Binary Exploitation — Winter 24/25
Practical Course

Fabian Kilger & Manuel Andreas
Julian Kirsch & Clemens Jonischkeit

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

2024-07-02
Exploiting buggy C programs on modern x86_64 Linux systems.
What is this?

Exploiting buggy C programs\(^1\) on modern x86\_64 Linux systems.

\(^1\)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...
Exploiting buggy C programs\textsuperscript{1} on modern x86\textunderscore 64\textsuperscript{2} Linux\textsuperscript{3} systems.

\textsuperscript{1}Disclaimer: There might be a little C++ as well...
\textsuperscript{2}Disclaimer: There might be a little 32-bit x86 as well...
\textsuperscript{3}Just kidding — no Windows (yet). We kindly refer you to abx.$\smile$
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)
<table>
<thead>
<tr>
<th>Team</th>
<th>pwn10</th>
<th>pwn11</th>
<th>pwn12</th>
<th>pwn13</th>
<th>pwn14</th>
<th>pwn15</th>
<th>pwn16</th>
<th>pwn17</th>
<th>pwn18</th>
<th>pwn19</th>
<th>pwn20</th>
<th>pwn21</th>
<th>pwn22</th>
<th>pwn23</th>
<th>pwn24</th>
</tr>
</thead>
<tbody>
<tr>
<td>team404</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>team203</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>team0xce</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>team202</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>team205</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>team207</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>team208</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>team209</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>team210</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

**Graphs**

![Graphs](image-url)
Process — Phase I

- 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)
<table>
<thead>
<tr>
<th><strong>When?</strong></th>
<th><strong>Wednesday, 14:00</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Where?</strong></td>
<td><strong>TBA</strong></td>
</tr>
</tbody>
</table>
Registration

- Solve our qualification challenge **individually**!
  - Connect via *netcat* or in Python via *socket* module
  - *GDB* might be helpful
  - Dockerfile provided, but not strictly necessary
  - You will **not** need to do any *heap* exploitation
Registration

- Solve our qualification challenge **individually**!
  - Connect via `netcat` or in Python via `socket` module
  - `GDB` might be helpful
  - Dockerfile provided, but not strictly necessary
  - You will **not** need to do any `heap` exploitation

- Available at:
  
  courses.sec.in.tum.de:22019

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

- **Deadline**: 2024-07-16 (23:59 pm)

- Registration using the **matching system** (formally required)

- 26 slots - no further priorization from our side
Contact us at {kilger, andreas}@sec.in.tum.de
Contact us at {kilger, andreas}@sec.in.tum.de

Questions?