Sources

Linux TPM PCR Registry

TPM PCRs are a scarce resource, there are only 24 of them in typical standards compliant TPMs. According to the TCG PC Client Specific Platform Firmware Profile Specification | Trusted Computing Group the OS can make use of PCRs 8…15. This document lists which component is using which PCR on a Linux platform in order to minimize conflicts.

PCRs owned by the firmware, i.e. PCRs 0–7 are described here just for convenience. The authoriative description is in the TCG document.

How other operating systems — in particular Windows — use PCRs, is out of scope of this document.

This document is informational in nature: it just describes what is, it is not intended to formally declare "ownership" of a specific PCR, but simply is supposed to reflect which PCR assignments are common in the Linux ecosystems. That said, co-opting PCR usage will likely create problems down the line, in particular if measurement logs are maintained separately. (To be more explicit: on `systemd` systems the warranty is voided if you write to the PCRs it also uses, as per the list below.)

PCR measurements most commonly serve two distinct purposes:

- To implement access policy on TPM sealed objects: policy can dictate that unsealing of such objects shall only be allowed if some PCRs are in a specific literal state, or in any state for which a signature by a specific key pair can be provided. For this it is essential that PCRs only contain measurements for a clearly defined set of objects, that typically is known in advance so that the PCR value can be pre-calculated (hence this is in a way a *forward*-looking use)
- To permit reasoning about the boot process and runtime *so far*, for example for the purpose of remote attestation. In this case it is not that important what objects are measured as long as a record is kept in a measurement log about what it was. The PCRs are in this case used to validate that log (hence this is in a way a *backward*-looking use)

In both cases it is important that data measured into the PCRs is carefully chosen. PCRs that shall be useful for policy binding should only cover data objects known in advance, and thus not contain runtime data that cannot be pre-calculated in advance. PCRs that shall be useful for backward-looking validation should only cover objects that are also written to the appropriate log for the PCR.

PCR#	Used by	From Location	Measured Objects	Log	
0	`Firmware` 💻	UEFI Boot Component	Core system firmware executable code	UEFI TPM event log	n/a
1	`Firmware` 💻	UEFI Boot Component	Core system firmware data/host platform configuration; typically contains serial and model numbers	UEFI TPM event log	n/a
2	`Firmware` 💻	UEFI Boot Component	Extended or pluggable executable code; includes option ROMs on pluggable hardware	UEFI TPM event log	n/a
3	`Firmware` 💻	UEFI Boot Component	Extended or pluggable firmware data; includes information about pluggable hardware	UEFI TPM event log	n/a
4	`Firmware` 💻	UEFI Boot Component	Boot loader and additional drivers; binaries and extensions loaded by the boot loader	UEFI TPM event log	n/a

5	`Firmware` 💻	UEFI Boot Component	GPT/Partition table	UEFI TPM event log	n/a
7	`Firmware` 💻	UEFI Boot Component	SecureBoot state	UEFI TPM event log	n/a
8	`grub` 🐵	UEFI Boot Component	Commands and kernel command line	UEFI TPM event log	n/a
9	`grub` 🐵	UEFI Boot Component	All files read (including kernel image)	UEFI TPM event log	n/a
	Linux kernel 🌰	Kernel	All passed initrds (when the new `LOAD_FILE2` initrd protocol is used)	UEFI TPM event log	n/a
10	IMA 📐	Kernel	Protection of the IMA measurement log	IMA event log	n/a
11	`systemd-stub` 🚀	UEFI Stub	All components of unified kernel images (UKIs)	UEFI TPM event log	in E `Stu`
	`systemd-pcrphase` 🚀	Userspace	Boot phase strings, indicating various milestones of	Journal (for now)	n/a

			the boot process		
12	`systemd-stub` 🚀	UEFI Stub	Kernel command line, system credentials and system configuration images	UEFI TPM event log	in E `Stu`
13	`systemd-stub` 🚀	UEFI Stub	All system extension images for the initrd	UEFI TPM event log	in E `Stu`
14	`shim` 🔑	UEFI Boot Component	“MOK” certificates and hashes	UEFI TPM event log	n/a
15	`systemd-cryptsetup@.service` 🚀	Userspace	Root file system volume encryption key	Journal (for now)	n/a
	`systemd-pcrmachine.service` 🚀	Userspace	Machine ID (`/etc/machine-id`)	Journal (for now)	n/a
	`systemd-pcrfs@.service` 🚀	Userspace	File system mount point, UUID, label, partition UUID label of root file system and `/var/`	Journal (for now)	n/a

PCR 0 changes on firmware updates; PCR 1 changes on basic hardware/CPU/RAM replacements.

PCR 4 changes on boot loader updates. The shim project will measure the PE binary it chain loads into this PCR. If the Linux kernel is invoked as UEFI PE binary, it is measured here, too.

systemd-stub measures system extension images read from the ESP here too (see systemd-sysext and Extension Images).

PCR 5 changes when partitions are added, modified, or removed.

PCR 7 changes when UEFI SecureBoot mode is enabled/disabled, or firmware certificates (PK, KEK, db, dbx, …) are updated. The shim project will measure most of its (non-MOK) certificates and SBAT data into this PCR.

PCR 11 and 15 as shown in the list above are used by multiple components of systemd. These are not conflicting uses; the involved components are properly ordered to cooperatively guarantee predictable behaviour.

systemd-stub measures the ELF kernel image, embedded initrd and other payload of the PE image into PCR 11. Unlike PCR 4 (where the same data should be measured too), those values should be easy to pre-calculate, as they only reflect the static parts of the PE binary. systemd-pcrphase.service measures boot phase strings into this PCR at various milestones of the boot process. Use PCR 11 to bind TPM policies to a specific kernel image, possibly with an embedded initrd, and to a specific boot phase.

systemd-boot measures the kernel command line into PCR 12. systemd-stub measures any manually specified kernel command line (i.e. a kernel command line that overrides the one embedded in the UKI) and loaded credentials into this PCR. This means that if `systemd-boot` and `systemd-stub` are used together, the command line might be measured twice.

systemd-stub measures any Extension Images it passes to the booted kernel into PCR 13.

systemd-cryptsetup optionally measures the volume key of activated LUKS volumes into this PCR. systemd-pcrmachine.service measures the machine-id into this PCR. systemd-pcrfs@.service measures mount points, file system UUIDs, labels, partion UUIDs of the root and `/var/` filesystems into this PCR.

Sources

- systemd-cryptenroll(1)
- TCG PC Client Specific Platform Firmware Profile Specification
- shim's README.tpm
- Measured Boot - GNU GRUB Manual 2.06
- Integrity Measurement Architecture (IMA)
- edk2-TrustedBootChain/4_Other_Trusted_Boot_Chains.md
- Trusted Platform Module - ArchWiki