The dissertation work is original, the contribution is significant, and the impact is high.

This dissertation addresses several important and challenging problems in secure software testing and verification. The problems are important because vulnerabilities hidden in software code can be extremely dangerous and costly to users. The problems are challenging because they are hard to solve. Often times, the input space of a program is extremely large, which means finding a security vulnerability is like "looking for a needle in the haystack."

Despite these challenges, the candidate proposes several innovative solutions that, in my opinion, are not only effective but also practical.

In HAWKEYE, the candidate proposes a technique that combines static analysis and heuristic seed selection to guide fuzz testing tool toward a set of target locations. The technique is based on some excellent insights, e.g., on why state-of-the-art fuzz testing tools do not work well, and what information may be used to guide the search. I am particularly impressed by the creativity of some of the proposed methods, e.g., the way distance metrics are defined and computed, and the way seeds are prioritized. I am also impressed by the experimental results, which show that HAWKEYE outperforms state-of-the-art fuzz testing tools such as AFL and AFLGo. Since fuzz testing is indispensable in practice for detecting security vulnerability, the proposed technique is expected to have a large impact in the real world.

In DOUBLADE, the candidate extends the testing technique from sequential to multithreaded software. In this case, the main problem is due to nondeterminism caused by thread scheduling. Due to nondeterminism, running the software multiple times may lead to different behaviors even under the same input, which can make security vulnerability detection more difficult. Due to this reason, concurrency-induced vulnerabilities were significantly under studied. The candidate fills the gap by developing a thread-aware seed selection technique for fuzz testing multithreaded software. I am impressed by the excellent insights and good experimental results. As far as I know, this is indeed the first thread-aware, general-purpose, fuzz testing tool.

The candidate also proposes FOT, a unified framework for implementing and comparing various fuzz testing tool and techniques. While there are already many fuzz testing tools, in terms of the implementation, they are all tied to specific algorithms and thus hard to modify and extend. In contrast, FOT gives users the ability to quickly implement and evaluate a wide range of fuzz testing tools and techniques. Therefore, it is a valuable resource for the research community.

In addition to the extremely practical (fuzz testing) techniques, the dissertation also contains results that are of significant theoretical depth. Specifically, in the security type system work, the candidate solves a challenging and long-standing problem, i.e., designing a type system to more accurately handle non-monotonic security policies. While the problem was well-understood, for years, there was no good solution. The candidate solves it by leveraging an existing type system called the BN system. While the proposed solution remains to be implemented and evaluated on

Android applications (for which it is designed), the new type system is elegant and effective, and therefore I expect it to work well in practice.

To summarize, this is a high-quality PhD dissertation in all aspects.