CS6570 - Secure Systems Engineering: Assignment-2

Submission guidelines

- **Deadline: 22nd February 2024**
- We expect you to submit an arhive (tar or zip) that contains the following
    - The arhive should be named as your Roll Number (`<roll_no>.tar`)
    - A file `payload` that contains the exploit string you have crafted
    - Any scripts that were used to genarate the exploit string, commented and formatted.
    - The provided binary (unmodified)
    - A PDF report (preferably in LaTeX) that should contain the following things compulsarily:
        - Your Name and Roll-Number.
        - Are there vulnerabilities present in the provided code? If yes, then why do they exist? How can they be fixed?
        - How do the ***gcc*** flags (in Makefile) affect how "secure" the binary is?

Files provided

- Makefile
- assignment_2.c
- assignment_2 (binary) [`sha256sum:12407ba5f5dc90974e008d8e2b46aaa9d1a38e3a3c95e5d634925fb712551542`]
- This README

Description

- The provided binary expects a command-line argument.

```
❯ ./assignment_2
Usage:
./assignment_2 your_name
```

- A normal execution of the program would list the files in your working directory followed by text a prompt.

```
❯ ./assignment_2 your_name
assignment_2  assignment_2.c  Makefile
Hi your_name!, can you make me run /bin/sh ?
```

- Your goal in this assignemnt is to identify such an input `payload` , which when provided to the binary as an input executes a shell (`/bin/sh`)

```
❯ ./assignment_2 $(cat payload)
assignment_2  assignment_2.c  Makefile  payload
Hi <secret_sauce>!, can you make me run /bin/sh ?
$ whoami
sse
$
```

Testing

- Ensure that your exploit string is self-contained in `payload`, any additional steps or modifications are not allowed.
- Changes of any form to the provided binary are not allowed, the exploit string `payload` should work with the provided binary.
- Your submission would be tested in the following way `./assignment_2 $(cat payload)`, ensure you can run it on your end.
- Submissions of the form `'python -c "secret_sauce"'` will not be accepted, write your exploit to `payload` file.

General Guidelines

- Please ensure that your observations and exploit string `payload` work on the course VM.
- Using *GDB* is suffecient for solving this assignment, you are also free to use other tools if you wish.
- You can write scripts to generate `payload`
- The internet is your friend, there a lot of excellent resources available (please properly reference any extra tools/ repositories used).