

# Binary Exploitation I — Winter 2020

## Practical Course

Clemens Jonischkeit & Julian Kirsch

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

2020-07-13

What is this?

Exploiting buggy C programs on modern x86\_64 Linux systems.

# What is this?

Exploiting buggy C programs<sup>1</sup> on modern x86\_64 Linux systems.

---

<sup>1</sup>Disclaimer: There might be a little C++ as well...

# What is this?

Exploiting buggy C programs<sup>1</sup> on modern x86\_64<sup>2</sup> Linux systems.

---

<sup>1</sup>Disclaimer: There might be a little C++ as well...

<sup>2</sup>Disclaimer: There might be a little 32-bit x86 as well...

# What is this?

Exploiting buggy C programs<sup>1</sup> on modern x86\_64<sup>2</sup> Linux<sup>3</sup> systems.

---

<sup>1</sup>Disclaimer: There might be a little C++ as well...

<sup>2</sup>Disclaimer: There might be a little 32-bit x86 as well...

<sup>3</sup>Just 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)

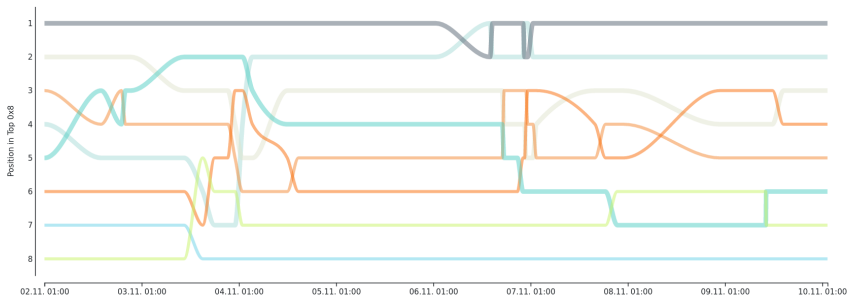
Phase II (~ 4 weeks):

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



Team	pwn00	pwn01	pwn02	pwn03	pwn04	pwn05	pwn06	pwn07	pwn08	pwn09	pwn10	pwn11	pwn12	pwn13	pwn14	pwn15	pwn16	pwn17	pwn18	pwn19	pwn20	pwn21	pwn22	pwn23	pwnr
w0rmi_b0is	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Oxdeadbeef	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
team208	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
# whoami	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
team202	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓
Ox400000	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗
Mantasr0x	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
team203	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗	✓	✓	✗
team209	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
team210	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

## 🔄 Graphs



# 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?** Online

# Registration

- ▶ Solve our **qualification challenge!**
- ▶ Available at:  
`bxqual.sec.in.tum.de:55555`
- ▶ Description <https://www.sec.in.tum.de/i20/teaching/ws2020/binary-exploitation>
- ▶ **Deadline:** 2020-07-26 (23:59 pm)
- ▶ Details: See the course web page after the premeeting
- ▶ Registration using the **matching system** (formally required)
- ▶ **2<sup>4</sup>** 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](mailto:jonischk@sec.in.tum.de)
- ▶ PGP fingerprint:
  - ▶ A903 76D1 65F3 25F9 8594 280A 2BA0 1592 EFAC B551

Questions?