

Resign a GPG key for rpm

the reason why

In RHEL9 the rpm system no longer accepts SHA1 signed gpg keys. This is because OpenSSL, which is used in compilation of gpg, no longer accepts SHA1, and has a minimum requirement of SHA512.

The packages in a repository are signed with the SHA1 signed key, now we resign the gpg key it self but the signage on the package is still valid. (hopefully)

get the original keys

N.B you will also need the passphrase for the secret key.

On the system where the keys were originally made with `gpg --gen-key`.

List the private keys on the system: `gpg --list-secret-keys`

```
/root/.gnupg/pubring.gpg
-----
sec    2048R/C3FAC3BD 2019-02-09
uid                               EXAMPLEKEY
ssb    2048R/4AEACD3A 2019-02-09
```

Export the key with :

```
gpg --export-secret-keys C3FAC3BD >secret-key.gpg
```

Or copy the entire directory `~/.gnupg` to the other system and adjust the owner and group of the files.

Import the private keys

On the new system you need to import the private key, transport the exported key to the system you want to use it on.

First install some software you need:

```
dnf install -y pinentry
```

Then import the key with:

```
gpg --import secret-key.gpg
```

When prompted enter the passphrase or "pin" for the secret key.

```
Please enter the passphrase to import the OpenPGP secret key:
"EXAMPLEKEY"
```

2048-bit RSA key, ID 0447A2B8C3FAC3BD,
created 2016-02-09.

Passphrase: _____

<OK>

<Cancel>

```
gpg: key 0447A2B8C3FAC3BD: "EXAMPLEKEY" not changed
gpg: key 0447A2B8C3FAC3BD: secret key imported
gpg: Total number processed: 1
gpg:           unchanged: 1
gpg:       secret keys read: 1
gpg:   secret keys imported: 1
```

Resign the secret key

Do this by:

```
gpg --cipher-algo IDEA --cert-digest-algo sha512 --expert --edit-key secret-
key.gpg
```

There is NO WARRANTY, to the extent permitted by law.

Secret key is available.

```
sec  rsa2048/0447A2B8C3FAC3BD
      created: 2016-02-09  expires: never      usage: SC
      trust: unknown      validity: unknown
ssb  rsa2048/5CA7D2244AEACD3A
      created: 2016-02-09  expires: never      usage: E
[ unknown] (1). EXAMPLEKEY
```

```
gpg> sign
```

"RWSBUILD" was already signed by key 0447A2B8C3FAC2BD

Do you want to sign it again anyway? (y/N) y

```
sec  rsa2048/0447A2B8C3FAC3BD
      created: 2016-02-09  expires: never      usage: SC
      trust: unknown      validity: unknown
Primary key fingerprint: FF7E B743 48CB CA81 256B  28C7 0447 A2B8 C3FA C3BD

      EXAMPLEKEY
```

Are you sure that you want to sign this key with your
key "EXAMPLEKEY" (0447A2B8C3FAC3BD)

This will be a self-signature.

```
Really sign? (y/N) y
```

In the test popup, set the pin to the same it was before.

And save:

```
gpg> save
```

Check the signatures: `gpg --check-sigs`

```
gpg: checking the trustdb
gpg: no ultimately trusted keys found
/home/abel/.gnupg/pubring.kbx
-----
pub      rsa2048 2019-02-09 [SC]
          FF7EB74348BCA81256B28C70447A2B8C3FAC3BD
uid           [ unknown] EXAMPLEKEY
sig!3        0447A2B8C3FAC3BD 2019-02-09  EXAMPLEKEY
sig!3        0447A2B8C3FAC3BD 2022-06-23  EXAMPLEKEY
sub      rsa2048 2019-02-09 [E]
sig!         0447A2B8C3FAC3BD 2019-02-09  EXAMPLEKEY

gpg: 3 good signatures
```

Export the public key

```
gpg --output public.pgp --armor --export
```

WORK IN PROGRESS

check if an rpm is signed with the same key

download the public key from the repository: `curl https://<repourl>/gpg_key_content -o retrieved-gpg-pub.key`

display some content of the key: `gpg retrieved-gpg-pub.key`

```
gpg: WARNING: no command supplied. Trying to guess what you mean ...
pub      rsa2048 2020-02-09 [SC]
          FF7EB12345BCA81256B28C70447A2B8C3FAC3BD
uid           EXAMPLEKEY
sub      rsa2048 2020-02-09 [E]
```

list the signature from the rpm file: `rpm -qip --nosignature example.rpm | grep Signature`

```
Signature      : RSA/SHA2, Wed 15 Jun 2022 02:19:41 PM CEST, Key ID
0447a2b8c3fac3bd
```

the last 16 characters of the long HEX number in the gpg output should match the characters in the signature of the rpm (albeit lower case)

alternative method

After importing the public key : `gpg --list-keys`

```
/root/.gnupg/pubring.gpg
-----
pub    2048R/C3FAC3BD 2019-02-09
uid           EXAMPLEKEY
sub    2048R/4AEACD3A 2019-02-09
```

`rpm --checksig filename.rpm`

```
./filename.rpm:
  Header V4 RSA/SHA1 Signature, key ID c3fac3bd: OK
  Header SHA1 digest: OK (07ae83890795d12adc57c39d32bc7166ed4727a6)
  V4 RSA/SHA1 Signature, key ID c3fac3bd: OK
  MD5 digest: OK (378df6a644641074b949460c22bf941f)
```

As you can see the key ID should be the same as the number in the pub part of the gpg key, and are the same as the last 8 digits in the full key ID.

Bronnen

https://wiki.cdote.senecacollege.ca/wiki/Signing_and_Creating_a_Repository_for_RPM_Packages

<https://superuser.com/questions/1009623/gpg-tools-location-of-private-keys>

<https://www.redhat.com/sysadmin/rpm-gpg-verify-packages>

<https://github.com/rpm-software-management/rpm/issues/1977>

<https://unix.stackexchange.com/questions/537795/compare-key-id-of-rpm-package-with-key-fingerprint-of-rpm-gpg-key>

https://access.redhat.com/documentation/en-us/red_hat_satellite/6.4/html/content_management_guide/importing_custom_content

From:
<https://wiki.auriel.nl/> -

Permanent link:
https://wiki.auriel.nl/doku.php?id=werkinstructies:pgp_key_resign&rev=1655988381

Last update: **2022/06/23 14:46**

