Enna Basic, Örebro University

Abstract

Large Language Models (LLMs) are increasingly used for programming and security tasks, but they can introduce vulnerabilities, miss real ones, hallucinate nonexistent ones, and may be further compromised by data poisoning attacks. This talk builds on a systematic literature review of LLMs in code security. The review shows that LLMs have been widely explored for vulnerability detection, vulnerability remediation, and secure code generation. At the same time, it highlights important limitations: LLMs may introduce insecure code, miss existing vulnerabilities, report false positives, and behave inconsistently depending on prompting strategies. These findings motivate my current work on more structured and interpretable approaches to LLM-assisted vulnerability detection.

The ongoing work investigates the use of behaviour trees as an intermediate representation for LLM-based vulnerability detection. Instead of analyzing source code directly, the approach first represents the program as a behaviour tree and then provides this representation to an LLM for vulnerability analysis. The motivation is that the behaviour tree structure may provide a clearer representation of the code logic and execution flow, potentially helping the model discover vulnerabilities that may be missed when analyzing the source code directly.