Secure GPG In Emacs, Featuring Agent Smith

Aug 30, 2020 • emacs cryptography • 1 comment

Oh, no. More GPG in Emacs? I'll try to keep it short.

One day I was pondering upon my opsec ("operational security")[1] when I realized something. Any Emacs buffers I had opened with `epa`'s transparent decryption would stay open, indefinitely.

My mechanical keyboard.

Moreover, the password I had used during decryption would stay encached by `gpg-agent` for ten minutes (the default). If someone were to usurp my computer while a `.gpg` buffer was open, he could not only read it, but view all of my encrypted files.

Paranoid? Yes. *Excessively* paranoid? Yes.

Killing GPG Buffers

Step one is to make sure GPG buffers don't stay open too long. Put the following code somewhere in your init:

```
;; Adapted from https://stackoverflow.com/a/15854362/6085242.
(defun kill-gpg-buffers ()
  "Kill GPG buffers."
  (interactive)
  (let ((buffers-killed 0))
    (dolist (buffer (buffer-list))
      (with-current-buffer buffer
        (when (string-match ".*\.gpg$" (buffer-name buffer))
          (message "Auto killing .gpg buffer '%s'" (buffer-name buffer))
          (when (buffer-modified-p buffer)
            (save-buffer))
          (kill-buffer buffer)
          (setq buffers-killed (+ buffers-killed 1)))))
    (unless (zerop buffers-killed)
      ;; Kill gpg-agent.
      (shell-command "gpgconf --kill gpg-agent")
      (message "%s .gpg buffers have been autosaved and killed" buffers-killed))))

(run-with-idle-timer 60 t 'kill-gpg-buffers)
```

If Emacs doesn't receive input for 60 seconds your GPG buffers get saved and killed. This time frame is configurable and should factor in the sensitivity of your files and what you personally feel comfortable with.

This code will run even if your computer is locked, at least on OSX. If you unlock your computer and someone happens to be peeping over your shoulder, you're good.

Enter gpg-agent

Everyone has their own mental model for `gpg-agent` . I like to picture Agent Smith: the guy that nobody likes from the old 90's movie `The Memento` . Smug and annoying, his only goal is to ruin your life.

If you're using GPG2, you're using `gpg-agent`. There's nothing you can do about it: GPG2 launches it automatically. You can kill it with a cron job, but I did something a bit different. I snuck this line into the big code block above:

```
(shell-command "gpgconf --kill gpg-agent")
```

This makes sure that the cache is cleared when Emacs is idle, but not while you're actively working in it.

You may also want to set some of the relevant cache expiry options in `~/.gnupg/gpg-agent.conf`. You can set them to a few minutes, or zero to disable caching altogether. For example:

```
default-cache-ttl 120
max-cache-ttl 600
```

Conclusion

Told you I'd keep it short. GPG and Emacs' `epa` are software disasters, but we had to take a couple minutes to revisit them here.

Footnotes

[1]: Don't worry, I'll keep the cool hacker's lingo to a minimum. ↩

Related posts:

- GPG In Emacs

Filed under emacs cryptography

top ⇡

⇠ Upgrading The OSX Dock

Anki Add-Ons You Didn't Know You Needed ⇢

Note: anonymous comments require approval.

Login

Add a comment

M↓ MARKDOWN

□ COMMENT ANONYMOUSLY

ADD COMMENT

Powered by **Commento**