

Lec 1: Introduction

IS561: Binary Code Analysis and Secure Software Systems

Sang Kil Cha

Who am I?

Sang Kil Cha



- Researcher and Software Engineer.

Sang Kil Cha



- Researcher and Software Engineer.
- Leader of SoftSec. Lab.

Sang Kil Cha



- Researcher and Software Engineer.
- Leader of SoftSec. Lab.
- Director of CSRC (Cyber Security Research Center)
- Chief professor of GSIS.

Sang Kil Cha



- Researcher and Software Engineer.
- Leader of SoftSec. Lab.
- Director of CSRC (Cyber Security Research Center)
- Chief professor of GSIS.
- Research Keywords:
 - Binary Analysis
 - Vulnerability Discovery
 - Exploit Verification
 - Malware Analysis

Contact

- Office: N5, #2319
- Office hour: by appointment.
 - 🏠 Homepage: <https://softsec.kaist.ac.kr/~sangkilc/>
 - ✉ Email: sangkilc@kaist.ac.kr

My Research

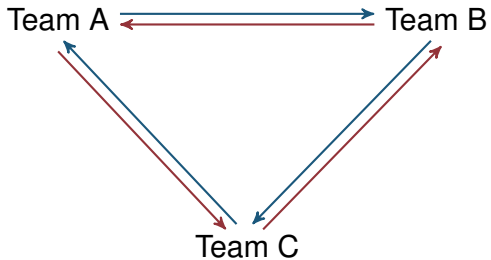
My research is all about building ***large*** and ***complex*** systems that automatically analyze programs to resolve security problems.

My Research: Automatic Exploit Generation

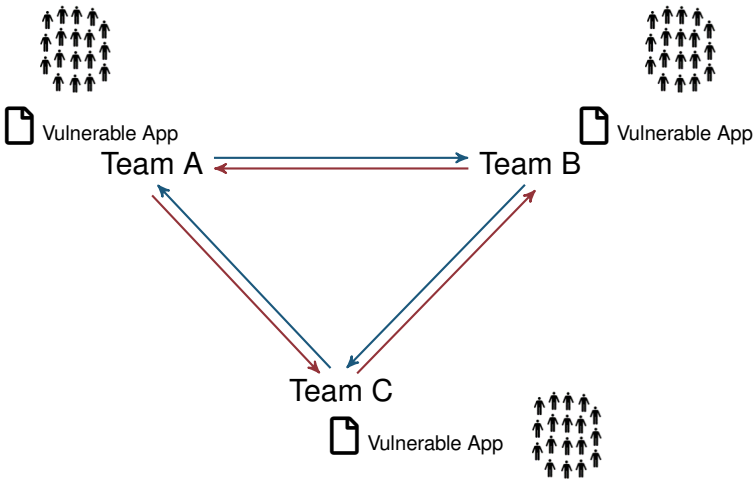
Fully automated hacking and defense
(e.g., DARPA's Cyber Grand Challenge)

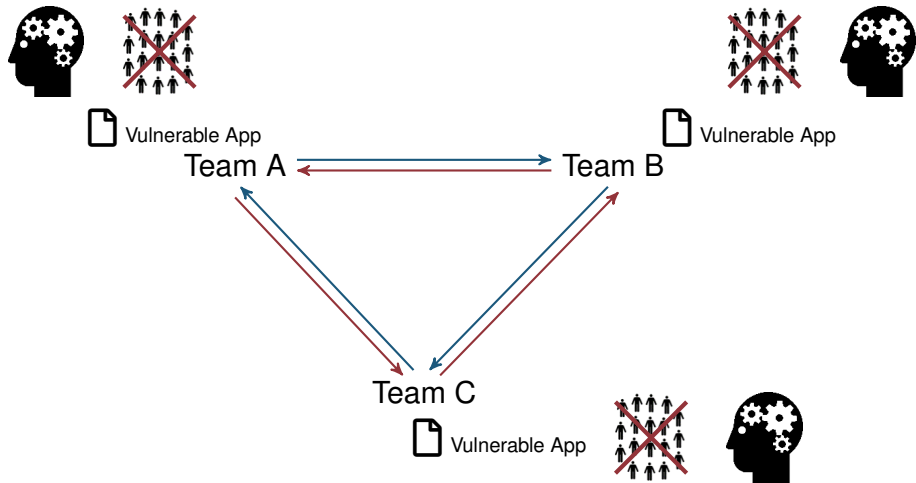


Normal CTFs

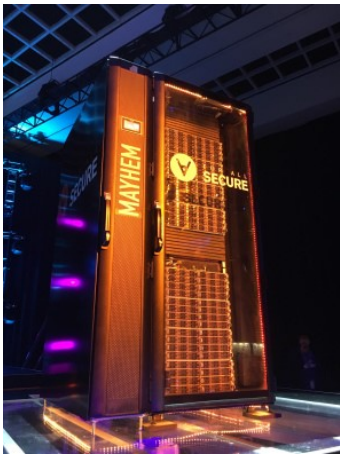


Normal CTFs





Winner = Mayhem



ForAllSecure (Carnegie Mellon University)

Image from <https://forallsecure.com/wp-content/uploads/2016/08/mayhem-crs.jpg>

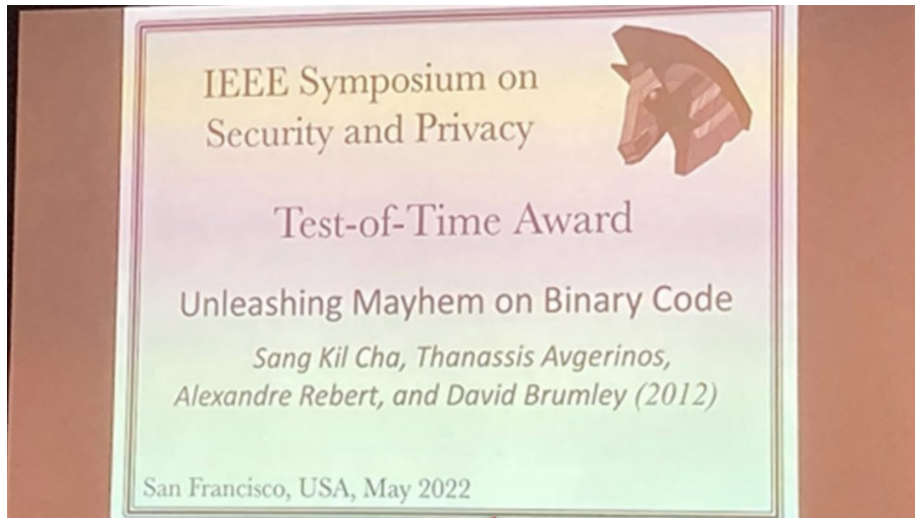
A large, illuminated, orange-colored safe or vault door stands prominently in a dark, industrial setting, likely a museum or exhibition space. The door is open, revealing a grid of small, glowing compartments inside. The word 'MAYHEM' is written vertically on the left side of the door, and 'ALL SECURE' is written horizontally on the right side. A yellow circular logo with a stylized 'V' is visible on the right side of the door. The background features a dark, industrial ceiling with a grid of lights.

2012 IEEE Symposium on Security and Privacy

Sang Kil Cha, Thanassis Avgerinos, Alexandre Rebert and David Brumley
Carnegie Mellon University
Pittsburgh, PA

Image from <https://forallsecure.com/wp-content/uploads/2016/08/mayhem-crs.jpg>

IEEE S&P Test-of-Time Award 2022



My Research: Windows Error Reporting

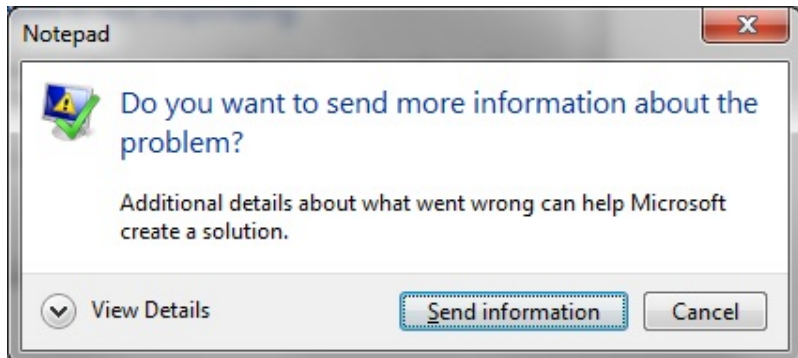
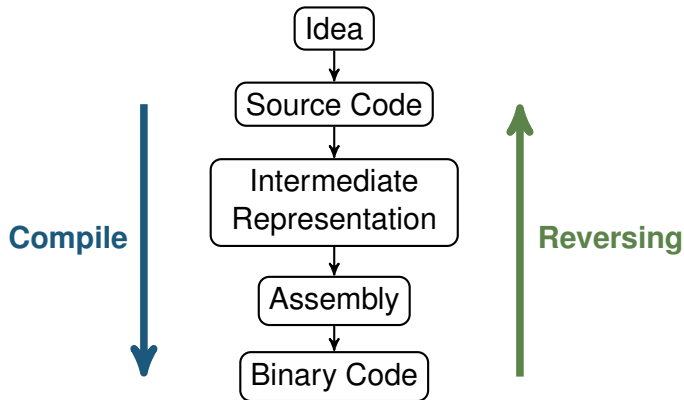


Image from <https://goo.gl/PLekyZ>

My Research: Automatic Bug Finding

- Targeting ***various software products***: browsers, kernels, smart contracts, etc.
- Practical impact: numerous CVEs, algorithms used by mainstream fuzzers, etc.
- Academic impact
 - 2020 FSE Distinguished Paper Award.
 - 2022 TSE Best Paper Award.
 - 2022 ASE Distinguished Paper Award.

My Research: Binary Analysis



My Research: B2R2

- The fastest binary analysis frontend.
- Academic impact
 - 2019 NDSS BAR Best Paper Award.



My Research: Windows Binary Analysis

- Windows binary analyzer built on top of **B2R2**.
- Published in 2021 IEEE Security & Privacy.
- Earned **25,000 USD** bounty for finding zero days.

My Research: Online Game Security

- Automatic aimbot detection.
- Reverse engineered CS:GO binaries to collect real-time game data.
- 2023 USENIX Security ***Distinguished Paper Award***.



Software Security

Software Security

- Software security is a broad topic.
 - Binary code analysis
 - Program analysis
 - Exploit verification
 - Vulnerability discovery
 - Bug detection & classification
 - Malware analysis
 - ...

What is the difference between science and engineering?

Science vs. Engineering



Where there is engineering, there is a
security problem.

Engineering Failure



Why?

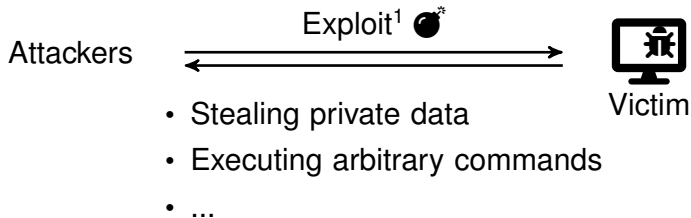
Why?

Humans always make *mistakes*.

And Software is No Exception

- We see hacking attacks, i.e., software failures, every day.
- Because software engineers make ***mistakes***.

Software Bug is the Root of Evil



¹Exploit is a malicious input that triggers bug(s).

Software Security is About Software Bugs

- Find software bugs
- Exploit software bugs
- Patch software bugs

Goal of This course

Understand general ***principles*** of

- How attackers find and exploit bugs
- How current defense techniques work
- How to engineer secure software systems

This course is ***NOT*** about learning hacking skills.

This course is about learning the ***principles***.

Software Security Researcher = Hacker?

- Security research vs. Hacking research?
- Security conference vs. Hacking conference?

Key Takeaway

Security researchers do not need to be like a hacker, but should be like an ***engineer***, because

- There's no provably secure software artifacts,
- Thus, We should make things work (and secure).

Key Takeaway (cont'd)

Good Hacker $\nrightarrow$ Good Engineer
Good Engineer $\rightarrow$ Good Hacker

Key Takeaway (cont'd)

Good Hacker $\nrightarrow$ Good Engineer
Good Engineer $\rightarrow$ Good Hacker

You don't need to be a hacker to do security research, although you can always become a hacker if you need to.

The Course

Target Audience

- Possess ***systems programming*** skills (C/C++)
(prerequisite)
- Possess knowledge about how to ***compile***, ***link***, and ***debug*** programs on Linux
(e.g., gcc, gdb, etc.)
(prerequisite)
- Possess basic knowledge about ***computer architecture*** and ***compiler***
(prerequisite)

Course Resources

- Course web:
<https://softsec.kaist.ac.kr/courses/2024f-is561/>.
- We use **KLMS** for Q&A and announcements.

Course Logistics

- 5% Reading critique, in-class participation
- 20% Homework
- 35% Midterm
- 40% Final

Our Environment

- Linux on x86-64
- GNU Debugger (GDB)
- GCC compiler
- No commercial tools, such as IDA Pro
- We provide a VM.

Vagrant VM

- Vagrant VM
 - Install latest version of VirtualBox
 - Install latest version of Vagrant
 - mkdir YOUR_PATH; cd YOUR_PATH
 - Download box from (redacted: please find the URL from KLMS)
- Read the Vagrant manual
- Basic Vagrant usage (in cmdline prompt)
 - vagrant box add is561 is561_2024.box
 - vagrant init is561
 - vagrant up
 - vagrant ssh

Reading Critique (by tomorrow)

See the course web page.

What Makes Good RC?

- Show your own unique view.
- Do not nitpick on aesthetics. Focus more on the idea.
- Make it easy to read.
- Be concise. Avoid lengthy sentences and paragraphs.

Question?