New issue

heap-buffer-overflow in gravity_vm_exec with many string literals + recursion #437

Closed

segv0x opened 2 days ago

The VM fiber stack can be written out of bounds during execution. The stack growth check in `gravity_check_stack()` doesn't seem to catch the case where the register window exceeds the remaining stack capacity, so `gravity_vm_exec` writes past the allocated region. The attached PoC triggers this with a recursive function and many string literals at global scope.

Tested on latest master (`87c9c90`), Ubuntu x86_64, clang 20.

To reproduce, build with ASAN (`-fsanitize=address` added to CFLAGS/LDFLAGS in the Makefile) and run the attached PoC:

```
./gravity poc.gravity
```

ASAN output:

```
==15096==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6e2deade3900 at pc
0x5eb247c034ee bp 0x7ffd9b49be00 sp 0x7ffd9b49bdf8
WRITE of size 8 at 0x6e2deade3900 thread T0
    #0 0x5eb247c034ed in gravity_vm_exec src/runtime/gravity_vm.c:522:17
    #1 0x5eb247c0867a in gravity_vm_runmain src/runtime/gravity_vm.c:1809:19

0x6e2deade3900 is located 0 bytes after 4096-byte region [0x6e2deade2900,0x6e2deade3900)
allocated by thread T0 here:
    #0 0x5eb247b12f9d in calloc
    #1 0x5eb247c5c75d in gravity_fiber_new src/shared/gravity_value.c:1377:39
    #2 0x5eb247bd2d12 in gravity_vm_new src/runtime/gravity_vm.c:1524:17

SUMMARY: AddressSanitizer: heap-buffer-overflow src/runtime/gravity_vm.c:522:17 in
gravity_vm_exec
```

Without ASAN the regular binary crashes with `realloc(): invalid next size`.

PoC attached.

poc.zip

marcobambini added a commit that references this issue 2 days ago

`Bump version to 0.9.6 — OOM safety, init-chain fix, docs, and test su.` ··· `18b9195`

marcobambini 2 days ago Owner •••

@**segv0x** thanks for the report!
Fixed by `18b9195`

marcobambini closed this as completed 2 days ago

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants