Reversing with R2

/by pancake 2018 @ Sant Esteve De Les Roures

Reversing with R2

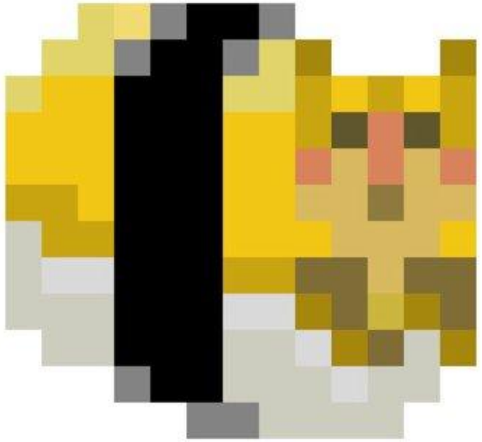
/by pancake 2018 @ ~~Sant Esteve De Les Roures~~
Girona -- OverdriveCon 2018

OverDrive
Conference

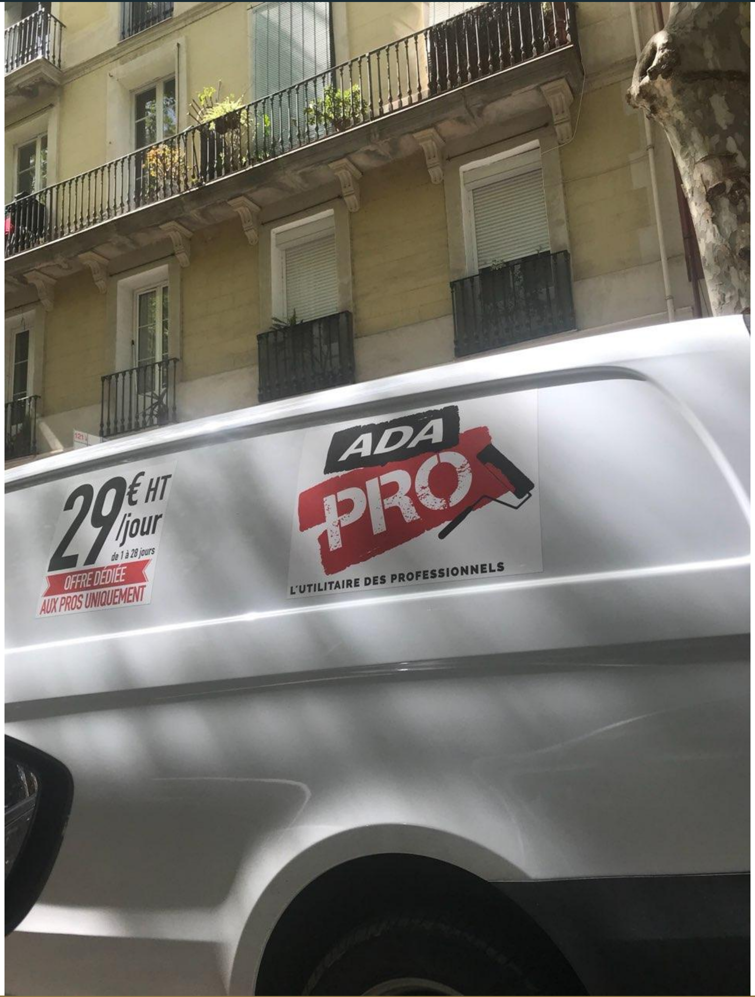
$ whoami

pancake aka Sergi Àlvarez

- @trufae on Twitter
- @radare on Github
- Mobile Security Analyst
- At NowSecure



wat


29€ HT
/jour
de 1 à 28 jours
OFFRE DÉDIÉE
AUX PROS UNIQUEMENT
ADA
PRO
L'UTILITAIRE DES PROFESSIONNELS

$ man radare2

$ whereis r2con

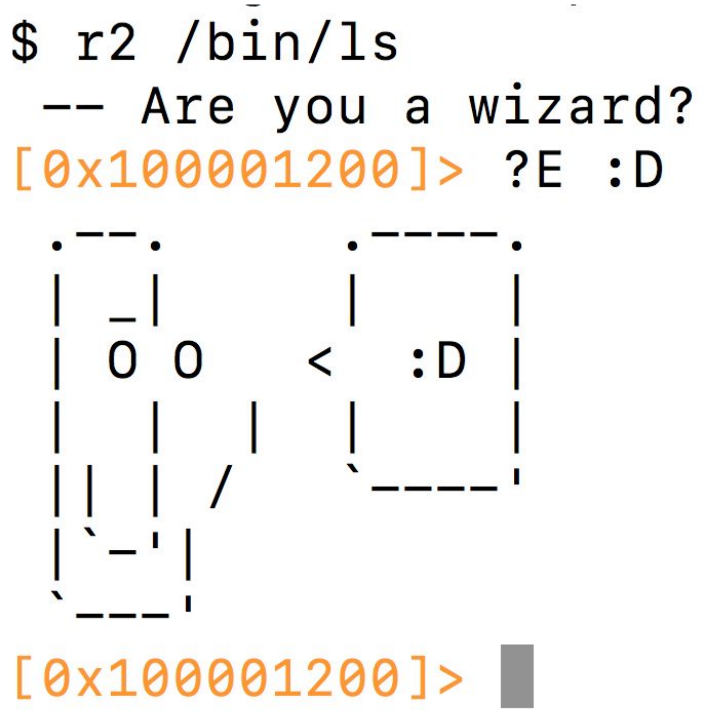
$ help

- Self documented
- Tons of talks in youtube
- 2 open-source books
- Several blogposts solving stuff

$ what

Reverse Engineering Framework

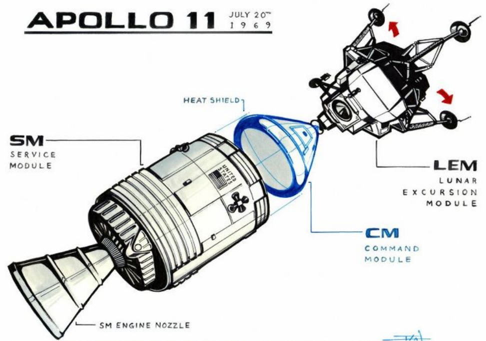
- Forensics
- Reversing
- Exploiting
- Cracking
- Analysis
- Emulation
- ....

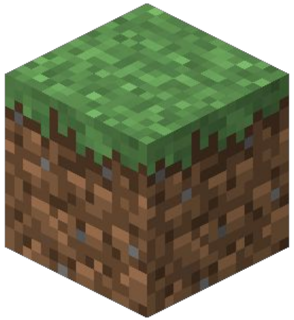
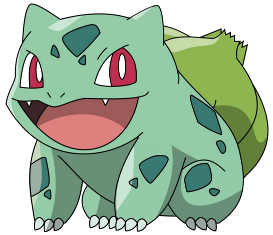
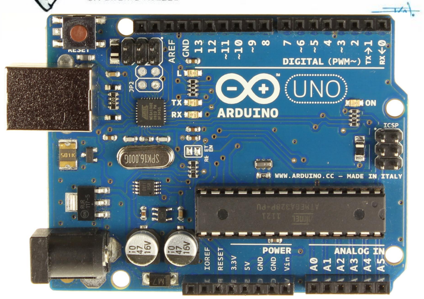


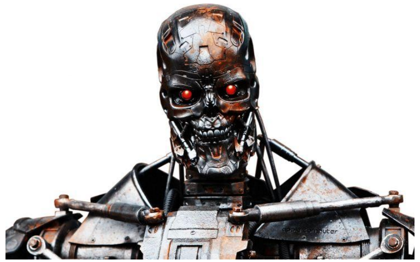
```
$ r2 /bin/ls
 -- Are you a wizard?
[0x100001200]> ?E :D
  .--.      .----.
  | _|      |    |
  | O O   <   :D |
  |  |  |   |    |
  || | /    `----'
  |`-'|
  `---'
[0x100001200]> 
```

$ where

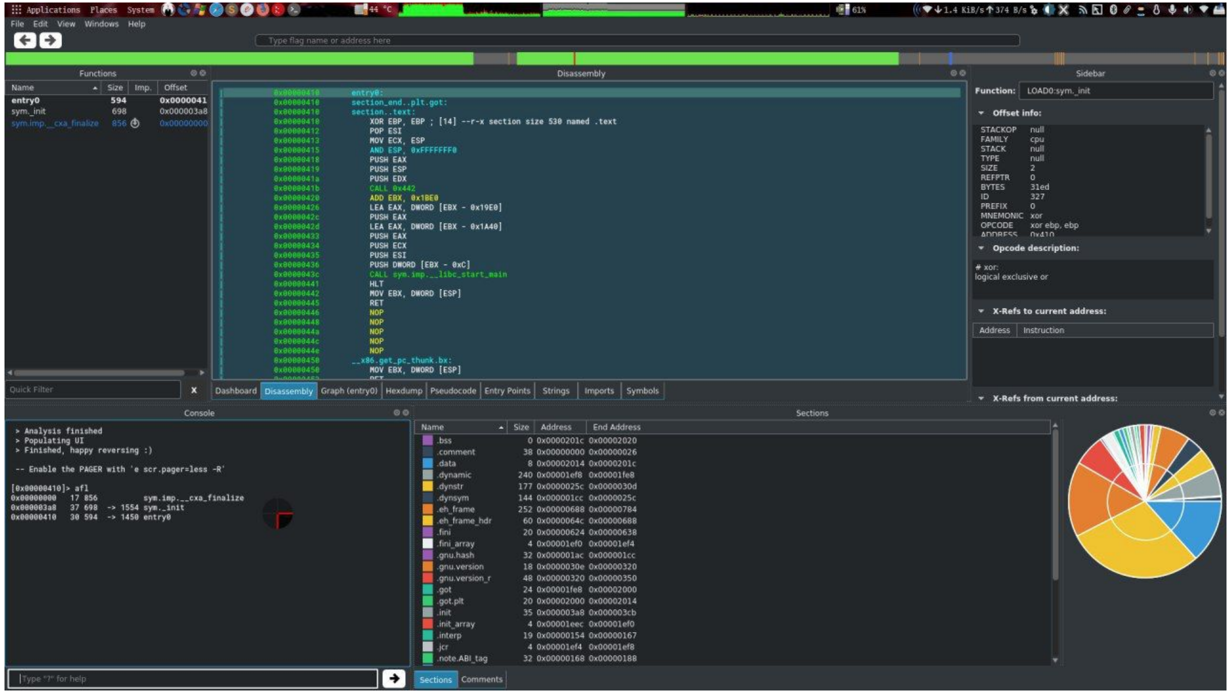
Anywhere!





$ startx



Name	Size	Imp.	Offset
entry0	594		0x0000041
sym._init	698		0x000003a8
sym.imp.__cxa_finalize	856		0x00000000

```
0x00000410        entry0:
0x00000410        section_end..plt.got:
0x00000410        section..text:
0x00000410            XOR EBP, EBP ; [14] --r-x section size 530 named .text
0x00000412            POP ESI
0x00000413            MOV ECX, ESP
0x00000415            AND ESP, 0xFFFFFFF0
0x00000418            PUSH EAX
0x00000419            PUSH ESP
0x0000041a            PUSH EDX
0x0000041b            CALL 0x442
0x00000420            ADD EBX, 0x1BE0
0x00000426            LEA EAX, DWORD [EBX - 0x19E0]
0x0000042c            PUSH EAX
0x0000042d            LEA EAX, DWORD [EBX - 0x1A40]
0x00000433            PUSH EAX
0x00000434            PUSH ECX
0x00000435            PUSH ESI
0x00000436            PUSH DWORD [EBX - 0xC]
0x0000043c            CALL sym.imp.__libc_start_main
0x00000441            HLT
0x00000442            MOV EBX, DWORD [ESP]
0x00000445            RET
0x00000446            NOP
0x00000448            NOP
0x0000044a            NOP
0x0000044c            NOP
0x0000044e            NOP
0x00000450        __x86.get_pc_thunk.bx:
0x00000450            MOV EBX, DWORD [ESP]
```

Function: LOAD0:sym._init

Offset info:

STACKOP	null
FAMILY	cpu
STACK	null
TYPE	null
SIZE	2
REFPTR	0
BYTES	31ed
ID	327
PREFIX	0
MNEMONIC	xor
OPCODE	xor ebp, ebp
ADDRESS	0x410

Opcode description:

xor:
logical exclusive or

X-Refs to current address:

Address	Instruction

X-Refs from current address:

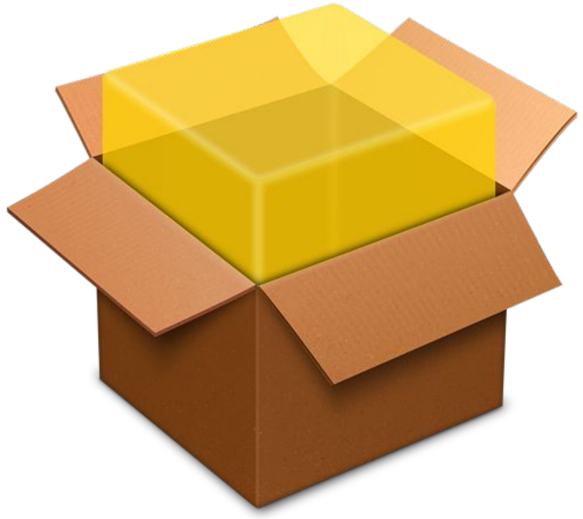
```
 > Analysis finished
 > Populating UI
 > Finished, happy reversing :)

 -- Enable the PAGER with 'e scr.pager=less -R'

[0x00000410]> afl
0x00000000   17 856          sym.imp.__cxa_finalize
0x000003a8   37 698  -> 1554 sym._init
0x00000410   30 594  -> 1450 entry0
```

Name	Size	Address	End Address
.bss	0	0x0000201c	0x00002020
.comment	38	0x00000000	0x00000026
.data	8	0x00002014	0x0000201c
.dynamic	240	0x00001ef8	0x00001fe8
.dynstr	177	0x0000025c	0x0000030d
.dynsym	144	0x000001cc	0x0000025c
.eh_frame	252	0x00000688	0x00000784
.eh_frame_hdr	60	0x0000064c	0x00000688
.fini	20	0x00000624	0x00000638
.fini_array	4	0x00001ef0	0x00001ef4
.gnu.hash	32	0x000001ac	0x000001cc
.gnu.version	18	0x0000030e	0x00000320
.gnu.version_r	48	0x00000320	0x00000350
.got	24	0x00001fe8	0x00002000
.got.plt	20	0x00002000	0x00002014
.init	35	0x000003a8	0x000003cb
.init_array	4	0x00001eec	0x00001ef0
.interp	19	0x00000154	0x00000167
.jcr	4	0x00001ef4	0x00001ef8
.note.ABI_tag	32	0x00000168	0x00000188

$ r2pm

Missing something? Just check out the package manager

- Frida
- R2k
- Tox
- Lldb
- Kaitai
- ...



$ gcc -undo

- Pdc
- Pdd
- Retdec
- Snowman

Decompiling a crackme

$ r2pipe

Easiest way to script r2

- Showing how to automate actions in r2

$ ESIL

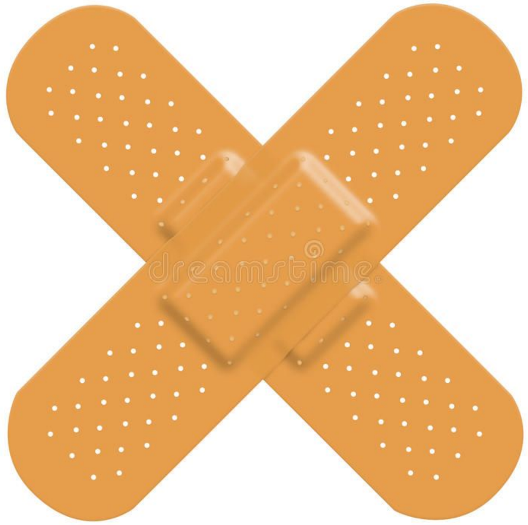
- IL of r2
- Used for emulation
- And other things..

- Demo bruteforcing password on arm binary with esil

$ r2fix

- Demo showing tool crashing and fixing the binary with r2

?TOOL EDITION



$ gc

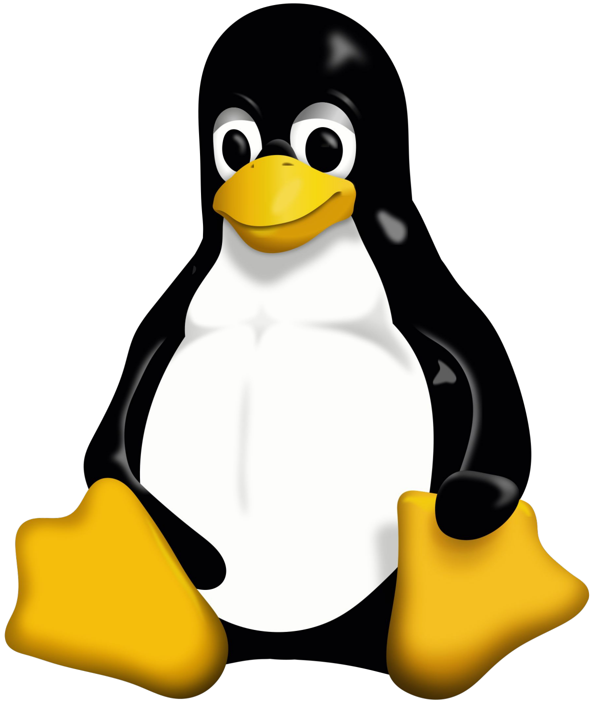
Cleaning up trash code is something common when doing RE

- Demo pdR with finfisher binary
- Reusing work from SkUaTeR

$ r2k

Raising priviledges to a bash process using r2k and volatility

- Dumping vbox ram
- Taking task_struct pointers from volatility’s linux_pslist
- Finding the struct cred pointer right before the process name
- Filling those 0x3e8 with zeroes



$ Questions?

kthxby