Research Statement

Mary Hannoush

maryhannoush3@gmail.com

Introduction:

I am applying for the PhD position at the Software Lab of the University of Stuttgart with a focus on advancing tools and techniques for building reliable, efficient, and secure software systems. My background in Telecommunications Engineering, combined with hands-on experience in Software-Defined Networking (SDN), distributed software architectures, and cloud-native environments, has prepared me to address complex software challenges. My research ambitions are aligned with the Software Lab's cutting-edge work in program analysis, automated vulnerability detection, and performance optimization—as demonstrated in recent projects such as *VulGen*, *SecBench.js*, and *LExecutor*. I aim to extend these contributions through novel AI-driven methods for semantic vulnerability detection, automated repair, and intelligent optimization of distributed systems.

Research Interests:

1. AI-Aided Vulnerability Discovery and Automatic Fix

Problem Statement:Modern vulnerability detection tools are typically not semantic-aware and lack the ability to detect subtle vulnerabilities in cloud-native and distributed applications. Old scanners cannot cope with cross-service vulnerabilities because of complex inter-service communication. Besides, while vulnerability generation is an extensively researched topic, real-time detection and automatic fixing are primarily open problems.

*State of the Art and Limitations:*Software Lab's latest studies, such as VulGen: Realistic Vulnerability Generation Via Pattern Mining and Deep Learning (ICSE 2023) and SecBench.js: An Executable Security Benchmark Suite for Server-Side JavaScript (ICSE 2023), are advances in vulnerability generation and benchmarking. These studies are primarily detection within controlled settings with little real-time detection and no inherent repair feature.

*Proposed Approach:*I propose VulFix, an AI-powered extension of VulGen incorporating:

- Semantic Vulnerability Detection: LLM-based program analysis to uncover cross-cutting vulnerabilities across disparate microservices and distributed components.
- Contextual Automated Repair: Learning patterns to patch found vulnerabilities in real time.
- Dynamic Adaptation: Real-time analysis of traffic for dynamic adaptation of detection patterns in response to emergent attack vectors.

Research Questions:

1. How do techniques of semantic analysis improve the discovery of cross-service vulnerabilities in cloud-native systems?

2. Which models are optimum for real-time patch generation for distributed systems?

3. In what way will live traffic monitoring enhance detection mechanisms' flexibility?

*Expected Impact:*The project aims to enhance Software Lab's contribution by providing real-time vulnerability detection and patching of vulnerabilities in distributed software systems, hence providing a cloud-native application with a self-healing mechanism.

2. Test Generation and Optimization for Distributed Systems

*Problem Statement:*Distributed programs are inherently complex because of concurrency, timing and communication between nodes that a conventional testing generation technique might not be able to handle. Latency, synchronization failures and packet loss expose faults that cannot be detected by isolated testing.

*State of the Art and Limitations:*Software Lab's work, i.e., LExecutor: Learning-Guided Execution (FSE 2023) and Beware of the Unexpected: Bimodal Taint Analysis (ISSTA 2023), does path exploration and taint analysis but mostly on single-node or isolated configurations, without multi-service interactions.

Proposed Approach: I recommend NetTest, a distributed-aware test generation platform with:

- Context-Aware Test Generation: LLM-based analysis to generate tests for distributed interactions, such as latency-induced bugs.
- Dynamic Dependency Mapping: In situ modeling of service dependencies to improve detection of concurrency faults.
- Adaptive Failure Injection: Chaos engineering techniques to introduce faults and synchronization faults like those encountered in real applications.

Research Questions:

1. What are the optimal methods for in situ detection of latency faults in distributed systems?

2. How can in situ service dependency mapping expose hidden concurrency faults?

3. How does fault injection help discover vulnerabilities in multi-node interactions?

*Expected Impact:*NetTest extends Software Lab's efforts with distributed-aware testing, revealing hidden vulnerabilities in real cloud-native production environments.

3. Intelligent Performance Optimization of CI/CD Pipelines

*Problem Statement:*CI/CD pipelines are essential to software development but are inefficient, have long build times, and wasteful use of resources. Current approaches are largely static and do not adapt intelligently to workload changes.

*State of the Art and Limitations:*Software Lab's effort on Resource Usage and Optimization Opportunities in Workflows of GitHub Actions (ICSE 2024) identifies optimization opportunities but does not refer to adaptive, real-time optimization techniques.

Proposed Approach:I propose OptiCI, an AI-based optimization layer for CI/CD pipelines that introduce:

- Real-Time Resource Adjustment: Predictive workload analysis-driven dynamic scaling.
- Intelligent Caching and Dependency Management: Intelligent caching of modules for build time optimization.
- Contextual Optimization Suggestions: LLM-based analysis for workflow optimization.

Research Questions:

1. How might real-time predictive models enhance the efficiency of CI/CD pipelines?

2. What are the most effective caching strategies for reducing build times in distributed environments?

3. How might LLM-based analysis help support adaptive resource management?

*Expected Impact:*OptiCI would contribute to Software Lab's research by integrating real-time, adaptive optimization into CI/CD pipelines, significantly increasing pipeline efficiency and cloud resource utilization.

My experience in distributed network optimization, program analysis automation, and secure communication systems prepares me well to support Software Lab's mission. I think that VulFix, NetTest, and OptiCI complement and extend Software Lab's current work, advancing the frontiers of secure, optimized, and reliable software systems. I am eager to join the team and support effective research leading to cloud-native and distributed software reliability.

Kind regards,

Mary Hannoush

maryhannoush3@gmail.com