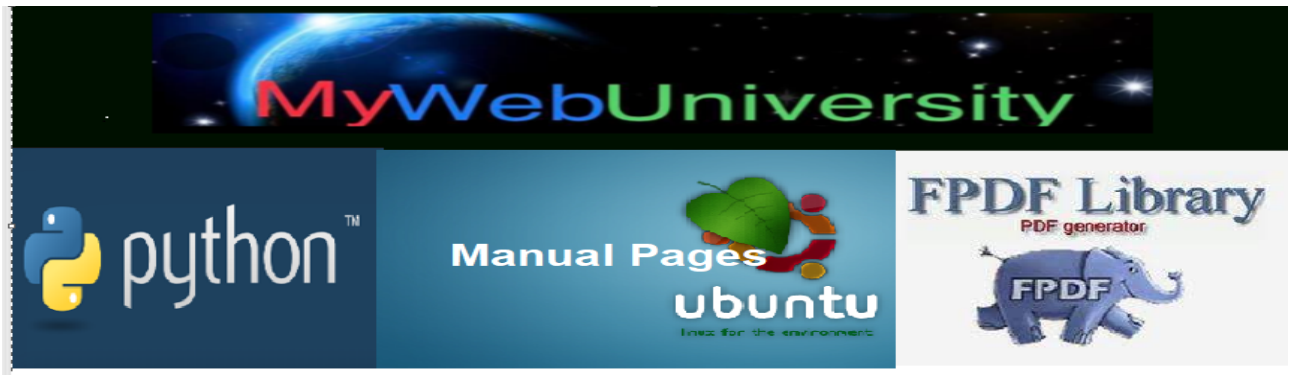

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'systemd-measure.1'

$ man systemd-measure.1

SYSTEMD-MEASURE(1) systemd-measure SYSTEMD-MEASURE(1)

NAME

systemd-measure - Pre-calculate and sign expected TPM2 PCR 11 values for booted unified kernel images

SYNOPSIS

/usr/lib/systemd/systemd-measure [OPTIONS...]

DESCRIPTION

Note: this command is experimental for now. While it is likely to become a regular component of systemd, it might still change in behaviour and interface.

systemd-measure is a tool that may be used to pre-calculate and sign the expected TPM2 PCR 11 values that should be seen when a Linux UAPI.5 Unified Kernel Image (UKI)[1] based on systemd-stub(7) is booted up. It accepts paths to the ELF kernel image file, initrd image file, devicetree file, kernel command line file, os-release(5) file, boot splash file, and TPM2 PCR PEM public key file that make up the unified kernel image, and determines the PCR values expected to be in place after booting the image. Calculation starts with a zero-initialized PCR 11, and is executed in a fashion compatible with what systemd-stub does at boot. The result may optionally be signed cryptographically, to allow TPM2 policies that can only be unlocked if a certain set of kernels is booted, for which such a PCR signature can be provided.

It usually does not make sense to call this tool directly when constructing a UKI. Instead, ukify(1) should be used; it will invoke systemd-measure and take care of embedding the resulting measurements into the UKI.

COMMANDS

The following commands are understood:

status

This is the default command if none is specified. This queries the local system's TPM2 PCR 11 values and displays them. The data is written in a similar format as the calculate command below, and may be used to quickly compare expectation with reality. Added in version 252.

calculate

Pre-calculate the expected values seen in PCR register 11 after boot-up of a unified kernel image consisting of the components specified with --linux=, --osrel=, --cmdline=, --initrd=, --ucode=, --splash=, --dtb=, --uname=, --sbat=, --pcrpkey=, --profile=, --dtbauto=, --hwids=, see below. Only --linux= is mandatory. (Alternatively, specify --current to use the current values of PCR register 11 instead.) Added in version 252.

sign

As with the calculate command, pre-calculate the expected value seen in TPM2 PCR register 11 after boot-up of a unified kernel image. Then, cryptographically sign the resulting values with the private/public key pair (RSA) configured via --private-key= and --public-key=. This will write a JSON object to standard output that contains signatures for all specified PCR banks (see the --bank= option below), which may be used to unlock encrypted credentials (see systemd-creds(1)) or LUKS volumes (see systemd-cryptsetup@.service(8)). This allows binding secrets to a set of kernels for which such PCR 11 signatures can be provided.

Note that a TPM2 device must be available for this signing to take place, even though the result is not tied to any TPM2 device or its state.

Added in version 252.

policy-digest

As with the sign command, pre-calculate the expected value seen in TPM2 PCR register 11 after boot-up of a unified kernel image. Then, compute the resulting TPM2 policy and print its digest. This will write a JSON object to standard output that contains the policy digests for all specified PCR banks (see the --bank= option below), so that it may be signed offline, for the cases where the private key is not directly accessible. If --public-key= or --certificate= are specified, the JSON object will also contain the key fingerprint.

Added in version 258.

OPTIONS

The following options are understood:

--linux=PATH, --osrel=PATH, --cmdline=PATH, --initrd=PATH, --ucode=PATH, --splash=PATH, --dtb=PATH, --uname=PATH, --sbat=PATH, --pcrpkey=PATH, --profile=PATH, --dtbauto=PATH, --hwids=PATH

When used with the calculate or sign verb, configures the files to read the unified kernel image components from. Each option corresponds with the equally named section in the unified kernel PE file. The --linux= switch expects the path to the ELF kernel file that the unified PE kernel will wrap. All switches except --linux= are optional. Each option may be used at most once.

Added in version 252.

With the exception of --profile=, --dtbauto= and --hwids=, which have been added in version 257.

--current

When used with the calculate or sign verb, takes the PCR 11 values currently in effect for the system (which should typically reflect the hashes of the currently booted kernel). This can be used in place of --linux= and the other switches listed above.

Added in version 252.

--bank=DIGEST

Controls the PCR banks to pre-calculate the PCR values for ? in case calculate or sign is invoked ?, or the banks to show in the status output. May be used more then once to specify multiple banks. If not specified, defaults to the four banks "sha1", "sha256", "sha384", "sha512".

Note that some operating systems disable support for SHA1-based signatures, in which case this operation will fail. Please consult your OS' documentation for details on how to override the OS security policy around this.

Added in version 252.

--private-key=PATH, --public-key=PATH, --certificate=PATH

These switches take paths to a pair of PEM encoded RSA key files, for use with the sign command.

Note the difference between the --pcrpkey= and --public-key= switches. The former selects the data to include in the ".pcrpkey" PE section of the unified kernel image,

the latter picks the public key of the key pair used to sign the resulting PCR 11 values. The former is the key that the booted system will likely use to lock disk and credential encryption to, the latter is the key used for unlocking such resources again. Hence, typically the same PEM key should be supplied in both cases.

If the --public-key= is not specified but --private-key= is specified, the public key is automatically derived from the private key.

--certificate= can be used to specify an X.509 certificate as an alternative to --public-key= since v256.

Added in version 252.

--private-key=PATH/URI, --private-key-source=TYPE[:NAME], --certificate=PATH/URI, --certificate-source=TYPE[:NAME]

As an alternative to --public-key= for the sign command, these switches can be used to sign with an hardware token. The private key option can take a path or a URI that will be passed to the OpenSSL engine or provider, as specified by --private-key-source= as a type:name tuple, such as engine:pkcs11. The specified OpenSSL signing engine or provider will be used to sign.

The --certificate= option also takes a path or a URI that will be passed to the OpenSSL provider, as specified by --certificate-source= as a "type:name" tuple, such as "provider:pkcs11". Note that unlike --private-key-source= this option only supports providers and not engines.

Added in version 256.

--tpm2-device=PATH

Controls which TPM2 device to use. Expects a device node path referring to the TPM2 chip (e.g. /dev/tpmrm0). Alternatively the special value "auto" may be specified, in order to automatically determine the device node of a suitable TPM2 device (of which there must be exactly one). The special value "list" may be used to enumerate all suitable TPM2 devices currently discovered.

Added in version 252.

--phase=PHASE

Controls which boot phases to calculate expected PCR 11 values for. This takes a series of colon-separated strings that encode boot "paths" for entering a specific phase of the boot process. Each of the specified strings is measured by the systemd-pcrphase-initrd.service, systemd-pcrphase-sysinit.service, and systemd-

pcrphase.service(8) into PCR 11 during different milestones of the boot process. This switch may be specified multiple times to calculate PCR values for multiple boot phases at once. If not used defaults to "enter-initrd", "enter-initrd:leave-initrd", "enter-initrd:leave-initrd:sysinit", "enter-initrd:leave-initrd:sysinit:ready", i.e. calculates expected PCR values for the boot phase in the initrd, during early boot, during later boot, and during system runtime, but excluding the phases before the initrd or when shutting down. This setting is honoured both by calculate and sign. When used with the latter it is particularly useful for generating PCR signatures that can only be used for unlocking resources during specific parts of the boot process.

For further details about PCR boot phases, see systemd-pcrphase.service(8).

Added in version 252.

--append=PATH

When generating a PCR JSON signature (via the sign command), combine it with a previously generated PCR JSON signature, and output it as one. The specified path must refer to a regular file that contains a valid JSON PCR signature object. The specified file is not modified. It will be read first, then the newly generated signature appended to it, and the resulting object is written to standard output. Use this to generate a single JSON object consisting from signatures made with a number of signing keys (for example, to have one key per boot phase). The command will suppress duplicates: if a specific signature is already included in a JSON signature object it is not added a second time.

Added in version 253.

--json=MODE

Shows output formatted as JSON. Expects one of "short" (for the shortest possible output without any redundant whitespace or line breaks), "pretty" (for a pretty version of the same, with indentation and line breaks) or "off" (to turn off JSON output, the default).

--no-pager

Do not pipe output into a pager.

-h, --help

Print a short help text and exit.

--version

Print a short version string and exit.

EXAMPLES

Example 1. Generate a unified kernel image, and calculate the expected TPM PCR 11 value

```
$ ukify build \
    --linux=vmlinux \
    --initrd=initrd.cpio \
    --os-release=@os-release.txt \
    --cmdline=@cmdline.txt \
    --splash=splash.bmp \
    --devicetree=devicetree.dtb \
    --measure \
    --output=vmlinux.efi
11:sha1=d775a7b4482450ac77e03ee19bda90bd792d6ec7
11:sha256=bc6170f9ce28eb051ab465cd62be8cf63985276766cf9faf527ffefb66f45651
11:sha384=1cf67dff4757e61e5...7f49ad720be02fd07263e1f93061243aec599d1ee4b4
11:sha512=8e79acd3ddbbc8282...0c3e8ec0c714821032038f525f744960bcd082d937da
```

ukify(1) internally calls systemd-measure. The output with hashes is from systemd-measure.

Example 2. Generate a private/public key pair, a unified kernel image, and a TPM PCR 11 signature for it, and embed the signature and the public key in the image

```
$ openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out tpm2-pcr-private-key.pem
..+.+++++++++.....+.........+.....+.......+....+.....+.+...+.........
$ openssl rsa -pubout -in tpm2-pcr-private-key.pem -out tpm2-pcr-public-key.pem
$ systemd-measure sign \
    --linux=vmlinux \
    --osrel=os-release.txt \
    --cmdline=cmdline.txt \
    --initrd=initrd.cpio \
    --splash=splash.bmp \
    --dtb=devicetree.dtb \
    --pcrpkey=tpm2-pcr-public-key.pem \
    --bank=sha1 \
    --bank=sha256 \
    --private-key=tpm2-pcr-private-key.pem \
    --public-key=tpm2-pcr-public-key.pem >tpm2-pcr-signature.json
$ ukify build \
```

```
    --linux=vmlinux \
    --initrd=initrd.cpio \
    --os-release=@os-release.txt \
    --cmdline=@cmdline.txt \
    --splash=splash.bmp \
    --devicetree=devicetree.dtb \
    --pcr-private-key=tpm2-pcr-private-key.pem \
    --pcr-public-key=tpm2-pcr-public-key.pem \
    --pcr-banks=sha1,sha256 \
    --output=vmlinuz.efi
```

Later on, enroll the signed PCR policy on a LUKS volume:

```
# systemd-cryptenroll --tpm2-device=auto \
    --tpm2-public-key=tpm2-pcr-public-key.pem \
    --tpm2-signature=tpm2-pcr-signature.json \
    /dev/sda5
```

And then unlock the device with the signature:

```
# systemd-cryptsetup attach \
    volume5 /dev/sda5 - \
    tpm2-device=auto,tpm2-signature=/path/to/tpm2-pcr-signature.json
```

Note that when the generated unified kernel image vmlinux.efi is booted, the signature and public key files will be placed at locations systemd-cryptenroll and systemd-cryptsetup will look for anyway, and thus these paths do not actually need to be specified.

Example 3. Introduce a second public key, signing the same kernel PCR measurements, but only for the initrd boot phase

This example extends the previous one, but we now introduce a second signing key that is only used to sign PCR policies restricted to the initrd boot phase. This can be used to lock down root volumes in a way that they can only be unlocked before the transition to the host system. Thus we have two classes of secrets or credentials: one that can be unlocked during the entire runtime, and the other that can only be used in the initrd.

```
$ openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out tpm2-pcr-private-key.pem
.+.....+.+.....+.....+...+...+.....+....+.....+..+.....
$ openssl rsa -pubout -in tpm2-pcr-private-key.pem -out tpm2-pcr-public-key.pem
$ openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out tpm2-pcr-initrd-private-key.pem
```

```
..+.......++........+........+......+........+....+.....+.+..+.........
$ openssl rsa -pubout -in tpm2-pcr-initrd-private-key.pem -out tpm2-pcr-initrd-public-key.pem
$ ukify build \
    --linux=vmlinux-1.2.3 \
    --initrd=initrd.cpio \
    --os-release=@os-release.txt \
    --cmdline=@cmdline.txt \
    --splash=splash.bmp \
    --devicetree=devicetree.dtb \
    --pcr-private-key=tpm2-pcr-private-key.pem \
    --pcr-public-key=tpm2-pcr-public-key.pem \
    --phases=enter-initrd,enter-initrd:leave-initrd,enter-initrd:leave-initrd:sysinit,enter-initrd:leave-initrd:sysinit:ready \
    --pcr-banks=sha1,sha256 \
    --pcr-private-key=tpm2-pcr-initrd-private-key.pem \
    --pcr-public-key=tpm2-pcr-initrd-public-key.pem \
    --phases=enter-initrd \
    --uname=1.2.3 \
    --output=vmlinux-1.2.3.efi
+ /usr/lib/systemd/systemd-measure sign --linux=vmlinux-1.2.3 \
--osrel=os-release.txt --cmdline=cmdline.txt --dtb=devicetree.dtb \
--splash=splash.bmp --initrd=initrd.cpio --bank=sha1 --bank=sha256 \
--private-key=tpm2-pcr-private-key.pem --public-key=tpm2-pcr-public-key.pem \
--phase=enter-initrd --phase=enter-initrd:leave-initrd \
--phase=enter-initrd:leave-initrd:sysinit \
--phase=enter-initrd:leave-initrd:sysinit:ready
+ /usr/lib/systemd/systemd-measure sign --linux=vmlinux-1.2.3 \
--osrel=os-release.txt --cmdline=cmdline.txt --dtb=devicetree.dtb \
--splash=splash.bmp --initrd=initrd.cpio --bank=sha1 --bank=sha256 \
--private-key=tpm2-pcr-initrd-private-key.pem \
--public-key=tpm2-pcr-initrd-public-key.pem \
--phase=enter-initrd
Wrote unsigned vmlinux-1.2.3.efi
```

ukify prints out both invocations of systemd-measure as informative output (the lines

starting with "+"); this allows us to see how systemd-measure is called. It then merges the output of both invocations into the ".pcrsig" section. systemd-measure may also do this merge itself using the --append= option.

Note that in this example the ".pcrpkey" PE section contains the key specified by the first --pcr-private-key= option, covering all boot phases. The ".pcrpkey" section is used in the default policies of systemd-cryptenroll and systemd-creds. To use the stricter policy bound to tpm2-pcr-initrd-public-key.pem, specify --tpm2-public-key= on the command line of those tools.

EXIT STATUS

On success, 0 is returned, a non-zero failure code otherwise.

SEE ALSO

systemd(1), systemd-stub(7), ukify(1), systemd-creds(1), systemd-cryptsetup@.service(8), systemd-pcrphase.service(8)

NOTES

1. UAPI.5 Unified Kernel Image (UKI)

   https://uapi-group.org/specifications/specs/unified_kernel_image/