SEOULTECH
Seoul National University of Science & Technology

Assignments for ITM 424

1. Overview

Students will have two project assignments and three HW assignments.

2. Project Assignments

➢ Description

This is a team project. Basically, each team consists of two or three students.

We have two topics and have the presentation for each topic. For each topic, teams are formatted randomly by the e-class system.

Present your results. Each presentation should be done within 5 minutes

➢ Grading

The total scores assigned for the projects are 20 points. That is, each project has 10 points.

Basically, the evaluations are conducted by the teams. However, individual contributions on each team could be considered as well.

➢ Evaluation criteria

Refer to the check list for each project. According to each point in the check list, the score will be quantitatively measured

➢ Feedback strategy

The feedbacks will be provided through e-class system

SEOULTECH
Seoul National University of Science & Technology

➢ 2nd Project (ROPME)

- Refer to the following source code.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>

void func(){
        char overflowme[32];
        read(0, overflowme, 0x200);
}

int main(int argc, char* argv[]){
        setvbuf(stdout, 0, _IOLBF, 0);
        setvbuf(stdin, 0, _IOLBF, 0);
        printf("The address of setvbuf : %16p\n",
setvbuf);
        func();
        write(1, "DONE\n", 5);
        return 0;
}
```

- Guideline
  - Executable file and libc file will be provided. (`ropme` and `libc.so.6`)
  - Using stack based buffer overflow, build ROP chain payload to establish a remote shell connection.and exit normally.

- Check list
  - Explain what is the vulnerability and how to fix it.
  - Explain how to find the offset of return address
  - Explain how to find the libc base address
  - Explain how to find the gadgets and how your gadget chains work.
  - The gadgets need to obtain remote shell access.
  - After a buffer overflow attack, the program should terminate gracefully without crashing.

- **Be aware of cheating! The cheating will be detected based on the submitted source codes.**

- Presentation and submission
  - Presentation: 6 Dec. 2024 at 02:00 PM @ Frontier 511
  - Submit the presentation slide and supplementary materials (including source codes) before the presentation; the results will be verified after the presentation
  - Late submissions are not allowed