

Secure Software Development

Introduction and Low Level

Daniel Gruss, Vedad Hadzic, Andreas Kogler, Martin Schwarzl, Marcel Nageler

October 1, 2021

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

Introduction

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

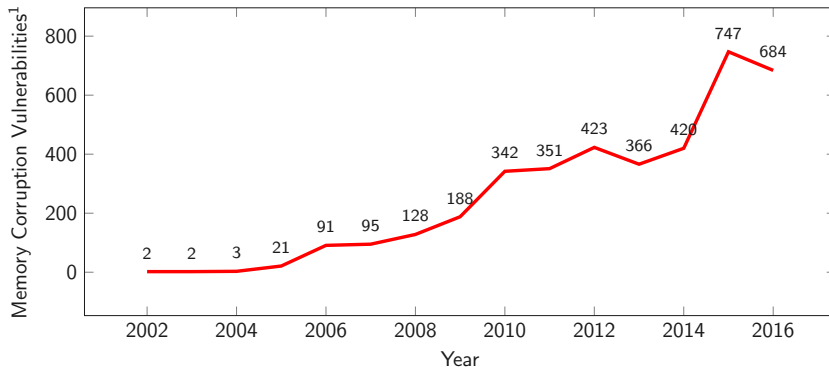
- Many programs are written in native code
- Programs get more and more complex



“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.”

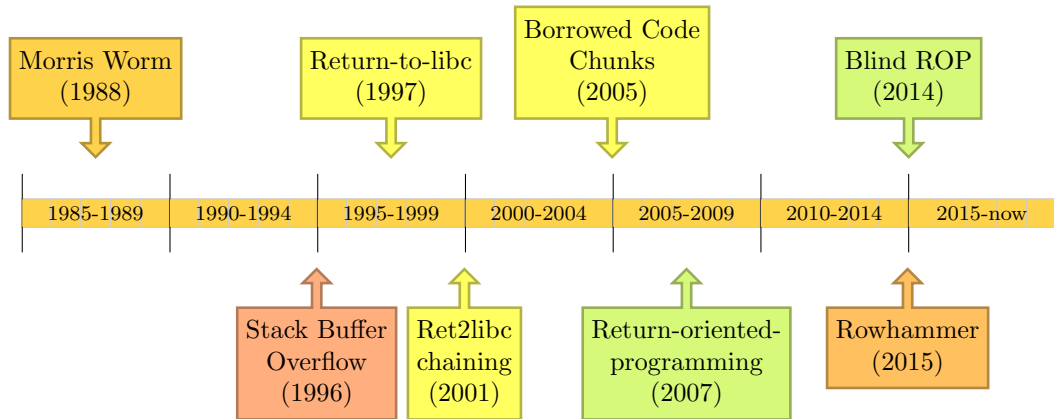
— M. Gosser

The more complex the programs, the more bugs



¹Source: <http://www.cvedetails.com/vulnerabilities-by-types.php>

- We had three decades of memory corruption attacks



IMPACT

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

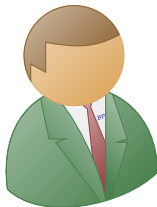


Vendors

- Who is interested in zero-days?



Criminals



Vendors



Governments



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



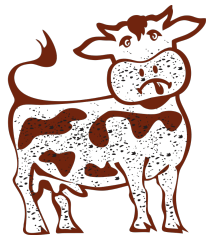
- A lot of zero-days received fancy names
 - Heartbleed: remote information leakage through OpenSSL



- 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



DIRTY COW

- 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)







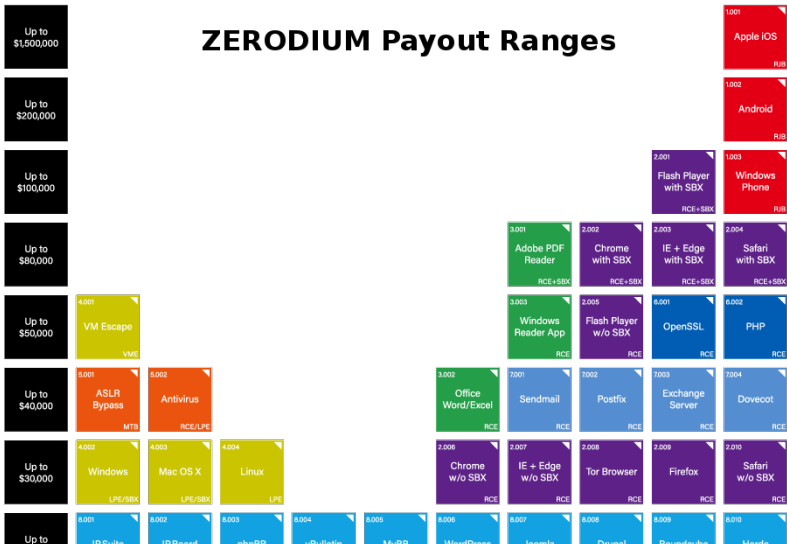
	High-quality report with functional exploit [1]	High-quality report [2]	Baseline [3]	Low-quality report [4]
Sandbox Escape [5]	\$15,000	\$10,000	\$2,000 - \$5,000	\$500
Renderer Remote Code Execution	\$7,500	\$5,000	\$1,000 - \$3,000	\$500
Universal XSS (local bypass or equivalent)	\$7,500	\$5,000	N/A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





- 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)

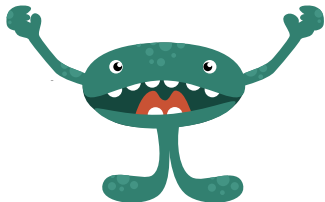


- Computer and network surveillance
- Sometimes use state-sponsored trojan horses (govware)



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

This Course



- We will learn about bugs...



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



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



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



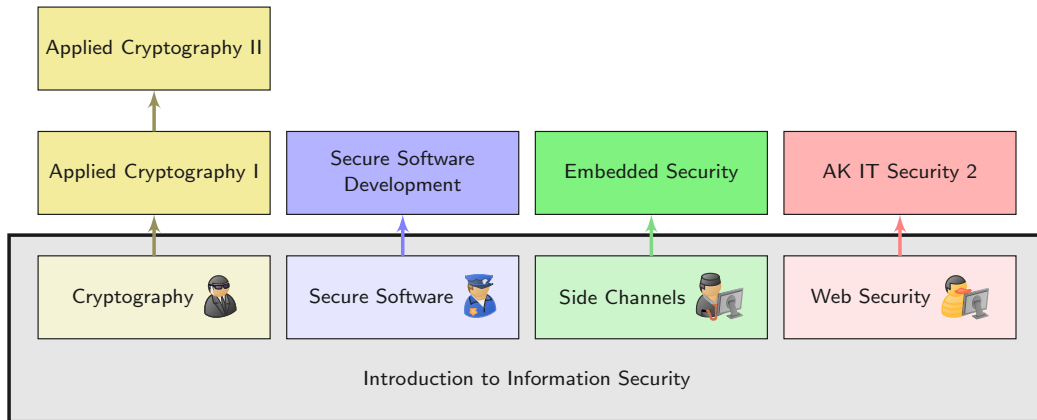
- We will learn about bugs...
- ...and even more bugs
 - Types of bugs
 - How to find them
 - How to exploit them



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



- 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



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

01.10. (Daniel) Introduction

01.10. (Daniel) Introduction + Low Level

09.10. (Daniel) Introduction

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

- 01.10.** (Daniel) Introduction + Low Level
- 09.10.** (Daniel) Introduction + Low Level
- 16.10.** (Martin) Memory Corruption 1
- 23.10.** (Martin) Memory Corruption 2 + Environment
- 29.10.** (Vedad) Exploits
- 5.11.** (Vedad) Finding Bugs 1
- 12.11.** (Vedad) Finding Bugs 2

- 01.10.** (Daniel) Introduction + Low Level
- 09.10.** (Daniel) Introduction + Low Level
- 16.10.** (Martin) Memory Corruption 1
- 23.10.** (Martin) Memory Corruption 2 + Environment
- 29.10.** (Vedad) Exploits
- 5.11.** (Vedad) Finding Bugs 1
- 12.11.** (Vedad) Finding Bugs 2
- 19.11.** (Andreas) Defensive Programming I

- 01.10.** (Daniel) Introduction + Low Level
- 09.10.** (Daniel) Introduction + Low Level
- 16.10.** (Martin) Memory Corruption 1
- 23.10.** (Martin) Memory Corruption 2 + Environment
- 29.10.** (Vedad) Exploits
- 5.11.** (Vedad) Finding Bugs 1
- 12.11.** (Vedad) Finding Bugs 2
- 19.11.** (Andreas) Defensive Programming I
- 26.11.** (Andreas) Defensive Programming II

- 01.10.** (Daniel) Introduction + Low Level
- 09.10.** (Daniel) Introduction + Low Level
- 16.10.** (Martin) Memory Corruption 1
- 23.10.** (Martin) Memory Corruption 2 + Environment
- 29.10.** (Vedad) Exploits
- 5.11.** (Vedad) Finding Bugs 1
- 12.11.** (Vedad) Finding Bugs 2
- 19.11.** (Andreas) Defensive Programming I
- 26.11.** (Andreas) Defensive Programming II
- 03.12.** (Lukas) Defensive Programming III (RUST)

- 01.10.** (Daniel) Introduction + Low Level
- 09.10.** (Daniel) Introduction + Low Level
- 16.10.** (Martin) Memory Corruption 1
- 23.10.** (Martin) Memory Corruption 2 + Environment
- 29.10.** (Vedad) Exploits
- 5.11.** (Vedad) Finding Bugs 1
- 12.11.** (Vedad) Finding Bugs 2
- 19.11.** (Andreas) Defensive Programming I
- 26.11.** (Andreas) Defensive Programming II
- 03.12.** (Lukas) Defensive Programming III (RUST)
- 10.12.** (Pietro) Guest Lecture

- 01.10.** (Daniel) Introduction + Low Level
- 09.10.** (Daniel) Introduction + Low Level
- 16.10.** (Martin) Memory Corruption 1
- 23.10.** (Martin) Memory Corruption 2 + Environment
- 29.10.** (Vedad) Exploits
- 5.11.** (Vedad) Finding Bugs 1
- 12.11.** (Vedad) Finding Bugs 2
- 19.11.** (Andreas) Defensive Programming I
- 26.11.** (Andreas) Defensive Programming II
- 03.12.** (Lukas) Defensive Programming III (RUST)
- 10.12.** (Pietro) Guest Lecture

- 01.10.** (Daniel) Introduction + Low Level
- 09.10.** (Daniel) Introduction + Low Level
- 16.10.** (Martin) Memory Corruption 1
- 23.10.** (Martin) Memory Corruption 2 + Environment
- 29.10.** (Vedad) Exploits
- 5.11.** (Vedad) Finding Bugs 1
- 12.11.** (Vedad) Finding Bugs 2
- 19.11.** (Andreas) Defensive Programming I
- 26.11.** (Andreas) Defensive Programming II
- 03.12.** (Lukas) Defensive Programming III (RUST)
- 10.12.** (Pietro) Guest Lecture

- 17.12** Exam

- 01.10.** (Daniel) Introduction + Low Level
- 09.10.** (Daniel) Introduction + Low Level
- 16.10.** (Martin) Memory Corruption 1
- 23.10.** (Martin) Memory Corruption 2 + Environment
- 29.10.** (Vedad) Exploits
- 5.11.** (Vedad) Finding Bugs 1
- 12.11.** (Vedad) Finding Bugs 2
- 19.11.** (Andreas) Defensive Programming I
- 26.11.** (Andreas) Defensive Programming II
- 03.12.** (Lukas) Defensive Programming III (RUST)
- 10.12.** (Pietro) Guest Lecture

- 17.12** Exam
- 28.01.** Exam

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

Register for the Practicals in TUGRAZonline!

- Register at Discord server using <https://discord.gg/DCpzjqWBD3>
- Ask questions regarding the lecture and practicals
- Questions in English or German
- For more detailed questions use dm or email

6.10. Introduction + Warmup Presentation

13.10. Defenselets I+II release 

20.10. Tutorial

27.10. Tutorial

3.11. Question hour Defenselets

10.11. Defensive programming release / Question hour


12.11. Deadline Defenselets I 23:59

17.11. Defensive Programming Tutorial I

19.11. Deadline Defenselets II 23:59

24.11. Defensive Programming Tutorial II

1.12. Question hour Defensive Programming

15.12. Christmas Special??? 

- We use Mentimeter for Quizzes and FAQs
- We give bonus points for the exam
- Send us your username and prove that you won the quiz via discord/email

- <https://sasectf.student.iaik.tugraz.at/>
- Register an account (**student mail address** recommended)
- Hacklet challenges used in practical part of 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

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

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

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

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

- 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

- 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

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

```
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
```

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
```

```
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
```



```
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
```

```
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
```

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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```



```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```



```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

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
```

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**

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

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

```
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
```

```
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
```

```
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
```

```
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
```

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`

```
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
```



```
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
```

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

```
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
```

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**

```
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
```

```
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
```

```
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
```

```
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
```



```
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
```

```
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
```

```
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
```

```
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
```

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

e868f70000

488345f801

48837df809

76e6

`jbe 0x1c`

bf0a000000

`movl $0xa, %edi`

e852f70000

`callq 0xf792`

b800000000

`movl $0, %eax`

c9

`leave`

c3

`retq`

```
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
```

488345f801

48837df809

76e6

bf0a000000

`movl $0xa, %edi`

e852f70000

`callq 0xf792`

b800000000

`movl $0, %eax`

c9

`leave`

c3

`retq`

48837df809

76e6

bf

0a00 `orb (%rax), %al`

0000 `addb %al, (%rax)`

e852f70000 `callq 0xf792`

b800000000 `movl $0, %eax`

c9 `leave`

c3 `retq`

```
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
```

48837df809

76e6

bf0a00

0000 `addb %al, (%rax)`

e852f70000 `callq 0xf792`

b800000000 `movl $0, %eax`

c9 `leave`

c3 `retq`

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

48837df809

76e6

bf0a000000

e852f70000

`callq 0xf792`

b800000000

`movl $0, %eax`

c9

`leave`

c3

`retq`

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

76e6	
bf0a000000	
e852f7	
0000	<code>addb %al, (%rax)</code>
b800000000	<code>movl \$0, %eax</code>
c9	<code>leave</code>
c3	<code>retq</code>

76e6

bf0a000000

e852f700

00b800000000 `addb %bh, (%rax)`

c9 `leave`

c3 `retq`


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

bf0a000000

e852f70000

b8

0000 `addb %al, (%rax)`

0000 `addb %al, (%rax)`

c9 `leave`

c3 `retq`

bf0a000000

e852f70000

b800

0000

`addb %al, (%rax)`

00c9

`addb %cl, %cl`

c3

`retq`

bf0a000000

e852f70000

b80000

0000

`addb %al, (%rax)`

c9

`leave`

c3

`retq`

bf0a000000

e852f70000

b8000000

00c9 `addb %cl, %cl`

c3 `retq`

bf0a000000

e852f70000

b800000000

c9

leave

c3

retq

