

New issue

[Jump to bottom](#)

## heap-buffer-overflow in re\_parse\_char\_escape #3871

Closed owl337 opened this issue on Jun 7, 2020 · 0 comments · Fixed by #3875

Assignees



Labels

bug ecma builtins

owl337 commented on Jun 7, 2020

JerryScript revision

cae6cd0

Build platform

Ubuntu 16.04.6 LTS (Linux 4.15.0-99-generic x86\_64)

Build steps

```
./tools/build.py --clean --debug --compile-flag=-fsanitize=address \
--compile-flag=-m32 --compile-flag=-fno-omit-frame-pointer \
--compile-flag=-fno-common --compile-flag=-g --strip=off \
--system-allocator=on --logging=on --linker-flag=-fuse-ld=gold \
--error-messages=on --profile=es2015-subset --lto=off
```

Test case

```
new RegExp('"\\u', 'u').exec(1)
```

Output

```
=====
==105443==ERROR: AddressSanitizer: heap-buffer-overflow on address 0xf600073f at pc 0x080e94ba bp 0xffe285a8 sp 0xffe28598
READ of size 1 at 0xf600073f thread T0
#0 0x80e94b9 in re_parse_char_escape /home/jerryscript/jerry-core/parser/regexp/re-parser.c:615
#1 0x80ea114 in re_parse_next_token /home/jerryscript/jerry-core/parser/regexp/re-parser.c:753
#2 0x80eba19 in re_parse_alternative /home/jerryscript/jerry-core/parser/regexp/re-parser.c:1152
#3 0x80e642e in re_compile_bytecode /home/jerryscript/jerry-core/parser/regexp/re-compiler.c:131
#4 0x80a7bd2 in ecma_op_create_regexp_from_pattern /home/jerryscript/jerry-core/ecma/operations/ecma-regexp-object.c:347
#5 0x8129048 in ecma_builtin_regexp_dispatch_helper /home/jerryscript/jerry-core/ecma/builtin-objects/ecma-builtin-regexp.c:174
#6 0x81290c1 in ecma_builtin_regexp_dispatch_construct /home/jerryscript/jerry-core/ecma/builtin-objects/ecma-builtin-regexp.c:218
#7 0x8080fed in ecma_builtin_dispatch_construct /home/jerryscript/jerry-core/ecma/builtin-objects/ecma-builtins.c:1260
#8 0x809866e in ecma_op_function_construct /home/jerryscript/jerry-core/ecma/operations/ecma-function-object.c:1275
#9 0x80f0e29 in opfunc_construct /home/jerryscript/jerry-core/vm/vm.c:866
#10 0x80ffbd2 in vm_execute /home/jerryscript/jerry-core/vm/vm.c:4237
#11 0x81000d5 in vm_run /home/jerryscript/jerry-core/vm/vm.c:4318
#12 0x80eefc0 in vm_run_global /home/jerryscript/jerry-core/vm/vm.c:338
#13 0x804e1ce in jerry_run /home/jerryscript/jerry-core/api/jerry.c:595
#14 0x804acbf in main /home/jerryscript/jerry-main/main-unix.c:759
#15 0xf784a636 in __libc_start_main (/lib/i386-linux-gnu/libc.so.6+0x18636)
#16 0x8048fb0 (/home/jerryscript/build/bin/jerry+0x8048fb0)
```

0xf600073f is located 0 bytes to the right of 15-byte region [0xf6000730,0xf600073f)  
allocated by thread T0 here:

```
#0 0xf7a7edee in malloc (/usr/lib32/libasan.so.2+0x96dee)
#1 0x80be178 in jmem_heap_alloc /home/jerryscript/jerry-core/jmem/jmem-heap.c:254
#2 0x80be248 in jmem_heap_gc_and_alloc_block /home/jerryscript/jerry-core/jmem/jmem-heap.c:289
#3 0x80be2c7 in jmem_heap_alloc_block /home/jerryscript/jerry-core/jmem/jmem-heap.c:323
#4 0x8100214 in ecma_alloc_string_buffer /home/jerryscript/jerry-core/ecma/base/ecma-alloc.c:194
#5 0x80619da in ecma_new_ecma_string_from_utf8_buffer /home/jerryscript/jerry-core/ecma/base/ecma-helpers-string.c:258
#6 0x80619da in ecma_new_ecma_string_from_utf8 /home/jerryscript/jerry-core/ecma/base/ecma-helpers-string.c:353
#7 0x8073bf5 in ecma_find_or_create_literal_string /home/jerryscript/jerry-core/ecma/base/ecma-literal-storage.c:134
#8 0x80c4501 in parser_compute_indices /home/jerryscript/jerry-core/parser/js/js-parser.c:130
#9 0x80c61c6 in parser_post_processing /home/jerryscript/jerry-core/parser/js/js-parser.c:973
#10 0x80cbcb9 in parser_parse_source /home/jerryscript/jerry-core/parser/js/js-parser.c:2192
#11 0x80cf05a in parser_parse_script /home/jerryscript/jerry-core/parser/js/js-parser.c:2813
#12 0x804dc81 in jerry_parse /home/jerryscript/jerry-core/api/jerry.c:447
#13 0x804ac76 in main /home/jerryscript/jerry-main/main-unix.c:750
#14 0xf784a636 in __libc_start_main (/lib/i386-linux-gnu/libc.so.6+0x18636)
```

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/jerryscript/jerry-core/parser/regexp/re-parser.c:615 re\_parse\_char\_escape

Shadow bytes around the buggy address:

```
0x3ec00090: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ec000a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ec000b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ec000c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ec000d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x3ec000e0: fa 00 05 fa 00[07]fa fa fd fa fa fd fd
0x3ec000f0: fa fa 00 00 fa fa 00 00 fa fa 00 06 fa fa 00 00
0x3ec00100: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ec00110: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ec00120: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ec00130: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
```

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
Heap right redzone: fb
Freed heap region: fd
Stack left redzone: f1
```

```
Stack mid redzone: f2
Stack right redzone: f3
Stack partial redzone: f4
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
==105443==ABORTING
```

Credits: This vulnerability is detected by chong from OWL337.

 **dbatyai** added **bug** **ecma builtins** labels on Jun 7, 2020

 **dbatyai** self-assigned this on Jun 7, 2020

 **dbatyai** added a commit to dbatyai/jerryscript that referenced this issue on Jun 8, 2020

 Add missing end-of-string checks to RegExp parser in unicode mode ...


0044a35

 **dbatyai** mentioned this issue on Jun 8, 2020

**Add missing end-of-string checks to RegExp parser in unicode mode #3875**

 Merged

 **rerobika** closed this as completed in [#3875](#) on Jun 8, 2020

 **rerobika** pushed a commit that referenced this issue on Jun 8, 2020

 Add missing end-of-string checks to RegExp parser in unicode mode ([#3875](#)) ...

✓ fed1b0c

#### Assignees

 **dbatyai**

#### Labels

**bug** **ecma builtins**

#### Projects


None yet

#### Milestone

No milestone

#### Development

Successfully merging a pull request may close this issue.

 Add missing end-of-string checks to RegExp parser in unicode mode  
dbatyai/jerryscript

#### 2 participants

