Xiao (Lester) Yu

☎ +1 (609) 951 2988 • ✉ xiao@nec-labs.com • 🖱 lesteryu.com

Education

Doctor of Philosophy in Computer Science **July 2018**
North Carolina State University *Raleigh, NC, USA*
Advisor: Guoliang Jin

Master of Science in Computer Software & Theory **June 2011**
East China Normal University *Shanghai, China*
Advisor: Geguang Pu

Bachelor of Engineering in Software Engineering **June 2008**
East China Normal University *Shanghai, China*

Employment and Visiting Experience

NEC Laboratories America *Princeton, NJ, USA*
Researcher, Data Science & System Security Department *8/2018–Present*

NEC Laboratories America *Princeton, NJ, USA*
Research Intern, Autonomic Management Department *5/2014–12/2014*

University of Illinois at Urbana-Champaign *Urbana, IL, USA*
Visiting Student, Department of Computer Science *9/2013–5/2014*

U.S. Food and Drug Administration *Silver Spring, MD, USA*
Visiting Student, Center for Devices and Radiological Health *5/2013–8/2013*

Microsoft Research *Beijing, China*
Research Intern, Software Analytics Group *5/2012–8/2012*

Research Interests

Understanding and addressing reliability, performance, and security problems in complex software systems through a variety of techniques including trace/log analysis, program analysis, and advanced machine learning.

Research Experience

Natural Language Processing for Security
NEC Laboratories America *2020–Present*

- Realizing the need of automated processing and transformation of enormous amount of threat intelligence text data, identified key roadblocking challenges being highly domain-specific and low-resource that hinder the adoption of advanced natural language processing techniques in the cybersecurity area.
- To overcome the identified challenges, designed a pipeline to build security-specific language models with: (1) model adaptation that adapts pre-trained general language models to learn semantics and representations in security-specific text, (2) distant supervision that leverages external knowledge bases to automatically generate labeled security text data at the cost of high noise ratio, and (3) joint named entity and relation extraction that works with text data with noisy distant labels to extract structural information while handle label noises.

System Visibility Research

NEC Laboratories America *2020–Present*

- Investigated the use of whole-system provenance in building system and program models for threat modeling, and its challenges in provenance-graph verbosity and high-level system visibility.
- Designed machine learning algorithms to map low-level activities in provenance graphs built from system-call traces to high-level activities in form of user-program functions to reduce provenance-graph verbosity and improve high-level system visibility.
- Investigated fusion of heterogenous logs with machine learning techniques to embed log messages and fields into a uniform embedding space to support log analytics across logs from different sources.

Provenance-based Intrusion Detection

NEC Laboratories America *2018–2020*

- Investigated emerging cybersecurity threats in the software supply chain, such as attacks leveraging the software installation channel.
- Investigated whole-system provenance and its potential usage in detecting supply-chain threats.
- Designed both statistical model based and graph deep learning algorithms on the whole-system provenance to detect malicious software installations.
- Developed multiple proof-of-concept systems for (1) detection of malware drops, and (2) detection of malicious packages in Python package installations.

Understanding and Addressing Inefficiencies in Web-based Applications

North Carolina State University *2015–2018*

- Realizing the prevalence of web-based applications, investigated their performance inefficiencies related to two common aspects: the request-based execution model and frequent database accesses.
- On inefficiencies related to the request-based execution model,
  - Proposed a data-driven analysis approach combining program analysis and tracing to infer data dependencies between request-handler methods.
  - Investigated how the inferred data dependencies could enable inter-request analysis, which works across request-handler methods to identify potential inefficiencies, such as repetitive computations.
- On inefficiencies related to database accesses,
  - Conducted an empirical study on performance bugs related to database accesses with the goal of finding future research directions to address inefficiencies in database accesses.
  - Identified root causes and related factors on both the application side and the database side, summarized and explained how such bugs could be introduced, triggered, and fixed.

Log-based Monitoring for Cloud Infrastructures

NEC Laboratories America and North Carolina State University *2014–2015*

- Proposed a lightweight workflow monitoring approach on distributed and interleaved logs from complex task executions in order to address the management complexity in multi-tenant cloud infrastructures.
- Implemented and applied the approach on OpenStack to monitor and detect failures and performance problems in task executions.

Analyzing Execution Traces for Bottlenecks in Windows Kernel Drivers

Microsoft Research and North Carolina State University *2012–2013*

- From large-scale real-world execution traces, identified that system performance could be compromised by *cost propagation*, an adverse effect caused by synchronizations and component dependencies in kernel drivers and the Windows kernel.
- Proposed and developed a practical two-step approach with effective data and pattern abstractions to (1) measure performance impacts manifested through cost propagation, and (2) discover runtime behavioral patterns closely related to performance problems.

Dynamic Test Generation for C Programs

East China Normal University *2007–2011*

- Built an automated test generation tool for C programs with the goal of improving the efficiency and usability of dynamic symbolic execution.
- Designed practical algorithms as supplements to the linear constraint solver used in the test generation tool to improve the handling of complex data structures and pointers.
- Adopted the side-effect analysis that eliminates irrelevant path conditions, and a partial execution technique that reduces paths to explore in loop iterations, in order to relieve the path explosion problem.
- Designed parallel algorithms for dynamic symbolic execution based on partial orders in program paths.

Validation of Design Patterns in Object-Oriented Programs

East China Normal University *2009–2010*

- Designed a relational calculus and an object model to describe object-oriented design patterns.
- Implemented a tool to detect and validate the use of design patterns in Java program with the relational calculus and the object model.

Analysis and Verification of rCOS

East China Normal University *2007–2008*

- Extended rCOS, a component-based modeling system, to support assertions, invariants, and parallelism.
- Proposed an rCOS-to-SPIN transformation approach to enable model verification.
- Constructed operational semantics for the extended rCOS as a guide for the implementation of rCOS.

Publications

[1] Xueyuan Han, **Xiao Yu**, Thomas Pasquier, Ding Li, Junghwan Rhee, James Mickens, Margo Seltzer, and Haifeng Chen. SIGL: Securing Software Installations Through Deep Graph Learning. In *30th USENIX Security Symposium* (*USENIX Security '21*), August 2021.

[2] Peng Fei, Zhou Li, Zhiying Wang, **Xiao Yu**, Ding Li, and Kangkook Jee. SEAL: Storage-efficient Causality Analysis on Enterprise Logs with Query-friendly Compression. In *30th USENIX Security Symposium* (*USENIX Security '21*), August 2021.

[3] Tao Wang, **Xiao Yu**, Zhengyi Qiu, Guoliang Jin, and Frank Mueller. BARRIERFINDER: Recognizing Ad Hoc Barriers [Jounral version]. *Empirical Software Engineering*), 2020.

[4] Qi Wang, Wajih Ul Hassan, Ding Li, Kangkook Jee, **Xiao Yu**, Kexuan Zou, Junghwan Rhee, Zhengzhang Chen, Wei Cheng, Carl A. Gunter, and Haifeng Chen. You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis. In *27th Aannual Network and Distributed Systems Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020.

[5] Shudi Shao, Zhengyi Qiu, **Xiao Yu**, Wei Yang, Guoliang Jin, Tao Xie, and Xintao Wu. Database-Access Performance Antipatterns in Database-Backed Web Applications. In *2020 International Conference on Software Maintenance and Evolution (ICSME)*, September 2020.

[6] Kyungtae Kim, Chung Hwan Kim, Junghwan Rhee, **Xiao Yu**, Haifeng Chen, Dave (Jing) Tian, and Byoungyoung Lee. Vessels: Efficient and Scalable Deep Learning Prediction on Trusted Processors. In *ACM Symposium on Cloud Computing 2020 (SoCC'20)*, Octobor 2020.

[7] Wajih Ul Hassan, Ding Li, Kangkook Jee, **Xiao Yu**, Kexuan Zou, Dawei Wang, Zhengzhang Chen, Zhichun Li, Junghwan Rhee, Jiaping Gui, and Adam Bates. This is Why We Can't Cache Nice Things: Lightning-Fast Threat Hunting using Suspicion-Based Hierarchical Storage. In *Annual Computer Security Applications Conference 2020 (ACSAC '20)*, pages 165–178, December 2020.

[8] Jiaping Gui, Zhengzhang Chen, **Xiao Yu**, Cristian Lumezanu, and Haifeng Chen. Anomaly detection on web-user behaviors through deep learning. In *Security and Privacy in Communication Networks*, pages 467–473, Cham, 2020. Springer International Publishing.

[9] Tao Wang, **Xiao Yu**, Zhengyi Qiu, Guoliang Jin, and Frank Mueller. BARRIERFINDER: Recognizing Ad Hoc Barriers. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 323–327, September 2019.

[10] Shen Wang, Zhengzhang Chen, **Xiao Yu**, Ding Li, Jingchao Ni, Lu-An Tang, Jiaping Gui, Zhichun Li, Haifeng Chen, and Philip S. Yu. Heterogeneous Graph Matching Networks for Unknown Malware Detection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3762–3770. International Joint Conferences on Artificial Intelligence Organization, July 2019.

[11] Shen Wang, Zhengzhang Chen, Jingchao Ni, **Xiao Yu**, Zhichun Li, Haifeng Chen, and Philip S. Yu. Adversarial Defense Framework for Graph Neural Network. arXiv:1905.03679, 2019.

[12] **Xiao Yu** and Guoliang Jin. Dataflow Tunneling: Mining Inter-request Data Dependencies for Request-based Applications. In *Proceedings of the 40th ACM/IEEE International Conference on Software Engineering*, ICSE '18, pages 586–597, New York, NY, USA, 2018. ACM.

[13] **Xiao Yu**, Pallavi Joshi, Jianwu Xu, Guoliang Jin, Hui Zhang, and Guofei Jiang. CloudSeer: Workflow Monitoring of Cloud Infrastructures via Interleaved Logs. In *Proceedings of the 21st International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '16, pages 489–502, New York, NY, USA, 2016. ACM.

[14] **Xiao Yu**, Shi Han, Dongmei Zhang, and Tao Xie. Comprehending Performance from Real-World Execution Traces: A Device-Driver Case. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '14, pages 193–206, New York, NY, USA, 2014. ACM.

[15] **Xiao Yu**, Shuai Sun, Geguang Pu, Siyuan Jiang, and Zheng Wang. A Parallel Approach to Concolic Testing with Low-cost Synchronization. *Electronic Notes in Theoretical Computer Science*, 274:83 – 96, 2011.

[16] Kang Miao, Siyuan Jiang, **Xiao Yu**, and Ji Zhao. Run-time Discovery of Java Design Patterns. In *the 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce*, AIMSEC '11, pages 3329–3332, 2011.

[17] Kang Miao, **Xiao Yu**, Ji Zhao, and Yu Shen. Java design pattern recognition based on relational calculus. *Application Research of Computers (in Chinese)*, 27(9):3425–3430, 1 2010.

[18] Libo Feng, **Xiao Yu**, Geguang Pu, Siyuan Jiang, Huibiao Zhu, and Bing Gu. Property Checking for Design Patterns. In *Proceedings of the IASTED International Conference on Software Engineering*, SE '10, pages 87–94. ACTA Press, 2010.

[19] Zheng Wang, **Xiao Yu**, Tao Sun, Geguang Pu, Zuohua Ding, and JueLiang Hu. Test Data Generation for Derived Types in C Program. In *the Third IEEE International Symposium on Theoretical Aspects of Software Engineering*, TASE '09, pages 155–162, 2009.

[20] Tao Sun, Zheng Wang, Geguang Pu, **Xiao Yu**, Zongyan Qiu, and Bing Gu. Towards Scalable Compositional Test Generation. In *the Ninth International Conference on Quality Software*, QSIC '09, pages 353–358, 2009.

[21] Zheng Wang, **Xiao Yu**, Geguang Pu, Libo Feng, Huibiao Zhu, and Jifeng He. Execution Semantics for rCOS. In *Proceedings of the 15th Asia-Pacific Software Engineering Conference*, APSEC '08, pages 119–126, 2008.

[22] **Xiao Yu**, Zheng Wang, Geguang Pu, Dingding Mao, and Jing Liu. The Verification of rCOS Using Spin. *Electronic Notes in Theoretical Computer Science*, 207:49 – 67, 2008.

Patents

Securing Software Installation Through Deep Graph Learning
__Xiao Yu__, Xueyuan Han, Ding Li, Junghwan Rhee, and Haifeng Chen *Pending*

Provenance-based Threat Detection Tools and Stealthy Malware Detection
Ding Li, __Xiao Yu__, Junghwan Rhee, Haifeng Chen, and Qi Wang *Pending*

Efficient and Scalable Enclave Protection for Machine Learning Programs
Chung Hwan Kim, Junghwan Rhee, __Xiao Yu__, LuAn Tang, Haifeng Chen, and Kyungtae Kim *Pending*

Real-time Threat Alert Forensic Analysis
Ding Li, Kangkook Jee, Zhichun Li, Zhengzhang Chen, and __Xiao Yu__ *Pending*

CloudSeer: Using Logs to Detect Errors in the Cloud Infrastructure
Pallavi Joshi, Hui Zhang, Jianwu Xu, __Xiao Yu__, and Guofei Jiang *US9720753, 8/1/2017*

Mentoring

Yufei Li *The University of Texas at Dallas*
Research Assistant @ NEC Laboratories America *Summer 2021*

Fei Zuo *University of South Carolina*
Research Assistant @ NEC Laboratories America *Summer 2020*

Xueyuan (Michael) Han *Harvard University*
Research Assistant @ NEC Laboratories America *Summer - Fall 2019*

Teaching Experience

Teaching Assistant
North Carolina State University

- **CSC 501: Operating Systems Principles** *Spring 2017, Fall 2017*
  Lecturers: Xiaohui (Helen) Gu and Guoliang Jin
- **CSC 568: Enterprise Storage Architecture** *Spring 2015*
  Lecturer: Vincent W. Freeh
- **CSC 506: Architecture of Parallel Computers** *Spring 2012, Spring 2015*
  Lecturer: Edward F. Gehringer
- **CSC 510: Software Engineering** *Fall 2011*
  Lecturer: Annie I. Antón

Conference Presentations and Invited Talks

Finding Troublemakers in Complex Software Systems: A Data-Driven Perspective
University of Memphis, Memphis, TN, USA (*March 12, 2018*)

Understanding and Debugging Interconnected Software Systems via Data-Driven Analysis
NEC Laboratories America, Princeton, NJ, USA (*March 5, 2018*)

CloudSeer: Workflow Monitoring of Cloud Infrastructures via Interleaved Logs
ASPLOS 2016, Atlanta, GA, USA

Comprehending Performance from Real-World Execution Traces: A Device-Driver Case
ASPLOS 2014, Salt Lake City, UT, USA

Professional Activities

Program Committee Member: ASPLOS 2018 (Shadow), TaPP 2020, TaPP 2021

Reviewer: Empirical Software Engineering (Special Issue on Automatic Software Repair), IEEE Transactions on Big Data, ACM Transactions on Data Science

Subreviewer: ISSTA 2012 & 2013, ICSM 2012, OOPSLA 2013, MSR 2014, ICST 2014, ASE 2014

Student Volunteer: FSE 2012

References

References are available upon request.