NAME

nitrocli − access Nitrokey devices

SYNOPSIS

nitrocli *command* [*arguments*]

DESCRIPTION

nitrocli provides access to Nitrokey devices. It supports the Nitrokey Pro, the Nitrokey Storage, and the Librem Key. It can be used to access the encrypted volume, the one-time password generator, and the password safe.

Device selection

Per default, **nitrocli** connects to any attached Nitrokey device. You can use the **−−model**, **−−serial-number** and **−−usb-path** options to select the device to connect to. **nitrocli** fails if more than one attached Nitrokey device matches this filter or if multiple Nitrokey devices are attached and none of the filter options is set. Use the **list** command to list all attached devices with their USB path, model, and serial number (if available).

OPTIONS

−m, **−−model librem|pro|storage**
: Restrict connections to the given device model, see the Device selection section.

−−serial-number *serial-number*
: Restrict connections to the given serial number, see the Device selection section. *serial-number* must be a hex string with an optional 0x prefix. This option can be set multiple times to allow any of the given serial numbers. Nitrokey Storage devices never match this restriction as they do not expose their serial number in the USB device descriptor.

−−usb-path *usb-path*
: Restrict connections to the given USB path, see the Device selection section.

−−no−cache
: If this option is set, nitrocli will not cache any inquired secrets using **gpg−agent**(1) but ask for them each time they are needed. Note that this option does not cause any cached secrets to be cleared. If a secret is already in the cache it will be ignored, but left otherwise untouched. Use the **pin clear** command to clear secrets from the cache.

−v, **−−verbose**
: Enable additional logging and control its verbosity. Logging enabled through this option will appear on the standard error stream. This option can be supplied multiple times. A single occurrence will show additional warnings. Commands sent to the device will be shown when supplied three times and full device communication is available with four occurrences. Supplying this option five times enables the highest verbosity.

−V, **−−version**
: Print the nitrocli version and exit.

COMMANDS

General

nitrocli list [**-n|−−no-connect**]
: List all attached Nitrokey devices. This command prints a list of the USB path, the model and the serial number of all attached Nitrokey devices. To access the serial number of a Nitrokey Storage device, **nitrocli** has to connect to it. To omit the serial number of Nitrokey Storage devices instead of connecting to them, set the **−−no-connect** option.

nitrocli status
: Print the status of the connected Nitrokey device, including the stick serial number, the firmware version, and the PIN retry count. If the device is a Nitrokey Storage, also print storage related information including the SD card serial number, the SD card usage during this power cycle, the encryption status, and the status of the volumes.

nitrocli lock
Lock the Nitrokey. This command locks the password safe (see the Password safe section). On the Nitrokey Storage, it will also close any active encrypted or hidden volumes (see the Storage section).

nitrocli reset [**−−only-aes-key**]
Perform a factory reset on the Nitrokey. This command performs a factory reset on the OpenPGP smart card, clears the flash storage and builds a new AES key. The user PIN is reset to 123456, the admin PIN to 12345678.

If the **−−only-aes-key** option is set, the command does not perform a full factory reset but only creates a new AES key. The AES key is for example used to encrypt the password safe.

This command requires the admin PIN. To avoid accidental calls of this command, the user has to enter the PIN even if it has been cached.

Storage

The Nitrokey Storage comes with a storage area. This area is comprised of an *unencrypted* region and an *encrypted* one of fixed sizes, each made available to the user in the form of block devices. The encrypted region can optionally further be overlayed with up to four *hidden* volumes. Because of this overlay (which is required to achieve plausible deniability of the existence of hidden volumes), the burden of ensuring that data on the encrypted volume does not overlap with data on one of the hidden volumes is on the user.

nitrocli unencrypted set *mode*
Change the read-write mode of the volume. *mode* is the type of the mode to change to: **read-write** to make the volume readable and writable or **read-only** to make it only readable. This command requires the admin PIN.

Note that this command requires firmware version 0.51 or higher. Earlier versions are not supported.

nitrocli encrypted open
Open the encrypted volume on the Nitrokey Storage. The user PIN that is required to open the volume is queried using **pinentry**(1) and cached by **gpg−agent**(1).

nitrocli encrypted close
Close the encrypted volume on the Nitrokey Storage.

nitrocli hidden create *slot start end*
Create a new hidden volume inside the encrypted volume. *slot* must indicate one of the four available slots. *start* and *end* represent, respectively, the start and end position of the hidden volume inside the encrypted volume, as a percentage of the encrypted volume's size. This command requires a password which is later used to look up the hidden volume to open. Unlike a PIN, this password is not cached by **gpg−agent**(1).

As a guide line for creating new hidden volumes, the **status** command provides a range of the SD card that has not been written to during this power cycle.

nitrocli hidden open
Open a hidden volume. The volume to open is determined based on the password entered, which must have a minimum of six characters. Only one hidden volume can be active at any point in time and previously opened volumes will be automatically closed. Similarly, the encrypted volume will be closed if it was open.

nitrocli hidden close
Close a hidden volume.

nitrocli fill [**−a**|**−−attach**
Fills the SD card with random data, overwriting all existing data. This operation takes about one hour to finish for a 16 GiB SD card. It cannot be canceled, even if the **nitrocli** process is terminated before it finishes.

This command requires the admin PIN. To avoid accidental calls of this command, the user has to enter the PIN even if it has been cached.

If the **−−attach** option is set, this command will not start a new fill operation. Instead it checks whether a fill operation is currently running on the device and shows its progress.

One-time passwords

The Nitrokey Pro, the Nitrokey Storage, and the Librem Key support the generation of one-time passwords using the HOTP algorithm according to RFC 4226 or the TOTP algorithm according to RFC 6238. The required data – a name and the secret – is stored in slots. Currently, the Nitrokey devices provide three HOTP slots and 15 TOTP slots. The slots are numbered per algorithm starting at zero.

The TOTP algorithm is a modified version of the HOTP algorithm that also uses the current time. Therefore, the Nitrokey clock must be synchronized with the clock of the application that requests the one-time password.

nitrocli otp get *slot* [**−a**|**−−algorithm** *algorithm*] **[−t**|**−−time** *time*]
Generate a one-time password. *slot* is the number of the slot to generate the password from. *algorithm* is the OTP algorithm to use. Possible values are **hotp** for the HOTP algorithm according to RFC 4226 and **totp** for the TOTP algorithm according to RFC 6238 (default). Per default, this commands sets the Nitrokey's time to the system time if the TOTP algorithm is selected. If **−−time** is set, it is set to *time* instead, which must be a Unix timestamp (i.e., the number of seconds since 1970-01-01 00:00:00 UTC). This command might require the user PIN (see the Configuration section).

nitrocli otp set *slot name secret* [**−a**|**−−algorithm** *algorithm*] [**−d**|**−−digits** *digits*] [**−c**|**−−counter** *counter*] [**−t**|**−−time-window** *time-window*] [**-f**|**−−format ascii**|**base32**|**hex**]
Configure a one-time password slot. *slot* is the number of the slot to configure. *name* is the name of the slot (may not be empty). *secret* is the secret value to store in that slot.

The **−−format** option specifies the format of the secret. If it is set to **ascii**, each character of the given secret is interpreted as the ASCII code of one byte. If it is set to **base32**, the secret is interpreted as a base32 string according to RFC 4648. If it is set to **hex**, every two characters are interpreted as the hexadecimal value of one byte. The default value is **base32**.

algorithm is the OTP algorithm to use. Possible values are **hotp** for the HOTP algorithm according to RFC 4226 and **totp** for the TOTP algorithm according to RFC 6238 (default). *digits* is the number of digits the one-time password should have. Allowed values are 6 and 8 (default: 6). *counter* is the initial counter if the HOTP algorithm is used (default: 0). *time-window* is the time window used with TOTP in seconds (default: 30).

nitrocli otp clear *slot* [**−a**|**−−algorithm** *algorithm*]
Delete the name and the secret stored in a one-time password slot. *slot* is the number of the slot to clear. *algorithm* is the OTP algorithm to use. Possible values are **hotp** for the HOTP algorithm according to RFC 4226 and **totp** for the TOTP algorithm according to RFC 6238 (default).

nitrocli otp status [**−a**|**−−all**]
List all OTP slots. If **−−all** is not set, empty slots are ignored.

Configuration

Nitrokey devices have four configuration settings: the Num Lock, Caps Lock and Scroll Lock keys can be mapped to an HOTP slot, and OTP generation can be set to require the user PIN.

nitrocli config get
Print the current Nitrokey configuration.

nitrocli config set [[**−n**|**−−num-lock** *slot*] | [**−N**|**−−no−num-lock**]] [[**−c**|**−−caps-lock** *slot*] | [**−C**|**−−no−caps-lock**]] [[**−s**|**−−scroll-lock** *slot*] | [**−S**|**−−no−scroll-lock**]] [[**−o**|**−−otp−pin**] | [**−O**|**−−no−otp−pin**]]
Update the Nitrokey configuration. This command requires the admin PIN.

With the **−−num-lock**, **−−caps-lock** and **−−scroll-lock** options, the respective bindings can be set. *slot* is the number of the HOTP slot to bind the key to. If **−−no−num-lock**, **−−no−caps-lock** or **−−no−scroll-lock** is set, the respective binding is disabled. The two corresponding options are mutually exclusive.

If **−−otp−pin** is set, the user PIN will be required to generate one-time passwords using the **otp get** command. If **−−no−otp−pin** is set, OTP generation can be performed without PIN. These two options are mutually exclusive.

Password safe

The Nitrokey Pro, the Nitrokey Storage, and the Librem Key provide a password safe (PWS) with 16 slots. In each of these slots you can store a name, a login, and a password. The PWS is not encrypted, but it is protected with the user PIN by the firmware. Once the PWS is unlocked by one of the commands listed below, it can be accessed without authentication. You can use the **lock** command to lock the password safe.

nitrocli pws get *slot* [**−n**|**−−name**] [**−l**|**−−login**] [**−p**|**−−password**] [**−q**|**−−quiet**]
Print the content of one PWS slot. *slot* is the number of the slot. Per default, this command prints the name, the login and the password (in that order). If one or more of the options **−−name**, **−−login**, and **−−password** are set, only the selected fields are printed. The order of the fields never changes.

The fields are printed together with a label. Use the **−−quiet** option to suppress the labels and to only output the values stored in the PWS slot.

nitrocli pws set *slot name login password*
Set the content of a PWS slot. *slot* is the number of the slot to write. *name*, *login*, and *password* represent the data to write to the slot.

nitrocli pws add *slot name login password*
Add a new PWS slot. This command locates the first free PWS slot and sets its content to the given values. It fails if all PWS slots are programmed.

nitrocli pws update *slot* [**−n**|**−−name** *name*] [**−l**|**−−login** *login*] [**−p**|**−−password** *password*]
Update the content of a programmed PWS slot. *slot* is the number of the slot to write. This command only sets the data given with the **−−name**, **−−login**, and **−−password** options and does not overwrite the other fields of the slot.

nitrocli pws clear *slot*
Delete the data stored in a PWS slot. *slot* is the number of the slot clear.

nitrocli pws status [**−a**|**−−all**]
List all PWS slots. If **−−all** is not set, empty slots are ignored.

PINs

Nitrokey devices have two PINs: the user PIN and the admin PIN. The user PIN must have at least six, the admin PIN at least eight characters. The user PIN is required for commands such as **otp get** (depending on the configuration) and for all **pws** commands. The admin PIN is usually required to change the device configuration.

Each PIN has a retry counter that is decreased with every wrong PIN entry and reset if the PIN was entered

correctly. The initial retry counter is three. If the retry counter for the user PIN is zero, you can use the **pin unblock** command to unblock and reset the user PIN. If the retry counter for the admin PIN is zero, you have to perform a factory reset using the **reset** command or **gpg**(1). Use the **status** command to check the retry counters.

nitrocli pin clear
: Clear the PINs cached by the other commands. Note that cached PINs are associated with the device they belong to and the **clear** command will only clear the PIN for the currently used device, not all others.

nitrocli pin set *type*
: Change a PIN. *type* is the type of the PIN that will be changed: **admin** to change the admin PIN or **user** to change the user PIN. This command only works if the retry counter for the PIN type is at least one. (Use the **status** command to check the retry counters.)

nitrocli pin unblock
: Unblock and reset the user PIN. This command requires the admin PIN. The admin PIN cannot be unblocked. This operation is equivalent to the unblock PIN option provided by **gpg**(1) (using the **−−change−pin** option).

Extensions

In addition to the above built-in commands, **nitrocli** supports user-provided functionality in the form of extensions. An extension can be any executable file whose filename starts with "nitrocli-" and that is discoverable through lookup via the **PATH** environment variable. Those executables can be invoked as regular subcommands (without the need of the prefix; e.g., an extension with the name "nitrocli-otp-cache" could be invoked as "nitrocli otp-cache").

More information on how to write extensions can be found in the Extensions section below.

CONFIG FILE

nitrocli tries to read the configuration file at **${XDG_CONFIG_HOME}/nitrocli/config.toml** (or **${HOME}/nitrocli/config.toml** if the **XDG_CONFIG_HOME** environment variable is not set). It is used to set default values for the options listed below.

You can also set the environment variable **NITROCLI_***KEY* to overwrite the configuration for *key* (see the Environment section). Note that command-line arguments overwrite both the configuration file and the environment variables.

The following values can be set in the configuration file:

model Restrict connections to the given device model (string, default: not set, see **−−model**).

serial_numbers
: Restrict connections to the given serial numbers (list of strings, default: empty, see **−−serial-number**).

usb_path
: Restrict connections to the given USB path (string, default: not set, see **−−usb-path**).

no_cache
: If set to true, do not cache any inquired secrets (boolean, default: false, see **−−no−cache**).

verbosity
: Set the log level (integer, default: 0, see **−−verbose**).

The configuration file must use the TOML format, for example:

```
model = "pro"
serial_numbers = ["0xf00baa", "deadbeef"]
usb_path = "0001:0006:02"
no_cache = false
verbosity = 0
```

ENVIRONMENT

The program honors two sets of environment variables, all prefixed by "NITROCLI_".

Configuration

The first set controls basic configuration of the program. These variables mirror the respective command line options and configuration file settings. They are:

NITROCLI_MODEL
Restrict connections to the given device model (string, default: not set, see **−−model**).

NITROCLI_SERIAL_NUMBERS
Restrict connections to the given list of serial numbers (comma-separated list of strings, default: empty, see **−−serial-number**).

NITROCLI_USB_PATH
Restrict connections to the given USB path (string, default: not set, see **−−usb-path**).

NITROCLI_NO_CACHE
If set to true, do not cache any inquired secrets (boolean, default: false, see **−−no−cache**).

NITROCLI_VERBOSITY
Set the log level (integer, default: 0, see **−−verbose**).

Password & PIN entry

The second set can be used to provide password & PIN data to the program to suppress interactive entry through **pinentry**(1) for operations that otherwise would ask for it. The following variables are recognized:

NITROCLI_ADMIN_PIN
The admin PIN to use.

NITROCLI_USER_PIN
The user PIN to use.

NITROCLI_NEW_ADMIN_PIN
The new admin PIN to set. This variable is only used by the **pin set** command for the **admin** type.

NITROCLI_NEW_USER_PIN
The new user PIN to set. This variable is only used by the **pin set** command for the **user** type.

NITROCLI_PASSWORD
A password used by commands that require one (e.g., **hidden open**).

EXTENSIONS

nitrocli supports user-provided extensions that are executable files whose filename starts with "nitrocli-" and that are discoverable through lookup via the **PATH** environment variable.

The program conveys basic configuration information to any extension being started this way. Specifically, it will set each environment variable as described in the Configuration subsection of the Environment section above, if the corresponding **nitrocli** program configuration was set. In addition, the following variables will be set:

NITROCLI_BINARY
The absolute path to the **nitrocli** binary through which the extension was invoked. This path may be used to recursively invoke **nitrocli** to implement certain functionality.

NITROCLI_RESOLVED_USB_PATH
The USB path of the device that **nitrocli** would connect to based on the **−−model**, **−−serial-number**, and **−−usb-path** options. If there is no matching Nitrokey device, or if multiple devices match the options, the environment variable is not set.

All other variables present in the environment will be passed through to the extension verbatim.

Newer versions of the program reserve the right to set additional environment variables inside the

"NITROCLI_" namespace. As such, extensions are advised to not define custom variables with this prefix. However, "NITROCLI_EXT_" is provided specifically for this purpose. To further avoid conflicts between extensions, it is recommended that this prefix be followed by the extension's name (uppercased).

Extensions may optionally read or write persistent data of various forms. Similar to the main program, extensions should follow the XDG Base Directory Specification as a guideline where to store such data. More specifically, the following conventions should be followed:

For configuration data, **${XDG_CONFIG_HOME}/***extension/* is the preferred directory, where *extension* is the full extension name, including the "nitrocli-" prefix. The recommended configuration format is TOML. If only a single configuration file is used, **config.toml** is the recommended name.

Similarly, regular data should reside in **${XDG_DATA_HOME}/***extension/* and cached data be stored in **${XDG_CACHE_HOME}/***extension/*.

FILES

${XDG_CONFIG_HOME}/nitrocli/config.toml

${HOME}/nitrocli/config.toml
: User configuration file, see the Config file section.

EXAMPLES

Storage

Create a hidden volume in the first available slot, starting at half the size of the encrypted volume (i.e., 50%) and stretching all the way to its end (100%):

$ **nitrocli hidden create 0 50 100**

One-time passwords

Configure a one-time password slot with a hexadecimal secret representation:

$ **nitrocli otp set 0 test−rfc4226 31323334353637383930313233343536373839 −−format hex −−algorithm hotp**

$ **nitrocli otp set 1 test−foobar 666F6F626172 −−format hex −−algorithm hotp**

$ **nitrocli otp set 0 test−rfc6238 31323334353637383930313233343536373839 −−format hex −−algorithm totp −−digits 8**

Configure a one-time password slot with an ASCII secret representation:

$ **nitrocli otp set 0 test−rfc4226 12345678901234567890 −−format ascii −−algorithm hotp**

$ **nitrocli otp set 1 test−foobar foobar −−format ascii −−algorithm hotp**

$ **nitrocli otp set 0 test−rfc6238 12345678901234567890 −−format ascii −−algorithm totp −−digits 8**

Configure a one-time password slot with a base32 secret representation:

$ **nitrocli otp set 0 test−rfc4226 gezdgnbvgy3tqojqgezdgnbvgy3tqojq −−algorithm hotp**

$ **nitrocli otp set 1 test−foobar mzxw6ytboi====== −−algorithm hotp**

$ **nitrocli otp set 0 test−rfc6238 gezdgnbvgy3tqojqgezdgnbvgy3tqojq −−algorithm totp −−digits 8**

Generate a one-time password:

$ **nitrocli otp get 0 −−algorithm hotp**
755224

$ **nitrocli otp get 0 −−algorithm totp −−time 1234567890**
89005924

Clear a one-time password slot:

$ **nitrocli otp clear 0 −−algorithm hotp**

Configuration

Query the configuration:

$ **nitrocli config get**

```
Config:
  numlock binding:         not set
  capslock binding:        not set
  scrollock binding:       not set
  require user PIN for OTP: true
```

Change the configuration:

$ **nitrocli config set −−otp−pin**

Password safe

Configure a PWS slot:

$ **nitrocli pws set 0 example.org john.doe passw0rd**

Get the data from a slot:

$ **nitrocli pws get 0**

```
name:     example.org
login:    john.doe
password: passw0rd
```

Copy the password to the clipboard (requires **xclip**(1)).

$ **nitrocli pws get 0 −−password −−quiet | xclip −in**

Query the PWS slots:

$ **nitrocli pws status**

```
slot    name
0       example.org
```