Song Liu

✉ vancirprince@gmail.com
⌂ vancir.github.io
✆ (+1) 8143216052

Education

The Pennsylvania State University, College of Information Sciences and Technology — PA, USA
Doctor of Philosophy — *Aug 2022 - Now*

Research Keywords: Software Security, DBMS Fuzzing, Directed Fuzzing

Xiamen University, School of Informatics — Xiamen, China
Bachelor of Engineering — *Sep 2015 - Jun 2019*

Relevant Coursework: Algorithms, Principles of Operating Systems, Principles of Compilers

Publications

- VIPER: Spotting Syscall-Guard Variables for Data-Only Attacks (link), **USENIX 2023**
- Can We Trust the Phone Vendors? Comprehensive Security Measurements on the Android Firmware Ecosystem, **TSE 2023**
- Detecting Logical Bugs of DBMS with Coverage-based Guidance (link), **USENIX 2022**
- Large-scale Security Measurements on the Android Firmware Ecosystem (link), **ICSE 2022**

Work

QI-ANXIN Technology Research Institute — Beijing, China
Research and Development Engineer, Supervisor: ***Lingyun Ying*** — *Aug 2019 - Aug 2022*

- **MacOS Sandbox for Malware Analysis**: Designed and implemented the **first** macOS sandbox system in China. For macOS versions below 10.15, I read the source code of **XNU kernel**, used the Mandatory Access Control Framework(**MACF**) to monitor **process behavior** and **file operations**, wrote Network Kernel Extension(**NKE**) to capture **network traffic and behavior** respectively, and use threaded to send the monitoring data from kernel mode to user mode, and wrote a client at user mode to receive the behavior data from kernel mode. For macOS versions 10.15 and above, the previous mechanism was deprecated, so I developed a new system using the EndPoint Security Framework(**ESF**) and Network Extension(**NE**) for monitoring and capturing behavioral data. **Apple Script** was used to write a dynamic traversal tool for macOS apps to simulate user interaction and trigger malicious behavior.
- **Static and Dynamic Analysis Tool for Android Apps**: Developed a static analysis tool based on **AndroGuard**, with the same analysis capability as **VirusTotal**. Combined the depth-first search(**DFS**) algorithm and breadth-first search(**BFS**) algorithm to traverse the UI of Android App.
- **Continuous Fuzzing Platform**: A **fuzzing infrastructure** is used to schedule and allocate resources, perform **continuous** and **large-scale** fuzzing of target software, and obtain various monitoring data and performance metrics. Various frameworks and middleware are used, such as MongoDB, InfluxDB, Fluentd, Redis, Amazon S3. Integrated multiple **fuzzing engines**, and supports fuzzing multiple targets on multiple operating system platforms.
- **TianWen: Dependency Analysis Platform for Software Supply Chain**: Developed a crawler to continuously crawl software binaries. Built and deployed **graph database cluster**. Optimized database query performance to support querying **extremely complex** graph information in seconds, and wrote tools to visualize large amounts of graph data.
- **CVE Information Extraction Tool**: Based on Named Entity Recognition(**NER**) theory, we use **BiLSTM-CRF** model to extract the **vulnerability function**, **vulnerability version** and **vulnerability source path** from the unstructured official description information of CVE vulnerabilities, with an accuracy rate of about **88%**.
- **pySnoopSnitch: Android Firmware Patch Existence Detection Tool**: Rewrote the **SnoopSnitch** project code in Python. Supports checking the presence of patches in Android firmware using the **full core performance** of the server and generating a heat map of patch misses.

Institute of Information Engineering, Chinese Academy of Sciences — Beijing, China
Research Intern, Supervisor: ***Feng Li*** — *Jul 2018 - Sep 2018*

- **IO2BO Vulnerability Automatic Detection**: Performed the **infra-procedural** analysis and **inter-procedural** analysis on the source code based on **Klee**. Modified the Klee to read target files as input for **concolic execution**. When executing, add constraints to the state that may have vulnerabilities and pass the symbolic expression to the **constraint solver**, and determine whether there is an Integer-Overflow-to-Buffer-Overflow(**IO2BO**) vulnerability based on the result.

Projects

- **CTF Wiki**: The CTF Wiki is an open source knowledge base about CTF competitions. I contributed most of the content of the **reverse engineering** chapter of the Wiki (github, 7.1k stars)
- **Awesome-Binary-Similarity**: An awesome list of binary code similarity papers (github, 359 stars)
- **Awesome-Binary-Rewriting**: An awesome list of binary rewriting papers (github, 172 stars)
- **PyPi-Typosquatting-Graph**: Analyzing and visualizing **typosquatting** for Python packages hosted on PyPi.org. I obtained all python package names and created a **trie tree**, then calculated the **edit distance** between the names and used a **force-directed algorithm** to draw the graph (github)
- **Fugitive Consortium**: A blockchain platform based on **Hyperledger Fabric**. It is written in Java and starts a network topology consisting of docker containers. Chaincode is written to add, delete, query and change information in the ledger (github)