What is this?

Exploiting buggy C programs on modern x86_64 Linux systems.
Exploiting buggy C programs\textsuperscript{1} on modern x86\_64 Linux systems.

\textsuperscript{1}Disclaimer: There might be a little C++ as well...
What is this?

Exploiting buggy C programs\(^1\) on modern x86\(_{64}\)\(^2\) Linux systems.

\(^1\)Disclaimer: There might be a little C++ as well...
\(^2\)Disclaimer: There might be a little 32-bit x86 as well...
What is this?

Exploiting buggy C programs on modern x86_64 Linux systems.

---

1 Disclaimer: There might be a little C++ as well...
2 Disclaimer: There might be a little 32-bit x86 as well...
3 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 | pwn24 |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| team404   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| team203   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| team0xce  |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| team202   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| team205   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| team207   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| team208   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| team209   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| team210   |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |

**Graphs**

- Position in Top 6:

  - Position 1
  - Position 2
  - Position 3
  - Position 4
  - Position 5
  - Position 6

  - Time Range: 08/11/01:00 to 24/01/01
Teams of two

Every week: Introduction to a new topic
  - Submission of solutions before the following week’s meeting
  - Presentation of the solution during that meeting
Final project

- Development of a **vulnerable application**
- Creation of an **exploit** (ab)using the vulnerability/ies
- **Presentation** (about 20 minutes)
- **Hack the other teams’** applications 😊
- Create **Write-Up(s)** about 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? TBA
Registration

- Solve our qualification challenge **individually**!
  - Connect via netcat or in Python via socket module
  - **GDB** might be helpful
  - Dockerfile in case program behaves differently on your system
  - stderr will be helpful for debugging

Available at: courses.sec.in.tum.de:39227

Registration courses.sec.in.tum.de/bx

Deadline: 2024-02-19 (23:59 pm)

Registration using the matching system (formally required)

20 slots - our prioritization is FCFS
Registration

- Solve our qualification challenge individually!
  - Connect via netcat or in Python via socket module
  - GDB might be helpful
  - Dockerfile in case program behaves differently on your system
  - stderr will be helpful for debugging

- Available at:
  
  courses.sec.in.tum.de:39227

- Registration courses.sec.in.tum.de/bx

- **Deadline**: 2024-02-19 (23:59 pm)

- Registration using the matching system (formally required)

- 20 slots - our prioritization is FCFS
Contact me at kilger@sec.in.tum.de
Contact me at kilger@sec.in.tum.de

Questions?