Yaohui Chen

Facebook
1 Hacker Way
Menlo Park, CA, 94025

E-mail: yaohway@gmail.com

Education

2017 - 2019 **Ph.D.**, *College of Computer and Information Science* , Northeastern University, Boston, MA, Advisor: Prof. Long Lu.

2014 - 2017 **Ph.D.**, *Department of Computer Science* , Stony Brook University (transferred to Northeastern University), Stony Brook, NY, Advisor: Prof. Long Lu.
GPA:3.90/4.00

2013 - 2013 **Exchange Student**, Polytechnic University of Catalonia, Barcelona, Spain.

2010 - 2014 **B.Sc.**, *Department of Computer Science*, Tongji University, Shanghai, China.

Research Interests

My research interests mainly include **Bug Finding Automation, Program Analysis, Binary Exploit Mitigations, Virtualization, Operating System and Mobile Security.**

Experiences

Dec. 2019 - present **Research Scientist**, *Infrastructure Security Team*, Facebook.

May. 2017 - Oct. 2019 **Research Assistant**, *RiS3 Lab*, Northeastern University.

Sep. 2018 - Dec. 2018 **Research Intern**, *Software Analysis Team*, Google.

June. 2018 - Aug. 2018 **Research Intern**, *System Security and Privacy Research*, Microsoft Research.

Feb. 2018 - May. 2018 **Research Intern**, *X-Lab*, Baidu USA.

May. 2017 - Sep. 2017 **Research Intern**, *B2B Lab*, Samsung Research America.

May. 2016 - Sep. 2016 **Research Intern**, *B2B Lab*, Samsung Research America.

Aug. 2014 - May. 2017 **Research Assistant**, *RiS3 Lab*, Stony Brook University.

Publications

SAVIOR: Towards Bug-Driven Hybrid Testing, ***Yaohui Chen***, *Peng Li, Jun Xu, Shengjian Guo, Rundong Zhou, Yulong Zhang, Taowei, Long Lu*, In the proceeding of the 41th IEEE Symposium on Security and Privacy (Oakland'20).

FUDGE: Fuzz Driver Generation at Scale, *Domagoj Babic, Stefan Bucur,* ***Yaohui Chen****, Franjo Ivancic, Tim King, Markus Kusano, Caroline Lemieux, Laszlo Szekeres and Wei Wang*, In the proceeding of The ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'19), (*Best Paper Award*).

Ptrix: Efficient Hardware-Assisted Fuzzing for COTS Binary, ***Yaohui Chen***, *Dongliang Mu, Zhichuang Sun, Jun Xu, Wenguo Shen, Xinyu Xing, Long Lu, Bing Mao*, In Proceedings of the 14th ACM on Asia Conference on Computer and Communications Security (AsiaCCS'19).

Compiler-assisted Code Randomization, *Hyungjoon Koo,* ***Yaohui Chen****, Long Lu, Vasileios P. Kemerlis, Michalis Polychronakis*, In the proceeding of the 39th IEEE Symposium on Security and Privacy (Oakland'18), (*Final list, CSAW 2018*).

InstaGuard: Instantly Deployable Hot-patches for Vulnerable System Programs on Android, ***Yaohui Chen***, *Yuping Li, Long Lu, Yueh-Hsun Lin, Hayawardh Vijayakumar, Zhi Wang, Xinming Ou*, In Proceedings of the 2018 Network and Distributed System Security Symposium (NDSS'18).

Norax: Enabling Execute-Only Memory for COTS Binaries on AArch64, ***Yaohui Chen***, *Dongli Zhang, Ruowen Wang, Rui Qiao, Ahmed Azab, Long Lu, Hayawardh Vijayakumar, Wenbo Shen*, In the proceeding of the 38th IEEE Symposium on Security and Privacy (Oakland'17), (*Third Place of Best Paper Award in Applied Cyber Security Research, CSAW 2017*).

Secure Integration of Web Content and Applications on Commodity Mobile Operating Systems, *Drew Davidson,* ***Yaohui Chen****, Franklin George, Long Lu, Somesh Jha*, In Proceedings of the 12th ACM on Asia Conference on Computer and Communications Security (AsiaCCS'17).

Shreds: Fine-grained Execution Units with Private Memory, ***Yaohui Chen***, *Sebassujeen Reymondjohnson, Zhichuang Sun, Long Lu*, In the proceeding of the 37th IEEE Symposium on Security and Privacy (Oakland'16).

Research Projects

June. 2018 - Aug. 2018 **Hyper-V hypervisor coverage feedback-guided fuzz testing**.

- Studied the existing automatic testing solutions for Hyper-V hypervisor.
- Designed and implemented code coverage guided fuzzer for Hyper-V hypervisor.
- The end-to-end solution involved Intel-PT virtualization support for hardware tracing.
- Discovered 3 real bugs in the newest Hyper-V hypervisor and they have been fixed.
- The solution has been adopted into Hyper-V hypervisor testing process before release.

May. 2017 - present **Harnessing Efficiency of Fuzzing and Soundness of Symbolic Execution**.

- Studied the inefficiency of symbolic execution and unsoundness of fuzzing.
- Modified KLEE to support on-demand symbolic execution.
- Developed light-weight static analysis to guide symbolic execution.
- Developed infrastructure to coordinate KLEE, AFL and static analysis components.
- Evaluated the framework on real-world programs including ARM TrustZone TAs, commonly fuzzed applications and showed it out-performed the state-of-the-art.

Services

2016 - Present **IET Information Security**, *Journal Reviewer.*

2016 **Transactions on Dependable and Secure Computing**, *External Reviewer.*

2020 **The 29th USENIX Security Symposium (Security)**, *External Reviewer.*

2020 - 2019 **The Network and Distributed System Security Symposium (NDSS)**, *External Reviewer.*

2020 **IEEE Symposium on Security and Privacy (S&P)**, *External Reviewer.*

2019 **ACM Symposium on Information, Computer and Communications Security (ASIACCS)**, *External Reviewer.*

2018 **ACM Conference on Computer and Communications Security (CCS)**, *External Reviewer.*

2018 **IEEE Secure Development Conference (SecDev)**, *External Reviewer.*

Honors and Awards

2019 **Best Paper Award**, ACM ESEC/FSE.

2019 **Google Fellowship Nomination**, Northeastern Univerisity.

2018 **RSA Scholarship**, *RSA Conference 2018*, U.S.A.

2017 **Best Paper Award**, *CSAW*, U.S.A.

2014 **Chair Fellowship**, *Department of Computer Science*, Stony Brook Univeristy.

2013 **Exchange Student Scholarship**, *Department of Computer Science*, Tongji Univerisity.

2012 **Second-Class Scholarship**, *Department of Computer Science*, Tongji Univerisity.

Professional Skills

OS **Linux, Android, Windows**.

Programming **C, C++, Java, Python, System-verilog, ARM/x86/Sparc/MIPS assembly**.

Compiler **Clang, LLVM**.

Misc **Kubernetes, Reverse Engineering, Program Analysis, ELF Binary Linking Loading Toolchain**.