CS 260 02: Information Security Lab

Chengyu Song

Slides based on CS6265 taught by
Prof. Taesoo Kim
Today’s Topics

• Quick introduction to GDB
• In-class tutorial
  • Walk over x86 asm and tools
  • Be familiarized with GDB and x86 (32-bit)
  • Let’s crack crackme0x00 ~ crackme0x06
Preparation

• Step 1: Setup environment (Wednesday)

• Step 2: Tutorial (in VM)

  $ cd /vagrant/cs26002/
  $ git pull
  $ cd lab/lab01/tut
  $ cat README.md
References

- GDB tutorial
- x86 instructions
- x86 architecture
GDB Summary

- run/continue
- backtrace/examine
- break/tbreak/rbreak/delete
- stepi/nexti/advance/finish
- watch/rwatch/awatch
- info reg/proc/break
- python, .gdbinit
- etc.
Bomblab

• This week's challenge: Bomblab!

$ cd /vagrant/cs26002/
$ git pull
$ cd lab/lab01/
$ ls
$ bomb bomblab1_01 bomblab1_02 bomblab1_03 bomblab1_04 bomblab1_05 bomblab1_06 bomblab1_07 bomblab1_08 bomblab1_09 bomblab1_10 init.sh README
$ cat README
Welcome to my fiendish little bomb. You have $N$ phases with which to blow yourself up. See you alive!
(hint: security question)
Note on Explosion
Note on Flag

- Random looking bytes, but be careful. It is designed to include tons of information unique to you, so we can easily check plagiarism

```
$ cat /proc/flag
```
Note on Submission

$ ./bin/submit -h
Usage: submit [options]

Options:
- -h, --help show this help message and exit
  -c, --checkin checkin
  -l LAB, --lab=LAB lab
  -p PROBLEM, --problem=PROBLEM problem
  -f, --flag submit flag
  -e, --exploit exploit

$ ./bin/submit -l lab01 -p bomblab1_01 -f saved_flag
$ ./bin/submit -l lab01 -p bomblab1_01 -e writeup