Chen Peng 陈鹏

ShanghaiTech University
School of Information Science and Technology

Phone: +86 185 1655 7902
Email: spinpx@gmail.com
Homepage: http://spinpx.com/

Education

M.S., Computer Science, ShanghaiTech University — Sep 2015 – Jul 2018

Exchange student, Computer Science, National Chiao Tung University — Feb 2013 – Jun 2013

B.S., Computer Science, Harbin Institute of Technology — Sep 2011 – Jul 2015

Experience

Intern at Compound Search Department, Baidu. — Jun 2014 – Sep 2014

TA Data Structure course (Outstanding TA 2015 award) — 2015

Publications

[1] Peng Chen and Hao Chen. "Angora: efficient fuzzing by principled search". In: *IEEE Symposium on Security and Privacy (IEEE S&P)*. San Francisco, CA, May 21–23, 2018.

[2] Peng Chen and Hao Chen. "Security Analysis of Personal Unmanned Aerial Vehicles". In: *International Conference on Security and Privacy in Communication Networks (SECURECOMM)*. Guangzhou, China, Oct. 10–12, 2016.

Projects

Security Analysis of Unmanned Aerial Vehicles — Mar 2015 – Sep 2015
We studied the risks of UAVs and conducted an empirical analysis of three popular DJI UAVs. We discovered a series of vulnerabilities, including insecure communication channels, misuse of cryptography, and insecure UAV activation and developer authorization. By exploiting these vulnerabilities, we designed a series of attacks for UAV.
Related skills: TCP/UDP, Android Security

Source Code Author Deanonymization — Nov 2015 – Jan 2016
In this work, we explored neural network based approaches (Recurrent neural network) towards the source code author deanonymization problem. With a dataset extracted from Google Code Jam, our char-level model performs competitively on normal size dataset comparing to previous state-of-art work.
Related skills: Crawler, Lua, Torch, Deep learning

Angora: Efficient, Coverage-Directed Fuzzing — Oct 2016 – Dec 2017
We propose Angora, a new mutation-based fuzzer that outperforms the state-of-the-art fuzzers by a wide margin. The main goal of Angora is to increase branch coverage by solving path constraints without symbolic execution. To solve path constraints efficiently, we introduce several key techniques: scalable byte-level taint tracking, context-sensitive branch count, search based on gradient descent, and input length exploration.
Related skills: C/C++, LLVM, IPC, Taint Tracking, Rust, Intel Pin, Machine learning