pinentry-tty mishandles ctrl-C

Closed, Resolved | Public

Description

Run gpg to decrypt a file, configured to use pinentry-tty. Enter the passphrase wrong, and on the second try, type ctrl-C. gpg exits, and the terminal mode is screwed up. Characters don't echo, etc.

Details

Version: gnupg 2.2.4, pinentry 1.1.0, Ubuntu 18.04.2

Assigned To: gniibe

Authored By: maiden_taiwan, 2019-06-26 06:00:07 (UTC+2)

Tags: Bug Report; Testing (Backlog)

Subscribers: dkg, gniibe, gouttegd, gv, maiden_taiwan, werner

Related Objects

Task Graph | Mentions

Status	Assigned	Task
Open	None	T4771 pinentry-tty/pinentry-curses interact a user as background process
Resolved	gniibe	~~T4585~~ pinentry-tty mishandles ctrl-C
Resolved	gouttegd	~~T4659~~ Release Pinentry-1.1.1

maiden_taiwan created this task. 2019-06-26 06:00:07 (UTC+2)

gniibe claimed this task. 2019-06-26 10:10:14 (UTC+2)
gniibe triaged this task as *Normal* priority.
gniibe added a subscriber: **gniibe**.

Ah, yes, that signal thing should be handled correctly, when we support line edit by tty.

For master branch of GnuPG, it was fixed. See ~~T2011: gnupg should notify cancellation of its operation to gpg-agent to kill pinentry~~.

gniibe added a comment. 2019-06-26 10:16:37 (UTC+2)

I meant, GnuPG side was fixed in master, it sends SIGINT to pinentry process when gpg exits.

We need to fix pinentry side to receive SIGINT in order to recover TTY state (for curses and tty).

gniibe added a comment. 2019-06-28 02:49:55 (UTC+2)

Because my fix was incomplete, I pushed another change to GnuPG master:
rG374a0775546b: agent: Close a dialog cleanly when gpg/ssh is killed for CONFIRM.
I also pushed my changes to pinentry master: **rPf6e84ce0a34c: tty: Confirmation is not by line edit mode.**, **rP531b92300c58: tty: Support line editing by system.**, **rPb176a8ac0dcd: Exit the loop on an error with GPG_ERR_FULLY_CANCELED.**

The changes for pinentry is also related to ~~T4583: pinentry-tty should accept backspace, delete, and ctrl-U~~, fixing the support of line edit by the TTY discipline.

gniibe changed the task status from *Open* to *Testing*. 2019-06-28 02:50:11 (UTC+2)

gniibe mentioned this in **E506: Weekly Standup**. 2019-07-01 06:25:19 (UTC+2)

gv added a subscriber: **gv**. 2019-12-03 12:58:55 (UTC+1)

I'm sorry, this issue is far from fixed.

[testing@c8 ~]$ gpg2 --sign b.log
gpg: using "xxxxxxxx" as default secret key for signing
Please enter the passphrase to unlock the OpenPGP secret key:
"<stripped info>"
4096-bit RSA key, ID , <stripped info>
created <stripped info>.

Passphrase:
gpg: signal Interrupt caught ... exiting

[testing@c8 ~]$
<== The cursors is here!

After pinentry-tty is terminated using CTRL-C the terminal cursor is under the terminal prompt (i.e.: [testing@c8 ~]$). A new CTR-C or CTRL-L is required. Also in midnight commander the screen us garbled an reset command has to be run to have a proper interface back.
Same with pinentry-curses (if that matter). In this case bash prompt is not visible at all.

I use the latest pinentry (git) and gpg is properly patched (I'm using 2.2.9).

Thank you.

maiden_taiwan added a comment. 2019-12-03 14:02:25 (UTC+1)

@gv : I am another user (not the developer), but here is a workaround I found. Type **ctrl-D** instead of ctrl-C to terminate pinentry-tty.

It works only if ctrl-D is the first character typed on the line (so it's treated as an end-of-file marker). If you've already started typing your passphrase and want to quit, first press ENTER (so you're submitting an incomplete/wrong passphrase), and when gpg2 prompts you a second time, type ctrl-D.

gv added a comment. 2019-12-03 15:08:49 (UTC+1)

@maiden_taiwan Thank you. Nice trick. Works fine for for one file and covers almost all of my issues.
Still, for example, when used together with rpmsign and I have to sign multiple rpms files, is inconvenient to type ctrl-D for each rpm file (for whatever reason I want to stop the signing process) . ctrl-c just stop the process.
This worked fine with gpg 1.x. Not so much with gpg2.

dkg added a subscriber: **dkg**. 2019-12-03 22:46:42 (UTC+1)

`pinentry-tty` is pretty fragile, and designed to be handled in a particular way. I strongly recommend a different workflow if you're using gpg secret key operations in a regular process. either:

- have no passphrase on your secret key
- `--pinentry-mode=loopback --passphrase-…=…` (that is, supplying the passphrase to the process directly, read the manpage for various `--passphrase-…` options)
- a robust graphical-mode pinentry

gv added a comment. 2019-12-04 11:59:09 (UTC+1)

@dkg I use gnupg 1.x for a very, very long time. I like the way it works. And most, I like that the terminal is not hidden from me when I type a password and that the characters in password does not appear on terminal as "*". Sometime the text in terminal is important to me. pinentry-tty have more or less the same behavior as gnupg 1.x. With pinentry-curses the terminal is hidden and there are '*' for each character in password that I type. Also, there is not GUI on my servers so no pinentry-(qt|gtk|anything else).

For a long time, every year, I try to switch to gnupg2, and I can't. The _most_ important issue for me was/is ctrl-c that confuse the email clients that I use (text mode apps), Midnight Commander, break the terminal, etc. No, pressing "Cancel" is not an option for me (when using pinentry-curses). If I forget to do that when using the email app, I must open a new terminal, ssh to the remote server, kill the mail app and all text that I wrote is lost.

> *have no passphrase on your secret key*

That a joke, right? :-)

> *--pinentry-mode=loopback*

This has its own issues. For example, when using for something like this:
gpg2 --symmetric --output b.log.enc --armor --quiet --sign --interactive --no-emit-version b.log
I have to guess when the encryption / signing passwords should be typed. Saying "Enter passphrase: " is not very intuitive on what to do.
Works fine with gnupg 1.x

> *a robust graphical-mode pinentry*

If you mean GUI there is none. Not for me.

Up to the above series of patches, there was no way to switch from gnupg 1.x to gnupg 2. Now, with the help of gniibe, maybe I can. If he or someone else can fix the above issue it will be great.

maiden_taiwan added a comment. 2019-12-04 13:43:52 (UTC+1)

I agree with everything in the previous comment. Just hoping for simple, robust UI like gpg 1.x that works over an SSH connection (no GUI) for ordinary file decryption on the command line.

pinentry-tty seems to be the closest. If it handled ^C properly, it would be perfect!

gniibe added a comment. 2019-12-05 00:52:12 (UTC+1)

Please note that pinentry-tty/curses is a kind of emulation of CLI user interface, it's not the real one (I'm going to explain in the next paragraph).
It is, by any means, not robust, as users would expect, from the implementation's view. It only works specific simple use cases (while I do my best to stabilize it in master branch of GnuPG).

The mechanism of user I/O is like: gpg --> gpg-agent -> pinentry ==> TTY
... where gpg-agent and pinentry run in background.
Note that even if gpg is invoked by a parent process synchronously (foreground), pinentry runs as a background process.
Doing input/output by a background process is unusual thing in POSIX. I'd say, it's pretty much unusual in my experience since 80's.
It doesn't work well with an application which uses PTY for its sub-process invocation (like midnight commander), since assumption of TTY usage doesn't match.
It doesn't work well with Emacs, because Emacs does its own screen setting and it doesn't expect another background process does input/output.

It is the best to specify `--pinentry-mode=loopback` for your invocation of gpg, so that gpg (the foreground process) can directly ask you when passphrase is needed.
Then, it is gpg (the foreground process), which does user I/O.

Another option for tmux or GNU screen users would be to have a dedicated screen for pinentry-curses interaction. By writing a glue script, the screen pops up for user interaction.

gniibe added a comment. 2019-12-05 01:10:56 (UTC+1)

My message above is: The reported issue of ^C was fixed in pinentry-tty and GnuPG in master branch. Please test that fixes.

The fundamental issue discussed would be: screen usage (when gpg is invoked by main application like email client), mutual exclusion of TTY by multiple processes

gniibe added a parent task: **T4771: pinentry-tty/pinentry-curses interact a user as background process**. 2019-12-05 06:59:24 (UTC+1)

gniibe added a subtask: ~~T4659: Release Pinentry-1.1.1~~. 2019-12-05 07:20:43 (UTC+1)

maiden_taiwan added a comment. 2019-12-05 13:01:47 (UTC+1)

@gniibe - Thanks for your explanation. Is `--pinentry-mode=loopback` the same as specifying in *~/.gnupg/gpg-agent.conf*:

```
pinentry-program /usr/bin/pinentry-tty
allow-loopback-pinentry
```

If not, is there a way to force `--pinentry-mode=loopback` in *~/.gnupg/gpg-agent.conf*?

werner added a subscriber: **werner**. 2019-12-05 20:57:28 (UTC+1)

allow-loopback-pinentry in gpg-agent.conf is actually the default. This options advises gpg-agent to accept a request for a loopback-pinentry. If you would configure no-allow-loopback-pinentry, requests from gpg to use a loopback pinentry are rejected.

Although possible, you should not use pinentry-mode=loopback in gpg.conf. The reason is that other applications don't assume that and reply on a pinentry. Thus --pinentry-mode=loopback should only be used on the command line.

gv added a comment. 2019-12-06 11:18:39 (UTC+1)

@gniibe Thank you!

Pressing ctrl-c when asked for password when gpg2 is invoked with --pinentry-mode=loopback command line argument works (almost, see bellow) fine in terminal. When gpg2 is invoked by email client and I press ctrl-c also works. However, when i quit the email client the terminal is garbled. Something like this:
[gpg@c8 ~]$ alpine
gpg: using "XXXXXXXX" as default secret key for signing
Enter passphrase:
gpg: signal Interrupt caught ... exiting
Alpine finished -- Closed empty folder "INBOX"
[gpg@c8 ~]$ [gpg@c8 ~]$ [gpg@c8 ~]$ [gpg@c8 ~]$ [gpg@c8 ~]$ [gpg@c8 ~]$
(Enter key pressed 5 times).

Tested with gpg2 && pinentry from git.

This I may fix fit a simple script/alias that call /usr/bin/reset when email client quit.

The big problem that I see with --pinentry-mode=loopback is that you do not get a second chance to enter the password again in case you made a typing mistake. And, when used with symmetric encryption and signing (gpg2 --symmetric --output file.enc --armor --quiet --sign file.txt), the encryption password is requested only once and is not clear what password you have to type (encryption password/gpg key password). And if you make a mistake... well...
gpg2 --pinentry-mode=loopback ... is not usable this way (at least for me).

werner added a comment. 2019-12-06 15:34:01 (UTC+1)

In case you use gpgme we have a flag which can be queried to see whether a redraw is required:

@item "redraw"
This flag is normally not changed by the caller because GPGME sets and
clears it automatically: The flag is cleared before an operation and
set if an operation noticed that the engine has launched a Pinentry.
A Curses based application may use this information to redraw the
screen; for example:

```
err = gpgme_op_keylist_start (ctx, "foo@@example.org", 0);
while (!err)
  {
    err = gpgme_op_keylist_next (ctx, &key);
    if (err)
      break;
    show_key (key);
    gpgme_key_release (key);
  }
if ((s = gpgme_get_ctx_flag (ctx, "redraw")) && *s)
  redraw_screen ();
gpgme_release (ctx);
```

If you use gpg directly you need to watch out for a status line (see --status-fd):

```
PINENTRY_LAUNCHED <pid>[:<extra>]
 This status line is emitted by gpg to notify a client that a
 Pinentry has been launched.  <pid> is the PID of the Pinentry.  It
 may be used to display a hint to the user but can't be used to
 synchronize with Pinentry.  Note that there is also an Assuan
 inquiry line with the same name used internally or, if enabled,
 send to the client instead of this status line.  Such an inquiry
 may be used to sync with Pinentry
```

gniibe added a project: **Testing**. 2020-03-12 06:38:23 (UTC+1)

gouttegd added a subscriber: **gouttegd**. 2021-01-18 21:02:47 (UTC+1)

Any news about this bug? It has been in “Testing” for quite a while now. For what it’s worth, handling of ^C seems to work here as I would expect, so I am inclined to close here and let pinentry-1.1.1 go out. @gniibe , as you did the fix, do you have any comment?

gniibe closed this task as *Resolved*. 2021-01-19 03:01:13 (UTC+1)

For a bug which requires more tests (like this one with GnuPG and pinentry), I had a practice to put "Testing" tag.

For me, it's done. So, I'm closing.

gouttegd mentioned this in ~~T4583: pinentry-tty should accept backspace, delete, and ctrl-U~~. 2021-01-19 19:48:40 (UTC+1)

gouttegd closed subtask ~~T4659: Release Pinentry-1.1.1~~ as *Resolved*. 2021-01-23 23:22:19 (UTC+1)

Log In to Comment