SoK: DARPA's AI Cyber Challenge (AIxCC): Competition Design, Architectures, and Lessons Learned

Cen Zhang[1,5] · Younggi Park* · Fabian Fleischer[1] · Yu-Fu Fu[1] · Jiho Kim[1] · Dongkwan Kim[5] · Youngjoon Kim[1] · Qingxiao Xu[2] · Andrew Chin[1] · Ze Sheng[2] · Hanqing Zhao[1] · Michael Pelican[3]
David J. Musliner[3] · Jeff Huang[2] · Jon Silliman[4] · Mikel Mcdaniel[4] · Jefferson Casavant[4] · Isaac Goldthwaite[4] · Nicholas Vidovich[4] · Matthew Lehman[4] · Taesoo Kim[1,5]

[1] Georgia Institute of Technology · [2] Texas A&M University · [3] Smart Information Flow Technologies · [4] Kudu Dynamics · [5] Microsoft · * Independent Researcher

42 → 7 entrants to finalists · **142.7 h** fully autonomous · **2023–2025** two-year competition · **$29.5M** prize purse · **$135K** compute + LLM credits

Official Archive · SoK Website · OpenSSF OSS-CRS / CRSBench

I · MOTIVATION

Why an AIxCC SoK?

A milestone competition—but no systematic account of its design, systems, or results.

WHO NEEDS THE ANSWERS?

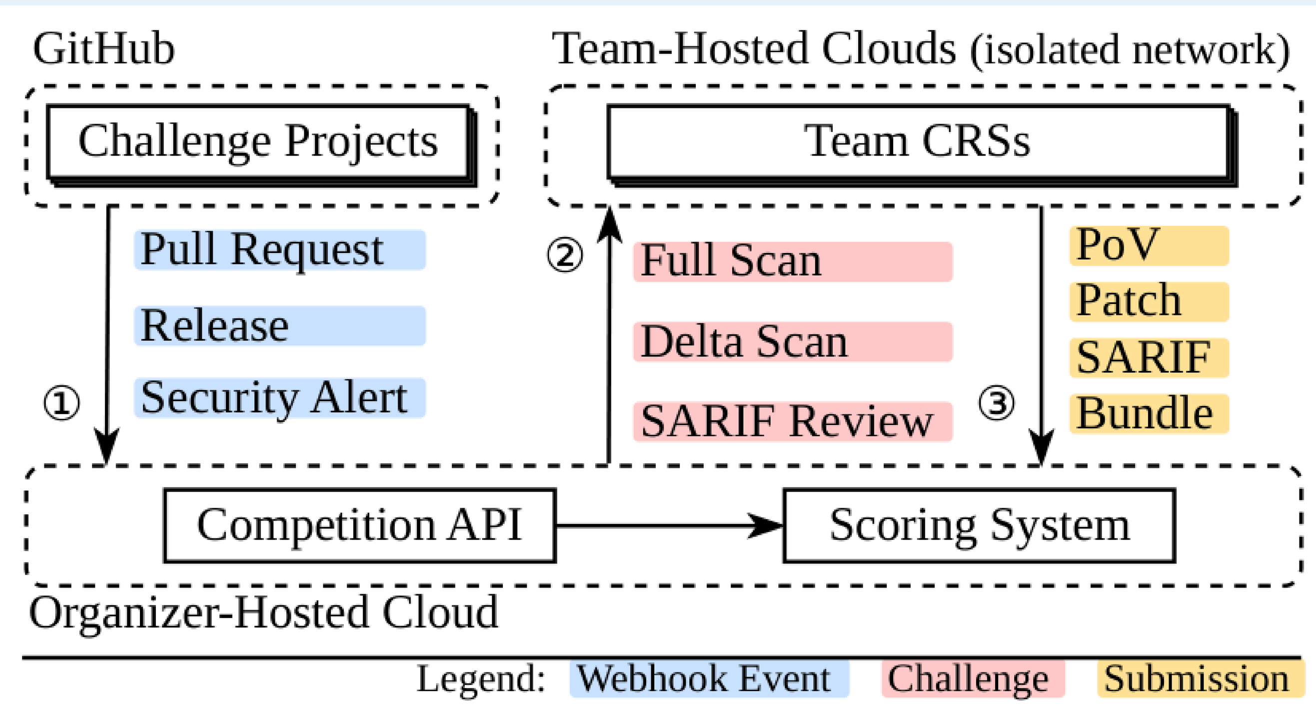
ORGANIZERS	Designing future competitions like AIxCC
RESEARCHERS	Building vulnerability discovery and repair techniques
PRACTITIONERS	Deploying autonomous security systems in practice

What evidence did we use?

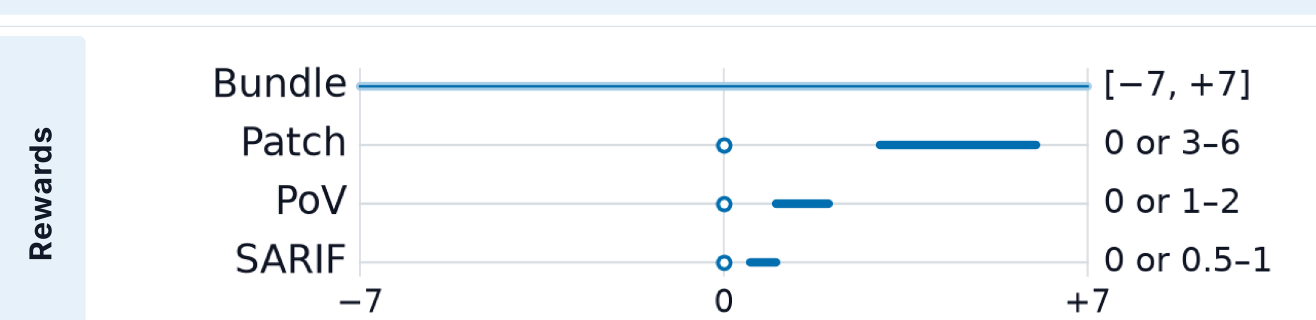
1. **Full competition database** — DARPA-approved access to challenges, logs, traces, and scores
2. **Regular discussions with organizers** — Kudu Dynamics—competition design and organizer reflections
3. **Communication with all seven finalist teams** — One-on-one meetings or a structured questionnaire
4. **Submission-version codebases** — Code as the grounded fact · cross-validation by ≥2 authors

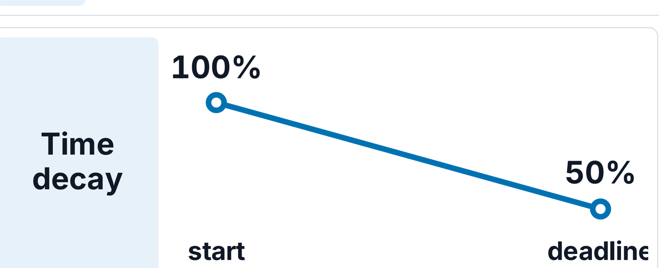
II · COMPETITION DESIGN

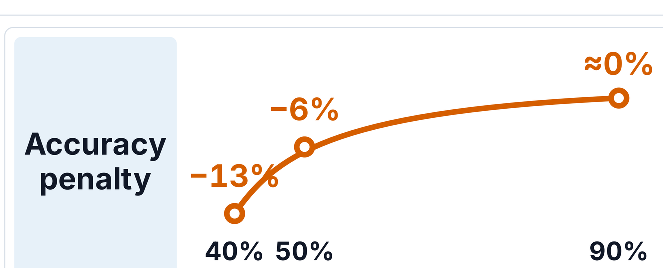
AIxCC ran as a real competition embedded in the GitHub development workflow.



Scoring rewarded autonomous outputs by the developer effort they saved.



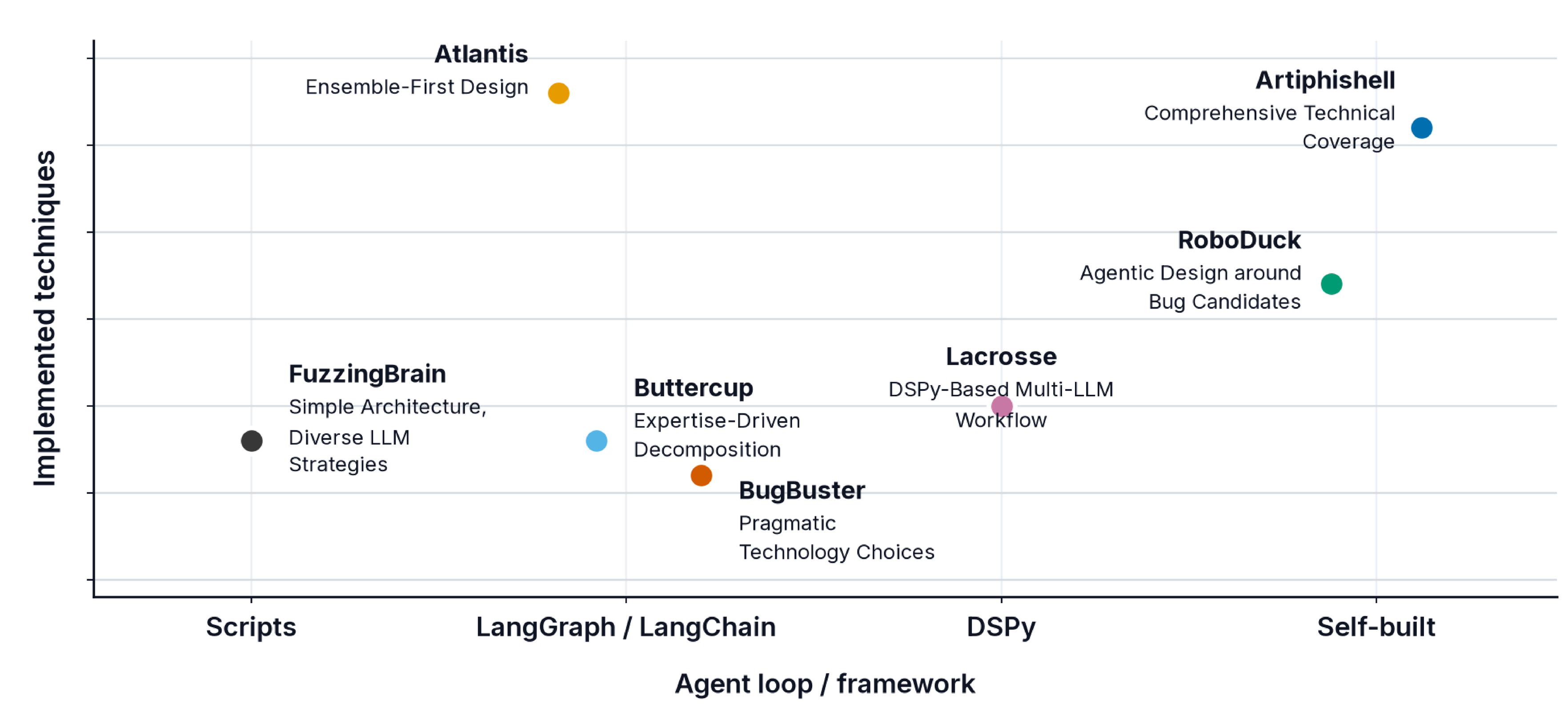




24	53	63	34	301
OSS projects	challenges	injected vulns	CWE types	harnesses

III · FINALIST SYSTEMS TECHNIQUE TAXONOMY

Seven finalist systems



PoV Generation Overview

Two pipelines cooperate through LLM-guided fuzzing and candidate-fed PoV generation.

Fuzzing Pipeline: Seed preparation → Fuzzing engines → PoV

Seed preparation		Fuzzing engines	
Pre-Comp Corpus	5/7	Engine Refinement	4/7
Seed Gen Agent	6/7	Semantic feedback	1/7
Bootstrap	5/7	Improved sanitizer	2/7
Coverage blocker	4/7	Dictionary generation	4/7
Mutator / generator	1/7	Directed fuzzing	2/7
Grammar-aware	3/7	Parallel Fuzzing	7/7
		Corpus sync	6/7
		Added C fuzzers	4/7
		Added JVM fuzzers	1/7
		Concolic Fuzzing	1/7

Pipeline Cooperation: ↑ LLM PoV Gen → Fuzz 5/7 · ↓ Fuzz → LLM PoV Gen 3/7

LLM-Based PoV Gen Pipeline: Bug-candidate analysis → LLM-based PoV generator → PoV

Bug-candidate analysis		LLM-based PoV generator	
Bug Candidate ID	5/7	PoV Gen Agent	5/7
Candidate filter	5/7	CWE guidance	4/7
Non-PoV Gen usage	4/7	Reach-then-exploit	3/7

Patch Generation Overview

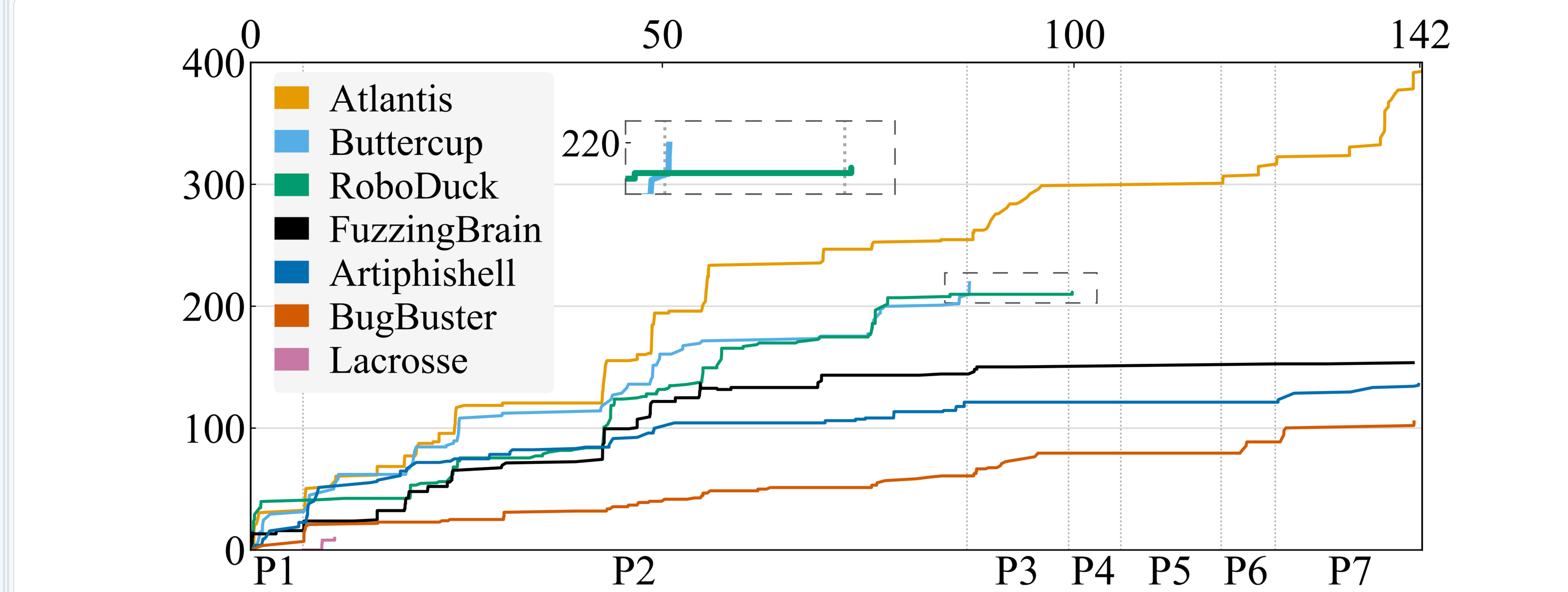
One shared patch pipeline; agent architectures and stage-level techniques differ.

AGENT ARCHITECTURE: Multi-Arch 2/7 · Multi-Agent 2/7 · Single-Agent 3/7

ITERATE: [RCA] → Generate → Validate → Dedup → Submit

[RCA]		Generate		Validate		Dedup		Submit	
Standalone RCA	4/7	Contextualization	7/7	Basic checks	7/7	Minimal patch set	4/7	No-PoV delayed submission	3/7
Multi-PoV RCA	3/7	Code indexer	5/7	Build	7/7				
Non-LLM RCA	1/7	SAST	4/7	PoV test (generation)	7/7				
		CWE guidance	2/7	Project tests	6/7				
		Fine-tuned LLM	1/7	PoV test (submission)	7/7				
		Agentic code search	7/7	LLM-as-judge	3/7				
		Dynamic information	4/7	Post-patch fuzzing	2/7				
		PoV bytes	3/7	Rebuild optimization	2/7				
		LLM reflection	5/7						
		No-PoV patch generation	3/7						

IV · COMPETITION RESULTS ANALYSIS



Scoreboard reveals

- **3 / 7** System stability — Stopped scoring before the final phases
- **−16.3** Submission accuracy — RoboDuck moved from second to third

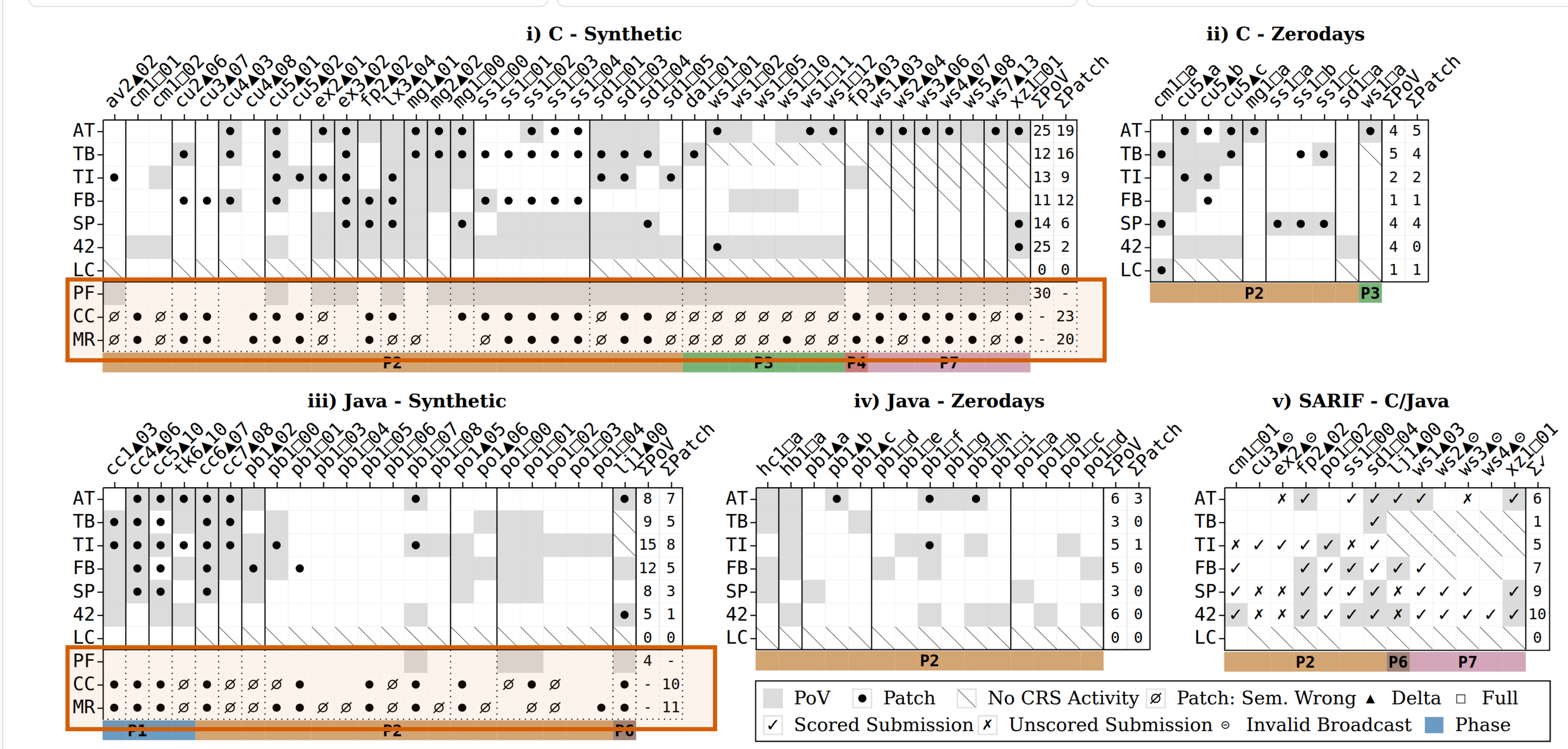
Direct attribution unavailable

- ✗ Ablation? — Prohibitive cost
- ✗ Postmortem? — Incomplete telemetry

Why annotation? Score cannot explain why a team missed → Run representative PoV / patch techniques per challenge → Create an analysis reference for each outcome

Annotation Results Overview

Parallel Fuzzing	MultiRetrieval	Claude Code
34 / 63	31 / 63	33 / 63



Overall findings

- AIxCC challenges emphasize real-world breadth—not extreme per-bug difficulty.
- Reliable use of representative techniques would place an autonomous system in the top three.

PoV findings

	Finalist teams find	Finalist teams miss
PF finds	**PF coverage differs by language** • C: 30 / 40 (75%) • Java: 4 / 23 (17%)	**System robustness issues** • Broken dependencies and resource-heavy builds • Sanitizer or reproduction mismatch • Coarse crash deduplication
PF misses	**LLMs complement coverage-based fuzzing** • Targeted detection: diff-guided triggering inputs • Constraint-heavy inputs: structured formats and semantic guards	**Few unresolved cases** • Complex reasoning: cryptographic transformations • Fuzzing-unfriendly nested timeout/OOM cases

Patch findings

	Finalist teams patch	Finalist teams miss
MR / CC patch	**Automatic checks do not establish semantic correctness** • Build, PoV reproduction, and project tests all pass • CC: 37.7% wrong · MR: 45.6% wrong • High-accuracy teams: still 16–21% wrong	**Patch capability can be lost before delivery** • 9 cases patchable by MR / CC • No PoV, scheduling, outages, or time limits stop the patch loop
MR / CC miss	**Finalist teams solve beyond MR / CC** • 16 additional vulnerabilities patched • Deeper root causes and functionality preservation	**Seven Java vulnerabilities expose two limits** • Reasoning gaps: infinite loops, integer overflow, ReDoS • Rigid localization from timeout / JVM-crash signals

V · Reflections

AIxCC is a trifecta — Research, engineering, and strategy jointly determine outcomes.

Translation is not automatic — Research needs analyzable telemetry and baselines; deployment needs efficient systems and standard interfaces.

Competitions and AI-powered scanning reinforce each other — Competitions foster practical exploration; stronger techniques reshape future competitions.