NAME

ntfsdecrypt − decrypt or update NTFS files encrypted according to EFS

SYNOPSIS

ntfsdecrypt [*options*] -k *key.pfx device file*

DESCRIPTION

ntfsdecrypt decrypts a file from an unmounted device and print the decrypted data on the standard output. It can also update an encrypted file with the encryption key unchanged.

The NTFS file encryption (known as EFS) uses a two-level encryption : first, the file contents is encrypted with a random symmetric key, then this symmetric key is encrypted with the public keys of each of the users allowed to decrypt the file (RSA public key encryptions).

Three symmetric encryption modes are currently implemented in ntfsdecrypt : DESX (a DES variant), 3DES (triple DES) and AES_256 (an AES variant).

All the encrypted symmetric keys are stored along with the file in a special extended attribute named "$LOGGED_UTILITY_STREAM". Usually, at least two users are allowed to read the file : its owner and the recovery manager who is able to decrypt all the files in a company. When backing up an encrypted file, it is important to also backup the corresponding $LOGGED_UTILITY_STREAM, otherwise the file cannot be decrypted, even by the recovery manager. Also note that encrypted files are slightly bigger than apparent, and the option "efs_raw" has to be used when backing up encrypted files with ntfs-3g.

When ntfsdecrypt is used to update a file, the keys and the $LOGGED_UTILITY_STREAM are kept unchanged, so a single key file has to be designated.

Note : the EFS encryption is only available in professional versions of Windows;

OPTIONS

Below is a summary of all the options that **ntfsdecrypt** accepts. Nearly all options have two equivalent names. The short name is preceded by− and the long name is preceded by −−. An y single letter options, that don't take an argument, can be combined into a single command, e.g. **−fv** is equi valent to **−f −v**. Long named options can be abbreviated to any unique prefix of their name.

−i, **−−inode** NUM
: Display or update the contents of a file designated through its inode number instead of its name.

−e, **−−encrypt**
: Update an existing encrypted file and get the new contents from the standard input. The full public and private key file has to be designated, as the symmetric key is kept unchanged, so the private key is needed to extract it.

−f, **−−force**
: This will override some sensible defaults, such as not using a mounted volume. Use this option with caution.

−k, **−−keyfile−name** key.pfx
: Define the file which contains the public and private keys in PKCS#12 format. This file obviously contains the keys of one of the users allowed to decrypt or update the file. It has to be extracted from Windows in PKCS#12 format (its usual suffix is .p12 or .pfx), and it is protected by a passphrase which has to be typed in for the keys to be extracted. This can be the key file of any user allowed to read the file, including the one of the recovery manager.

−h, **−−help**
: Show a list of options with a brief description of each one.

−q, **−−quiet**
: Suppress some debug/warning/error messages.

−V, **−−version**
: Show the version number, copyright and license of **ntfsdecrypt**.

−v, **−−verbose**
Display more debug/warning/error messages.

EXAMPLES

Display the contents of the file hamlet.doc in the directory Documents of the root of the NTFS file system on the device /dev/sda1

ntfsdecrypt -k foo.key /dev/sda1 Documents/hamlet.doc

Update the file hamlet.doc

ntfsdecrypt -k foo.key /dev/sda1 Documents/hamlet.doc < new.doc

BUGS

There are no known problems with **ntfsdecrypt**. If you find a bug please send an email describing the problem to the development team:
ntfs−3g−devel@lists.sf.net

AUTHORS

ntfsdecrypt was written by Yuval Fledel, Anton Altaparmakov and Yura Pakhuchiy. It was ported to ntfs-3g by Erik Larsson and upgraded by Jean-Pierre Andre.

AVAILABILITY

ntfsdecrypt is part of the **ntfs-3g** package and is available from:
https://github.com/tuxera/ntfs-3g/wiki/

SEE ALSO

Read **ntfs-3g**(8) for details on option efs_raw,
ntfscat(8), **ntfsprogs**(8)