CS6265: Information Security Lab

Taesoo Kim

CS6265: Information Security Lab

- A special course: supervised, hands-on laboratory
- Designed for seniors and above
- Prerequisite: OS, system programming, architecture
- Background: low-level programming (e.g., C)

Course Goal: Capture-the-flag

Topics

- Reverse engineering
- Binary exploitation
- Binary analysis
- Memory forensic
- etc.

Schedule: https://tc.gtisc.gatech.edu/cs6265/2016/cal.html

Weekly Structure

- Fri: Cover single topic (e.g., stack overflow)
- Wed: Optional recitation at 5-6pm
- Thr: Solve 10 challenge binaries (i.e., problem set)
    - Submit: flag, writeup/exploit of each challenge

Course Grading

- 100% Lab (if you didn't turn in a single lab, you will get F)
- No midterm and final exams
- 40%: A , 30-40%: B , 30-20%: C and below
- But, grading in each group: undergraduates, Masters and PhDs
- See Game Rules

Scoring Each Lab (Game Rules)

- Approximately **10 challenges**
- 20 pt (flag) x 1.0 (write-up/exploit) = 20 pt (each challenge)
- So, 200 pt (20 pt x 10 challenges) in each lab
- Bonus : two fastest solvers get 10 and 5 bonus pt
- Late policy : 50% of the original points (an extra week)

Misconduct Policy

- Cheating vs. collaboration
- Refer GT's Academic Misconduct Policy
- **Never ever** use/copy other students' code/write-up
- Please write down names of your collaborators

In-class Meeting

- 30 min : discus last week's challenges (be prepared, two fastest solvers)
- 30 min : cover this week's topic
- 30-60 min : in-class tutorial (so bring a laptop!)
- 30-60 min : TA-ing

NOTE . recitation on Wed at 5-6pm (Klaus 1447)

Administrivia

- TA: Insu Yun, Wen Xu, Ren Ding, Yeongjin Jang
- Contact: 6265-staff@cc.gatech.edu
- Website: https://tc.gtisc.gatech.edu/cs6265/2016/index.html
- Scoreboard: https://tc.gtisc.gatech.edu/cs6265/2016/submit/handin.py
- Join Piazza
  - Be active: bonus points on two participants

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
CB25682B33EF8BF23545A767562A1D5AA33C88EEACC1AE562D950CB9F1E5725D
864725DB51460902ECBD52BA4CBED86A10F3A98A35F6FB71871019702A0E9199
5BC59332C390A3C27D0EC2CE85BC13E956A6027E3171352F90467A8C12346D9A
2A26EE914B3078ED031FDB14BB6224C3D743D79A733FB49EB4E9C1F383CF810E
F6841EE935FE2DA2C57DB4804B6823884B36AE62B08848486918C120E4C2AA94
E1D3F8A6E9E2251AC39E5F37971FB07DFF839E0BC1C4E6C1D4A24E0948F8751B
25BFFE854CD84A8D8E28814398FF192CD9AD37150D83DA872E944DF1552F97DD
...
```

Next Two Weeks

Monday	Tuesday	Wednesday	Thursday	Friday
Aug 22 *First day of class (**No class**)*	Aug 23	Aug 24	Aug 25	Aug 26 **LEC 1:** Warm-up: x86, Tools **TUT 1:** GDB **Assigned:** Lab 01: Bomb Lab1 *ADD/DROP DATE*
Aug 29	Aug 30	Aug 31	Sep 1 **DUE:** Lab 01	Sep 2 **LEC 2:** Warm-up: x86_64, Shellcode, Tools **TUT 2:** PEDA, Shellcode **Assigned:** Lab 02: Bomb Lab2 / Shellcode
Sep 5 Labor day	Sep 6	Sep 7	Sep 8 **DUE:** Lab 02	Sep 9 **Assigned:** Lab 03: Stack Overflow

Today's Topics

- This week: Bomblab !
- 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

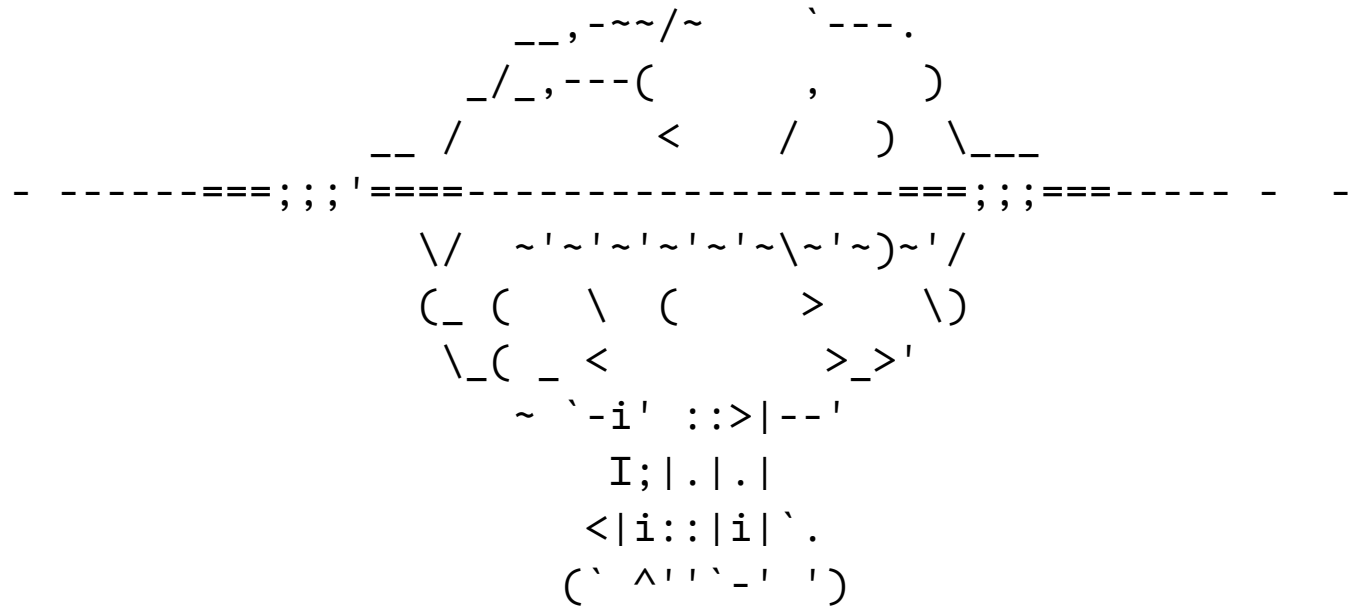
Note on Bomblab

```
$ ./bomb

          ,--.!,   ____                    _     _        _
       __/   -*-  | __ )  ___  _ __ ___ | |__ | | __ _| |__
     ,d08b.  '|`  |  _ \ / _ \| '_ ` _ \| '_ \| |/ _` | '_ \
    0088MM        | |_) | (_) | | | | | | |_) | | (_| | |_) |
     `9MMP'       |____/ \___/|_| |_| |_|_.__/|_|\__,_|_.__/
           cs6265

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



DEMO: GDB Summary

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

In-class Tutorial

- Step 1: Setup environment
  - https://tc.gtisc.gatech.edu/cs6265/2016/rules.html
- Step 2: Tutorial (in VM)

```
$ git git@clone tc.gtisc.gatech.edu:seclab-pub cs6265
$ cd cs6265/lab01
$ cat README
$ cd tut
$ cat README
```

References

- GDB tutorial
- x86 instructions
- x86 architecture