Qingkai Shi

ADDRESS

Department of Computer and Science
Purdue University
305 N. University Street
West Lafayette, IN 47907
Email: shi553@purdue.edu
URL: https://qingkaishi.github.io/
ORCID: 0000-0002-8297-8998

EDUCATION & EMPLOYMENT

Purdue University 2021 - Now
Postdoctoral Research Associate

Hong Kong University of Science and Technology / Sourcebrella Inc. 2015 - 2021
Ph.D. in Computer Science (2020) / Co-founder of Sourcebrella Inc. (2021)
Thesis: Precise and Scalable Static Bug Finding for Industrial-Sized Code

RESEARCH & AWARDS

My research focuses on the use of programming language and compiler techniques, especially static analysis, for rigorously ensuring software security, including static analysis for (1) bug scanning, (2) reverse engineering, and (3) fuzz testing. My research has been extensively published in premium venues and has let me win ACM SIGPLAN Distinguished Paper Award, ACM SIGSOFT Distinguished Paper Award, Google Research Paper Reward, and Hong Kong Ph.D. Fellowship.

My research on bug scanning, known as the *Pinpoint Static Analyzer*, has received 200M CNY ($\approx$ 30M USD) investment. The startup that commercializes my research, Sourcebrella Inc., has been acquired by Ant Group, where *Pinpoint* is deployed in daily operations for improving the quality of Alipay, a popular digital payment app with over a billion monthly active users. To date, we have discovered hundreds of vulnerabilities and CVEs in mature software.

For further information, see https://qingkaishi.github.io/.

REFEREED PUBLICATIONS

My research has been published extensively at premium venues of programming language (PLDI, OOPSLA), software engineering (ICSE, FSE), and cybersecurity (S&P, CCS).

Representative Papers:

1. **Qingkai Shi**, Xiangzhe Xu, and Xiangyu Zhang. Extracting Protocol Format as State Machine via Controlled Static Loop Analysis. In *Proceedings of the USENIX Security Symposium (Security'23)*. 2023.
2. **Qingkai Shi**, Junyang Shao, Yapeng Ye, Mingwei Zheng, and Xiangyu Zhang. Lifting Network Protocol Implementation to Precise Format Specification with Security Applications. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS'23)*. 2023.
3. **Qingkai Shi**, Yongchao Wang, Peisen Yao, and Charles Zhang. Indexing the Extended Dyck-CFL Reachability for Context-Sensitive Program Analysis. In *Proceedings of the ACM on Programming Languages (OOPSLA'22)*. 2022.

4. **Qingkai Shi**, Peisen Yao, Rongxin Wu, and Charles Zhang. Path-Sensitive Sparse Analysis without Path Conditions. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'21)*. 2021.

5. **Qingkai Shi** and Charles Zhang. Pipelining Bottom-up Data Flow Analysis. In *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'20)*. 2020.

6. **Qingkai Shi**, Rongxin Wu, Gang Fan, and Charles Zhang. Conquering the Extensional Scalability Problem for Value-Flow Analysis Frameworks. In *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'20)*. 2020.

7. **Qingkai Shi**, Xiao Xiao, Rongxin Wu, Jinguo Zhou, Gang Fan, and Charles Zhang. Pinpoint: Fast and Precise Sparse Value Flow Analysis for Million Lines of Code. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'18)*. 2018.

8. **Qingkai Shi**, Jeff Huang, Zhenyu Chen, and Baowen Xu. Verifying Synchronization for Atomicity Violation Fixing. In *the IEEE Transactions on Software Engineering (TSE'16)*. 2016.

9. **Qingkai Shi**, Zhenyu Chen, Chunrong Fang, Yang Feng, and Baowen Xu. Measuring the Diversity of a Test Set with Distance Entropy. In *the IEEE Transactions on Reliability (TRel'16)*. 2016.

Other Papers:

10. Yapeng Ye, Zhuo Zhang, **Qingkai Shi**, Yousra Aafer, and Xiangyu Zhang. D-ARM: Disassembling ARM Binaries by Lightweight Superset Instruction Interpretation and Graph Modeling. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P'23)*. 2023.

11. Xiangzhe Xu, Z. Xun, S. Feng, S. Chen, Y. Ye, **Qingkai Shi**, G. Tao, L. Yu, Z. Zhang, and Xiangyu Zhang. PEM: Representing Binary Program Semantics for Similarity Analysis via A Probabilistic Execution Model. In *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'23)*. 2023.

12. Xiangzhe Xu, S. Feng, Y. Ye, G. Shen, Z. Su, S. Chen, G. Tao, **Qingkai Shi**, Z. Zhang, and Xiangyu Zhang. Improving Binary Code Similarity Transformer Models by Semantics-driven Instruction Deemphasis. In *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'23)*. 2023.

13. Chengpeng Wang, Wenyang Wang, Peisen Yao, **Qingkai Shi**, Jinguo Zhou, Xiao Xiao, and Charles Zhang. Anchor: Fast and Precise Value-Flow Analysis for Containers via Memory Orientation. In *the ACM Transactions on Software Engineering and Methodology (TOSEM'23)*. 2023.

14. Chengpeng Wang, Peisen Yao, Wensheng Tang, **Qingkai Shi**, and Charles Zhang. Complexity-Guided Container Replacement Synthesis. In *Proceedings of the ACM on Programming Languages (OOPSLA'22)*. 2022. **ACM SIGPLAN Distinguished Paper Award**

15. Yuandao Cai, Chengfeng Ye, **Qingkai Shi**, and Charles Zhang. Peahen: Fast and Precise Static Deadlock Detection via Context Reduction. In *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'22)*. 2022.

16. Yiyuan Guo, Jinguo Zhou, Peisen Yao, **Qingkai Shi**, and Charles Zhang. Precise Divide-By-Zero Detection with Affirmative Evidence. In *Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'22)*. 2022.

17. Heqing Huang, Yiyuan Guo, **Qingkai Shi**, Peisen Yao, Rongxin Wu, and Charles Zhang. Beacon: Directed Grey-Box Fuzzing with Provable Path Pruning. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P'22)*. 2022. **Google Research Paper Reward**

18. Peisen Yao, **Qingkai Shi**, Heqing Huang, and Charles Zhang. Program Analysis via Efficient Symbolic Abstraction. In *Proceedings of the ACM on Programming Languages (OOPSLA'21)*. 2021.

19. Peisen Yao, Heqing Huang, Wensheng Tang, **Qingkai Shi**, Rongxin Wu, and Charles Zhang. Skeletal Approximation Enumeration for SMT Solver Testing. In *Proceedings of the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'21)*. 2021.

20. Peisen Yao, Heqing Huang, Wensheng Tang, **Qingkai Shi**, Rongxin Wu, and Charles Zhang. Fuzzing SMT Solvers via Two-Dimensional Input Space Exploration. In *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'21)*. 2021.

21. Heqing Huang, Peisen Yao, Rongxin Wu, **Qingkai Shi**, and Charles Zhang. Pangolin: Incremental Hybrid Fuzzing with Polyhedral Path Abstraction. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P'20)*. 2020.

22. Peisen Yao, **Qingkai Shi**, Heqing Huang, and Charles Zhang. Fast Bit-Vector Satisfiability. In *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. 2020.

23. Gang Fan, Chengpeng Wang, Rongxin Wu, Xiao Xiao, **Qingkai Shi**, and Charles Zhang. Escaping Dependency Hell: Finding Build Dependency Errors with the Unified Dependency Graph. In *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. 2020.

24. Yang Feng, **Qingkai Shi**, Xinyu Gao, Jun Wan, Chunrong Fang, and Zhenyu Chen. DeepGini: Prioritizing Massive Tests to Enhance the Robustness of Deep Neural Networks. In *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. 2020.

25. Chunrong Fang, Zixi Liu, Yangyang Shi, Jeff Huang, and **Qingkai Shi**. Functional Code Clone Detection with Syntax and Semantics Fusion Learning. In *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. 2020.

26. Gang Fan, Rongxin Wu, **Qingkai Shi**, Xiao Xiao, Jinguo Zhou, and Charles Zhang. Smoke: Scalable Path-Sensitive Memory Leak Detection for Millions of Lines of Code. In *Proceedings of the ACM/IEEE International Conference on Software En-*

gineering (ICSE'19). 2019. **ACM SIGSOFT Distinguished Paper Award**

PATENTS

Defect detection method, device. system, and computer readable medium.

- US Patent No. 20190108003
- China Patent No. 201811013103, 201811013000, 201811015751, 2018110146864

INVITED TALKS

Fast and Precise Static Bug Detection for Industrial-Sized Code. Texas A&M University, College Station, United States, July 2021.

Taming Path-Sensitivity for Static Code Analysis on Industrial Scale. Nanjing University, Nanjing, China, April 2021.

The Next Generation Software Security Analysis. Nanjing University, Nanjing, China, May 2020.

Marrying Taint Analysis with Big Data Analytics. Southern University of Science and Technology, Shenzhen, China, November 2018.

PROFESSIONAL SERVICES

Reviewer, *ACM Computing Surveys*

Reviewer, *IEEE Transactions on Software Engineering*

Reviewer, *IEEE Transactions on Dependable and Secure Computing*

Reviewer, *IEEE Transactions on Reliability*

Reviewer, *IEEE Transactions on Emerging Topics in Computing*

Member of Artifact Evaluation Committee, *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'22)*

Member of Artifact Evaluation Committee, *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'23)*

Member of External Review Committee and Artifact Evaluation Committee, *European Conference on Object-Oriented Programming (ECOOP'23)*

TEACHING EXPERIENCE

Teaching Assistant, COMP3111/3111H: Software Engineering (Fall 2018)

Teaching Assistant, COMP4111: Software Engineering Practices (Spring 2018)

Teaching Assistant, COMP4111: Software Engineering Practices (Fall 2016)