Information Security Laboratory

01-Introduction

Sang Kil Cha
Instructor: Sang Kil Cha

• Researcher, Engineer, and Hacker

• Office: N5 2319
• Office hour: by appointment
KaisHack

• KAIST Hacking

• Post-undergraduate hacking team

• Jointly work with GoN (undergraduate hacking team)
  - http://kaishackgon.blogspot.com
Information Security
What are the virtues of being a good security researcher?
Several Necessary Conditions

• Being ethical

• Being able to think like adversaries
Great Engineer vs. Security Expert

• Great engineers can easily become a great security researcher

• But, many security experts (hackers) that I know are poor engineers
Engineering

Math

Science

Social knowledge

Policy
The empire state building

- Year: 1931
- Cost: $41 million
- Completed in 16 months
- Engineer: Homer G. Balcom
Lockheed SR-71 Blackbird

- Year: 1964
- Speed: 3529.6 km/h
- Engineer: Kelly Johnson
Less Successful Example

RMS Titanic (1912)
Touted as an “unsinkable ship”
Breaking things is far easier than engineering things
Vision of this Course

Learn foundations of computer security through hands-on engineering exercises
Is this Course about S/W Exploitation?

• We do learn hacking, but we focus more on the engineering aspects in this course

• We are not going to study assembly code nor binary-based exploitation in this course

• Refer to *IS-561*: Binary Code Analysis and Secure Software Systems
How to Become a Good Engineer?

Practice (read and write)
Read The Manual
Read The Manual (a.k.a. RTFM)

Hackles

…I tried compiling a new driver for my network card, but I’m still getting errors –

New Sys Admin

By Drake Emko & Jen Brodzik

http://hackles.org

Copyright © 2001 Drake Emko & Jen Brodzik

Image from http://hackles.org/cgi-bin/archives.pl?request=2
BYOL Course

• Bring Your Own Laptop
  − You can use desktops in the lab, but it is going to be easier for you to use your own laptop

• We do *hands-on activities* everyday
Open-Sourced Course

Nearly everything will be open-sourced!

https://github.com/KAIST-IS521

For more information about Github-based education, see:
The Emergence of GitHub as a Collaborative Platform for Education, 
CSCW 2015
Open-Sourced Course (cont’d)

• Every student works in a public repository (@ GitHub)
• Everyone can see each other’s work
  – Concern about cheating/plagiarism?
  – You can explicitly fork off from a peer’s branch
  – Everyone (in the world) can see who is cheating!
• Clearly show your progress with frequent commits
Course Logistics

• (Roughly) 1 Lecture + 1 Lab
  − Lecture on Mon. (Prof.): 14:30 ~ 16:15
  − Lab on Wed. (TA): 14:30 ~ 17:10

• Bring your laptop to all sessions
  − We will do hands on activities even in a lecture
Course Web

• Syllabus and slides
  @ http://softsec.kaist.ac.kr/courses/2017s-is521/

• Assignment and activities
  @ https://github.com/KAIST-IS521/
Course Logistics

• 70% Individual
  – Offline (class) + Online activity
  – Homework and exam

• 30% Team project
Difficulty Level

• If you are an average CS student, you will not be able to finish all the activities during the course, but you should be able to finish them by the deadline if you work hard enough 😊

• If you are a fast coder, then you should be able to finish everything during the lecture

• If you are a good engineer, then you can finish writing your code in the lecture, but you will revise your code for later use
What if I am a theoretician?

Course staffs will do their best, but hey, this is a hands-on course! 😊
What if I am done with all the activities?

- Please remain seated
- Help other peers (we will give bonus points)
- Do you own stuff if you want, but do not distract others
Activity-based Scoring System

Github records the history of all your activities!

• Help peers (commenting on Github): +1
• Help revising course materials
  (accepted pull request on the main repo): +5
• Share problems with peers (issue on Github): +1
• In-class extra points: +1 ~ +5
Grading Activity Score

• Relative evaluation
  – Sort students based on their total activity score, and divide them into 9 groups: A+/A0/A-/B+/B0/B-/C+/C0/C-

• Be active!
Grading Each Activity (Assignment)

• Each commit should convey clear idea (why you made this change)
  – You should *make multiple commits* instead of making a single monolithic commit

• Each assignment should contain a document (README.md) completely explaining what you did

• Your code should be clearly commented (make them concise)
Grading (cont’d)

• C
  – All the problems addressed (although solution doesn’t work).

• B
  – Solution works. No major flaws.

• A
  – Excellent design. Well engineered piece of work.
Plagiarism?

• Condition: if you did not cite nor explicitly fork off from someone else’s repo, and if we detected that you have copied your peer’s work

• Result in ‘F’

• If you are really concerned about others seeing your solution, commit locally, and push right before the deadline 😊
Private Repository

• 1 repo per student (for exams only)
• We will use some private repos for the final mock CTF
Mock CTF

- Team-based CTF at the end of the course
- 6/7 from 9:00am-
Call For Proposals

• We call for activity proposals
  – Write what kind of activities you want to do in this course? And why?
  – If you are a native Korean, *write in Hangul*
  – If you are not a native Korean, write in English
  – We provide a template (in Latex) in GitHub

• Accepted activity proposals: +20
Lab Configuration

• Use a Vagrant VM:
  − http://143.248.8.33/deb32-jessie-base-vb5.1.12.box
  − http://143.248.8.33/deb32-jessie-base-vb5.1.12.sha1sum
  − If you have your own Linux system you can just use it

• OS: Linux (Ubuntu, Intel 32-bit)
• Language:
  − In some activities, you may have to write in C/C++
  − For the other activities you can use any language you want
    ▪ Examples of good language: F#, Haskell, OCaml, Scala, etc.
Setup

• Install Git
• Install Virtualbox
• Install Vagrant
• $ mkdir IS521; cd IS521
• Download the BOX
• $ vagrant box add IS521 deb32-jessie-base-vb5.1.12.box
• $ vagrant init IS521
• $ vagrant up
• $ vagrant ssh
Activities Outline (Tentative)

• Create your own botnet
• Create your own anti-malware
• Create your own debugger
• Create your own network sniffer/spoofer
• Create your own CTF
• Etc.
Activity #0: LaTeX!

- Install LaTeX on the host machine (not in the VM)
- Visit: [https://www.latex-tutorial.com/tutorials/](https://www.latex-tutorial.com/tutorials/)
- Finish up to “step 8”, using your own “local” git repository
  - We will ask you to push (sync with a remote server) your commits in the next class

- Deadline: before the next class (Mon. 3/6)
  - No need to submit anything
  - Just come to the class with your own local repo
Question?