MyWebUniversity
python
Rocky Linux
Manual Pages
FPDF Library
PDF generator
FPDF
Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'tpm2_policypcr.1'

$ man tpm2_policypcr.1

tpm2_policypcr(1)   General Commands Manual   tpm2_policypcr(1)

NAME

tpm2_policypcr(1) - Create a policy that includes specific PCR values.

SYNOPSIS

tpm2_policypcr [OPTIONS]

DESCRIPTION

tpm2_policypcr(1) - Generates a PCR policy event with the TPM. A PCR

policy event creates a policy bound to specific PCR values and is use?

ful within larger policies constructed using policyor and policyautho?

rize events. See tpm2_policyor(1) and tpm2_policyauthorize(1) respec?

tively for their usages. The PCR data factored into the policy can be

specified in one of 3 ways: 1. A file containing a concatenated list

of PCR values as in the output from tpm2_pcrread. 2. Requiring the

PCR values be read off the TPM by not specifying a PCR file input.   3.

The digest of all the PCR values directly specified as an argument.

OPTIONS

? -L, --policy=FILE:

File to save the policy digest.

? -f, --pcr=FILE:

Optional Path or Name of the file containing expected PCR values for the specified index. Default is to read the current PCRs per the set list.

? -l, --pcr-list=PCR:

The list of PCR banks and selected PCRs? ids for each bank.

? -S, --session=FILE:

The policy session file generated via the -S option to tpm2_star? tauthsession(1).

? ARGUMENT: The calculated digest of all PCR values specified as a hex byte stream. Eg: openssl dgst -sha256 -binary pcr.bin | xxd -p -c 32

References

Context Object Format

The type of a context object, whether it is a handle or file name, is determined according to the following logic in-order:

? If the argument is a file path, then the file is loaded as a restored TPM transient object.

? If the argument is a prefix match on one of:

? owner: the owner hierarchy

? platform: the platform hierarchy

? endorsement: the endorsement hierarchy

? lockout: the lockout control persistent object

? If the argument argument can be loaded as a number it will be treat as a handle, e.g. 0x81010013 and used directly._OBJECT_.

Authorization Formatting

Authorization for use of an object in TPM2.0 can come in 3 different forms: 1. Password 2. HMAC 3. Sessions

NOTE: ?Authorizations default to the EMPTY PASSWORD when not speci? fied?.

Passwords

Passwords are interpreted in the following forms below using prefix identifiers.

Note: By default passwords are assumed to be in the string form when they do not have a prefix.

String

A string password, specified by prefix ?str:? or it?s absence (raw string without prefix) is not interpreted, and is directly used for au?thorization.

Examples

```
foobar

str:foobar
```

Hex-string

A hex-string password, specified by prefix ?hex:? is converted from a hexidecimal form into a byte array form, thus allowing passwords with non-printable and/or terminal un-friendly characters.

Example

```
hex:0x1122334455667788
```

File

A file based password, specified be prefix ?file:? should be the path of a file containing the password to be read by the tool or a ?-? to use stdin. Storing passwords in files prevents information leakage, passwords passed as options can be read from the process list or common shell history features.

Examples

```
# to use stdin and be prompted

file:-

# to use a file from a path

file:path/to/password/file

# to echo a password via stdin:

echo foobar | tpm2_tool -p file:-

# to use a bash here-string via stdin:

tpm2_tool -p file:- <<< foobar
```

Sessions

When using a policy session to authorize the use of an object, prefix the option argument with the session keyword. Then indicate a path to a session file that was created with tpm2_startauthsession(1). Option?

ally, if the session requires an auth value to be sent with the session handle (eg policy password), then append a + and a string as described in the Passwords section.

Examples

To use a session context file called session.ctx.

```
session:session.ctx
```

To use a session context file called session.ctx AND send the authvalue mypassword.

```
session:session.ctx+mypassword
```

To use a session context file called session.ctx AND send the HEX auth?value 0x11223344.

```
session:session.ctx+hex:11223344
```

PCR Authorizations

You can satisfy a PCR policy using the ?pcr:? prefix and the PCR mini?language. The PCR minilanguage is as follows: <pcr-spec>=<raw-pcr-file>

The PCR spec is documented in in the section ?PCR bank specifiers?.

The raw-pcr-file is an optional argument that contains the output of the raw PCR contents as returned by tpm2_pcrread(1).

PCR bank specifiers (pcr.md)

Examples

To satisfy a PCR policy of sha256 on banks 0, 1, 2 and 3 use a specifi?er of:

```
pcr:sha256:0,1,2,3
```

specifying AUTH.

PCR Bank Specifiers

PCR Bank Selection lists follow the below specification:

```
<BANK>:<PCR>[,<PCR>] or <BANK>:all
```

multiple banks may be separated by `+'.

For example:

```
sha1:3,4+sha256:all
```

will select PCRs 3 and 4 from the SHA1 bank and PCRs 0 to 23 from the SHA256 bank.

Note

PCR Selections allow for up to 5 hash to pcr selection mappings.   This is a limitation in design in the single call to the tpm to get the pcr values. PCR.

COMMON OPTIONS

This collection of options are common to many programs and provide in?formation that many users may expect.

? -h, --help=[man|no-man]: Display the tools manpage. By default, it attempts to invoke the manpager for the tool, however, on failure will output a short tool summary. This is the same behavior if the ?man? option argument is specified, however if explicit ?man? is re?quested, the tool will provide errors from man on stderr. If the ?no-man? option if specified, or the manpager fails, the short op?tions will be output to stdout.

To successfully use the manpages feature requires the manpages to be installed or on MANPATH, See man(1) for more details.

? -v, --version: Display version information for this tool, supported tctis and exit.

? -V, --verbose: Increase the information that the tool prints to the console during its execution. When using this option the file and line number are printed.

? -Q, --quiet: Silence normal tool output to stdout.

? -Z, --enable-errata: Enable the application of errata fixups. Useful if an errata fixup needs to be applied to commands sent to the TPM. Defining the environment TPM2TOOLS_ENABLE_ERRATA is equivalent. in?formation many users may expect.

TCTI Configuration

The TCTI or ?Transmission Interface? is the communication mechanism with the TPM. TCTIs can be changed for communication with TPMs across different mediums.

To control the TCTI, the tools respect:

1. The command line option -T or --tcti

2. The environment variable: TPM2TOOLS_TCTI.

Note: The command line option always overrides the environment vari?
able.

The current known TCTIs are:

? tabrmd - The resource manager, called tabrmd
  (https://github.com/tpm2-software/tpm2-abrmd). Note that tabrmd and
  abrmd as a tcti name are synonymous.

? mssim - Typically used for communicating to the TPM software simula?
  tor.

? device - Used when talking directly to a TPM device file.

? none - Do not initalize a connection with the TPM. Some tools allow
  for off-tpm options and thus support not using a TCTI. Tools that do
  not support it will error when attempted to be used without a TCTI
  connection. Does not support ANY options and MUST BE presented as
  the exact text of ?none?.

The arguments to either the command line option or the environment
variable are in the form:

<tcti-name>:<tcti-option-config>

Specifying an empty string for either the <tcti-name> or <tcti-op?
tion-config> results in the default being used for that portion respec?
tively.

TCTI Defaults

When a TCTI is not specified, the default TCTI is searched for using
dlopen(3) semantics. The tools will search for tabrmd, device and
mssim TCTIs IN THAT ORDER and USE THE FIRST ONE FOUND. You can query
what TCTI will be chosen as the default by using the -v option to print
the version information. The ?default-tcti? key-value pair will indi?
cate which of the aforementioned TCTIs is the default.

Custom TCTIs

Any TCTI that implements the dynamic TCTI interface can be loaded. The
tools internally use dlopen(3), and the raw tcti-name value is used for
the lookup. Thus, this could be a path to the shared library, or a li?
brary name as understood by dlopen(3) semantics.

This collection of options are used to configure the various known TCTI modules available:

? device: For the device TCTI, the TPM character device file for use by the device TCTI can be specified. The default is /dev/tpm0.

Example: -T device:/dev/tpm0 or export TPM2TOOLS_TCTI=?de?
vice:/dev/tpm0?

? mssim: For the mssim TCTI, the domain name or IP address and port number used by the simulator can be specified. The default are 127.0.0.1 and 2321.

Example: -T mssim:host=localhost,port=2321 or export TPM2TOOLS_TC?
TI=?mssim:host=localhost,port=2321?

? abrmd: For the abrmd TCTI, the configuration string format is a se?
ries of simple key value pairs separated by a `,' character. Each key and value string are separated by a `=' character.

? TCTI abrmd supports two keys:

1. `bus_name' : The name of the tabrmd service on the bus (a string).
2. `bus_type' : The type of the dbus instance (a string) limited to `session' and `system'.

Specify the tabrmd tcti name and a config string of bus_name=com.ex?
ample.FooBar:

```
\--tcti=tabrmd:bus_name=com.example.FooBar
```

Specify the default (abrmd) tcti and a config string of bus_type=ses?
sion:

```
\--tcti:bus_type=session
```

NOTE: abrmd and tabrmd are synonymous. the various known TCTI mod?
ules.

EXAMPLES

Starts a trial session, builds a PCR policy and uses that policy in the creation of an object. Then, it uses a policy session to unseal some data stored in the object.

Step 1: create a policy

```
tpm2_createprimary -C e -g sha256 -G ecc -c primary.ctx
```

```
tpm2_pcrread -o pcr.dat "sha1:0,1,2,3"

tpm2_startauthsession -S session.dat

tpm2_policypcr -S session.dat -l "sha1:0,1,2,3" -f pcr.dat -L policy.dat

tpm2_flushcontext session.dat
```

Step 2: create an object using that policy

```
tpm2_create -Q -u key.pub -r key.priv -C primary.ctx -L policy.dat \

-i- <<< "12345678"

tpm2_load -C primary.ctx -u key.pub -r key.priv -n unseal.key.name \

-c unseal.key.ctx
```

Step 3: Satisfy the policy

```
tpm2_startauthsession --policy-session -S session.dat

tpm2_policypcr -S session.dat -l "sha1:0,1,2,3" -f pcr.dat -L policy.dat
```

Step 4: Use the policy

```
tpm2_unseal -psession:session.dat -c unseal.key.ctx

12345678

tpm2_flushcontext session.dat
```

Returns

Tools can return any of the following codes:

? 0 - Success.

? 1 - General non-specific error.

? 2 - Options handling error.

? 3 - Authentication error.

? 4 - TCTI related error.

? 5 - Non supported scheme. Applicable to tpm2_testparams.

Limitations

It expects a session to be already established via tpm2_startauthses?

sion(1) and requires one of the following:

? direct device access

? extended session support with tpm2-abrmd.

Without it, most resource managers will not save session state between

command invocations.

BUGS

Github Issues (https://github.com/tpm2-software/tpm2-tools/issues)

HELP

See the Mailing List (https://lists.01.org/mailman/listinfo/tpm2)