CURRICULUM VITAE

Name: Yang Hu **Website:** huyang-utspark.github.io **Email:** huyang@utexas.edu

EDUCATION

The University of Texas at Austin, Austin, TX, USA **Sep. 2019 - Present**
Ph.D. student in the Department of Electrical and Computer Engineering
Research Interests: Software Security, Software Engineering, Automated Reasoning, Machine Learning
Supervisor: Prof. Mohit Tiwari

Xi'an Jiaotong University, Xi'an, Shaanxi, China **Sep. 2014 - Jun. 2017**
Master of Engineering in Software Engineering
Research Area: Security and Privacy
Supervisor: Prof. Jianming Chen
Graduate with honor (granted to top %5 students)

Xi'an Jiaotong University, Xi'an, Shaanxi, China **Sep. 2010 - Jun. 2014**
Bachelor of Engineering in Software Engineering
Postgraduate recommendation (granted to top 10% students)

PUBLICATIONS

1. Interactive Greybox Penetration Testing on Cloud Access Control with IAM Modeling and Deep Reinforcement Learning
Yang Hu*, Wenxi Wang*, Sarfraz Khurshid, Mohit Tiwari
Under Submission **[**arXiv preprint**]**

2. NeuroComb: Improving SAT Solving with Graph Neural Networks
Wenxi Wang, **Yang Hu**, Mohit Tiwari, Sarfraz Khurshid, Kenneth McMillan, Risto Miikkulainen
Under Submission [arXiv preprint**]**

3. Fixing Privilege Escalations in Cloud Access Control with MaxSAT and Graph Neural Networks
Yang Hu*, Wenxi Wang*, Sarfraz Khurshid, Kenneth McMillan, Mohit Tiwari
The 38th IEEE/ACM International Conference on Automated Software Engineering (ASE'23)
To Appear

4. SymMC: Approximate Model Enumeration and Counting Using Symmetry Information for Alloy Specifications
Wenxi Wang, **Yang Hu**, Kenneth McMillan, Sarfraz Khurshid
The 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'22) [Paper]

5. SapientML: Synthesizing Machine Learning Pipelines by Learning from Human-Written Solutions
Ripon Saha, Akira Ura, Sonal Mahaja, Chenguang Zhu, Linyi Li, **Yang Hu**, Hiroaki Yoshida, Sarfraz Khurshid, Mukul R. Prasad
The 44th International Conference on Software Engineering (ICSE'22) [Paper]

6. ACHyb: A Hybrid Analysis Approach to Detect Kernel Access Control Vulnerabilities
Yang Hu, Wenxi Wang, Casen Hunger, Riley Wood, Sarfraz Khurshid, Mohit Tiwari
The 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'21) [Paper]

7. Re-factoring based Program Repair applied to Programming Assignments
Yang Hu, Umair Z. Ahmed, Sergey Mechtaev, Ben Leong, Abhik Roychoudhury
The 34th IEEE/ACM International Conference on Automated Software Engineering (ASE'19) [Paper]

8. Exploiting Non-Uniform Program Execution Time to Evade Record/Replay Forensic Analysis
Yang Hu, Mingshen Sun, John C.S. Lui
Journal of Computers & Security. Elsevier, Apr. 2019. [Paper]

9. Taming Energy Cost of Disk Encryption Software on Data-Intensive Mobile Devices
Yang Hu, John C.S. Lui, Wenjun Hu, Xiaobo Ma, Jianfeng Li, Xiao Liang
Journal of Future Generation Computer Systems. Elsevier, Sep. 2017. [Paper]

SELECTED RESEARCH EXPERIENCE

1. Role: Applied Scientist Intern at Automated Reasoning Group, Amazon Web Services (AWS)

1.1 Cloud Access Control Policy Search via Large Language Models **May 2023 - Aug. 2023**

1.2 Hyperparameter Optimization for SMT Solvers **May 2022 - Aug. 2022**

2. Role: Ph.D. Candidate at The University of Texas at Austin

2.1 Interactive Greybox Penetration Testing on Cloud Access Control **Sep. 2021 - Present**

Overview:

Identity and Access Management (IAM) is an access control service in cloud platforms. To securely manage cloud resources, cloud customers are required to configure IAM to specify the access control rules for their cloud organizations. However, IAM misconfiguration may be exploited to perform privilege escalation attacks, which can cause severe economic loss for cloud customers. To detect privilege escalations due to IAM misconfigurations, existing third-party cloud security services apply whitebox penetration testing techniques with the requirement of the access of complete IAM configurations. This requirement causes several problems, such as information disclosure and anonymization. To mitigate the limitation of the whitebox penetration testing, we propose a precise greybox penetration testing approach called TAC which interacts with customers by selectively querying only the essential information needed.

Contributions:

- Proposed abstract IAM modeling, enabling TAC to detect IAM privilege escalations based on the partial information collected from queries.
- Minimized the interactions with customers by applying Reinforcement Learning with Graph Neural Networks, allowing TAC to learn to make as few queries as possible.
- Proposed an IAM privilege escalation task generator for evaluating the penetration testing tool.
- Results on our generated task set show that TAC detects IAM privilege escalations as precisely as state-of-the-art whitebox approaches using less than 100 queries for each privilege escalation task.
- This work is under submission.

2.2 Repairing Privilege Escalations in Cloud Access Control **Mar. 2023 - Present**

Overview:

As introduced, addressing IAM privilege escalations is crucial for improving security assurance for cloud customers. However, the area of repairing IAM privilege escalations due to IAM misconfigurations is relatively underexplored. To our knowledge, the only existing IAM repair tool called IAM-Deescalate focuses on a limited number of IAM privilege escalation patterns. To mitigate this limitation, we propose a novel IAM Privilege Escalation Repair Engine called IAMPERE that efficiently generates an approximately minimal patch for repairing a broader range of IAM privilege escalations.

Contributions:

- Proposed to formulate the repair problem as a MaxSAT problem, and apply Graph Neural Networks to improve the scalability of solving the formulated MaxSAT problem.
- Experimental results on the synthesized datasets demonstrate that IAMPERE can successfully repair 2.4x more IAM misconfigurations than the baseline IAM-Deescalate within a time limit of 600 seconds.
- This work was accepted by ASE’23.

2.3 Detecting Kernel Access Control Vulnerabilities Sep. 2019 - Aug. 2021

Overview:

Access control is essential for the Operating System (OS) security. Incorrect implementation of access control can introduce new attack surfaces to the OS, known as Kernel Access Control Vulnerabilities (KACVs). Our project aims to detect KACVs precisely and efficiently.

Contributions:

- Conducted an empirical study on KACVs to understand their security impacts and root causes.
- Proposed a precise, scalable hybrid analysis approach called ACHyb to detect KACVs due to missing or misusing permission checks.
- Experimental results show that ACHyb outperforms PeX, the state-of-the-art KACV detector, in terms of both the detection precision and the efficiency. Furthermore, ACHyb detects 7 new KACVs, 2 of which have been confirmed by the kernel developers.
- This work was accepted by ESEC/FSE’21.

2.4 Improving SAT Solving with Graph Neural Networks Sep. 2020 - Present

Overview:

Propositional satisfiability (SAT) is an NP-complete problem that impacts many research fields, such as planning, verification, and security. Mainstream modern SAT solvers are based on the Conflict-Driven Clause Learning (CDCL) algorithm. Recent work aimed to enhance CDCL SAT solvers by improving their variable branching heuristics through predictions generated by Graph Neural Networks (GNNs). However, so far this approach either has not made solving more effective, or has required online access to substantial GPU resources. This project aims to make GNN improvements practical in CDCL SAT solvers.

Contributions:

- Proposed an GNN-assisted SAT solving approach called NeuroComb based on two insights: (1) predictions of important variables and clauses can be combined with dynamic branching into a more effective hybrid branching strategy;(2) it is sufficient to query the GNN model only once for the predictions before the SAT solving starts.
- Results show that our approach allowed the MiniSat solver to solve 11% and the Glucose solver to solve 5% more problems on the recent SATCOMP-2021 competition problem set, with the computational resource requirement of only one GPU. Technical details have been summarized in our arXiv preprint.

2.5 Approximate Model Counting with Symmetry Information for Alloy Specifications Jan. 2020 - Nov. 2022

Overview:

Alloy is a mature tool-set that provides first-order relational logic for writing specifications, and a fully automatic powerful backend for analyzing the specifications. It has been widely applied in areas including verification, security, and synthesis. Symmetry breaking is a useful approach for pruning the search space to efficiently check the satisfiability of combinatorial problems. Alloy does the partial symmetry breaking for Alloy specifications. While full symmetry breaking remains challenging to scale, a recent study showed that Alloy partial symmetry breaking could significantly reduce the model counting time, albeit at the cost of producing only partial model counts. However, the desired term is either the isomorphic count under no symmetry breaking, or the non-isomorphic count under full symmetry breaking. This paper presents an approach called SymMC, which utilizes the symmetry information to compute both the desired counts for Alloy specifications.

Contributions:

- proposed our approximate model counting approach SymMC based on sampling to scalably estimate the counts.
- Proved that our proposed approximate counting algorithm has consistency and upper bound properties.
- Results show that SymMC outperforms two state-of-the-art model counters in over 77% problems.
- This work was accepted by ESEC/FSE'22.

3. Role: Research Intern at Fujitsu Research of America - AI Lab

Machine Learning Pipeline Synthesis for Automatic Machine Learning (AutoML) Jun. 2021 - Aug. 2021

Overview:

AutoML holds the promise of truly democratizing the use of machine learning by substantially automating the work of data scientists. However, the huge combinatorial search space of candidate pipelines means that current AutoML techniques generate sub-optimal pipelines, or none at all, especially on large, complex datasets. This project aims to learn from a corpus of existing datasets and their human-written pipelines to efficiently generate a high-quality pipeline for a predictive task on a new dataset.

Contributions:

- Presented a learning-based AutoML tool that can efficiently synthesize high-quality supervised ML pipelines.
- Evaluated the tool on 41 datasets. The results show that it outperforms SOTA tools in 27 datasets.
- This work was accepted by ICSE'22.

4. Role: Research Assistant at National University of Singapore

Automated Program Repair Jul. 2018 - Jul. 2019

Overview:

Our project aims at automatically repairing severely incorrect programs given at least one reference program. This is achieved by conducting software refactoring on reference programs to generate diverse correct programs, which are then used to facilitate block-level patch synthesis. Our approach has been applied to intelligent tutoring for programming education at National University of Singapore.

Contributions:

- Proposed a refactoring-based program repair approach that can fix severely incorrect programs.
- Developed a program repair engine for Python programs based on our approach.
- Evaluated our repair engine on one benchmark with 7290 buggy student programs. Our repair engine can fix 40% more buggy programs in 50% less average repair time, compared with the baseline tool Clara.
- This work was accepted by ASE'19.

5. Role: Research Assistant at The Chinese University of Hong Kong

Record and Replay Forensic Analysis Mar. 2016 - Oct. 2016

Overview:

Record and replay analysis is initially utilized in software debugging. It records the events that may affect program flow and reproduces the bug by replaying the same events. Record and replay analysis is now used in software forensics. However, the program under forensic analysis may use tricks to prevent malicious behavior from being replayed. Our goal is to explore the vulnerabilities which a program may exploit to hinder a faithful replay.

Contributions:

- Discovered one type of vulnerability associated with the phenomenon that program execution time cannot be exactly the same as the program re-execution time due to the non-uniform noise from hardware.
- Conducted proof-of-concept attacks on three platforms.
- This work was accepted by Journal of Security & Privacy

FUTURE RESEARCH DIRECTIONS

AI for Security:

- AI for Cloud Computing Security
    - *AI for Cloud Access Control Security:* use AI techniques to test and verify cloud access control, fix cloud access control vulnerabilities, and synthesize secure cloud access control policies.
    - *AI for Cloud Intrusion Detection and Prevention:* use AI techniques to identify unusual attack patterns targeting the cloud infrastructure, and proactively neutralize the threat.
- AI for Software Security
    - *AI for Fuzzing:* use machine learning techniques to improve the effectiveness and efficiency of fuzzing critical software systems such as operating system kernels and automated driving systems.
    - *AI for Software Vulnerability Repair*: use machine learning techniques to generate high-quality repairs for software vulnerabilities such as access control vulnerabilities.

Security for AI: utilize formal methods or software analysis techniques to facilitate the security analysis of AI systems.

RELEASED SOFTWARE

● *ACHyb: A Kernel Access Control Vulnerability Detector*	https://github.com/githubhuyang/achyb
● *Refactory: A Program Repair Tool for Python*	https://github.com/githubhuyang/refactory
● *Malicious URL Detector*	https://bitbucket.org/huyang0905/murl_detector

ACADEMIC SERVICE EXPERIENCE

● *Program Committee:* AAAI'24, ISSTA'23 AE, PLDI'23 AE, USENIX SEC'23 AE, ISSTA'22 AE	
● *Reviewer in Journal of Information and Software Technology*	Fall 2021

SELECTED AWARDS, HONORS & QUALIFICATIONS

Awards

● *Professional Development Award, UT Austin*	Fall 2019
● *Chen Qi Scholarship, Xi'an Jiaotong University (1%)*	2015-2016
● *Weizhizhu Scholarship, Weizhizhu Talent Pool (1%)*	2014-2015
● *Fuji Xerox Scholarship, Fuji Xerox Inc. (5%)*	2014-2015
● *Lu Shidi Scholarship, Xi'an Jiaotong University. (5%)*	2012-2013
● *Si Yuan Scholarship, Xi'an Jiaotong University (5%)*	2011-2012

Honors & Qualifications

● *Graduate with Honor from Xi'an Jiaotong University*	2017
● *IBM Academic Qualification in Software Testing and Evaluation*	2016
● *Silver Medal, ACM-ICPC Asia China Shaanxi Provincial Programming Contest*	2013
● *Postgraduate Recommendation, Xi'an Jiaotong University*	2013
● *Outstanding Student, Xi'an Jiaotong University*	2012

SKILLS

Programming Languages:	C, C++, Java, C#, Python, golang, etc

Tools and Frameworks

Machine Learning:	scikit-learn, PyTorch, PyTorch Geometric, Tensorflow, LangChain
Mobile Computing:	Android (framework and kernel level)
Distributed Computing:	Hadoop, Spark
Miscellaneous:	LaTeX, Vim, Git