GPG (GNU Privacy Guard) aka PGP

...

Terminology

- Key: a chunk of data that you can use to encrypt or certify data
- Private Key: the secret part of the key that only you hold, and that you never share (used for decryption and signing)
- Public Key: the public part of the key that you share (used for encryption)
- Keyserver: A public directory of keys

What is it?

- A means of verifying who something came from
  - Or who approved it

- A means of encrypting data for only one recipient

How is it useful?

On the Internet... No one knows you're a potato...

No One.

From thebloggingpotatoes2015.blogspot.com which might be distributing malware

How is it useful?

- Web of Trust
  - Not the browser extension
  - Know who the people you trust trust, so you can trust those other people too
    - This gets quite extensive
- Verify who...
  - Published a file
  - Made a commit
  - Made a software release
  - Sent an email
- Send something to someone that only they can decrypt

Who uses it?

- Technical people
  - It's not particularly intuitive so there's a barrier to entry

- That aside, anyone who wants to
  - Maybe even you! Package managers use GPG signatures

A few very important things about keys, before we make one

Keep your private keys close...

- **DO NOT** publish, share, or upload your private key for any reason
  - Anyone who holds this can impersonate you and intercept messages intended only for you

- **Back up your private key** and revocation certificate
  - If you lose it, you're screwed
  - Physical media in a safe deposit box is a good option

- **DO NOT** handle your public keys through shared computers or services
  - Yes, keybase.io is cool. No, you shouldn't let them touch your private key.

“I know them from the Internet” is not good enough

- Signing someone else’s key is serious business, it means you know who they are and trust them
  - It also means you’re saying other people can trust them
  - It means that you’ve verified who they are, *in person*

- If you aren’t sure of someone’s identity or if you can trust them, *don’t sign their key*
  - Otherwise, the web of trust loses meaning because your friends can no longer trust who you trust

Know how and when to revoke your keys

- Revoking a key means that you...
  - Had the private key stolen from you or compromised, or uploaded to a public place
  - Have a new key and will no longer use the old one
  - You lost the password (and have a revocation certificate)

- Revoking a key is PERMANENT
  - An expired key is not revoked because you can “unexpire” it

- If you have the private key and password, you can revoke it by editing the key
  - If you don’t, you need your revocation certificate

Creating a Key

From the very beginning...

- Install gnupg through your package manager
  - Run gpg --help to make sure it's set

- If you're using a shared computer, feel free to make a key, but don't make it your actual key

Setting up

- Add these lines to ~/.gnupg/gpg.conf (they tell gpg to use stronger hashes, the default isn't great)

```
personal-digest-preferences SHA512

cert-digest-algo SHA512

default-preference-list SHA512 SHA384 SHA256 SHA224 AES256 AES192 AES CAST5 ZLIB BZIP2 ZIP Uncompressed
```

Generating a key

- Run `gpg2 --full-gen-key` and follow the prompts
  - What kind of key? 4 (RSA sign only)
  - What size? 4096 (the max for RSA)
  - Pick an expiration (note that expired keys can be unexpired)
  - Verify and say yes
  - Put in your real name, email, etc (you can add additional emails later)
  - Set your password (you can change it, but you cannot lose it)
  - Do lots of other things until it finishes generating (this may take a few minutes)

So now we have a super limited key...

- Edit your key to add encryption and signing, and any other email addresses
  - `gpg --edit-key <your key id>`
    - $ addkey
      - Do it for a signing key and an encryption key (twice)
    - $ adduid
      - Add all your other emails and names

Last Steps

- Generate a revocation certificate and save it somewhere safe
- (Optional) Separate off your secret key from your other keys
  - Put it somewhere safe
  - You can use your encryption and signing keys, but you can’t sign other keys without it

Upload your public key to a keyserver

- These stay up in the GPG network permanently so only do this if you're keeping the key you made
- `gpg2 --keyserver pgp.mit.edu --send-key <your key id>`
  - Get your key ID by running `gpg2 --list-secret-keys` and finding your key in the list

Signing someone's key

- This validates you trust them and know who they are, and that the key is theirs
- Download their key: `pgp --recv-keys <their key id>`
- Sign their key with your key `pgp --sign-key <their key id>`
- Push their key to the keyserver
    - Alternatively, detach and email the signature to them so they can verify and upload it themselves