Lec 1: Introduction

IS521: Information Security Laboratory

Sang Kil Cha
Who am I?
Sang Kil Cha

- Researcher, Engineer, and Hacker.

Research Keywords:
- Binary Analysis
- Vulnerability Discovery
- Exploit Verification
- Malware Analysis
Sang Kil Cha

• Researcher, Engineer, and Hacker.
• Leader of SoftSec. Lab.
Sang Kil Cha

- Researcher, Engineer, and Hacker.
- Leader of SoftSec. Lab.
- 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

Team A

Team B

Team C
Normal CTFs
Normal CTFs

Team A

Team B

Team C

Vulnerable App

Vulnerable App

Vulnerable App

Vulnerable App
Winner = Mayhem

ForAllSecure (Carnegie Mellon University)

Winner = Mayhem

ForAllSecure (Carnegie Mellon University)

2012 IEEE Symposium on Security and Privacy

Unleashing MAYHEM on Binary Code

Sang Kil Cha, Thanassis Avgerinos, Alexandre Rebert and David Brumley
Carnegie Mellon University
Pittsburgh, PA
{sangkilc, thanassis, alexandre.rebert, dbrumley}@cmu.edu

K-CGC (Korean CGC)

- Officially called “KISA Data Challenge”.
K-CGC (Korean CGC)

- Officially called “KISA Data Challenge”.
- Started in 2018.
K-CGC (Korean CGC)

- Officially called “KISA Data Challenge”.
- Started in 2018.
- SoftSec. Lab. won the challenge twice in a row (2018 and 2019).
K-CGC (Korean CGC)

- Officially called “KISA Data Challenge”.
- Started in 2018.
- SoftSec. Lab. won the challenge twice in a row (2018 and 2019).
- Built our own system from scratch!
My Research: Windows Error Reporting

Image from https://goo.gl/PLekyZ
My Research: macOS Kernel Vulnerabilities

About the security content of macOS High Sierra 10.13.2, Security Update 2017-002 Sierra, and Security Update 2017-005 El Capitan

This document describes the security content of macOS High Sierra 10.13.2, Security Update 2017-002 Sierra, and Security Update 2017-005 El Capitan.
A vulnerability was found in Apple iOS up to 12.1. It has been classified as critical. This affects code of the component WebKit. The manipulation with an unknown input leads to a memory corruption vulnerability. CWE is classifying the issue as CWE-119. This is going to have an impact on confidentiality, integrity, and availability.

The weakness was released 12/05/2018 by HyungSeok Han, DongHyeon Oh and Sang Kil Cha with KAIST Softsec Lab as HT209340 as confirmed advisory (Website). The advisory is shared at support.apple.com. This vulnerability is uniquely identified as CVE-2018-4437. It is possible to initiate the attack remotely. No form of authentication is needed for exploitation. Neither technical details nor an exploit are publicly available. The price for an exploit might be around USD $25k-$100k at the moment (estimation calculated on 12/06/2018). It expected to see the exploit prices for this product increasing in the near future. The advisory points out:

<table>
<thead>
<tr>
<th>CVSS Meta Temp Score</th>
<th>Current Exploit Price ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>$25k-$100k</td>
</tr>
</tbody>
</table>
My Research: Binary Analysis

- Idea
- Source Code
- Intermediate Representation
- Assembly
- Binary Code

Compile

Reversing
My Research: B2R2

- The fastest binary analysis frontend.
- The basis of our system for K-CGC.
Let me thank you for the great work you did!

Carnegie Mellon University

Thank you for doing this research!

Thanks for the hard work and the help in making better binary platforms.
What is IS521?
Check the Course Webpage

How can you find it?

- Find my name on Google.
- Click the link for IS-521, 2020.
This is a Laboratory Course

- Lots of hands-on exercises.
- Discussion-oriented course (very few lectures including this one).
Course Description

The primary goal of this course is to “learn by doing”. Students will learn fundamental ideas in computer security by creating and solving CTF-style problems. They will also learn how to present their ideas in a concise manner, which is indeed an essential aspect of being a good security researcher.
CTF (Capture The Flag)

CTF is a **game**.
CTF (Capture The Flag)

CTF is a game. And it is really an effective way to learn security concepts.
CTF as a Education Platform

We will create our own CTF, and we will play it to learn things!
CTF as a Education Platform

We will create our own CTF, and we will play it to learn things!

But, in a principled way.
Typical CTF Experience

• You play the game; you solve several problems but not all.
• You look for solutions after the end of the game to learn. However, you may not find all the solutions.
Our CTF

• You create CTF problems.
• You solve CTF problems made by others.
• You learn from the authors of *all* the problems.
• You learn by analyzing other players’ solutions.
FAQ

Q: Can I take this course if I don’t have any experience on CTFs?

A: Yes!
Course Logistics

Grading:

• 40% CTF.
• 20% Writeup.
• 40% Presentation.
The Goal

The goal of this course is *NOT* becoming a good hacker. It is becoming a *good security researcher*. 
Good Security Researcher?

• You should be able to deliver your ideas in a concise manner.
• You should be able to write down your thoughts.
Good Opportunity for Practicing Presentation

Each student will have at least seven times to present in front of me and TAs.

This is why this course is so unique!
Communication = Slack

TAs will invite you to our Slack. You are free to ping TAs.
CTF Problem Development

• Use Git and GitHub.
• We will provide private repositories to each student.
• Each problem will be eventually made public during the CTF: we open the source code of your problem during the event, but not the git commit histories.
Open-Sourced CTF

What we do is really an open-sourced CTF, which you may never have heard of. However, there are several good reasons why we do this.

• Unlike classic CTF, players can focus more on the problem itself, not on “reversing” the problem.
• Players can come up with more brilliant ideas to hide their vulnerabilities.
• You will naturally learn how to think like an adversary while engineering your problems.
What Kind of Problems Should I Make?

There are many types of CTF problems: pwnable, reversing, web, crypto, etc. There is really no limit in terms of problem types. Use your own expertise to create your problem. But remember, you should be able to explain the problem well during presentation sessions.
CTF Problem Specification

There is a specification that you should follow:

• Your problem should be dockerized.
• Your problem should be solvable from remote: at least one dedicated port should be used.
• You should provide a script that builds your Docker image with a random string as input. The random string is then used as a flag in the container.

In other words, TAs should be able to easily deploy your problems on our server.
Question?