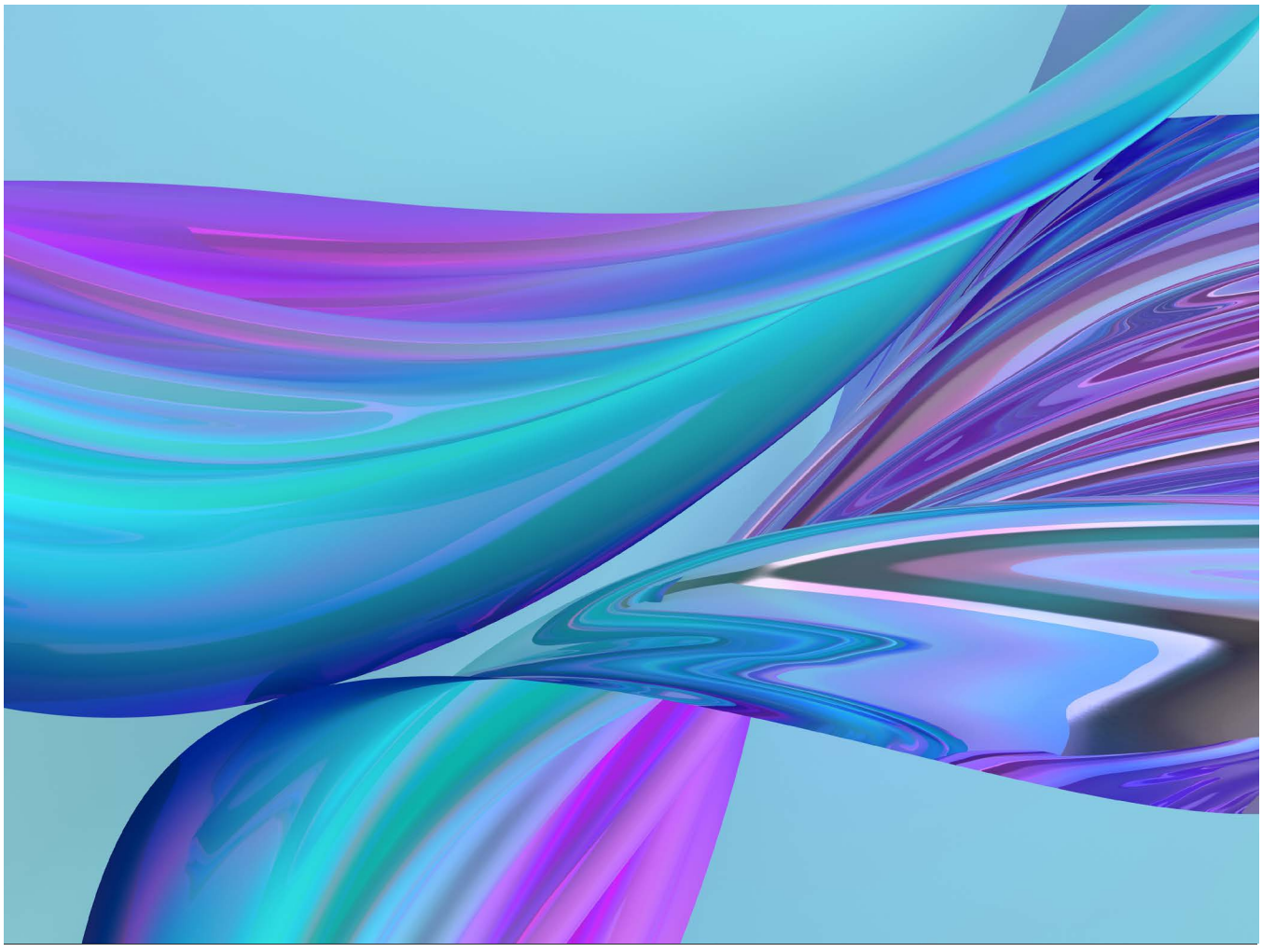

WHITE PAPER
The Human-AI Partnership:
A Practical Approach to
Vulnerability Remediation

Contents

A TL;DR Overview

Claims that AI can significantly expedite vulnerability remediation are true. It's very effective in triaging static application security testing (SAST) findings and can produce good and valuable fixes if the scope is chosen properly and tailored to the organisation's coding practices. Without any tuning, you can expect one-third of fixes to be successful.

A hybrid approach that combines AI suggestions with the experience of developers and security engineers currently yields the best results. The biggest trap to watch out for is transferring effort from coding to verification, rather than gaining efficiency by reducing effort overall. Our goal is to bring security use cases into the processes and tools that engineering teams use to accelerate development.



How is Vulnerability Remediation Challenging Developers?

Organizations face a significant number of vulnerabilities discovered by security tools, and with AI development on the rise, the situation is unlikely to improve. The use of generative AI to expedite remediation at the pace of development is one of the most promising applications of the technology in security.

A **recent survey** reveals that vulnerability and risk management is the top area where security leaders believe AI will offer the most value, as cited by 74% of respondents.

AI explosion is expected to exacerbate the situation, as many efforts focus on AI augmentation for engineering productivity, increasing the pace of code development, but leaving code security and other aspects of development behind.

In the January 2025 "**Emerging Tech: AI Developer Tools Must Span SDLC Phases to Deliver Value**" report, Gartner® recommends product leaders to, "Direct efforts to capitalize on opportunities in software engineering beyond code generation for GenAI tools by focusing on multiple phases of the SDLC, including requirements, design, architecture, testing, refactoring, documentation, DevOps, releases management, security audit and operations."*

We asked Eitan Worcel, CEO of Mobb.ai, a leading solutions provider in auto remediation using algorithmic fixes, his opinion on how challenging SAST remediation is for the developers:

"They find impossible to even triage the number of findings they get and understand that, with an average of five hours to fix a single finding, the company will need to have their dev workforce spend years just addressing those, instead of on capabilities that help drive more business."

However, as our practical testing has shown, automatic remediation is not as simple as dropping a few lines of code into LLM and copying and pasting the result back, so GitHub Copilot is not the answer.

*Gartner®: Emerging Tech: AI Developer Tools Must Span SDLC Phases to Deliver Value. Ray Valdes, et al. 29 January 2025.

Current Market & Research Landscape

Current approaches to AI-based vulnerability remediation generally fall under two categories:

01

Pure AI approaches, which utilize LLMs to generate complete code fixes based on vulnerability descriptions and surrounding code context. While these solutions offer an impressive breadth of coverage across vulnerability types, they encounter hallucinations, generating plausible but incorrect fixes.

02

Algorithmic or hybrid approaches that combine traditional rule-based systems with AI assistance. These solutions typically offer higher precision for specific vulnerability types but more limited coverage across the full spectrum of security issues.

Various academic research published last year has demonstrated encouraging results, with pure AI approaches successfully generating proper fixes for up to 60% of certain vulnerability types after applying sophisticated prompt techniques and remediation-pipeline improvements. Among the above-mentioned studies include:

APPATCH: Automated Adaptive Prompting Large Language Models for Real-World Software Vulnerability Patching

Enhanced Automated Code Vulnerability Repair Using Large Language Models

A Case Study of LLM for Automated Vulnerability Repair: Assessing Impact of Reasoning and Patch Validation Feedback

How Good Are LLMs at Patching Vulnerabilities?

AI-powered Patching: The Future of Automated Vulnerability Fixes

Introducing AutoPatchBench: A Benchmark for AI-Powered Security Fixes

The market offers tools with both hybrid and pure AI approaches, where the latter prevails in the form of copilots.

Experimentation

To evaluate technology readiness, we created auto-fix tools based on pure AI and arranged a POC with the market leader on algorithmic fixes. We then evaluated them against the OWASP WebGoat project, an intentionally vulnerable application designed for security training, and two EPAM production projects. The focus was on vulnerabilities with relatively straightforward mitigations, such as SQLi, Path Traversal, XEE, etc.

Figure 1: Our testing environment.

Initial results for AI-generated fixes with WebGoat were promising, with approximately one-third of AI-generated fixes being immediately applicable and correct, another third requiring minor modifications but providing a solid foundation, and the remaining third being incorrect or potentially breaking application functionality.

However, when we expanded testing to production applications with actual security vulnerabilities, the rate of successful fixes that can be adapted without tuning dropped even lower due to the complexity of the production codebase.

It is worth mentioning that the projects have already fixed most of the low-hanging fruit, and fixing the remaining vulnerabilities was not straightforward even for human experts. So, the results on other projects could be better.

As for algorithmic fixes from specialized tools, they addressed only a limited fraction of the vulnerability backlog (15-25%, counting only high and medium severity issues) on our test projects.

It's important to note that the effectiveness of AI-powered remediation varies significantly based on project context. Factors such as codebase complexity, framework usage and the nature of vulnerabilities all impact success rates. Rather than providing specific percentages or estimates that might be misleading across different environments and considering the rapid evolution of LLMs, organizations should conduct their own evaluations within their specific context.

Overall, our testing revealed several critical insights:

01

SUPERFICIAL CORRECTNESS

LLMs frequently generate fixes that appear correct at first but contain subtle flaws that could leave vulnerabilities exploitable or break functionality. This "plausible incorrectness" is particularly dangerous as it may pass a casual review.

02

TRIAGE & VERIFICATION CHALLENGES

AI-accelerated fix implementation is only part of the full vulnerability remediation flow. Triage, analysis, and testing often consume more time than the actual coding, presenting significant challenges for both development and security teams.

03

NEED FOR PROPER SCOPING

Fixing vulnerabilities across multiple repositories or components, such as updating outdated cryptographic algorithms used in the authorization flow, requires a more proactive approach and will demand more attention. So, it's worth starting with vulnerabilities that can be addressed within one component.

04

INTEGRATION OBSTACLES

It's one thing to have a solution that technically solves the problem. But having developers or security teams to use it is an entirely different story. Our experiments and overall market research revealed that the adoption of GenAI is not straightforward.

Our experiments and overall market research revealed that the adoption of GenAI is not straightforward. Eitan also shared his perspective on the limitations of LLMs.

"Unlike 'regular' software, which is deterministic, GenAI follows a probabilistic approach. This means that, given similar and even the same tasks, a GenAI solution often provides different responses. When this happens, it is usually referred to as AI hallucination by some and AI creativity by others. In any case, it means that the answers are unpredictable and therefore unreliable.

It's been proven time and time again that following well-defined, battle-tested best practices when dealing with security requirements is the path forward. And, while there's value in creativity when writing code, very rarely is that the case with fixing code vulnerabilities.

Using GenAI for code remediation without proper guardrails forces developers to extensively review every suggested change to ensure it doesn't break the code, alter application behavior, or fail to resolve the issue. Many developers lack the skills for this. For example, guardrails could include not asking the model to address issues it cannot handle, such as weak encryption, since it won't respond with 'I can't.'"

We agree that security must be deterministic. However, if, as an industry, we are going to trust LLMs with writing code, it should become reliable enough to produce secure code or rewrite existing code with proper security. The path to achieve this is described in the next section, but before jumping into it, it is worth looking into the power of GenAI in triage of the vulnerabilities.

During the remediation experimentation, we estimated that up to 90% of the total backlog consisted of issues not worth fixing — false positives or true positives with virtually no practical risk in the context of these specific projects. This directed our second experiment: **creating an agent** capable of triaging SAST findings.

The architecture and workflow are very straightforward. We created a comprehensive description of the project and its preferences regarding the definition of false positives. Then the agent pulls findings from SAST, retrieves insights and relevant code from the repository, and suggests an analysis with a conclusion on whether the vulnerability should be fixed or suppressed.

To evaluate performance, we crafted a benchmark based on the OWASP benchmark for SASTs that includes tricky examples where some code is not exploitable and has labelled data on which findings are true positives. On this benchmark, we got ~95% precision and ~70% recall for false-positive detection. Meaning that 70% of all false positives were detected, reducing potential backlog for developers, and 95% of the false positives were correctly triaged, leaving 5% of true positives missed and required to be caught by a human.

So far, we haven't observed any major issues preventing this use of tech and gaining immediate value. We are currently implementing this approach in a real project with partially labelled data and expect results soon.

Path to Address AI Limitations

As experiments showed, AI out of the box can provide gains in low-risk activities like triage. To further increase gains for high-risk activities like remediation, we are working on the following steps:

Implementation of self-verification with ReAct and Reflection based on tests

Further improvement of the triage agent

Focus on further customization, language and vulnerability-specific

Creating a library of verifiable fixes

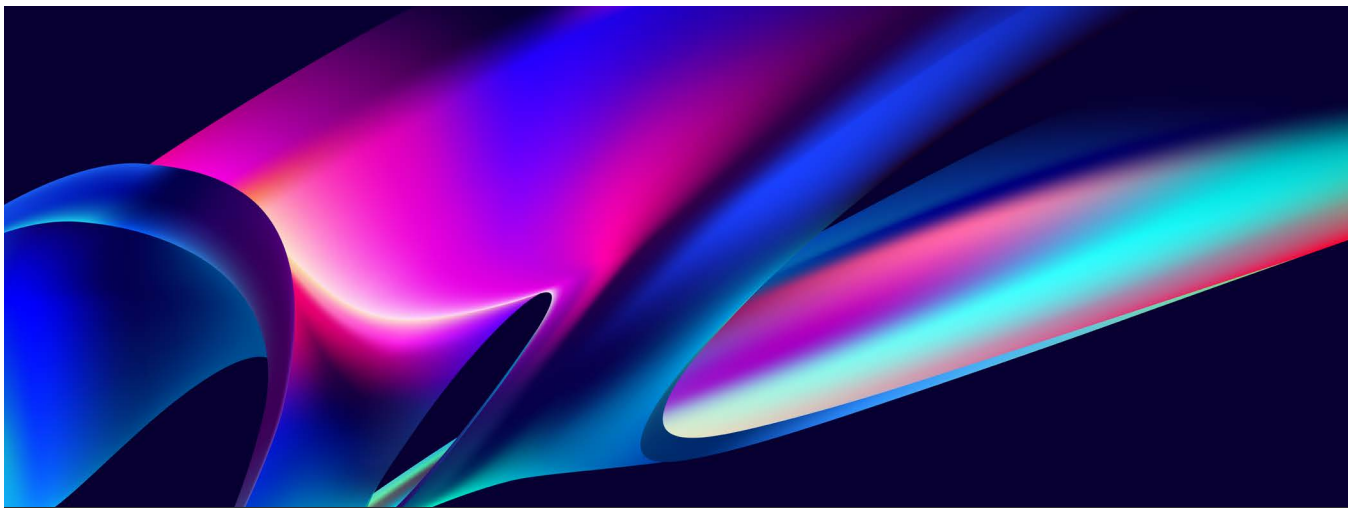
Regarding AI's ability to overcome limitations, Eitan had the following perspective to share:

"Advancing technology is quickly closing gaps once thought out of reach. A key hurdle is the lack of large, well-curated datasets for proper model training. Companies that acquire such datasets will unlock major improvements in model output quality. By combining GenAI with deterministic guidance, guardrails, and verification, some solutions already address many vulnerabilities. Over time, as GenAI evolves, these systems may autonomously resolve most vulnerabilities."

While gradual, this process has begun, and within 1-3 years, it could cover up to 90-100% of vulnerabilities, though not all may be addressable.

Focusing on further customization, language and vulnerability-specific issues, and creating a library of verifiable fixes, provides precisely the deterministic guidance.

To take it further, a group of EPAM engineers have initiated the **OWASP "Cheatcode" project** that extends known security best practices with concrete, testable code examples. So far, it covers known techniques for remediating path traversal in Java but with the collaborative effort of the community, it can become a fantastic source of truth and practical extension of the OWASP Cheat Sheet project.



Integrations Into the Development Workflow

The challenge to address is integrating GenAI into organizations, including security use cases, as demonstrated by analysts:

"Integrating AI agents into developer workflows requires intuitive UI design that enables developers to effectively interact with and use the AI agents."

Gartner®: How AI Agents Will Disrupt Software Engineering. Adrian Leow. 28 September 2024.

"Implementation challenges will stall more than 50% of agentic and AI agent efforts."

Forrester

"Our data shows that organizations have conducted an average of 37 proofs of concept, but only about five have moved into production. It's been a year of intense experimentation. (author talks about 2024) Now, the big question is: What will it take to move from experimentation to adoption? The key areas we see are having an enterprise AI strategy, a unified governance model and managing the technology costs associated with genAI to present a compelling business case to the executive team."

CIO

To ensure security keeps up with the speed of AI-driven coding, we are integrating security use cases into the AI/Run™ methodology. The framework encompasses all aspects of integration, ranging from change management and education to a platform for running an ecosystem of tools and agents. And we have already made vulnerability triage and remediation agents accessible to engineering teams through tools and processes that they're currently using for test creation and code refactoring.

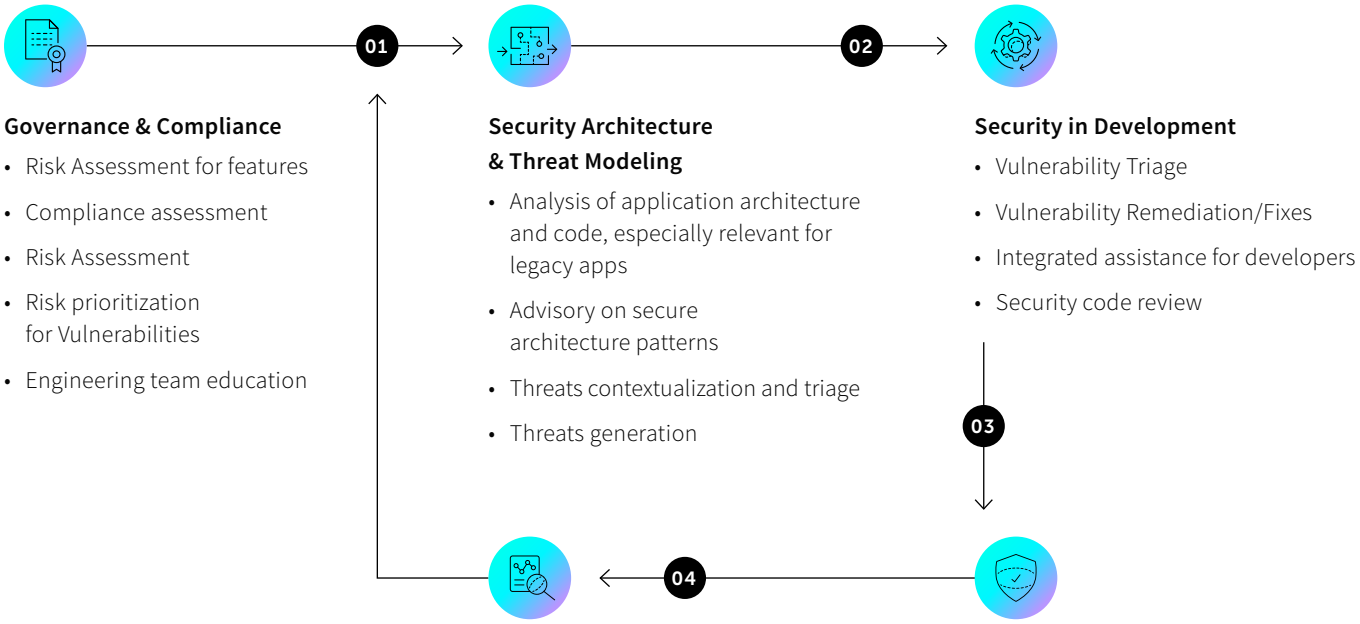
This facilitates a hybrid approach, where AI generates candidate fixes that human experts then verify and adapt without leaving the development environment, offering the best balance of efficiency and reliability.

It's too early to say about the effectiveness of this approach compared to introducing a new, purely security-related platform. However, we believe that the capabilities of GenAI open a unique opportunity for engineering teams to become better at security, giving everyone their personal security advisor with a centralized and standardized approach.

Some of the security use cases we plan to onboard are depicted on the diagram below. We look forward to continuing to publish as we progress with implementation.

What's Next — Onboard Security Use Cases to SDLC Acceleration Programs

AI/RUN FRAMEWORK



Governance & Compliance

- Risk Assessment for features
- Compliance assessment
- Risk Assessment
- Risk prioritization for Vulnerabilities
- Engineering team education

Security Architecture & Threat Modeling

- Analysis of application architecture and code, especially relevant for legacy apps
- Advisory on secure architecture patterns
- Threats contextualization and triage
- Threats generation

Security in Development

- Vulnerability Triage
- Vulnerability Remediation/Fixes
- Integrated assistance for developers
- Security code review

Security Operation

- Connectors creation and rules migration
- Detection and response acceleration
- Identification and onboarding of IoCs

Offensive Security

- Attack surface discovery
- AI for massive social engineering attacks simulation
- Fuzz testing and vulnerability discovery
- Report enrichment

Conclusion

Even with current technology, GenAI solutions for vulnerability triage and remediation provide significant benefits to security and engineering teams. Although existing solutions have limitations, rapid advancements indicate that significant improvements are likely in the near future. Organizations need to keep pace with AI-driven development, and should therefore actively integrate AI into their vulnerability remediation efforts while maintaining human oversight.

About EPAM

Since 1993, EPAM Systems, Inc. (NYSE: EPAM) has used its software engineering expertise to become a leading global provider of digital engineering, cloud and AI-enabled transformation services, and a leading business and experience consulting partner for global enterprises and ambitious startups.

We address our clients' transformation challenges by fusing EPAM Continuum's integrated strategy, experience and technology consulting with our 30+ years of engineering execution to speed our clients' time to market and drive greater value from their innovations and digital investments.

We leverage AI and GenAI to deliver transformative solutions that accelerate our clients' digital innovation and enhance their competitive edge. Through platforms like EPAM AI/RUN™ and initiatives like DIALX Lab, we integrate advanced AI technologies into tailored business strategies, driving significant industry impact and fostering continuous innovation.

We deliver globally, but engage locally with our expert teams of consultants, architects, designers and engineers, making the future real for our clients, our partners and our people around the world.

We believe the right solutions are the ones that improve people's lives and fuel competitive advantage for our clients across diverse industries. Our thinking comes to life in the experiences, products and platforms we design and bring to market.

Added to the S&P 500 and the Forbes Global 2000 in 2021 and recognized by Glassdoor and Newsweek as Most Loved Workplace, our multidisciplinary teams serve customers across six continents. We are proud to be among the top 15 companies in Information Technology Services in the Fortune 1000 and to be recognized as a leader in the IDC MarketScapes for Worldwide Experience Build Services, Worldwide Experience Design Services and Worldwide Software Engineering Services.

Learn more at **www.EPAM.com** and follow us on **LinkedIn**.

Headquarters

41 University Drive, Suite 202
Newtown, PA 18940, USA

P: +1-267-759-9000
F: +1-267-759-8989