Yu Hao

Last updated: May, 2024
Email: yhao016@ucr.edu
Phone: 951-783-7821
Website: zhyfeng.github.io
[Google Scholar] [DBLP] [Github]

My research focuses on system security and program analysis. My research improves multiple program analysis and testing techniques (*e.g.,* fuzzing, symbolic execution, static analysis and large language models for program analysis), thereby combining the advantages of those techniques to systematically and automatically improve the security for real world software system (*e.g.,* Linux kernel, concurrent program). My research has led to multiple papers published in several prestigious conferences (*e.g.,* S&P, USENIX Security, CCS, NDSS, ICSE, FSE) and journal (*e.g.,* TSE). The open source tools attract interest from and applied in academia, community and industry. Besides open source tools, I also report bugs and CVE for Linux kernel and contribute patches.

Research Objective

System Security	• **Operating System Security**, Concurrency Security.
Program Analysis	• **Fuzzing**, **LLMs for Security**, **Symbolic Execution**, **Static Analysis**.

Education

09/2018 – 08/2024 **Ph.D.** in Computer Science, University of California, Riverside, USA. Advisor: Zhiyun Qian
Operating System Security, LLMs for Security, Fuzzing, Symbolic Execution, Static Analysis

09/2015 – 06/2018 **Master** in Control Science and Engineering, Xi'an Jiaotong University, China. Advisor: Ting Liu
Concurrency Security, Symbolic Execution

09/2011 – 06/2015 **Bachelor** in Control Science and Engineering, Xi'an Jiaotong University, China.

09/2008 – 06/2011 Hohhot No.2 High School, Inner Mongolia, China.

Experience

09/2018 – present **Graduate Student Researcher**, Advisor: Zhiyun Qian
Center for Research and Education in Cyber Security and Privacy (CRESP)
University of California, Riverside, USA
Operating System Security, LLMs for Security, Fuzzing, Symbolic Execution, Static Analysis

06/2023 – 09/2023 [Proj4.3] **Research Intern**, Mentor: Weidong Cui
Systems Security and Privacy Research Group, Microsoft Research, USA
LLMs for Security

06/2022 – 09/2022 [Proj3.1] **Research Intern**, Mentor: Kris Alder
Android Kernel Security Team, Google, USA
Linux/Android Kernel Security, Fuzzing, Static Analysis

06/2015 – 06/2018 **Research Assistant**, Advisor: Ting Liu
Ministry of Education Key Lab For Intelligent Networks and Network Security
Xi'an JiaoTong University, China
Concurrency Security, Symbolic Execution

06/2013 – 06/2015 **Undergraduate Research Assistant**, Advisor: Ting Liu
Ministry of Education Key Lab For Intelligent Networks and Network Security
Xi'an JiaoTong University, China
Symbolic Execution

Projects

Thread 4 - Improve Linux/Android Kernel Fuzzing

- [Proj4.1, 2018-2021, Lead] Demystifying the Dependency Challenge in Kernel Fuzzing.
  I undertake a substantial measurement study (e.g., static analysis based on LLVM, experimental solutions of fuzzer based on syzkaller) to systematically understand the real challenge behind dependency challenge, which is critical for improving kernel fuzzing. I demystify the real root causes behind the dependency challenge, which provides valuable guidance to future research in kernel fuzzing.
  - 10.9k C++, 4.7k Golang, 0.5k Protobuf LoC for static analysis based on LLVM, fuzzing solutions based on syzkaller, and gRPC between static analysis and fuzzer.
  - [C9, ICSE 22] Tool Result [Google Research Paper Rewards]
- [Proj4.2, 2021-2023, Lead] SyzDescribe: Principled, Automated, Static Generation of Syscall Descriptions for Kernel Drivers.
  Following previous project to improve kernel fuzzing, I present SyzDescribe that can automatically generate syscall descriptions for Linux kernel drivers by static analysis based on LLVM. SyzDescribe are better than manually-curated ones, and much better than prior work. Besides some patches for fuzzer (i.e., syzkaller), SyzDescribe also finds many new bugs in the Linux kernel and 18 bugs in Android kernel of Pixel 6.
  - 8.2k C++, 0.3k Golang LoC for static analysis based on LLVM.
  - [C10, S&P 23] Tool Result
  - [Linux Security Summit 23] [Qualcomm Product Security Summit 23] [Symposium on the Science of Security 24]
  - Selected Patches for syzkaller: [Patch1] [Patch2] [Patch3]
  - Selected CVEs for Linux: [CVE-2023-31081] [CVE-2023-31082] [CVE-2023-31083] [CVE-2023-31084] [CVE-2023-31085]
- [Proj4.3, 2022, Internship at Google, Lead] Identify Priority Fuzzing Targets in the Android Kernel (Pixel 7/7 pro).
  There is a limited time to test the pixel phone before its release. I developed a tool to identify priority fuzzing targets based on customized patterns from engineers in the Android kernel (Pixel 7/7 pro) by static analysis based on LLVM.
- [Proj4.4, Ongoing, Lead] Generation of Syscall Descriptions for Complex Subsystems in the Linux Kernel.
  Some subsystems in the Linux kernel are complex and static analysis can not work well. In this project, I am working on generating syscall descriptions with under constrained symbolic execution (based on KLEE/LLVM) for complex subsystems (e.g., GPU/DRM, V4L2).
  - C++ LoC for under constrained symbolic execution based on KLEE/LLVM and static analysis based on LLVM.

Thread 3 - Analysis of Fuzzing-Discovered Bugs of Linux Kernel

- [Proj3.1, 2023, Internship at Microsoft Research, Lead] Triaging Bugs from Linux Kernel Fuzzing with LLMs.
  I leverage LLMs for backward taint analysis for triaging bugs, which is to figure out the function where the patch of the bugs should be. The evaluation shows that the approach can achieve an average accuracy of more than 80% for 170 bugs on seven different critical bug categories from the Linux kernel.
  - [P1, arXiv] [AGI Leap Summit 2024] [Symposium on the Science of Security 24]
- [Proj3.2, 2022-2023, Code Contribution] SyzBridge: Bridging the Gap in Exploitability Assessment of Linux Kernel Bugs.
  We present SyzBridge, which provides the possibility of bringing Linux upstream kernel bug PoCs to the downstream kernels. It is a fully automatic system that adapts upstream PoCs by tuning race condition, removing unnecessary setup, and loading missing kernel modules. The evaluation shows that SyzBridge can adapt 50+ highly exploitable bugs on downstream kernels (e.g., Ubuntu, Debian, Fedora, and Suse).
  - 3k C++ LoC for static analysis based on LLVM.
  - [C12, NDSS 24] Tool
  - Selected CVEs for Linux kernel: [CVE-2022-27666] [CVE-2021-42008]
- [Proj3.3, 2022-2024, Research Mentoring] SymBisect: Accurate Bisection for Fuzzer-Exposed Vulnerabilities.
  We present SymBisect, which would verify the underlying bug logic and do accurate bisection for fuzzer-exposed vulnerabilities of Linux kernel. We apply under-constrained symbolic execution with several principled guiding techniques to efficiently search for the presence of the bug logic. We show that our approach achieves significantly better accuracy than the state-of-the-art solution by 16% (from 74.7% to 90.7%).
  - C++ Code for under constrained symbolic execution based on KLEE/LLVM.
  - [C5, USENIX Security 24]

Thread 2 - Detect Use-before-Initialization Bugs in Linux Kernel

- [Proj2.1, 2018-2021, Code Contribution] By Static Analysis and Under-Constrained Symbolic Execution.
  We employed a static analysis designed for UBI vulnerability identification across the Linux kernel. To address the false positive issue, I spearheaded the integration of under-constrained symbolic execution as a post-processing mechanism to refine the results generated through static analysis. I successfully eliminated approximately 87.4% of the initially reported false positives, thereby significantly enhancing the reliability of UBI vulnerability detection in the Linux kernel.
  - 5k C++ LoC for under constrained symbolic execution based on KLEE/LLVM.
  - [C4, FSE 20] Tool [C8, NDSS 22] Tool [2023 Cyber Security CRA Capstone Poster]
  - Selected Patches for Linux: [Patch1] [Patch2] [Patch3] [Patch4] [Patch5] [Patch6]
- [Proj2.2, 2022-2023, Research Mentoring] By Static Analysis and LLMs (Large Language Models).
  To mitigate the path explosion issue of symbolic execution, we subsequently employ LLMs for path sensitive program analysis, aiming to bolster the overall accuracy of the detection.
  - [C14, OOPSLA 24] Tool [C11, FSE-IVR 23] Tool [arXiv] [S&P 23 Poster]

Thread 1 - Concurrency Security

- [Proj1.1, 2015-2016, Code Contribution] Debugging Interleaving of Concurrent Program with Concolic Execution.
  I develop a tool to recover possible interleavings of concurrent programs with concolic execution. This tool based on the synergistic integration of symbolic analysis and dynamic analysis techniques.
  - 4k C++ LoC for concolic execution based on KLEE/LLVM.
  - [C1, SATE] [J1, *IEEE Access*] [C2, ICST] Tool v1 Tool v2
- [Proj1.2, 2016-2017, Code Contribution] Dynamic Taint Analysis for Concurrent Program with Concolic Execution.
  I designed and developed DSTAM, a tool designed to systematically identify tainted instances across all conceivable thread interleavings. To the best of our knowledge, DSTAM represents the first-ever solution that addresses the challenges of conducting taint analysis on concurrent programs while keeping the input variables constant.
  - 4k C++ LoC for Concolic Execution based on KLEE/LLVM, Taint Analysis, Shadow Memory and benchmarks.
  - [J2, TSE] [S&P 17 Poster] Patent: [PCT] [CN] Tool Benchmarks Result
- [Proj1.3, 2017-2018, Lead] Malware Hidden in Concurrent Programs.
  I present a new security threat that hides malware in nondeterministic thread interleavings. The malicious behavior can be triggered by certain thread interleavings that rarely happen (e.g., <1%) under a normal execution environment. I can activate such malware with a very high probability (e.g., >90%) by remotely disturbing thread scheduling. This can bypass most of the antivirus engines in VirusTotal and four well-known online dynamic malware analysis systems.
  - [C3, QRS, Best Paper Award] [J3, *IEEE Transactions on Reliability*] Code Result

Publications

[C]: Conference, [J]: Journal, [P]: Preprint

[C15] USENIX Security 24
SymBisect: Accurate Bisection for Fuzzer-Exposed Vulnerabilities
Zheng Zhang, **Yu Hao**, Weiteng Chen, Xiaochen Zou, Xingyu Li, Haonan Li, Yizhuo Zhai, Zhiyun Qian, Billy Lau
USENIX Security Symposium 2024.
Operating System Security, Symbiolic Execution

[C14] OOPSLA 24
Enhancing Static Analysis for Practical Bug Detection: An LLM-Integrated Approach
Haonan Li, **Yu Hao**, Yizhuo Zhai, Zhiyun Qian
ACM SIGPLAN International Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA 24. Tool

[C13] S&P 24
SyzGen++: Dependency Inference for Augmenting Kernel Driver Fuzzing
Weiteng Chen, **Yu Hao**, Zheng Zhang, Xiaochen Zou, Dhilung Kirat, Shachee Mishra, Douglas Schales, Jiyong Jang, Zhiyun Qian
IEEE Symposium on Security and Privacy, S&P 24. Tool

[C12] NDSS 24
SyzBridge: Bridging the Gap in Exploitability Assessment of Linux Kernel Bugs in the Linux Ecosystem
Xiaochen Zou, **Yu Hao**, Zheng Zhang, Juefei Pu, Weiteng Chen, Zhiyun Qian
Network and Distributed System Security Symposium, NDSS 24. Tool

[P1]
E&V: Prompting Large Language Models to Perform Static Analysis by Pseudo-code Execution and Verification
Yu Hao, Weiteng Chen, Ziqiao Zhou, Weidong Cui *arXiv*

[C11] FSE 23 IVR
Assisting Static Analysis with Large Language Models: A ChatGPT Experiment
Haonan Li, **Yu Hao**, Yizhuo Zhai, Zhiyun Qian
The ACM International Conference on the Foundations of Software Engineering, Ideas, Visions and Reflections, FSE 23 IVR Tool *arXiv*

[C10] SyzDescribe: Principled, Automated, Static Generation of Syscall Descriptions for Kernel Drivers
S&P 23 **Yu Hao**, Guoren Li, Xiaochen Zou, Weiteng Chen, Shitong Zhu, Zhiyun Qian, Ardalan Amiri Sani
IEEE Symposium on Security and Privacy, S&P 23. Tool Result

[C9] Demystifying the Dependency Challenge in Kernel Fuzzing
ICSE 22 **Yu Hao**, Hang Zhang, Guoren Li, Xingyun Du, Zhiyun Qian, Ardalan Amiri Sani
IEEE/ACM International Conference on Software Engineering, ICSE 22. Tool Result

[C8] Progressive Scrutiny: Incremental Detection of UBI bugs in the Linux Kernel
NDSS 22 Yizhuo Zhai, **Yu Hao**, Zheng Zhang, Weiteng Chen, Guoren Li, Zhiyun Qian, Chengyu Song, Manu Sridharan, Srikanth V. Krishnamurthy, Trent Jaeger, Paul Yu
Network and Distributed System Security Symposium, NDSS 22. Tool

[C7] Eluding ML-based Adblockers With Actionable Adversarial Examples
ACSAC 21 Shitong Zhu, Zhongjie Wang, Xun Chen, Shasha Li, Keyu Man, Umar Iqbal, Zhiyun Qian, Kevin S Chan, Srikanth V Krishnamurthy, Zubair Shafiq, **Yu Hao**, Guoren Li, Zheng Zhang, Xiaochen Zou
Annual Computer Security Applications Conference, ACSAC 21. Tool

[C6] Themis: Ambiguity-Aware Network Intrusion Detection based on Symbolic Model Comparison
CCS 21 Zhongjie Wang, Shitong Zhu, Keyu Man, Pengxiong Zhu, **Yu Hao**, Zhiyun Qian, Srikanth V. Krishnamurthy, Tom La Porta, Michael J. De Lucia
ACM SIGSAC Conference on Computer and Communications Security, CCS 21. Tool

[C5] Statically Discovering High-Order Taint Style Vulnerabilities in OS Kernels
CCS 21 Hang Zhang, Weiteng Chen, **Yu Hao**, Guoren Li, Yizhuo Zhai, Xiaochen Zou, Zhiyun Qian
ACM SIGSAC Conference on Computer and Communications Security, CCS 21. Tool

[C4] UBITect: A Precise and Scalable Method to Detect Use-before-Initialization Bugs in Linux Kernel
FSE 20 Yizhuo Zhai, **Yu Hao**, Hang Zhang, Daimeng Wang, Chengyu Song, Zhiyun Qian, Mohsen Lesani, Srikanth V. Krishnamurthy, Paul Yu
ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE 20. Tool

[J3] ConcSpectre: Be Aware of Forthcoming Malware Hidden in Concurrent Programs
Yang Liu, Ming Fan, Ting Liu, **Yu Hao**, Zisen Xu, Kai Chen, Hao Chen, and Yan Cai
IEEE Transactions on Reliability Code Result

[C3] ConcSpectre: Be Aware of Forthcoming Malware Hidden in Concurrent Programs
QRS 21 Yang Liu, Ming Fan, Ting Liu, **Yu Hao**, Zisen Xu, Kai Chen, Hao Chen, and Yan Cai
IEEE International Conference on Software Quality, Reliability, and Security, QRS 21. Code Result

[J2] Tell You a Definite Answer: Whether Your Data is Tainted During Thread Scheduling
TSE Xiaodong Zhang, Zijiang Yang, Qinghua Zheng, **Yu Hao**, Pei Liu, Ting Liu
IEEE Transactions on Software Engineering, ***TSE*** Tool Benchmarks Result

[J1] Debugging Multithreaded Programs as if They Were Sequential
Xiaodong Zhang, Zijiang Yang, Qinghua Zheng, **Yu Hao**, Pei Liu, Lechen Yu, Ting Liu
IEEE Access Tool

[C2] Automated Testing of Definition-Use Data Flow for Multithreaded Programs
ICST 17 Xiaodong Zhang, Zijiang Yang, Qinghua Zheng, Pei Liu, Jialiang Chang, **Yu Hao**, Ting Liu
IEEE International Conference on Software Testing, Verification and Validation, ICST 17. Tool

[C1] Debugging Multithreaded Programs as if They Were Sequential
SATE 17 Xiaodong Zhang, Zijiang Yang, Qinghua Zheng, **Yu Hao**, Pei Liu, Lechen Yu, Ming Fan, Ting Liu
IEEE International Conference on Software Analysis, Testing and Evolution, SATE 16. Tool

Thesis

[Master] Research on Malicious Code Hiding Methods Based on Uncertain Interleaving of Concurrent Programs
- [Proj1.3][C3, QRS 21, Best Paper Award][J3, *IEEE Transactions on Reliability*]

Patent

WO2017181628 - TAINT ANALYSIS METHOD EMPLOYING SYMBOLIC COMPUTATION AND USED FOR DYNAMIC PARALLEL PROGRAM [PCT] [CN]

Presentations, Talks

- E&V: Prompting Large Language Models to Perform Static Analysis by Pseudo-code Execution and Verification
  - *AGI Leap Summit 2024* Website
  - *Symposium on the Science of Security, HoTSoS 2024* Website
- SyzDescribe: Principled, Automated, Static Generation of Syscall Descriptions for Kernel Drivers
  - *Linux Security Summit North America 2023* Website Slides Recording
  - *Qualcomm Product Security Summit 2023* Website
  - *44th IEEE Symposium on Security and Privacy, S&P 2023* Website Recording
  - *Symposium on the Science of Security, HoTSoS 2024* Website
- Demystifying the Dependency Challenge in Kernel Fuzzing
  - *44rd IEEE/ACM International Conference on Software Engineering, ICSE 2022* Website Slides Recording

Honors, Awards

2024	Laxmi Bhuyan Fellowship Award, University of California, Riverside
	Dissertation Completion Fellowship Award, University of California, Riverside
2023	IEEE S&P Student Travel Grant
	Google Research Paper Rewards
	The Linux Foundation's Travel Fund
2021	QRS 2021 Best Paper Award
2018	Dean's Distinguished Fellowship, University of California, Riverside
	Excellent Graduated Student, Xi'an JiaoTong University
2017	Excellent Graduate Student, Xi'an JiaoTong University
	Graduate Student Fellowship, Xi'an JiaoTong University
	Sheng Han High IQ Association • Website IQ >= 135, Club at the same level as Mensa
2016	Graduate Student Fellowship, Xi'an JiaoTong University
2015	Graduate Student Fellowship, Xi'an JiaoTong University
2014	Siyuan Scholarship, Xi'an JiaoTong University
2013	Siyuan Scholarship, Xi'an JiaoTong University
	Second Prize, Xi'an JiaoTong University ACM-ICPC Competition
	Second Prize, Xi'an JiaoTong University MCM Competition
2010	First Prize, Mathematical Olympiad in Provinces, Inner Mongolia • Ranking 4th at that year in the province with population of 25 million
	Third Prize, Physics Olympiad in Provinces, Inner Mongolia
	Third Prize, Chemistry Olympiad in Provinces, Inner Mongolia
2007	First Prize, National Mathematics Invitational Competition, Inner Mongolia • A total of 4 at that year in the province with population of 25 million

Service

Reviewer	TIFS 2023, 2024. EAI SecureComm 2023.
Sub-Reviewer	S&P 2020, 2021, 2024. CCS 2024. USENIX Security 2021. NDSS 2020, 2021.
Program Committee	FORGE@ICSE 2024. MSR 2024(Junior PC).
AEC	EuroSys 2023. ISSTA 2024. ECOOP 2024.

References

Advisor of Ph.D. Degree	**Zhiyun Qian** *Everett and Imogene Ross Professor* Computer Science and Engineering department, University of California, Riverside, USA Contact: zhiyunq@cs.ucr.edu
Collaborator	**Ardalan Amiri Sani** *Associate Professor* Computer Science Department, University of California, Irvine, USA Contact: ardalan@uci.edu

Collaborator	**Srikanth V. Krishnamurthy** *Professor, IEEE Fellow* Computer Science and Engineering department, University of California, Riverside, USA Contact: krish@cs.ucr.edu
Intern Mentor at Microsoft Research	**Weidong Cui** *Partner Research Manager* Systems Security and Privacy Research Group, Microsoft Research, Redmond, USA Contact: Weidong.Cui@microsoft.com
Advisor of Master Degree	**Ting Liu** *Professor, Associate Dean* School of Cyber Science and Engineering, Xi'an Jiaotong University, China Contact: tingliu@mail.xjtu.edu.cn