PGP keyid collisions

64 bits are kinda small

Evil 32

- evil32.com
- GPG did not verify to full fingerprint
- patched since June 2014
- GPG v1 still displays 32bit keyids

PGP keyid

- **last 64 bits of the PGP fingerprint**

```
gpg --keyid-format long --list-key sprecher.m@gmail.com
pub   rsa4096/BB773AC8D5836ABD 2011-09-18 [SCA]
      A5B48124C99B59CAB3548849BB773AC8D5836ABD
```

PGP fingerprint

- SHA1(timestamp + public key)
- with some extra seasoning

PGP fingerprint

```
00000000: 9900 8d04 2c85 36df 0104 00ae c406 0100
00000010: bda0 fbfe 94fb b305 13a8 a118 7ca1 86d3
00000070: 3baf dcab d02f 3224 f1d8 eee5 abbd 4c4b
00000080: 5a81 0f6a 31d8 378a 2423 8300 1101 0001
```

- ID, length (16bit big endian), version, timestamp (32bit big endian)
- algorithm, length public key (in bits, 16bit big endian), public key
- length exponent (in bits, 16bit big endian), exponent

bc1196ce6574ba12cb3a6a37b10ba5ed6bb70707

creating collision

- 64 bit is still a large number
- 18,446,744,073,709,551,616
- using a GTX 1080 would take ~36 years
- guestimated (don't take my word for it)
- what if we control both keys?

naive approach

1. generate many RSA keys
2. store the keyids in a lookup table
3. generate a new RSA key
4. check if the keyid in the lookup table
5. repeat step 3 and 4 until one matches

RSA is slow (compared to SHA1)

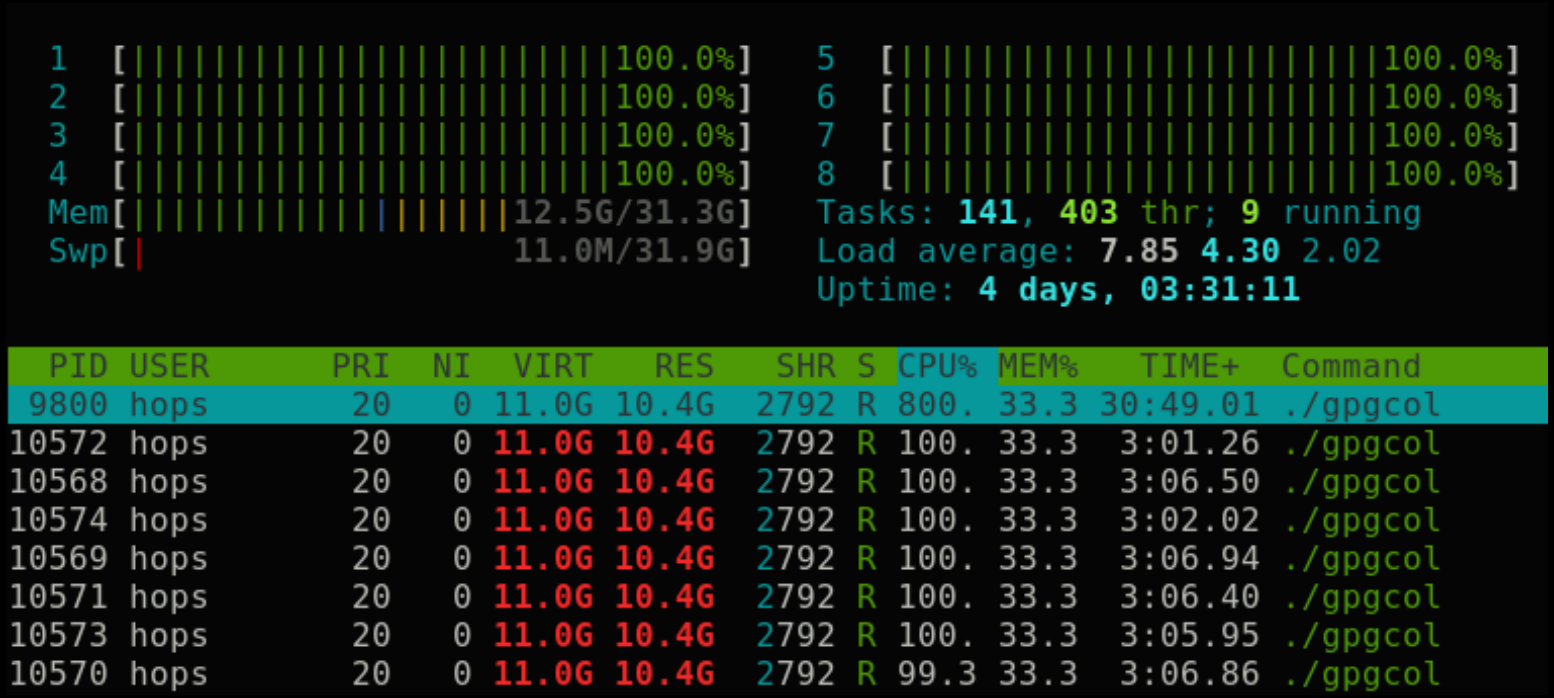
better approach

- timestamp is part of the fingerprint
- 2^32 fingerprints for each RSA key

POC

1. generate one RSA key (let's call it good)
2. change timestamp (1.1.2000 – 2.11.2018)
3. store keyid and timestamp in a Judy array
4. generate evil RSA key
5. loop through possible timestamps
6. check if keyid exists in Judy array
7. profit

POC


1 [||||||||||||||||||||||||100.0%]
2 [||||||||||||||||||||||||100.0%]
3 [||||||||||||||||||||||||100.0%]
4 [||||||||||||||||||||||||100.0%]
Mem[||||||||||||||||||12.5G/31.3G]
Swp[| 11.0M/31.9G]
5 [||||||||||||||||||||||||100.0%]
6 [||||||||||||||||||||||||100.0%]
7 [||||||||||||||||||||||||100.0%]
8 [||||||||||||||||||||||||100.0%]
Tasks: 141, 403 thr; 9 running
Load average: 7.85 4.30 2.02
Uptime: 4 days, 03:31:11
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
9800 hops 20 0 11.0G 10.4G 2792 R 800. 33.3 30:49.01 ./gpgcol
10572 hops 20 0 11.0G 10.4G 2792 R 100. 33.3 3:01.26 ./gpgcol
10568 hops 20 0 11.0G 10.4G 2792 R 100. 33.3 3:06.50 ./gpgcol
10574 hops 20 0 11.0G 10.4G 2792 R 100. 33.3 3:02.02 ./gpgcol
10569 hops 20 0 11.0G 10.4G 2792 R 100. 33.3 3:06.94 ./gpgcol
10571 hops 20 0 11.0G 10.4G 2792 R 100. 33.3 3:06.40 ./gpgcol
10573 hops 20 0 11.0G 10.4G 2792 R 100. 33.3 3:05.95 ./gpgcol
10570 hops 20 0 11.0G 10.4G 2792 R 99.3 33.3 3:06.86 ./gpgcol

got it!

```
gpu:  ~/src/gpgcol
→ gpg --homedir gpg-good --list-key --keyid-format LONG
gpg: WARNING: unsafe permissions on homedir `gpg-good'
gpg-good/pubring.gpg
---------------------
pub   1024R/5F7B7605CE786392 2005-07-14
uid                          good person <good@person.com>


gpu:  ~/src/gpgcol
→ gpg --homedir gpg-evil --list-key --keyid-format LONG
gpg: WARNING: unsafe permissions on homedir `gpg-evil'
gpg-evil/pubring.gpg
---------------------
pub   1024R/5F7B7605CE786392 2017-07-27
uid                          evil hacker <evil@hacker.com>
```

real impact?

- GPG won't import duplicate keyids
- GitHub and gitea do the same check
- «unnamed software» doesn't (POC pending)
- your idea?

Questions?