TPM 2.0 Linux sysfs interface

LPC 2019: System Boot and Security MC

Piotr Król

3MDEB

About me

Piotr Król
Founder & Embedded Systems Consultant

- open-source firmware
- platform security
- trusted computing

@pietrushnic
piotr.krol@3mdeb.com
linkedin.com/in/krolpiotr
facebook.com/piotr.krol.756859

Problem statement

- trivia: TPM1.2 and 2.0 are not compatible
- but not all users and developers know that
- there are some tools/scripts that rely on TPM1.2 sysfs structure (e.g. TPM detection in QubesOS)
- people sometimes want to update TPM to 2.0 and use the same software
- there are TPM models that can work as 1.2 and 2.0
- tpm2-software moves so fast that distros are not up to date

Problem statement

2. Check TPM version

Version 1.2 TPMs are currently supported. Read the TPM device ID file to discover the TPM version:

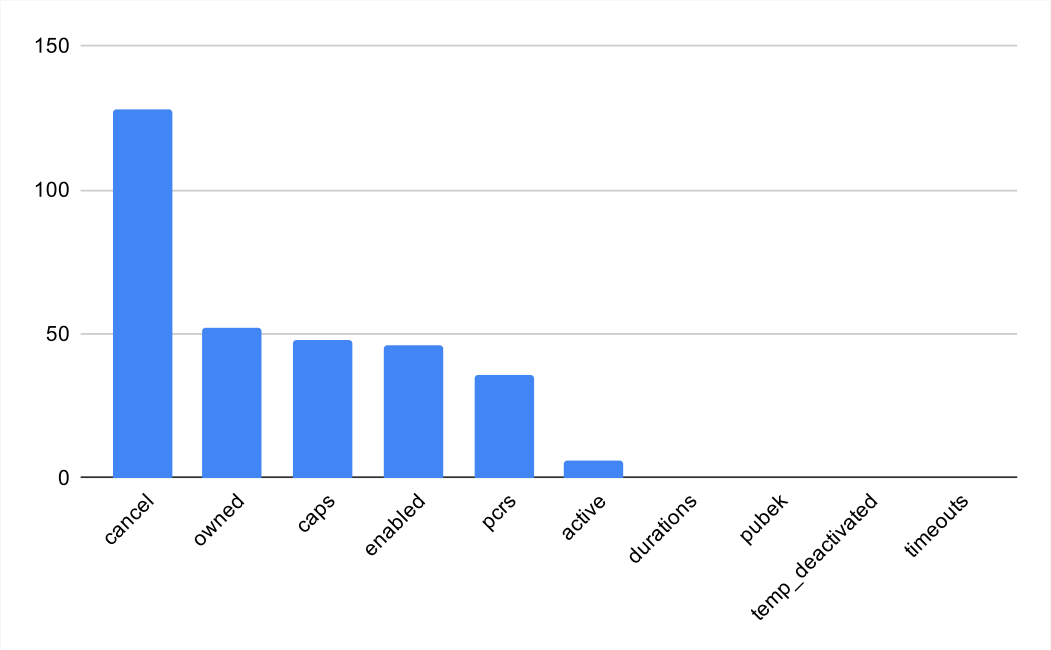
```
# cat /sys/class/tpm/tpm0/device/id
```

The contents of the `id` file vary for supported version 1.2 TPMs. It is simplest to check that the file does *not* contain the known string for unsupported version 2.0 TPMs, `MSFT0101`. Almost any other non-zero, non-error output from reading the `id` file indicates a supported version 1.2 TPM.

Support for version 2.0 TPMs identified with the `MSFT0101` string will be added in a future Container Linux release.

- Software has no idea if it deals with TPM 1.2 or 2.0
- What should be the official way to figure out with what TPM module/s?
- Even if we do not support sysfs for TPM 2.0 we should have consistent method to get information about TPMs we have available
- Similar issue will touch the bootloaders

Whose affected



Graph shows Github stats of various sysfs uses (Google, CoreOS, LTP, ...)

Whose affected

- QubesOS AEM and HCL scripts, which detect TPM
- LTP IMA tests
- CoreOS
- Google ChromeOS vboot-utils
- fwupd
- `ima-evm-utils` - tools for producing and verifying signatures
- USRP by Ettus Research

https://github.com/QubesOS/qubes-core-admin/blob/master/qvm-tools/qubes-hcl-report#L114

https://codesearch.debian.net/search?q=%2Fsys%2Fclass%2Ftpm%2Ftpm0

https://github.com/fwupd/fwupd/blob/master/plugins/uefi/fu-uefi-pcrs.c#L190

https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/refs/heads/master/utility/tpm-dad-lock#10

https://github.com/EttusResearch/uhd/blob/master/mpm/python/usrp_mpm/bist.py#L239

sysfs ABI

- https://www.kernel.org/doc/Documentation/ABI/stable/sysfs-class-tpm
- Jarkko Sakkinen patches for required attributes: https://patchwork.kernel.org/patch/5274701/
- TPM 2.0 support

sysfs attributes problems

This patch set enables TPM2 protocol and provides drivers for FIFO and CRB interfaces. This patch set does not export any sysfs attributes for TPM 2.0 because existing sysfs attributes have three non-trivial issues:

- They are associated with the platform device instead of character device.
- They are are not trivial key-value pairs but contain text that is not easily parsed by a computer.
- Raciness as described in http://kroah.com/log/blog/2013/06/26/how-to-create-a-sysfs-file-correctly/

https://lwn.net/Articles/624241/

TPM1.2 support

- At the time of writing this paper the Linux kernel supported TPM 1.2 functionalities in sysfs. To these functionalities we include:

```
$ ls /sys/devices/pnp0/00:04/tpm/tpm0
active  caps  device     enabled  pcrs   ppi    subsystem         timeouts
cancel  dev   durations  owned    power  pubek  temp_deactivated  uevent
$ ls /sys/devices/pnp0/00:04/tpm/tpm0/ppi
request  response  tcg_operations  transition_action  version  vs_operations
```

TPM2.0 support

```
$ ls /sys/devices/pnp0/00\:04/tpm/tpm0
dev  device  power  ppi  subsystem  uevent
```

Mailing list discussion summary

- PCRs in sysfs rather NAK
- TPM 1.2 vs 2.0 identification - some NAKs, but majority seem to need that

Proposals so far

- Jarkko
  - `/sys/class/tpm/tpm0/protocol_major`
- Mimi and Petr
  - `/sys/class/tpm/tpm0/version`
- Tadeusz
  - `new /proc/tpminfo entry`
- securityfs
- ioctl
- user space C code

Open questions

- is there a method for TPM presence check that will be compatible between TPM versions?
- is there a way to unify interface between kernel, user space and TPM?

Platform Security Summit 2019

Oct 1-3, 2019 · Redmond, WA

"Give me a place to stand on,
and I will move the earth."
—Archimedes

PSEC 2019 brings together security architects, researchers and developers from the ecosystems of hyperscalers, service operators, product vendors, academia and open-source.

Q&A