Security Software Development

Introduction and Low Level

Daniel Gruss, Vedad Hadzic, Martin Schwarzl, Samuel Weiser

02.10.2020

Winter 2020/21, www.iaik.tugraz.at

Introduction

Motivation

An error that effects only one bit can be enough to give an attacker full privileges on your computer

Motivation

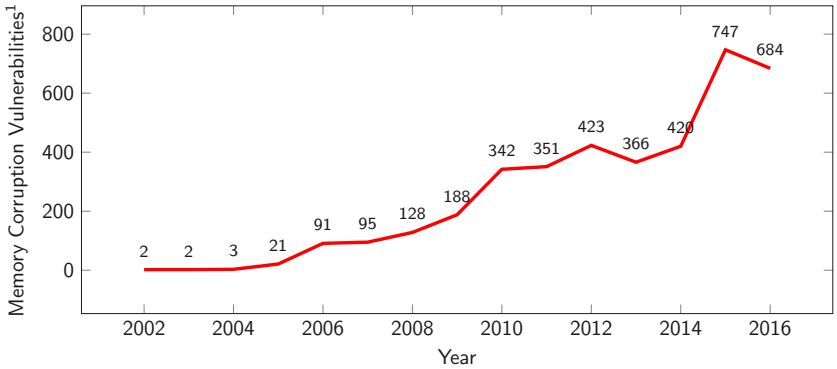
- Many programs are written in native code
- Programs get more and more complex

Motivation

"Securing a computer system has traditionally been a battle of wits: the penetrator tries to find the holes, and the designer tries to close them."

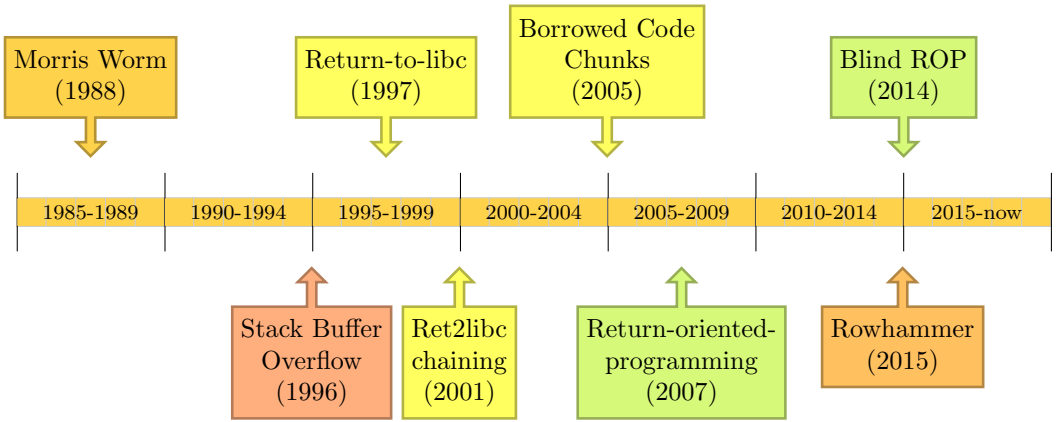
— M. Gosser

The complexer the programs, the more bugs



[1]Source: http://www.cvedetails.com/vulnerabilities-by-types.php

- We had three decades of memory corruption attacks



aka “Do we really have to care about that?”

Zero-day - Wikipedia

A zero-day (also known as zero-hour or 0-day or day zero) vulnerability

Zero-day - Wikipedia

A zero-day (also known as zero-hour or 0-day or day zero) vulnerability is an undisclosed computer-software vulnerability

Zero-day - Wikipedia

A zero-day (also known as zero-hour or 0-day or day zero) vulnerability is an undisclosed computer-software vulnerability that hackers can exploit to adversely affect computer programs, data, additional computers or a network.

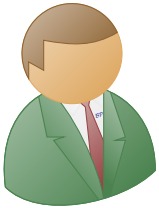
- Who is interested in zero-days?

- Who is interested in zero-days?

Criminals

- Who is interested in zero-days?

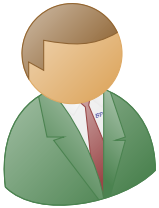
Criminals



Vendors

- Who is interested in zero-days?

Criminals



Vendors

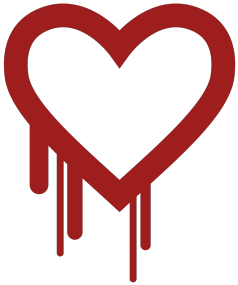
Governments

Zero-Days Prices on the Black Market

Adobe Reader	5000 $ - 30 000 $
Mac OS X	20 000 $ - 50 000 $
Android	30 000 $ - 60 000 $
Flash/Java Browser Plugin	10 000 $ - 100 000 $
Microsoft Word	50 000 $ - 100 000 $
Windows	60 000 $ - 120 000 $
Firefox/Safari	60 000 $ - 150 000 $
Chrome/Internet Explorer	80 000 $ - 200 000 $
iOS	100 000 $ - 250 000 $

Source: Forbes

- A lot of zero-days received fancy names



- A lot of zero-days received fancy names
  - Heartbleed: remote information leakage through OpenSSL

- A lot of zero-days received fancy names
  - Heartbleed: remote information leakage through OpenSSL
  - Shellshock: arbitrary command execution in Bash

- A lot of zero-days received fancy names
  - Heartbleed: remote information leakage through OpenSSL
  - Shellshock: arbitrary command execution in Bash
  - Stagefright: remote code execution on Android through MMS

- A lot of zero-days received fancy names
  - Heartbleed: remote information leakage through OpenSSL
  - Shellshock: arbitrary command execution in Bash
  - Stagefright: remote code execution on Android through MMS
  - Dirty COW: root privileges on Linux

- Jailbreaks (e.g., getting root) on various devices:

- Jailbreaks (e.g., getting root) on various devices:
  - iOS (multiple exploits)

- Jailbreaks (e.g., getting root) on various devices:
  - iOS (multiple exploits)
  - Wii (buffer overflow in *The Legend of Zelda: Twilight Princess*).

- Jailbreaks (e.g., getting root) on various devices:
  - iOS (multiple exploits)
  - Wii (buffer overflow in *The Legend of Zelda: Twilight Princess*).
  - PS2 (buffer overflow in the BIOS)

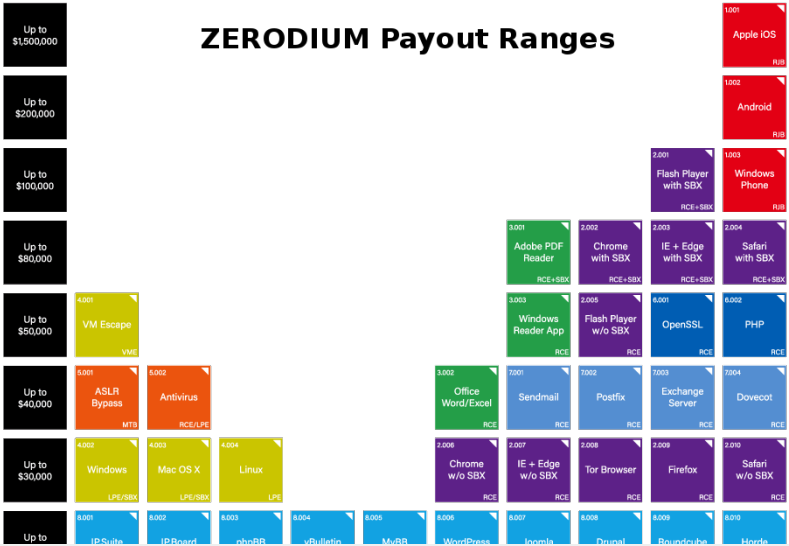
- Jailbreaks (e.g., getting root) on various devices:
  - iOS (multiple exploits)
  - Wii (buffer overflow in *The Legend of Zelda: Twilight Princess*).
  - PS2 (buffer overflow in the BIOS)
  - PS3 (heap overflow)

- Jailbreaks (e.g., getting root) on various devices:
  - iOS (multiple exploits)
  - Wii (buffer overflow in *The Legend of Zelda: Twilight Princess*).
  - PS2 (buffer overflow in the BIOS)
  - PS3 (heap overflow)
  - Xbox (buffer overflow in savegames)

Bug Bounty Programs (Google Chrome)

	High-quality report with functional exploit [1]	High-quality report [2]	Baseline [3]	Low-quality report [4]
Sandbox Escape [5]	$15,000	$10,000	$2,000 - $5,000	$500
Renderer Remote Code Execution	$7,500	$5,000	$1,000 - $3,000	$500
Universal XSS (local bypass or equivalent)	$7,500	$5,000	N/A	N/A
Information Leak	$4,000	$2,000	$0 - $1000	$0
Download Protection bypass [6]	N/A	$1,000	$0 - $500	$0

Severity	Complete Report* + PoC	Payment range (if report includes an exploit leading to Kernel compromise)**	Payment range (if report includes an exploit leading to TEE compromise)**
Critical	Required	Up to $150,000	Up to $200,000
High	Required	Up to $75,000	Up to $100,000
Moderate	Required	Up to $20,000	Up to $35,000
Low	Required	Up to $330	Up to $330


ZERODIUM Payout Ranges
Up to $1,500,000
Up to $200,000
Up to $100,000
Up to $80,000
Up to $50,000
Up to $40,000
Up to $30,000
Up to
1.001 Apple iOS RJB
1.002 Android RJB
2.001 Flash Player with SBX RCE+SBX
1.003 Windows Phone RJB
3.001 Adobe PDF Reader RCE+SBX
2.002 Chrome with SBX RCE+SBX
2.003 IE + Edge with SBX RCE+SBX
2.004 Safari with SBX RCE+SBX
4.001 VM Escape VME
3.003 Windows Reader App RCE
2.005 Flash Player w/o SBX RCE
6.001 OpenSSL RCE
6.002 PHP RCE
5.001 ASLR Bypass MTB
5.002 Antivirus RCE/LPE
3.002 Office Word/Excel RCE
7.001 Sendmail RCE
7.002 Postfix RCE
7.003 Exchange Server RCE
7.004 Dovecot RCE
4.002 Windows LPE/SBX
4.003 Mac OS X LPE/SBX
4.004 Linux LPE
2.006 Chrome w/o SBX RCE
2.007 IE + Edge w/o SBX RCE
2.008 Tor Browser RCE
2.009 Firefox RCE
2.010 Safari w/o SBX RCE
8.001 IP.Suite
8.002 IP.Board
8.003 phpBB
8.004 vBulletin
8.005 MyBB
8.006 WordPress
8.007 Joomla
8.008 Drupal
8.009 Roundcube
8.010 Horde

- Nation-state malware, such as Stuxnet, Duqu, Duqu2, Flame, and Gauss

- Computer and network surveillance

- Computer and network surveillance
- Sometimes use state-sponsored trojan horses (govware)

- Computer and network surveillance
- Sometimes use state-sponsored trojan horses (govware)

- Bundestrojaner (Germany)

- Computer and network surveillance
- Sometimes use state-sponsored trojan horses (govware)

- Bundestrojaner (Germany)
- MiniPanzer and MegaPanzer (Switzerland)

- Computer and network surveillance
- Sometimes use state-sponsored trojan horses (govware)

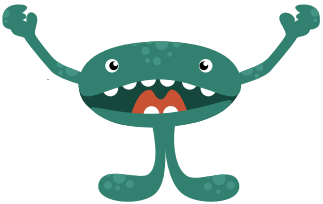
- Bundestrojaner (Germany)
- MiniPanzer and MegaPanzer (Switzerland)
- "Sicherheitspaket" (Austria)

- Computer and network surveillance
- Sometimes use state-sponsored trojan horses (govware)

- Bundestrojaner (Germany)
- MiniPanzer and MegaPanzer (Switzerland)
- "Sicherheitspaket" (Austria)
- NSA Exploits (Shadow Broker Leak)

This Course

Goal of this Course



- We will learn about bugs...

Goal of this Course

- We will learn about bugs...
- ...and even more bugs

Goal of this Course

- We will learn about bugs...
- ...and even more bugs
  - Types of bugs

Goal of this Course

- We will learn about bugs...
- ...and even more bugs
  - Types of bugs
  - How to find them

Goal of this Course

- We will learn about bugs...
- ...and even more bugs
  - Types of bugs
  - How to find them
  - How to exploit them

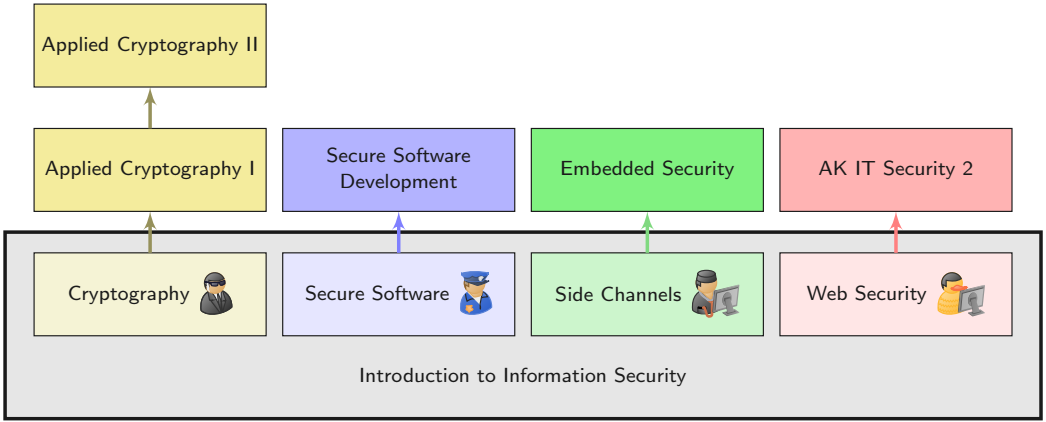
Goal of this Course

- We will learn about bugs...
- ...and even more bugs
  - Types of bugs
  - How to find them
  - How to exploit them
  - How to fix them

Goal of this Course

- We will learn about bugs...
- ...and even more bugs
  - Types of bugs
  - How to find them
  - How to exploit them
  - How to fix them
  - How to prevent them in the first place

SSD in the Curriculum



- Starts on time (12:00)
- Presentation in English
- Always held online
- Questions in English or German

02.10. (Daniel) Introduction

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction

Lecture Schedule

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction + Low Level
16.10. (Martin) Memory Corruption 1
23.10. (Martin) Memory Corruption 2 + Environment

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction + Low Level
16.10. (Martin) Memory Corruption 1
23.10. (Martin) Memory Corruption 2 + Environment
30.10. (Vedad) Exploits
6.11. (Daniel) Finding Bugs 1
13.11. (Vedad) Finding Bugs 2

Lecture Schedule

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction + Low Level
16.10. (Martin) Memory Corruption 1
23.10. (Martin) Memory Corruption 2 + Environment
30.10. (Vedad) Exploits
6.11. (Daniel) Finding Bugs 1
13.11. (Vedad) Finding Bugs 2
20.11. (Samuel) Defensive Programming I

Lecture Schedule

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction + Low Level
16.10. (Martin) Memory Corruption 1
23.10. (Martin) Memory Corruption 2 + Environment
30.10. (Vedad) Exploits
6.11. (Daniel) Finding Bugs 1
13.11. (Vedad) Finding Bugs 2
20.11. (Samuel) Defensive Programming I
27.11. (Samuel) Defensive Programming II

Lecture Schedule

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction + Low Level
16.10. (Martin) Memory Corruption 1
23.10. (Martin) Memory Corruption 2 + Environment
30.10. (Vedad) Exploits
6.11. (Daniel) Finding Bugs 1
13.11. (Vedad) Finding Bugs 2
20.11. (Samuel) Defensive Programming I
27.11. (Samuel) Defensive Programming II
04.12. (Samuel) Defensive Programming III (RUST)

Lecture Schedule

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction + Low Level
16.10. (Martin) Memory Corruption 1
23.10. (Martin) Memory Corruption 2 + Environment
30.10. (Vedad) Exploits
6.11. (Daniel) Finding Bugs 1
13.11. (Vedad) Finding Bugs 2
20.11. (Samuel) Defensive Programming I
27.11. (Samuel) Defensive Programming II
04.12. (Samuel) Defensive Programming III (RUST)

Lecture Schedule

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction + Low Level
16.10. (Martin) Memory Corruption 1
23.10. (Martin) Memory Corruption 2 + Environment
30.10. (Vedad) Exploits
6.11. (Daniel) Finding Bugs 1
13.11. (Vedad) Finding Bugs 2
20.11. (Samuel) Defensive Programming I
27.11. (Samuel) Defensive Programming II
04.12. (Samuel) Defensive Programming III (RUST)

11.12/18.12? Exam

02.10. (Daniel) Introduction + Low Level
09.10. (Daniel) Introduction + Low Level
16.10. (Martin) Memory Corruption 1
23.10. (Martin) Memory Corruption 2 + Environment
30.10. (Vedad) Exploits
6.11. (Daniel) Finding Bugs 1
13.11. (Vedad) Finding Bugs 2
20.11. (Samuel) Defensive Programming I
27.11. (Samuel) Defensive Programming II
04.12. (Samuel) Defensive Programming III (RUST)

11.12/18.12? Exam
29.01. Exam

Exam

- 90 minutes
- Questions in English
- Answers in English or German
- Covers everything from Low Level to Countermeasures

Practicals

Register for the Practicals in TUGRAZonline!

Support Channels

- Register at Discord server using `https://discord.com/invite/rvwAdYb`
- Ask questions regarding the lecture and practicals
- Questions in English or German
- For more detailled questions use dm or email

7.10. Introduction + Warmup Presentation

14.10. H1+H2 release 

21.10. Tutorials + Questions

28.10. ROP Tutorial

4.11. Question hour H1

6.11. Deadline H2 23:59

11.11. D1+D2 release (Question hour H2)

13.11. Deadline H2 23:59

Practicals Schedule II

18.11. Tutorial Defensive I

25.11. Tutorial Defensive II

2.12. Tutorial Defensive III (RUST)

9.12. Question hour D1

11.12. Deadline D1 23:59

16.12. D3 release (Question hour D2)

18.12. Deadline D2 23:59

- https://sasectf.student.iaik.tugraz.at/
- Register an account (**student mail address** recommended)
- Challenges during the lecture
- Solved challenges → bonus points for the written exam

The x86 Architecture

- Backwards-compatibility
- Heavily stack-oriented
  - Function calls require a stack
  - Context switches (interrupts) require a stack
- Complex instruction set with variable-length op codes
- Unaligned memory access and execution

x86-32: Registers (short version)

- General purpose: eax, ebx, ecx, edx, edi, esi

x86-32: Registers (short version)

- General purpose: eax, ebx, ecx, edx, edi, esi
- Stack pointer: esp
- Base pointer: ebp

x86-32: Registers (short version)

- General purpose: eax, ebx, ecx, edx, edi, esi
- Stack pointer: esp
- Base pointer: ebp
- Instruction pointer: eip

x86-32: Registers (short version)

- General purpose: eax, ebx, ecx, edx, edi, esi
- Stack pointer: esp
- Base pointer: ebp
- Instruction pointer: eip
- Control registers: cr3 (cr0-cr15)

x86-32: Registers (short version)

- General purpose: eax, ebx, ecx, edx, edi, esi
- Stack pointer: esp
- Base pointer: ebp
- Instruction pointer: eip
- Control registers: cr3 (cr0-cr15)
- Segment registers: cs, ds, ss, es, fs, gs

x86-64: Registers (short version)

- General purpose: rax, rbx, rcx, rdx, rdi, rsi, r8-r15
- Stack pointer: rsp
- Base pointer: rbp
- Instruction pointer: rip
- Control registers: cr3 (cr0-cr15)
- Segment registers: cs, ds, ss, es, fs, gs

- High-level abstraction for
  - Op codes
  - Addresses
  - Variable storage

Op codes

Op code example


```
#include <stdio.h>
int main()
{
  puts("hello world");
  for (size_t i = 0; i < 10; ++i)
    putchar('0'+i);
  putchar('\n');
  return 0;
}
```

Op code example


```
55                      pushq %rbp
4889e5                  movq  %rsp, %rbp
4883ec10                subq  $0x10, %rsp
bf84304a00              movl  $0x4a3084, %edi
e870f50000              callq 0xf582
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

Op code example

```
55

4889e5                  movq  %rsp, %rbp
4883ec10                subq  $0x10, %rsp
bf84304a00              movl  $0x4a3084, %edi
e870f50000              callq 0xf582
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
55
48
89e5                    movl  %esp, %ebp
4883ec10                subq  $0x10, %rsp
bf84304a00              movl  $0x4a3084, %edi
e870f50000              callq 0xf582
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
55
48
89e5                    movl  %esp, %ebp
4883ec10                subq  $0x10, %rsp
bf84304a00              movl  $0x4a3084, %edi
e870f50000              callq 0xf582
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
55
4889
e548                    inl   $0x48, %eax
83ec10                  subl  $0x10, %esp
bf84304a00              movl  $0x4a3084, %edi
e870f50000              callq 0xf582
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
55
4889e5

4883ec10                    subq  $0x10, %rsp
bf84304a00                  movl  $0x4a3084, %edi
e870f50000                  callq 0xf582
48c745f800000000            movq  $0, -8(%rbp)
eb13                        jmp   0x2f
488b45f8                    movq  -8(%rbp), %rax
83c030                      addl  $0x30, %eax
89c7                        movl  %eax, %edi
e868f70000                  callq 0xf792
488345f801                  addq  $1, -8(%rbp)
48837df809                  cmpq  $9, -8(%rbp)
76e6                        jbe   0x1c
bf0a000000                  movl  $0xa, %edi
e852f70000                  callq 0xf792
b800000000                  movl  $0, %eax
c9                          leave
c3                          retq
```

But what if we misalign?


```
55
4889e5
48
83ec10                  subl   $0x10, %esp
bf84304a00              movl   $0x4a3084, %edi
e870f50000              callq  0xf582
48c745f800000000        movq   $0, -8(%rbp)
eb13                    jmp    0x2f
488b45f8                movq   -8(%rbp), %rax
83c030                  addl   $0x30, %eax
89c7                    movl   %eax, %edi
e868f70000              callq  0xf792
488345f801              addq   $1, -8(%rbp)
48837df809              cmpq   $9, -8(%rbp)
76e6                    jbe    0x1c
bf0a000000              movl   $0xa, %edi
e852f70000              callq  0xf792
b800000000              movl   $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
55
4889e5
4883
ec                  inb   %dx, %al
10bf84304a00        adcb  %bh, 0x4a3084(%rdi)
e870f50000          callq 0xf582
48c745f800000000    movq  $0, -8(%rbp)
eb13                jmp   0x2f
488b45f8            movq  -8(%rbp), %rax
83c030              addl  $0x30, %eax
89c7                movl  %eax, %edi
e868f70000          callq 0xf792
488345f801          addq  $1, -8(%rbp)
48837df809          cmpq  $9, -8(%rbp)
76e6                jbe   0x1c
bf0a000000          movl  $0xa, %edi
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
4889e5
4883ec10
bf
8430                    testb %dh, (%rax)
4a00e8                  addb  %bpl, %al
70f5                    jo    5
0000                    addb  %al, (%rax)
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
```

But what if we misalign?


```
4889e5
4883ec10
bf84
304a00                  xorb  %cl, (%rdx)
e870f50000              callq 0xf582
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
4889e5
4883ec10
bf8430
4a00e8                  addb   %bpl, %al
70f5                    jo     5
0000                    addb   %al, (%rax)
48c745f800000000        movq   $0, -8(%rbp)
eb13                    jmp    0x2f
488b45f8                movq   -8(%rbp), %rax
83c030                  addl   $0x30, %eax
89c7                    movl   %eax, %edi
e868f70000              callq  0xf792
488345f801              addq   $1, -8(%rbp)
48837df809              cmpq   $9, -8(%rbp)
76e6                    jbe    0x1c
bf0a000000              movl   $0xa, %edi
e852f70000              callq  0xf792
b800000000              movl   $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
4883ec10
bf84304a00
e8
70f5                jo    5
0000                addb  %al, (%rax)
48c745f800000000    movq  $0, -8(%rbp)
eb13                jmp   0x2f
488b45f8            movq  -8(%rbp), %rax
83c030              addl  $0x30, %eax
89c7                movl  %eax, %edi
e868f70000          callq 0xf792
488345f801          addq  $1, -8(%rbp)
48837df809          cmpq  $9, -8(%rbp)
76e6                jbe   0x1c
bf0a000000          movl  $0xa, %edi
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
4883ec10
bf84304a00
e870
f5                  cmc
0000                addb  %al, (%rax)
48c745f800000000    movq  $0, -8(%rbp)
eb13                jmp   0x2f
488b45f8            movq  -8(%rbp), %rax
83c030              addl  $0x30, %eax
89c7                movl  %eax, %edi
e868f70000          callq 0xf792
488345f801          addq  $1, -8(%rbp)
48837df809          cmpq  $9, -8(%rbp)
76e6                jbe   0x1c
bf0a000000          movl  $0xa, %edi
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
4883ec10
bf84304a00
e870f5
0000                    addb  %al, (%rax)
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
4883ec10
bf84304a00
e870f500
0048c7          addb  %cl, -0x39(%rax)
45f8            clc
0000            addb  %al, (%rax)
0000            addb  %al, (%rax)
eb13            jmp   0x2f
488b45f8        movq  -8(%rbp), %rax
83c030          addl  $0x30, %eax
89c7            movl  %eax, %edi
e868f70000      callq 0xf792
488345f801      addq  $1, -8(%rbp)
48837df809      cmpq  $9, -8(%rbp)
76e6            jbe   0x1c
bf0a000000      movl  $0xa, %edi
e852f70000      callq 0xf792
b800000000      movl  $0, %eax
c9              leave
c3              retq
```

But what if we misalign?

```
4883ec10
bf84304a00
e870f50000
48c745f800000000        movq  $0, -8(%rbp)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
bf84304a00
e870f50000
48
c745f800000000          movl   $0, -8(%rbp)
eb13                    jmp    0x2f
488b45f8                movq   -8(%rbp), %rax
83c030                  addl   $0x30, %eax
89c7                    movl   %eax, %edi
e868f70000              callq  0xf792
488345f801              addq   $1, -8(%rbp)
48837df809              cmpq   $9, -8(%rbp)
76e6                    jbe    0x1c
bf0a000000              movl   $0xa, %edi
e852f70000              callq  0xf792
b800000000              movl   $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
bf84304a00
e870f50000
48c7
45f8                    clc
0000                    addb  %al, (%rax)
0000                    addb  %al, (%rax)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
bf84304a00
e870f50000
48c745
f8                  clc
0000                addb  %al, (%rax)
0000                addb  %al, (%rax)
eb13                jmp   0x2f
488b45f8            movq  -8(%rbp), %rax
83c030              addl  $0x30, %eax
89c7                movl  %eax, %edi
e868f70000          callq 0xf792
488345f801          addq  $1, -8(%rbp)
48837df809          cmpq  $9, -8(%rbp)
76e6                jbe   0x1c
bf0a000000          movl  $0xa, %edi
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
bf84304a00
e870f50000
48c745f8
0000                    addb  %al, (%rax)
0000                    addb  %al, (%rax)
eb13                    jmp   0x2f
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
bf84304a00
e870f50000
48c745f800
0000          addb  %al, (%rax)
00eb          addb  %ch, %bl
13488b        adcl  -0x75(%rax), %ecx
45f8          clc
83c030        addl  $0x30, %eax
89c7          movl  %eax, %edi
e868f70000    callq 0xf792
488345f801    addq  $1, -8(%rbp)
48837df809    cmpq  $9, -8(%rbp)
76e6          jbe   0x1c
bf0a000000    movl  $0xa, %edi
e852f70000    callq 0xf792
b800000000    movl  $0, %eax
c9            leave
c3            retq
```

But what if we misalign?


```
bf84304a00
e870f50000
48c745f80000
0000                    addb   %al, (%rax)
eb13                    jmp    0x2f
488b45f8                movq   -8(%rbp), %rax
83c030                  addl   $0x30, %eax
89c7                    movl   %eax, %edi
e868f70000              callq  0xf792
488345f801              addq   $1, -8(%rbp)
48837df809              cmpq   $9, -8(%rbp)
76e6                    jbe    0x1c
bf0a000000              movl   $0xa, %edi
e852f70000              callq  0xf792
b800000000              movl   $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
bf84304a00
e870f50000
48c745f8000000
00eb                addb  %ch, %bl
13488b              adcl  -0x75(%rax), %ecx
45f8                clc
83c030              addl  $0x30, %eax
89c7                movl  %eax, %edi
e868f70000          callq 0xf792
488345f801          addq  $1, -8(%rbp)
48837df809          cmpq  $9, -8(%rbp)
76e6                jbe   0x1c
bf0a000000          movl  $0xa, %edi
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
bf84304a00
e870f50000
48c745f800000000
eb13                jmp   0x2f
488b45f8            movq  -8(%rbp), %rax
83c030              addl  $0x30, %eax
89c7                movl  %eax, %edi
e868f70000          callq 0xf792
488345f801          addq  $1, -8(%rbp)
48837df809          cmpq  $9, -8(%rbp)
76e6                jbe   0x1c
bf0a000000          movl  $0xa, %edi
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
e870f50000
48c745f800000000
eb
13488b                  adcl  -0x75(%rax), %ecx
45f8                    clc
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?

```
e870f50000
48c745f800000000
eb13
488b45f8                movq  -8(%rbp), %rax
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
48c745f800000000
eb13
48
8b45f8          movl  -8(%rbp), %eax
83c030          addl  $0x30, %eax
89c7            movl  %eax, %edi
e868f70000      callq 0xf792
488345f801      addq  $1, -8(%rbp)
48837df809      cmpq  $9, -8(%rbp)
76e6            jbe   0x1c
bf0a000000      movl  $0xa, %edi
e852f70000      callq 0xf792
b800000000      movl  $0, %eax
c9              leave
c3              retq
```

But what if we misalign?


```
48c745f800000000
eb13
488b
45f8                    clc
83c030                  addl  $0x30, %eax
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?

```
48c745f800000000
eb13
488b45
f8                      clc
83c030                  addl   $0x30, %eax
89c7                    movl   %eax, %edi
e868f70000              callq  0xf792
488345f801              addq   $1, -8(%rbp)
48837df809              cmpq   $9, -8(%rbp)
76e6                    jbe    0x1c
bf0a000000              movl   $0xa, %edi
e852f70000              callq  0xf792
b800000000              movl   $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?

```
eb13
488b45f8
83
c03089                  salb  $0x89, (%rax)
c7                      invalid opcode
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?

```
eb13
488b45f8
83c0
3089c7e868f7          xorb  %cl, -0x8971739(%rcx)
0000                  addb  %al, (%rax)
488345f801            addq  $1, -8(%rbp)
48837df809            cmpq  $9, -8(%rbp)
76e6                  jbe   0x1c
bf0a000000            movl  $0xa, %edi
e852f70000            callq 0xf792
b800000000            movl  $0, %eax
c9                    leave
c3                    retq
```

But what if we misalign?


```
eb13
488b45f8
83c030
89c7                    movl  %eax, %edi
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
488b45f8
83c030
89
c7                      invalid opcode
e868f70000              callq 0xf792
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
488b45f8
83c030
89c7
e868f70000                callq 0xf792
488345f801                addq  $1, -8(%rbp)
48837df809                cmpq  $9, -8(%rbp)
76e6                      jbe   0x1c
bf0a000000                movl  $0xa, %edi
e852f70000                callq 0xf792
b800000000                movl  $0, %eax
c9                        leave
c3                        retq
```

But what if we misalign?

```
83c030
89c7
e8
68f7000048          pushq  $0x480000f7
8345f801            addl   $1, -8(%rbp)
48837df809          cmpq   $9, -8(%rbp)
76e6                jbe    0x1c
bf0a000000          movl   $0xa, %edi
e852f70000          callq  0xf792
b800000000          movl   $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
83c030
89c7
e868
f70000488345        testl $0x45834800, (%rax)
f8                  clc
014883              addl  %ecx, -0x7d(%rax)
7df8                jge   0x2b
0976e6              orl   %esi, -0x1a(%rsi)
bf0a000000          movl  $0xa, %edi
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?

```
83c030
89c7
e868f7
0000                    addb  %al, (%rax)
488345f801              addq  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
83c030
89c7
e868f700
004883          addb  %cl, -0x7d(%rax)
45f8            clc
014883          addl  %ecx, -0x7d(%rax)
7df8            jge   0x2b
bf0a000000      movl  $0xa, %edi
e852f70000      callq 0xf792
b800000000      movl  $0, %eax
c9              leave
c3              retq
```

But what if we misalign?


```
83c030
89c7
e868f70000
488345f801                    addq  $1, -8(%rbp)
48837df809                    cmpq  $9, -8(%rbp)
76e6                          jbe   0x1c
bf0a000000                    movl  $0xa, %edi
e852f70000                    callq 0xf792
b800000000                    movl  $0, %eax
c9                            leave
c3                            retq
```

But what if we misalign?


```
89c7
e868f70000
48
8345f801                addl  $1, -8(%rbp)
48837df809              cmpq  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
89c7
e868f70000
4883
45f8                    clc
014883                  addl  %ecx, -0x7d(%rax)
7df8                    jge   0x2b
0976e6                  orl   %esi, -0x1a(%rsi)
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
89c7
e868f70000
488345
f8                  clc
014883              addl   %ecx, -0x7d(%rax)
7df8                jge    0x2b
0976e6              orl    %esi, -0x1a(%rsi)
bf0a000000          movl   $0xa, %edi
e852f70000          callq  0xf792
b800000000          movl   $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
89c7
e868f70000
488345f8
014883                  addl   %ecx, -0x7d(%rax)
7df8                    jge    0x2b
0976e6                  orl    %esi, -0x1a(%rsi)
bf0a000000              movl   $0xa, %edi
e852f70000              callq  0xf792
b800000000              movl   $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
89c7
e868f70000
488345f801
48837df809          cmpq  $9, -8(%rbp)
76e6                jbe   0x1c
bf0a000000          movl  $0xa, %edi
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
e868f70000
488345f801
48
837df809                cmpl  $9, -8(%rbp)
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
e868f70000
488345f801
4883
7df8                    jge   0x2b
0976e6                  orl   %esi, -0x1a(%rsi)
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
e868f70000
488345f801
48837d
f8                      clc
0976e6                  orl   %esi, -0x1a(%rsi)
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
e868f70000
488345f801
48837df8
0976e6                  orl   %esi, -0x1a(%rsi)
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
e868f70000
488345f801
48837df809
76e6                    jbe   0x1c
bf0a000000              movl  $0xa, %edi
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
488345f801
48837df809
76
e6bf            outb  %al, $0xbf
0a00            orb   (%rax), %al
0000            addb  %al, (%rax)
e852f70000      callq 0xf792
b800000000      movl  $0, %eax
c9              leave
c3              retq
```

But what if we misalign?


```
488345f801
48837df809
76e6
bf0a000000                    movl  $0xa, %edi
e852f70000                    callq 0xf792
b800000000                    movl  $0, %eax
c9                            leave
c3                            retq
```

But what if we misalign?


```
48837df809
76e6
bf
0a00                    orb   (%rax), %al
0000                    addb  %al, (%rax)
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
48837df809
76e6
bf0a
0000                    addb  %al, (%rax)
00e8                    addb  %ch, %al
52                      pushq %rdx
f70000b80000            testl $0xb800, (%rax)
0000                    addb  %al, (%rax)
c9                      leave
c3                      retq
```

But what if we misalign?

```
48837df809
76e6
bf0a00
0000                    addb  %al, (%rax)
e852f70000              callq 0xf792
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?

```
48837df809
76e6
bf0a0000
00e8                    addb  %ch, %al
52                      pushq %rdx
f70000b80000            testl $0xb800, (%rax)
0000                    addb  %al, (%rax)
c9                      leave
c3                      retq
```

But what if we misalign?


```
48837df809
76e6
bf0a000000
e852f70000          callq 0xf792
b800000000          movl  $0, %eax
c9                  leave
c3                  retq
```

But what if we misalign?


```
76e6
bf0a000000
e8
52                      pushq %rdx
f70000b80000            testl $0xb800, (%rax)
0000                    addb  %al, (%rax)
c9                      leave
c3                      retq
```

But what if we misalign?


```
76e6
bf0a000000
e852f7
0000                    addb  %al, (%rax)
b800000000              movl  $0, %eax
c9                      leave
c3                      retq
```

But what if we misalign?


```
76e6
bf0a000000
e852f700
00b800000000                    addb   %bh, (%rax)
c9                              leave
c3                              retq
```

But what if we misalign?


```
76e6
bf0a000000
e852f70000
b800000000                    movl  $0, %eax
c9                            leave
c3                            retq
```

But what if we misalign?


```
bf0a000000
e852f70000
b8
0000                    addb   %al, (%rax)
0000                    addb   %al, (%rax)
c9                      leave
c3                      retq
```

But what if we misalign?


```
bf0a000000
e852f70000
b800
0000                    addb  %al, (%rax)
00c9                    addb  %cl, %cl
c3                      retq
```

But what if we misalign?


```
bf0a000000
e852f70000
b80000
0000                    addb  %al, (%rax)
c9                      leave
c3                      retq
```

But what if we misalign?


```
bf0a000000
e852f70000
b8000000
00c9                    addb  %cl, %cl
c3                      retq
```

But what if we misalign?


```
bf0a000000
e852f70000
b800000000
c9                              leave
c3                              retq
```