Lab 5
Main Goals

• **Analyze** potential return-to-libc vulnerabilities in source code

• **Exploit** these vulnerabilities in different scenarios

• **Gain** a deeper understanding of the function call convention
Task 1: Inspect the Program

• Analyze the provided source code
• Determine the potential ret2libc vulnerability
• Understand the stack layout during a function call
Task 2: Using system function

Four subtasks

• Subtask 1:
  • Program should not exit gracefully
  • /bin/sh is an environment variable
  • ASLR is disabled

• Subtask 2:
  • Program **should exit** gracefully
  • /bin/sh is an environment variable
  • ASLR is disabled
Task 2: Using system function

Four subtasks [Cont’d]

• Subtask 3:
  • Program **should exit** gracefully
  • `/bin/sh **is not** an environment variable
  • ASLR is disabled

• Subtask 4:
  • Program **should exit** gracefully
  • `/bin/sh **is not** an environment variable
  • ASLR is **enabled**

**Note:** gdb disables randomization
Task 3: Using `exec` function

- Implement this chain: `printf` → `exec` → `exit`
- With proper inputs and return addresses!

Recall `execve`:

```
int execve(char *file, char *argv[], char *env[])
```

`execl`:

```
int execl(char *file, const char *arg, ...)
```

Calling `execl`:

```
execl("/bin/sh", "/bin/sh", NULL);
```
Task 3: Using `exec1` function

- `fmt_str = "\%5$n"`
- `/bin/sh`
- ret will cause `FUNC` to be called
- `%n` needs an address
Task 3: Using `exec1` function

```c
printf

pop ebp
ret

exec1()

exit()

fmt_str = "%5$n"

"/bin/sh"

"/bin/sh"

0xbfffffaaa

0xbfffffd5d

0xbfffffd5d

0xbfffffe0b0

0xb7da4bac

0xb7da4bff

0xbfffffaaa

0xb7da4da0

0xb7d989d0

0xb7da4dfe

ret will cause FUNC to be called

0xb7da4da0

0xb7d989d0

0xbfffe0b0
```
Task 3: Using `exec1` function

```
0xbffffaaa
fmt_str = "%5$n"

0xbfffd5d
"/bin/sh"

0xbfffd5d
"/bin/sh"

0xbffffe0b0

ret will cause FUNC to be called

0xb7da4bac
printf

0xb7da4dff
pop ebp
ret

0xb7da4da0
exec1()

0xb7d989d0
exit()

0xb7da4bac

0xb7da4b0
0xb7da4b0
0xbfffffaa
0xb7da4da0
0xb7d989d0
0xbfffffd5d
0xbfffffd5d
0xbfffffd5d
0xbffffe0b0

address of this location
```
Questions?