2.30.7

Git v2.30.7 Release Notes ==========================

This release addresses the security issues CVE-2022-41903 and CVE-2022-23521.

Fixes since v2.30.6 ----------

- CVE-2022-41903:

```
git log has the ability to display commits using an arbitrary
format with its --format specifiers. This functionality is also
exposed to git archive via the export-subst gitattribute.

When processing the padding operators (e.g., %<(, %<|(, %>(,
%>>(, or %><( ), an integer overflow can occur in
pretty.c::format_and_pad_commit() where a size_t is improperly
stored as an int, and then added as an offset to a subsequent
memcpy() call.

This overflow can be triggered directly by a user running a
command which invokes the commit formatting machinery (e.g., git
log --format=...). It may also be triggered indirectly through
git archive via the export-subst mechanism, which expands format
specifiers inside of files within the repository during a git
archive.

This integer overflow can result in arbitrary heap writes, which
may result in remote code execution.
```

- CVE-2022-23521:

```
gitattributes are a mechanism to allow defining attributes for
paths. These attributes can be defined by adding a `.gitattributes`
file to the repository, which contains a set of file patterns and
the attributes that should be set for paths matching this pattern.

When parsing gitattributes, multiple integer overflows can occur
```

```
when there is a huge number of path patterns, a huge number of
attributes for a single pattern, or when the declared attribute
names are huge.

These overflows can be triggered via a crafted `.gitattributes` file
that may be part of the commit history. Git silently splits lines
longer than 2KB when parsing gitattributes from a file, but not when
parsing them from the index. Consequentially, the failure mode
depends on whether the file exists in the working tree, the index or
both.

This integer overflow can result in arbitrary heap reads and writes,
which may result in remote code execution.
```

Credit for finding CVE-2022-41903 goes to Joern Schneeweisz of GitLab. An initial fix was authored by Markus Vervier of X41 D-Sec. Credit for finding CVE-2022-23521 goes to Markus Vervier and Eric Sesterhenn of X41 D-Sec. This work was sponsored by OSTIF.

The proposed fixes have been polished and extended to cover additional findings by Patrick Steinhardt of GitLab, with help from others on the Git security mailing list.

Patrick Steinhardt (21): attr: fix overflow when upserting attribute with overly long name attr: fix out-of-bounds read with huge attribute names attr: fix integer overflow when parsing huge attribute names attr: fix out-of-bounds write when parsing huge number of attributes attr: fix out-of-bounds read with unreasonable amount of patterns attr: fix integer overflow with more than INT_MAX macros attr: harden allocation against integer overflows attr: fix silently splitting up lines longer than 2048 bytes attr: ignore attribute lines exceeding 2048 bytes attr: ignore overly large gitattributes files pretty: fix out-of-bounds write caused by integer overflow pretty: fix out-of-bounds read when left-flushing with stealing pretty: fix out-of-bounds read when parsing invalid padding format pretty: fix adding linefeed when placeholder is not expanded pretty: fix integer overflow in wrapping format utf8: fix truncated string lengths in `utf8_strnwidth()` utf8: fix returning negative string width utf8: fix overflow when returning string width utf8: fix checking for glyph width in `strbuf_utf8_replace()` utf8: refactor `strbuf_utf8_replace` to not rely on preallocated buffer pretty: restrict input lengths for padding and wrapping formats

---

□□□ #2
□ □□□ □□□ 2026-10-31 19:58:27 UTC
□ □□□ □□□ 2026-10-31 19:58:28 UTC