CS6570 - Secure Systems Engineering: Assignment-3

Submission guidelines

- **Deadline: 14th March 2024**
- We expect you to submit an arhive (tar or zip) that contains the following
    - The archive should be named as your Roll Number (`<roll_no>.tar`)
    - Files `payload_Q#` that contain the exploit strings you have crafted, and any scripts that were used to genarate the exploit strings.
    - The provided binary (unmodified)
    - A PDF report (preferably in LaTeX) that should contain the following things compulsarily:
        - Your Name and Roll-Number.
        - The explanation of the ROPchains crafted for each question
        - The gadgets you have used
        - Pictures of your working exploits

Files provided

- main
- This README

Description

- The provided binary simply prints the following and waits for input

```
> ./main
This program ONLY adds 21 to itself
21 + 21 = 42
Anything to say?
```

- Upon normal execution, the binary ONLY adds 21 to itself.
- Can we make it do something else?
- There are 3 tasks given to you based on this binary.

Tasks

1. The programmer intended to multiply 73 and 21 instead. Can you make the program compute 73 x 21 ? (40 points)
2. Taking it a step further, we would like to make the binary compute a factorial. Create a payload to compute 7! (60 points)
3. Bonus: Using your experience in crafting ROPchains, can you come up with a way to calculate the nth number in the fibonacci sequence? (**Extra** 20 points)

Testing

- Ensure that your exploit string is self-contained in `payload_Q#`, any additional steps or modifications are not allowed.
- Your input will be passed to the program in the following manner:

```
> cat payload_Q1 - | ./main >&1
```

General Guidelines

- Atempting the bonus task is optional and will not affect the maximum grade of this assignment
- In the bonus task, you are expected to read the value for n from `stdin`
- You are expected to write ROPchains to compute the required expressions. Solutions such as simply printing 1533 in Q1 would not be accepted
- ROPgadget is a tool installed on the VM to identify gadgets that you can use in your ROPchain
- Please ensure that your `payload_Q#` works on the course VM.
- You can write scripts to generate `payload` , include these in your archive.
- The internet is your friend, there a lot of excellent resources available (please properly reference any extra tools/repositories used).