Isfahan University of Technology
Network Security
Dr. Manshaei
Homework 5: Buffer Overflow
Deadline: Friday 22^{nd} **Dey, at** 23 : 59

We only accept the homework **delivered via Yekta (yekta.iut.ac.ir), before the deadline**. If you have any question or concerns about this homework, feel free to contact Mr. Bagher Mohammadzadeh via Telegram: *@BGH*1998 (Preferred) or email: b.mohammadzadeh@ec.iut.ac.ir.

Theoretical Questions

Q1: What are the main causes of Buffer overflow vulnerabilities?

Q2: How can you prevent buffer overflow vulnerabilities in your applications? What methods are recommended in compilers? What is the best way to avoid vulnerability (Explain your reasons)?

Q3: Does Java have a buffer overflow vulnerability?

Q4- Bonus Question: Suppose this code is safeguarded against buffer overflows using the "baggy bound" method, which involves checking 16-byte chunks (as described in the article by Akritidis et al.) You can find the paper in Yekta and a short video at: (Video).
For each pair of values for A and B provided below, assess whether the checks implemented by this method lead to the termination of the program. If termination occurs, identify the specific line of code where the program ends.

```
1. char *p1 = malloc (90)
2. char *p2 = p1 + A;
3. char *p3 = p2 - B;
4. *p3 = '\ 0 ';
```

Figure 1: code

- A = 95 , B = -10
- A = -10 , B = 95
- A = 130 , B = -5
- A = -5 , B = 130
- A = 140 , B = -15
- A = 135 , B = -5

Practical Questions

Perform the following steps on a 32-bit Kali Linux system and prepare a written report detailing each step. Additionally, record the entire process as a video.

- Write code in the C++ language, define a buffer, and store the input data in memory.
- Store an entry in memory with a size smaller than the defined buffer value. Evaluate the current state of the stack and determine the values stored in the stack registers.
- Store an entry in memory with a size exceeding the defined buffer value. Evaluate the current state of the stack and determine the values stored in the stack registers.
- Perform a shellcode injection attack, as we already addressed in the videos. As a result of the attack, the terminal would be ready to get new commands.You need to show this in the captured video.
- Follow the previous steps on 64-bit Kali. Was the attack unsuccessful? If yes, what were the reasons for its failure?