

APRIL 18-19, 2024

MARINA BAY SANDS / SINGAPORE

AutoFix: Automated Vulnerability Remediation Using Static Analysis and LLMs

Asankhaya Sharma





Asankhaya Sharma, Co-Founder & CTO, https://patched.codes









2007

2014

2019

2023

2003



2010

[:]SourceClear

2018



2022









DIDAR - Database Intrusion Detection with Automated Recovery HIP/SLEEK: Automatic
Verification and Specification
Inference System









building security tools for developers

V/S

developer tools for security



OWASP Top Ten

		2003/2004		2007		2010		2013		2017		2021
	A1	Unvalidated Input		Cross-Site Scripting (XSS)		Injection		Injection		Injection		Broken Access Control
	A2	Broken Access Control		Injection Flaws		Cross-Site Scripting (XSS)		Broken Authentication and Session Management		Broken Authentication		Cryptographic Failures
	А3	Broken Authentication and Session Management		Malicious File Execution		Broken Authentication and Session Management		Cross-Site Scripting (XSS)		Sensitive Data Exposure		Injection
	A4	Cross-Site Scripting (XSS)		Insecure Direct Object References		Insecure Direct Object References		Insecure Direct Object References		XML External Entities (XXE)		Insecure Design
	A5	Buffer Overflow		Cross-Site Request Forgery (CSRF)	-	Cross-Site Request Forgery (CSRF)		Security Misconfiguration		Broken Access Control		Security Misconfiguration
	A6	Injection Flaws		Information Leakage and Improper Error Handling		Security Misconfiguration		Sensitive Data Exposure		Security Misconfiguration		Vulnerable and Outdated Components
	A7	Improper Error Handling		Broken Authentication and Session Management		Insecure Cryptographic Storage		Missing Function Level Access Control		Cross-Site Scripting		Identification and Authentication Failures
	A8	Insecure Storage		Insecure Cryptographic Storage		Failure to Restrict URL Access		Cross-Site Request Forgery (CSRF)		Insecure Deserialization		Software and Data Integrity Failures
	A9	Application Denial of Service		Insecure Communications		Insufficient Transport Layer Protection		Using Components with Known Vulnerabilities		Using Components with Known Vulnerabilities		Security Logging and Monitoring Failures
	A10	Insecure Configuration Management		Failure to Restrict URL Access		Unvalidated Redirects and Forwards		Unvalidated Redirects and Forwards		Insufficient Logging & Monitoring		Server-Side Request Forgery (SSRF)



Breaking the Cycle: Beyond Scan-and-Fix in AppSec

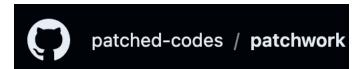


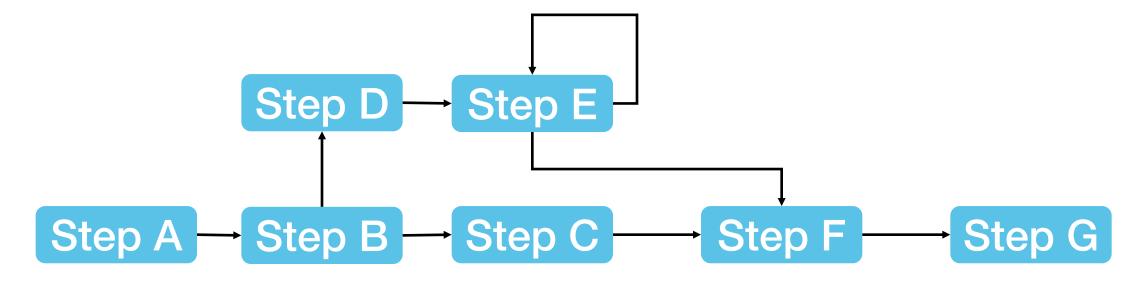
- Old Ways, New Challenges: Stuck in a "scan and fix" loop, traditional SAST/DAST tools leave us chasing vulnerabilities, not proactive security.
- Shift Left Illusion: Moving security earlier in the SDLC doesn't stop the cycle; it starts it sooner, overburdening developers with endless issues to fix.
- IDE Interruptions: Real-time scanning in IDEs promises security but disrupts developer workflow, compromising productivity with constant alerts and system overhead.

This pattern is just wrong. It's broken. We've seen a history of the challenges following this pattern does in working with developers." — Chris Romeo



Patchflow



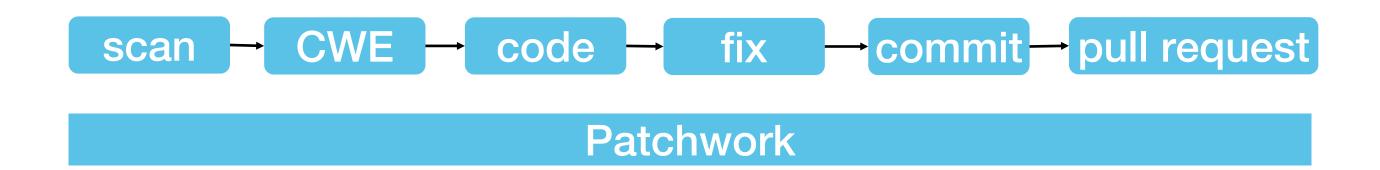


Patchwork

https://github.com/patched-codes/patchwork



AutoFix



patchwork AutoFix -sarif=results.sarif -createBranch=patch-main -severity=critical -patch=customprompts.json



Extensible with steps

Add, modify, or remove steps easily to extend the patchflow to suit your needs.

Integrations with development tools

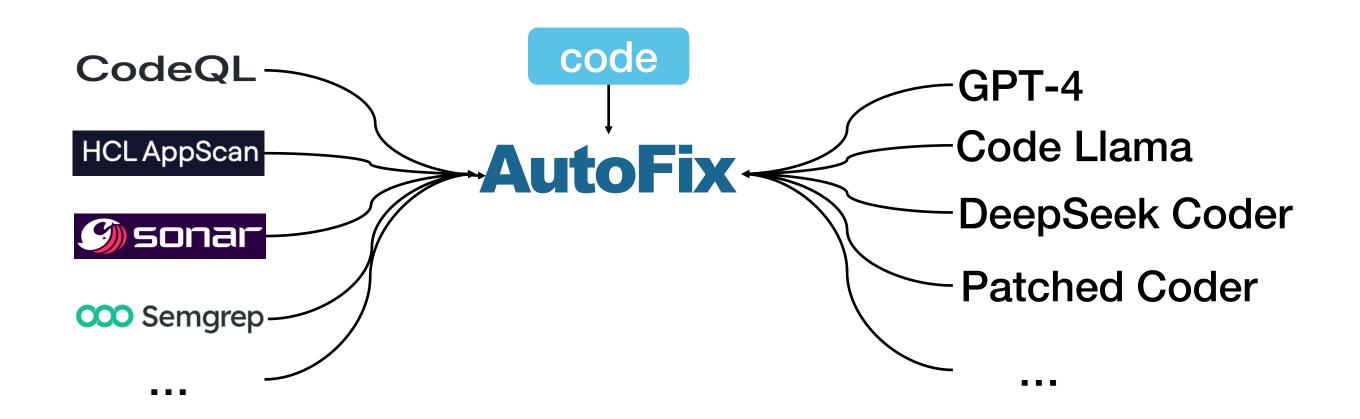
Run anywhere CLI, CI, IDE etc.

Customizable with patchprompts

Use of customizable natural language prompts allows for highly tailored solutions to specific vulnerabilities or coding standards. This flexibility ensures that the automated fixes are aligned with the project's requirements and coding practices.



SAST + LLMs





DEMO

blackhat[®] ARSENAL

APRIL 18-19, 2024
MARINA BAY SANDS / SINGAPORE

Thanks!

https://github.com/patched-codes/patchwork

asankhaya@patchedcodes.com