Jiahao Cai

85 Engineer's Way, Charlottesville, VA, 22903 434-227-3265
jc4mf@virginia.edu jiahao-cai.info github.com/jiahao42

Education

University of Virginia Charlottesville, VA, United States
Ph.D. in Computer Science (Drop out), GPA: 3.94/4.0 *Aug 2018 - Jun 2020*

- Research area: Program analysis, Software security, Reverse engineering, Software engineering
- Course: Compiler (A+), Software Security (A+), Analysis of Software Artifacts, Mobile and IoT security

Beijing Information Science & Technology University Beijing, China
Bachelor of Software Engineering, GPA: 3.59/4.00 *Sep 2014 - Jul 2018*

- Course: Data Structure, Object-oriented programming, Operating Systems, Database, Mobile Development
- Awards: First-Class Scholarship for Outstanding Students ***GPA 4.0/4.0, Rank 1/91***, 2017

Working Experience

***RA* - Evil Mastermind: Delivering Malicious Payloads via Innocent Actors** Charlottesville, VA, United States
Advisor: Yonghwi Kwon *Mar 2019 – Sep 2019*

- (In submission) Cai, Jiahao, et al. "Evil Mastermind: Delivering Malicious Payloads via Innocent Actors." Proceedings of The Network and Distributed System Security Symposium (NDSS). 2020.
- Designed a transformation to make malware evade from state-of-the-art detection, with low transformation overhead (6-8s) and negligible runtime overhead (<1s). Embedded transformed malware into existing benign programs.
- Evaluated the transformation on 573 real world PHP malware in 8 categories with 15 static analysis tools, 4 symbolic execution tools and 4 fuzzers to prove it's extremely difficult to analyze and reverse engineering.
- Developed a website profiler with Flask, JQuery and Bootstrap, which can visualize a DOM tree with statistical annotations based on gigabytes of webpages. The profiler is hosted on AWS. The whole system is shipped with Docker.

***RA* - Platform Agnostic Binary code via Causality-based Signature** Charlottesville, VA, United States
Advisor: Yonghwi Kwon *Dec 2018 – Mar 2019*

- Leveraged LLVM to do program analysis (e.g., detect loops/invariants) and instrumentation (e.g., insert log functions) on vulnerable C/C++ programs. Hooked Linux kernel system calls to monitor operations.
- Wrote Perl and Shell scripts to automate the process of instrumenting executable, executing test cases, and preprocessing gigabytes of logs. Developed an algorithm to automatically deduce templates from logs, simplify logs with templates, and align different logs. Utilized Python to extract vulnerability signature from logs and visualize as HTML table.

***TA* - CS 4414: Operating Systems:** offered office hour & grading about xv6 OS *Jan 2019 - May 2019*
***TA* - IT Forensics and Digital Evidence:** supervised experiments on forensics in filesystem (FAT32) *June 2017*
***TA* - Object-oriented programming:** taught concepts in OOP with C++. *February 2017 - May 2017*

Notable Projects

Compiler for Meggy Java Charlottesville, VA, United States
github.com/jiahao42/MeggyJava-Compiler *Sep 2018 – Dec 2018*

- Developed a compiler to translate Java into Assembly using visitor design pattern. Implemented features such as function, call stack, class, static scope, type checking/cast, dynamic memory allocation.
- Visualized abstract syntax tree (AST) and symbol table via Graphviz. Wrote high coverage regression test for the compiler.

Kernel for embedded OS Halmstad, Sweden
github.com/jiahao42/SimpleKernel *Jan 2018 – Apr 2018*

- Implemented task administration by Task Control Block (TCB) including stack, context, deadline, etc.
- Enabled asynchronous task and inter process communication (IPC) by creating a mailbox for exchanging messages. Wrote high coverage tests for the kernel. Leveraged QEMU to simulate the embedded environment.

Zhihu Daily Android Client Beijing, China
github.com/jiahao42/Simplified-Zhihu-Daily *Sep 2016 – Oct 2016*

- Implemented functions such as splash, animation, viewing questions/answers. Enabled sharing function by integrating sharing SDK. Designed cache by serializing retrieved data and storing in SQLite database.
- Created a web crawler to simulate HTTP request to login and crawl user profile. Utilized regular expression to parse crawled content. Note that login and viewing user profile was ***not even implemented in the official app*** back then.
- Uploaded it to application store and got hundreds of downloads.

Skills

Languages: C/C++, Python, Java, Javascript, Racket, PHP, Ruby, Shell script, Perl, SQL, HTML, CSS
Framework: Flask, Django, Spring, Bootstrap, JQuery, AJAX, Ruby on Rails
System & Tools: GNU/Linux, Git, Docker, Apache, Vim, Make, MySQL, PostgreSQL, GCC, GDB