RESEARCH REPORT 2026

Can AI Really Review Code for Security?

A Practical Benchmark on a Vulnerable Application

AUTHORED BY

ENCIPHERS

Comprehensive Evaluation

Claude Codex Gemini

BENCHMARK OVERVIEW

Executive Summary

This benchmark evaluates the practical capabilities of leading AI models in security code review. Beyond simple code completion, we test their ability to function as reliable security analysts.

Models Evaluated

Claude Code Opus	OpenAI Codex	Gemini Pro
Version 4.6	Version 5.3	Version 2.5

Evaluation Scope

We utilized an intentionally vulnerable web application containing a validated ground truth of **39 security issues**. The stack included Python/Flask, PostgreSQL, and GraphQL endpoints.

SQL Injection | XSS | IDOR | SSRF

To ensure fairness, all models received the **same scope** and the **same standardized prompt**. We measured performance not just on raw detection, but on practical utility for an enterprise workflow.

Key Metrics

- **Detection Rate:** Percentage of valid issues found.
- **Noise Levels:** Frequency of false positives.
- **Severity Accuracy:** Correct risk classification.

Operational Factors

- **Review Time:** Speed of analysis.
- **Analyst Effort:** Amount of prompting required.
- **Business Logic:** Ability to find logic flaws.

Goal for AppSec Teams

This report presents measured results and neutral observations to help security leaders understand where AI can augment human review and where it falls short.

CONTEXT & MOTIVATION

Why This Benchmark Matters

When AI joins the security team, the standards change. This isn't just about writing code, it's about protecting it.

The Security Review "Trust Gap"

A normal code review tool can get away with being "mostly useful." A security review tool cannot. In security, the cost of error is asymmetric:

RISK 1: MISSED FINDINGS

If it misses critical vulnerabilities, it creates a false sense of security that is worse than no review at all.

RISK 2: ALERT FATIGUE

If it reports too much noise, developers lose trust and stop checking the tool's output entirely.

What Enterprises Actually Need

Genuine Vulnerability Detection
Must find real, exploitable issues, not just stylistic nits.

Noise Reduction
Must distinguish between a theoretical risk and a practical threat.

Operational Reliability
Should save time during manual review without needing constant hand-holding.

Not Just Another AI Comparison

This evaluation avoids generic "coding benchmarks" to focus specifically on the **AppSec workflow**. We aren't asking "Can it write code?" We are asking "Can a CISO or Engineering Lead trust it to audit their infrastructure?"

METHODOLOGY

Evaluation Scope & Testing Protocol

A rigorous, standardized approach to measuring AI security capabilities against a known ground truth.

01 The Target

An **intentionally vulnerable web application** designed to simulate a realistic modern stack. This was not a simple single-file test case.

Python / Flask | PostgreSQL | REST & GraphQL | Docker | HTML/CSS/JS

GROUND TRUTH

39 Validated Issues

02 Vulnerability Classes

The 39 validated vulnerabilities covered a comprehensive range of OWASP Top 10 and critical security flaws, including:

- Cross-Site Scripting (XSS)
- SQL Injection
- Server-Side Request Forgery (SSRF)
- Insecure Direct Object Reference
- XXE Injection
- Insecure File Handling
- Sensitive Info Exposure
- Misconfigurations

Prompt Strategy

All models received the same structured instruction to ensure fair comparison.

Finding_ID	Unique Identifier
Severity	Risk Level
Vulnerability_Type	Classification
CWE_ID / OWASP	Standard Mapping
Steps_To_Reproduce	Verification Path
Remediation	Fix Advice

Validation Logic

Every model output was manually verified against the ground truth dataset.

True Positive
Correctly identified valid issue

False Positive
Reported issue was invalid

Missed Finding
Valid issue not reported

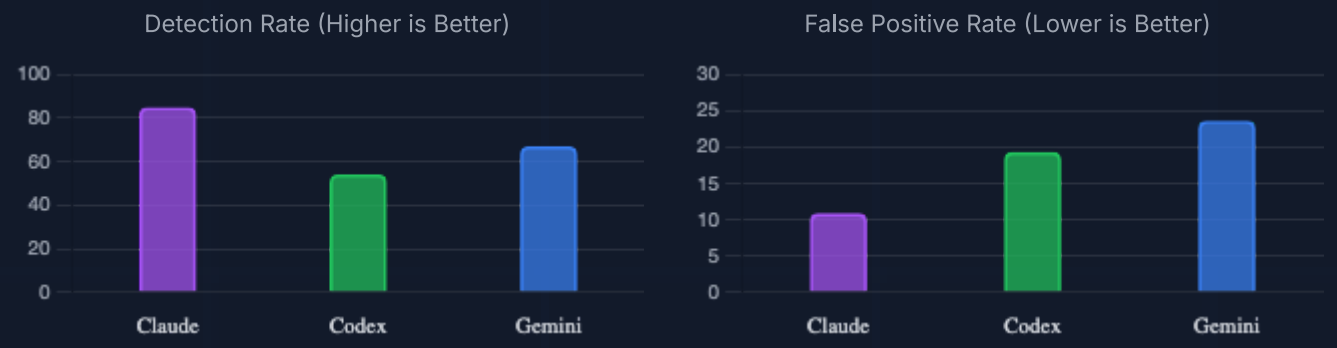
Also tracked: Time Taken & Prompting Effort

PERFORMANCE OVERVIEW

Benchmark Summary

A direct comparison of detection capabilities, accuracy, and operational efficiency across the three evaluated models.

Detection vs. Noise



Claude

Opus 4.6

#1

VALID FINDINGS

33 / 39

DETECTION RATE 84.62%

SEVERITY ACCURACY 72.97%

REVIEW TIME

7m 52s

WEIGHTED SCORE

72.19

Codex

Version 5.3

#2

VALID FINDINGS

21 / 39

DETECTION RATE 53.85%

SEVERITY ACCURACY 88.46%

REVIEW TIME

7m 15s

WEIGHTED SCORE

62.78

Gemini

Pro 2.5

#3

VALID FINDINGS

26 / 39

DETECTION RATE 66.67%

SEVERITY ACCURACY 88.24%

REVIEW TIME

17m 40s

WEIGHTED SCORE

58.64

Business Logic Gap: Zero business logic vulnerabilities were detected by any model in this benchmark.

CORE CAPABILITIES

Detection Comparison

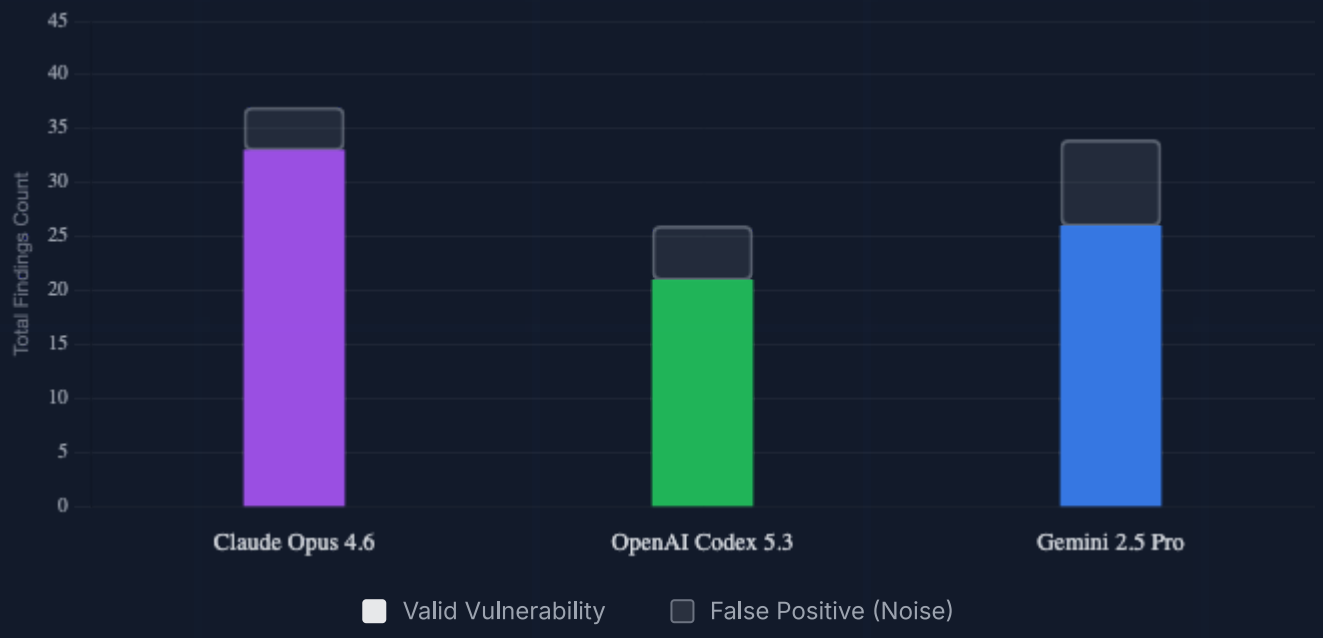
Evaluating the primary function of a security reviewer: the ability to identify valid vulnerabilities within the code.

Signal vs. Noise

A high number of findings is not necessarily better. Ideally, a model should maximize **Valid Vulnerabilities** (Signal) while minimizing **False Positives** (Noise). The chart below visualizes the total findings reported by each model, broken down by validity.

Findings Composition

TOTAL VS. VALID



Claude

LEADER

DETECTION RATE

84.6%

Valid Issues	**33**
Total Findings	37

Codex

DETECTION RATE

53.8%

Valid Issues	**21**
Total Findings	26

Gemini

DETECTION RATE

66.7%

Valid Issues	**26**
Total Findings	34

ANALYSIS PHASE 02

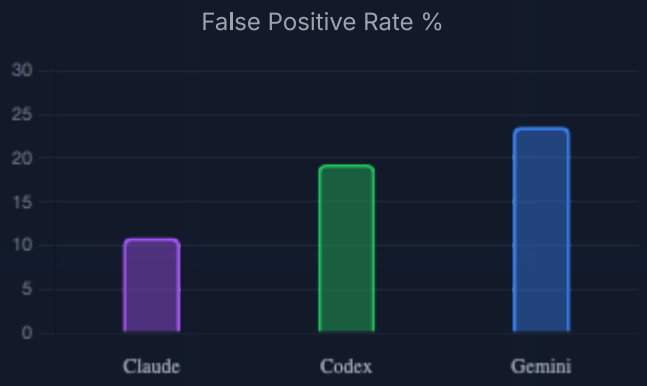
False Positives & Missed Findings

Examining the trust gap: how much noise did the models create, and how many real vulnerabilities slipped through the cracks?

The Noise Factor

LOWER IS BETTER

Claude Opus 4.6	4	FALSE POSITIVES
Codex Version 5.3	5	FALSE POSITIVES
Gemini Pro 2.5	8	FALSE POSITIVES



The Coverage Gap

LOWER IS BETTER

Claude Least Missed	6	MISSED ISSUES
Codex Most Missed	18	MISSED ISSUES
Gemini Moderate Gap	13	MISSED ISSUES

Neutral Observation

Performance varied significantly across models. **Claude** kept noise and blind spots relatively low. **Codex** had a high miss rate, while **Gemini** struggled with higher noise levels. This inconsistency reinforces that current AI models cannot yet replace human validation.

RISK ASSESSMENT QUALITY

Severity Accuracy

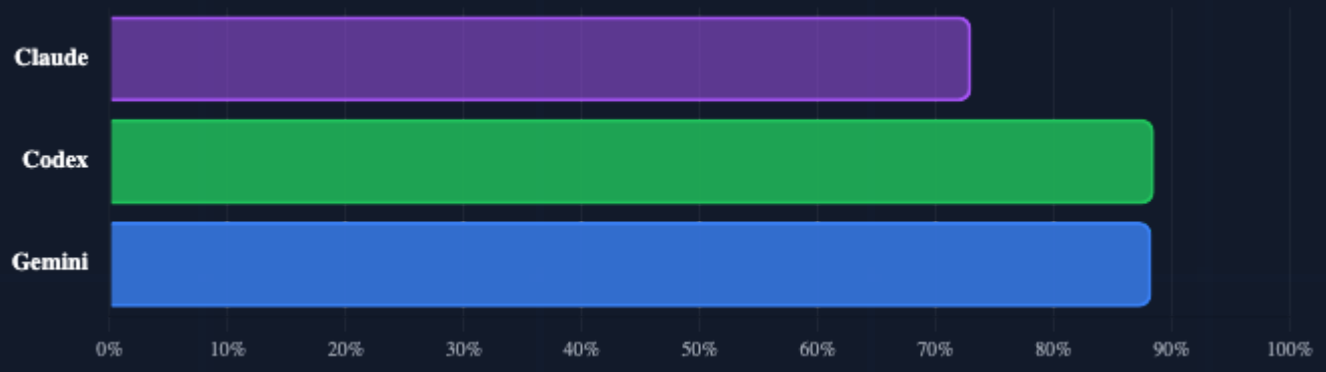
Finding a vulnerability is only step one. Correctly assessing its severity is critical for prioritization and remediation workflows.

The Grading Gap

A model that marks a "Critical" issue as "Low" creates danger. A model that marks "Low" issues as "Critical" creates panic. This metric evaluates how often the AI's severity rating matched the validated ground truth.

Severity Accuracy Rate

Higher is Better



Detailed Classification Metrics

MODEL	ACCURACY	MISCLASSIFIED	ERROR RATE
Claude Opus 4.6	72.97%	10 Issues	27.03%
Codex Ver 5.3	88.46%	3 Issues	11.54%
Gemini Pro 2.5	88.24%	4 Issues	11.76%

Time & Effort

Beyond just finding bugs: Measuring the speed of analysis and the human effort required to extract value from each model.

Analysis Duration

Minutes:Seconds

Workflow & Prompting Friction

1

Claude Opus 4.6

Autonomous

Produced a **strong result in a single prompt**. It behaved most like a trusted senior reviewer, requiring minimal hand-holding to find the majority of issues.

Codex Version 5.3

Two-Pass

Fastest initial speed, but **benefited from a second pass**. The additional review cycle helped surface high/critical findings that were missed in the first sweep.

Gemini Pro 2.5

High Friction

Required **repeated follow-ups**. Initial output was much smaller. A broader vulnerability list only emerged after several rounds of prompting, adding significant operational overhead.

Operational Reality

While Gemini's 17-minute runtime is high, the **hidden cost** lies in the analyst's time spent crafting follow-up prompts. Claude's ability to deliver in one shot makes it significantly more viable for high-volume enterprise pipelines.

CRITICAL LIMITATION

Business Logic Gaps

Identifying technical flaws like SQLi is one thing. Understanding complex application workflows and logic errors is another challenge entirely.

VULNERABILITIES DETECTED

Across All Models

- Claude: **0**
- Codex: **0**
- Gemini: **0**

Why This Matters

Business logic vulnerabilities, such as bypassing payment steps, manipulating order totals, or abusing workflow-based privilege paths, require understanding the **intent** of the application, not just the syntax of the code.

Practical Limitation

The benchmark confirmed a major gap: **AI models struggle to reason beyond obvious technical flaws.** While they performed well on pattern-matching issues like XSS and SQLi, they completely missed logic-level issues in this dataset.

Takeaway for AppSec Teams

Do not rely on AI for logic review. Human reviewers must continue to focus on architectural flaws and business workflow validation, using AI primarily as an assistant for syntax-heavy and pattern-based vulnerability detection.

COMPREHENSIVE EVALUATION

Weighted Scoring

Combining detection capability, accuracy, speed, and usability into a single unified benchmark score.

Scoring Weights

30%	20%	15%	15%	10%	10%
DETECTION	FP HANDLING	LOGIC	SEVERITY	SPEED	ENTERPRISE

Component Scores (Normalized)

METRIC	CLAUDE	CODEX	GEMINI
Detection (30%)	**84.62**	53.85	66.67
FP Handling (20%)	**89.19**	80.77	76.47
Business Logic (15%)	0.00	0.00	0.00
Severity Accuracy (15%)	72.97	**88.46**	**88.24**
Performance (10%)	92.16	**100.00**	41.04
Enterprise Suitability (10%)	**88.00**	72.00	60.00

Claude
Opus 4.6
72.2 /100
TOP RANKED

Codex
Version 5.3
62.8 /100
RUNNER UP

Gemini
Pro 2.5
58.6 /100
MODERATE

Practical Implications

For security teams adopting AI code review, success depends on understanding limitations and integrating tools into the right part of the workflow.

Human Validation Remains Essential

AI assistance is a force multiplier, not a replacement. The benchmark shows that even top models miss critical issues or hallucinate vulnerabilities. Experienced AppSec judgment is required to triage findings and validate risk.

Workflow Fit: The Trilemma

Teams must balance **Coverage**, **Noise**, and **Speed**. A tool like Claude fits autonomous pipelines due to low noise, while Codex may serve better as a second pair of eyes for quick sanity checks despite lower coverage.

Prompt Efficiency = Operational ROI

In enterprise workflows, time to value matters. Models that require fewer follow-up prompts significantly reduce analyst fatigue and operational overhead compared to models needing constant nudging.

Severity Accuracy ≠ Coverage

Do not confuse classification quality with detection power. A model might be excellent at rating the severity of the few bugs it finds while still missing a large portion of the attack surface.

The Logic Blind Spot

Since no model detected business logic vulnerabilities, security teams must retain manual review or specialized testing for logic-heavy features such as authorization workflows, fraud controls, and approval paths.

Key Takeaways

A high-level view of performance across detection, accuracy, speed, and overall suitability for enterprise security workflows.

The Ground Truth

Validated vulnerabilities in target application

39
TOTAL VALID ISSUES

Detection Rate

Claude	TOP	84.62%
Gemini		66.67%
Codex		53.85%

False Positive Rate

Lower is better

Claude	BEST	10.81%
Codex		19.23%
Gemini		23.53%

Severity Accuracy

Codex	TOP	88.46%
Gemini		88.24%
Claude		72.97%

Time to Review

Faster is better

Codex	FASTEST	7m 15s
Claude		7m 52s
Gemini		17m 40s

Business Logic Detection

Performance on logic-level vulnerabilities across all models

0
ISSUES FOUND

Final Weighted Scores

Claude Opus 4.6	OpenAI Codex 5.3	Gemini 2.5 Pro
72.19	62.78	58.64

FINAL VERDICT

The Future of AI Security Review

AI can **support** application security review, but it cannot **replace** it.

These models are not plug-and-play security auditors. Their findings still require **validation**, **context**, and **judgment** from experienced application security professionals.

A Tool, Not A Teammate

The best model in this benchmark was the most useful tool, effectively reducing manual effort but still requiring oversight.

Validation is Key

False positives and missed issues mean blind trust in AI output can waste engineering time and leave real risk behind.

Can AI Really Review Code for Security?

A Practical Benchmark on a Vulnerable Application

RESEARCH & ANALYSIS
ENCIPHERS

REPORT GENERATED
March 2026

CLASSIFICATION
Public Benchmark

END OF DOCUMENT