r2con 2019

down the business with r2dwarf

Giovanni - iGio90 - Rocca

Content

What is Dwarf

-> Built with the concept to create a frontend for Frida

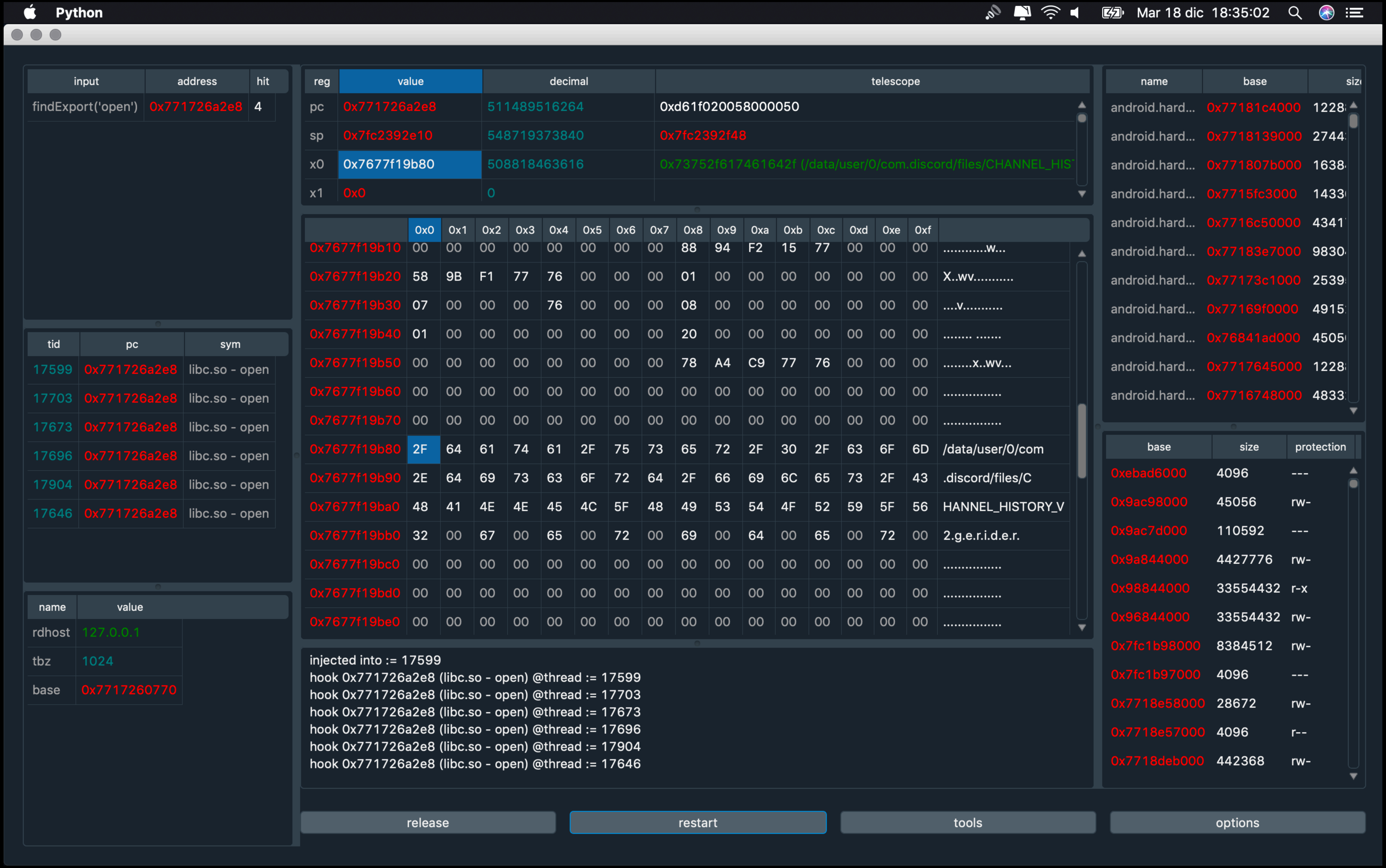
-> Nowadays it can be named "a framework"

-> Allows to debug target processes

    -> API to insert breakpoints and watchpoints
    -> UI components to interact with Frida in runtime
    -> Trace and step native / java code
    -> Multi arch/os

-> Extendible

    -> Plugin development allows UI injection and easy way to speak with Frida



input	address	hit
findExport('open')	0x771726a2e8	4

reg	value	decimal	telescope
pc	0x771726a2e8	511489516264	0xd61f020058000050
sp	0x7fc2392e10	548719373840	0x7fc2392f48
x0	0x7677f19b80	508818463616	0x73752f617461642f (/data/user/0/com.discord/files/CHANNEL_HIS
x1	0x0	0	

name	base	siz
android.hard...	0x77181c4000	1228
android.hard...	0x7718139000	2744
android.hard...	0x771807b000	1638
android.hard...	0x7715fc3000	1433
android.hard...	0x7716c50000	4341
android.hard...	0x77183e7000	9830
android.hard...	0x77173c1000	2539
android.hard...	0x77169f0000	4915
android.hard...	0x76841ad000	4505
android.hard...	0x7717645000	1228
android.hard...	0x7716748000	4833

```
              0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb 0xc 0xd 0xe 0xf
0x7677f19b10  00  00  00  00  00  00  00  00  88  94  F2  15  77  00  00  00  ...........w...
0x7677f19b20  58  9B  F1  77  76  00  00  00  01  00  00  00  00  00  00  00  X..wv..........
0x7677f19b30  07  00  00  00  76  00  00  00  08  00  00  00  00  00  00  00  ....v..........
0x7677f19b40  01  00  00  00  00  00  00  00  20  00  00  00  00  00  00  00  ........ .......
0x7677f19b50  00  00  00  00  00  00  00  00  78  A4  C9  77  76  00  00  00  ........x..wv...
0x7677f19b60  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  ................
0x7677f19b70  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  ................
0x7677f19b80  2F  64  61  74  61  2F  75  73  65  72  2F  30  2F  63  6F  6D  /data/user/0/com
0x7677f19b90  2E  64  69  73  63  6F  72  64  2F  66  69  6C  65  73  2F  43  .discord/files/C
0x7677f19ba0  48  41  4E  4E  45  4C  5F  48  49  53  54  4F  52  59  5F  56  HANNEL_HISTORY_V
0x7677f19bb0  32  00  67  00  65  00  72  00  69  00  64  00  65  00  72  00  2.g.e.r.i.d.e.r.
0x7677f19bc0  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  ................
0x7677f19bd0  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  ................
0x7677f19be0  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  ................
```

tid	pc	sym
17599	0x771726a2e8	libc.so - open
17703	0x771726a2e8	libc.so - open
17673	0x771726a2e8	libc.so - open
17696	0x771726a2e8	libc.so - open
17904	0x771726a2e8	libc.so - open
17646	0x771726a2e8	libc.so - open

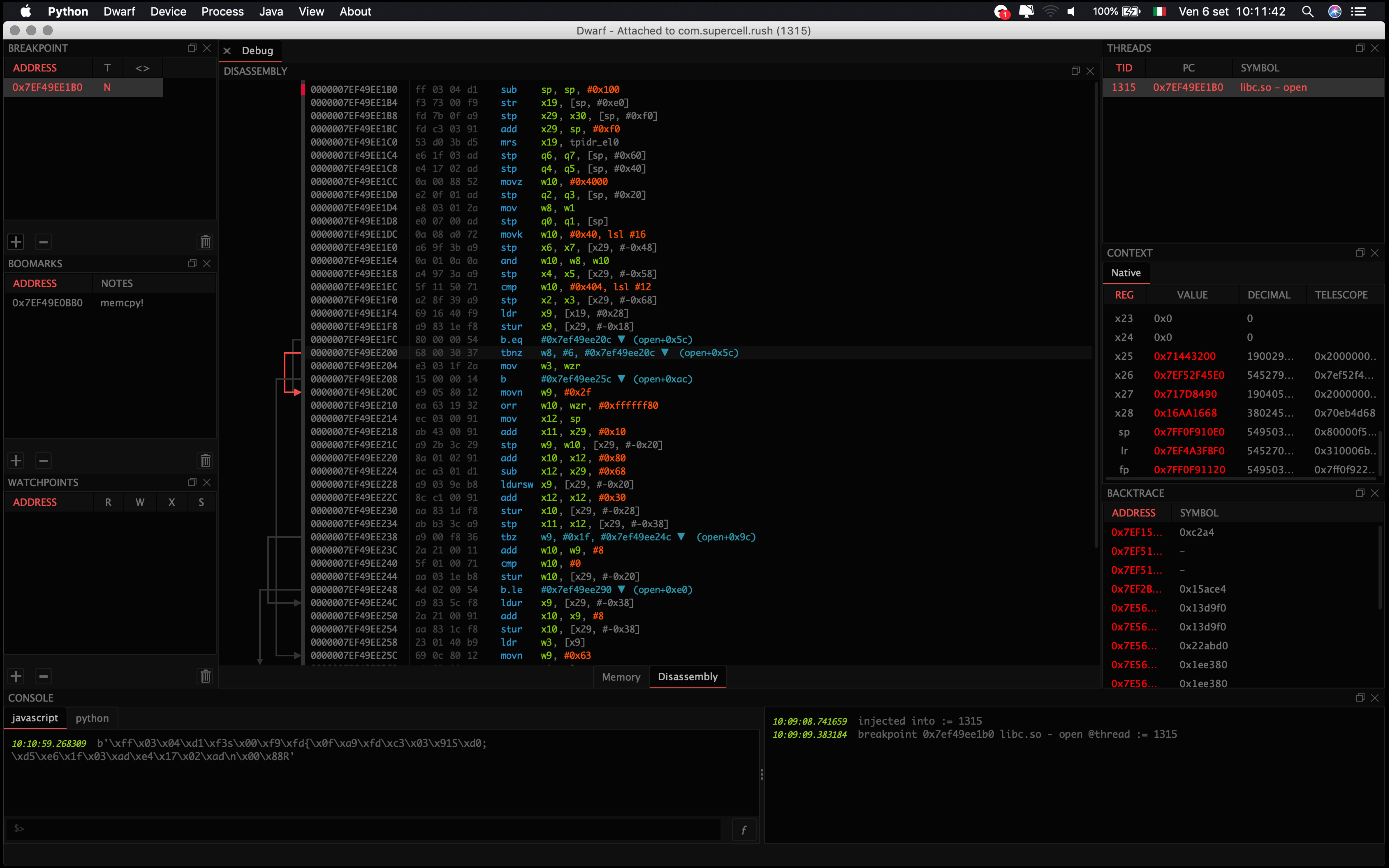
base	size	protection
0xebad6000	4096	---
0x9ac98000	45056	rw-
0x9ac7d000	110592	---
0x9a844000	4427776	rw-
0x98844000	33554432	r-x
0x96844000	33554432	rw-
0x7fc1b98000	8384512	rw-
0x7fc1b97000	4096	---
0x7718e58000	28672	rw-
0x7718e57000	4096	r--
0x7718deb000	442368	rw-

name	value
rdhost	127.0.0.1
tbz	1024
base	0x7717260770

```
injected into := 17599
hook 0x771726a2e8 (libc.so - open) @thread := 17599
hook 0x771726a2e8 (libc.so - open) @thread := 17703
hook 0x771726a2e8 (libc.so - open) @thread := 17673
hook 0x771726a2e8 (libc.so - open) @thread := 17696
hook 0x771726a2e8 (libc.so - open) @thread := 17904
hook 0x771726a2e8 (libc.so - open) @thread := 17646
```

Dwarf in early 2k19

r2con 2019



ADDRESS	T	<>
0x7EF49EE1B0	N	

ADDRESS	NOTES
0x7EF49E0BB0	memcpy!

ADDRESS	R	W	X	S

```
0000007EF49EE1B0  ff 03 04 d1  sub    sp, sp, #0x100
0000007EF49EE1B4  f3 73 00 f9  str    x19, [sp, #0xe0]
0000007EF49EE1B8  fd 7b 0f a9  stp    x29, x30, [sp, #0xf0]
0000007EF49EE1BC  fd c3 03 91  add    x29, sp, #0xf0
0000007EF49EE1C0  53 d0 3b d5  mrs    x19, tpidr_el0
0000007EF49EE1C4  e6 1f 03 ad  stp    q6, q7, [sp, #0x60]
0000007EF49EE1C8  e4 17 02 ad  stp    q4, q5, [sp, #0x40]
0000007EF49EE1CC  0a 00 88 52  movz   w10, #0x4000
0000007EF49EE1D0  e2 0f 01 ad  stp    q2, q3, [sp, #0x20]
0000007EF49EE1D4  e8 03 01 2a  mov    w8, w1
0000007EF49EE1D8  e0 07 00 ad  stp    q0, q1, [sp]
0000007EF49EE1DC  0a 08 a0 72  movk   w10, #0x40, lsl #16
0000007EF49EE1E0  a6 9f 3b a9  stp    x6, x7, [x29, #-0x48]
0000007EF49EE1E4  0a 01 0a 0a  and    w10, w8, w10
0000007EF49EE1E8  a4 97 3a a9  stp    x4, x5, [x29, #-0x58]
0000007EF49EE1EC  5f 11 50 71  cmp    w10, #0x404, lsl #12
0000007EF49EE1F0  a2 8f 39 a9  stp    x2, x3, [x29, #-0x68]
0000007EF49EE1F4  69 16 40 f9  ldr    x9, [x19, #0x28]
0000007EF49EE1F8  a9 83 1e f8  stur   x9, [x29, #-0x18]
0000007EF49EE1FC  80 00 00 54  b.eq   #0x7ef49ee20c ▼ (open+0x5c)
0000007EF49EE200  68 00 30 37  tbnz   w8, #6, #0x7ef49ee20c ▼  (open+0x5c)
0000007EF49EE204  e3 03 1f 2a  mov    w3, wzr
0000007EF49EE208  15 00 00 14  b      #0x7ef49ee25c ▼ (open+0xac)
0000007EF49EE20C  e9 05 80 12  movn   w9, #0x2f
0000007EF49EE210  ea 63 19 32  orr    w10, wzr, #0xffffff80
0000007EF49EE214  ec 03 00 91  mov    x12, sp
0000007EF49EE218  ab 43 00 91  add    x11, x29, #0x10
0000007EF49EE21C  a9 2b 3c 29  stp    w9, w10, [x29, #-0x20]
0000007EF49EE220  8a 01 02 91  add    x10, x12, #0x80
0000007EF49EE224  ac a3 01 d1  sub    x12, x29, #0x68
0000007EF49EE228  a9 03 9e b8  ldursw x9, [x29, #-0x20]
0000007EF49EE22C  8c c1 00 91  add    x12, x12, #0x30
0000007EF49EE230  aa 83 1d f8  stur   x10, [x29, #-0x28]
0000007EF49EE234  ab b3 3c a9  stp    x11, x12, [x29, #-0x38]
0000007EF49EE238  a9 00 f8 36  tbz    w9, #0x1f, #0x7ef49ee24c ▼  (open+0x9c)
0000007EF49EE23C  2a 21 00 11  add    w10, w9, #8
0000007EF49EE240  5f 01 00 71  cmp    w10, #0
0000007EF49EE244  aa 03 1e b8  stur   w10, [x29, #-0x20]
0000007EF49EE248  4d 02 00 54  b.le   #0x7ef49ee290 ▼ (open+0xe0)
0000007EF49EE24C  a9 83 5c f8  ldur   x9, [x29, #-0x38]
0000007EF49EE250  2a 21 00 91  add    x10, x9, #8
0000007EF49EE254  aa 83 1c f8  stur   x10, [x29, #-0x38]
0000007EF49EE258  23 01 40 b9  ldr    w3, [x9]
0000007EF49EE25C  69 0c 80 12  movn   w9, #0x63
```

TID	PC	SYMBOL
1315	0x7EF49EE1B0	libc.so - open

REG	VALUE	DECIMAL	TELESCOPE
x23	0x0	0	
x24	0x0	0	
x25	0x71443200	190029…	0x2000000..
x26	0x7EF52F45E0	545279…	0x7ef52f4…
x27	0x717D8490	190405…	0x2000000..
x28	0x16AA1668	380245…	0x70eb4d68
sp	0x7FF0F910E0	549503…	0x80000f5…
lr	0x7EF4A3FBF0	545270…	0x310006b..
fp	0x7FF0F91120	549503…	0x7ff0f922..

ADDRESS	SYMBOL
0x7EF15…	0xc2a4
0x7EF51…	-
0x7EF51…	-
0x7EF2B…	0x15ace4
0x7E56…	0x13d9f0
0x7E56…	0x13d9f0
0x7E56…	0x22abd0
0x7E56…	0x1ee380
0x7E56…	0x1ee380

```
10:10:59.268309  b'\xff\x03\x04\xd1\xf3s\x00\xf9\xfd{\x0f\xa9\xfd\xc3\x03\x91S\xd0;
\xd5\xe6\x1f\x03\xad\xe4\x17\x02\xad\n\x00\x88R'
```

```
10:09:08.741659  injected into := 1315
10:09:09.383184  breakpoint 0x7ef49ee1b0 libc.so - open @thread := 1315
```

Dwarf today

R2Dwarf

-> A pipe between Dwarf and R2

-> Enrich the debug UI with graph and decompiler (r2dec)

-> Automated analysis

-> Expose a javascript sync API to run R2 commands in the Frida agent

-> R2 console

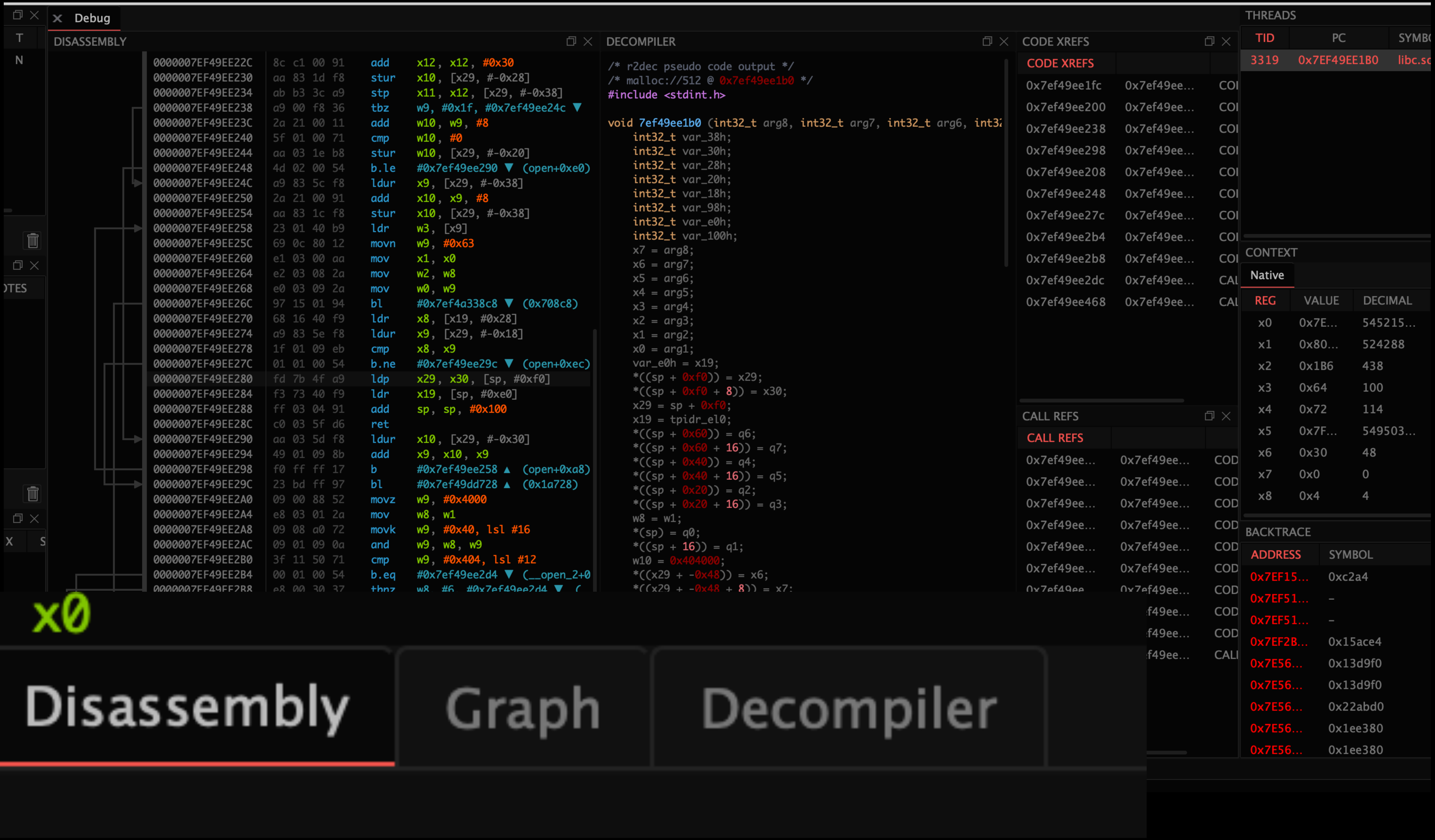
R2Dwarf

-> installing plugin in Dwarf is easy but still manual

  -> there is a wip on a plugin manager which will make it easier

-> clone the plugin into ~/.dwarf/plugins/

R2Dwarf



```
0000007EF49EE22C  8c c1 00 91   add    x12, x12, #0x30
0000007EF49EE230  aa 83 1d f8   stur   x10, [x29, #-0x28]
0000007EF49EE234  ab b3 3c a9   stp    x11, x12, [x29, #-0x38]
0000007EF49EE238  a9 00 f8 36   tbz    w9, #0x1f, #0x7ef49ee24c ▼
0000007EF49EE23C  2a 21 00 11   add    w10, w9, #8
0000007EF49EE240  5f 01 00 71   cmp    w10, #0
0000007EF49EE244  aa 03 1e b8   stur   w10, [x29, #-0x20]
0000007EF49EE248  4d 02 00 54   b.le   #0x7ef49ee290 ▼ (open+0xe0)
0000007EF49EE24C  a9 83 5c f8   ldur   x9, [x29, #-0x38]
0000007EF49EE250  2a 21 00 91   add    x10, x9, #8
0000007EF49EE254  aa 83 1c f8   stur   x10, [x29, #-0x38]
0000007EF49EE258  23 01 40 b9   ldr    w3, [x9]
0000007EF49EE25C  69 0c 80 12   movn   w9, #0x63
0000007EF49EE260  e1 03 00 aa   mov    x1, x0
0000007EF49EE264  e2 03 08 2a   mov    w2, w8
0000007EF49EE268  e0 03 09 2a   mov    w0, w9
0000007EF49EE26C  97 15 01 94   bl     #0x7ef4a338c8 ▼ (0x708c8)
0000007EF49EE270  68 16 40 f9   ldr    x8, [x19, #0x28]
0000007EF49EE274  a9 83 5e f8   ldur   x9, [x29, #-0x18]
0000007EF49EE278  1f 01 09 eb   cmp    x8, x9
0000007EF49EE27C  01 01 00 54   b.ne   #0x7ef49ee29c ▼ (open+0xec)
0000007EF49EE280  fd 7b 4f a9   ldp    x29, x30, [sp, #0xf0]
0000007EF49EE284  f3 73 40 f9   ldr    x19, [sp, #0xe0]
0000007EF49EE288  ff 03 04 91   add    sp, sp, #0x100
0000007EF49EE28C  c0 03 5f d6   ret
0000007EF49EE290  aa 03 5d f8   ldur   x10, [x29, #-0x30]
0000007EF49EE294  49 01 09 8b   add    x9, x10, x9
0000007EF49EE298  f0 ff ff 17   b      #0x7ef49ee258 ▲ (open+0xa8)
0000007EF49EE29C  23 bd ff 97   bl     #0x7ef49dd728 ▲ (0x1a728)
0000007EF49EE2A0  09 00 88 52   movz   w9, #0x4000
0000007EF49EE2A4  e8 03 01 2a   mov    w8, w1
0000007EF49EE2A8  09 08 a0 72   movk   w9, #0x40, lsl #16
0000007EF49EE2AC  09 01 09 0a   and    w9, w8, w9
0000007EF49EE2B0  3f 11 50 71   cmp    w9, #0x404, lsl #12
0000007EF49EE2B4  00 01 00 54   b.eq   #0x7ef49ee2d4 ▼ (__open_2+0
0000007EF49EE2B8  e8 00 30 37   tbnz   w8, #6, #0x7ef49ee2d4 ▼ (
```

```
/* r2dec pseudo code output */
/* malloc://512 @ 0x7ef49ee1b0 */
#include <stdint.h>

void 7ef49ee1b0 (int32_t arg8, int32_t arg7, int32_t arg6, int32
    int32_t var_38h;
    int32_t var_30h;
    int32_t var_28h;
    int32_t var_20h;
    int32_t var_18h;
    int32_t var_98h;
    int32_t var_e0h;
    int32_t var_100h;
    x7 = arg8;
    x6 = arg7;
    x5 = arg6;
    x4 = arg5;
    x3 = arg4;
    x2 = arg3;
    x1 = arg2;
    x0 = arg1;
    var_e0h = x19;
    *((sp + 0xf0)) = x29;
    *((sp + 0xf0 + 8)) = x30;
    x29 = sp + 0xf0;
    x19 = tpidr_el0;
    *((sp + 0x60)) = q6;
    *((sp + 0x60 + 16)) = q7;
    *((sp + 0x40)) = q4;
    *((sp + 0x40 + 16)) = q5;
    *((sp + 0x20)) = q2;
    *((sp + 0x20 + 16)) = q3;
    w8 = w1;
    *(sp) = q0;
    *((sp + 16)) = q1;
    w10 = 0x404000;
    *((x29 + -0x48)) = x6;
    *((x29 + -0x48 + 8)) = x7;
```

CODE XREFS

CODE XREFS		
0x7ef49ee1fc	0x7ef49ee...	COI
0x7ef49ee200	0x7ef49ee...	COI
0x7ef49ee238	0x7ef49ee...	COI
0x7ef49ee298	0x7ef49ee...	COI
0x7ef49ee208	0x7ef49ee...	COI
0x7ef49ee248	0x7ef49ee...	COI
0x7ef49ee27c	0x7ef49ee...	COI
0x7ef49ee2b4	0x7ef49ee...	COI
0x7ef49ee2b8	0x7ef49ee...	COI
0x7ef49ee2dc	0x7ef49ee...	CAL
0x7ef49ee468	0x7ef49ee...	CAL

CALL REFS

CALL REFS		
0x7ef49ee...	0x7ef49ee...	COD
0x7ef49ee...	0x7ef49ee...	COD
0x7ef49ee...	0x7ef49ee...	COD
0x7ef49ee...	0x7ef49ee...	COD
0x7ef49ee...	0x7ef49ee...	COD
0x7ef49ee...	0x7ef49ee...	COD
0x7ef49ee...	0x7ef49ee...	COD
	f49ee...	COD
	f49ee...	COD
	f49ee...	CALI

THREADS

TID	PC	SYMB
3319	0x7EF49EE1B0	libc.sc

CONTEXT

Native

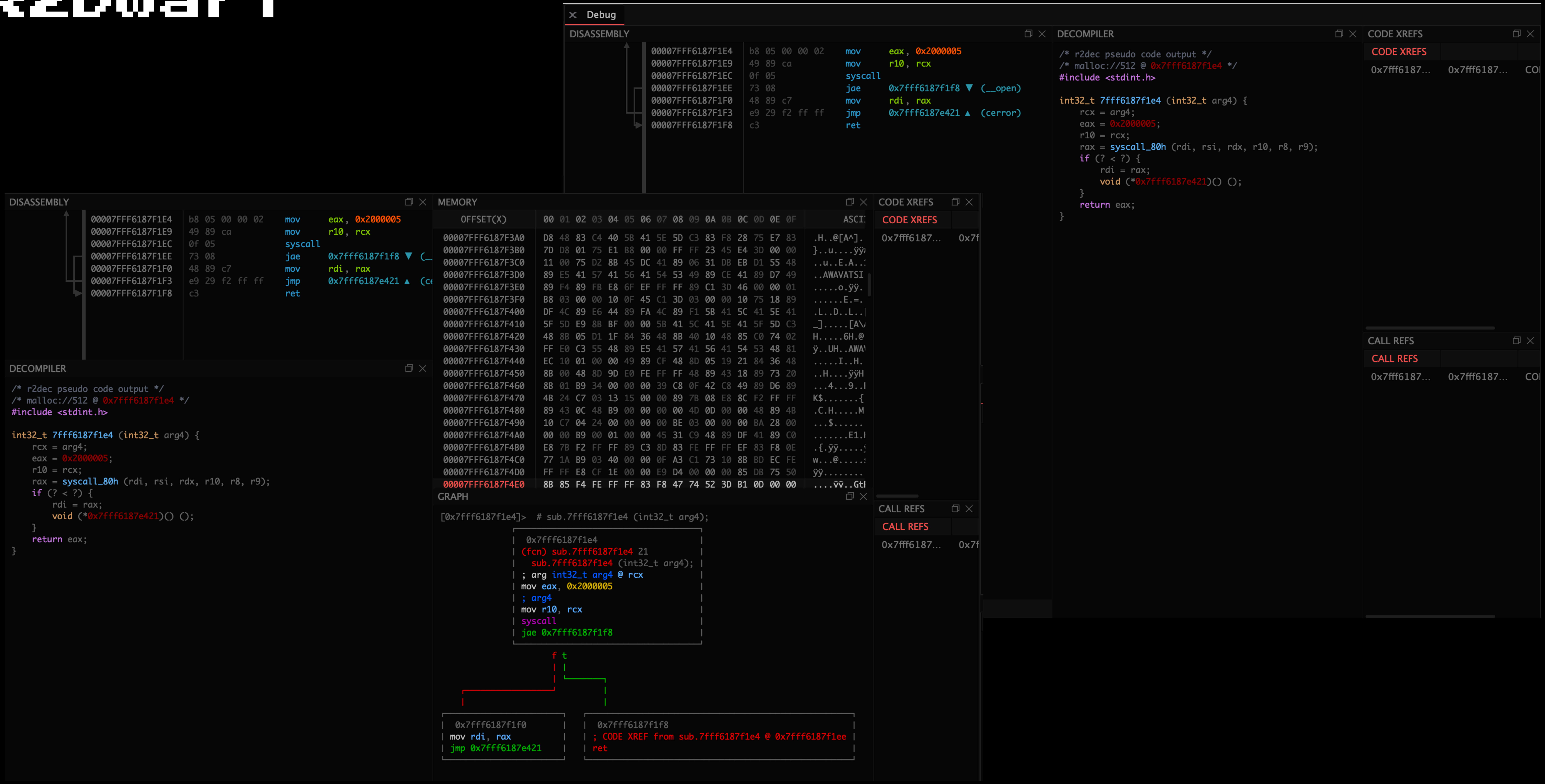
REG	VALUE	DECIMAL
x0	0x7E...	545215...
x1	0x80...	524288
x2	0x1B6	438
x3	0x64	100
x4	0x72	114
x5	0x7F...	549503...
x6	0x30	48
x7	0x0	0
x8	0x4	4

BACKTRACE

ADDRESS	SYMBOL
0x7EF15...	0xc2a4
0x7EF51...	-
0x7EF51...	-
0x7EF2B...	0x15ace4
0x7E56...	0x13d9f0
0x7E56...	0x13d9f0
0x7E56...	0x22abd0
0x7E56...	0x1ee380
0x7E56...	0x1ee380

```
mov        x1, x0
```

R2Dwarf



Debug

DISASSEMBLY

```
00007FFF6187F1E4   b8 05 00 00 02   mov      eax, 0x2000005
00007FFF6187F1E9   49 89 ca         mov      r10, rcx
00007FFF6187F1EC   0f 05            syscall
00007FFF6187F1EE   73 08            jae      0x7fff6187f1f8 ▼ (__open)
00007FFF6187F1F0   48 89 c7         mov      rdi, rax
00007FFF6187F1F3   e9 29 f2 ff ff   jmp      0x7fff6187e421 ▲ (cerror)
00007FFF6187F1F8   c3               ret
```

DECOMPILER

```
/* r2dec pseudo code output */
/* malloc://512 @ 0x7fff6187f1e4 */
#include <stdint.h>

int32_t 7fff6187f1e4 (int32_t arg4) {
    rcx = arg4;
    eax = 0x2000005;
    r10 = rcx;
    rax = syscall_80h (rdi, rsi, rdx, r10, r8, r9);
    if (? < ?) {
        rdi = rax;
        void (*0x7fff6187e421)() ();
    }
    return eax;
}
```

CODE XREFS

CODE XREFS

0x7fff6187...   0x7fff6187...   CO

CALL REFS

CALL REFS

0x7fff6187...   0x7fff6187...   CO

DISASSEMBLY

```
00007FFF6187F1E4   b8 05 00 00 02   mov      eax, 0x2000005
00007FFF6187F1E9   49 89 ca         mov      r10, rcx
00007FFF6187F1EC   0f 05            syscall
00007FFF6187F1EE   73 08            jae      0x7fff6187f1f8 ▼ (_
00007FFF6187F1F0   48 89 c7         mov      rdi, rax
00007FFF6187F1F3   e9 29 f2 ff ff   jmp      0x7fff6187e421 ▲ (ce
00007FFF6187F1F8   c3               ret
```

DECOMPILER

```
/* r2dec pseudo code output */
/* malloc://512 @ 0x7fff6187f1e4 */
#include <stdint.h>

int32_t 7fff6187f1e4 (int32_t arg4) {
    rcx = arg4;
    eax = 0x2000005;
    r10 = rcx;
    rax = syscall_80h (rdi, rsi, rdx, r10, r8, r9);
    if (? < ?) {
        rdi = rax;
        void (*0x7fff6187e421)() ();
    }
    return eax;
}
```

MEMORY

```
   OFFSET(X)        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F      ASCI
00007FFF6187F3A0    D8 48 83 C4 40 5B 41 5E 5D C3 83 F8 28 75 E7 83   .H..@[A^].
00007FFF6187F3B0    7D D8 01 75 E1 B8 00 00 FF FF 23 45 E4 3D 00 00   }..u....ÿÿ
00007FFF6187F3C0    11 00 75 D2 8B 45 DC 41 89 06 31 DB EB D1 55 48   ..u..E.A..
00007FFF6187F3D0    89 E5 41 57 41 56 41 54 53 49 89 CE 41 89 D7 49   ..AWAVATSI
00007FFF6187F3E0    89 F4 89 FB E8 6F EF FF FF 89 C1 3D 46 00 00 01   .....o.ÿÿ.
00007FFF6187F3F0    B8 03 00 00 10 0F 45 C1 3D 03 00 00 10 75 18 89   ......E.=.
00007FFF6187F400    DF 4C 89 E6 44 89 FA 4C 89 F1 5B 41 5C 41 5E 41   .L..D..L..
00007FFF6187F410    5F 5D E9 8B BF 00 00 5B 41 5C 41 5E 41 5F 5D C3   _].....[A\
00007FFF6187F420    48 8B 05 D1 1F 84 36 48 8B 40 10 48 85 C0 74 02   H.....6H.@
00007FFF6187F430    FF E0 C3 55 48 89 E5 41 57 41 56 41 54 53 48 81   ÿ..UH..AWA
00007FFF6187F440    EC 10 01 00 00 49 89 CF 48 8D 05 19 21 84 36 48   .....I..H.
00007FFF6187F450    8B 00 48 8D 9D E0 FE FF FF 48 89 43 18 89 73 20   ..H....ÿÿH
00007FFF6187F460    8B 01 B9 34 00 00 00 39 C8 0F 42 C8 49 89 D6 89   ...4...9..
00007FFF6187F470    4B 24 C7 03 13 15 00 00 89 7B 08 E8 8C F2 FF FF   K$.......{
00007FFF6187F480    89 43 0C 48 B9 00 00 00 00 4D 0D 00 00 48 89 4B   .C.H.....M
00007FFF6187F490    10 C7 04 24 00 00 00 00 BE 03 00 00 00 BA 28 00   ...$......
00007FFF6187F4A0    00 00 B9 00 01 00 00 45 31 C9 48 89 DF 41 89 C0   .......E1.
00007FFF6187F4B0    E8 7B F2 FF FF 89 C3 8D 83 FE FF FF EF 83 F8 0E   .{.ÿÿ.....
00007FFF6187F4C0    77 1A B9 03 40 00 00 0F A3 C1 73 10 8B BD EC FE   w...@.....
00007FFF6187F4D0    FF FF E8 CF 1E 00 00 E9 D4 00 00 00 85 DB 75 50   ÿÿ........
00007FFF6187F4E0    8B 85 F4 FE FF FF 83 F8 47 74 52 3D B1 0D 00 00   ....ÿÿ..Gt
```

CODE XREFS

CODE XREFS

0x7fff6187...   0x7f

GRAPH

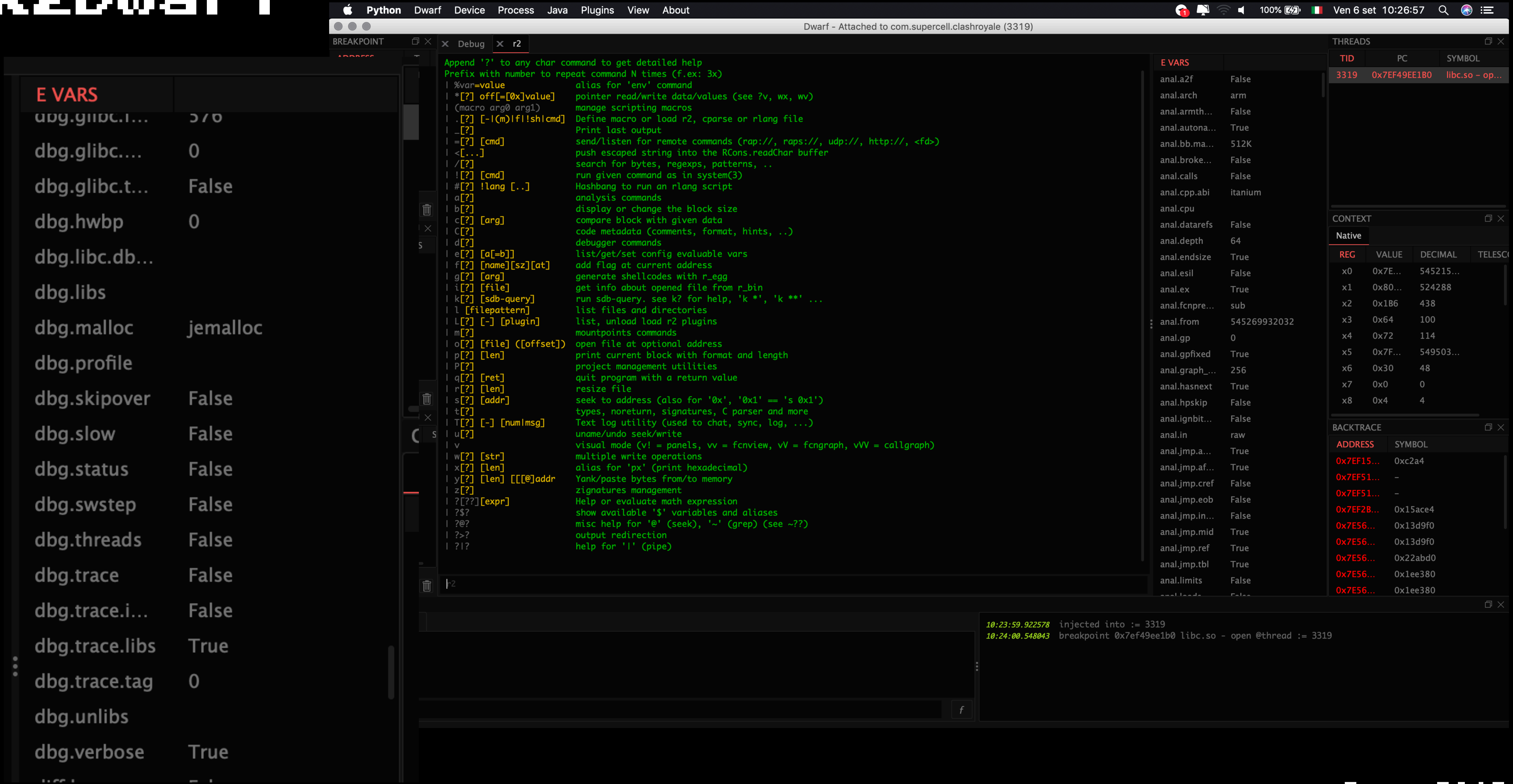
```
[0x7fff6187f1e4]>  # sub.7fff6187f1e4 (int32_t arg4);

              | 0x7fff6187f1e4                                |
              | (fcn) sub.7fff6187f1e4 21                     |
              |   sub.7fff6187f1e4 (int32_t arg4);            |
              | ; arg int32_t arg4 @ rcx                      |
              | mov eax, 0x2000005                            |
              | ; arg4                                        |
              | mov r10, rcx                                  |
              | syscall                                       |
              | jae 0x7fff6187f1f8                            |

                       f t

 | 0x7fff6187f1f0     |    | 0x7fff6187f1f8                                               |
 | mov rdi, rax       |    | ; CODE XREF from sub.7fff6187f1e4 @ 0x7fff6187f1ee          |
 | jmp 0x7fff6187e421 |    | ret                                                          |
```

CALL REFS

CALL REFS

0x7fff6187...   0x7f

R2Dwarf



E VARS	
dbg.glibc.f...	576
dbg.glibc....	0
dbg.glibc.t...	False
dbg.hwbp	0
dbg.libc.db...	
dbg.libs	
dbg.malloc	jemalloc
dbg.profile	
dbg.skipover	False
dbg.slow	False
dbg.status	False
dbg.swstep	False
dbg.threads	False
dbg.trace	False
dbg.trace.i...	False
dbg.trace.libs	True
dbg.trace.tag	0
dbg.unlibs	
dbg.verbose	True

```
Append '?' to any char command to get detailed help
Prefix with number to repeat command N times (f.ex: 3x)
| %var=value              alias for 'env' command
| *[?] off[=[0x]value]    pointer read/write data/values (see ?v, wx, wv)
| (macro arg0 arg1)       manage scripting macros
| .[?] [-|(m)|f|!sh|cmd]  Define macro or load r2, cparse or rlang file
| _[?]                    Print last output
| =[?] [cmd]              send/listen for remote commands (rap://, raps://, udp://, http://, <fd>)
| <[...]                  push escaped string into the RCons.readChar buffer
| /[?]                    search for bytes, regexps, patterns, ..
| ![?] [cmd]              run given command as in system(3)
| #[?] !lang [..]         Hashbang to run an rlang script
| a[?]                    analysis commands
| b[?]                    display or change the block size
| c[?] [arg]              compare block with given data
| C[?]                    code metadata (comments, format, hints, ..)
| d[?]                    debugger commands
| e[?] [a[=b]]            list/get/set config evaluable vars
| f[?] [name][sz][at]     add flag at current address
| g[?] [arg]              generate shellcodes with r_egg
| i[?] [file]             get info about opened file from r_bin
| k[?] [sdb-query]        run sdb-query. see k? for help, 'k *', 'k **' ...
| l [filepattern]         list files and directories
| L[?] [-] [plugin]       list, unload load r2 plugins
| m[?]                    mountpoints commands
| o[?] [file] ([offset])  open file at optional address
| p[?] [len]              print current block with format and length
| P[?]                    project management utilities
| q[?] [ret]              quit program with a return value
| r[?] [len]              resize file
| s[?] [addr]             seek to address (also for '0x', '0x1' == 's 0x1')
| t[?]                    types, noreturn, signatures, C parser and more
| T[?] [-] [num|msg]      Text log utility (used to chat, sync, log, ...)
| u[?]                    uname/undo seek/write
| v                       visual mode (v! = panels, vv = fcnview, vV = fcngraph, vVV = callgraph)
| w[?] [str]              multiple write operations
| x[?] [len]              alias for 'px' (print hexadecimal)
| y[?] [len] [[[@]addr    Yank/paste bytes from/to memory
| z[?]                    zignatures management
| ?[??][expr]             Help or evaluate math expression
| ?$?                     show available '$' variables and aliases
| ?@?                     misc help for '@' (seek), '~' (grep) (see ~??)
| ?>?                     output redirection
| ?|?                     help for '|' (pipe)
```

E VARS	
anal.a2f	False
anal.arch	arm
anal.armth...	False
anal.autona...	True
anal.bb.ma...	512K
anal.broke...	False
anal.calls	False
anal.cpp.abi	itanium
anal.cpu	
anal.datarefs	False
anal.depth	64
anal.endsize	True
anal.esil	False
anal.ex	True
anal.fcnpre...	sub
anal.from	545269932032
anal.gp	0
anal.gpfixed	True
anal.graph_...	256
anal.hasnext	True
anal.hpskip	False
anal.ignbit...	False
anal.in	raw
anal.jmp.a...	True
anal.jmp.af...	True
anal.jmp.cref	False
anal.jmp.eob	False
anal.jmp.in...	False
anal.jmp.mid	True
anal.jmp.ref	True
anal.jmp.tbl	True
anal.limits	False

TID	PC	SYMBOL
3319	0x7EF49EE1B0	libc.so - op...

REG	VALUE	DECIMAL	TELESC
x0	0x7E...	545215...	
x1	0x80...	524288	
x2	0x1B6	438	
x3	0x64	100	
x4	0x72	114	
x5	0x7F...	549503...	
x6	0x30	48	
x7	0x0	0	
x8	0x4	4	

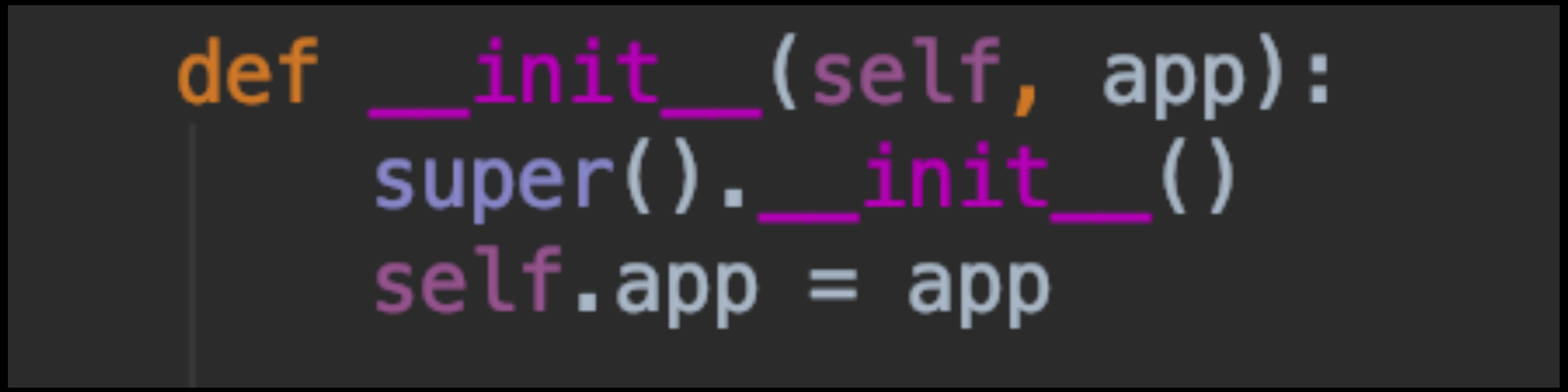
ADDRESS	SYMBOL
0x7EF15...	0xc2a4
0x7EF51...	-
0x7EF51...	-
0x7EF2B...	0x15ace4
0x7E56...	0x13d9f0
0x7E56...	0x13d9f0
0x7E56...	0x22abd0
0x7E56...	0x1ee380
0x7E56...	0x1ee380

```
10:23:59.922578  injected into := 3319
10:24:00.548043  breakpoint 0x7ef49ee1b0 libc.so - open @thread := 3319
```

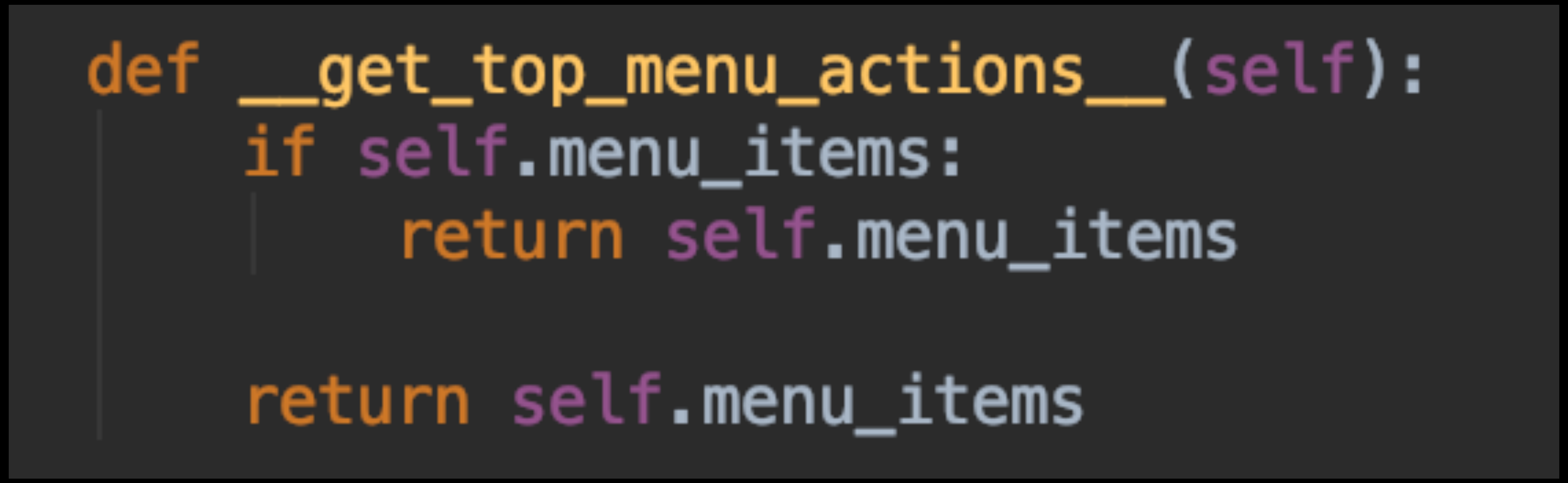
r2con 2019

Build your Dwarf plugins

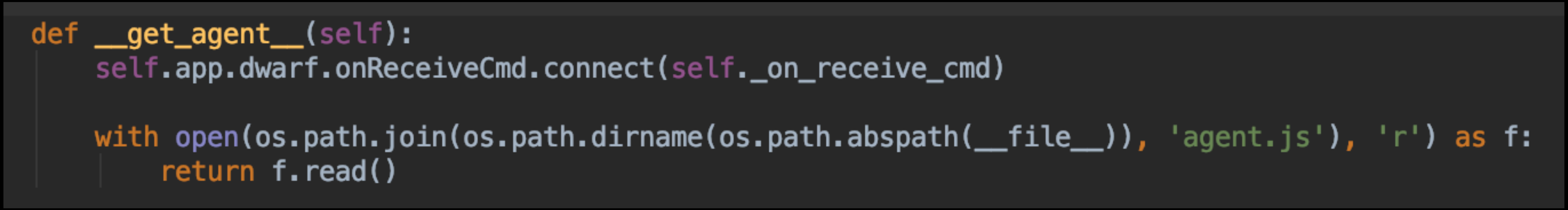
Has been made super “hacky” and easy due to the nature of the tool (OSS)



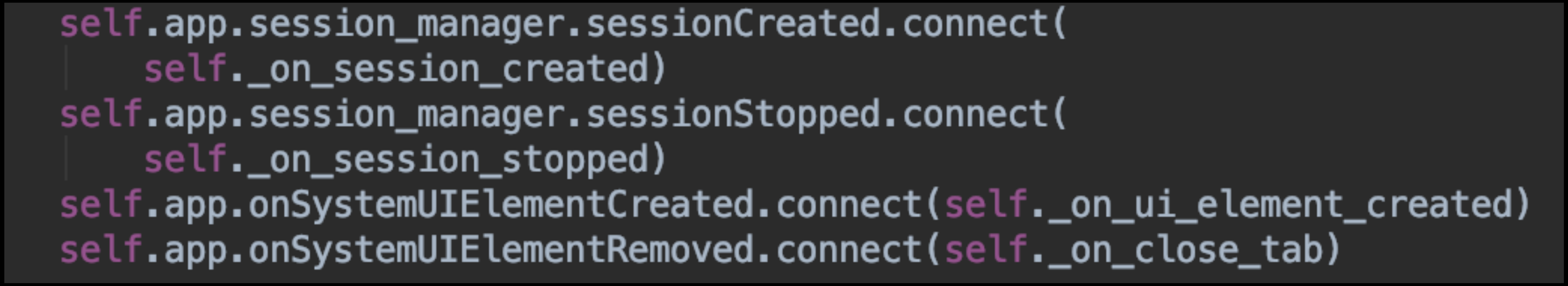
```python
def __init__(self, app):
    super().__init__()
    self.app = app
```



```python
def __get_top_menu_actions__(self):
    if self.menu_items:
        return self.menu_items

    return self.menu_items
```



```python
def __get_agent__(self):
    self.app.dwarf.onReceiveCmd.connect(self._on_receive_cmd)

    with open(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'agent.js'), 'r') as f:
        return f.read()
```



```python
self.app.session_manager.sessionCreated.connect(
    self._on_session_created)
self.app.session_manager.sessionStopped.connect(
    self._on_session_stopped)
self.app.onSystemUIElementCreated.connect(self._on_ui_element_created)
self.app.onSystemUIElementRemoved.connect(self._on_close_tab)
```

The global challenge of OSS developers

Involve users to use your tool

Standard mobile security analysis approach of a Frida user

-> Create the JavaScript agent

-> Copy paste the injector (Python / Node) from another project

-> Inject trash and redundant code to understand wtf is going on

The global challenge of OSS developers

iGio90 APPROVED!

The global challenge of OSS developers

Making it easy for the user to run it

but wait… I got a tool, which provides more JavaScript api.
How can I take my self to use my tool?

-> dwarf-creator: from 0 to IDE in 17 seconds.

-> dwarf-injector: quickly inject agents with Dwarf api and no UI

-> re-coded the whole JavaScript core in TypeScript

-> Giving typings and inline documentation on popular IDE

-> Separate breakpoints from Interceptor

-> pushing the right dude into fix Frida Stalker issue which was preventing native code step and tracers to work

My U.S. trip has been killed by a Frida check

-> During my holidays in NY, an engineer from Finland contacted me

-> I got asked to check a govn application and crack various layers

-> The application crashed with and without Frida running

-> The end of my holidays

My U.S. trip has been killed by a Frida check

TLDR; full documentation can be found at http://giovanni-rocca.com

The application

-> is used to prove the identity of the owner (ID)

-> was crashing without Frida in a device rooted with Magisk

-> my lovely @enovella found out the protector with apkid in 0.2
which was totally unknown at us back in the days

-> the effort required was just crazy

My U.S. trip has been killed by a Frida check

Giovanni Rocca si trova qui: **Pier A Harbor House**.
8 giu alle 03:45 • New York •

Sunset in NYC is just crazy!

… ok, super crazy. Now why the *@)! is the app crashing.

Hopefully they are not doing the crap way with sockets.

My U.S. trip has been killed by a Frida check

TLDR; full documentation can be found at http://giovanni-rocca.com

The goal

-> crack the various security layers: you are cool

-> take out my data and picture (with my friend engineer credentials, simulating a compromised device context): you are suppa cool

-> fake data and picture: GG WP

My U.S. trip has been killed by a Frida check

TLDR; full documentation can be found at http://giovanni-rocca.com

Chained checks are painful asf

-> the application was crashing without Frida running

   -> Giving evidence of root checks

-> debugging ^ with Frida was not really helpful

   -> A check for Frida was there before root check

-> debugging ^ with ptrace (strace, gdb) was not really helpful

   -> A check for ptrace was there before Frida check

My U.S. trip has been killed by a Frida check

TLDR; full documentation can be found at http://giovanni-rocca.com

The solution?

-> An initial - very unstable solution - was achieved by bypassing one by one all the checks

-> An interested CMP instruction was there just after any of the checks

-> That CMP was calling an additional function returning a bool

-> Altering the return of that functions tango downed everything

How the insecure world is preventing code injection?

TLDR; of my public researches

-> application signature verification

-> tracer check on /proc/self/status followed by a Frida check

-> inline syscall checks

    -> fstat (common su / binaries paths)

    -> socket (Frida)

    -> open/read (/proc/net/unix | /proc/self/maps) (Frida | substrate)

-> memory crc | fd notify

-> stack manipulation before crash

How the insecure world is preventing replication?

TLDR; of my public researches

-> obfuscation (data | code)

-> JNI <-> JVM “ping-pong”

-> encryptions

Detecting Frida

- -> /proc/self/maps
    - -> grep for "frida"
    - -> iterate all regions and scan for patterns
- -> /proc/net/unix
    - -> grep for "frida"
- -> ping listening sockets with Frida AUTH

Using Dwarf

- -> with UI
  - -> understanding complex functions
  - -> filter hundred of log lines
  - -> we are unsure about what is the target doing
  - -> test dynamic code and patches
- -> without UI
  - -> as Frida api extender
  - -> perform quick analysis and tests
  - -> build complex solutions

Using Dwarf

-> DEMO of a JVM Frida check crack with 5 lines of code

-> I though people wouldn't waste time developing and selling for thousand dollars some anti Frida running in the JVM.
Someone would say, better than nothing.
I was super wrong.

-> Nothing a-side inline syscall is really efficient if you know the system that is running the code

Detecting detections

-> strace is your friend as far as there are no ptrace checks chained

```
console.log(Process.id);
Thread.sleep(10);

strace -y -yy -x -i -f -o /sdcard/strace.log -p pid
```

-> fuzz strings

- -> strstr, strcmp, memcpy
- -> grep common strings (frida, agent)

-> prevent memory accesses in Frida space

- -> not ez. lowest possible level by reading /proc/self/maps

Stalker and R2

-> some recent fixes in the Frida gum allowed to:

- -> step native code
- -> trace native code with context arguments
  - -> trace specific instructions
    - -> trace inline syscalls

-> R2 plugin:

- -> backend analysis
  - -> functions details
  - -> graphs
  - -> decompiler
- -> sync js api
- -> … own usage case

Stalker

-> one shoot hook hit (Frida trampoline restored before stalking)

-> Stalker.follow -> transform

```
| module space - first block of code moved to frida space
| module space - jump to frida space

| frida space - do things
| frida space - 5x RET
| frida space - arm64 only: x1 block of code
| frida space - execute first block of code
| frida space - jump back to module space

| module space - continue execution
```

-> 5 RET instructions on both arm64 and x86-64

-> arm64 will execute 1 more block of code before the target one

Code step

```
| module space - first block of code moved to frida space
| module space - jump to frida space

|  frida space - do things
|  frida space - 5x RET
|  frida space - arm64 only: x1 block of code
|  frida space - execute first block of code
|  |
|  |---------------------------------------> some breakpoints needed here
|  |
|  frida space - jump back to module space

| module space - continue execution
```

Using Dwarf

-> DEMO of R2Dwarf

-> using r2 for quickly grab DT_INIT_ARRAY

-> nowadays Android linker unzip and read shared library in runtime

dlopen('/path/to/application.apk!/path/to/lib.so')

-> using Dwarf to step the code

r2con 2019

thank you

FOR NO QUESTIONS

<3

http://www.giovanni-rocca.com

https://github.com/iGio90

https://twitter.com/iGio90