Critique on
Suman Jana, Yuan Jochen Kang, Samuel Roth, and Baishakhi Ray. "Automatically Detecting Error Handling Bugs Using Error Specifications."

Astha Sharma
asharma6@uno.edu

SUMMARY

The paper introduces a tool called EPEx that is designed for detecting error handling mechanism or bugs in programs written in C. The core idea of the tool is to check the validity of the return type obtained from a called function. The return value must be a positive 0 or 1 but if there's a bug and it returns a negative integer as the return type, then the function is not really doing the job as described. Meaning, the execution is not properly handled and there is a bug in the system. In the paper, the authors have given few examples of some renown libraries that failed the test. Also, the tool here identifies the error paths by performing under constrained symbolic execution at the caller function. It further locates and pinpoints the error.

The EPEx tool was used to find the error handling bugs in different open-source SSL/TLS libraries and applications.

STRENGTHS

1. Security error bugs are difficult to find in C as there are no builtin error handling mechanism. This paper helps get a solution to this problem.
2. The EPEx tool efficiently finds the error handling bugs with 78% precision rate.
3. Focuses on targeted solutions very effectively.
4. EPEx could also be useful to the developers for checking error handling code.
5. The paper also presents a comparison between the working of EPEx tool and other existing ones.

WEAKNESSES

1. The tool created here is dependent on the Clang Static Analyzer Engine for under constrained symbolic execution.
2. The execution speed of EPEx depends on presence and absence of inbuilt checker in the Clang tool which is used only for finding the error execution path.
3. There is a limitation in solution provided by the tool as it cannot detect all kinds of error handling bugs. It can only detect a bug that is encountered in the error path.
4. Also the detected error handling bugs cannot be fixed by the tool; the tool only finds them.