Using PGP With GnuPG

BV

[2015-07-07 Tue 20:23]

In 2005 I made a PGP Key. This note documents various aspects of interacting with my key.

1 Keys

My new 2015 key. See next section for how I upgraded.

```
pub   4096R/97A51B47 2015-07-08
      Key fingerprint = F37E 5F67 394D 3592 5D37  EE66 5B00 71E0 97A5 1B47
uid                  Brett Viren <bv@bnl.gov>
sub   4096R/3F58C5C6 2015-07-08
```

My initial 2005 key.

```
$ gpg --fingerprint --list-keys 9D1D282C
pub   1024D/9D1D282C 2005-12-20
      Key fingerprint = 04A6 309A D560 D339 D027  2924 122C 47D9 9D1D 282C
uid                  Brett Viren <bv@bnl.gov>
sub   2048g/C26A51C6 2005-12-20
```

2 Key server

I'm told I should use an sks keyserver pool which is authenticated by a CA cert.

```
$ wget https://sks-keyservers.net/sks-keyservers.netCA.pem
$ openssl x509 -fingerprint -in sks-keyservers.netCA.pem|head -1
SHA1 Fingerprint=79:1B:27:A3:8E:66:7F:80:27:81:4D:4E:68:E7:C4:78:A4:5D:5A:17
```

gpg.conf:

```
keyserver hkps://hkps.pool.sks-keyservers.net
keyserver-options ca-cert-file=~/.gnupg/CA/sks-keyservers.netCA.pem
```

3 Upgrading

The 2005 key is a 1024D. I want a larger 4096R one.

3.1 Getting started

I mostly followed this post which seems to be what everyone follows. It starts by recommending a few `gpg.conf` settings. However, it looks like `SHA512` is more common today that the `SHA256` that this 2009 blog post recomends.

```
# default-key  9D1D282C
personal-digest-preferences SHA512
cert-digest-algo SHA512
default-preference-list SHA512 SHA384 SHA256 SHA224 AES256 AES192 AES CAST5 ZLIB BZIP2
```

3.2 Creating the new key

Thanks Debian. Create the key with `gpg --gen-key` selecting option:

```
$ gpg --gen-key
...
   (1) RSA and RSA (default)
```

No comment as that's apparently considered "bad". Meet the new me:

```
pub   4096R/97A51B47 2015-07-08
      Key fingerprint = F37E 5F67 394D 3592 5D37  EE66 5B00 71E0 97A5 1B47
uid                  Brett Viren <bv@bnl.gov>
sub   4096R/3F58C5C6 2015-07-08
```

Make it default in `gpg.conf`

```
default-key  97A51B47
```

3.3 Add my Gmail identity

I try to keep my work identity tied to my `bv@bnl.gov` email address and all the rest tied to my `brett.viren@gmail.com` address. I don't currently have GPG well integrated with Gmail, even when sending via GNUS, but I follow the guide and add this identity nonetheless.

```
$ gpg --edit-key 97A51B47

gpg> adduid
Real name: Brett Viren
Email address: brett.viren@gmail.com
Comment:
You selected this USER-ID:
    "Brett Viren <brett.viren@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o

You need a passphrase to unlock the secret key for
user: "Brett Viren <bv@bnl.gov>"
4096-bit RSA key, ID 97A51B47, created 2015-07-08


pub  4096R/97A51B47  created: 2015-07-08  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/3F58C5C6  created: 2015-07-08  expires: never       usage: E
[ultimate] (1)  Brett Viren <bv@bnl.gov>
[ unknown] (2). Brett Viren <brett.viren@gmail.com>

gpg> save
```

3.4 Sign new with old

This is so that the new me recognizes the old me.

```
$ gpg --default-key 9D1D282C --sign-key 97A51B47
```

3.5 Transition statement

It is recommended to write a transition statement and sign it with both keys.
The key components of such a statement are apparently:

- Affirm the transition is desired.
- List the key ID, date and fingerprint for the old and the new keys.
- State what you request others do with this transition information.

- For convenience of others, provide instructions on how to carry out the request.

My statements:

- `./transition-2005-to-2015.txt`, signed by old key, signed by new key

4 Subkeys

It seems that when the world wants to know about subkeys the world turns to the Debian subkey wiki page. Here is a more coloquial coverage of the issue with some background. Lots of good walk throughs at this ubuntu wiki page.

This here post gives the run down on the key types that `gpg -[kK]` shows

`sec` SECret key

`ssb` Secret SuBkey

`pub` PUBlic key

`sub` public SUBkey

Using `gpg --edit-key` prints a usage code

`S` signing

`C` certifying other signatures

`E` encrypting

`A` authentication

I'm going to start using pass and I (think I) want to use a dedicated encrypting subkey.

```
$ gpg --edit-key 0x5B0071E097A51B47
gpg> addkey
   ...
   (6) RSA (encrypt only)
Your selection? 6
What keysize do you want? (2048) 4096
```

```
Please specify how long the key should be valid.
         0 = key does not expire
...
Key is valid for? (0) 0
```