Binary Exploitation I — Summer 2019 Practical Course

Clemens Jonischkeit

Chair of IT Security / 120 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...

Exploiting buggy C programs¹ on modern $x86_{64^2}$ 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. ③

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

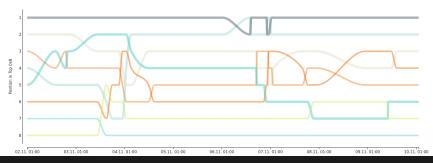
Phase I (\sim 10 weeks):

► "Usual" practical course (weekly meetings and assignments)
Phase II (~ 4 weeks):

► Final project (vulnerable program, exploit and presentation)

Team		punco.	punol	pwm02	pwn03	punoA	punos	punos	pomol	poun08	pwm09	pwn10	pount	punt?	punt3	punia	pount5	pwn16	pountly	ponts.	pwn19	pwn20	pwn21	pwn22	pwn23	pun
w0rmi_b0is		Ø			Ø	V	V		Ø						Ø	Ø					V	V	Ø			V
0xdeadbeef			Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø			Ø	Ø	Ø	Ø			Ø	Ø	Ø	Ø		v
team208		V	Ø	Z	Z	Z	Ø	V	Ø	Z	Z	Z			V	Ø	Ø	Ø			Ø	Ø	V	Ø	V	V
# whoami			Ø	Z	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø			Ø	Ø	Ø	Ø			Ø	Ø	Ø	Ø		V
team202	V	\checkmark	Ø	Z	Ø	Z	Ø	Ø	Ø	Ø	Ø	Z			Ø	Ø	Ø	Ø	\checkmark		Ø	Ø	\checkmark	Ø		V
0x400000			Ø	Z	Ø	Ø	Ø	V	Ø	Ø	Ø	Ø	V		V	Ø	Ø	\swarrow		Ø	Ø	Ø	V	Ø	V	×
Mantasr0x		\checkmark	\swarrow	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø			Ø	Ø	Ø	Ø			Ø	Ø	Ø	Ø		×
team203	V	V	Ø	Z	Ø	Ø	×	×	×	Ø	Ø	Ø	V	V	V	Ø	Ø	×	\mathbf{X}	Ø	Ø	Ø	×	V	V	×
team209	\boxtimes	×	×	\mathbf{X}	\boxtimes	\boxtimes	×	×	×	\mathbf{X}	\mathbf{X}	×	×	×	×	\mathbf{X}	×	×	×	×	\boxtimes	×	×	×	\boxtimes	×
team210	\mathbf{X}	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

🔀 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

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?