using-signed-tag-in-pull-request

From: Junio C Hamano <gitster@pobox.com> Date: Tue, 17 Jan 2011 13:00:00 -0800 Subject: Using signed tag in pull requests Abstract: Beginning v1.7.9, a contributor can push a signed tag to her publishing repository and ask her integrator to pull it. This assures the integrator that the pulled history is authentic and allows others to later validate it. Content-type: text/asciidoc

How to use a signed tag in pull requests
==========

A typical distributed workflow using Git is for a contributor to fork a project, build on it, publish the result to her public repository, and ask the "upstream" person (often the owner of the project where she forked from) to pull from her public repository. Requesting such a "pull" is made easy by the `git request-pull` command.

Earlier, a typical pull request may have started like this:

```
The following changes since commit 406da78032179...:

  Froboz 3.2 (2011-09-30 14:20:57 -0700)

are available in the Git repository at:

  example.com:/git/froboz.git for-xyzzy
```

followed by a shortlog of the changes and a diffstat.

The request was for a branch name (e.g. `for-xyzzy`) in the public repository of the contributor, and even though it stated where the contributor forked her work from, the message did not say anything about the commit to expect at the tip of the for-xyzzy branch. If the site that hosts the public repository of the contributor cannot be fully trusted, it was unnecessarily hard to make sure what was pulled by the integrator was genuinely what the contributor had produced for the project. Also there was no easy way for third-party auditors to later verify the resulting history.

Starting from Git release v1.7.9, a contributor can add a signed tag to the commit at the tip of the history and ask the integrator to pull that signed tag. When the integrator runs `git pull`, the signed tag is automatically verified to assure that the history is not tampered with. In addition, the resulting merge commit records the content of the signed tag, so that other people can verify that

the branch merged by the integrator was signed by the contributor, without fetching the signed tag used to validate the pull request separately and keeping it in the refs namespace.

This document describes the workflow between the contributor and the integrator, using Git v1.7.9 or later.

A contributor or a lieutenant ----------

After preparing her work to be pulled, the contributor uses `git` `tag` `-s` to create a signed tag:

```
$ git checkout work
$ ... "git pull" from sublieutenants, "git commit" your own work ...
$ git tag -s -m "Completed frotz feature" frotz-for-xyzzy work
```

Note that this example uses the `-m` option to create a signed tag with just a one-liner message, but this is for illustration purposes only. It is advisable to compose a well-written explanation of what the topic does to justify why it is worthwhile for the integrator to pull it, as this message will eventually become part of the final history after the integrator responds to the pull request (as we will see later).

Then she pushes the tag out to her public repository:

```
$ git push example.com:/git/froboz.git/ +frotz-for-xyzzy
```

There is no need to push the `work` branch or anything else.

Note that the above command line used a plus sign at the beginning of `+frotz-for-xyzzy` to allow forcing the update of a tag, as the same contributor may want to reuse a signed tag with the same name after the previous pull request has already been responded to.

The contributor then prepares a message to request a "pull":

```
$ git request-pull v3.2 example.com:/git/froboz.git/ frotz-for-xyzzy >msg.txt
```

The arguments are:

1. the version of the integrator’s commit the contributor based her work on;
2. the URL of the repository, to which the contributor has pushed what she wants to get pulled; and
3. the name of the tag the contributor wants to get pulled (earlier, she could write only a branch name here).

The resulting msg.txt file begins like so:

```
The following changes since commit 406da78032179...:

  Froboz 3.2 (2011-09-30 14:20:57 -0700)

are available in the Git repository at:

  example.com:/git/froboz.git tags/frotz-for-xyzzy

for you to fetch changes up to 703f05ad5835c...:

  Add tests and documentation for frotz (2011-12-02 10:02:52 -0800)

----------------------------------------------
Completed frotz feature
----------------------------------------------
```

followed by a shortlog of the changes and a diffstat. Comparing this with the earlier illustration of the output from the traditional `git request-pull` command, the reader should notice that:

1. The tip commit to expect is shown to the integrator; and
2. The signed tag message is shown prominently between the dashed lines before the shortlog.

The latter is why the contributor would want to justify why pulling her work is worthwhile when creating the signed tag. The contributor then opens her favorite MUA, reads msg.txt, edits and sends it to her upstream integrator.

Integrator ----------

After receiving such a pull request message, the integrator fetches and integrates the tag named in the request, with:

```
$ git pull example.com:/git/froboz.git/ tags/frotz-for-xyzzy
```

This operation will always open an editor to allow the integrator to fine tune the commit log message when merging a signed tag. Also, pulling a signed tag will always create a merge commit even when the integrator does not have any new commit since the contributor’s work forked (i.e. ’fast forward’), so that the integrator can properly explain what the merge is about and why it was made.

In the editor, the integrator will see something like this:

```
Merge tag 'frotz-for-xyzzy' of example.com:/git/froboz.git/

Completed frotz feature
# gpg: Signature made Fri 02 Dec 2011 10:03:01 AM PST using RSA key ID 96AFE6CB
# gpg: Good signature from "Con Tributor <nitfol@example.com>"
```

Notice that the message recorded in the signed tag "Completed frotz feature" appears here, and again that is why it is important for the contributor to explain her work well when creating the signed tag.

As usual, the lines commented with `#` are stripped out. The resulting commit records the signed tag used for this validation in a hidden field so that it can later be used by others to audit the history. There is no need for the integrator to keep a separate copy of the tag in his repository (i.e. `git tag -l` won't list the `frotz-for-xyzzy` tag in the above example), and there is no need to publish the tag to his public repository, either.

After the integrator responds to the pull request and her work becomes part of the permanent history, the contributor can remove the tag from her public repository, if she chooses, in order to keep the tag namespace of her public repository clean, with:

```
$ git push example.com:/git/froboz.git :frotz-for-xyzzy
```

Auditors --------

The `--show-signature` option can be given to `git log` or `git show` and shows the verification status of the embedded signed tag in merge commits created when the integrator responded to a pull request of a signed tag.

A typical output from `git show --show-signature` may look like this:

```
$ git show --show-signature
commit 02306ef6a3498a39118aef9df7975bdb50091585
merged tag 'frotz-for-xyzzy'
gpg: Signature made Fri 06 Jan 2012 12:41:49 PM PST using RSA key ID 96AFE6CB
gpg: Good signature from "Con Tributor <nitfol@example.com>"
Merge: 406da78 703f05a
Author: Inte Grator <xyzzy@example.com>
Date:   Tue Jan 17 13:49:41 2012 -0800

    Merge tag 'frotz-for-xyzzy' of example.com:/git/froboz.git/

    Completed frotz feature
```

```
    * tag 'frotz-for-xyzzy' (100 commits)
      Add tests and documentation for frotz
      ...
```

There is no need for the auditor to explicitly fetch the contributor’s signature, or to even be aware of what tag(s) the contributor and integrator used to communicate the signature. All the required information is recorded as part of the merge commit.

□□□ #2
□ □□□ □□□ 2026-10-31 20:15:22 UTC
□ □□□ □□□ 2026-10-31 20:15:23 UTC