SOFTWARE SECURITY 1

TOOLING AND SETUP

HG SOFTWARE SECURITY

EXERCISE SESSIONS

- Thursday 10–12 in MC 5/222
  - That's the CIP pool on the north side of the building
  - Take the **north elevator** to the fifth floor
  - Ring the bell if the door is closed
- We'll discuss the assignments and your questions there
  - Optional, but please make use of these sessions
  - For now: **Bring a laptop** until the computers are set up
- Possible backup slots: Thursday 12–14 or 14–16 (see Moodle)

ASSIGNMENTS

- Assignments consist of CTF-style challenges
  - Each assignment is a small task designed to test specific topics discussed in the lecture
  - For solving them, you get a **flag**
  - Each flag you submit gives you points on our scoreboard
- A flag looks like this:

```
softsec{tGh-P8c_rNlXzICp8kF1HdRvXCBsL57vG8xNc2veJdwelOR6ys91xxOx68Yi9EyM}
```

ASSIGNMENTS

- Assignments are not directly part of your final grade (see lecture slides)
- However, you need to be able to solve them
  - The exam tasks will be similar in style to the assignments, and cover the same topics
  - Of course, they'll be appropriately scaled in difficulty
- Please upload your commented exploits, even if they don't fully work
  - This lets us know whether we need to go into more depth on some topics
  - If there are unexpected solutions, we can also discuss those (and fix them)
  - Also may be needed in case there is a bonus for first/second/third solves

ASSIGNMENTS

- You can find the tasks at `scoreboard.ws24.softsec.rub.de`

- I will do my best to give out a small prize for the best participants on the scoreboard at the end of the semester (details are still unclear)

CHALLENGE SETUP

- All of our challenges use Docker for easier setup
  - macOS users: Use Docker for Mac directly (Colima had issues last year)
  - Apple Silicon users: You'll need `--platform linux/amd64`
- We can only provide limited support for non-{Linux, x64} setups
  - Please help each other and let us know how you fixed problems
- Commands to run each challenge are included in the `Dockerfile`

TOOLING

- Reverse engineering and binary analysis
    - **IDA Pro** ($$$), **Binary Ninja** ($), or **Ghidra** to decompile binaries
        - IDA Free with the cloud decompiler should be sufficient for this course
        - Watch this spot (it might be worth getting acquainted with IDA)
    - **objdump** if you just need a quick peek at the disassembly
- Debugging and dynamic analysis
    - We recommend **GDB** with either **pwndbg** or **gef**
    - **strace** to look at system calls
- Exploit writing
    - **Python** is highly recommended
    - **pwntools** is not required, but very helpful
- Lots of specialized tooling exists for specific tasks (e. g. ROP chains)
- We'll mention those when we get to those topics

SHELLCODE TOOLING

- **pwntools**
  - **shellcraft** can generate shellcode for you (but try getting used to writing your own)
  - `pwn.asm(" ... ")` to assemble instructions directly
  - Make sure you set it to 64-bit mode: `pwn.context.arch = 'amd64'`
- You can use a dedicated assembler (**gas**, **nasm**, **masm**, **yasm**, ...)
- **gcc** or **clang** will treat `.s` files as assembly code, just extract the shellcode with **objcopy**:

```
gcc -Wl,-N -ffreestanding -nostdlib -static code.s -o code.elf
objcopy -j .text -O binary code.elf code.bin
od -tx1 -An code.bin | paste -sd' ' | tr -d ' '
```

SHELLCODE CAVEATS

mov rax, qword ptr [rsi + rcx * 8]
movq (%rsi, %rcx, 8), %rax
YOU
imgflip.com

SHELLCODE CAVEATS

	Intel syntax	AT&T syntax
Registers	`rax, rsi, rip, …`	`%rax, %rsi, %rip, …`
Integers	`42, 0×10, …`	`$42, $0×10, …`
Indexing	`[rdi + rcx * 4]`	`(%rdi, %rcx, 4)`
	`[rsi + 16]`	`16(%rsi)`
Order	`cmp r8, r9`	`cmpq %r9, %r8`
	`jl target` (if r8 < r9)	`jl target` (if r8 < r9)
Width	`mov rsi, qword ptr [rax]`	`movq (%rax), %rsi`
	`add edi, ecx`	`addd %ecx, %edi`
Indirection	`jmp rax`	`jmp *%rax`

SHELLCODE CAVEATS

- For GCC/GAS, use `.intel_syntax noprefix` to switch to Intel syntax
- For `objdump`, use `-M intel`
- Most other tools should use Intel/MASM syntax by default