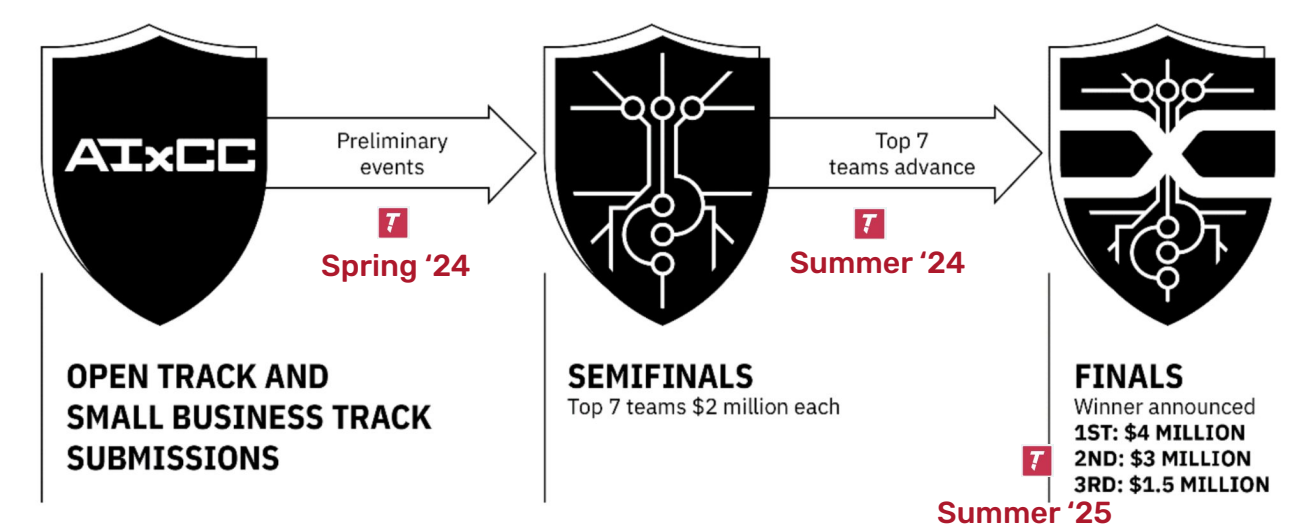
Buttercup and DARPA's AI Cyber Challenge

AI Cyber Challenge (AIxCC)

Competition to create AI systems that find and fix bugs in open-source software



Competition Rules

- **Fully automated solution: no human in the loop**
- **Points for:**
  - Patches - Worth the most points
  - Vulnerabilities - Requires input which triggers bug
  - Static analysis alerts - Minor points
  - Bundling - Match patches, vulnerabilities, and alerts
- **Scoring modifiers:**
  - Speed - Earlier submissions get more points
  - Accuracy - Incorrect/duplicate submissions reduce points
- **Budget**
  - $50,000 LLM API
  - $85,000 Azure

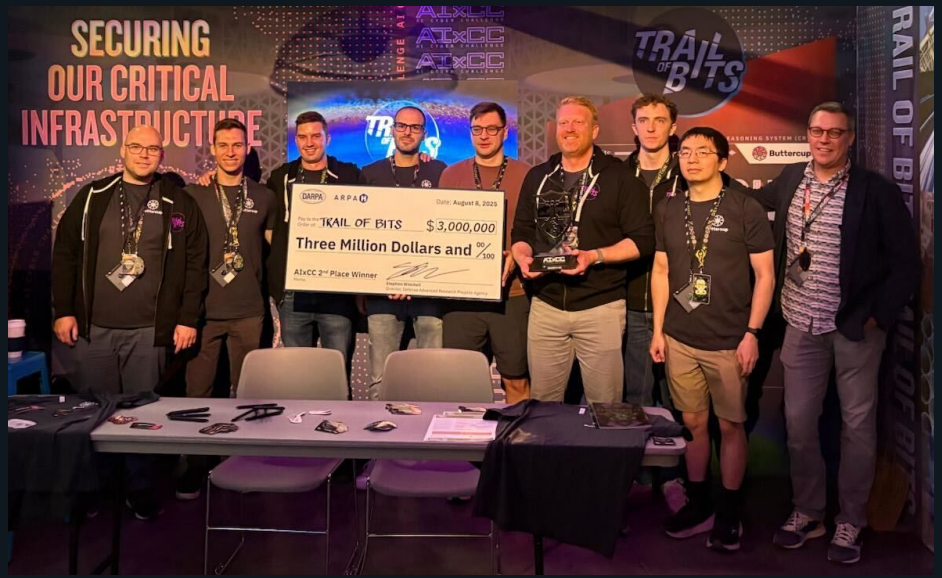
How would you solve this?

Results: $3 Million Prize

Buttercup won 2nd place and a $3 Million prize

Keys to success:

- **Accuracy**
- **Versatility**
- **Reliability**

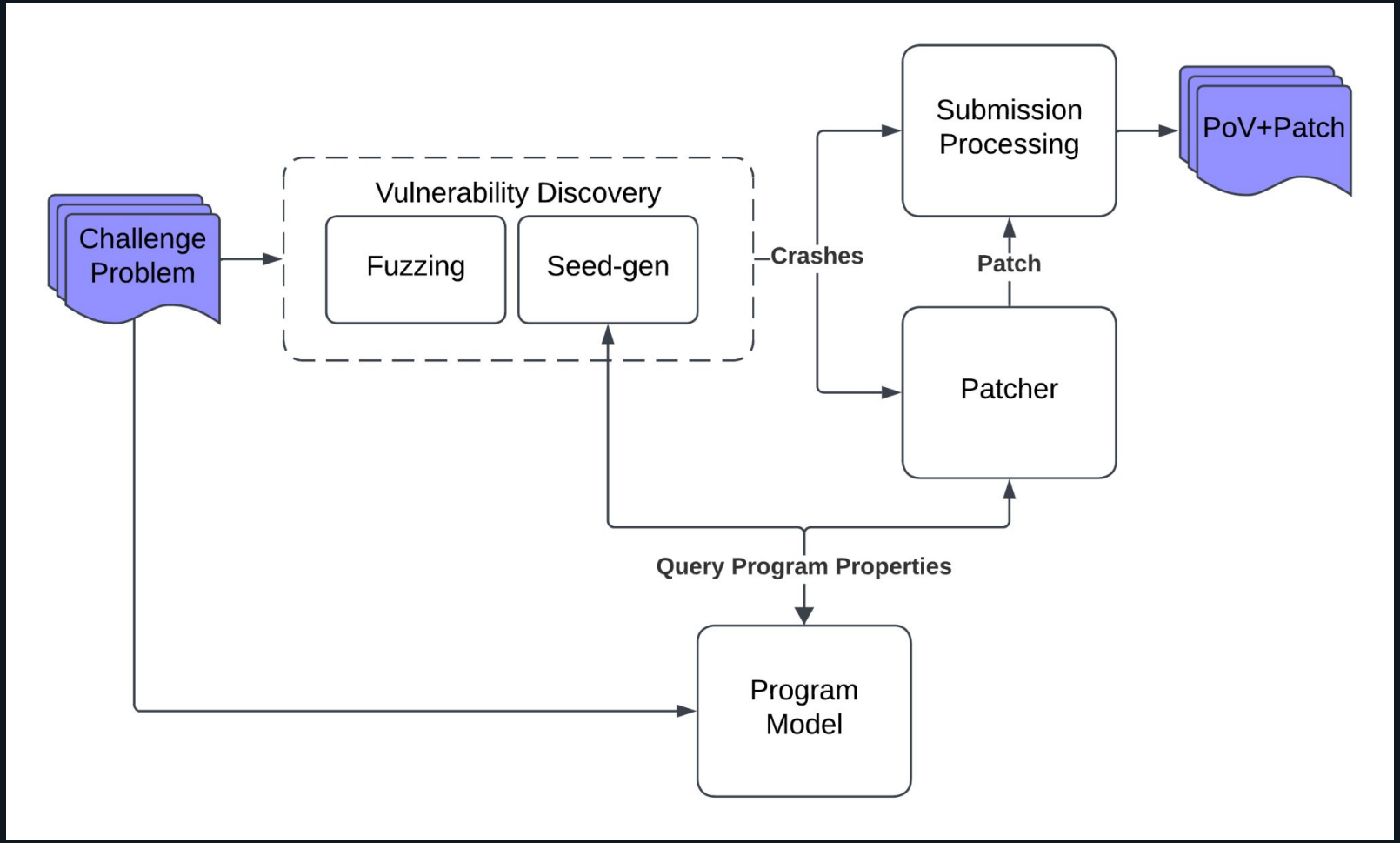


Team	LLM spend	Compute spend	Total spend	Cost per point
Team Atlanta	$29.4k	$73.9k	$103.3k	$263
Trail of Bits	**$21.1k**	**$18.5k**	**$39.6k**	**$181**
Theori	$11.5k	$20.3k	$31.8k	$151
fuzzing_brain	$12.2k	$63.2k	$75.4k	$490
Shellphish	$2.9k	$54.9k	$57.8k	$425
42-b3yond-6ug	$1.1k	$38.7k	$39.8k	$379
LACROSSE	$631	$7.1k	$7.2k	$751

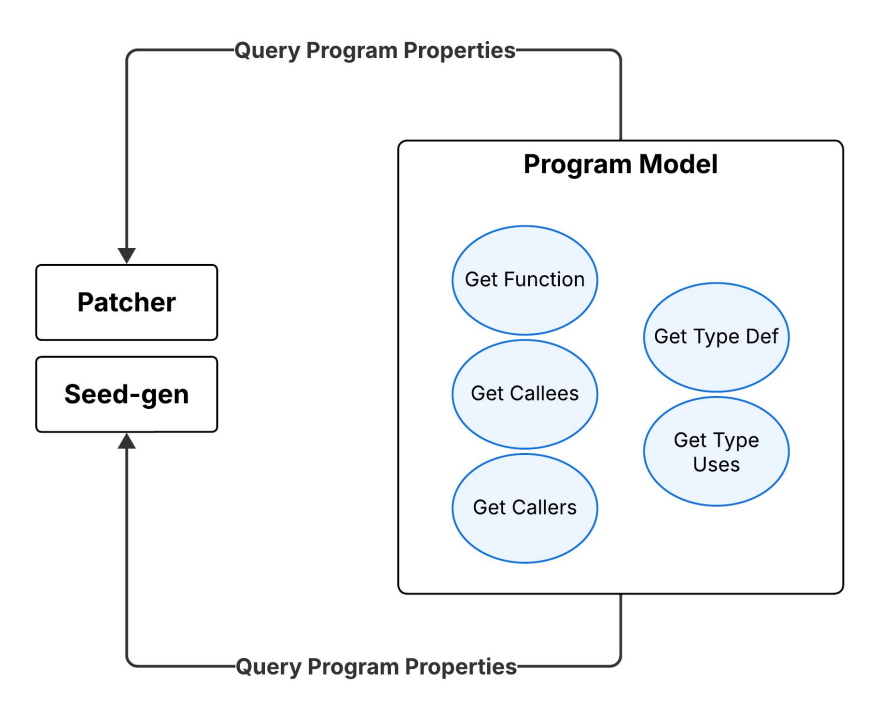
Results!

2nd in LLM Spend
6th in Compute Spend
2nd in $ per Point

Buttercup System Design



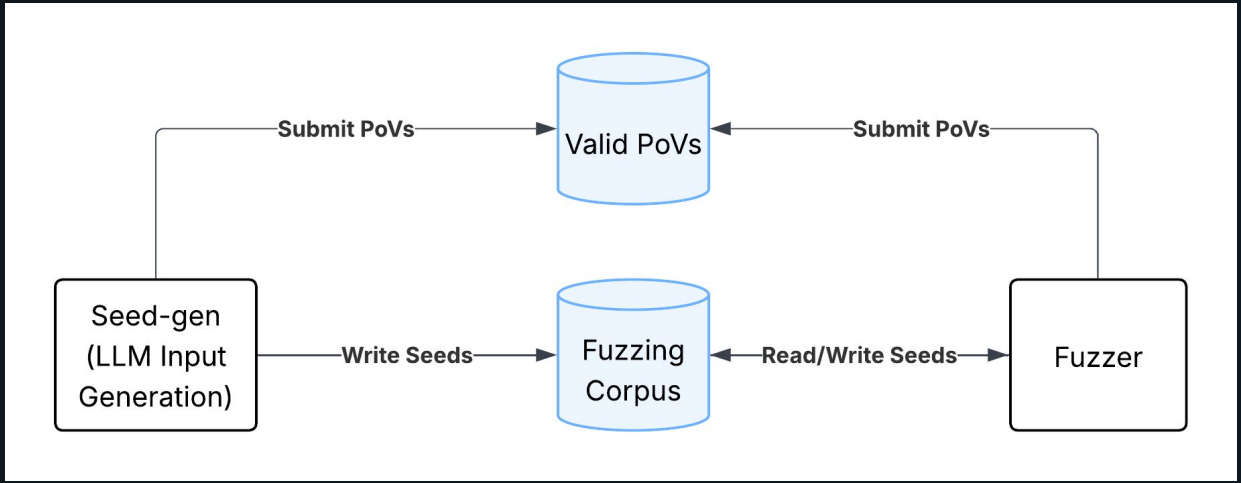
Program Model



- Constructs program model using CodeQuery + Tree-sitter
- Supports querying program properties (functions & types)
- Used by LLM components

Vulnerability Discovery

Approach: Combine fuzzing and LLM input generation



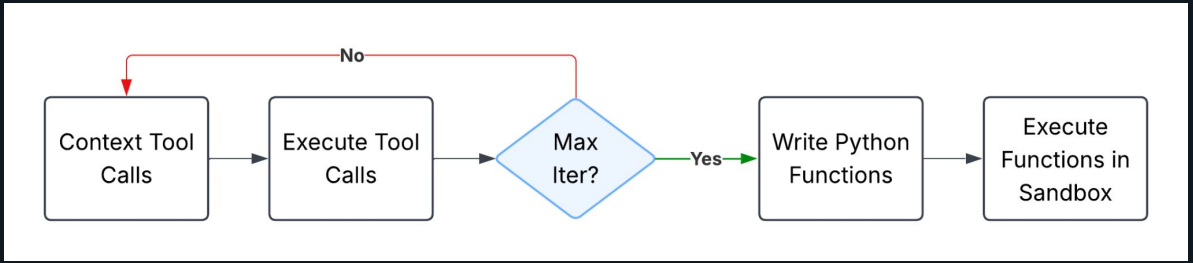
Note: PoVs stands for Proofs of Vulnerability

Fuzzing

- **Standard OSS-Fuzz fuzzers:**
  - LibFuzzer for C/C++
  - Jazzer for Java
- **Fuzzer bots sample active harnesses to run short fuzz campaigns**
- **Merger bots save inputs which improve coverage**
  - Merge a fuzzer bot's local corpus to the shared corpus

Seed-Gen: Enhancing Fuzzing

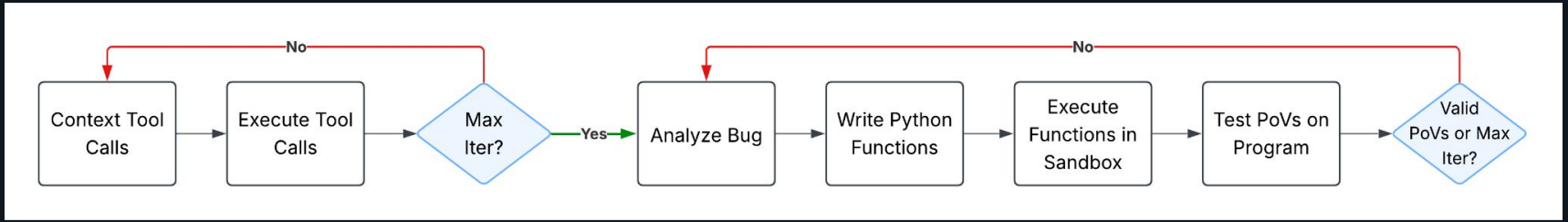
- Generate inputs with LLM to support the fuzzer
- **Initialize Task:** Bootstrap fuzzer with initial seed inputs that exercise harness
- **Explore Task:** Increase coverage for a target function
  - Sample function with low coverage
  - Generate reaching inputs



Initialize and Explore Tasks

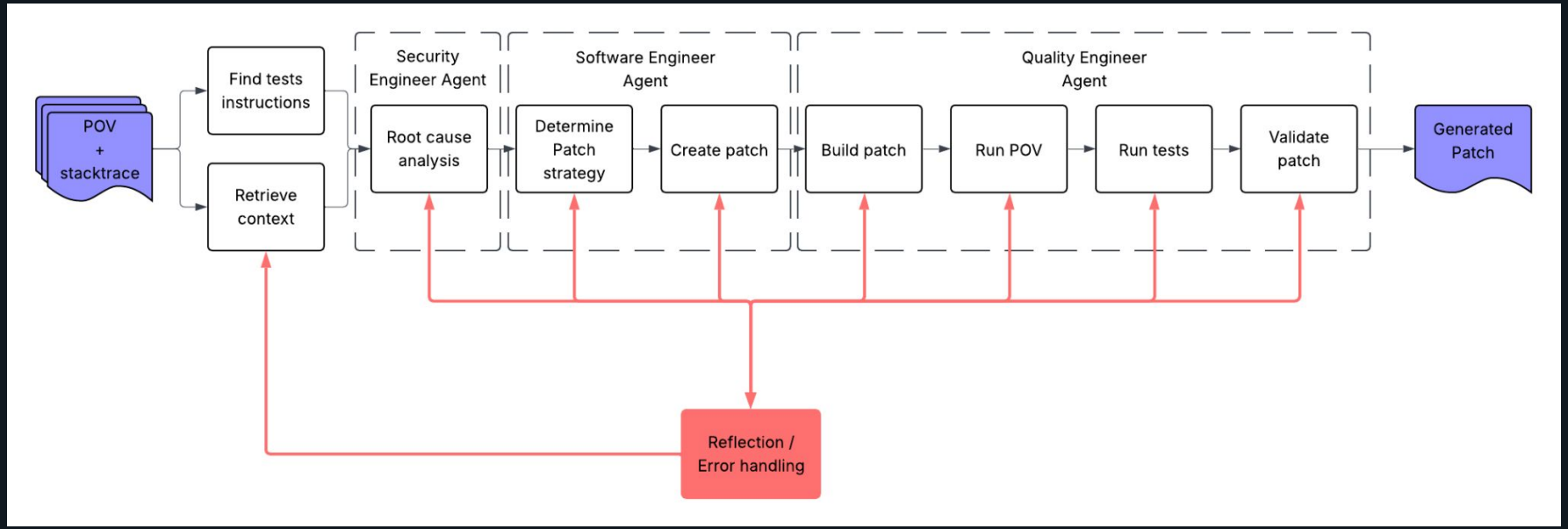
Seed-Gen: Vulnerability Discovery

- **Independently find bugs with an LLM agent**
- **Vuln discovery task: Identify vulnerabilities and create PoVs**
  - Similar to fuzzing

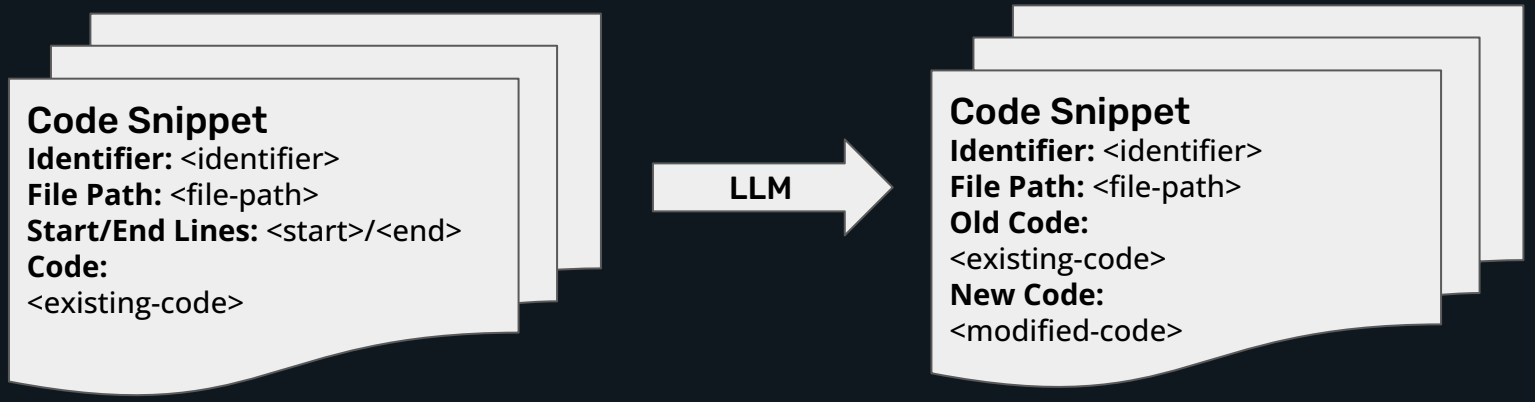


Vulnerability Discovery Task

Patcher Design



Patch Creation



Using Buttercup

Repo: github.com/trailofbits/buttercup

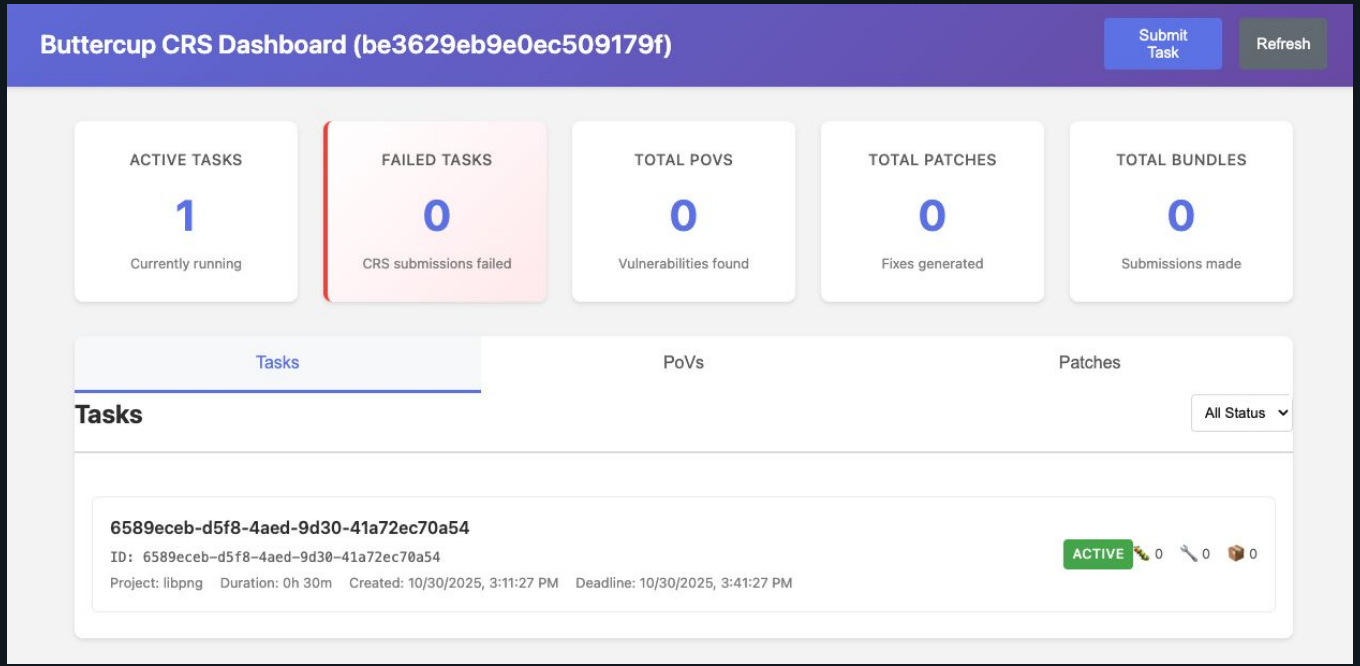
Requires:

- **LLM API Key (OpenAI, Anthropic, or Gemini)**
- **OSS-Fuzz compatible project as a target**

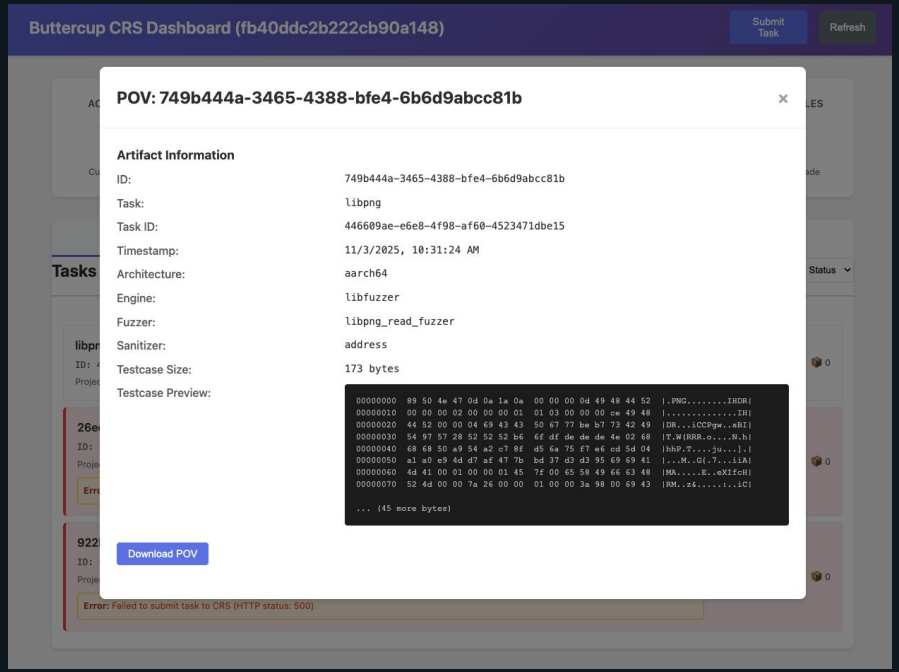
Easy to use:

- **Seamless configuration script**
- **Web UI to task Buttercup and monitor results**

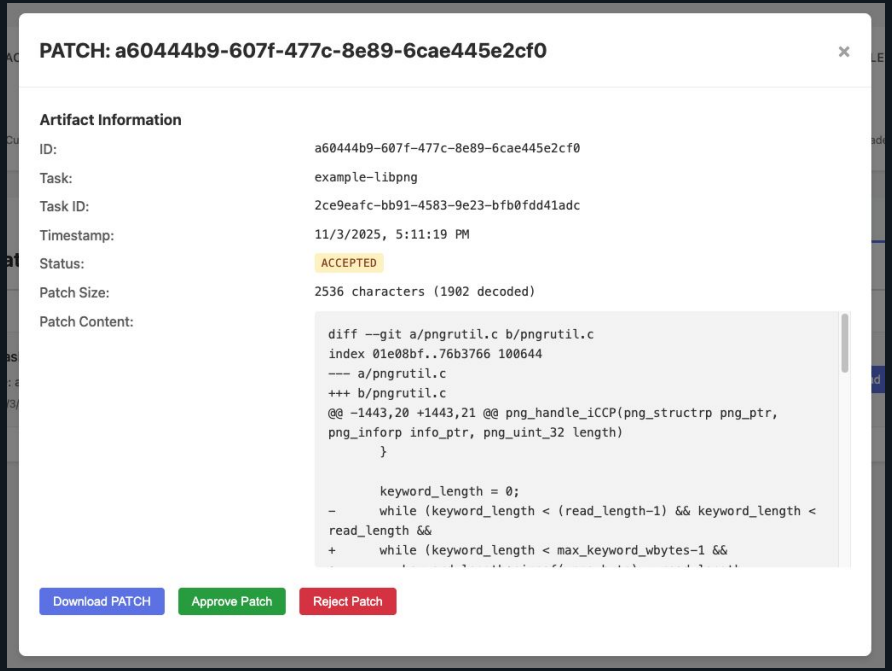
Web UI


Buttercup CRS Dashboard (be3629eb9e0ec509179f)
Submit Task
Refresh
ACTIVE TASKS
1
Currently running
FAILED TASKS
0
CRS submissions failed
TOTAL POVS
0
Vulnerabilities found
TOTAL PATCHES
0
Fixes generated
TOTAL BUNDLES
0
Submissions made
Tasks
PoVs
Patches
Tasks
All Status
6589eceb-d5f8-4aed-9d30-41a72ec70a54
ID: 6589eceb-d5f8-4aed-9d30-41a72ec70a54
Project: libpng Duration: 0h 30m Created: 10/30/2025, 3:11:27 PM Deadline: 10/30/2025, 3:41:27 PM
ACTIVE 0 0 0

Web UI - PoV Review

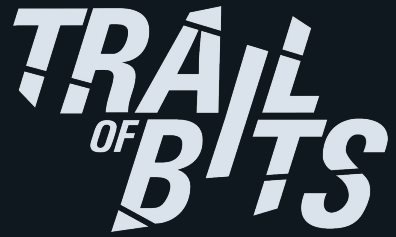


Web UI - Patch Approval



Try out Buttercup:

github.com/trailofbits/buttercup



Website: trailofbits.com

Careers: trailofbits.com/careers

Blog: blog.trailofbits.com