AI for Software Vulnerability Management: Detection, Prioritization, and Remediation

Conference Summary

SophI.A Summit 2025, November 19–21, Sophia Antipolis

Abstract. Merve Sahin from SAP Product Security presented research on using Large Language Models for software vulnerability management, covering detection, prioritization, and remediation. The results show strong promise for LLMs in automated repair, but also highlight persistent challenges in detection and reasoning that require human intervention. The most practical paths for future combine LLMs with traditional security tools within agentic frameworks.

Keywords: Software Vulnerability Management · Large Language Models · SAST · Security Automation

1 Introduction

Software vulnerabilities like weaknesses in design, implementation, or configuration , enable attackers to compromise confidentiality, integrity, and availability, with consequences ranging from data breaches to supply-chain attacks. The presentation emphasized a growing gap between vulnerability discovery and remediation, supported by recent reports [1,2]. Traditional Static Application Security Testing (SAST) tools struggle with false positives and false negatives and still rely heavily on expert judgment. SAP examined whether Large Language Models can help across the three stages of vulnerability management: detection, prioritization, and remediation.

2 Vulnerability Detection with LLMs

More than 300 studies have evaluated LLMs across many benchmark datasets with mixed outcomes [3]. In controlled, synthetic settings some LLMs outperform traditional SAST tools [4], but repository-level evaluations report high false-positive rates and inconsistent reasoning [5]. LLMs can be non-deterministic and often struggle with complex or poorly documented code.

3 Prioritization and Remediation

Research on automated prioritization remains limited. In SAP's collaboration with TU Braunschweig, the team constructed a ground-truth dataset of paired

Table 1. Vulnerability remediation results (optimal temperature, top-p)

LLM	Prompting	Fix Success Rate	# Vulnerable samples
GPT 4.1	Zero shot	85.9%	71
Gemini 2.5 Flash	Zero shot	77.3%	73
Claude 4 Sonnet	Zero shot	72.0%	75
Gemini 2.5 Pro	Zero shot	71.1%	73

fixes focusing on common web languages to evaluate remediation quality. LLMs produced well-structured fixes in many cases,often preferring helper functions over one-line patches,but some human fixes were still incomplete or had design issues.

4 Limitations and Future Directions

LLMs are not yet reliable as standalone detection tools because of false positives and inconsistent outputs. Commercial auto-fix offerings exist and can accelerate remediation workflows, but expert review remains necessary to verify correctness and security. The likely future is hybrid, agentic systems that integrate LLMs with SAST/DAST/SCA tooling.This raises important questions about how to ensure deterministic detection and avoid introducing AI-generated vulnerabilities.

5 Conclusions

The presentation showed that LLMs offer powerful assistance for remediation and can produce high-quality fixes in many cases, but they do not replace human experts. Detection accuracy and consistent reasoning remain key obstacles, and prioritization methods are still early-stage.

Hybrid approaches that combine LLMs with traditional security tools and human oversight appear most practical. These AI-augmented workflows can help manage the widening gap between discovery and remediation, provided deployments include verification steps and appropriate safeguards.

References

1. Action1: Software Vulnerability Ratings Report 2025
2. Cyentia: Why Your MTTR is Probably Bogus
3. Sheng et al.: LLMs in Software Security. ACM Comput. Surv. (2025)
4. Tamberg et al.: Harnessing LLMs for Vulnerability Detection. IEEE Access (2025)
5. Gaucher et al.: Finding Vulnerabilities in Modern Web Apps. Semgrep (2025)