GNU libmicrohttpd SOS Fund Audit Fix Log

Identified Vulnerabilities

A: Use of a file descriptor before it is initialized, when compiled with all warnings on (Low)

Christian Grothoff writes: Issue A was a really transient issue (recently introduced before the audit revision, shortly afterwards fixed, and Ram already said that he also believes it is already fixed). So that one (being the "most serious") requires no action as the report already states.

VERIFIED

B: Use of unbounded string functions (Medium)

Fix:
https://gnunet.org/git/libmicrohttpd.git/commit/?id=ef49636130061c379821d60c58ef51468bf9e039

Christian Grothoff writes: Commit ef49636130061c379821d60c58ef51468bf9e039 replaces the remaining uses of sprintf with MHD_snprintf(), which has the added benefit of reducing the number of functions used from libc(). None of the changes would seem to be indicative of a real security issue; they are more preventative against future changes in the code that might cause problems (i.e. very long HTTP status code strings).

The proposed change to strcpy() are pretty useless. In the first case, the bounds check would still rely on correct calculation of size/off, which while not trivial here would not really improve the situation. Furthermore, this is in code that is performance-critical (executed for each HTTP reply) and would increase the code size.

Similarly, the second one is _after_ an explicit strlen()-check which is _already_ redundant as the code provably only passes strings of a length below the limit (which is also already documented to be like that). So replacing the strcpy() is really unnecessary, and trivial replacements (memcpy, strncpy()) are likely to introduce more subtle issues (off-by-one, 0-termination).

We may change these in the future to get rid of all implicit string-length computations for performance reasons.

vfprintf(): past security issues in libc remain security issues in libc. The suggestion to not use any libc function that ever had a security issue in its implementation is nonsense. Worse, the suggested replacement either requires dynamic memory allocation (bad, additional error

cases INSIDE logging!), truncating log messages, and/or large static buffers (bigger stacks / larger memory footprint) and are thus also all unacceptable / worse. So the proposed cure is way worse than the hypothetical disease.

VERIFIED

C: Whitespace RFC 7230 header rules noncompliance (Low)

Fix:
https://gnunet.org/git/libmicrohttpd.git/commit/?id=e95ec4874da57b153ecea27fa553ae8a19b4a280
Fix:
https://gnunet.org/git/libmicrohttpd.git/commit/?id=b57456c2e6536764ad7b065c70b999f876269a2c

Christian Grothoff writes: Commit e95ec487 enforces the no-whitespace rule, albeit for now only if MHD_USE_PEDANTIC_CHECKS is set in the options. Whether this should fall under PEDANTIC or not is still up for debate, but at least the diff makes it clear what needs to be changed.

b57456c2 makes the no-whitespace rule the default, adding a flag MHD_USE_PERMISSIVE_CHECKS to disable it and allow the previous semantics if explicitly desired by the application.

For the 0.9.54 release we ultimately decided to merge the PEDANTIC_CHECKS and PERMISSIVE_CHECKS flags into a new option “MHD_OPTION_STRICT_FOR_CLIENT”. But the fundamental logic change is in the commits mentioned above.

VERIFIED

Miscellaneous Issues

1: Improvements to daemon.c (info)

Both suggestions were already resolved in the meantime.

VERIFIED