Qingkai Shi

Department of Computer Science and Engineering, Hong Kong University of Science and Technology

Email: qingkaishi@gmail.com **Phone:** +852-5578-1990 **Web:** https://qingkaishi.github.io/

Research Interest

My research interest centers around cybersecurity (**SEC**), programming language (**PL**), and software engineering (**SE**). I aim to address **SEC** problems by developing **PL** and **SE** methods, or address **PL** and **SE** problems to support **SEC** analysis. More specifically, I focus on the use of both static and dynamic program analysis for making software systems more secure and reliable.

Education and Employment

Ph.D., Computer Science and Engineering, 2015 - 2020
The Hong Kong University of Science and Technology, GPA: 3.94/4.0
Thesis: Precise and Scalable Static Bug Finding for Industrial-Sized Code
Thesis Supervisor: Dr. Charles Zhang

B.S., Software Engineering, 2008 - 2012
Nanjing University, GPA: 3.94/4.0

Co-founder, 2014 - 2020, Sourcebrella Inc., Shenzhen, China
Web: https://www.sourcebrella.com/
Sourcebrella Inc. commercializes my research on static bug finding (read my PhD thesis for details).
Sourcebrella Inc. was acquired by Ant Financial Services Group in 2020.

Honors

- ACM SIGSOFT Distinguished Paper Award (2019)
- Champion of NASAC Prototype Competition (2016, 2018a, 2018b)
- Hong Kong PhD Fellowship (2015)
- China National Scholarship (2010, 2015)

Selected Publications

- Peisen Yao, **Qingkai Shi***, Heqing Huang, and Charles Zhang. Fast Bit-Vector Satisfiability. In **ISSTA 2020**: the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis. Los Angeles, CA, USA. July 2020.
(***Corresponding Author – Paper under My Supervision**)
- Gang Fan, Chengpeng Wang, Rongxin Wu, **Qingkai Shi**, and Charles Zhang. Escaping Dependency Hell: Finding Build Dependency Errors with the Unified Dependency Graph. In **ISSTA 2020**: the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis. Los Angeles, CA, USA. July 2020.
- Chunrong Fang, Zixi Liu, Yangyang Shi, Jeff Huang, and **Qingkai Shi***. Functional Code Clone Detection with Syntax and Semantics Fusion Learning. In **ISSTA 2020**: the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis. Los Angeles, CA, USA. July 2020.
- Yang Feng, **Qingkai Shi***, Xinyu Gao, Jun Wan, Chunrong Fang, and Zhenyu Chen. DeepGini: Prioritizing Massive Tests to Enhance the Robustness of Deep Neuron Networks. In **ISSTA 2020**: the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis. Los Angeles, CA, USA. July 2020.

- Heqing Huang, Peisen Yao, Rongxin Wu, **Qingkai Shi***, Charles Zhang. Pangolin: Incremental Hybrid Fuzzing with Polyhedral Path Abstraction. In **S&P 2020**: the 41st IEEE Symposium on Security and Privacy. San Francisco, CA, United States. May 2020.
- **Qingkai Shi**, Charles Zhang. Pipelining Bottom-up Data Flow Analysis. In **ICSE 2020**: the 42nd ACM/IEEE International Conference on Software Engineering. Seoul, South Korea. May 2020.
- **Qingkai Shi**, Rongxin Wu, Gang Fan, Charles Zhang. Conquering the Extensional Scalability Problem for Value-Flow Analysis Frameworks. In **ICSE 2020**: the 42nd ACM/IEEE International Conference on Software Engineering. Seoul, South Korea. May 2020.
- Gang Fan, Rongxin Wu, **Qingkai Shi**, Xiao Xiao, Jinguo Zhou, Charles Zhang. SMOKE: Scalable Path-Sensitive Memory Leak Detection for Millions of Lines of Code. In **ICSE 2019**: the 41st ACM/IEEE International Conference on Software Engineering. Montreal, QC, Canada. May 2019.
  (**ACM SIGSOFT Distinguished Paper Award**)
- **Qingkai Shi**, Xiao Xiao, Rongxin Wu, Jinguo Zhou, Gang Fan, Charles Zhang. Pinpoint: Fast and Precise Sparse Value Flow Analysis for Million Lines of Code. In **PLDI 2018**: the 39th annual ACM SIGPLAN conference on Programming Language Design and Implementation. Philadelphia, PA, United States. June 2018.
- **Qingkai Shi**, Zhenyu Chen, Chunrong Fang, Yang Feng, Baowen Xu. Measuring the Diversity of a Test Set with Distance Entropy. In **TRel 2016**: IEEE Transactions on Reliability, Vol. 65, No. 1, 2016.
- **Qingkai Shi**, Jeff Huang, Zhenyu Chen, and Baowen Xu. Verifying Synchronization for Atomicity Violation Fixing. In **TSE 2016**: IEEE Transactions on Software Engineering, Vol. 42, No. 3, 2016.

CVE IDs

- CVE-2017-14739: ImageMagick 7.0.74 mishandles failed memory allocation, which allows remote attackers to cause a denial of service.
- CVE-2017-14952: International Components for Unicode (ICU) for C/C++ through 59.1 contains a double free that allows remote attackers to execute arbitrary code.
- CVE-2017-15096: GlusterFS in versions prior to 3.10 contains a null pointer dereference that may cause denial of service.
- CVE-2017-16892: Bftpd 4.6 contains a memory leak which occurs if a mal-crafted sequence of FTP requests are received.
- CVE-2017-1000445: ImageMagick 7.0.71 and older version are vulnerable to null pointer dereference in the MagickCore component and might lead to denial of service.
- CVE-2018-20786: libvterm through 0+bzr726, as used in Vim and other products, mishandles certain out-of-memory conditions, leading to a denial of service (application crash), related to screen.c, etc.
- CVE-2019-13238: An issue was discovered in Bento4 1.5.1.0. A memory allocation failure is unhandled in Core/Ap4SdpAtom.cpp and leads to crashes. When parsing input video, the program allocates a new buffer to parse an atom in the stream. The unhandled memory allocation failure causes a direct copy to a NULL pointer.
- CVE-2019-13959: In Bento4 1.5.1-627, AP4_DataBuffer::SetDataSize does not handle reallocation failures, leading to a memory copy into a NULL pointer.
- CVE-2019-13960: In libjpeg-turbo 2.0.2, a large amount of memory can be used during processing of an invalid progressive JPEG image containing incorrect width and height values in the image header.

Notable Projects

- Pinpoint Static Analyzer
  - Pinpoint is an industrial-strength next-generation automated bug finding tool through static program analysis. This is my main research work when I was a Ph.D. student.
  - It has found about a hundred vulnerabilities in many mature open-source projects, including Apache, MySQL, Firefox, Python, OpenSSL, etc. (https://whichbug.github.io/).
  - Some of the detected vulnerabilities have been assigned CVE IDs (see last section).
  - The project has been commercialized in Sourcebrella Inc. (https://www.sourcebrella.com/).
  - Related technical papers were published in PLDI 2018, ICSE 2019, ICSE 2020a, ICSE 2020b, ISSTA 2020a, ISSTA 2020b, ISSTA 2020c.
  - Read my PhD thesis for the core techniques.
- Pangolin Fuzzing System
  - Pangolin is an industrial-strength fuzzing system that aims to find software vulnerabilities through fuzz testing.
  - We propose domain-specific testing criteria to guide fuzzing in different application scenarios (TSE 2016, TRel 2016, ISSTA 2020d).
  - We propose incremental techniques to reduce the cost of random input mutation and SMT solving (S&P 2020).
  - Some of the detected vulnerabilities have been assigned CVE IDs (see last section).

Professional Services

- Reviewer or sub-reviewer for IEEE Transactions on Software Engineering, IEEE Transactions on Reliability, ICSE, FSE, ISSTA, ASE.
- Student volunteer for PLDI 2016, QSIC 2013, and PLDI 2012.

Teaching Experience

- Teaching Assistant for COMP3111/3111H: Software Engineering (Fall 2018)
- Teaching Assistant for COMP4111: Software Engineering Practices (Fall 2016, Spring 2018)

April 28, 2020