Lanner

tpm2-tools
Quick Start Guide

Version: 1.1

Date of Release:2019-07-18

Online Resources

To obtain additional documentation resources and software updates for your system, please visit the Lanner Download Center. As certain categories of documents are only available to users who are logged in, please be registered for a Lanner Account at http://www.lannerinc.com/ to access published documents and downloadable resources.

For troubleshooting the issues with your system, please check the Lanner Q&A page for a diagnostic procedure and troubleshooting steps.

Technical Support

In addition to contacting your distributor or sales representative, you could use other available methods to get support from Lanner:

Submitting a Ticket

Visit the **Lanner Technical Support** page at http://www.lannerinc.com/technical-support where you can fill in a support ticket to our technical support department.

Calling Us

A toll-free phone support is offered to our customers in the United States and Canada; it is:

+1-855-852-6637

Copyright and Trademarks

Documentation Feedback

Your feedback is valuable to us, as it will help us continue to provide you with more accurate and relevant documentation. To provide any feedback, comments or to report an error, please email to contact@lannerinc.com, Thank you for your time.

Table of Contents

TPM2-Tools Overview

The **tpm2-tools** is a collection of both low-level and aggregate command line tools that provide access to a tpm2.0 compatible device. This is an open source project and available at https://github.com/tpm2-software/tpm2-tools.

Installation by Pre-built Package of Linux Distribution

tpm2-tools resides in GitHub and is available in source code. Many Linux distributions already had it pre-built and packaged into the repository for user to install, including:

- Arch Linux
- CentOS
- Debian
- Fedora
- Ubuntu
- RedHat

Detailed information could be found in https://pkgs.org/download/tpm2-tools.

Varied distributions adopt different ways to install **tpm2-tools** package. See below for an installation example with **CentOS 7**:

```
[root@localhost ~]# uname -a
Linux localhost.localdomain 3.10.0-693.21.1.el7.x86_64 #1 SMP Wed Mar 7 19:03:37 UTC 2018 x86_64 x86_64 x86_64
GNU/Linux
[root@localhost ~]#
[root@localhost ~]# yum install tpm2-tools
Loaded plugins: fastestmirror, langpacks
base                                                     | 3.6 kB     00:00
docker-ce-edge                                           | 3.5 kB     00:00
docker-ce-stable                                         | 3.5 kB     00:00
docker-ce-test                                           | 3.5 kB     00:00
epel/x86_64/metalink                                     | 4.4 kB     00:00
epel                                                     | 5.3 kB     00:00
extras                                                   | 3.4 kB     00:00
updates                                                  | 3.4 kB     00:00
(1/13): base/7/x86_64/group_gz                             | 166 kB   00:00
(2/13): docker-ce-stable/x86_64/updateinfo                 |   55 B   00:00
(3/13): docker-ce-edge/x86_64/primary_db                   |  33 kB   00:00
(4/13): docker-ce-stable/x86_64/primary_db                 |  29 kB   00:00
(5/13): docker-ce-test/x86_64/updateinfo                   |   55 B   00:00
(6/13): docker-ce-edge/x86_64/updateinfo                   |   55 B   00:00
(7/13): epel/x86_64/group_gz                               |  88 kB   00:00
(8/13): docker-ce-test/x86_64/primary_db                   |  92 kB   00:00
(9/13): epel/x86_64/updateinfo                             | 988 kB   00:01
(10/13): extras/7/x86_64/primary_db                        | 205 kB   00:00
(11/13): base/7/x86_64/primary_db                          | 6.0 MB   00:04
(12/13): updates/7/x86_64/primary_db                       | 6.5 MB   00:04
(13/13): epel/x86_64/primary_db                            | 6.8 MB   00:10
```

```
Determining fastest mirrors
 * base: free.nchc.org.tw
 * epel: my.fedora.ipserverone.com
 * extras: free.nchc.org.tw
 * updates: free.nchc.org.tw

Resolving Dependencies
--> Running transaction check
---> Package tpm2-tools.x86_64 0:3.0.4-2.el7 will be installed
--> Processing Dependency: tpm2-tss(x86-64) >= 1.3.0-1.el7 for package: tpm2-tools-3.0.4-2.el7.x86_64
--> Processing Dependency: libtcti-tabrmd.so.0()(64bit) for package: tpm2-tools-3.0.4-2.el7.x86_64
--> Processing Dependency: libtcti-socket.so.0()(64bit) for package: tpm2-tools-3.0.4-2.el7.x86_64
--> Processing Dependency: libtcti-device.so.0()(64bit) for package: tpm2-tools-3.0.4-2.el7.x86_64
--> Processing Dependency: libsapi.so.0()(64bit) for package: tpm2-tools-3.0.4-2.el7.x86_64
--> Running transaction check
---> Package tpm2-abrmd.x86_64 0:1.1.0-10.el7 will be installed
--> Processing Dependency: tpm2-tss-devel(x86-64) >= 1.4.0-1.el7 for package:
tpm2-abrmd-1.1.0-10.el7.x86_64
---> Package tpm2-tss.x86_64 0:1.4.0-2.el7 will be installed
--> Running transaction check
---> Package tpm2-tss-devel.x86_64 0:1.4.0-2.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package              Arch          Version              Repository    Size
================================================================================
Installing:
 tpm2-tools           x86_64        3.0.4-2.el7          base        365 k
Installing for dependencies:
 tpm2-abrmd           x86_64        1.1.0-10.el7         base         84 k
 tpm2-tss             x86_64        1.4.0-2.el7          base         62 k
 tpm2-tss-devel       x86_64        1.4.0-2.el7          base         55 k

Transaction Summary
================================================================================
Install  1 Package (+3 Dependent packages)

Total download size: 567 k
Installed size: 2.4 M
Is this ok [y/d/N]: y
Downloading packages:
(1/4): tpm2-abrmd-1.1.0-10.el7.x86_64.rpm               |  84 kB   00:00
(2/4): tpm2-tss-devel-1.4.0-2.el7.x86_64.rpm            |  55 kB   00:00
(3/4): tpm2-tss-1.4.0-2.el7.x86_64.rpm                  |  62 kB   00:00
(4/4): tpm2-tools-3.0.4-2.el7.x86_64.rpm                | 365 kB   00:00
--------------------------------------------------------------------------------
Total                                          1.5 MB/s | 567 kB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

```
  Installing : tpm2-tss-1.4.0-2.el7.x86_64                                  1/4
  Installing : tpm2-tss-devel-1.4.0-2.el7.x86_64                            2/4
  Installing : tpm2-abrmd-1.1.0-10.el7.x86_64                               3/4
  Installing : tpm2-tools-3.0.4-2.el7.x86_64                                4/4
  Verifying  : tpm2-tss-1.4.0-2.el7.x86_64                                  1/4
  Verifying  : tpm2-tss-devel-1.4.0-2.el7.x86_64                            2/4
  Verifying  : tpm2-tools-3.0.4-2.el7.x86_64                                3/4
  Verifying  : tpm2-abrmd-1.1.0-10.el7.x86_64                               4/4

Installed:
  tpm2-tools.x86_64 0:3.0.4-2.el7

Dependency Installed:
  tpm2-abrmd.x86_64 0:1.1.0-10.el7         tpm2-tss.x86_64 0:1.4.0-2.el7
  tpm2-tss-devel.x86_64 0:1.4.0-2.el7

Complete!
```

With **Ubuntu**, the concept is quite similar:

```
# sudo apt-get update
# sudo apt-get install tpm2-tools
```

Installation by GitHub Source Tree (CentOS 7 for example)

To deploy **tpm2-tools** on target platform from source code, first you shall install **tpm2-tss**, which is available at https://github.com/tpm2-software/tpm2-tss. The following illustrate steps from scratch:

tpm2-tss

Before you start, it is important to have a look at https://github.com/tpm2-software/tpm2-tss and https://github.com/tpm2-software/tpm2-tss/blob/master/INSTALL.md to acquire base knowledge and to learn all the requirements.

1. First, clone the source from GitHub

```
[lanner@localhost ~]$ git clone https://github.com/tpm2-software/tpm2-tss.git
Cloning into 'tpm2-tss'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 19849 (delta 2), reused 5 (delta 1), pack-reused 19839
Receiving objects: 100% (19849/19849), 18.95 MiB | 484.00 KiB/s, done.
Resolving deltas: 100% (15904/15904), done.
[lanner@localhost ~]$ cd tpm2-tss
[lanner@localhost tpm2-tss]$ git log -1
commit b55b642fc48b5a5cdbbf21e46397e5c8c2ecafcb
Author: Tadeusz Struk <tadeusz.struk@intel.com>
Date:   Thu Jun 27 17:17:21 2019 -0700

    sys: Add missing definition of TPMS_TAGGED_POLICY struc

    For whatever reson we don't have the definition of the
    TPMS_TAGGED_POLICY struct even though its size is used
    to calculate the TPM2_MAX_TAGGED_POLICIES value.

    Signed-off-by: Tadeusz Struk <tadeusz.struk@intel.com>
[lanner@localhost tpm2-tss]$
```

Note

The git log above shows the commit hash id as my writing time. If you should encounter any error while building, you can checkout to this state and try again.

2. Second, **bootstrap** shall be run in advance:

```
[lanner@localhost tpm2-tss]$ ./bootstrap
Generating file lists: src_vars.mk
aclocal: installing 'm4/ax_ac_append_to_file.m4' from '/usr/share/aclocal/ax_ac_append_to_file.m4'
aclocal: installing 'm4/ax_ac_print_to_file.m4' from '/usr/share/aclocal/ax_ac_print_to_file.m4'
aclocal: installing 'm4/ax_add_am_macro_static.m4' from '/usr/share/aclocal/ax_add_am_macro_static.m4'
aclocal: installing 'm4/ax_add_fortify_source.m4' from '/usr/share/aclocal/ax_add_fortify_source.m4'
aclocal: installing 'm4/ax_am_macros_static.m4' from '/usr/share/aclocal/ax_am_macros_static.m4'
aclocal: installing 'm4/ax_check_compile_flag.m4' from '/usr/share/aclocal/ax_check_compile_flag.m4'
aclocal: installing 'm4/ax_check_enable_debug.m4' from '/usr/share/aclocal/ax_check_enable_debug.m4'
aclocal: installing 'm4/ax_check_link_flag.m4' from '/usr/share/aclocal/ax_check_link_flag.m4'
aclocal: installing 'm4/ax_code_coverage.m4' from '/usr/share/aclocal/ax_code_coverage.m4'
aclocal: installing 'm4/ax_file_escapes.m4' from '/usr/share/aclocal/ax_file_escapes.m4'
aclocal: installing 'm4/ax_is_release.m4' from '/usr/share/aclocal/ax_is_release.m4'
aclocal: installing 'm4/ax_normalize_path.m4' from '/usr/share/aclocal/ax_normalize_path.m4'
aclocal: installing 'm4/ax_prog_doxygen.m4' from '/usr/share/aclocal/ax_prog_doxygen.m4'
aclocal: installing 'm4/ax_valgrind_check.m4' from '/usr/share/aclocal/ax_valgrind_check.m4'
aclocal: installing 'm4/libtool.m4' from '/usr/share/aclocal/libtool.m4'
aclocal: installing 'm4/ltoptions.m4' from '/usr/share/aclocal/ltoptions.m4'
aclocal: installing 'm4/ltsugar.m4' from '/usr/share/aclocal/ltsugar.m4'
aclocal: installing 'm4/ltversion.m4' from '/usr/share/aclocal/ltversion.m4'
aclocal: installing 'm4/lt~obsolete.m4' from '/usr/share/aclocal/lt~obsolete.m4'
aclocal: installing 'm4/pkg.m4' from '/usr/share/aclocal/pkg.m4'
libtoolize: putting auxiliary files in `.'.
libtoolize: linking file `./ltmain.sh'
configure.ac:23: installing './config.guess'
configure.ac:23: installing './config.sub'
configure.ac:12: installing './install-sh'
configure.ac:12: installing './missing'
Makefile.am: installing './depcomp'
parallel-tests: installing './test-driver'
[lanner@localhost tpm2-tss]$
```

Once **bootstrap** is run without an error, the next step is to execute **configure**. Although there are several arguments over the necessity of setting up **intent setting**, the example here shows only how to enable physical tpm device and test function.


Note

–disbale-doxygen-man will be applied to the following experiment due to doxygen compatible issue here.


Note

You may encounter errors in configuration phase as below:
configure: error: TPM device provided does not exist or is not writable

You can fix it by changing **mode of /dev/tpm0**:
`# sudo chmod go+rw /dev/tpm0`

```
[lanner@localhost tpm2-tss]$ ./configure --disable-doxygen-man --with-udevrulesdir=/etc/udev/rules.d
--with-ptpm=/dev/tpm0 --with-ptpmtests="mandatory,optional" --enable-unit --enable-integration
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
...
<snip>
config.status: executing depfiles commands
config.status: executing libtool commands

    tpm2-tss 2.3.0-dev
    esapi:             yes
    tctidefaultmodule:  libtss2-tcti-default.so
    tctidefaultconfig:
    unit:              yes
    fuzzing:           none
    debug:             info
    maxloglevel:       trace
    doxygen:           1
    tcti-device-async:  no
    tcti-partial-read:  no
    crypto backend:    ossl

[lanner@localhost tpm2-tss]$
```

3. Let's build it:

```
[lanner@localhost tpm2-tss]$ make
git.mk: Generating .gitignore
make  all-am
make[1]: Entering directory `/home/lanner/tpm2-tss'
 cd . && /bin/sh /home/lanner/tpm2-tss/missing automake-1.13 --foreign Makefile
 cd . && /bin/sh ./config.status Makefile depfiles
config.status: creating Makefile
config.status: executing depfiles commands
make[1]: Leaving directory `/home/lanner/tpm2-tss'
make[1]: Entering directory `/home/lanner/tpm2-tss'
  CC       src/tss2-mu/base-types.lo
  CC       src/tss2-mu/tpm2b-types.lo
<snip>
...
<snip>
  GEN      man/man7/tss2-tcti-device.7
  GEN      man/man7/tss2-tcti-mssim.7
make[1]: Leaving directory `/home/lanner/tpm2-tss'
[lanner@localhost tpm2-tss]$
```

tpm2-tss includes codes for unit and integration test.

Run these test by:

```
[lanner@localhost tpm2-tss]$ make check
make  test/integration/libtest_utils.la   test/unit/CommonPreparePrologue test/unit/CopyCommandHeader
test/unit/io
<snip>
...
<snip>
PASS: test/integration/esys-auto-session-flags.int
make[3]: Entering directory `/home/lanner/tpm2-tss'
make  all-am
make[4]: Entering directory `/home/lanner/tpm2-tss'
make[4]: Leaving directory `/home/lanner/tpm2-tss'
make[3]: Leaving directory `/home/lanner/tpm2-tss'
============================================================================
Testsuite summary for tpm2-tss 2.3.0-dev
============================================================================
# TOTAL: 104
# PASS:  91
# SKIP:  13
# XFAIL: 0
# FAIL:  0
# XPASS: 0
# ERROR: 0
============================================================================
make[2]: Leaving directory `/home/lanner/tpm2-tss'
make[1]: Leaving directory `/home/lanner/tpm2-tss'
[lanner@localhost tpm2-tss]$
```

Install it by `# sudo make install`.

`tpm2-tss libtss2-*.a` and `;ibtss2-*.so` are installed to `/usr/local/lib`.

tpm2-abrmd

Take a look at https://github.com/tpm2-software/tpm2-abrmd and https://github.com/tpm2-software/tpm2-abrmd/blob/master/INSTALL.md before you start to build from source.

1. First, clone the source code and record your commit id:

```
[lanner@localhost ~]$ git clone https://github.com/tpm2-software/tpm2-abrmd.git
Cloning into 'tpm2-abrmd'...
remote: Enumerating objects: 75, done.
remote: Counting objects: 100% (75/75), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 6736 (delta 15), reused 21 (delta 8), pack-reused 6661
Receiving objects: 100% (6736/6736), 2.02 MiB | 672.00 KiB/s, done.
Resolving deltas: 100% (5224/5224), done.
[lanner@localhost ~]$ cd tpm2-abrmd
[lanner@localhost tpm2-abrmd]$ git log -1
commit b41fbe23089b8701d229db1988a2811315288dfc
Author: Philip Tricca <philip.b.tricca@intel.com>
Date:   Fri Jun 7 16:47:12 2019 -0700

    util: remove use of objid

    This was originally used to track / trace objects moving through the
    daemon. These are largely useless now and tracing / debugging should be
    done with gdb and the like.

    Signed-off-by: Philip Tricca <philip.b.tricca@intel.com>
[lanner@localhost tpm2-abrmd]$
```

Proceed with the same steps previously done in **tpm2-tss**, bootstrap and configure it. Since we will work with real tpm2 chip instead of a simulation, add **–enable-test-hwtpm**. You may encounter errors as below while configuring:

```
configure: error: Package requirements (tss2-sys >= 2.0.0) were not met:

No package 'tss2-sys' found
```

Export correct reference path (**tpm2-tss** is installed into **/usr/local/lib** before you configure it.) The command shall be

```
[lanner@localhost tpm2-abrmd]$ export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig ;./configure
--enable-unit --enable-integration --enable-test-hwtpm
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
<snip>
...
<snip>
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating dist/tss2-tcti-tabrmd.pc
config.status: creating dist/tpm2-abrmd.service
config.status: creating dist/tpm2-abrmd.preset
config.status: executing libtool commands
config.status: executing depfiles commands
[lanner@localhost tpm2-abrmd]$
```

2. Install it and start **tpm2-abrmd** service:

```
[lanner@localhost tpm2-abrmd]$ sudo make install
[sudo] password for lanner:
make  install-am
make[1]: Entering directory `/home/lanner/tpm2-abrmd'
make[2]: Entering directory `/home/lanner/tpm2-abrmd'
 /usr/bin/mkdir -p '/usr/local/lib'
...
<snip>
 /usr/bin/mkdir -p '/usr/local/lib/systemd/system'
 /usr/bin/install -c -m 644 dist/tpm2-abrmd.service '/usr/local/lib/systemd/system'
make[2]: Leaving directory `/home/lanner/tpm2-abrmd'
make[1]: Leaving directory `/home/lanner/tpm2-abrmd'
[lanner@localhost tpm2-abrmd]$ sudo systemctl restart tpm2-abrmd.service
[lanner@localhost tpm2-abrmd]$ sudo systemctl status tpm2-abrmd.service
● tpm2-abrmd.service - TPM2 Access Broker and Resource Management Daemon
   Loaded: loaded (/usr/local/lib/systemd/system/tpm2-abrmd.service; disabled; vendor preset: disabled)
   Active: activating (auto-restart) since Fri 2019-07-12 18:47:35 CST; 3s ago
  Process: 20537 ExecStart=/usr/local/sbin/tpm2-abrmd (code=exited, status=0/SUCCESS)
 Main PID: 20537 (code=exited, status=0/SUCCESS)

Jul 12 18:47:35 localhost.localdomain systemd[1]: Started TPM2 Access Broker ...
Hint: Some lines were ellipsized, use -l to show in full.
[lanner@localhost tpm2-abrmd]$
```

tpm2-tools

Read the **README** and **INSTALL** sections in https://github.com/tpm2-software/tpm2-tools.

Proceed with the similar steps to clone, bootstrap and configure in **tpm2-tss** and **tpm2-abrmd**:

```
[lanner@localhost ~]$ git clone https://github.com/tpm2-software/tpm2-tools.git
Cloning into 'tpm2-tools'...
remote: Enumerating objects: 53, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 17800 (delta 17), reused 27 (delta 12), pack-reused 17747
Receiving objects: 100% (17800/17800), 6.24 MiB | 671.00 KiB/s, done.
Resolving deltas: 100% (14167/14167), done.
[lanner@localhost ~]$ cd tpm2-tools
[lanner@localhost tpm2-tools]$ git log -1
commit 350fae1f410e95b5f9a21059e7c6d1e279857801
Author: William Roberts <william.c.roberts@intel.com>
Date:   Thu Jul 11 14:32:59 2019 -0500

    tpm2_checkquote: remove stdout outputs

    They didn't make any sense or offer any real value.

    Progresses: #1042

    Signed-off-by: William Roberts <william.c.roberts@intel.com>
[lanner@localhost tpm2-tools]$ ./bootstrap
Generating file lists: src_vars.mk
...
<snip>
Makefile.am: installing './depcomp'
parallel-tests: installing './test-driver'
[lanner@localhost tpm2-tools]$ ./configure --disable-hardening
checking whether to enable debugging... info
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler.
...
<snip>
config.status: lib/config.h is unchanged
config.status: executing libtool commands
config.status: executing depfiles commands

    tpm2-tools    3.0.2-1303-g350fae1
    Unit tests:   no

[lanner@localhost tpm2-tools]$ make
  CC       lib/lib_libcommon_a-files.o
  CC       lib/lib_libcommon_a-log.o
....
<snip>
  CCLD     tools/tpm2_unseal
  CC       tools/tpm2_verifysignature.o
  CCLD     tools/tpm2_verifysignature
[lanner@localhost tpm2-tools]$
```

Congratulations! **tpm2_tools** has been installed into **/usr/local/bin**. Run any command to see if it works or not.

Here, I will run **tpm2_pcrlist** to dump values:

```
[lanner@localhost tpm2-tools]$ tpm2_pcrlist
sha1:
  0 : 0xC4C7206F7469620F071E7B8596443C1CB7C02C2A
  1 : 0xC20332C8631D42B6A1779A6C1A043ADB6E35E526
  2 : 0x4039D466FE90E64473F912725CAEE4830E25441C
  3 : 0xB2A83B0EBF2F8374299A5B2BDFC31EA955AD7236
  4 : 0xB2A83B0EBF2F8374299A5B2BDFC31EA955AD7236
  5 : 0xB2A83B0EBF2F8374299A5B2BDFC31EA955AD7236
  6 : 0xB2A83B0EBF2F8374299A5B2BDFC31EA955AD7236
  7 : 0x4037336FA7BC0EABE3778FCFFF5FCD0EE6ADCDE3
  8 : 0x0000000000000000000000000000000000000000
  9 : 0x0000000000000000000000000000000000000000
  10: 0xD3747CCBAAA910A4395EE943D7B7771EA9943FB5
  11: 0x0000000000000000000000000000000000000000
  12: 0x0000000000000000000000000000000000000000
  13: 0x0000000000000000000000000000000000000000
  14: 0x0000000000000000000000000000000000000000
  15: 0x0000000000000000000000000000000000000000
  16: 0xB587C8CE13DFD50E056B409D3FF7F40A48032E6D
  17: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  18: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  19: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  20: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  21: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  22: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  23: 0x0000000000000000000000000000000000000000
sha256:
  0 : 0x9B7E76DE6F71D564D672A84AEC658BBA68D81B9884AA95AAD96C759868692BDD
  1 : 0x035D6E443AAA21AD2614942F355917C7FF17B19234AA5434B6B8C46FB980B975
  2 : 0x05C3B7FAAC93CE48D79F79FC77B62D408440FDAE8F50D154F9DEF705B941E256
  3 : 0x3D458CFE55CC03EA1F443F1562BEEC8DF51C75E14A9FCF9A7234A13F198E7969
  4 : 0x3D458CFE55CC03EA1F443F1562BEEC8DF51C75E14A9FCF9A7234A13F198E7969
  5 : 0x3D458CFE55CC03EA1F443F1562BEEC8DF51C75E14A9FCF9A7234A13F198E7969
  6 : 0x3D458CFE55CC03EA1F443F1562BEEC8DF51C75E14A9FCF9A7234A13F198E7969
  7 : 0xB5710BF57D25623E4019027DA116821FA99F5C81E9E38B87671CC574F9281439
  8 : 0x0000000000000000000000000000000000000000000000000000000000000000
  9 : 0x0000000000000000000000000000000000000000000000000000000000000000
  10: 0xFA94DF975D0FA8E6EFF1C63CB2EC3C749AA3A16CAB3CF0EE6A07EC0B17838895
  11: 0x0000000000000000000000000000000000000000000000000000000000000000
  12: 0x0000000000000000000000000000000000000000000000000000000000000000
  13: 0x0000000000000000000000000000000000000000000000000000000000000000
  14: 0x0000000000000000000000000000000000000000000000000000000000000000
  15: 0x0000000000000000000000000000000000000000000000000000000000000000
  16: 0x0A2FD798F6938F88B8275F5D34C109C3D83FEAA990BACCB8378C281244495C1F
  17: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  18: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  19: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  20: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  21: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  22: 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  23: 0x0000000000000000000000000000000000000000000000000000000000000000
[lanner@localhost tpm2-tools]$
```

Check tpm2-tools Functions (CentOS 7 for example)

1. First, check if tpm2-abrmd service is running by **"systemctl status tpm2-abrmd"**.
   If this service is inactive or dead, you will get the below message:

```
[root@localhost ~]# systemctl status tpm2-abrmd | grep Active
   Active: inactive (dead)
```

2. Start the service by **"systemctl start tpm2-abrmd"** and check again, it shall become active:

```
[root@localhost ~]# systemctl status tpm2-abrmd | grep Active
   Active: active (running) since Thu 2019-07-11 02:40:38 CST; 1h 11
```

3. **tpm2-tools** includes a variety of tools for hashing, NVRAM read/write, display PCR, etc. To check the available functions provided, type "tpm2" and press [TAB]:

```
[root@localhost ~]# tpm2
tpm2-abrmd              tpm2_hash               tpm2_pcrlist
tpm2_activatecredential  tpm2_hmac               tpm2_quote
tpm2_certify            tpm2_listpersistent     tpm2_rc_decode
tpm2_create             tpm2_load               tpm2_readpublic
tpm2_createpolicy        tpm2_loadexternal       tpm2_rsadecrypt
tpm2_createprimary       tpm2_makecredential     tpm2_rsaencrypt
tpm2_dictionarylockout   tpm2_nvdefine           tpm2_send
tpm2_encryptdecrypt      tpm2_nvlist             tpm2_sign
tpm2_evictcontrol        tpm2_nvread             tpm2_startup
tpm2_getcap             tpm2_nvreadlock         tpm2_takeownership
tpm2_getmanufec          tpm2_nvrelease          tpm2_unseal
tpm2_getpubak            tpm2_nvwrite            tpm2_verifysignature
tpm2_getpubek            tpm2_pcrevent
tpm2_getrandom           tpm2_pcrextend
[root@localhost ~]# tpm2
```

For detailed instructions on usage of these functions, use "man page" to see how to work, or access https://github.com/tpm2-software/tpm2-tools/tree/master/man.

Please note, since **tpm2-tools** version may differ by the distribution version regarding GitHub master tree and function support, it is better to build from source for up-to-date features.