settings
```

SAVESEC, LISTSEC as well as RESEC come with file code 700 und 800 as well.

I recommend to put SECURE and the three supportive utilities in a search path to allow an easy access.

Give SECURE and the three tools to the System Adminmistrator, and set the security vector to: "OOAO"

SECURE nor any of the tools does require any privilege to run.

Command syntax

SECURE fills a gap: It allows to secure a set of files, described by a wildcard, depending of their ownership, type, code and actual security setting. The main idea was born when I had the challenge to secure all the objects in $SYSTEM.SYS*.* for local EXECUTE access only. Without this small program, this really is a challenge.

The command syntax is:

```
SECURE [<file-set> <security-target> [filler][key-word[=]key-item]...
                                    ...[filler][key-word[=]key-item]]
```

where

<file-set>
is a set of files, described by a wildcard.
wildcard characters are asterisks (*) and question marks (?)
e.g. **$DATA.CARL.*** -> all files on $DATA, in subvol CARL
$SYSTEM.SYS*.* -> all files in $SYSTEM in all subvols, beginning with SYS (SYSTEM, SYS01 etc.)

Beginning with 09Mar2004, the SECURE FreeWare tool supports the extendet wildcard character set:

```
* or {*}  = any number of any type of characters
? or {?}  = ONE character of any type
{..a}     = any number of alphabetical characters
{..n}     = any number of numerical    characters
{a}       = ONE alphabetical characters
{n}       = ONE numerical    characters
```

<security-target>
The security target describes the new security string to be given to the files, matching the <file-set>. Currently SECURE supports three targets:

"rwep"-template
describes the new security string to be given to the files in <file-set>.
Valid characters for Read, Write, Execute and Purge are:
* -> don't change the current setting
L -> make the current setting a Local one
R/E -> make the current setting an Remote (=Expand) one
A,G,O,N,C,U,- -> change the current setting to this one
The quotation marks, surrounding the security template, are optional.

NETDUP
In case NETDUP is specified as new security setting, SECURE assumes the following templates:
"N*"** for all OBJECT type files (file code 100)
"N*-*" for all other files

MYSEC

In case MYSEC is specified as new security setting, OR there is NO setting specified, SECURE assumes the following template:
"UUOO" (that is Carl's favorite template setting)"

filler

SECURE supports three fillers, making the command string readable:

- **WHERE**
- **AND**
- **=** (equal sign; does NOT need to be surrounded by blanks!)

These fillers are ignored by SECURE.
e.g.

```
SECURE * MYSEC WHERE CODE = 100    is equal to
SECURE * MYSEC WHERE CODE=100      is equal to
SECURE * MYSEC CODE 100 OWNER GHS.*
```

key-word

SECURE supports five key-words:

- **CODE**
- **OWNER**
- **SECURITY**
- **TYPE**
- **SHOW**

describing file attributes, or directing SECURE to display action counters. The sequence of the key words is up to the user.

CODE [=] file-code

describes the CODE of a disk file that has to match in addition to the file-set as well as all the other parameters. File-code is in the range of 0 to 65535. In case CODE is supported, it is used as an AND condition: There must be a file, matching the file-set AND that file must have the given CODE.

Default: NO CODE check

OWNER [=] file-owner

describes the OWNER of a disk file that has to match in addition to the file-set as well as all the other parameters. SECURE supports both forms of file-owners:
= the ID form, e.g. 100,5
the ID format does NOT support wild cards
= the name form, e.g. SA.CARL
the name format does support wild cards, e.g. sa.c* or ghs.u? or *.manager
In case OWNER is supported, it is used as an AND condition: There must be a file, matching the file-set AND that file must have the mentioned OWNER.

Default: NO OWNER check

SECURITY [=] file-security

describes the file SECURITY that has to match in addition to the file-set as well as all the other parameters. The file-security is in the well known "RWEP" form.
Beside the RWEP security attributes, the asterisk (*) is allowed to mask a security settings. E.g. the string "***C" addresses only those security settings, where the PURGE flag is set to "C".
In case SECURITY is supported, it is used as an AND condition: There must be a file, matching the file-set AND that file must have the mentioned SECURITY.

Default: NO SECURITY check

TYPE [=] file-type

describes the file TYPE that has to match in addition to the file-set as well as all the other parameters. We know about four file TYPEs:

= U Unstructured
= R Relative
= E Entry Sequenced
= K Key Sequenced

In case TYPE is supported, it is used as an AND condition: There must be a file, matching the file-set AND that file must have the mentioned TYPE.

Default: NO TYPE check

SHOW

The SHOW command asks SECURE, to display the activity counters:

= Number of checked files
= Number of changed files
= Number of files in error

Default: NO SHOW

case sensitivity

The command string is NOT case sensitive

delimiters

Delimiters in the command string are either spaces or commas. e.g.

```
SECURE,*,"NNNN",WHERE,CODE,=,100  is equal to
SECURE *,"NNNN", WHERE CODE = 100 is equal to
SECURE *,MYSEC CODE,100           etc.
```

help

In case SECURE is started without any parameter, it displays a brief help text:

```
$GHS1 SECURE 50> secure
SECURE (303) - T7172H06 - (05Jun2007)
Copyright (c) GreenHouse Software & Consulting  1992 .. 2007
Syntax is:
SECURE [<file-set> <security-target> [filler][key-word[=]key-item]
                                  ...[filler][key-word[=]key-item]]
 <file-set>        = list of files to secure; extended wildcard support
<security-target> = new security settings
                    "rwep" format
                     where rwep can be *  ->  don't change
                                       L  ->  translate to local
                                     E,R  ->  translate to remote
                           A,G,O,N,C,U,-  ->  change to
                 OR  NETDUP -> "N***" for OBJECTS AND
                               "N*-*" for all other files
                 OR  MYSEC  -> "UUOO"
 filler            = the words WHERE and AND
 key-word/key-item = CLP (clear-on-purge
                     CODE <file-code> (0 to 65535)
                     OWNER <file-owner> in ID or name format
                     SECURITY <file-security> in "rwep" format
                     TYPE <file-type> (U,R,E or K)
                     SHOW
e.g.:  SECURE $SYSTEM.SYS*.* "**L*" WHERE CODE = 100
secures ALL OBJECTS in $SYSTEM.SYSTEM and SYSnn for local EXECUTE access
only by translating the current EXECUTE setting to the equivalent local
setting: N is translated to A, C to G, and U to O.
The security string supports extended wildcards, and does not require
surrounding quotation marks.

$GHS1 SECURE 51>
```

Examples

SECURE *
secures all files in the current subvol to "UUOO"

SECURE *,MYSEC
secures all files in the current subvol to "UUOO" (that is Carl's favorite security setting)

SECURE *,MYSEC,WHERE CODE = 101
secures all EDIT type files in the current subvol to "UUOO"

SECURE * NETDUP
sets the security target template to **"N***"** for all OBJECTS, and to **"N*-*"** for all other file types

SECURE $SYSTEM.SYS*.* **L* WHERE CODE = 100
translates the EXECUTE settings to the equivalent LOCAL EXECUTE ACCESS

SECURE $DATA?.CARL.*SRC OO-- WHERE CODE = 4711 AND TYPE = K
secures all key sequenced files in the mentioned directories with a file code of 4711 to "OO--"

SECURE $SYSTEM.SYS*.* ***O WHERE OWNER = super.* AND SECURITY = "*N"**
Sets the PURGE flag of all files in the subvols $SYSTEM.SYS* to "O" where the owner belongs to the SUPER-Group and the current PURGE flag is set to "N".

Supportive utilities

The following three supportive utilities come along with the SECURE FreeWare tool:

1. SaveSec Save Security Settings
2. ListSec Lists the file, produced by SaveSec
3. ReSec Reestabish Security Settings from a SaveSec created file.

SaveSec

Sometimes it does make sense, to save the current GUARDIAN security settings BEFORE unsing the SECURE product to secure a large set of file for recoverey reasons. SaveSec does exatly this.

Command syntax:

```
SAVESEC/OUT <file>/[<template>]
```

where:

OUT <file> defines a file, to which the security settings have to be saved. In case the file exists, an error message is generatedm and the opration aborted.
The OUT file will be created by SAVESEC as an unstructured file.

<template> defiles the files to be accessed.
Extended wildcards are allowed.
An incomplete template is resolved with the users current location.
A missing template causes SAVESEC to save the security attributes of the files in the current users location.

Restriction: SAVESEC will NOT collect security settings from OSS type files.

ListSec

The OUT file, produced from SAVESEC, can be listed by the LISTSEC program.

Command syntax:

```
LISTSEC/IN <file>/[<template>]
```

where:

IN <file> defines the SAVESEC type file to be listed.
In case IN is empty, or not a SAVESEC file, an error message is displayed, and the operation aborted.

<template> defiles the files to be listed from the IN file.
Extended wildcards are supported.
An incomplete template is resolved with the users current location.
A missing template causes LISTSEC to list the security attributes of the files, matching the current users location.

ReSec

The OUT file, produced from SAVESEC, can be used to re-establish the original GUARDIAN file security settings for RWEP.

Command syntax:

```
RESEC/IN <file>/[<template>]
```

where:

IN <file> defines the SAVESEC type file to be processed.
In case IN is empty, or not a SAVESEC file, an error message is displayed, and the operation aborted.

<template> defiles the files, for which the RWEP settings have to be re-established.
Extended wildcards are supported.
An incomplete template is resolved with the users current location.
A missing template causes RESEC to retrieve the security attributes of the files, matching the current users location.

PayWare

The PayWare version has one more hard coded command available:

```
SECURE PAYTOK
```

Supplying the key word PAYTOK causes SECURE to check the actual license constellation and to display one of the following messages:

```
SECURE will run for the next xx days.
To license SECURE, please send the following
line to GreenHouse at: Info@GreenHouse.de
SECURE;5JIECRES
```

This message is displayed when SECURE is used within the 62 day test period.

```
SECURE is not licensed.
To license SECURE, please send the following
line to GreenHouse at: Info@GreenHouse.de
SECURE;5JIECRES
```

This message is displayed when SECURE runs outside the test phase and it NOT licensed.

```
The license to run SECURE will expire at the end of April.
To renew the license, please send the following
line to GreenHouse at: Info@GreenHouse.de
SECURE;5JIECRES
```

This message is displayed when SECURE is licensed, but the license is from last year. SECURE will run until the end of April for a smooth transition and allowing the user to get a new PayTok file.

```
SECURE is licensed OK and will run until the end of April
next year.
```

This message is displayed when SECURE is licensed for the current year.

When SECURE is used as PayWare und the license is going to expire, you get informed about the situation with similar messages.

To get a PayTok file, please send the small data string as highlightes above to Info@GreenHoude.de and you get a PayTok name in return, allowing you to use the product for one more year.

INDEX