Ruilin Wang

Edmonton, AB | 587-589-0128 | ruilin.wang517@gmail.com | linkedin.com/in/ruilin-wang001 | github.com/RWuilin
Authorized to work in Canada on an open PGWP valid until Oct 2029

SOFTWARE DEVELOPER

PROFILE

Computer Science graduate from the University of Alberta and Master of Information Technology candidate at UNSW with hands-on experience in Python, Java, software testing, data processing, security testing, and research software development. Built projects involving data validation, black-box fuzzing, protocol simulation, and computer vision pipelines.

CORE SKILLS

Languages: Python, Java, TypeScript, SQL, C, Shell
Development: OOP, Modular Design, CLI Tools, REST APIs, Debugging, Testing, Input Validation
Data / ML: Pandas, NumPy, Matplotlib, OpenCV, PyTorch, TensorFlow, scikit-learn
Tools / Security: Git, GitHub, Docker, Linux, Firebase, Android Studio, SQLite, Fuzzing, Secure Coding
Spoken Languages: English (C1 – Advanced), Mandarin Chinese (Native)

EDUCATION

Master of Information Technology, Cyber Security Specialization | University of New South Wales, Sydney, Australia | Expected Aug 2026
Bachelor of Science in Computer Science, Minor in Mathematics | University of Alberta, Edmonton, Canada | Dec 2024

PROJECT EXPERIENCE

OpenET 2: Remote Eye-Tracking Data Benchmarking Tools | **Client-Based Capstone Project** | Python, Pandas, NumPy, Matplotlib, Data Validation, Visualization
UNSW | Client: Dr Peter Wagner | 2026

- Contributing to a client-based research software project extending OpenET, a Python framework for validating, visualizing, and benchmarking remote eye-tracking device data.
- Designed data quality checks for missing samples, duplicate timestamps, irregular sampling intervals, invalid gaze coordinates, incomplete recordings, and inconsistent metadata.
- Developed visualization and documentation workflows for gaze traces, fixation stability, missing-data timelines, sampling-frequency plots, and user/developer guides.

Format-Aware Black-Box Fuzzer | **Python, Docker, Shell, Linux, Security Testing**
GitHub: github.com/RWuilin/simplefuzzer

- Developed a Dockerized Python fuzzer for stdin-driven binaries using mutation-based testing to identify crashes, hangs, slow paths, and unusual program behaviours.
- Implemented format-aware mutation strategies for CSV, JSON, XML, JPEG, and plaintext inputs, with timeout handling and reproducible crash/hang input saving.
- Added output-signature tracking as a lightweight coverage approximation to preserve inputs that triggered new behaviors.

DIMY Contact Tracing Protocol Simulator | **Python, TCP/UDP Networking, Threads, Bloom Filters, Shamir Secret Sharing**
GitHub: github.com/RWuilin/dimy-contact-tracing-simulator

- Built a Python simulator for a privacy-preserving contact tracing protocol with frontend nodes, a backend TCP server, and a passive attacker model.
- Implemented rotating ephemeral identifiers, Shamir Secret Sharing, Bloom filter encounter storage, UDP broadcasting, and TCP request-response communication.
- Used threading, locks, and background loops to coordinate node behaviour, server upload/query operations, and interactive commands.