**SECURENET**

EU Horizon Europe Project

# AutoPoV

## Autonomous Proof-of-Vulnerability Generation & Validation

*Discovery-First · CWE-Agnostic · Runtime-Validated at Scale*

**697 Scans · 26 Repositories · 6 AI Models · 328 Confirmed Vulnerabilities**

Funded by the European Union — Horizon Europe, Grant Agreement No. 101217315

# The Problem & Our Approach

## THE GAP TODAY

### Scanners lack proof

Static tools flag possible issues but can't confirm they are exploitable — false positives erode trust.

### Manual exploits don't scale

Writing proof-of-concept exploits needs deep expertise and creates a pipeline bottleneck.

### Locked to known bug classes

Most tools target predefined CWE categories and miss novel or unconventional bugs.

## WHAT AUTOPOV DOES

An end-to-end autonomous framework that scans source code, investigates findings, writes proof-of-vulnerability scripts, and validates them by running them in isolated Docker containers.

- **Discovery-first** — broad static analysis before targeted investigation
- **CWE-agnostic** — no predefined vulnerability categories required
- **Runtime-validated** — proofs executed in Docker with crash / sanitizer oracles
- **Multi-model** — 6 LLM backends — 4 offline, 2 online

# How AutoPoV Works

*From source code to a runtime-proven vulnerability in one autonomous pipeline.*

**1**

### Discover

CodeQL + Semgrep + heuristics scout the code (7 languages)

**2**

### Investigate

Per-finding LLM analysis with RAG context (ChromaDB)

**3**

### Generate PoV

Model writes a Python / JS / shell exploit script

**4**

### Validate

Run in a language-matched Docker container

**5**

### Oracle & Log

Crash / sanitizer / exit-code signal = confirmed

**Self-healing loop:** if a proof fails, a coordinator analyzes the full attempt history and retries up to 3 times with error context and diversity hints — before logging a final verdict.

# Scale of the Study

| 697 | 26 | 6 | 5,060 |
|---|---|---|---|
| TOTAL SCANS | REPOSITORIES | AI MODELS | TOTAL FINDINGS |

**OFFLINE MODELS · Ollama, local inference**

**qwen3** — 134 scans · 134 confirmed
**glm-4.7-flash** — 139 scans · 90 confirmed
**llama4** — 131 scans · 44 confirmed
**qwen3.6** — 134 scans · 49 confirmed

**ONLINE MODELS · OpenRouter API**

**claude-opus-4.6** — 22 scans · 6 confirmed
**gpt-5.2** — 15 scans · 5 confirmed

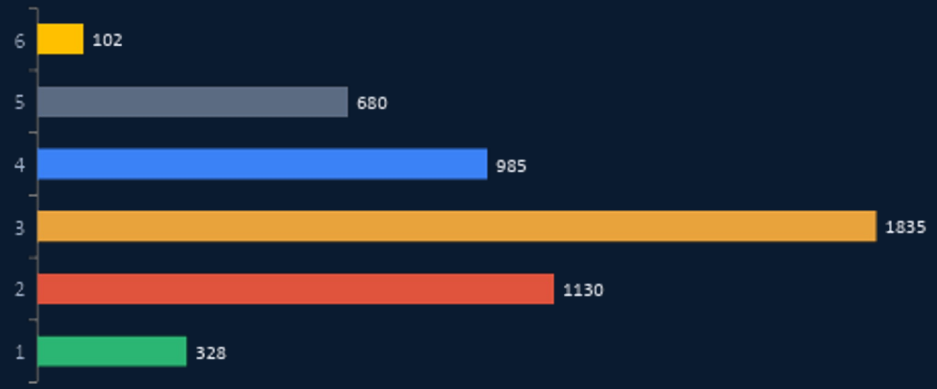
*~1.4M tokens per confirmed vuln — 30× the cost of the best offline model.*

**Repository coverage:** C, C++, Java, JS, TS, Python, Go · 3K–263K LOC · libarchive, curl, tinyxml2, jhead, express, axios, KaTeX

# Overall Results

| 328 | 6.5% | 70 | 77 | 123 |
|---|---|---|---|---|
| CONFIRMED | CONFIRM RATE | UNIQUE CWEs | UNIQUE SITES | REPO·SITE·CWE |

## WHERE THE 5,060 FINDINGS LANDED




### READ THIS AS

**328 vulnerabilities were proven at runtime** — not flagged, but demonstrated with a working exploit.

They span **70 distinct CWE types**, showing the CWE-agnostic approach finds a broad range of bug classes, not one category.

**The 6.5% rate is deliberately conservative** — only crashes we could reproduce count.

# Which Model Wins

BEST OVERALL

## qwen3

*offline · local*

**11.4%** detection rate — best of all 6

**13.3%** PoV success rate — highest

**46.5K** tokens per confirmed — most efficient

*Strong on C / C++ buffer overflows.*

| Model | Detect | FP rate | Tok / confirmed |
|---|---|---|---|
| qwen3 | 11.4% | 5.8% | 46.5K |
| glm-4.7-flash | 7.2% | 4.3% | 193.9K |
| llama4 | 3.9% | 6.2% | 183.2K |
| qwen3.6 | 4.2% | 5.9% | harness only |
| claude-opus-4.6 | 3.3% | 4.9% | 1.45M |
| gpt-5.2 | 3.4% | 28.9% | 1.38M |

**Takeaway:** the best small local model beats the big online models on detection AND is 30× cheaper per confirmed vulnerability.

# Deterministic Harness vs. LLM Exploits

*Two ways AutoPoV proves a bug — and they are complementary.*

**DETERMINISTIC HARNESS**

**55.1%**

**confirmation rate**

256 findings processed · 141 confirmed
7.4K tokens per confirmed
But covers only ~5% of all findings

**BEST LLM (qwen3)**

**10.0%**

**confirmation rate**

1,155 findings processed · 116 confirmed
46.5K tokens per confirmed
Covers broad, novel CWE classes

**Key insight:** pre-built harnesses hit **5.5× the confirmation rate at 6× lower token cost** — but only on the ~5% of findings they cover. LLMs handle the long tail. The winning system uses both.

# Key Findings

1

### qwen3 leads offline models

11.4% detection, 30× more token-efficient than online models.

2

### Deterministic harnesses dominate

55.1% vs 10% for the best LLM, at 6× lower cost — but only 5% coverage.

3

### Small C codebases are most provable

jhead 20.7%, libarchive 11.7%, enchive 10.1% — known bug patterns.

4

### CWE classification is inconsistent

Only 16.9% cross-model agreement; 73.5% even within a model across runs.

5

### The oracle gap is the bottleneck

37% of failures are 'no crash detected' — the highest-impact area to fix.

6

### Online models aren't cost-effective

1.4M tokens per confirmed vuln vs 46.5K for qwen3 — a 30× gap.

# Conclusion & What's Next

Autonomous vulnerability proving is feasible at scale — combining static analysis, LLM reasoning, and runtime validation in one pipeline.

| 328 | 70 | 73.5% | 46.5K |
|---|---|---|---|
| CONFIRMED | UNIQUE CWEs | CWE AGREEMENT | TOK / CONF (BEST) |

## FUTURE DIRECTIONS

- **Oracle refinement** — cut the 37% 'no crash detected' failure class — highest impact
- **Harness expansion** — extend deterministic harnesses to more CWE types
- **Ensemble models** — exploit cross-model disagreement for broader coverage
- **Multi-language PoV** — improve LLM exploit generation for JS / TS / Python targets

# Thank You

AutoPoV — Autonomous Proof-of-Vulnerability Generation & Validation

*Discovery-First · CWE-Agnostic · Runtime-Validated*

---

**Questions?**

SECURENET — Enhancing Cross-Sectoral Collaboration in Cybersecurity · Funded by the EU, Horizon Europe GA No. 101217315