NAME

swtpm_setup − Swtpm tool to simulate the manufacturing of a TPM 1.2 or 2.0

SYNOPSIS

swtpm_setup [OPTIONS]

DESCRIPTION

swtpm_setup is a tool that prepares the initial state for a libtpms-based TPM.

The following options are supported:

−−runas <userid>
Use this userid to run swtpm_setup.sh as. Only 'root' can use this option.

−−config <file>
Path to configuration file containing the tool to use for creating certificates; see also **swtpm_setup.conf**

If this parameter is not provided, the default configuration file will be used. The search order for the default configuration file is as follows. If the environment variable XDG_CONFIG_HOME is set, ${XDG_CONFIG_HOME}/swtpm_setup.conf will be used if available, otherwise if the environment variable HOME is set, ${HOME}/.config/swtpm_setup.conf will be used if available. If none of the previous ones are available, /etc/swtpm_setup.conf will be used.

−−tpm−state <dir> or **−−tpmstate <dir>**
Path to a directory where the TPM's state will be written into; this is a mandatory argument

−−tpm <path to executable>
Path to the TPM executable; this is an optional argument and by default the swtpm executable found in the PATH will be used.

−−tpm2
Do setup on a TPM 2; by default a TPM 1.2 is setup.

−−createek
Create an endorsement key (EK).

−−allow−signing
Create an EK that can sign. This option requires −−tpm2.

This option will create a non-standard EK. When re-creating the EK, TPM 2 tools have to use the EK Template that is witten at an NV index corresponding to the created EK (e.g., NV index 0x01c00004 for RS 2048 EK). Otherwise the tool-created EK will not correspond to the actual key being used or the modulus shown in the EK certificate.

Note that the TCG specification ''EK Credential Profile For TPM Family 2.0; Level 0'' suggests in its section on ''EK Usage'' that ''the Endorsement Key can be a created as a decryption or signing key.'' However, some platforms will not accept an EK as a signing key, or as a signing and encryption key, and therefore this option should be used very carefully.

−−decryption
Create an EK that can be used for key encipherment. This is the default unless −−allow−signing is passed. This option requires −−tpm2.

−−ecc
Create elliptic curve crypto (ECC) keys; by default RSA keys are generated.

−−take−ownership
Take ownership; this option implies −−createek. This option is only available for TPM 1.2.

−−ownerpass <password>
Provide custom owner password; default is 'ooo'. This option is only available for TPM 1.2.

−−owner−well−known
Use a password of all zeros (20 bytes of zeros) as the owner password. This option is only available for TPM 1.2.

−−srkpass <password>
Provide custom SRK password; default is 'sss'. This option is only available for TPM 1.2.

−−srk−well−known
Use a password of all zeros (20 bytes of zeros) as the SRK password. This option is only available for TPM 1.2.

−−create−ek−cert
Create an EK certificate; this implies −−createek.

−−create−platform−cert
Create a platform certificate; this implies −−create−ek−cert.

−−lock−nvram
Lock NVRAM access to all NVRAM locations that were written to.

−−display
At the end display as much info as possible about the configuration of the TPM.

−−logfile <logfile>
The logfile to log to. By default logging goes to stdout and stderr.

−−keyfile <keyfile>
The key file contains an ASCII hex key consisting of 32 hex digits with an optional leading '0x'. This is the key to be used by the TPM emulator for encrypting the state of the TPM.

−−keyfile−fd <file descriptor>
Like **−−keyfile** but the key will be read from the file descriptor.

−−pwdfile <passphrase file>
The passphrase file contains a passphrase from which the TPM emulator will derive the encryption key from and use the key for encrypting the TPM state.

−−pwdfile−fd <file descriptor>
Like **−−pwdfile** but the passphrase will be read from the file descriptor.

−−ciper <cipher>
The cipher may be either aes-cbc or aes−128−cbc for 128 bit AES encryption, or aes−256−cbc for 256 bit AES encryption. The same cipher must be used on the *swtpm* command line later on.

−−overwrite
Overwrite existing TPM state. All previous state will be erased. If this option is not given and an existing state file is found, an error code is returned.

−−not−overwrite
Do not overwrite existing TPM state. If existing TPM state is found, the program ends without an error.

−−vmid <VM ID>
Optional VM ID that can be used to keep track of certificates issued for VMs (or containers). This parameter will be passed through to the tool used for creating the certificates and may be required by that tool.

−−pcr−banks <PCR banks>
Optional comma-separated list of PCR banks to activate. Providing '−' allows a user to skip the selection and activates all PCR banks. By default the sha1 and sha256 banks are activated.

−−swtpm_ioctl <executable>
Pass the path to the swtpm_ioctl executable. By default the swtpm_ioctl in the PATH is used.

−−tcsd−system−ps−file <file>
This option is deprecated and has no effect (since v0.4).

−−rsa−keysize <keysize> (since v0.4)
This option allows to pass the size of a TPM 2 RSA EK key, such as 2048 or 3072. The supported keysizes for a TPM 2 can be queried for using the *−−print−capabilities* option. The default size is 2048 bits for both TPM 1.2 and TPM 2. If 'max' is passed, the largest possible key size is used.

−−print−capabilities (since v0.2)
Print capabilities that were added to swtpm_setup after version 0.1. The output may contain the following:

```
{
  "type": "swtpm_setup",
  "features": [
    "cmdarg−keyfile−fd",
    "cmdarg−pwdfile−fd",
    "tpm2−rsa−keysize−2048",
    "tpm2−rsa−keysize−3072",
    "tpm12−not−need−root"
  ]
}
```

The meaning of the feature verbs is as follows:

cmdarg-key-fd
The *−−keyfile−fd* option is supported.

cmdarg-pwd-fd
The *−−pwdfile−fd* option is supported.

tpm2−rsa−keysize−2048, ...
The shown RSA key sizes are supported for a TPM 2's EK key. If none of the tpm2−rsa−keysize verbs is shown then only RSA 2048 bit keys are supported.

tpm12−not−need−root (since 0.4.0)
This option implies that any user can setup a TPM 1.2. Previously only root or the 'tss' user, depending on configuration and availability of this account, could do that.

−−help, −h
Display the help screen

EXAMPLE USAGE

To simulate manufacturing of a TPM, one would typically run the following command:

```
#> sudo swtpm_setup −−tpmstate /tmp/mytpm1/ \
    −−create−ek−cert −−create−platform−cert −−lock−nvram
```

Note: since v0.4 TPM 1.2 setup does not require root rights anymore.

Any user can also simulate the manufacturing of a TPM using the swtpm-localca plugin. The following example assumes that the user has set the environment variable XDG_CONFIG_HOME as follows (using bash for example):

```
export XDG_CONFIG_HOME=~/.config
```

Note: The XDG_CONFIG_HOME variable is part of the XDG Base Directory Specification.

The following configuration files need to be created:

~/.config/swtpm_setup.conf:

```
# Program invoked for creating certificates
create_certs_tool= /usr/share/swtpm/swtpm-localca
create_certs_tool_config = ${XDG_CONFIG_HOME}/swtpm-localca.conf
create_certs_tool_options = ${XDG_CONFIG_HOME}/swtpm-localca.options
```

~/.config/swtpm-localca.conf:

```
statedir = ${XDG_CONFIG_HOME}/var/lib/swtpm-localca
signingkey = ${XDG_CONFIG_HOME}/var/lib/swtpm-localca/signkey.pem
issuercert = ${XDG_CONFIG_HOME}/var/lib/swtpm-localca/issuercert.pem
certserial = ${XDG_CONFIG_HOME}/var/lib/swtpm-localca/certserial
```

~/.config/swtpm-localca.options:

```
--platform-manufacturer Fedora
--platform-version 2.12
--platform-model QEMU
```

Note: The tool swtpm-create-user-config-files can be used to create such files (with different content):

```
#> /usr/share/swtpm/swtpm-create-user-config-files
Writing /home/stefanb/.config/swtpm_setup.conf.
Writing /home/stefanb/.config/swtpm-localca.conf.
Writing /home/stefanb/.config/swtpm-localca.options.
```

The following commands now create a TPM 2 with an EK and platform certificate. The state of the TPM 2 will be stored in the directory ${XDG_CONFIG_HOME}/mytpm1.

```
#> mkdir -p ${XDG_CONFIG_HOME}/mytpm1
#> swtpm_setup --tpm2 --tpmstate ${XDG_CONFIG_HOME}/mytpm1 \
    --create-ek-cert --create-platform-cert --lock-nvram
```

SEE ALSO

swtpm_setup.conf

REPORTING BUGS

Report bugs to Stefan Berger <stefanb@linux.ibm.com>