Proposed Topic/Title of Research:

Machine-learning-guide typestate analysis for Detecting UAF vulnerabilities

Yujian Gan

Background:

Zero-day Use-After-Free (UAF) vulnerabilities are increasingly popular and highly dangerous, but few mitigations exist. UAF vulnerabilities, i.e., dangling pointer dereferences in C/C++ programs can cause data corruption, information leaks, denial-of-service attacks via program crashes, and control-flow hijacking attacks. While other
memory corruption bugs, such as buffer overflows, have become harder to exploit due to various mitigation techniques, UAF has recently become a significantly more important target for exploitation.

Typestate analysis represents a fundamental approach for detecting statically temporal memory safety errors, such as use-after-free, in C/C++ programs. However, typestate analysis relies on precise pointer analysis for accurate software vulnerabilities detection.

Aims:

To improve the accuracy of UAF vulnerabilities detection and overcome the above limitations, our approach is to design a static analysis approach based on machine learning that bridges the gap between the existing typestate and pointer analyses by capturing the correlations between program features and complicated aliases that are answered conservatively by the state-of-the-art pointer analysis. The proposed project aims to learn and predict complicated likelihood software bugs by steering typestate analysis using Support Vector Machine (SVM) or TensorFlow.

Approach:

The proposed project will achieve the above goal through transfer learning based method. The size of data is an important factor, which significantly affects the accuracy of the detection results. However, the UAF sample data is limited in real scenarios, especially for real-world large programs. Thus, it is not enough for model training. One of the solutions is to use other program languages, such as C#, Java, because UAF in C/C++ is similar to object dispose exception in C#. After data collection and feature extraction, it can use transfer-learning model which is trained by some other languages to detect UAF in C/C++.

In addition, there are some other relevant approaches, such as UAFPrediction(a tool uses the SVM machine learning model to determine the likelihood of a use-after-free bug within C/C++ source files) developed by Dr. Yulei's research team. It will be more efficient to finish the project by leveraging the existing research work from Yulei's team.

Expected Outcomes:

The expected outcomes of the project are 1) an open-source tool for automatically detecting UAF vulnerabilities based on machine learning; 2) high quality publications in the area of software engineering and artificial intelligence.