NAME

ntfssecaudit − NTFS Security Data Auditing

SYNOPSIS

ntfssecaudit [*options***]** *args*

Where *options* is a combination of :

-a full auditing of security data (Linux only)
-b backup ACLs
-e setting extra backed-up parameters (in conjunction with -s)
-h displaying hexadecimal security descriptors saved in a file
-r recursing in a directory
-s setting backed-up ACLs
-u getting a user mapping proposal
-v verbose (very verbose if set twice)

and args define the parameters and the set of files acted upon.

Typing secaudit with no args will display a summary of available options.

DESCRIPTION

ntfssecaudit displays the ownership and permissions of a set of files on an NTFS file system, and checks their consistency. It can be started in terminal mode only (no graphical user interface is available.)

When a *volume* is required, it has to be unmounted, and the command has to be issued as **root**. The *volume* can be either a block device (i.e. a disk partition) or an image file.

When acting on a directory or volume, the command may produce a lot of information. It is therefore advisable to redirect the output to a file or pipe it to a text editor for examination.

OPTIONS

Below are the valid combinations of options and arguments that **ntfssecaudit** accepts. All the indicated arguments are mandatory and must be unique (if wildcards are used, they must resolve to a single name.)

-h *file* Displays in an human readable form the hexadecimal security descriptors saved in *file*. This can be used to turn a verbose output into a very verbose output.

-a[rv] *volume*

Audits the volume : all the global security data on *volume* are scanned and errors are displayed. If option **-r** is present, all files and directories are also scanned and their relations to global security data are checked. This can produce a lot of data.

This option is not effective on volumes formatted for old NTFS versions (pre NTFS 3.0). Such volumes have no global security data.

When errors are signalled, it is advisable to repair the volume with an appropriate tool (such as **chkdsk** on Windows.)

[-v] *volume file*

Displays the security parameters of *file* : its interpreted Linux mode (rwx flags in octal) and Posix ACL[1], its security key if any, and its security descriptor if verbose output.

-r[v] *volume directory*

displays the security parameters of all files and subdirectories in *directory* : their interpreted Linux mode (rwx flags in octal) and Posix ACL[1], their security key if any, and their security descriptor if verbose output.

-b[v] *volume [directory]*

Recursively extracts to standard output the NTFS ACLs of files in *volume* and *directory*.

-s[ev] *volume [backup-file]*

Sets the NTFS ACLS as indicated in *backup-file* or standard input. The input data must have been created on Linux. With option **-e**, also sets extra parameters (currently Windows attrib).

volume perms file

Sets the security parameters of file to perms. Perms is the Linux requested mode (rwx flags, expressed in octal form as in chmod) or a Posix ACL[1] (expressed like in setfacl -m). This sets a new ACL which is effective for Linux and Windows.

-r[v] *volume perms directory*

Sets the security parameters of all files and subdirectories in *directory* to *perms*. Perms is the Linux requested mode (rwx flags, expressed in octal form as in **chmod**), or a Posix ACL[1] (expressed like in **setfacl -m**.) This sets new ACLs which are effective for Linux and Windows.

[-v] *mounted-file*

Displays the security parameters of *mounted-file* : its interpreted Linux mode (rwx flags in octal) and Posix ACL[1], its security key if any, and its security descriptor if verbose output. This is a special case which acts on a mounted file (or directory) and does not require being root. The Posix ACL interpretation can only be displayed if the full path to *mounted-file* from the root of the global file tree is provided.

-u[v] *mounted-file*

Displays a proposed contents for a user mapping file, based on the ownership parameters set by Windows on *mounted-file*, assuming this file was created on Windows by the user who should be mapped to the current Linux user. The displayed information has to be copied to the file **.NTFS-3G/UserMapping** where **.NTFS-3G** is a hidden subdirectory of the root of the partition for which the mapping is to be defined. This will cause the ownership of files created on that partition to be the same as the original *mounted-file*.

NOTE

[1] provided the POSIX ACL option was selected at compile time. A Posix ACL specification looks like "**[d:]{ugmo}:[id]:[perms],...**" where id is a numeric user or group id, and perms an octal digit or a set from the letters r, w and x.

Example : "**u::7,g::5,o:0,u:510:rwx,g:500:5,d:u:510:7**"

EXAMPLES

Audit the global security data on /dev/sda1

ntfssecaudit -ar /dev/sda1

Display the ownership and permissions parameters for files in directory /audio/music on device /dev/sda5, excluding sub-directories :

ntfssecaudit /dev/sda5 /audio/music

Set all files in directory /audio/music on device /dev/sda5 as writeable by owner and read-only for everybody :

ntfssecaudit -r /dev/sda5 644 /audio/music

EXIT CODES

ntfssecaudit exits with a value of 0 when no error was detected, and with a value of 1 when an error was detected.

KNOWN ISSUES

Please see

https://github.com/tuxera/ntfs-3g/wiki/NTFS-3G-FAQ/

for common questions and known issues. If you would find a new one in the latest release of the software then please send an email describing it in detail. You can contact the development team on the ntfs−3g−devel@lists.sf.net address.

AUTHORS

ntfssecaudit has been developed by Jean-Pierre André.

THANKS

Several people made heroic efforts, often over five or more years which resulted the ntfs-3g driver. Most importantly they are Anton Altaparmakov, Richard Russon, Szabolcs Szakacsits, Yura Pakhuchiy, Yuval Fledel, and the author of the groundbreaking FUSE filesystem development framework, Miklos Szeredi.

SEE ALSO

ntfsprogs(8), **attr**(5), **getfattr**(1)