libmicrohttpd

HTTP/1.x server C library (MHD 1.x, stable)

Log | Files | Refs | Submodules | README | LICENSE

---

commit 6d6846e20bfdf4b3eb1b592c97520a532f724238
parent 08f933a6ad2138ad0fad86496eb3ba88eaf3f173
Author: Christian Grothoff <christian@grothoff.org>
Date: Sun, 26 Feb 2023 17:51:24 +0100

fix parser bug that could be used to crash servers using the MHD_PostProcessor

Diffstat:

M	ChangeLog	14	+++++++++-----
M	src/microhttpd/postprocessor.c	2	+-

2 files changed, 10 insertions(+), 6 deletions(-)

---

diff --git a/ChangeLog b/ChangeLog

```
@@ -1,3 +1,7 @@
+Sun Feb 26 05:49:30 PM CET 2023
+    Fix potential DoS vector in MHD_PostProcessor discovered
+    by Gynvael Coldwind and Dejan Alvadzijevic. -CG
+
 Sun 26 Dec 2021 20:30:00 MSK
     Releasing GNU libmicrohttpd 0.9.75 -EG

@@ -23,7 +27,7 @@ December 2021
     Some code improvements for new test test_client_put_stop.
     Added special log message if thread creation failed due to system limits.
     Fully restructured new_connection_process_() to correctly handle errors,
-    fixed missing decrement of number of daemon connections if any error
+    fixed missing decrement of number of daemon connections if any error
     encountered, fixed app notification of connection termination when app has
     not been notified about connection start, fixed (highly unlikely) reset of
     the list of connections if reached daemon's connections limit.
@@ -67,7 +71,7 @@ November 2021
     for testing of MHD.
     Renamed 'early_response' connection flag to 'discard_request' and reworked
     handling of connection's flags.
-    Clarified request termination reasons doxy, fixed reporting of
+    Clarified request termination reasons doxy, fixed reporting of
     MHD_REQUEST_TERMINATED_READ_ERROR (previously this code was not really used
     in reporting).
     Enforce all libcurl tests exit code to be zero or one.
@@ -76,7 +80,7 @@ November 2021
     of the last LF in termination chunk, handle correctly chunk sizes with more
     than 16 digits (leading zeros are valid according to HTTP RFC), fixed
     handling of CRCR, LFCR, LFLF, and bare CR as single line delimiters, report
-    error when invalid chunk format is received without waiting to receive
+    error when invalid chunk format is received without waiting to receive
     (possibly missing) end of the line, reply to the client with special error
     if chunk size is too large to be handled by MHD (>16 EiB).
     Added error reply if client used too large request payload (>16 EiB).
@@ -92,7 +96,7 @@ October 2021
     Added test family test_toolarge to check correct handling of the buffers
     when the size of data is larger than free space.
     Fixed missing updated of read and write buffers sizes.
-    Added detection and use of supported "noreturn" keyword for function
+    Added detection and use of supported "noreturn" keyword for function
```

```
+    Added detection and use of supported "noreturn" keyword for function
     declaration. It should help compiler and static analyser.
     Added support for leak sanitizer.
     Fixed analyser errors on W32.
@@ -290,7 +294,7 @@ June 2021
     used for the next request data.
     Fixed completely broken calculation of request header size.
     Chunked response: do not ask app callback for more data then
-    it is possible to process (more than 16 MBytes).
+    it is possible to process (more than 16 MBytes).
     Check and report if app used wrong response code (>999 or <100)
     Refuse to add second "Transfer-Encoding" header.
     HTTPS tests: check whether all libcurl function succeeded.
diff --git a/src/microhttpd/postprocessor.c b/src/microhttpd/postprocessor.c
@@ -83,7 +83,7 @@ MHD_create_post_processor (struct MHD_Connection *connection,
       return NULL; /* failed to determine boundary */
     boundary += MHD_STATICSTR_LEN_ ("boundary=");
     blen = strlen (boundary);
-    if ( (blen == 0) ||
+    if ( (blen < 2) ||
          (blen * 2 + 2 > buffer_size) )
       return NULL;              /* (will be) out of memory or invalid boundary */
     if ( (boundary[0] == '"') &&
```