

Laboratory_10: Commit Signing in Git

Git's Default Behavior

In the command line, this requires executing the following commands:

```
git config --global user.email "santi@example.org"  
git config --global user.name "Santi Figuerola"
```

What happens when someone else uses your email address, and then pushes changes remotely?

```
git config --global user.email "santi@example.org"  
git commit -m "Santi did this."  
git push origin main
```

How to Sign Commits in Git

You can get a list of your GPG keys with:

```
gpg --list-keys
```

Generate key pair:

```
gpg --full-gen-key
```

We'll be using the full GPG fingerprint, which we can get with the command:

```
gpg --list-keys
```

You'll get output like the following:

```
pub  rsa4096 2021-05-23 [SC]
      C6656513A0F9B7B7F4E76389EF39187D04795745
uid          [ultimate] Santi Figuerola <santi@example.org>
sub  rsa4096 2021-05-23 [E]
```

You need to export the public-key so you can upload it to GitHub. We use the `--armor` argument to indicate that we want to export it in an [ASCII armored](#) format instead of binary. This writes the public-key to a file named `gpg-key.pub`.

```
gpg --export --armor C6656513A0F9B7B7F4E76389EF39187D04795745 > ./gpg-key.pub
```

How to Back Up Your Keys

You can export your private-key in the same way we exported the public-key, this writes the private-key to a file named `gpg-key.asc`:

```
gpg --export-secret-keys --armor
C6656513A0F9B7B7F4E76389EF39187D04795745 > ./gpg-key.asc
```

How to Enable Commit Signing

Then to enable signing all commits, set the `commit.gpgsign` setting using `git config`. This will make `git` commit sign commits by default.

```
git config --global commit.gpgsign true
```

If you have multiple GPG keys, or just for future reference, you may want to set `user.signingkey` as well. This will indicate specifically which key `Git` should use for signing to avoid ambiguity.

```
git config --global user.signingkey C6656513A0F9B7B7F4E76389EF39187D04795745
```

How to Use your Key

Finally, you have to upload your public key. You can use the same GPG key for both GitHub and GitLab, or any other Git provider.

Account settings

Profile

Account

Appearance New

Account security

Billing & plans

Security log

Security & analysis

Emails

Notifications

Scheduled reminders

SSH and GPG keys

GPG keys / Add new

Key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Swi9xT2RtnxZbNskCvHtjbk3cBo2zWIIWL+oQ9t78X4pMTdm33V+4K0YTkOagIIA
MD24MqDQLON95uWleu+ETgxMV9o64P9qVVAzyRoFssM8lNHNERwqVFqk0EjknvW
zxeycevsM7XxDMAb+oxs59zfpVB8Kxkk5Dy23aFjajSfoC2o67X2RK0mese6tdZ
o12xvWWvIO2DWkniTrJ9ve0Fm+HV7SabpilGz7mM7LNzHSqyewmrh2Nikjo0lWuyr
bqvmwzrvLv1TFTcem9585JjX+MeudPMD8pk6x2wLDhikw9JSpihRaTnBAWChUYu
YjkAyu0nVDgA2o4wKa+YlgwT3wGypi5Q7HQJtG13Ox2VUzCC7uxePigHjC12rExp
OI+CErlllFCN+ZhxYKKSZK3OpoDNuZ+YRcGYmgpzSjll0oUK/o2D5YIU-
-OYgG
-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key