

Binary Exploitation I — Summer 2019

Practical Course

Clemens Jonischkeit

Chair of IT Security / I20
Prof. Dr. Claudia Eckert
Technische Universität München

2020-01-28

What is this?

Exploiting buggy C programs on modern x86_64 Linux systems.

What is this?

Exploiting buggy C programs¹ on modern x86_64 Linux systems.

¹Disclaimer: There might be a little C++ as well...

What is this?

Exploiting buggy C programs¹ on modern x86_64² Linux systems.

¹Disclaimer: There might be a little C++ as well...

²Disclaimer: There might be a little 32-bit x86 as well...

What is this?

Exploiting buggy C programs¹ on modern x86_64² Linux³ systems.

¹Disclaimer: There might be a little C++ as well...

²Disclaimer: There might be a little 32-bit x86 as well...

³Just kidding — no Windows (yet). We kindly refer you to abx.☺

You should...

- ▶ ...understand **how computers work**
- ▶ ...know the basics of the Intel **x86 assembly** language
- ▶ ...have a reasonable grasp of the **C programming** language

...but **most importantly:**

You should...

- ▶ ...understand **how computers work**
- ▶ ...know the basics of the Intel **x86 assembly** language
- ▶ ...have a reasonable grasp of the **C programming** language

...but **most importantly:**

- ▶ ...enjoy **banging your head** against **tough challenges**

Process

Phase I (~ 10 weeks):

- ▶ “Usual” practical course (weekly meetings and assignments)

Phase II (~ 4 weeks):

- ▶ Final project (vulnerable program, exploit and presentation)

Process — Phase I

- ▶ Teams of two
- ▶ Every week: Introduction to a new topic
 - ▶ Submission of solutions **before** the following week's meeting
 - ▶ Private explanation of the solution during that meeting

Process — Phase II

Final project

- ▶ Development of a **vulnerable application**
- ▶ Creation of an **exploit** (ab)using the vulnerability/ies
- ▶ Short paper (about 5 pages)
- ▶ **Presentation** (about 15 minutes)
- ▶ **Hack** the **other teams'** applications 😊
- ▶ Details follow when the time has come

Contents

- ▶ Analysis and debugging tools
- ▶ Hijacking the control flow
- ▶ Shellcode
- ▶ Format string vulnerabilities
- ▶ Stack- and heap-based buffer overflows
- ▶ Exploiting heap management logic
- ▶ Bypassing protection mechanisms

Don't say we didn't warn you

- ▶ Assume up to **30h of workload per week**
- ▶ (But: You reach **state-of-the-art** ~~uber 1337 h4x0r skillz~~ knowledge about binary exploitation techniques on Linux systems)

Time and place

When? Tuesday, 14:00

Where? 01.05.013

Registration

- ▶ Solve our **qualification challenge!**
- ▶ Available at:
`bxqual.sec.in.tum.de:55555`
- ▶ Description <https://www.sec.in.tum.de/i20/teaching/ss2020/binary-exploitation>
- ▶ **Deadline:** 2020-02-17 (23:59 pm)
- ▶ Details: See the course web page after the premeeting
- ▶ Registration using the **matching system** (formally required)
- ▶ **2⁴** slots

▶ Contact me at jonischk@sec.in.tum.de

▶ PGP fingerprint:

▶ A903 76D1 65F3 25F9 8594 280A 2BA0 1592 EFAC B551

- ▶ Contact me at `jonischk@sec.in.tum.de`
- ▶ PGP fingerprint:
 - ▶ A903 76D1 65F3 25F9 8594 280A 2BA0 1592 EFAC B551

Questions?