Theo Documentation

Michele (macno) Azzolari

Jan 25, 2021

Setup

1 First steps **3**

2 Public test instance **5**
2.1 Setup . . . 5
2.1.1 Cookbook . . . 5
2.1.2 Full install . . . 8
2.1.3 Generate SSH keys . . . 8
2.2 theo installation . . . 8
2.2.1 With docker . . . 8
2.2.2 From sources . . . 8
2.3 theo-cli installation . . . 10
2.3.1 NPM . . . 10
2.3.2 Sources . . . 10
2.3.3 Configuration . . . 11
2.4 Sign SSH keys . . . 11
2.4.1 Use authorized key signature . . . 11
2.4.2 Setup . . . 11
2.4.3 Configure . . . 12
2.4.4 Usage . . . 12
2.5 theo-cli usage . . . 13
2.5.1 Accounts . . . 13
2.5.2 Groups . . . 15
2.5.3 SSH Keys . . . 17
2.5.4 Permissions . . . 18
2.5.5 Authorized Keys . . . 19
2.5.6 Examples . . . 19
2.6 Installation . . . 23
2.6.1 1. Download one of the binaries for your system: . . . 23
2.6.2 2. Make it executable . . . 23
2.6.3 3. Create a Theo Agent user: . . . 23
2.6.4 4. Install . . . 24
2.6.5 5. Restart openssh . . . 25
2.6.6 6. SELinux . . . 25
2.7 Options . . . 25
2.7.1 1. Installation . . . 25
2.7.2 2. Execution . . . 26

2.8 Configuration 26
2.9 Verify SSH keys 27
2.9.1 Use authorized key signature 27
2.9.2 Setup 27
2.9.3 Configure 27

3 Indices and tables **29**

Theo App is the authorized keys manager, you can use it as replacement for all of your *authorized_keys* It allows you to set fine permissions (specific user and host) or to use wildcard (ex, using host *%.test.sample.com*)

CHAPTER 1

First steps

Theo is based on 3 components:

1. theo, the core HTTP application
2. theo-cli, the command line interface to administer Theo
3. theo-agent, the program that will be executed by sshd to retrieve AuthorizedKeys

- **Getting started**: *Cookbook*

CHAPTER 2

Public test instance

A public test instance is available at theo.test.authkeys.io

Database will be reset every 6 hours (0am 6am 12pm 18pm UTC)

Configured tokens:

```
ADMIN_TOKEN=RMkqF4B8h6jtv3upvy3QubzNyTrMdgn8

CLIENT_TOKENS=h8LYYwGgTqKFYQ3mRN2hv8vK5CBGJvMs,gAWXaG9ZnhHAXsDbF6dv3NYEbPNuZKR7
```

Instance has the *REQUIRE_SIGNED_KEY* flag on, so you need to enable key sign/verify on your side

Be aware that the instance is public, so everyone has access to the data, please use fake email

2.1 Setup

2.1.1 Cookbook

While it's possible to install **theo** and the other components on one single server, you will appreciate all the power of **theo** with multiple servers. We'll illustrate here a scenario with 2 servers (one for **theo**, the other for **theo-agent**) and a computer for **theo-cli**.

Let's assume the server on which we will install **theo** is `server` and the other is `node-a`

theo

Install with docker

On `server` you can easily run **theo** as docker container

NOTE don't forget to replace *ADMIN_TOKEN* and *CLIENT_TOKENS* values!

```
$ docker run --rm -v /tmp/theo:/data \
    -e DB_STORAGE=/data/theo.db \
    -e ADMIN_TOKEN=12345 \
    -e CLIENT_TOKENS=abcde,fghij \
    -p 9100:9100 theoapp/theo
```

Executing the command will result in a running instance of **Theo** listening on port *9100* and accepting calls from:

- theo-agent using token *fghij* [1]
- theo-cli using token *12345* [1]

[1] Token are sent as HTTP header *Authorization: Bearer *token**

Install from sources

Please refert to *Full install* to install `theo` from sources

theo-cli

Install

To manage `theo` we use `theo-cli`

`theo-cli` is a node app, available on npm, install it on your computer

```
$ npm install -g theoapp-cli
```

`theo-cli` needs 2 variables: *THEO_URL* and *THEO_TOKEN*. You can set them as environment variables:

```
$ export THEO_URL=http://server:9100
$ export THEO_TOKEN=12345
```

Note Refer to `theo-cli` *install document* for other ways to set these variables

Create first account

Now you are ready to create the first account on `theo`

```
$ theo \
    accounts add \
    --name john.doe \
    --email john.doe@sample.com
```

Add public key to account

Now you need to add a public key to john.doe, you'll use your public key

Note if you don't have it or want to generate another one see *generate ssh key*

```
$ theo \
    keys add john.doe@sample.com \
    -k "$(cat ~/.ssh/id_rsa.pub)"
```

Add permission to account

Now we need to add permission to john.doe@sample.com to access `server` as `root` (or other existing linux user of `server`)

```
$ theo \
    add \
    --user john.doe@sample.com \
    --host node-a \
    --user root
```

theo-agent

Download

theo-agent is a program written in go. You need to connect to `node-a` and run

```
$ sudo curl -L -o /usr/sbin/theo-agent \
    https://github.com/theoapp/theo-agent/releases/download/$(curl -L -s -H 'Accept:␣
↪application/json' https://github.com/theoapp/theo-agent/releases/latest |sed -e 's/.
↪*"tag_name":"\([^"]*\)".*/\1/')/theo-agent-linux-amd64
```

And make it executable

```
$ sudo chmod 755 /usr/sbin/theo-agent
```

Install

You need to create a system user:

```
sudo useradd \
    --comment 'Theo Agent' \
    --shell /bin/false \
    --system \
    theo-agent
```

Configure

You can let **theo-agent** to configure itself automatically:

With this command you will: disable ssh password authentication, disable AuthorizedKeysFile from user's home (sshd will look for them in /var/cache/theo-agent/%u)

```
$ sudo theo-agent -install \
    -no-interactive \
    -sshd-config \
```

(continues on next page)

(continued from previous page)

```
    -url http://server:9100 \
    -token fghij
```

Final check

Now you're ready to test if everything is working, connect from your computer to `node-a`

```
ssh root@node-a
```

Congratulations!! You made it!

2.1.2 Full install

WIP

See

- **theo** *intallation guide*
- **theo-cli** *intallation guide*
- **theo-agent** *intallation guide*

2.1.3 Generate SSH keys

To generate SSH

```
$ ssh-keygen -b 4096
```

Leaving all the defaults, the command creates a new key in `~/.ssh/id_rsa`.

The public key is `~/.ssh/id_rsa.pub`

The public key will be used by the remote server to authorize the connection.

2.2 theo installation

2.2.1 With docker

```
$ docker pull theoapp/theo
```

2.2.2 From sources

Clone repo

```
$ git clone https://github.com/theoapp/theo-node.git
```

Install dependencies

```
$ npm i --no-optional
```

Build

```
$ npm run build
```

Configure

To configure there are 2 ways:

1. Using environment variables
2. Using `settings.json`

1. Environment variables

Name	Mandatory	Default value	Meaning	Type
PORT	NO	9100	The port on wichh the http server will listen	int
DB_ENGINE	NO	sqlite	Use sqlite3 as database. Needs DB_STORAGE	enum: * sqlite * mariadb
			Use mariadb as database. Needs DB_HOST, DB_USER, DB_PASSWORD, DB_NAME	
DB_STORAGE	NO	./data/theo.db	Path to sqlite3 db. Absolute or relative to node process	string
DB_HOST	YES(1)		Mariadb server hostname or ip	string
DB_USER	YES(1)		Mariadb username	string
DB_PASSWORD	YES(1)		Mariadb password	string
DB_NAME	YES(1)		Mariadb database	string
ADMIN_TOKEN	YES(2)		Admin token	string
CLIENT_TOKENS	YES(2)		Client tokens: comma separated	string
CORE_TOKEN	NO		Core token	string
CACHE_ENABLED	NO		if set, the cache server will be used	enum: * redis * memcached
CACHE_URI	YES(3)	localhost:11211	memcached connection url	string
		redis://localhost:6379	redis connection url	
CACHE_OPTIONS	NO		Optional cache parameters	string
REQUIRE_SIGNED_KEY	NO		Accept only signed keys	tinyint (0/1)
CLUSTER_MODE	NO		Enable features for cluster env	tinyint (0/1)
LOG_AUTH_KEYS_URL	NO		Enable remote log of successful requests	string
LOG_AUTH_KEYS_TOKEN	NO		Authorization Bearer for LOG_AUTH_KEYS_URL	string

(1) Mandatory if DB_ENGINE=mariadb

(2) Mandatory if CORE_TOKEN is not set

(3) Mandatory if CACHE_ENABLED=memcached or CACHE_ENABLED=redis

NOTE It's possible to save the variables in a *.env* file in the project's root

2. settings.json

It possibile to use `settings.json` file in the project's root to load **theo** configuration.

```
{
  "admin": {
    "token": "ch4ng3Me"
  },
  "client": {
    "tokens": [
      "njknsjd2412fnjkasnj",
      "knkjnknfjfnjenkln"
    ]
  },
  "sqlite": {
    "path": "./data/theo.db"
  },
  "server": {
    "http_port": 8890
  },
  "cache": {
    "type": "memcached",
    "settings": {
      "uri": "localhost:11211",
      "options": false
    }
  }
}
```

Run

```
$ npm start
```

2.3 theo-cli installation

2.3.1 NPM

```
$ npm i -g theoapp-cli
```

2.3.2 Sources

Clone repo

```
$ git clone https://github.com/theoapp/theo-cli.git
```

Install dependencies

```
$ npm install
```

Build

```
$ npm run build
```

2.3.3 Configuration

theo-cli needs 2 variables to work: *THEO_URL* and *THEO_TOKEN*. They can be set as environment variables:

```
THEO_URL=https://your.server.name THEO_TOKEN=your_secret_admin_token theo accounts␣
↪list
```

Or they can be stored in a file:

```
THEO_URL=https://your.server.name
THEO_TOKEN=your_secret_admin_token
```

theo-cli will look at (in this order):

```
$PWD/.env
$HOME/.theo/env
/etc/theo/env
```

2.4 Sign SSH keys

2.4.1 Use authorized key signature

Storing authorized key' signature along with the authorized key, let **theo-agent** to verify it before returning it to sshd. This will guarantee you that no one in any case will be able to inject unsolicited authorized keys and consequently get access to your server.

2.4.2 Setup

First, you need to create private/public keys, we'll use `openssl`

```
openssl genrsa 4096 | openssl pkcs8 -topk8 -v2 aes-256-cbc -out private.pem
```

It will prompt you to insert a pass phrase, memorize it!

now we need to extract the public key (we will use it with *theo-agent* to verify the signatures)

```
openssl rsa -in private.pem -pubout -out public.pem
```

It will ask you the pass phrase to unlock the private key.

The public key has to be copied on all the servers where **theo-agent** will run. *See theo-agent VERIFY*

2.4.3 Configure

To enable signing, you could set 2 variables: *THEO_PRIVATE_KEY* and *THEO_PRIVATE_KEY_PASSPHRASE*.

You can do it in 2 ways:

- Adding them as environment variables while executing *theo*.
- Adding them to the config file (the first file found will be used):

```
$PWD/.env
$HOME/.theo/env
/etc/theo/env
```

THEO_PRIVATE_KEY must point to your private key (use full path). *THEO_PRIVATE_KEY_PASSPHRASE* is the pass phrase to unlock the private key.

Since theo-cli 0.9.0 it's possible to pass private key path and passphrase as arguments.

```
--certificate, -c       Path to private key                             [string]
--passphrase, -p        passphrase for private key                      [string]
--passphrase-stdin, -i  read passphrase for private key from stdin     [boolean]
```

2.4.4 Usage

When adding a new authorized key to a user, to let *theo* signs the SSH public key add the *–sign* flag

```
theo keys add john.doe@example.com \
    --sign \
    --key "ssh-rsa AAAAB3NzaC1yc2E[...]7xUw== john.doe@laptop"
```

```
theo keys add john.doe@example.com \
    --passphrase-stdin \
    --sign \
    --key "ssh-rsa AAAAB3NzaC1yc2E[...]7xUw== john.doe@laptop"
```

```
theo keys add john.doe@example.com \
    --passphrase your-passphrase \
    --sign \
    --key "ssh-rsa AAAAB3NzaC1yc2E[...]7xUw== john.doe@laptop"
```

```
theo keys add john.doe@example.com \
    --passphrase-stdin \
    --certificate $HOME/private/theo-private.pem
    --sign \
    --key "ssh-rsa AAAAB3NzaC1yc2E[...]7xUw== john.doe@laptop"
```

```
theo keys add john.doe@example.com \
    --passphrase your-passphrase \
    --certificate $HOME/private/theo-private.pem
    --sign \
    --key "ssh-rsa AAAAB3NzaC1yc2E[...]7xUw== john.doe@laptop"
```

Since *theo-cli 0.10.0*, if you prefer to get the signature yourself (using OpenSSH or other tool) you can pass it to theo with the *–signature* argument

```
theo keys add john.doe@example.com \
    --key "ssh-rsa AAAAB3NzaC1yc2E[...]7xUw== john.doe@laptop" \
    --signature "81db52ca9a0d6d2[...]31a62663c0ce0a38c24cd7"
```

2.5 theo-cli usage

2.5.1 Accounts

```
theo accounts <command>

Manage accounts

Commands:
    theo accounts add [options]               Create account
    theo accounts rm <id>                     Remove account
    theo accounts edit <id> [options]         Edit account
    <group>
    theo accounts get <id>                    Get account
    theo accounts list                        List accounts
    theo accounts mod <id> [options]          Change account status
    theo accounts search                      Search accounts
```

List

```
theo accounts list

List accounts

Options:
  --version     Show version number                                    [boolean]
  --help        Show help                                              [boolean]
  --limit, -l   Number of accounts to retrieve                          [number]
  --offset, -o  Offset of the query                                     [number]
```

Search

```
theo accounts search

Search accounts

Options:
```

(continues on next page)

(continued from previous page)

```
  --version     Show version number                                    [boolean]
  --help        Show help                                              [boolean]
  --name, -n    Account name                                            [string]
  --email, -e   Account email                                           [string]
  --limit, -l   Number of accounts to retrieve                          [number]
  --offset, -o  Offset of the query                                     [number]
```

Get

```
theo accounts get <id>

Get account

Options:
  --version  Show version number                                       [boolean]
  --help     Show help                                                 [boolean]
```

Add/Create

```
theo accounts add [options]

Create account

Options:
  --version    Show version number                                     [boolean]
  --help       Show help                                               [boolean]
  --name, -n   Account name                                  [string] [required]
  --email, -e  Account email                                 [string] [required]
  --expire, -x Set account expiration (0 no expire). Use ISO 8601 date format
               (ex 2018-10-31)                                          [string]
```

Change status/expiration date

```
theo accounts mod <id> [options]

Edit account

Options:
  --version      Show version number                                   [boolean]
  --help         Show help                                             [boolean]
  --enable, -e   Enable Account                                        [boolean]
  --disable, -d  Disable Account                                       [boolean]
  --expire, -x  Set account expiration (0 no expire). Use ISO 8601 date format
                 (ex 2018-10-31)                                        [string]
```

Remove

```
theo accounts rm <id>
```

(continues on next page)

(continued from previous page)

```
Remove account

Options:
  --version  Show version number                                    [boolean]
  --help     Show help                                              [boolean]
```

Edit

```
theo accounts edit <id> [options] <group>

Edit account

Options:
  --version  Show version number                                    [boolean]
  --help     Show help                                              [boolean]
  --add, -a  Add account to group                                   [boolean]
  --rm, -d   Remove account from group                              [boolean]
```

2.5.2 Groups

```
theo groups <command>

Manage accounts

Manage groups

Commands:
  theo groups add [options]        Create group
  theo groups rm <id>              Remove group
  theo groups edit <id> [options]  Edit group
  theo groups get <id>             Get group
  theo groups list                 List groups
```

List

```
theo groups list

List groups

Options:
  --version     Show version number                                 [boolean]
  --help        Show help                                           [boolean]
  --limit, -l   Number of groups to retrieve                         [number]
  --offset, -o  Offset of the query                                  [number]
```

Get

```
theo groups get <id>

Get group

Options:
   --version  Show version number                                      [boolean]
   --help     Show help                                                [boolean]
```

Add

```
theo groups add [options]

Create group

Options:
  --version   Show version number                                      [boolean]
  --help      Show help                                                [boolean]
  --name, -n  Group name                                     [string] [required]
```

Change status

```
theo groups mod <id> [options]

Edit group

Options:
  --version     Show version number                                    [boolean]
  --help        Show help                                              [boolean]
  --action, -a  Action: enable|disable                       [string] [required]
```

Remove

```
theo groups rm <id>

Remove group

Options:
  --version  Show version number                                       [boolean]
  --help     Show help                                                 [boolean]
```

Edit

```
theo groups edit <id> [options] <account..>

Add/remove account(s) to/from group

Options:
  --version  Show version number                                       [boolean]
  --help     Show help                                                 [boolean]
```

(continues on next page)

(continued from previous page)

```
  --add, -a   Add accounts to group                                   [boolean]
  --rm, -d    Remove accounts from group                              [boolean]
```

2.5.3 SSH Keys

```
theo keys <command>

Manage accounts' keys

Commands:
  theo keys add <account> [options]     Add key to account
  theo keys import <account> [options]  Imporrt keys to account from a
                                           service (github/gitlab)
  theo keys rm <account> [options]      Remove key from account
```

Add

```
theo keys add <account> [options]

Add key to account

Options:
        --version            Show version number                          [boolean]
        --help               Show help                                    [boolean]
    -k, --key                Public ssh key                     [string] [required]
    -s, --sign               sign Public ssh key with private key. (Needs
                             THEO_PRIVATE_KEY env (or -c) and
                             THEO_PRIVATE_KEY_PASSPHRASE env (or -p / -i))[boolean]
    -c, --certificate        Path to private key                           [string]
    -p, --passphrase         passphrase for private key                    [string]
    -i, --passphrase-stdin   read passphrase for private key from stdin   [boolean]
    -g, --signature          Public ssh key' signature                     [string]
    -o, --ssh-options        SSH options                                   [string]
```

See examples for *–ssh-options* syntax

Edit

```
theo keys edit <account> [options]

Update SSH options for an account's key

Options:
    --version      Show version number                                [boolean]
    --help         Show help                                          [boolean]
-k, --key          Public ssh key ID                                 [required]
-o, --ssh-options  SSH options                               [string] [required]
```

See examples for *–ssh-options* syntax

Import

```
theo keys import <account> [options]

Imporrt keys to account from a service (github/gitlab)

Options:
  --version       Show version number                                  [boolean]
  --help          Show help                                            [boolean]
  --service, -s   Service to import from                     [string] [required]
  --username, -u  Service's username                         [string] [required]
```

Remove

```
theo keys rm <account> [options]

     Remove key from account

     Options:
       --version  Show version number                                  [boolean]
       --help     Show help                                            [boolean]
       --key, -k  Public ssh key ID                                   [required]
```

2.5.4 Permissions

```
theo permissions <command>

Manage accounts' permissions

Commands:
  theo permissions add <account>          Add permission to account       [options]
  theo permissions rm <account>           Remove permission from account  [options]
```

Add

```
theo permissions add [options]

     Add permission to account or group

     Options:
       --version      Show version number                              [boolean]
       --help         Show help                                        [boolean]
       --account, -a  Account id                                        [string]
       --group, -g    Group id                                          [string]
       --host, -h     Host name                              [string] [required]
       --user, -u     User name                              [string] [required]
```

Remove

```
theo permissions rm <account> [options]

     Remove permission from account

     Options:
       --version         Show version number                                  [boolean]
       --help            Show help                                            [boolean]
       --permission, -p  Permission ID                                       [required]
```

Search

```
theo permissions search [options]

     Check accounts by permissions

     Options:
       --version   Show version number                                        [boolean]
       --help      Show help                                                  [boolean]
       --host, -h  Host name                                        [string] [required]
       --user, -u  User name                                        [string] [required]
```

2.5.5 Authorized Keys

Fetch authorized keys

```
theo authorized_keys [options]

     Test authorized_keys

     Options:
       --version   Show version number                                        [boolean]
       --help      Show help                                                  [boolean]
       --host, -h  Host name                                        [string] [required]
       --user, -u  User name                                        [string] [required]
```

2.5.6 Examples

To create a new account with name *john.doe* and email *john.doe@sample.com*

```
$ THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    accounts add \
    --name john.doe \
    --email john.doe@sample.com

+-----------------------------------+
{
   "id": 1,
   "name": "john.doe",
   "email": "john.doe@sample.com",
   "active": 1,
   "public_keys": [],
```

(continues on next page)

(continued from previous page)

```
   "permissions": []
}
+---------------------------------+
```

To create a new account with name *Gary Cooper* and email *gary.cooper@sample.com* that will expire on Dec, 31 2018:

```
$ THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    accounts add \
    --name john.doe \
    --email john.doe@sample.com \
    --expire "2018-12-31"

+---------------------------------+
{
   "id": 1,
   "name": "john.doe",
   "email": "john.doe@sample.com",
   "expire_at": 1546214400000,
   "active": 1,
   "public_keys": [],
   "permissions": []
}
+---------------------------------+
```

To add a new key to account *john.doe* (Id 1):

```
$ THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    keys add john.doe@sample.com \
    -k "ssh-rsa AAAAB3N[.....]lS03D7xUw== john.doe@localhost"

  +----------------------------------------------------------------+
  {
     "account_id": "1",
     "keys": [
        {
           "key": "ssh-rsa AAAAB3N[.....]lS03D7xUw== john.doe@localhost"
        }
     ]
  }
  +----------------------------------------------------------------+
```

To add a new key with signature to account *john.doe* (Id 1):

```
$ THEO_PRIVATE_KEY="/home/macno/sign/private.pem" \
    THEO_PRIVATE_KEY_PASSPHRASE="abcd" \
    THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    keys add john.doe@sample.com \
    -k "ssh-rsa AAAAB3N[.....]lS03D7xUw== john.doe@localhost"
    -s

  +----------------------------------------------------------------+
  {
     "account_id": "1",
     "keys": [
        {
```

(continues on next page)

(continued from previous page)

```
            "key": "ssh-rsa AAAAB3N[.....]lS03D7xUw== john.doe@localhost",
            "signature": "1f01a031462da939ded812c9371e[...]b9c18ef6"
        }
    ]
  }
  +-----------------------------------------------------------------+
```

To import *John Doe*'s public keys from his github account (which is *jdoe80*):

```
THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    keys import john.doe@sample.com -s github -u jdoe80


+-----------------------------------------------------------------+
{
   "account_id": 1,
   "public_keys": [
      {
         "id": 8,
         "public_key": "ssh-rsa AAAAB3[....]aRcd099sfCzz"
      },
      {
         "id": 9,
         "public_key": "ssh-rsa AAAAB3[.....]lSasfd3ds=="
      }
   ]
}
+-----------------------------------------------------------------+
```

To add a new permission to *john.doe* to let him login as user *ubuntu* to host *srv-sample-01*

```
THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    permissions add \
    --account john.doe@sample.com \
    --host srv-sample-01 \
    --user ubuntu

+---------------------+
{
   "account_id": "1"
}
+---------------------+
```

To give permission to login as user *ubuntu* on all the servers named *test-xxxx*:

```
THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    permissions add \
    --account john.doe@sample.com \
    --host "test-%" \
    --user ubuntu
```

To create a new group *developers*

```
THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    groups add --name developers
```

To add *john doe* to *developer* group

```
THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    groups edit developers --add john.doe@sample.com
```

To grant access as user *deploy* on server *dev01* to group *developers*:

```
THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    permissions add \
    --group developers \
    --host "dev01" \
    --user deploy
```

To check who has access to server *dev01* with user *ubuntu*:

```
THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    permissions search \
    --host dev01
    --user ubuntu
```

SSH Options argument is a JSON string:

```
THEO_URL=http://localhost:9100 THEO_TOKEN=12345 theo \
    keys edit john.doe@sample.com \
    -k 20 --ssh-options '{"from": ["192.168.1.200"]}'
```

JSON schema

```
{
    "from": {
        "type": "array",
        "items": {
            "type": "string"
        }
    },
    "environment": {
        "type": "array",
        "items": {
            "type": "string"
        }
    },
    "command": {
        "type": "string"
    },
    "restrict": {
        "type": "boolean"
    },
    "agent-forwarding": {
        "type": "boolean"
    },
    "port-forwarding": {
        "type": "boolean"
    },
    "pty": {
        "type": "boolean"
    },
    "user-rc": {
        "type": "boolean"
    },
```

(continues on next page)

(continued from previous page)

```
    "X11-forwarding": {
        "type": "boolean"
    },
    "no-agent-forwarding": {
        "type": "boolean"
    },
    "no-port-forwarding": {
        "type": "boolean"
    },
    "no-pty": {
        "type": "boolean"
    },
    "no-user-rc": {
        "type": "boolean"
    },
    "no-X11-forwarding": {
        "type": "boolean"
    }
}
```

- if *restrict* is false (default) only *no-** properties are evaluated
- if *restrict* is true, only *agent-forwarding*, *port-forwarding*, *pty*, *user-rc*, *X11-forwarding* are evaluated

2.6 Installation

ATTENTION: OpenSSH must be version 6.2 or higher

2.6.1 1. Download one of the binaries for your system:

```
THEO_AGENT_LATEST=$(curl -L -s -H 'Accept: application/json' https://github.com/
↪theoapp/theo-agent/releases/latest |sed -e 's/.*"tag_name":"\([^"]*\)".*/\1/')
sudo curl -L -o /usr/sbin/theo-agent \
    https://github.com/theoapp/theo-agent/releases/download/${THEO_AGENT_LATEST}/theo-
↪agent-$(uname -s)-$(uname -m)
```

2.6.2 2. Make it executable

```
sudo chmod 755 /usr/sbin/theo-agent
```

2.6.3 3. Create a Theo Agent user:

```
sudo useradd \
    --comment 'Theo Agent' \
    --shell /bin/false \
    --system \
    theo-agent
```

2.6.4 4. Install

4.1. Full Automatic install

ATTENTION!!!

This command will:

- disable tunneled clear text passwords (no more user/password login!) [1]
- disable users' .ssh/authorized_keys
- set theo-agent as unique source for authorized_keys

We suggest to keep an open session until you're sure everything works as expected

```
sudo theo-agent -install \
    -no-interactive \
    -sshd-config \
    -url ${THEO_URL} \
    -token ${THEO_CLIENT_TOKEN}
```

[1] You can leave your PasswordAuthentication option unchanged adding the `-with-password-authentication` flag

4.2. Semi-Automatic install

```
sudo theo-agent -install \
    -no-interactive \
    -url ${THEO_URL} \
    -token ${THEO_CLIENT_TOKEN}
```

Edit `/etc/ssh/sshd_config` as suggested

4.3. Semi-manual install

```
sudo theo-agent -install
```

Answer to the questions and edit `/etc/ssh/sshd_config` as suggested

4.4. Manual install

ATTENTION: with OpenSSH older than 6.9 jump to section 4.5

Create a `config.yml` file (default is */etc/theo-agent/config.yml*):

```
url: THEO_URL
token: THEO_CLIENT_TOKEN
```

Create a cache directory (default is */var/cache/theo-agent*):

```
mkdir /var/cache/theo-agent
chmod 755 /var/cache/theo-agent
chown theo-agent /var/cache/theo-agent
```

Modify `/etc/ssh/sshd_config` (if you changed the default path, add the options to the command)

```
PasswordAuthentication no
AuthorizedKeysFile /var/cache/theo-agent/%u
AuthorizedKeysCommand /usr/sbin/theo-agent [-config-file /path/to/config.
↪yml] [-cache-path /path/to/cache/dir] %u
AuthorizedKeysCommandUser theo-agent
```

4.5. Manual install with OpenSSH older than 6.9

OpenSSH older than 6.9 does not support passing arguments to the command set with *Authorized-KeysCommand*, you must use the default values:

Create a `config.yml` file in */etc/theo-agent/config.yml*:

```
url: THEO_URL
token: THEO_CLIENT_TOKEN
cachedir: /var/cache/theo-agent
```

Create a cache directory */var/cache/theo-agent*:

```
mkdir /var/cache/theo-agent
chmod 755 /var/cache/theo-agent
chown theo-agent /var/cache/theo-agent
```

Modify `/etc/ssh/sshd_config`

```
PasswordAuthentication no
AuthorizedKeysFile /var/cache/theo-agent/%u
AuthorizedKeysCommand /usr/sbin/theo-agent
AuthorizedKeysCommandUser theo-agent
```

2.6.5 5. Restart openssh

```
sudo systemctl restart ssh.service
```

2.6.6 6. SELinux

If you're on a system with SELinux enabled (You can check it with: *getenforce*), you must switch sshd to permissive mode:

```
sudo semanage permissive -a sshd_t
```

2.7 Options

2.7.1 1. Installation

You can pass these arguments with `-install`

-no-interactive	It will use the value read from the arguments or it will use defaults
-config-file /path/to/config-file.yaml	It will use this path as config file
-user <value>	It will use <value> for executing theo-agent (default theo-agent)
-verify	It will set "verify: True" in configuration file
-public-key /path/to/public.key	It will add the path to the public key in configuration file
-cache-path /path/to/cache/dir	It will add the path to the cache directory in configuration file
-sshd-config	It will update sshd_config for you
-sshd-config-path /path/to/sshd_config	It will change this file if -sshd-config (default /etc/ssh/sshd_config)
-sshd-config-backup	It will make a copy of your sshd_config
-with-password-authentication	if -sshd-config, it will not change PasswordAuthentication value in sshd_config
-with-use-dns	if -sshd-config, it will set UseDNS to true in sshd_config This is needed if you will use hostnames/FQDN in 'from' authorized_keys options
-hostname-prefix <value>	It will set "hostname-prefix: <value>" in configuration file. The value will be prepend to hostname when querying theo server
-hostname-suffix <value>	It will set "hostname-suffix: <value>" in configuration file. The value will be append to hostname when querying theo server

2.7.2 2. Execution

theo-agent will accept these arguments (you can add them in sshd_config only if you have OpenSSH equal or greater than 6.9)

-config-file /path/to/config-file.yaml	It will use this path as config file
-verify	It will verify SSH public key signatures
-public-key /path/to/public.key	It will use this the public key to verify signatures
-cache-path /path/to/cache/dir	It will use this path as cache directory
-hostname-prefix <value>	The value will be prepend to hostname when querying theo server
-hostname-suffix <value>	The value will be append to hostname when querying theo server
-fingerprint <value>	It will send the value of SSH key fingerprint to the server You need to configure it in *sshd_config* in this way: `AuthorizedKeysCommand /usr/sbin/theo-agent -fingerprint %f %u`

2.8 Configuration

Full configuration example

```
url: https://example.authkeys.io
token: 132411349981792jkwqhqlwer4132345234
verify: True
public_key: /etc/theo-agent/public.pem
cachedir: /var/cache/theo-agent
hostname-prefix: dovm-
hostname-suffix: -test
timeout: 3000
```

2.9 Verify SSH keys

2.9.1 Use authorized key signature

If you store authorized key' signature along with the authorized key, **theo-agent** is able to verify it before returning it to sshd. This will guarantee you that no one, in any case, will be able to inject unsolicited authorized keys and consequently get access to your server.

2.9.2 Setup

Follow *theo-cli guide* to create private/public key. Then copy only the public key to the server where **theo-agent** will run.

2.9.3 Configure

With the `-verify` flag on, or `verify:  True` in config file, **theo-agent** will use the public key indicated in the `/etc/theo-agent/config.yml` to verifiy each signature.

`public_key` property in config.yml can be a path to a public key file, a public key, or an array

This is an example of *config.yml* using path value:

```
url: https://keys.sample.com
token: XXXXXXXX
verify: True
public_key: /etc/theo-agent/public.pem
```

This is an example of *config.yml* using an array of public keys paths

```
url: https://keys.sample.com
token: XXXXXXXX
verify: True
public_key:
- /etc/theo-agent/public.pem
- /etc/theo-agent/public2.pem
```

Here's an example og using embed public key

:: url: https://keys.sample.com token: XXXXXXXX verify: True public_key: |

> ——BEGIN PUBLIC KEY—— MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAvz9gLQyHy7EAdh6NggPc
> IIe0oHKLxUrim4Lhlq/xN1Eg+g5Iq9NSnMTBfiC4VS207X0G76tygOerh1m9ReqL
> FqUmaBB2g0OupcayKgJttKMC1jxRD6TWrvXVTHIgYya3FGD6/yBOiNztJccRgkbg
> pSLe5Nd3n8piJmVvaAmIjQmrAqGzV02Axjv/WY+qFVPvdG+YG4O18ypMJvdPAlBR

z6A9V7J1DHQNZTCWVPbjGTS0LZtgJeUXuKTQlNikqBz88IzTNDRpbMaUa2DCxzvL
U4N5qfhZjiVoYzEZtf91iXPC+Yl+Pj4yfZBxTMEgILWwWSZwd9M2EiHtJ+KDLcsb
lnoTKSJfmeu8pgBXvaFA8usBNi2sACPclwNDUq5CG4APmP3/AHKPfzR+3BLUTQ+s
wzT8AIJENRF4jMPzxcCUW4M2SjLElNan1F8lZdp8XwxNRlAOe6gBfyidAwGFRfmC
Wn1Y9x/sLyCBQG0FCGvuCvqxSVOLnYf4T0N5tK9OUJkaJvMnA98i0DvgYM6TbVjB
ULTiuVkBZ75qCicwMcfEvNN2SewYW2zZeJtnWDpZKXIEuv+ifG7mRXRzK75cIDsW
AFD/GwweoEG9WPf62um7BpKC3ewd+nERIjGrag+/+3OF8IW/xlicIVMtw+L9ZQ0T
o881E5rKe5WzEX90LbTD3xMCAwEAAQ== ——END PUBLIC KEY——

`/etc/ssh/sshd_config` must include the `-verify` flag in `AuthorizedKeysCommand`:

```
[...]
AuthorizedKeysCommand /usr/sbin/theo-agent -verify
AuthorizedKeysCommandUser theo-user
[...]
```

Remember to reload sshd

CHAPTER 3

Indices and tables

- genindex
- modindex
- search