NAME

zfs-tpm-list — print dataset tzpfms metadata

SYNOPSIS

zfs-tpm-list [**-H**] [**-r**|**-d** *depth*] [**-a**|**-b** *back-end*] [**-u**|**-l**] [*filesystem*|*volume*]...

DESCRIPTION

Lists the following properties on encryption roots:

name	
back-end	the **tzpfms** back-end (e.g. **TPM2** for zfs-tpm2-change-key(8) or **TPM1.X** for zfs-tpm1x-change-key(8)), or "**-**" if none is configured
keystatus	**available** or **unavailable**
coherent	**yes** if either both xyz.nabijaczleweli:tzpfms.backend and xyz.nabijaczleweli:tzpfms.key are present or missing, **no** otherwise

Incoherent datasets require immediate operator attention, with either the appropriate **zfs-tpm*-clear-key** program or **zfs change-key** and **zfs inherit** — if the key becomes unloaded, they will require restoration from back-up. However, this should never occur, unless something went horribly wrong with the dataset properties.

If no datasets are specified, all matching encryption roots are listed — by default, those managed by **tzpfms**.

OPTIONS

-H Scripting mode — remove headers and separate fields by a single tab instead of columnating them with spaces.

-r Recurse into all descendants of specified datasets.

-d *depth* Recurse at most *depth* datasets deep. Default: **0**.

-a List all encryption roots, even ones not managed by **tzpfms**.

-b *back-end*
List only encryption roots with the specified **tzpfms** *back-end*.

-l List only encryption roots whose keys are available.

-y List only encryption roots whose keys are unavailable.

EXAMPLES

```
$ zfs-tpm-list
NAME             BACK-END  KEYSTATUS    COHERENT
tarta-zoot       TPM1.X    available    yes
tarta-zoot/home  TPM2      unavailable  yes

$ zfs-tpm-list -ad0
NAME     BACK-END  KEYSTATUS  COHERENT
filling  -         available  yes

$ zfs-tpm-list -b TPM2
NAME             BACK-END  KEYSTATUS    COHERENT
tarta-zoot/home  TPM2      unavailable  yes

$ zfs-tpm-list -ra tarta-zoot
NAME             BACK-END  KEYSTATUS    COHERENT
tarta-zoot       TPM1.X    available    yes
```

```
tarta-zoot/home  TPM2      unavailable  yes
tarta-zoot/bkp   -         available    yes
tarta-zoot/vm    -         available    yes

$ zfs-tpm-list -al
NAME             BACK-END  KEYSTATUS  COHERENT
filling          -         available  yes
tarta-zoot       TPM1.X    available  yes
tarta-zoot/bkp   -         available  yes
tarta-zoot/vm    -         available  yes
```

SPECIAL THANKS

To all who support further development, in particular:

- ThePhD
- Embark Studios
- Jasper Bekkers

REPORTING BUGS

https://todo.sr.ht/~nabijaczleweli/tzpfms

`~nabijaczleweli/tzpfms@lists.sr.ht`, archived at **https://lists.sr.ht/~nabijaczleweli/tzpfms**.

SEE ALSO

https://git.sr.ht/~nabijaczleweli/tzpfms

NAME

zfs-tpm1x-change-key — change ZFS dataset key to one stored on the TPM

SYNOPSIS

zfs-tpm1x-change-key [**-b** *backup-file*] *dataset*

DESCRIPTION

To normalise the *dataset*, **zfs-tpm1x-change-key** will open its encryption root in its stead. **zfs-tpm1x-change-key** will *never* create or destroy encryption roots; use zfs-change-key(8) for that.

First, a connection is made to the TPM, which *must* be TPM-1.X-compatible.

If *dataset* was previously encrypted with **tzpfms** and the **TPM1.X** back-end was used, the metadata will be silently cleared. Otherwise, or in case of an error, data required for manual intervention will be printed to the standard error stream.

Next, a new wrapping key is generated on the TPM, optionally backed up (see **OPTIONS**), and sealed on the TPM; the user is prompted for an optional passphrase to protect the key with, and for the SRK passphrase, set when taking ownership, if not "well-known" (all zeroes).

The following properties are set on *dataset*:

- xyz.nabijaczleweli:tzpfms.backend=**TPM1.X**
- xyz.nabijaczleweli:tzpfms.key=*parent-key-blob*:*sealed-object-blob*

tzpfms.backend identifies this dataset for work with **TPM1.X**-back-ended **tzpfms** tools (namely zfs-tpm1x-change-key(8), zfs-tpm1x-load-key(8), and zfs-tpm1x-clear-key(8)).

tzpfms.key is a colon-separated pair of hexadecimal-string (i.e. "4F7730" for "Ow0") blobs; the first one represents the RSA key protecting the blob, and it is protected with either the password, if provided, or the SHA1 constant CE4CF677875B5EB8993591D5A9AF1ED24A3A8736; the second represents the sealed object containing the wrapping key, and is protected with the SHA1 constant B9EE715DBE4B243FAA81EA04306E063710383E35. There exists no other user-land tool for decrypting this; perhaps there should be.

Finally, the equivalent of **zfs change-key -o** keylocation=prompt **-o** keyformat=raw *dataset* is performed with the new key. If an error occurred, best effort is made to clean up the properties, or to issue a note for manual intervention into the standard error stream.

A final verification should be made by running **zfs-tpm1x-load-key -n** *dataset*. If that command succeeds, all is well, but otherwise the dataset can be manually rolled back to a password with **zfs-tpm1x-clear-key** *dataset* (or, if that fails to work, **zfs change-key -o** keyformat=passphrase *dataset*), and you are hereby asked to report a bug, please.

zfs-tpm1x-clear-key *dataset* can be used to clear the properties and go back to using a password.

OPTIONS

-b *backup-file*

Save a back-up of the key to *backup-file*, which must not exist beforehand. This back-up *must* be stored securely, off-site. In case of a catastrophic event, the key can be loaded by running

zfs load-key *dataset* < *backup-file*

ENVIRONMENT VARIABLES

`TZPFMS_PASSPHRASE_HELPER`

By default, passphrases are prompted for and read in on the standard output and input streams. If `TZPFMS_PASSPHRASE_HELPER` is set and nonempty, it will be run via `/bin/`**`sh -c`** to provide each passphrase, instead.

The standard output stream of the helper is tied to an anonymous file and used in its entirety as the passphrase, except for a trailing new-line, if any. The arguments are:

- `$1` Pre-formatted noun phrase with all the information below, for use as a prompt
- `$2` Either the dataset name or the element of the TPM hierarchy being prompted for
- `$3` "new" if this is for a new passphrase, otherwise blank
- `$4` "again" if it's the second prompt for that passphrase, otherwise blank

If the helper doesn't exist (the shell exits with **127**), a diagnostic is issued and the normal prompt is used as fall-back. If it fails for any other reason, the prompting is aborted.

TPM1.X back-end configuration

TPM selection

The **`tzpfms`** suite connects to a local `tcsd`(8) process (at `localhost:30003`) by default. Use the environment variable `TZPFMS_TPM1X` to specify a remote TCS hostname.

The TrouSerS `tcsd`(8) daemon will try `/dev/tpm0`, then `/udev/tpm0`, then `/dev/tpm`; by occupying one of the earlier ones with, for example, shell redirection, a later one can be selected.

See also

The TrouSerS project page at **https://sourceforge.net/projects/trousers**.

The TPM 1.2 main specification index at **https://trustedcomputinggroup.org/resource/tpm-main-specification**.

SPECIAL THANKS

To all who support further development, in particular:

- ThePhD
- Embark Studios
- Jasper Bekkers

REPORTING BUGS

https://todo.sr.ht/~nabijaczleweli/tzpfms

`~nabijaczleweli/tzpfms@lists.sr.ht`, archived at **https://lists.sr.ht/~nabijaczleweli/tzpfms**.

SEE ALSO

https://git.sr.ht/~nabijaczleweli/tzpfms

NAME

zfs-tpm1x-clear-key — rewrap ZFS dataset key in passsword and clear tzpfms TPM1.X metadata

SYNOPSIS

zfs-tpm1x-clear-key *dataset*

DESCRIPTION

After verifying *dataset* was encrypted with **tzpfms** backend **TPM1.X**:

1. performs the equivalent of **zfs change-key -o** keylocation=prompt **-o** keyformat=passphrase *dataset*,
2. removes the xyz.nabijaczleweli:tzpfms.{backend, key} properties from *dataset*.

See zfs-tpm1x-change-key(8) for a detailed description.

TPM1.X back-end configuration

TPM selection

The **tzpfms** suite connects to a local tcsd(8) process (at localhost:30003) by default. Use the environment variable TZPFMS_TPM1X to specify a remote TCS hostname.

The TrouSerS tcsd(8) daemon will try /dev/tpm0, then /udev/tpm0, then /dev/tpm; by occupying one of the earlier ones with, for example, shell redirection, a later one can be selected.

See also

The TrouSerS project page at **https://sourceforge.net/projects/trousers**.

The TPM 1.2 main specification index at **https://trustedcomputinggroup.org/resource/tpm-main-specification**.

SPECIAL THANKS

To all who support further development, in particular:

- ThePhD
- Embark Studios
- Jasper Bekkers

REPORTING BUGS

https://todo.sr.ht/~nabijaczleweli/tzpfms

~nabijaczleweli/tzpfms@lists.sr.ht, archived at **https://lists.sr.ht/~nabijaczleweli/tzpfms**.

SEE ALSO

https://git.sr.ht/~nabijaczleweli/tzpfms

NAME

zfs-tpm1x-load-key — load TPM1.X-encrypted ZFS dataset key

SYNOPSIS

zfs-tpm1x-load-key [**-n**] *dataset*

DESCRIPTION

After verifying *dataset* was encrypted with **tzpfms** backend **TPM1.X** will unseal the key and load it into *dataset*.

The user is first prompted for the SRK passphrase, set when taking ownership, if not "well-known" (all zeroes); then for the additional passphrase, set when creating the key, if one was set.

See `zfs-tpm1x-change-key`(8) for a detailed description.

OPTIONS

-n
Do a no-op/dry run, can be used even if the key is already loaded. Equivalent to **zfs load-key**'s **-n** option.

ENVIRONMENT VARIABLES

`TZPFMS_PASSPHRASE_HELPER`
By default, passphrases are prompted for and read in on the standard output and input streams. If `TZPFMS_PASSPHRASE_HELPER` is set and nonempty, it will be run via /bin/**sh -c** to provide each passphrase, instead.

The standard output stream of the helper is tied to an anonymous file and used in its entirety as the passphrase, except for a trailing new-line, if any. The arguments are:

- `$1` Pre-formatted noun phrase with all the information below, for use as a prompt
- `$2` Either the dataset name or the element of the TPM hierarchy being prompted for
- `$3` "new" if this is for a new passphrase, otherwise blank
- `$4` "again" if it's the second prompt for that passphrase, otherwise blank

If the helper doesn't exist (the shell exits with **127**), a diagnostic is issued and the normal prompt is used as fall-back. If it fails for any other reason, the prompting is aborted.

TPM1.X back-end configuration

TPM selection

The **tzpfms** suite connects to a local `tcsd`(8) process (at `localhost:30003`) by default. Use the environment variable `TZPFMS_TPM1X` to specify a remote TCS hostname.

The TrouSerS `tcsd`(8) daemon will try `/dev/tpm0`, then `/udev/tpm0`, then `/dev/tpm`; by occupying one of the earlier ones with, for example, shell redirection, a later one can be selected.

See also

The TrouSerS project page at **https://sourceforge.net/projects/trousers**.

The TPM 1.2 main specification index at **https://trustedcomputinggroup.org/resource/tpm-main-specification**.

SPECIAL THANKS

To all who support further development, in particular:

- ThePhD
- Embark Studios
- Jasper Bekkers

REPORTING BUGS

https://todo.sr.ht/~nabijaczleweli/tzpfms

`~nabijaczleweli/tzpfms@lists.sr.ht`, archived at **https://lists.sr.ht/~nabijaczleweli/tzpfms**.

SEE ALSO

https://git.sr.ht/~nabijaczleweli/tzpfms

NAME

zfs-tpm2-change-key — change ZFS dataset key to one stored on the TPM

SYNOPSIS

zfs-tpm2-change-key [**-b** *backup-file*] *dataset*

DESCRIPTION

To normalise *dataset*, **zfs-tpm2-change-key** will open its encryption root in its stead. **zfs-tpm2-change-key** will *never* create or destroy encryption roots; use zfs-change-key(8) for that.

First, a connection is made to the TPM, which *must* be TPM-2.0-compatible.

If *dataset* was previously encrypted with **tzpfms** and the **TPM2** back-end was used, the previous key will be freed from the TPM. Otherwise, or in case of an error, data required for manual intervention will be printed to the standard error stream.

Next, a new wrapping key is generated on the TPM, optionally backed up (see **OPTIONS**), and sealed to a persistent object on the TPM under the owner hierarchy; if there is a passphrase set on the owner hierarchy, the user is prompted for it; the user is always prompted for an optional passphrase to protect the sealed object with.

The following properties are set on *dataset*:

- xyz.nabijaczleweli:tzpfms.backend=**TPM2**
- xyz.nabijaczleweli:tzpfms.key=*ID of persistent object*

tzpfms.backend identifies this dataset for work with **TPM2**-back-ended **tzpfms** tools (namely zfs-tpm2-change-key(8), zfs-tpm2-load-key(8), and zfs-tpm2-clear-key(8)).

tzpfms.key is an integer representing the sealed object; if needed, it can be passed to **tpm2_unseal -c** ${tzpfms.key} [**-p** ${password}] or equivalent for back-up (see **OPTIONS**). If you have a sealed key you can access with that or equivalent tool and set both of these properties, it will funxion seamlessly.

Finally, the equivalent of **zfs change-key -o** keylocation=prompt **-o** keyformat=raw *dataset* is performed with the new key. If an error occurred, best effort is made to clean up the persistent object and properties, or to issue a note for manual intervention into the standard error stream.

A final verification should be made by running **zfs-tpm2-load-key -n** *dataset*. If that command succeeds, all is well, but otherwise the dataset can be manually rolled back to a password with **zfs-tpm2-clear-key** *dataset* (or, if that fails to work, **zfs change-key -o** keyformat=passphrase *dataset*), and you are hereby asked to report a bug, please.

zfs-tpm2-clear-key *dataset* can be used to free the TPM persistent object and go back to using a password.

OPTIONS

-b *backup-file*

Save a back-up of the key to *backup-file*, which must not exist beforehand. This back-up *must* be stored securely, off-site. In case of a catastrophic event, the key can be loaded by running

zfs load-key *dataset* < *backup-file*

ENVIRONMENT VARIABLES

TZPFMS_PASSPHRASE_HELPER

By default, passphrases are prompted for and read in on the standard output and input streams. If TZPFMS_PASSPHRASE_HELPER is set and nonempty, it will be run via /bin/**sh -c** to provide each passphrase, instead.

The standard output stream of the helper is tied to an anonymous file and used in its entirety as the passphrase, except for a trailing new-line, if any. The arguments are:

$1 Pre-formatted noun phrase with all the information below, for use as a prompt
$2 Either the dataset name or the element of the TPM hierarchy being prompted for
$3 "new" if this is for a new passphrase, otherwise blank
$4 "again" if it's the second prompt for that passphrase, otherwise blank

If the helper doesn't exist (the shell exits with **127**), a diagnostic is issued and the normal prompt is used as fall-back. If it fails for any other reason, the prompting is aborted.

TPM2 back-end configuration

Environment variables

TSS2_LOG Any of: **NONE**, **ERROR**, **WARNING**, **INFO**, **DEBUG**, **TRACE**. Default: **WARNING**.

TPM selection

The library **libtss2-tcti-default.so** can be linked to any of the libtss2-tcti-*.so libraries to select the default, otherwise /dev/tpmrm0, then /dev/tpm0, then localhost:2321 will be tried, in order (see ESYS_CONTEXT(3)).

See also

The tpm2-tss git repository at **https://github.com/tpm2-software/tpm2-tss** and the documentation at **https://tpm2-tss.readthedocs.io**.

The TPM 2.0 specifications, mainly at **https://trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.38.pdf** and related pages.

SPECIAL THANKS

To all who support further development, in particular:

- ThePhD
- Embark Studios
- Jasper Bekkers

REPORTING BUGS

https://todo.sr.ht/~nabijaczleweli/tzpfms

~nabijaczleweli/tzpfms@lists.sr.ht, archived at **https://lists.sr.ht/~nabijaczleweli/tzpfms**.

SEE ALSO

tpm2_unseal(1)

https://git.sr.ht/~nabijaczleweli/tzpfms

NAME

zfs-tpm2-clear-key — rewrap ZFS dataset key in passsword and clear tzpfms TPM2 metadata

SYNOPSIS

zfs-tpm2-clear-key *dataset*

DESCRIPTION

After verifying *dataset* was encrypted with **tzpfms** backend **TPM2**:

1. performs the equivalent of **zfs change-key -o** keylocation=prompt **-o** keyformat=passphrase *dataset*,
2. frees the sealed key previously used to encrypt *dataset*,
3. removes the xyz.nabijaczleweli:tzpfms.{backend, key} properties from *dataset*.

See zfs-tpm2-change-key(8) for a detailed description.

ENVIRONMENT VARIABLES

TZPFMS_PASSPHRASE_HELPER

By default, passphrases are prompted for and read in on the standard output and input streams. If TZPFMS_PASSPHRASE_HELPER is set and nonempty, it will be run via /bin/**sh -c** to provide each passphrase, instead.

The standard output stream of the helper is tied to an anonymous file and used in its entirety as the passphrase, except for a trailing new-line, if any. The arguments are:

$1 Pre-formatted noun phrase with all the information below, for use as a prompt
$2 Either the dataset name or the element of the TPM hierarchy being prompted for
$3 "new" if this is for a new passphrase, otherwise blank
$4 "again" if it's the second prompt for that passphrase, otherwise blank

If the helper doesn't exist (the shell exits with **127**) , a diagnostic is issued and the normal prompt is used as fall-back. If it fails for any other reason, the prompting is aborted.

TPM2 back-end configuration

Environment variables

TSS2_LOG Any of: **NONE**, **ERROR**, **WARNING**, **INFO**, **DEBUG**, **TRACE**. Default: **WARNING**.

TPM selection

The library **libtss2-tcti-default.so** can be linked to any of the libtss2-tcti-*.so libraries to select the default, otherwise /dev/tpmrm0, then /dev/tpm0, then localhost:2321 will be tried, in order (see ESYS_CONTEXT(3)) .

See also

The tpm2-tss git repository at **https://github.com/tpm2-software/tpm2-tss** and the documentation at **https://tpm2-tss.readthedocs.io**.

The TPM 2.0 specifications, mainly at **https://trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.38.pdf** and related pages.

SPECIAL THANKS

To all who support further development, in particular:

- ThePhD
- Embark Studios

• Jasper Bekkers

REPORTING BUGS

https://todo.sr.ht/~nabijaczleweli/tzpfms

`~nabijaczleweli/tzpfms@lists.sr.ht`, archived at **https://lists.sr.ht/~nabijaczleweli/tzpfms**.

SEE ALSO

https://git.sr.ht/~nabijaczleweli/tzpfms

NAME

zfs-tpm2-load-key — load TPM2-encrypted ZFS dataset key

SYNOPSIS

zfs-tpm2-load-key [**-n**] *dataset*

DESCRIPTION

After verifying *dataset* was encrypted with **tzpfms** backend **TPM2**, unseals the key and loads it into *dataset*.

The user is prompted for the additional passphrase, set when creating the key, if one was set.

See `zfs-tpm2-change-key`(8) for a detailed description.

OPTIONS

-n

Do a no-op/dry run, can be used even if the key is already loaded. Equivalent to **zfs load-key**'s **-n** option.

ENVIRONMENT VARIABLES

`TZPFMS_PASSPHRASE_HELPER`

By default, passphrases are prompted for and read in on the standard output and input streams. If `TZPFMS_PASSPHRASE_HELPER` is set and nonempty, it will be run via /bin/**sh -c** to provide each passphrase, instead.

The standard output stream of the helper is tied to an anonymous file and used in its entirety as the passphrase, except for a trailing new-line, if any. The arguments are:

- `$1` Pre-formatted noun phrase with all the information below, for use as a prompt
- `$2` Either the dataset name or the element of the TPM hierarchy being prompted for
- `$3` "new" if this is for a new passphrase, otherwise blank
- `$4` "again" if it's the second prompt for that passphrase, otherwise blank

If the helper doesn't exist (the shell exits with **127**), a diagnostic is issued and the normal prompt is used as fall-back. If it fails for any other reason, the prompting is aborted.

TPM1.X back-end configuration

TPM selection

The **tzpfms** suite connects to a local `tcsd`(8) process (at `localhost:30003`) by default. Use the environment variable `TZPFMS_TPM1X` to specify a remote TCS hostname.

The TrouSerS `tcsd`(8) daemon will try `/dev/tpm0`, then `/udev/tpm0`, then `/dev/tpm`; by occupying one of the earlier ones with, for example, shell redirection, a later one can be selected.

See also

The TrouSerS project page at **https://sourceforge.net/projects/trousers**.

The TPM 1.2 main specification index at **https://trustedcomputinggroup.org/resource/tpm-main-specification**.

SPECIAL THANKS

To all who support further development, in particular:

- ThePhD

- Embark Studios
- Jasper Bekkers

REPORTING BUGS

https://todo.sr.ht/~nabijaczleweli/tzpfms

`~nabijaczleweli/tzpfms@lists.sr.ht`, archived at **https://lists.sr.ht/~nabijaczleweli/tzpfms**.

SEE ALSO

https://git.sr.ht/~nabijaczleweli/tzpfms