~~Bug 1176034 - (CVE-2020-25125) VUL-0: CVE-2020-25125: gpg2: AEAD preference list overflow~~

Status: RESOLVED FIXED

Classification: openSUSE
Product: openSUSE Tumbleweed
Component: Security
Version: Current
Hardware: Other Other

Priority: P1 - Urgent **Severity**: Critical (vote)
Target Milestone: ---
Assigned To: Pedro Monreal Gonzalez
QA Contact: Security Team bot

URL:
Whiteboard:
Keywords:

Depends on:
Blocks:
Show dependency tree / graph

- Create test case
- Clone This Bug

Reported: 2020-09-01 21:38 UTC by Andreas Stieger
Modified: 2021-02-22 20:15 UTC (History)
CC List: 5 users (show)

See Also:
Found By: ---
Services Priority:
Business Priority:
Blocker: ---

Attachments

Add an attachment (proposed patch, testcase, etc.)

Note
You need to log in before you can comment on or make changes to this bug.

Andreas Stieger 2020-09-01 21:38:45 UTC Description

```
Unable to import ed25519 key. It is partially imported but it's certificate/WOT is
not verified. With the key in the keyring, verifying a signature made with this key
fails.

An empty profile:

$ gpg --version -v
gpg (GnuPG) 2.2.21
libgcrypt 1.8.6
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: /home/[...]/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2


From https://www.gnupg.org/signature_key.html
fetch this key:


 > pub   ed25519 2020-08-24 [expires: 2030-06-30]
 >       Key fingerprint = 6DAA 6E64 A76D 2840 571B  4902 5288 97B8 2640 3ADA
 > uid   Werner Koch (dist signing 2020)


$ gpg --import *.keyring
gpg: /home/[...]/.gnupg/trustdb.gpg: trustdb created
gpg: key 249B39D24F25E3B6: public key "Werner Koch (dist sig)" imported
gpg: key 2071B08A33BD3F06: public key "NIIBE Yutaka (GnuPG Release Key)
<gniibe@fsij.org>" imported
gpg: key BCEF7E294B092E28: public key "Andre Heinecke (Release Signing Key)"
imported
free(): invalid pointer
Aborted (core dumped)

Unable to list:

$ gpg --list-keys
free(): invalid pointer
Aborted (core dumped)

However it is locally stored:

$ gpg --armor --export 6DAA6E64A76D2840571B4902528897B826403ADA
-----BEGIN PGP PUBLIC KEY BLOCK-----

mDMEX0PliRYJKwYBBAHaRw8BAQdAz75Hlekc16JhhfI0MKdEVxLdkxhcMCO0ZG6W
MBAmNpe0Hldlcm5lciBLb2NoIChkaXN0IHNpZ25pbmcgMjAyMCmImgQTFgoAQhYh
BG2qbmSnbShAVxtJAlKI17gmQDraBQJfQ+wlAhsDBQkShccRBQsJCAcCAyICAQYV
CgkICwIEFgIDAQIeBwIXgAAKCRBSiJe4JkA62nmuAP9uL/HOdB0gvwWrH+FpURJL
s4bnaZaPIk9ARrU0EXRgJgD/YCGfHQXpIPT0ZaXuwJexK04Z+qMFR/bMlq1Leo5C
jgY=
=ukul
-----END PGP PUBLIC KEY BLOCK-----


$ gpg --armor --export 6DAA6E64A76D2840571B4902528897B826403ADA > gpg2.keyring
$ osc service localrun source_validator
gpg: assuming signed data in '/home/[...]/gpg2/gnupg-2.2.22.tar.bz2'
```

```
gpg: Signature made Thu 27 Aug 2020 02:44:50 PM CEST
gpg:                using EDDSA key 6DAA6E64A76D2840571B4902528897B826403ADA
free(): invalid pointer
/usr/lib/obs/service/source_validators/20-files-present-and-referenced: line 169:
11415 Aborted                 (core dumped) gpg $GPG_OPTIONS --verify "$i"
(E) signature /home/[...]/gpg2/gnupg-2.2.22.tar.bz2.sig does not validate
```

Andreas Stieger 2020-09-01 22:53:07 UTC Comment 1

```
Base:System/gpg2 r247 2.2.20 good
Base:System/gpg2 r248 2.2.20 good
Base:System/gpg2 r250 2.2.21 bad
https://build.opensuse.org/request/show/819712
```

Pedro Monreal Gonzalez 2020-09-02 10:30:04 UTC Comment 2

```
I don't get the crash but I can see an invalid read of size 1 with valgrind. I'll
check its non of our patches before reporting upstream.
```

Pedro Monreal Gonzalez 2020-09-02 15:22:02 UTC Comment 3

```
Upstream confirmed. They will land a fix soon.
```

Pedro Monreal Gonzalez 2020-09-02 16:13:10 UTC Comment 4

```
Upstream fix: https://dev.gnupg.org/rG8ec9573e57866dda5efb4677d4454161517484bc
```

Andreas Stieger 2020-09-02 20:21:50 UTC Comment 5

```
Thank you for raising with upstream. I took the upstream commit and the issue
remains in my form (comment #). I understand that you were unable to reproduce. We
may be speaking about different issues.

adding some info below

openat(AT_FDCWD, "/home/[...]/.gnupg/pubring.kbx", O_RDONLY) = 5
lseek(5, 0, SEEK_CUR)                   = 0
fstat(5, {st_mode=S_IFREG|0644, st_size=3543, ...}) = 0
read(5, "\0\0\0 \1\1\0\2KBXf\0\0\0\0_O\374&_O\374&\0\0\0\0\0\0\0\0"..., 4096) =
3543
lseek(5, 0, SEEK_CUR)                   = 3543
lseek(5, 0, SEEK_CUR)                   = 3543
lseek(5, 0, SEEK_CUR)                   = 3543
lseek(5, 0, SEEK_CUR)                   = 3543
lseek(5, 0, SEEK_CUR)                   = 3543
writev(2, [{iov_base="free(): invalid pointer", iov_len=23}, {iov_base="\n",
iov_len=1}], 2free(): invalid pointer
) = 24
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fbff17ce000
rt_sigprocmask(SIG_UNBLOCK, [ABRT], NULL, 8) = 0
rt_sigprocmask(SIG_BLOCK, ~[RTMIN RT_1], [], 8) = 0
getpid()                                = 10090
gettid()                                = 10090
tgkill(10090, 10090, SIGABRT)           = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
--- SIGABRT {si_signo=SIGABRT, si_code=SI_TKILL, si_pid=10090, si_uid=1000} ---
+++ killed by SIGABRT (core dumped) +++
Aborted (core dumped)



$ gdb --quiet --args gpg --import *.keyring
Reading symbols from gpg...
Reading symbols from /usr/lib/debug/usr/bin/gpg2-2.2.22-269.1.x86_64.debug...
(gdb) r
Starting program: /usr/bin/gpg --import gpg2.keyring
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
gpg: key 249B39D24F25E3B6: "Werner Koch (dist sig)" not changed
gpg: key 2071B08A33BD3F06: "NIIBE Yutaka (GnuPG Release Key) <gniibe@fsij.org>" not
changed
gpg: key BCEF7E294B092E28: "Andre Heinecke (Release Signing Key)" not changed
free(): invalid pointer

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50        return ret;



#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
#1  0x00007ffff7acd539 in __GI_abort () at abort.c:79
#2  0x00007ffff7b27827 in __libc_message (action=action@entry=do_abort,
fmt=fmt@entry=0x7ffff7c36e2d "%s\n")
    at ../sysdeps/posix/libc_fatal.c:155
#3  0x00007ffff7b2eb2c in malloc_printerr (str=str@entry=0x7ffff7c3505c "free():
invalid pointer") at malloc.c:5347
#4  0x00007ffff7b30d76 in free_check (mem=0x55555566f360, caller=<optimized out>)
at hooks.c:255
#5  0x00007ffff7d10f95 in ?? () from /usr/lib64/libgcrypt.so.20
#6  0x00005555555582ad3 in free_user_id (uid=0x55555566f160) at free-packet.c:326
#7  free_user_id (uid=0x55555566f160) at free-packet.c:316
#8  0x00005555555832d8 in free_packet (pkt=0x55555566d780, parsectx=0x0) at free-
packet.c:468
#9  0x000055555558f87a in free_packet (parsectx=0x0, pkt=<optimized out>) at free-
packet.c:416
#10 release_kbnode (n=0x55555566d7e0) at kbnode.c:120
#11 0x00005555555891bc in get_pubkey_byfprint (ctrl=ctrl@entry=0x555555667820,
pk=pk@entry=0x0,
    r_keyblock=r_keyblock@entry=0x0,
    fprint=fprint@entry=0x7fffffffcd10
"m\252nd\247m(@W\033I\002R\210\227\270&@:\332\377\177",
    fprint_len=fprint_len@entry=20) at getkey.c:1796
#12 0x00005555555893a7 in get_pubkey_byfprint (fprint_len=20,
    fprint=0x7fffffffcd10 "m\252nd\247m(@W\033I\002R\210\227\270&@:\332\377\177",
r_keyblock=0x0, pk=0x0,
    ctrl=0x555555667820) at getkey.c:1767
#13 get_user_id_byfpr (ctrl=0x555555667820, fpr=0x7fffffffcd10
"m\252nd\247m(@W\033I\002R\210\227\270&@:\332\377\177",
    rn=rn@entry=0x7fffffffcba0) at getkey.c:4085
#14 0x000055555558940f in get_user_id_byfpr_native (ctrl=<optimized out>, fpr=
<optimized out>) at getkey.c:4098
```

```
#15 0x00005555555bdacb in import_one_real (ctrl=ctrl@entry=0x555555667820,
keyblock=<optimized out>,
    keyblock@entry=0x55555566d710, stats=stats@entry=0x5555556678a0,
fpr=fpr@entry=0x0, fpr_len=fpr_len@entry=0x0,
    options=options@entry=2048, from_sk=<optimized out>, silent=<optimized out>,
screener=<optimized out>,
    screener_arg=<optimized out>, origin=<optimized out>, url=<optimized out>,
r_valid=<optimized out>) at import.c:2258
#16 0x00005555555bdfa9 in import_one (ctrl=ctrl@entry=0x555555667820,
keyblock=0x55555566d710,
    stats=stats@entry=0x5555556678a0, fpr=fpr@entry=0x0, fpr_len=fpr_len@entry=0x0,
options=options@entry=2048, from_sk=0,
    silent=0, screener=0x0, screener_arg=0x0, origin=0, url=0x0, r_valid=0x0) at
import.c:2336
#17 0x00005555555bf76b in import (ctrl=ctrl@entry=0x555555667820,
inp=inp@entry=0x555555667930,
    fname=fname@entry=0x7fffffffe33b "gpg2.keyring",
stats=stats@entry=0x5555556678a0, fpr=fpr@entry=0x0,
    fpr_len=fpr_len@entry=0x0, options=<optimized out>, screener=<optimized out>,
screener_arg=<optimized out>,
    origin=<optimized out>, url=<optimized out>) at import.c:643
#18 0x00005555555c0e32 in import_keys_internal (ctrl=0x555555667820, inp=<optimized
out>, fnames=0x7fffffffdec8,
    nnames=<optimized out>, stats_handle=0x0, fpr=0x0, fpr_len=0x0, options=2048,
screener=0x0, screener_arg=0x0, origin=0,
    url=0x0) at import.c:539
#19 0x00005555555556b955 in import_keys (stats_handle=0x0, url=<optimized out>,
origin=<optimized out>,
    options=<optimized out>, nnames=<optimized out>, fnames=<optimized out>,
ctrl=0x555555667820) at import.c:575
#20 main (argc=<optimized out>, argv=<optimized out>) at gpg.c:4646
```

Werner Koch 2020-09-03 14:16:58 UTC Comment 6

Thanks for the new info. I am able to replicate the bug and preparing a new release.

Werner Koch 2020-09-03 17:15:19 UTC Comment 7

A new release is available. See

https://lists.gnupg.org/pipermail/gnupg-announce/2020q3/000448.html

Our upstream bug is https://dev.gnupg.org/T5050

Thanks to everyone here for tracking down this brown paper bag bug of mine.

Andreas Stieger 2020-09-03 17:23:07 UTC Comment 8

From https://dev.gnupg.org/T5050

Importing a key with AEAD preferences with GnuPG 2.2 can lead to an array overflow. This is not trivial to exploit because the attacker can control only each second byte with the first byte being fixed at 0x04. But it can be exploited.

Affected versions are GnuPG 2.2.21 and 2.2.22. GnuPG 2.3 and versions before 2.2.21 are not affected.

From https://lists.gnupg.org/pipermail/gnupg-announce/2020q3/000448.html

This version fixes a *critical security bug* in versions 2.2.21 and 2.2.22.

Importing an OpenPGP key having a preference list for AEAD algorithms will lead to an array overflow and thus often to a crash or other undefined behaviour.

Importing an arbitrary key can often easily be triggered by an attacker and thus triggering this bug. Exploiting the bug aside from crashes is not trivial but likely possible for a dedicated attacker. The major hurdle for an attacker is that only every second byte is under their control with every first byte having a fixed value of 0x04.

References:
https://dev.gnupg.org/T5050
https://dev.gnupg.org/rGaeb8272ca8aad403a4baac33b8d5673719cfd8f0

Andreas Stieger 2020-09-03 17:30:10 UTC Comment 9

https://build.opensuse.org/request/show/831935

OBSbugzilla Bot 2020-09-03 18:50:06 UTC Comment 10

This is an autogenerated message for OBS integration:
This bug (1176034) was mentioned in
https://build.opensuse.org/request/show/831939 Factory / gpg2

Andreas Stieger 2020-09-06 09:44:06 UTC Comment 11

will be in next snapshot
