PGP Signing

Matthew Walster
Network Engineer, IX Reach
matthew.walster@ixreach.com

Why Sign?

- Kernel.org hacked
  - At *least* 17 days before detection (2011/08/28)
  - How do we know that the files haven’t been altered?
  - How do we know the signatures haven’t been faked?

CASE STUDY: Linux Kernel

- Firstly, download the files from either kernel.org, or your local mirror:

```
[dotwaffle@baud:staging/uknof20]$ ls
linux-3.0.tar.bz2  linux-3.0.tar.bz2.sign
```

- Now verify the signature matches the file:

```
[dotwaffle@baud:staging/uknof20]$ gpg --verify linux-3.0.tar.bz2.sign
gpg: Signature made Fri 22 Jul 2011 04:31:02 BST using DSA key ID 517D0F0E
gpg: requesting key 517D0F0E from hkp server pool.sks-keyservers.net
gpg: key 517D0F0E: public key "Linux Kernel Archives Verification Key <ftpadmin@kernel.org>" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:               imported: 1
gpg: Good signature from "Linux Kernel Archives Verification Key <ftpadmin@kernel.org>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: C75D C40A 11D7 AF88 9981  ED5B C86B A06A 517D 0F0E
```

What just happened?

- gpg checked the signature time/date, and discovered which key signed
- gpg saw I didn’t have the key, and automatically retrieved it for me
- gpg checked the signature and saw it was “good”.
- gpg warned me that I don’t yet trust that key

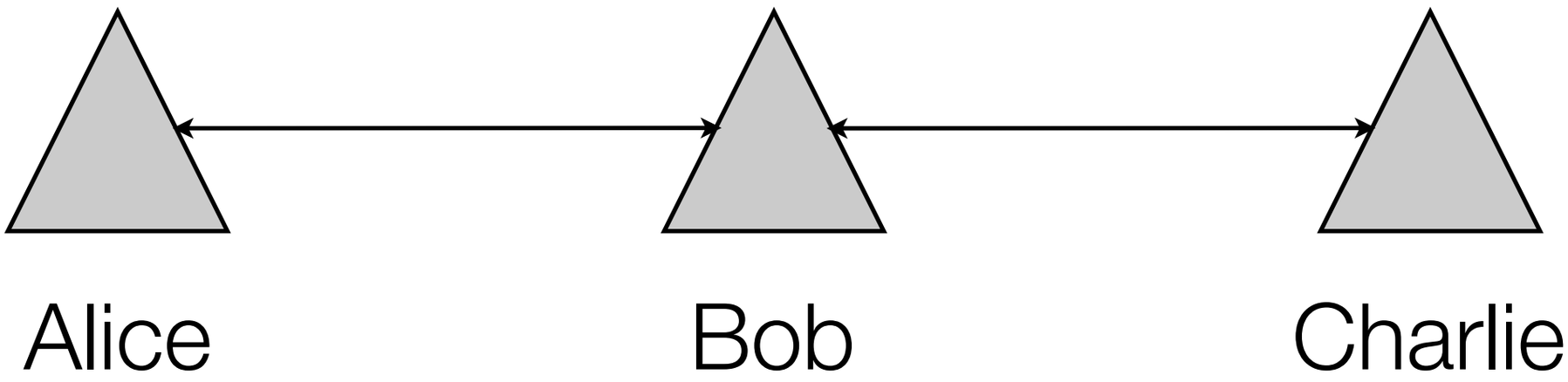
```
[dotwaffle@baud:staging/uknof20]$ gpg --verify linux-3.0.tar.bz2.sign
gpg: Signature made Fri 22 Jul 2011 04:31:02 BST using DSA key ID 517D0F0E
gpg: requesting key 517D0F0E from hkp server pool.sks-keyservers.net
gpg: key 517D0F0E: public key "Linux Kernel Archives Verification Key <ftpadmin@kernel.org>" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:               imported: 1
gpg: Good signature from "Linux Kernel Archives Verification Key <ftpadmin@kernel.org>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: C75D C40A 11D7 AF88 9981  ED5B C86B A06A 517D 0F0E
```

Trust?

- How can I trust this?
    - Who is “ftpadmin@kernel.org”?
- I can’t reasonably verify this is not a fake by itself.
    - Let’s assume for a moment I don’t know the SysOp at kernel.org...

The Web Of Trust

- Alice verifies that Bob is indeed Bob, and vice versa.
- Bob verifies that Charlie is indeed Charlie, and vice versa.
- If Charlie is satisfied that Bob will only verify people he knows are indeed who they say they are, Charlie can trust Bob that Alice is who she says she is.



Which Bob is right?

- I have dozens of people I trust - likewise, so does ftpadmin@kernel.org

- Non-trivial to find out the “best” path between us - we need a program!

sigtrace.pl

- Get it from http://www.chaosreigns.com/code/sigtrace/
- Requires you to have the path between you and your destination already
  - Quite a lot of effort preparing all the files, keys etc.

```
$ /usr/bin/time ./sigtrace.pl FAEBD5FC 0E9FF879
Data loaded, tracing....
level:0 keys:1 seconds:0
level:1 keys:76 seconds:0
level:2 keys:826 seconds:0
level:3 keys:2 seconds:0
4 hop path: FAEBD5FC 9D496584 F6F83318 80675E65 0E9FF879
FAEBD5FC Philip R. Zimmermann <prz@pgp.com>
9D496584 Network Associates TNS Division Employee Certification Key
F6F83318 Jason Bobier <jason@prismatix.com>
80675E65 Leonard D. Rosenthol <leonardr@lazerware.com>
0E9FF879 Darxus <Darxus@ChaosReigns.com>
```

- Isn’t there a really cool Web 1.0 solution to this problem?

pgp.cs.uu.nl

- Do a trace from 3AF59D16 to 517D0F0E...

PGP trust paths : Matthew Walster → Linux Kernel Archives Verification Key

from	**stats** Matthew Walster <matthew.at.walster.org>	3AF59D16
to	**stats** Linux Kernel Archives Verification Key <ftpadmin.at.kernel.org>	0B7BAE00
find	**reverse path**	trust paths reset
see also	paths **image** by wotsap	

0 3AF59D16 **stats** Matthew Walster <matthew.at.walster.org> #560 *signs*
1 C272A126 **stats** Andreas Jaeger <aj.at.suse.de> #185 *signs*
2 218D18D7 **stats** Robert Schiele <rschiele.at.gmail.com> #54 *signs*
3 517D0F0E **stats** Linux Kernel Archives Verification Key <ftpadmin.at.kernel.org> #1693

0 3AF59D16 **stats** Matthew Walster <matthew.at.walster.org> #560 *signs*
1 4813B5FE **stats** Andreas Scherbaum <ads.at.wars-nicht.de> #91 *signs*
2 1DFBA164 **stats** Bernhard Wiedemann <bernhardpgp.at.lsmod.de> #832 *signs*
3 517D0F0E **stats** Linux Kernel Archives Verification Key <ftpadmin.at.kernel.org> #1693

0 3AF59D16 **stats** Matthew Walster <matthew.at.walster.org> #560 *signs*
1 C139647C **stats** Ludovic Hirlimann (Normal Mail) <Ludovic.at.hirlimann.net> #430 *signs*
2 5D8CDA7B **stats** Guus Sliepen <guus.at.sliepen.eu.org> #300 *signs*
3 517D0F0E **stats** Linux Kernel Archives Verification Key <ftpadmin.at.kernel.org> #1693

0 3AF59D16 **stats** Matthew Walster <matthew.at.walster.org> #560 *signs*
1 20687895 **stats** Daniel Silverstone (DOB: 1980-04-09) <dsilvers.at.digital-scurf.org> #223 *signs*
2 A0B3E88B **stats** Martin Pool <mbp.at.sourcefrog.net> #670 *signs*
3 517D0F0E **stats** Linux Kernel Archives Verification Key <ftpadmin.at.kernel.org> #1693

Can we trust that file?

- I trust Daniel Silverstone.
- Daniel trusts Martin Pool.
- Martin trusts ftpadmin@kernel.org
- We can trust that the file was certified by ftpadmin@kernel.org
- NOTE: We can’t trust that the file is safe, just that it came from the right place!

How To Build The Web Of Trust

- Verify other people are who they say they are by “signing” their key
- Only do this if you’re *sure* - check their passport or other Government ID
- Verify their email address is valid by asking them to upload the signature
- If you upload it yourself, you haven’t done a proper check!

Signing

- I recommend using “caff” from debian signing-tools.
    - There is a sample caffrc supplied - easy to use
- You’ll need a working mail setup; if you want to smarthost, use ssmtp.
- I’ll regret this, but mail me if you need help!

Keysigning Party

- Here at UKNOF! Get your papers out!

MD5

7c07 fc6e a216 705d 1337 2d9d 8414 c64f

SHA1

b91f 99ae c642 9f93 b8cc
061a 8ffa 5e6u 43xf b70d

Paying attention?

MD5

7c07 fc6e a216 705d 120f 2d9d 8414 c64f

SHA1

b91f 99ae c642 9f93 b8cc
061a 8ffa 5e64 883b b70d

Signing

- At “Afternoon Tea” (15:15)
    - Gives people time to create keys if they want to join in
    - Write down the fingerprint from “gpg --fingerprint $MYKEY”

PGP Signing @ 15:15

Matthew Walster
Network Engineer, IX Reach
matthew.walster@ixreach.com