

BUILDING A CONCRETE ALTERNATIVE TO IDA

Radare2 to the rescue!

Jeffrey (crowell) Crowell – Julien (jvoisin) Voisin

June 20, 2015

REcon 2015 – Montreal



crowell

- Work at Google
- raxcity.com
- Shellphish
- Boston Key Party

jvoisin

- Soon graduated
- <redacted>
- dustri.org
- Knows some english

Professional

- IDA Pro
- ImmunityDBG
- WinDBG
-

Amateur

- IDA Pro
- WineDBG
- Hopper
- OllyDBG

Professional

- IDA Pro (\$5000)
- ImmunityDBG
- WinDBG
-

Amateur

- IDA Pro (pirated)
- WineDBG (pirated Windows)
- Hopper (probably not)
- OllyDBG (not maintained)

- Created by Ilfak Guilfanov





- Created by Ilfak Guilfanov
- First DataRescue, then Hex-Rays



- Created by Ilfak Guilfanov
- First DataRescue, then Hex-Rays
- Closed-source and expensive



- Created by Ilfak Guilfanov
- First DataRescue, then Hex-Rays
- Closed-source and expensive
- Lots of architectures are supported



- Created by Ilfak Guilfanov
- First DataRescue, then Hex-Rays
- Closed-source and expensive
- Lots of architectures are supported
- Decompilation!



- Created by Ilfak Guilfanov
- First DataRescue, then Hex-Rays
- Closed-source and expensive
- Lots of architectures are supported
- Decompilation!
- **Awesome** piece of software

RADARE2, CET INCONNU

- radare in 2006





- radare in 2006
- forensics tool



- radare in 2006
- forensics tool
- radare2 in 2009



- radare in 2006
- forensics tool
- radare2 in 2009
- written in pure C



- radare in 2006
- forensics tool
- radare2 in 2009
- written in pure C
- 350k LoC under *LGPL*



- radare in 2006
- forensics tool
- radare2 in 2009
- written in pure C
- 350k LoC under *LGPL*
- multi-purpose suite of tools



- likely packaged in your distribution



- likely packaged in your distribution
- install from source though ;-)



- likely packaged in your distribution
- install from source though ;-)
- more than 50 contributors for the latest release



- likely packaged in your distribution
- install from source though ;-)
- more than 50 contributors for the latest release
- RSoC (+GSoC)

- ragg2
- radiff2
- rabin2
- rafind2
- rahash2
- rarun2
- rasm2
- rax2
- radare2

Compile programs into tiny binaries for x86-32/64 and arm.

- ragg2
- **radiff2**
- rabin2
- rafind2
- rahash2
- rarun2
- rasm2
- rax2
- radare2

Binary diffing

- ragg2
- radiff2
- rabin2
- rafind2
- rahash2
- rarun2
- rasm2
- rax2
- radare2

Binary program info extractor (think *readelf*)

- ragg2
- radiff2
- rabin2
- **rafind2**
- rahash2
- rarun2
- rasm2
- rax2
- radare2

Search for byte patterns in files

- ragg2
- radiff2
- rabin2
- rafind2
- **rahash2**
- rarun2
- rasm2
- rax2
- radare2

Block based hashing utility

- ragg2
- radiff2
- rabin2
- rafind2
- rahash2
- rarun2
- rasm2
- rax2
- radare2

Run programs in exotic environments

- ragg2
- radiff2
- rabin2
- rafind2
- rahash2
- rarun2
- rasm2
- rax2
- radare2

Assembler/disassembler

- ragg2
- radiff2
- rabin2
- rafind2
- rahash2
- rarun2
- rasm2
- rax2
- radare2

Base converter

- ragg2
- radiff2
- rabin2
- rafind2
- rahash2
- rarun2
- rasm2
- rax2
- radare2

Combine **everything** together

Runs on

- Windows
- GNU/Linux
- *BSD
- OSX
- Android and iOS
- Smartwatch
- Web browser
- QNX
- ...

Handles

- MZ/PE+/PE/COFF
- ELF, ELF64
- Fatmach0/Mach0
- DEX/JAVA
- BIOS/TE
- GB/GBA/DS
- XBOX
- Plan9
- BIOS

- 8051
- arc
- arm
- avr
- brainfuck
- cr16
- csr
- dalvik
- dcpu16
- ebc
- gb
- h8300

- i4004
- i8080
- java
- LH5801
- m68k
- malbolge
- mips
- msil
- msp430
- nios2
- powerpc
- rar

- ART
- sh
- sparc
- spc700
- sysz
- tms320
- v850
- whitespace
- x86
- xcore
- z80
- propeller
- snes
- psosvm
- 6502

R2 INTERNALS

```
[1] pry(main)> require 'r2pipe';  
[2] pry(main)> r2p = R2Pipe.new '/bin/ls';  
[3] pry(main)> puts r2p.cmd 'ie'; # print entrypoint  
[Entrypoints]  
vaddr=0x004048c5 paddr=0x000048c5 baddr=0x00400000 laddr=0x00000000  
  
1 entrypoints
```

Bindings are boring, let's call r2 instead!

3rd party (or 1st party) plugins

- *r_asm*, assembler and disassembler
- *r_anal*, code analysis (opcode, type, esil)
- *r_reg*, registers
- *r_syscall*, system calls
- *r_debug*, debugger
- *r_io*, io layer
- *r_search*, search engine
- ...

FEATURE COMPARISON

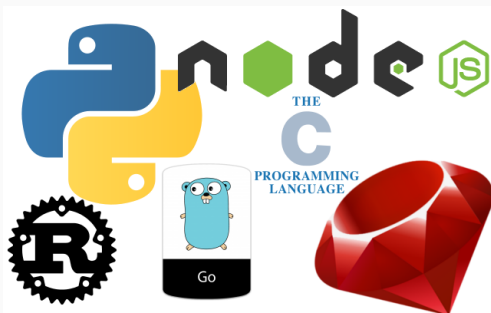
IDA HAS A BOOK, R2 IS SELF-DOCUMENTED (AND ALSO HAS A BOOK TOO)

```
[0x00000000]> aa?  
| Usage: aa[0*?] # see also 'af' and 'afna'  
| aa      alias for 'af@@ sym.*;af@entry0'  
| aa*    analyze all flags starting with sym. (af @@ sym.*)  
| aaa    autoname functions after aa (see afna)  
| aac [len] analyze function calls (af @@ `pi len~call[1]`)  
| aas [len] analyze symbols (af @@= `isq~[0]`)  
| aap    find and analyze function preludes  
[0x00000000]> █
```

- R2 is like vim
- Combine *intuitives* commands
- Just append ? everywhere

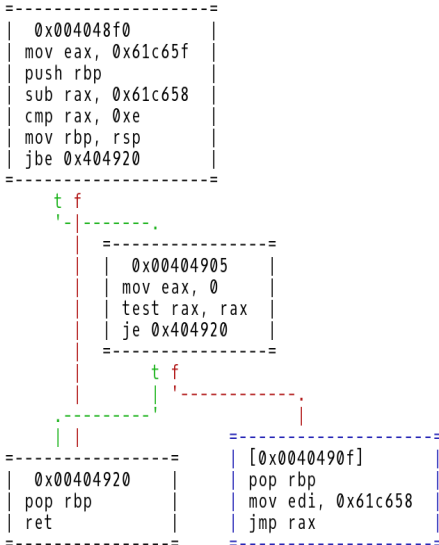
IDA HAS PLUGINS, R2 HAS MORE BINDINGS

- Python
- NodeJS
- C
- Lua
- Lisp
- Vala
- Ruby
- Go
- Rust
- Perl
- OCaml
- ...



IDA HAS SOME GRAPHS, R2 DOES TOO (BUT IN ASCII)

- Minimap
- Debugger-compliant
- Interactive



IDA IS CLEVER BUT ALSO INTERACTIVE, SO IS R2

- name functions
- mark *flags*
- define code/data
- leave comments
- name stack variables
- mark structures
- use types
- define/modify functions

```
[0x00401393 18% 190 /bin/false]> pd $r @ main+3 # 0x401393
    push rbx
    je 0x4013a0 ;[1]
    ; JMP XREF from 0x00401401 (section..text)
    ; JMP XREF from 0x00401429 (section..text)
    mov edi, 1
    call sym.imp.exit ;[2]
    ^- sym.imp.exit(unk)
    ; JMP XREF from 0x00401394 (section..text)
    mov rdi, qword [rsi]
    mov rbx, rsi
    call fcn.00401900 ;[3]
    ^- fcn.00401900()
    mov esi, 0x403f0e
    mov edi, 6
    call sym.imp.setlocale ;[4]
    ^- sym.imp.setlocale()
    mov esi, str._usr_share_locale
    ^- ; "/usr/share/locale" @ 0x403f97
    mov edi, 0x403f89 ; "coreutils" @ 0x403f89
```

```
[0x004048c5]>
```

It's not all that scary!

- *Visual Mode* - friendly enough?
- Familiar vim keybindings.
- *Web UI* - The future of collaborative reversing!
- Communicate over r2pipe.

IDA HAS AN OLD-SCHOOL TUI MODE, R2 HAS A BETTER ONE.

- Ncurses-like
- Static
- Dynamic
- Analysis
- Try it, really.

```
asm.size = false
asm.stackptr = false
asm.syntax = intel
asm.tabs = 0
asm.trace = false
asm.tracespace = false
asm.ucase = false
asm.vars = true
asm.varsub = true
> asm.varxs = false
asm.xrefs = true
```

Selected: asm.varxs (Show accesses of local variables)

```
;- entry0:
0x004048c5 31ed xor ebp, ebp
0x004048c7 4989d1 mov r9, rdx
0x004048ca 5e pop rsi
0x004048cb 4889e2 mov rdx, rsp
0x004048ce 4883e4f0 and rsp, 0xfffffffffffffff0
```

IDA HAS NO WEB-UI, R2 DOES.

The screenshot displays the Radare2 disassembler interface. The main window shows a control flow graph (CFG) with several nodes connected by arrows. The nodes contain assembly code snippets:

- Node 1 (0x40094c):

```
0x40094c  call sym.imp.puts
0x400951  call sym.imp.exit
```
- Node 2 (0x400956):

```
; JMP XREF from 0x400940
0x400956  lea rdx, [rbp-local_14]
0x40095a  mov rax, qword [rbp-local_16]
0x40095e  mov esi, 0x1000
0x400963  mov rdi, rax
0x400966  call sym.imp.SHA256
0x40096b  test rax, rax
0x40096e  jne 0x400984
```
- Node 3 (0x400970):

```
0x400970  mov edi, str.__error: could_not_create_SHA256
entry0
```
- Node 4 (0x400975):

```
0x400975  call sym.imp.puts
0x40097a  mov edi, 1
0x40097f  call sym.imp.exit
```
- Node 5 (0x400984):

```
; JMP XREF from 0x40096e
0x400984  mov dword [rbp-local_17], 0
0x40098e  jmp 0x4009cd
```
- Node 6 (0x4009cd):

```
; JMP XREF from 0x40098e
0x4009cd  cmp dword [rbp-local_17], 0x1f
0x4009d4  jle 0x400990
```
- Node 7 (0x400990):

```
; JMP XREF from 0x4009d4
0x400990  mov eax, dword [rbp-local_17]
0x400996  cdb
```

The interface includes a left sidebar with a Functions list (24 items), a Symbols list (42 items), Relocs (12), Imports (14), and Flags (130). The bottom status bar shows the current function is 'main' and the current instruction is 'switch view'. The right sidebar displays system information for the file being analyzed, including type (EXEC), file path, fd (6), size (0x2310), mode (r--), block (0x100), format (elf64), pic (false), canary (true), nx (true), crypto (false), va (true), bintype (elf), class (ELF64), lang (c), arch (x86), bits (64), machine (AMD x86-64), os (linux), subsys (linux), endian (little), stripped (false), static (false), linenum (true), lsyms (true), relocs (NONE), rpath (true), and binsz (7052).

IDA HAS A DEBUGGER, SO DOES R2

- Classic features
- Visual mode too
- Several backends
- Tracing
- Remote

```
- offset -      0 1 2 3 4 5 6 7 8 9 A B C D 0123456789ABCD
0x7ffd1aa98010 c2a4 a91a fd7f 0000 7885 a91a fd7f .....X.....
0x7ffd1aa9801e 0000 c548 4000 0000 0000 20ff f3ca ...H@.....
0x7ffd1aa9802c 987f 0000 3096 f2ca 987f 0000 b237 ....0.....7
0x7ffd1aa9803a f3ca 987f 0000 0d00 0000 0000 0000 .....
0x7ffd1aa98048 3096 f2ca 987f 0000                                0.....
r15 0x00000000                                r14 0x00000000                                r13 0x7ffd1aa98570
r12 0x004048c5                                rbp 0x7ffd1aa98578                                rbx 0x7ffd1aa9a4c2
r11 0x7f98ca752950                            r10 0x7f98cb14e188                                r9 0x7f98caf399d0
r8 0x7f98caaf7dd0                             rax 0x00000000                                rcx 0x00000000
rdx 0x7ffd1aa98588                            rsi 0x00000027                                rdi 0x7f98cb14e188
orax 0xffffffffffffffff rip 0x7f98caf388b9                                rflags = 1PI
rsp 0x7ffd1aa98010
                                0x7f98caf388b0                                4154                                push r12
                                0x7f98caf388b2                                55                                push rbp
                                0x7f98caf388b3                                4989fa                                mov r10, rdi
                                0x7f98caf388b6                                53                                push rbx
                                0x7f98caf388b7                                89f6                                mov esi, esi
;-- rip:
0x7f98caf388b9                                488d1476                                lea rdx, [rsi + rsi*2]
0x7f98caf388bd                                4883ec10                                sub rsp, 0x10
0x7f98caf388c1                                488b4768                                mov rax, qword [rdi + 0x68]
0x7f98caf388c5                                488b7808                                mov rdi, qword [rax + 8] ;
0x7f98caf388c9                                498b82f80000                                mov rax, qword [r10 + 0xf8] ;
0x7f98caf388d0                                488b4008                                mov rax, qword [rax + 8] ;
0x7f98caf388d4                                4c8d04d0                                lea r8, [rax + rdx*8]
0x7f98caf388d8                                498b4270                                mov rax, qword [r10 + 0x70]
0x7f98caf388dc                                498b4808                                mov rcx, qword [r8 + 8] ;
0x7f98caf388e0                                488b4008                                mov rax, qword [rax + 8] ;
```

IDA HAS KICK-ASS ANALYSIS, R2 HAS SOME TOO

- Functions detection
- Local var detection
- FLIRT integration
- *signatures*
- (X)REF
- DWARF and PDB

```
[0x00401393 18% 190 /bin/false]> pd $r @ main+3 # 0x401393
    push rbx
    je 0x4013a0 ;[1]
    ; JMP XREF from 0x00401401 (section..text)
    ; JMP XREF from 0x00401429 (section..text)
    mov edi, 1
    call sym.imp.exit ;[2]
    ^- sym.imp.exit(unk)
    ; JMP XREF from 0x00401394 (section..text)
    mov rdi, qword [rsi]
    mov rbx, rsi
    call fcn.00401900 ;[3]
    ^- fcn.00401900()
    mov esi, 0x403f0e
    mov edi, 6
    call sym.imp.setlocale ;[4]
    ^- sym.imp.setlocale()
    mov esi, str._usr_share_locale
    ^- ; "/usr/share/locale" @ 0x403f97
    mov edi, 0x403f89 ; "coreutils" @ 0x403f89
```

- ESIL
- RPN-ish
- Documented
- Emulation
- Decompilation
- Analysis

```
edi, edi, ^=, %z, zf, =, %p, pf, =, 0, cf, =, 0, of, =, %s, sf, =  
0x417cc0, 4, *, [4], esi, =  
5, rip, +, 8, rsp, -=, rsp, =[], 4254240, rip, =  
4280671, edi, =  
80, 0x21a799, rip, +, =[8]  
5, rip, +, 8, rsp, -=, rsp, =[], 4202944, rip, =  
rax, r12, =  
0x40, rsp, +, rax, =  
0, r12, r12, &, =, %z, zf, =, %p, pf, =, %s, sf, =, 0, cf, =, 0, of, =  
rax, 0x28, rsp, +, =[8]  
zf, ?{, 4205201, rip, =, }  
0, r12, [1], =, %z, zf, =, %b8, cf, =, %p, pf, =, %s, sf, =  
zf, !, ?{, 4208508, rip, =, }  
0x28, rsp, +, [8], rdx, =  
eax, eax, ^=, %z, zf, =, %p, pf, =, 0, cf, =, 0, of, =, %s, sf, =  
21523, esi, =  
1, edi, =
```

IDA HAS PLUGINS FOR PWNAGE, R2 PUT THIS IN CORE

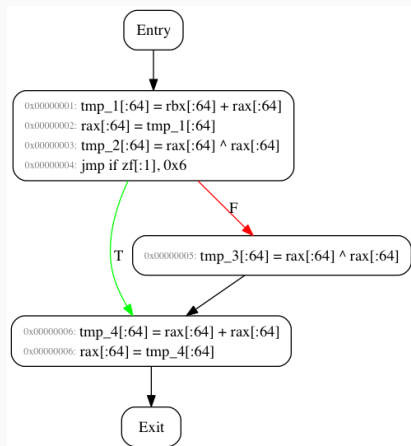
- Regexp ROP hunter
- Mitigations detection
- Emulation
- Patterns
- Environment control

```
[0x00402a35]> "/R/ pop r1.;pop.*;pop"  
0x0040524e 415c pop r12  
0x00405250 415d pop r13  
0x00405252 415e pop r14  
0x00405254 415f pop r15  
0x00405256 c3 ret  
  
0x0040524f 5c pop rsp  
0x00405250 415d pop r13  
0x00405252 415e pop r14  
0x00405254 415f pop r15  
0x00405256 c3 ret  
  
0x00405443 415d pop r13  
0x00405445 415e pop r14  
0x00405447 415f pop r15  
0x00405449 5d pop rbp  
0x0040544a c3 ret  
  
0x00405444 5d pop rbp  
0x00405445 415e pop r14  
0x00405447 415f pop r15  
0x00405449 5d pop rbp  
0x0040544a c3 ret
```


SUMMARY

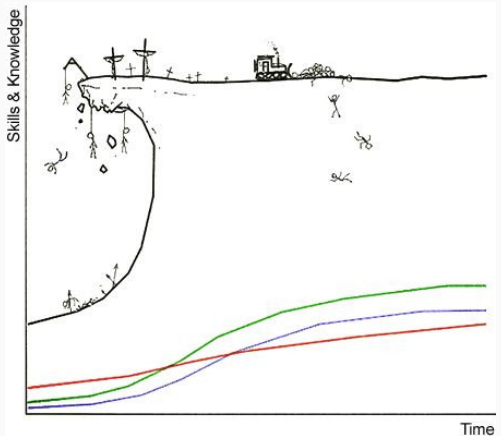
AND NOW?

- GSoC
- Stabilization
- A fresh release
- Second edition of our RSoC
- ~1000 LoC modified per week



CURRENT DRAWBACKS

- **Super**-steep learning curve
- A lot of features
- Fast-moving target
- IDA is friendlier



- Free-software
- Exotic arch support
- Active development
- A lot of features
- More and more users



WHO USES R2 CURRENTLY?

- Some top-notch ctf teams
 - Shellphish
 - Dragon Sector
 - ...
- Anti-malware companies
 - AlienVault
 - IOActive
 - ...
- Some popular RE projects
 - Coreboot
 - Magic lantern
 - ...
- Cool wargames
 - io from smashthestack
 - OverTheWire
 - ...

WHO USES R2 CURRENTLY?

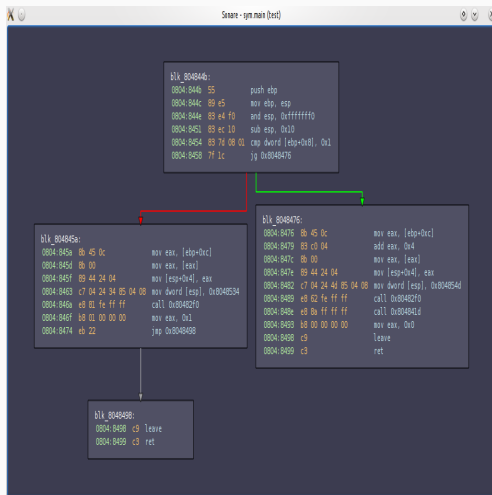
- Some top-notch ctf teams
 - Shellphish
 - Dragon Sector
 - ...
- Anti-malware companies
 - AlienVault
 - IOActive
 - ...
- Some popular RE projects
 - Coreboot
 - Magic lantern
 - ...
- Cool wargames
 - io from smashthestack
 - OverTheWire
 - ...

We do!

Do you?

AND TOMORROW?

- Complete-emulation
- Decompilation
- A complete GUI
- What do you want?



Question IDA supremacy¹.
Monoculture is bad.

¹And don't pirate it!

Radare2 is nice.
You should use it.¹

¹Or at least try it

- TV channel - <http://radare.tv/>
- Book - <http://maijin.gitbooks.io/radare2book/content/>
- Blog - <http://radare.today/>
- Homepage - <http://rada.re/>
- Source code - <http://github.com/radare/radare2/>
- IRC channel - <irc://irc.freenode.net/radare>

Come **talk** to us!

QUESTIONS?