Qiuchen Yan

https://www-users.cs.umn.edu/~yanxx297/ Email: yanxx297@umn.edu
https://github.com/yanxx297 Mobile : +1-651-235-4138

EDUCATION

University of Minnesota, Twin Cities Minneapolis, MN
Ph.D. in Computer Science May 2014 – 2020 (anticipated)
Master of Science in Computer Science Sep. 2012 – May 2014

Shandong University of Science and Technology Qingdao, China
Bachelor of Engineering in Computer Science Sep. 2008 – July 2012

SKILLS

Programming Languages: C/C++, Python, OCaml, Java, X86 assembly
Systems & Tools: Linux, FuzzBALL, Xed (Intel Pin), DWARF, Vine

RESEARCH PROJECTS

Testing Emulators Using Symbolic Execution 2018 - present

- To test the correctness of QEMU, explore it and other emulators with FuzzBALL (a symbolic execution platform written in OCaml), and perform triangle tests base on the out-coming expressions

Loop Summarization for Symbolic Execution 2014 - 2015, 2018 - present

- As a countermeasure of the path explosion problem, design a extended version of a trace-based loop summarization algorithm[1] and implement it on FuzzBALL.
- Evaluate this work with competition binaries from DARPA Cyber Grand Challenge

Fast & Automatic Emulator Testing System 2015 - 2018

- Speed up an automatic emulator testing tool by designing and implementing a novel approach to generate test cases.
- Implement an x86 assembly test case generator based on the previous work. Thegenerator is mostly written in Python. Also modified other components of the testing system written in C++.

Binary Level Type Inference 2013 – 2014

- Design a static type inference tool that can infer the signedness of variables in binaries with 96% true positive.
- Build this tool on top of Vine and libdwarf using C++.

PUBLICATION

Qiuchen Yan, Stephen McCamant,“Fast PokeEMU: Scaling Generated Instruction Tests Using Aggregation and State Chaining,” The 14th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE’18)

1 Patrice Godefroid and Daniel Luchaup. Automatic partial loop summarization indynamic test generation. InProceedings of the 2011 International Symposium onSoftware Testing and Analysis, ISSTA ’11, pages 23–33, New York, NY, USA, 2011.ACM.

Qiuchen Yan, Stephen McCamant, "Conservative Signed/Unsigned Type Inference for Binaries using Minimum Cut," Technical report

EXPERIENCE

Graduate Research Asistant, University of Minnesota 2014 – present
Work with Stephen McCamant on several research projects. Collaborate with Pen-Chung Yew's dynamic binary translation group on projects related to emulator testing.

DARPA Cyber Grand Challenge 2014 – 2015
Contribute vulnerability checking code for the FuzzBOMB group in CGC Qualification Event.

ACADEMIC PROJECTS

Reproduce the Lucky Thirteen attack 2014
Implement a timing side channel attack[2] to the TLS protocol. Course project.

Sybil attack study 2014
Survey about the Sybil attack in online social network and its state-of-art defence approach and collected data from real world sybil communities in sina weibo. Course project.

Encrypted address book for Android 2012
Design and implement an Android address book app that can send encrypted contact info via text message. Bachelor final project.

SERVICE

- Contribute code to FuzzBALL, an open source symbolic execution tool.
- Present my work on The 14th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE'18)
- Give guest lectures on security related courses at the University of Minnesota.

COURSEWORK

Introduction to Computer Security - A breadth of knowledge about software security and network security
Modern Cryptography - Introduction to widely used cryptography theories and algorithms
Machine Learning - Introduction to machine learning
Security and Privacy in Computing - A seminar discussing recent papers about security,privacy and cryptography

2 Nadhem J. Al Fardan and Kenneth G. Paterson. Lucky thirteen: Breaking the tls and dtls record protocols. In Proceedings of the 2013 IEEE Symposium on Security and Privacy, SP '13, pages 526–540, Washington, DC, USA, 2013. IEEE Computer Society.