Eric **Liang**

☎ (+1) 646-525-9401 | ✉ ericzhiliang@gmail.com

Summary

Currently a Vulnerabililty Researcher at Raytheon CODEX. 3+ years experience specializing in 0-day research for remote code execution in high value targets on modern systems with full mitigations. Interested in working on difficult problems, learning about new research to further vulnerability discovery, and developing new tools to aid in said difficult problems.

Education

NYU Tandon School of Engineering

Brooklyn, New York

B.S. in Computer Science; GPA: 3.26

2012 - 2016

- Minor in cybersecurity

Work Experience

Raytheon CODEX

Annapolis Junction, Maryland

Senior Cyber Engineer I

Jan. 2017 - Present

- Structured Fuzzing W/ Coverage
  - Structured fuzzing with coverage utilizing protobuf and libfuzzer to find memory corruption bugs. Fuzzing harnesses were created to recreate realistic behavior in various remote attack surfaces. Auditing was done in tandem with harness creation to statically find bugs, understand program design, and model various program states.
- Grammar Based Fuzzing
  - Developed and researched fuzzers that generated syntatically and semantically valid grammar to explore complex program states in target that takes a programming language as input. Modified popular grammar based fuzzer with compiler phases to have type aware grammar generation, and type aware arguments for procedure calls.
- Code Auditing
  - Audited through modern C++ to look for potential bugs in target application with complex program states. Research was done to understand various APIs that can change said program states and understand various esoteric metalanguages in the codebase.
- N-Day Analysis
  - Conducted research and created reports for multiple vulnerabilities. Analyzed usability of vulnerabilities for necessary context required for usage.

Raytheon CODEX

Annapolis Junction, Maryland

Vulnerability Research Intern

May 2015 - Aug. 2015

- Browser N-Day Research
  - Developed and demonstrated a patched remote code execution utilizing a use-after-free bug in Internet Explorer 9's Document Object Model. Development required grooming of the Window's Low Fragmentation Heap, a heap spray (32-bit address space at the time), and return orientated programming to pivot to shellcode execution.
- Coverage Guided Feedback Tool
  - Created an intel pintool for coverage-guided feedback on a target application. Work was done to record all basic blocks reached by user input and aggregate coverage results.

Skills

Languages	Python, C++14, C, Javascript, Bash
Architectures	x86, x86-64, ARM, AArch64
Skills	Reverse Enginnering, Code Auditing, Shellcoding, Remote Code Execution, Sandbox Escape, IPC, Return Oriented Programming, Heapspray, JIT, Heap Grooms, Exploit Development
Systems	Linux, Android, Windows, Mac

Miscellaneous

2018 **DEFCON CTF Finalist**, Spaceballs/Nasa Rejects *Las Vegas, Nevada*