GPG Keysigning

Matthew Walster, Fastly
UKNOF34, 21 April 2016

What is GPG?

- Making encryption practical
- Making signing practical
- Making the internet safer

Objective for this presentation

UKNOF wants you to:

- Create a PGP (GPG) Key, if you haven't already
- Sign a key someone else has, and have them sign yours
- Create a web-of-trust for the networking community

GPG:
How secure do you want to be?

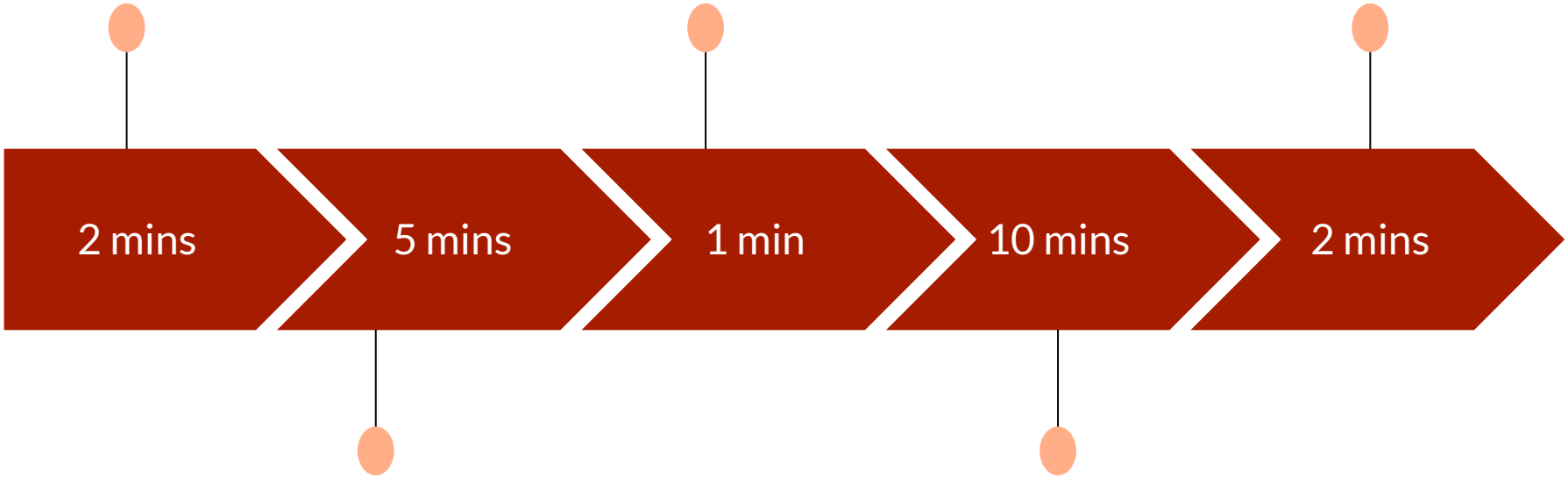
- Key length:
  - 2048-bit vs 4096-bit
- Key Expiry:
  - 1y vs 5y vs Never!
- Verifying Others
- Appropriate Usage
- SSH Keys!

Making a new GPG Key

Create your key:
Offline and as securely
as possible, please!

(Optionally) Generate
a revocation certificate

Sign someone's key
using "pius" or "caff"



(Optionally) Create
subkey for encryption
and authentication

(Optionally) Upload
your keys to a card

DON'T BE A HERO!

Don't use:

- Anything but RSA Keys
  - Your crazy ECC keys will probably be useless to others, especially on GPG <2.1
- Anything but SHA2 hashing
- A bigger keysize than needed
  - 4096-bit is only 16% stronger than 2048-bit
- A card to store your key unless you are going to use it -- it's a one-way process!

Do use:

- **GPG v2.1** (or later)
- A brand new key if generated before 2010
- rsa4096 for your certify key
- rsa2048: signing subkey
- rsa2048: encryption subkey
- A healthy amount of skepticism about my recommended GPG options and configuration
- A Yubikey 4!

The following links are for study -- skip if you are following along!

The following links will be quickly passed through during the presentation.

You should not attempt to do this at your desk unless you're comfortable with what's going on.

REMEMBER: Don't be a hero. Use these settings unless you *really* know what you're doing.

I will make myself available to walk people through it after the keysigning.

How to create a “secure” keyring:

https://www.jclement.ca/articles/2015/gpg-smartcard/

STRONGLY recommended GPG configuration:

https://gist.github.com/dotwaffle/5e1fe99bf66f8711d6d15d6c56450f9b

Using your key

An important note

Don’t carry it with you!

If you want to do so, STRONGLY consider the use of a card system!

NEVER submit your secret key

Keysigning

The purpose of keysigning

I verify you are who you say you are.

Three steps of verification:

1. FACE vs NAME
2. KEY vs NAME
3. EMAIL vs KEY

If you don't do all three parts, you're cheating the system and weaking the web of trust -- so be sure to check it!

Others will certify how much they trust your checking process!

Casual vs. Extensive Checking

How we keysign at UKNOF

You need a signing sheet, and a pen. No computers allowed!

If the hashes match, no need to read out key hashes!

1. When your number comes up, say if your ID is correct.
2. Show your Government issued ID for people to check it.

There is no Step 3!

Signing other people's keys

If you're a masochist, use "caff" from signing-party.

If you want an easy life, use "pius":

https://www.phildev.net/pius/

% pius ${keyid} ${keyid} ${keyid}

HINT: Start gpg-agent!

DO NOT USE "gpg2 --sign-key ${keyid}"

Observing trust

```
17:55:27 [mwalster@mwmbp:~] % gpg2 --check-sigs 0xA506E6D4DD4D5088
pub   dsa1024/0xA506E6D4DD4D5088 2001-10-09 [SC]
      Key fingerprint = 13C1 6D03 EDE7 2851 4473  AA73 A506 E6D4 DD4D 5088
uid                  [  full  ] Jonathan Riddell <jriddell@ubuntu.com>
sig!3        0x2404ED3A6AAAA569 2005-07-12  Martin Meredith <martin@sourceguru.
net>
sig!         0xF7B2C1C1B345BDD3 2007-06-21  Neil McGovern <kaulkin@halon.org.uk>
sig!         0x8991FF62CCBCBE9A 2007-07-08  Andy Davidson <andy@nosignal.org>

<snip>

gpg: 41 good signatures
gpg: 300 signatures not checked due to missing keys
```

Confused?

Come along and observe what happens.

Many will stick around afterwards to sign others and help.

Please try to sign the keys before the next meeting!

SSH Keys

Signing SSH Keys... Meh.

I want to give you access to my server.

You sign your ssh key fingerprint and send it over.

This is hassle and you probably (should) have lots of keys.

Did you know GPG can do SSH?

If gpg-agent is started with “enable-ssh-support” it’ll work!

You need an [A] subkey, “A” for Authentication.

https://wiki.archlinux.org/index.php/GnuPG#SSH_agent

No need for “ssh-agent” any longer!

Works just like “ssh-agent”

```
16:57:38 [mwalster@mwmbp:~] % export SSH_AUTH_SOCK="${HOME}/.gnupg/S.gpg-agent.
ssh"

17:08:49 [mwalster@mwmbp:~] % ssh-add -l
2048 SHA256:sk8qqt/0r4a/51WYkjBpov8CtiV6mY1Ss2b+zDKwW1U cardno:000603810757 (RSA)

17:08:52 [mwalster@mwmbp:~] % ssh-add -L
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDrwd2gK6IF/MjK8f+1X2Y7J/jiYGpXNa1rKemYl+oPJvnecyDcNI
On9Mg9SOPsoUI9Swkydw0Z8GC5uh3T1ZGelJtPjFdVYzXijBvU1fDPEqNC8bN5tAkp9oR8Rd6PB1Pm2ukV
EYwKTSFAm2Y5XTMJBRN8pyuJfeMMWBp7164zrlC4a1ldghwQqr8mZyd1eytUTyupY0a7AB+lgX2s9kv47a
UvYk1qurybC9AJJEuDGEFDiPmEVwqGy4SZQTJnTEi/XVrJZPTcI6oTid/F0QvWtK4A+feQPCfCuKJ4FScr
uZUfxKTfU5IXhbPo0I46cHsIK4BDY4AZ6XzU2HPwSYLx cardno:000603810757
```

It’ll even work with ssh keys

Best of both worlds:

- Your GPG Key where you can
- Your SSH Key where you must

Even better, migrate all SSH keys to GPG!

The private key can live on your Yubikey, meaning you don’t need one key per machine! You did do that, right?

Review

GPG is useful

You can sign to prove you sent it.

You can encrypt to make it private.

You can authenticate yourself with your key.

You can improve the web of trust.

You can meet new people!

GPG Keysigning

Questions?

Matthew Walster, Fastly
UKNOF34, 21 April 2016