Trustwave®
Smart security on demand

Trustwave® SpiderLabs®

Reverse Engineering with Radare2

How to start

Jarosław Górny
Vulnerability Researcher, SpiderLabs Research

October 15, 2017

Agenda

1. Why this talk
2. What is Reverse Engineering and why radare2
3. Installation and basics of radare2
4. 101 of Reverse Engineering on x86
5. Static analysis
6. Dynamic analysis

Who am I?

handles: jaroslav on freenode.net IRC; @JaroslavNahorny on twitter

work: Trustwave's SpiderLabs Research

- I work as a Vulnerability Researcher

HS-WAW, aka hackerspace.pl

- former member, still a big fan, I highly recommend!
- #hackerspace-pl @ freenode.net IRC

CTF

- playing with an awesome team, Dragon Sector (as a guest)

Why this talk?

- **I started with assembly when I was a kid (on Commodore 64)**
- **Then a bit of microprocessor assembly at the university**
- **I thought it would be cool to understand it on x86 too**
  - CTF competitions, crackmes turned out to be a lot of fun
- **Radare2 seemed to be a reasonable pick**
  - Free (as "free beer", but "freedom of speech" too)
  - I like command line ☺
  - People say it's hard, and I like challenges ☺

  - *I use Emacs, I'm weird; that can be a part of the reason too*
- **Disclaimer: I still consider myself as a beginner in RE field**

RE, radare2, basic facts

Reverse Engineering

What do we mean by this?

In general: the process of extracting knowledge or design information from anything man-made (Wikipedia)

In our case: we are "extracting knowledge" from a binary software (executables, libraries), by disassembling or decompiling it

What we want to achieve?

```
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x100000f20  5548 89e5 897d fc89 75f8 8b75 fc03 75f8  UH...}..u..u..u.
0x100000f30  89f0 5dc3 6666 662e 0f1f 8400 0000 0000  ..].fff.........
0x100000f40  5548 89e5 4883 ec20 b802 0000 00b9 0300  UH..H.. ........
0x100000f50  0000 897d fc48 8975 f089 c789 cee8 beff  ...}.H.u........
```

disassembly

decompilation

```
[0x100000f20] ;[ga]
      ; section 0 va=0x100000f20 pa=0x00000f20 sz=95 vsz=95 rwx=m-r-x 0.__TEXT.__text
   ;-- section.0.__TEXT.__text:
   ;-- func.100000f20:
(fcn) sym._add 20
   sym._add ();
; var int local_8h @ rbp-0x8
; var int local_4h @ rbp-0x4
    ; CALL XREF from 0x100000f5d (entry0)
0x100000f20 55              push rbp
0x100000f21 4889e5          mov rbp, rsp
0x100000f24 897dfc          mov dword [local_4h], edi
0x100000f27 8975f8          mov dword [local_8h], esi
0x100000f2a 8b75fc          mov esi, dword [local_4h]
0x100000f2d 0375f8          add esi, dword [local_8h]
0x100000f30 89f0            mov eax, esi
0x100000f32 5d              pop rbp
0x100000f33 c3              ret
```

```
int add(int a, int b) {
        return a + b;
}
```

Decompilation is not that easy…

What is radare2

According to https://radare.org/r/

Radare is a **portable reversing framework** that can...

- **Disassemble (and assemble for) many different architectures**
- **Debug with local native and remote debuggers (gdb, rap, webui, r2pipe, winedbg, windbg)**
- **Run on Linux, *BSD, Windows, OSX, Android, iOS, Solaris and Haiku**
- Perform forensics on filesystems and data carving
- **Be scripted in Python, Javascript, Go and more**
- Support collaborative analysis using the embedded webserver
- Visualize data structures of several file types
- **Patch programs to uncover new features or fix vulnerabilities**
- Use powerful analysis capabilities to speed up reversing
- Aid in software exploitation

Radare2: Installation and First Steps

https://twitter.com/r2gif/status/892135663289630720

Installation: always use version from git

Just:

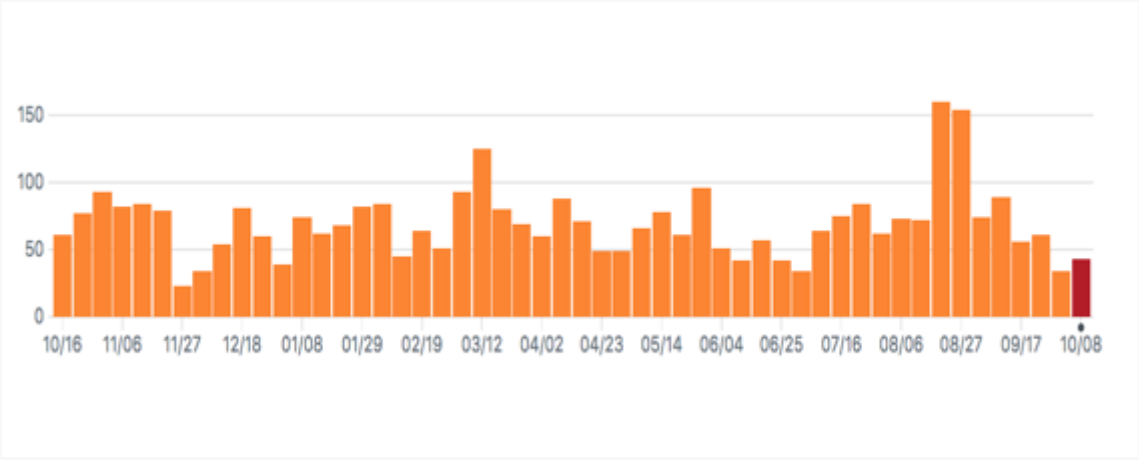
```
$ git clone
https://github.com/radare/radare2

$ cd radare2

$ ./sys/install.sh

$ r2 –v

radare2 2.0.1 16334 @ linux-x86-64
git.2.0.1-1-g00fdad81bcommit:
00fdad81b99ae9a010948e6f484e8aff994
d9915 build: 2017-10-13__11:15:44
```



n of commits per week

Installation: if you really, really, really can't from git

You can find latest binaries for many architectures here:

http://rada.re/r/down.html

What do we get?

Apart of radare2 itself (which is also aliased simply as 'r2'):

- Package manager (115 packages at the moment): **r2pm**
- Binary program info extractor: **rabin2**
- Binary diffing utility: **radiff2**
- Find byte patterns: **rafind2**
- Calculate, check and show hashes: **rahash2**
- Utility to run programs in exotic environments: **rarun2**
- Assemble / disassemble on the fly: **rasm2**
- Convert between bases, simple calc: **rax2**

What do we get? Examples

rasm2

```
$ rasm2 -a sparc -e -d "9de3bf90"

save sp, -0x70, sp

$ rasm2 -a arm "add r0, r0, #2”

020080e2

$ rasm2 -a x86 "add rsp, 0x20”

4883c420
```

What do we get? Examples

rax2

```
$ rax2 -s 41 42 43 0A
ABC
$ rax2 -S "flag{radare2}”
666c61677b726164617265327d
```

```
rax2 =16 0x09+0x02
0xb
$ rax2 0x09+0x02
11
$ rax2 =2 0x09+0x02
1011b
```

What do we get? Examples

rafind2

Many nice features:

- Search for specific string,
- Regular expressions,
- Limit to specific file range,
- Known file-format carving

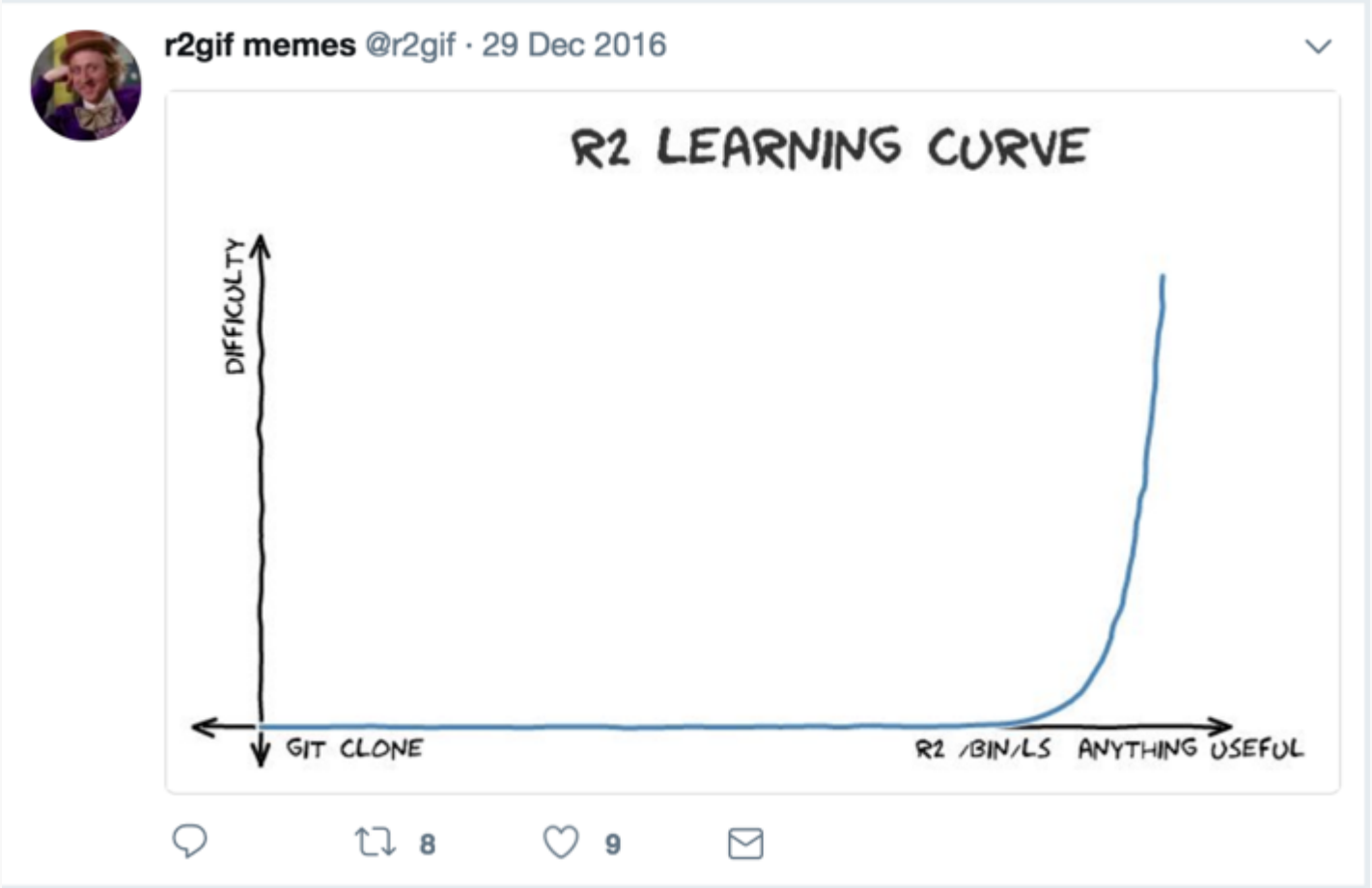
```
✔ 16:10 ~/priv/ctf/scs-2017-ctf/stego_big_foot $ rafind2 -m WhereIsBigFoot.jpg
-- 0 1e17f2
0x00000000 0x00000000 1 JPEG image , EXIF standard
0x0000000c 0x0000000c 1 TIFF image data, big-endian
0x00027776 0x00027776 1 FreeBSD/i386 executable not stripped
0x0003a682 0x0003a682 1 Macromedia Flash data, version 242
0x001e16fc 0x001e16fc 1 ZIP Zip archive data, at least v2.0 to extract
0x00000000 1 JPEG image , EXIF standard
0x0000000c 1 TIFF image data, big-endian
0x00027776 1 FreeBSD/i386 executable not stripped
0x0003a682 1 Macromedia Flash data, version 242
0x001e16fc 1 ZIP Zip archive data, at least v2.0 to extract
✔ 16:11 ~/priv/ctf/scs-2017-ctf/stego_big_foot $
```

What do we get? Examples

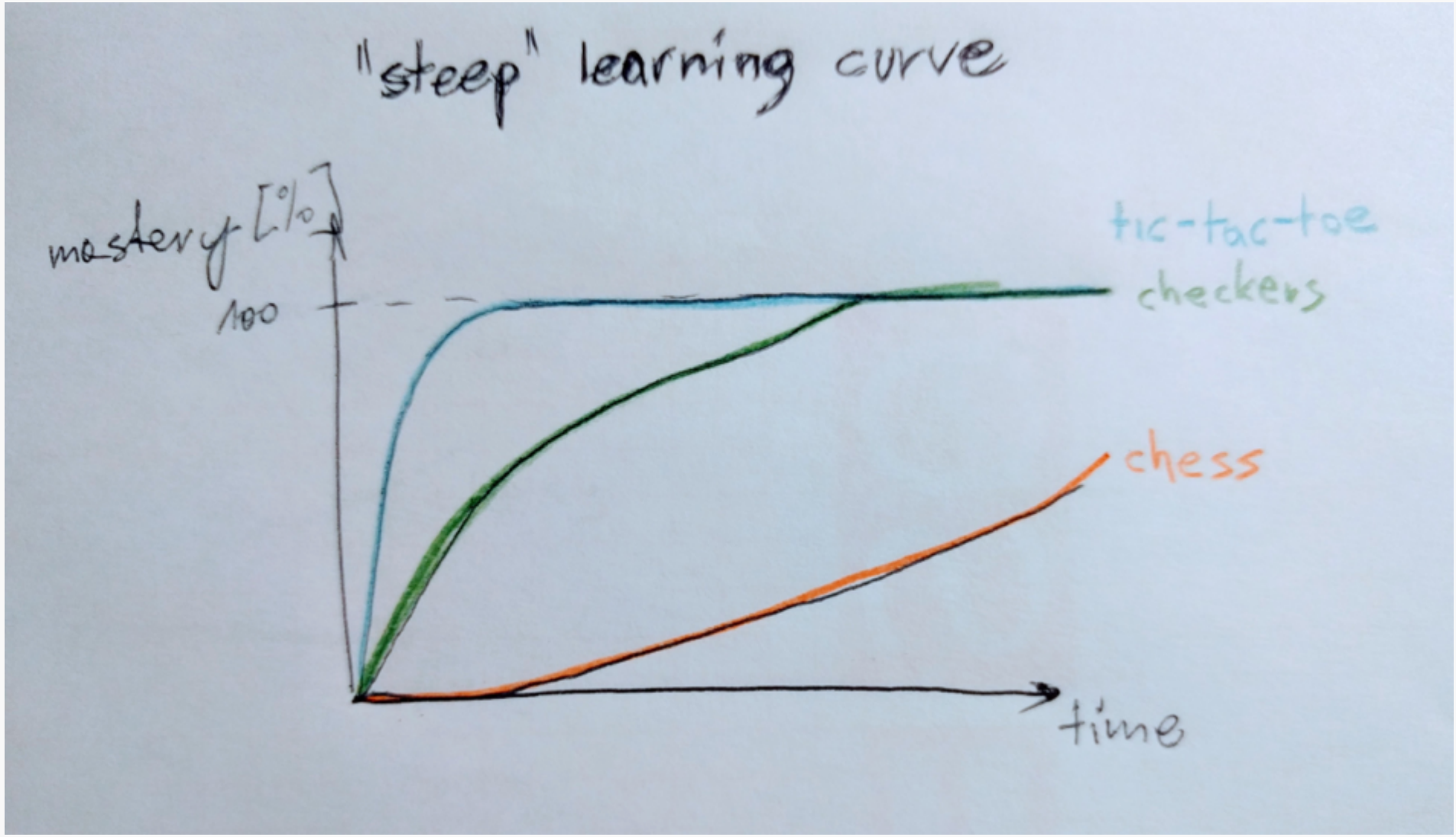
rabin2

```
$ rabin2 -I <binary>          # show info
$ rabin2 -i <binary>          # show imports
$ rabin2 -l <binary>          # linked libraries
$ rabin2 -S <binary>          # show sections
$ rabin2 -z <binary>          # show strings
```

The learning curve of radare2



The learning curve of radare2



The learning curve of radare2

Some people say radare2 doesn't have ANY learning curve at all. If you have a problem, you just ask on IRC, and somebody (usually @pancake) replies with radare2 commands.

(I saw it on twitter, can't find the author ☹)

That's how it looks on IRC....

```
[15:45] <   jaroslav> | lol, where's my cfg.write? ;)
[15:46] <     r2tgirc>| <pancake> Aaaaand its gone
[15:46] <     r2tgirc>| <pancake> Use oo+
```

Let’s dive into the command line

(demo)

My basic ~/.radare2rc

```
e cfg.fortunes = true

e cfg.fortunes.clippy = true

# e asm.pseudo = true

e cmd.stack = true

eco solarized

e scr.utf8 = true

e scr.utf8.curvy = true
```

101 of Reverse Engineering on x86 (mostly)

Disassembly Syntax: Intel vs AT&T

When you want to see a disassembly, using objdump:

```
objdump –d –M intel binary
objdump -d -x86-asm-syntax=intel binary
```

(on macOS) ¯_(ツ)_/¯

Or, if you have source, and want to see how it's going to be compiled:

```
gcc -S -masm=intel program.c
```

Intel vs AT&T (cont.)

AT&T

```
pushq %rbp
movq %rsp, %rbp
subq $0x20, %rsp
movl $2, %eax
movl $3, %ecx
movl %edi, local_4h
movq %rsi, local_10h
movl %eax, %edi
movl %ecx, %esi
```

Intel

```
push rbp
mov rbp, rsp
sub rsp, 0x20
mov eax, 2
mov ecx, 3
mov dword [local_4h], edi
mov qword [local_10h], rsi
mov edi, eax
mov esi, ecx
```

Calling conventions (x86)

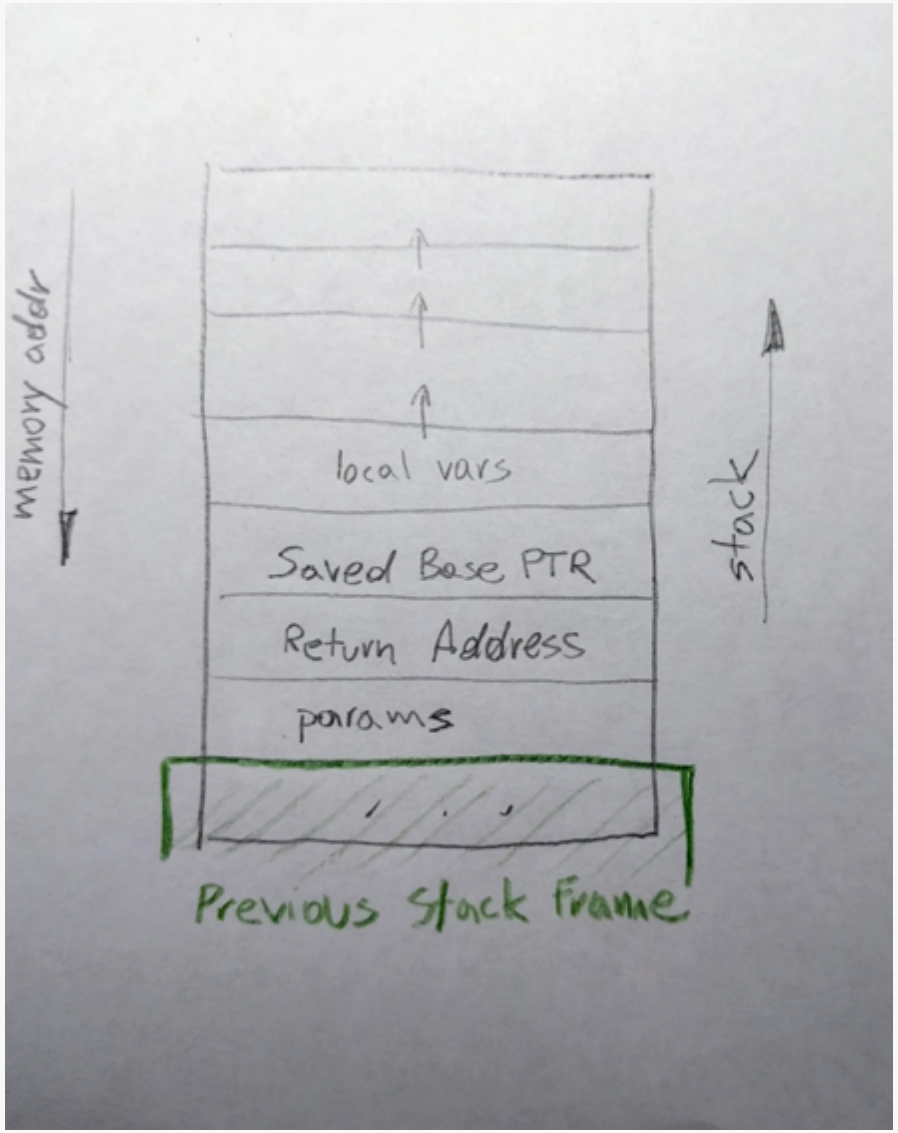
x86 (32 bit)

name	parameters	cleaning the stack
cdecl	on the stack	caller
stdcall	on the stack	callee
fastcall	ECX, EDX, stack	callee

x86-64 (64 bit)

name	parameters	cleaning the stack
?	RDI, RSI, RDX, RCX, R8, R9, stack	caller

How does the stack work



Let’s see…

(DEMO)

Good reads

Smashing The Stack For Fun And Profit

- 1996 ☺, by Aleph One (Elias Levy)
- http://phrack.org/issues/49/14.html#article

Smashing The Stack For Fun &Profit : Revived

- 2016, by avicoder
- https://avicoder.me/2016/02/01/smashsatck-revived/

Today: canary, nx, aslr, relro…

- https://en.wikipedia.org/wiki/Stack_buffer_overflow

Static analysis

Static or dynamic analysis, what does it mean?

Static:

- Without actually executing programs
  - Safer
  - Can be done for an architecture / configuration we don't have access

Dynamic:

- With executing programs on a real or virtual processor
  - Allows to inspect actual state of the CPU / memory step by step
  - Can give "wrong" results (debugger evasion)
  - Can reveal some extra information, helpful in finding vulnerabilities (eg. fuzzing)

Let's break something!

Example 1

Very simple binary (you'd be surprised ☺)

A fancy program for opening protected PDF files

Example 2 – “cracking”

Binary Patching (and diffing ;))

And what, if we don’t need to know the password, we just want to pass through the check

Example 3

Automation…

Q: What if reversing “by hand” is a bit of pain?

A: Automation!

Dynamic analysis

Example 4

Automation again

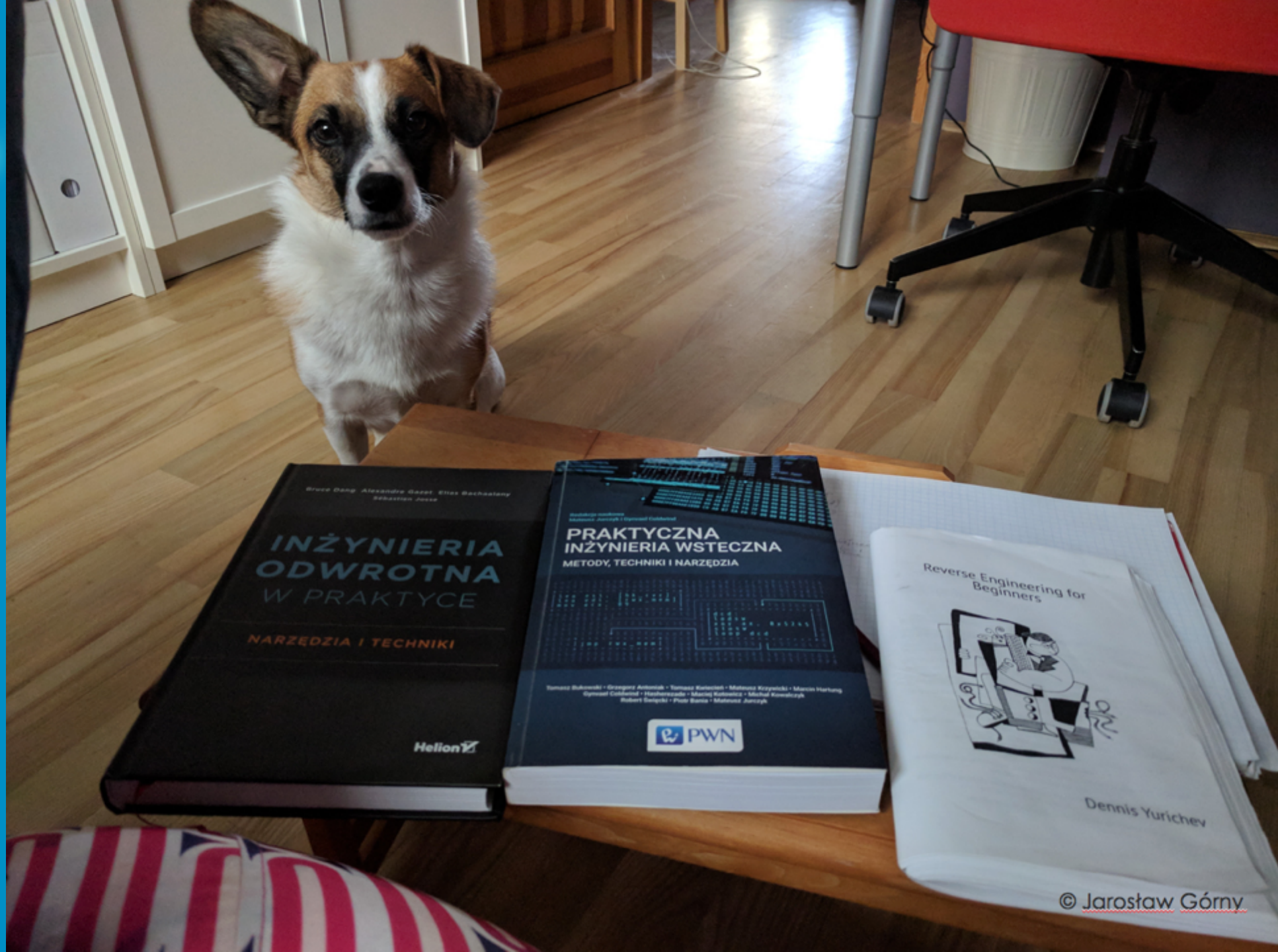
The same binary, slightly different way…

Summary

I hope I've managed to convince you…

- Reverse engineering is not a black magic;
- Radare2 is not **that** hard;
  - Be aware of it's specific
  - Use version from git ☺
  - Visit #radare on freenode
  - Remember about the "?" command
  - Don't use "aaaaaaaaa" blindly

Study!


INŻYNIERIA ODWROTNA
W PRAKTYCE
NARZĘDZIA I TECHNIKI
Helion
PRAKTYCZNA INŻYNIERIA WSTECZNA
METODY, TECHNIKI I NARZĘDZIA
PWN
Reverse Engineering for Beginners
Dennis Yurichev
© Jarosław Górny

Bibliography, useful links

RE in general

- B. Dang, A. Gazet & others: *Practical Reverse Engineering*
  - http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118787315,subjectCd-CSJ0.html
- (red) G. Coldwind, M. Jurczyk: Praktyczna inżynieria wsteczna
  - https://ksiegarnia.pwn.pl/Praktyczna-inzynieria-wsteczna,622427233,p.html
- D. Yurichev: *RE for beginners*
  - https://beginners.re/
- Intel 64 and IA-32 Architectures Software Developer Manuals
  - https://software.intel.com/en-us/articles/intel-sdm
- CTFTime: https://ctftime.org

Bibliography, useful links

Radare2 specific

- http://rada.re/r/docs.html
  - https://radare.gitbooks.io/radare2book/content/ (can be a bit outdated)
- http://radare.today/posts/analysis-by-default/ (and in general http://radare.today)
- http://rada.re/r/cmp.html (feature comparison, a bit biased ☺)

- https://twitter.com/r2gif

Questions?

IRC (freenode.org): jaroslav **| |** **twitter: @JaroslavNahorny**

T.Hanks!

Source: Wikipedia, PD-USGov lic.

IRC (freenode.org): jaroslav || twitter: @JaroslavNahorny