copr-keygen Documentation

Release 1.52

Valentin Gologuzov

Apr 24, 2019

Contents

CHAPTER 1

About

Keygen is an auxiliary component in the Copr build system. It generate gpg key-pair, which will be used by obs-signd to sign built packages.

CHAPTER 2

Content

2.1 An outline of package sign process in Copr

To sign packages we decided to use obs-signd 1. Unfortunately it doesn't manage user keys in any way, but it's possible to minimize Copr operations with gpg key-pairs.

2.1.1 Expected setup

host-sign: secure machine where key-pairs is stored in /usr/share/copr-keygen/gnupg/ it runs:

- [A] perl signd from *obs-signd*
- [B] copr-keygen service

host-build: backend where builds occurs and result rpms are signed by invocation of /bin/sign [C] from *obs-signd*

[C] is configured by /etc/sign.conf to access [A] at host-0

When user `foo` builds first package, service [B] will be invoked to generate new keys (they will be contained in the keyring in **GPGHOMEDIR**). Also it creates dummy file into **PHRASESDIR**, that file indicates that user `foo` exists for [A]. Finally [A] can sign packages for user `foo` without receiving keys through network.

copr-backend do everything related to sign through new module **backend.sign** which either runs [C] or calls [B].

2.1.2 Configuration notes

At host-build

sign should be executed from the root user
/etc/sign.conf:

- server: host of machine with signd

ensure that configs/sudoers/copr_signer is copied into /etc/sudoers.d/

At host-sign

/etc/sign.conf:

- allow: list of backend hosts
- phrases: /var/lib/copr-keygen/phrases – location of **PHRASESDIR**

NB: obs-signd always run as root and doesn't accept alternative **GPGHOMEDIR**. To overcome this obstacle we added `/usr/bin/gpg_copr` Bash script wrapper which calls `gpg2` with correct user and homedir

2.2 Installation instructions

NB: will be changed after addition of *copr-keygen* and *obs-signd* to Fedora repos.

At first we need to provide obs-signd packakge, which right now is provided only for f21:

```
wget https://kojipkgs.fedoraproject.org//packages/obs-signd/2.2.1/4.fc21/x86\_64/obs-
↪signd-2.2.1-4.fc21.x86\_64.rpm
sudo yum localinstall obs-signd-2.2.1-4.fc21.x86\_64.rpm -y
```

Install keygen service itself:

```
sudo yum install copr-keygen
```

Next we need to copy config for httpd:

```
cp /usr/share/copr-keygen/httpd/copr-keygen.conf.example /etc/httpd/conf.d/
```

Copy config for signd and edit allowed hosts:

```
cp -f /usr/share/copr-keygen/sign/sign.conf/example /etc/sign/conf
```

Enable services and run them:

```
systemctl enable signd httpd haveged
systemctl start signd httpd haveged
```

2.3 Autodoc

2.3.1 HTTP endpoints

`POST /gen_key`

Generates new key-pair

Example request:

```
POST /gen_key HTTP 1.1
Content-Type: application/json

{
```

(continues on next page)

(continued from previous page)

```
  "name_real": "foo_bar",
  "name_email": "foo_bar@example.com"
}
```

request fields:

- **name_real, name_email, name_comment**: for key identification
- **key_length**: now supports 1024 or 2048 bytes
- **expire**: [optional] key expire in days, default 0 means never

Return Http response with plain text content

Status Codes

- 201 Created – on success, returns empty data
- 200 OK – key already exists, nothing done
- 400 Bad Request – incorrect request
- 500 Internal Server Error – internal server error

GET /ping

Checks if server still alive

Status Codes

- 200 OK – server alive

2.3.2 Python modules

copr_keygen.logic.**create_new_key**(*app*, *name_real*, *name_email*, *key_length*, *expire=None*, *name_comment=None*)

Creates new key for user. WARNING! This method doesn't check for the key duplicity. :param app: Flask application object

Parameters

- **name_real** – name for key identification
- **name_email** – email for key identification
- **key_length** – length of key in bytes, accepts 1024 or 2048
- **expire** – [optional] days for key to expire, default 0 == never expire
- **name_comment** – [optional] comment for key

Returns (stdout, stderr) from *gpg* invocation

copr_keygen.logic.**ensure_passphrase_exist**(*app*, *name_email*)

Need this to tell signd server that *name_email* available in keyring Key not protected by passphrase, so we write *something* to passphrase file.

copr_keygen.logic.**get_passphrase_location**(*app*, *name_email*)

copr_keygen.logic.**user_exists**(*app*, *mail*)

Checks if the user identified by mail presents in keyring

Returns bool True when user present

Raises GpgErrorException

exception copr_keygen.exceptions.**BadRequestException**(**args*, ***kwargs*)

status_code = 400

exception copr_keygen.exceptions.**GpgErrorException**(**args*, ***kwargs*)

status_code = 500

exception copr_keygen.exceptions.**KeygenServiceBaseException**(**args*, ***kwargs*)

msg

status_code = 500

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

HTTP Routing Table

/gen_key

`POST /gen_key`, 4

/ping

`GET /ping`, 5

Python Module Index

C

Index