NAME

fido2-token - find and manage a FIDO2 authenticator

SYNOPSIS

fido2-token -C [**-d**] *device*
fido2-token -D [**-d**] **-i** *cred_id device*
fido2-token -D -b [**-d**] **-k** *key_path device*
fido2-token -D -b [**-d**] **-n** *rp_id* [**-i** *cred_id*] *device*
fido2-token -D -e [**-d**] **-i** *template_id device*
fido2-token -D -u [**-d**] *device*
fido2-token -G -b [**-d**] **-k** *key_path blob_path device*
fido2-token -G -b [**-d**] **-n** *rp_id* [**-i** *cred_id*] *blob_path device*
fido2-token -I [**-cd**] [**-k** *rp_id* **-i** *cred_id*] *device*
fido2-token -L [**-bder**] [**-k** *rp_id*] [device]
fido2-token -R [**-d**] *device*
fido2-token -S [**-adefu**] *device*
fido2-token -S [**-d**] **-i** *template_id* **-n** *template_name device*
fido2-token -S [**-d**] **-l** *pin_length device*
fido2-token -S -b [**-d**] **-k** *key_path blob_path device*
fido2-token -S -b [**-d**] **-n** *rp_id* [**-i** *cred_id*] *blob_path device*
fido2-token -S -c [**-d**] **-i** *cred_id* **-k** *user_id* **-n** *name* **-p** *display_name device*
fido2-token -S -m *rp_id device*
fido2-token -V

DESCRIPTION

fido2-token manages a FIDO2 authenticator.

The options are as follows:

-C *device*

Changes the PIN of *device*. The user will be prompted for the current and new PINs.

-D -i *id device*

Deletes the resident credential specified by *id* from *device*, where *id* is the credential's base64-encoded id. The user will be prompted for the PIN.

-D -b -k *key_path device*

Deletes a "largeBlob" encrypted with *key_path* from *device*, where *key_path* holds the blob's base64-encoded 32-byte AES-256 GCM encryption key. A PIN or equivalent user-verification gesture is required.

-D -b -n *rp_id* [**-i** *cred_id*] *device*

Deletes a "largeBlob" corresponding to *rp_id* from *device*. If *rp_id* has multiple credentials enrolled on *device*, the credential ID must be specified using **-i** *cred_id*, where *cred_id* is a base64-encoded blob. A PIN or equivalent user-verification gesture is required.

-D -e -i *id device*

Deletes the biometric enrollment specified by *id* from *device*, where *id* is the enrollment's template base64-encoded id. The user will be prompted for the PIN.

-D -u *device*

Disables the CTAP 2.1 "user verification always" feature on *device*.

-G -b -k *key_path blob_path device*

Gets a CTAP 2.1 "largeBlob" encrypted with *key_path* from *device*, where *key_path* holds the blob's base64-encoded 32-byte AES-256 GCM encryption key. The blob is written to *blob_path*. A PIN or equivalent user-verification gesture is required.

-G -b -n *rp_id* [**-i** *cred_id*] *blob_path device*

Gets a CTAP 2.1 "largeBlob" associated with *rp_id* from *device*. If *rp_id* has multiple credentials enrolled on *device*, the credential ID must be specified using **-i** *cred_id*, where *cred_id* is a base64-encoded blob. The blob is written to *blob_path*. A PIN or equivalent user-verification gesture is required.

-I *device*

Retrieves information on *device*.

-I -c *device*

Retrieves resident credential metadata from *device*. The user will be prompted for the PIN.

-I -k *rp_id* **-i** *cred_id device*

Prints the credential id (base64-encoded) and public key (PEM encoded) of the resident credential specified by *rp_id* and *cred_id*, where *rp_id* is a UTF-8 relying party id, and *cred_id* is a base64-encoded credential id. The user will be prompted for the PIN.

-L Produces a list of authenticators found by the operating system.

-L -b *device*

Produces a list of CTAP 2.1 "largeBlobs" on *device*. A PIN or equivalent user-verification gesture is required.

-L -e *device*
Produces a list of biometric enrollments on *device*. The user will be prompted for the PIN.

-L -r *device*
Produces a list of relying parties with resident credentials on *device*. The user will be prompted for the PIN.

-L -k *rp_id device*
Produces a list of resident credentials corresponding to relying party *rp_id* on *device*. The user will be prompted for the PIN.

-R Performs a reset on *device*. **fido2-token** will NOT prompt for confirmation.

-S Sets the PIN of *device*. The user will be prompted for the PIN.

-S -a *device*
Enables CTAP 2.1 Enterprise Attestation on *device*.

-S -b -k *key_path blob_path device*
Sets a CTAP 2.1 "largeBlob" encrypted with *key_path* on *device*, where *key_path* holds the blob's base64-encoded 32-byte AES-256 GCM encryption key. The blob is read from *blob_path*. A PIN or equivalent user-verification gesture is required.

-S -b -n *rp_id* [**-i** *cred_id*] *blob_path device*
Sets a CTAP 2.1 "largeBlob" associated with *rp_id* on *device*. The blob is read from *blob_path*. If *rp_id* has multiple credentials enrolled on *device*, the credential ID must be specified using **-i** *cred_id*, where *cred_id* is a base64-encoded blob. A PIN or equivalent user-verification gesture is required.

-S -c -i *cred_id* **-k** *user_id* **-n** *name* **-p** *display_name device*
Sets the *name* and *display_name* attributes of the resident credential identified by *cred_id* and *user_id*, where *name* and *display_name* are UTF-8 strings and *cred_id* and *user_id* are base64-encoded blobs. A PIN or equivalent user-verification gesture is required.

-S -e *device*
Performs a new biometric enrollment on *device*. The user will be prompted for the PIN.

-S -e -i *template_id* **-n** *template_name device*
Sets the friendly name of the biometric enrollment specified by *template_id* to *template_name* on *device*, where *template_id* is base64-encoded and *template_name* is a UTF-8 string. The

user will be prompted for the PIN.

-S -f *device*
Forces a PIN change on *device*. The user will be prompted for the PIN.

-S -l *pin_length device*
Sets the minimum PIN length of *device* to *pin_length*. The user will be prompted for the PIN.

-S -m *rp_id device*
Sets the list of relying party IDs that are allowed to retrieve the minimum PIN length of *device*. Multiple IDs may be specified, separated by commas. The user will be prompted for the PIN.

-S -u *device*
Enables the CTAP 2.1 "user verification always" feature on *device*.

-V Prints version information.

-d Causes **fido2-token** to emit debugging output on *stderr*.

If a *tty* is available, **fido2-token** will use it to prompt for PINs. Otherwise, *stdin* is used.

fido2-token exits 0 on success and 1 on error.

SEE ALSO

fido2-assert(1), fido2-cred(1)

CAVEATS

The actual user-flow to perform a reset is outside the scope of the FIDO2 specification, and may therefore vary depending on the authenticator. Yubico authenticators do not allow resets after 5 seconds from power-up, and expect a reset to be confirmed by the user through touch within 30 seconds.

An authenticator's path may contain spaces.

Resident credentials are called "discoverable credentials" in CTAP 2.1.

Whether the CTAP 2.1 "user verification always" feature is activated or deactivated after an authenticator reset is vendor-specific.