

**NAME**

**decryptcore** - decrypt a core dump of the operating system

**SYNOPSIS**

**decryptcore** [-fLv] -p *privatekeyfile* -k *keyfile* -e *encryptedcore* -c *core*

**decryptcore** [-fLv] [-d *crashdir*] -p *privatekeyfile* -n *dumpnr*

**DESCRIPTION**

The **decryptcore** utility first decrypts *keyfile* using *privatekeyfile* and then uses the resulting key to decrypt *encryptedcore* saved by `savecore(8)`. The result is saved in *core*.

Alternatively a user can decrypt a core dump numbered *dumpnr* from the *crashdir* directory. In this case a dump key from the *key.#* file is used and the result is saved in the *vmcore.#* file where "#" corresponds to *dumpnr*.

By default **decryptcore** does not overwrite an old core dump as a user might want to store the core somewhere else for the future. This behaviour can be changed using the **-f** flag.

The **decryptcore** utility can be started with the following command line arguments:

- f** Remove a decrypted core dump if it already exists.
- L** Write log messages to `syslogd(8)`.
- v** Print or log verbose/debugging information. This option can be specified multiple times to raise the verbosity level.
- p** *privatekeyfile* Specify location of a private key file which will be used to decrypt a dump key file.
- k** *keyfile* Specify location of a dump key file.
- e** *encryptedcore* Specify location of an encrypted core.
- c** *core* Specify location of a resulting decrypted core dump.
- d** *crashdir* Specify an alternative crash dump directory. The default crash dump directory is */var/crash*.
- n** *dumpnr* Specify a number of a crash dump to be decrypted.

**EXIT STATUS**

The **decryptcore** utility exits 0 on success, and >0 if an error occurs.

**SEE ALSO**

kgdb(1) (*ports/devel/gdb*), capsicum(4), dumpon(8), savecore(8), syslogd(8)

**AUTHORS**

The **decryptcore** utility was implemented by Konrad Witaszczyk <*def@FreeBSD.org*>.