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_policyticket.1'

$ man tpm2_policyticket.1

tpm2_ticket(1)   General Commands Manual   tpm2_ticket(1)

NAME

tpm2_ticket(1) - Enables policy authorization by verifying a ticket

that represents a validated authorization that had an expiration time

associated with it.

SYNOPSIS

tpm2_ticket [OPTIONS]

DESCRIPTION

tpm2_ticket(1) - Enables policy authorization by verifying a ticket

that represents a validated authorization that had an expiration time

associated with it.

OPTIONS

? -L, --policy=FILE:

File to save the compounded policy digest.

? -S, --session=FILE:

The policy session file generated via the -S option to tpm2_star?

tauthsession(1).

? -n, --name=FILE:

Name of the object that validated the authorization.

? --ticket=FILE:

The ticket file to record the authorization ticket structure.

? --timeout=FILE:

The file path to record the timeout structure returned.

? -q, --qualification=FILE_OR_HEX_STR:

Optional, the policy qualifier data that the signer can choose to include in the signature. Can be either a hex string or path.

References

COMMON OPTIONS

This collection of options are common to many programs and provide information 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 requested, the tool will provide errors from man on stderr. If the ?no-man? option if specified, or the manpager fails, the short options 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. information 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 library name as understood by dlopen(3) semantics.

TCTI OPTIONS

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=?device:/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_TCTI=?mssim:host=localhost,port=2321?

? abrmd: For the abrmd TCTI, the configuration string format is a series 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.example.FooBar:

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

Specify the default (abrmd) tcti and a config string of bus_type=session:

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

NOTE: abrmd and tabrmd are synonymous. the various known TCTI modules.

EXAMPLES

Authorize a TPM operation on an object whose authorization is bound to specific signing authority.

Create the signing authority and load the verification key

```
openssl genrsa -out private.pem 2048

openssl rsa -in private.pem -outform PEM -pubout -out public.pem

tpm2_loadexternal -C o -G rsa -u public.pem -c signing_key.ctx \

-n signing_key.name
```

Generate signature with the expiry time

```
EXPIRYTIME="FFFFFE0C"

echo $EXPIRYTIME | xxd -r -p | \

openssl dgst -sha256 -sign private.pem -out signature.dat
```

Create the policy

```
tpm2_startauthsession -S session.ctx

tpm2_policysigned -S session.ctx -g sha256 -s signature.dat -f rsassa \

-c signing_key.ctx -L policy.signed

tpm2_flushcontext session.ctx
```

Create a sealing object

```
tpm2_createprimary -C o -c prim.ctx -Q

echo "plaintext" > secret.dat

tpm2_create -u sealing_key.pub -r sealing_key.priv -c sealing_key.ctx \

-C prim.ctx -i secret.dat -L policy.signed -Q
```

Create ticket-able policy

```
tpm2_startauthsession -S session.ctx --nonce-tpm=nonce.test --policy-session

{ cat nonce.test & echo $EXPIRYTIME | xxd -r -p; } | \

openssl dgst -sha256 -sign private.pem -out signature.dat

tpm2_policysigned -S session.ctx -g sha256 -s signature.dat -f rsassa \

-c signing_key.ctx -x nonce.test --ticket tic.ket --timeout time.out \

-t 0xFFFFFE0C

tpm2_flushcontext session.ctx
```

##Test with policyticket instead of policysigned

```
tpm2_startauthsession -S session.ctx --policy-session

tpm2_policyticket -S session.ctx -n signing_key.name --ticket tic.ket \

--timeout time.out
```

```
tpm2_unseal -p session:session.ctx -c sealing_key.ctx
```

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)