NAME

swtpm_ioctl − Utility for sending control commands to swtpm

SYNOPSIS

swtpm_ioctl [COMMAND] [<device>]

DESCRIPTION

swtpm_ioctl implements a client tool for controlling the *swtpm_cuse* and *swtpm* TPM software emulators, such as for example their initialization and shutdown. Once it has been initialized, TPM commands can be sent to it.

Note: The environment variable SWTPM_IOCTL_BUFFERSIZE can be set to the size for the buffer for state blob transfer to use. If it is not set, the **ioctl()** interface is used for transferring the state. This environment variable is primarily used for testing purposes.

The following commands are supported:

−−tpm−device <device>

Use the given device. The full path to the character device must be provided, such as for example /dev/vtpm−200.

This option can be used instead of providing the device as the last parameter.

−−tcp <server>:<port>

Connect to the given server and port; if no server is given, 127.0.0.1 is used; if port is not given, the default port 6545 is used.

−−unix <path>

Connect to the given UnixIO path.

−c Get the capability flags indicating which commands are supported.

−i Send a hardware initialization signal to the swtpm_cuse/swtpm. Volatile state previously written by the TPM will be read and the file automatically delete.

−s Initiate a graceful shut down.

−−stop

Stop the swtpm_cuse/swtpm. This does not shut it down. The *−i* command can again be sent to it. After a stop it is also possible to load TPM stateblobs into the TPM using the *−−load* command.

−e Get the tpmEstablished bit.

−r locality

Reset the tpmEstablished bit using the given locality. Only localities 3 and 4 work. This operation will not permanently change the localty that was previously set using the *−l* option.

−l locality

Set the locality for the subsequent TPM commands.

−v Have the TPM write the volatile state to a file. Upon a TPM_Init (−i) the TPM state will be read and the TPM can then resume operation without further intialization.

−C Cancel an ongoing TPM command.

−h data

Reset and extend PCR 17 with the hash of the given data. If data is the single character ’−’, then all data are read from stdin.

−−save <TPM state blob name> <filename>

Save the TPM state blob into the given file. Valid TPM state blob names are ’permanent’, ’volatile’, and ’savestate’.

Note that this command can be executed at any time. However, to retrieve the latest volatile state, the *−v* command should have been run immediately before running this command. The savestate blob will only be returned if a TPM_SaveState command was executed in the TPM (TPM 1.2).

−−load <TPM state blob name> <filename>

Load the given TPM state blob from the given file. Valid TPM state blob names are ’permanent’, ’volatile’, and ’savestate’.

Note that this command can only be executed on a TPM that is shut down. To then start the TPM with the uploaded state, the *−i* command must be issued.

−g Get configuration flags that for example indicate which keys (file encryption or migration key) are in use by the TPM.

−−info <flag>

Get information about the TPM implementation in JSON format. The flag *TPMLIB_INFO_TPMSPECIFICATION*, which has the value 1, returns information about the specification the TPM implementation followed. The flag *TPMLIB_INFO_TPMATTRIBUTES*, which has the value 2, returns information about the manufacturer, model, and version of the TPM.

EXAMPLE

Start swtpm on port 10000 for the control port and emulate a TPM 1.2:

```
#> swtpm socket −−tpmstate dir=/tmp/myvtpm1 −−log level=4 −−ctrl type=tcp,port
```

Get information about the TPM implementation in JSON:

```
#> swtpm_ioctl −−tcp :10000 −−info 1
{"TPMSpecification":{"family":"1.2","level":2,"revision":116}}
#> swtpm_ioctl −−tcp :10000 −−info 2
{"TPMAttributes":{"manufacturer":"id:00001014","version":"id:00740001","model"
```

Shut down the swtpm

```
#> swtpm_ioctl −−tcp :10000 −s
```

SEE ALSO

swtpm_cuse