Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'tpm2_changeauth.1' command

$ man tpm2_changeauth.1

tpm2_changeauth(1) General Commands Manual tpm2_changeauth(1)

NAME

tpm2_changeauth - Changes authorization values for TPM objects.

SYNOPSIS

tpm2_changeauth [OPTIONS] [ARGUMENT]

DESCRIPTION

tpm2_changeauth - Configures authorization values for the various hier?

archies, NV indices, transient and persistent objects.

Note: For non-permanent objects (Transient objects and Persistent ob?

jects), copies of the private information (files or persistent handles)

created prior to changing auth are not invalidated.

OPTIONS

Passwords should follow the ?password authorization formatting stan?

dards?, see section ?Authorization Formatting?.

? -c, --object-context=OBJECT:

The key context object to be used for the operation.

? -p, --object-auth=AUTH:

The old authorization value for the TPM object specified with -c.

? -C, --parent-context=OBJECT:

The parent object. This is required if the object for the operation

is a transient or persistent object.

? -r, --private=FILE: The output file which contains the new sensitive

portion of the object whose auth was being changed. # Protection De?

tails

Objects that can move outside of TPM need to be protected (confiden?
tiality and integrity). For instance, transient objects require that
TPM protected data (key or seal material) be stored outside of the TPM.
This is seen in tools like tpm2_create(1), where the -r option outputs
this protected data. This blob contains the sensitive portions of the
object. The sensitive portions of the object are protected by the par?
ent object, using the parent?s symmetric encryption details to encrypt
the sensitive data and HMAC it.

In-depth details can be found in sections 23 of:

? https://trustedcomputinggroup.org/wp-content/up?
loads/TPM-Rev-2.0-Part-1-Architecture-01.38.pdf

Notably Figure 20, is relevant, even though it?s specifically referring
to duplication blobs, the process is identical.

If the output is from tpm2_duplicate(1), the output will be slightly
different, as described fully in section 23.

? --cphash=FILE

File path to record the hash of the command parameters. This is com?
monly termed as cpHash. NOTE: When this option is selected, The tool
will not actually execute the command, it simply returns a cpHash,
unless rphash is also required.

? --rphash=FILE

File path to record the hash of the response parameters. This is
commonly termed as rpHash.

? -S, --session=FILE:

The session created using tpm2_startauthsession. This can be used to
specify an auxiliary session for auditing and or encryption/decryp?
tion of the parameters.

? ARGUMENT the command line argument specifies the AUTH to be set for
the object specified with -c.

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.

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.

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=?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

Set owner, endorsement and lockout authorizations to newpass

```
tpm2_changeauth -c owner newpass
tpm2_changeauth -c endorsement newpass
tpm2_changeauth -c lockout newpass
```

Change owner, endorsement and lockout authorizations

```
tpm2_changeauth -c o -p newpass newerpass
tpm2_changeauth -c e -p newpass newerpass
tpm2_changeauth -c l -p newpass newerpass
```

Set owner authorization to empty password

```
tpm2_changeauth -c o -p oldpass
```

Modify authorization for a loadable transient object

```
tpm2_createprimary -Q -C o -c prim.ctx
tpm2_create -Q -g sha256 -G aes -u key.pub -r key.priv -C prim.ctx
tpm2_load -C prim.ctx -u key.pub -r key.priv -n key.name -c key.ctx
tpm2_changeauth -c key.ctx -C prim.ctx -r key.priv newkeyauth
```

Modify authorization for a NV Index

Requires Extended Session Support.

```
tpm2_startauthsession -S session.ctx
tpm2_policycommandcode -S session.ctx -L policy.nvchange TPM2_CC_NV_ChangeAuth
tpm2_flushcontext session.ctx
NVIndex=0x1500015
tpm2_nvdefine   $NVIndex -C o -s 32 -a "authread|authwrite" -L policy.nvchange
tpm2_startauthsession \--policy-session -S session.ctx
tpm2_policycommandcode -S session.ctx -L policy.nvchange TPM2_CC_NV_ChangeAuth
tpm2_changeauth -p session:session.ctx -c $NVIndex newindexauth
```

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.

BUGS

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

HELP

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