

CENTRAL TEXAS CYBER RANGE · CYBEARSECURITY LAB · BAYLOR UNIVERSITY

From Measurement to Co-Adaptive Defense

Autonomous Vulnerability Remediation on Live Systems

Abanisenioluwa Kolawole Orojo

PhD Candidate, Department of Computer Science

Advised by Dr. Erika Leal



Security and Operational Objectives

SECURITY OBJECTIVE

Close the exposure

Patch, pin, harden a directive, scope a bind address, revoke a privilege — or, where no vendor fix exists, stand up a compensating control.

OPERATIONAL OBJECTIVE

Keep serving

Every service that was up before the change is still up after it — and still answering under load, not merely bound to its port.

A remediation that closes a vulnerability by breaking the protected service is a failure.

WHERE WE ALL LEARNED THIS

We have all touched these images

2012

Metasploitable 2

2016–2020

Metasploitable 3

2012–2022

VulnHub

2015–2022

CCDC

HS20–23

Hivestorm

2018–2026

**The threats we
answer now**

These artifacts taught us to harden systems. None of them **scores** the hardening.

Three regimes, three different questions

OFFENSIVE AGENTS

**Did the exploit
land?**

SOURCE REPAIR

**Does the build
pass?**

THE OPERATOR

**Is it closed, and
is it still
serving?**



THE GAP

Nobody scores the cell the operator works in

SYSTEM / LINE OF WORK	LIVE EXEC	DUAL OBJ	MULTI-HOST	ROLLBACK	ADV. INPUT
SWE-bench line	—	—	—	—	—
DARPA AIxCC	✓	✓	—	—	—
CVE-Bench · PentestGPT	✓	—	—	—	—
CybORG	sim	~	✓	—	—
CAGE-4	sim	~	✓	—	—
Sys-Repair Bench	✓	✓	✓	✓	✓

Dual obj. = gates on service availability alongside security. ~ = a scalar reward where availability is tradable. Adv. input = evaluates robustness to poisoned remediation inputs.

SysRepair-Bench

313 scenarios · six suites two decades of operational threat

SUITE & ERA		
ccdc	50	2015–22
meta2	40	2008–12
vulnhub	30	2012–22
meta3/ubuntu	19	2014–20
meta3/windows	21	2016–20
meta4	137	2018–26
hivestorm	16	HS20–23

CVSS V3.1	
Critical	94
High	106
Medium	44
Low	1
Unscored	68
Critical and High together are 200 of 313 — operational triage priorities.	

REMEDIATION PRIMITIVE	
Config hardening	113
Dependency & pkg	71
Access control	64
Compensating ctrl	39
Network security	10
Free-roam	16
No single template — apt upgrade or systemctl restart — can dominate.	

39 scenarios forbid the direct fix outright — legacy dependency, end-of-life software, no maintenance window — and oblige a compensating control instead.

THE SCORING

Scoring conjunction

block 1

Proof of concept

The canonical exploit is re-run and must fail. This is the security objective.

block 2

Regression

Every service operational before the change is exercised again and must still respond.

block 3 · 39 scenarios

Attack surface

The required mitigation is present: a firewall rule, a loopback bind, a removed directive.

```
verify.sh → exit 0
```

Pass is the conjunction. An agent that closes the vulnerability by killing the service it was protecting fails outright.

Harness: Inspect AI, a single bash tool, no Python and no web access — the affordances of a responder in a competition round. 1800 s per scenario, success@1 and success@5 over five seeds.

Agents are weakest exactly where incident response is hardest



success@5 (%), report-informed, pooled over MiniMax-M3, Qwen3.5-35B-A3B and Qwen3.5-9B under ReAct.

The two reasoning-heavy categories trail access control by 25.4 and 38.7 percentage points. A blanket unattended-upgrades sweep clears 2 of 313 . This is not a stale-package problem.

Tell the agent what to look for and modern CVEs gain 23–35 points



Δ success@1, report-informed minus black-box, percentage points. Hivestorm is excluded — randomized artifact identities preclude a threat description.

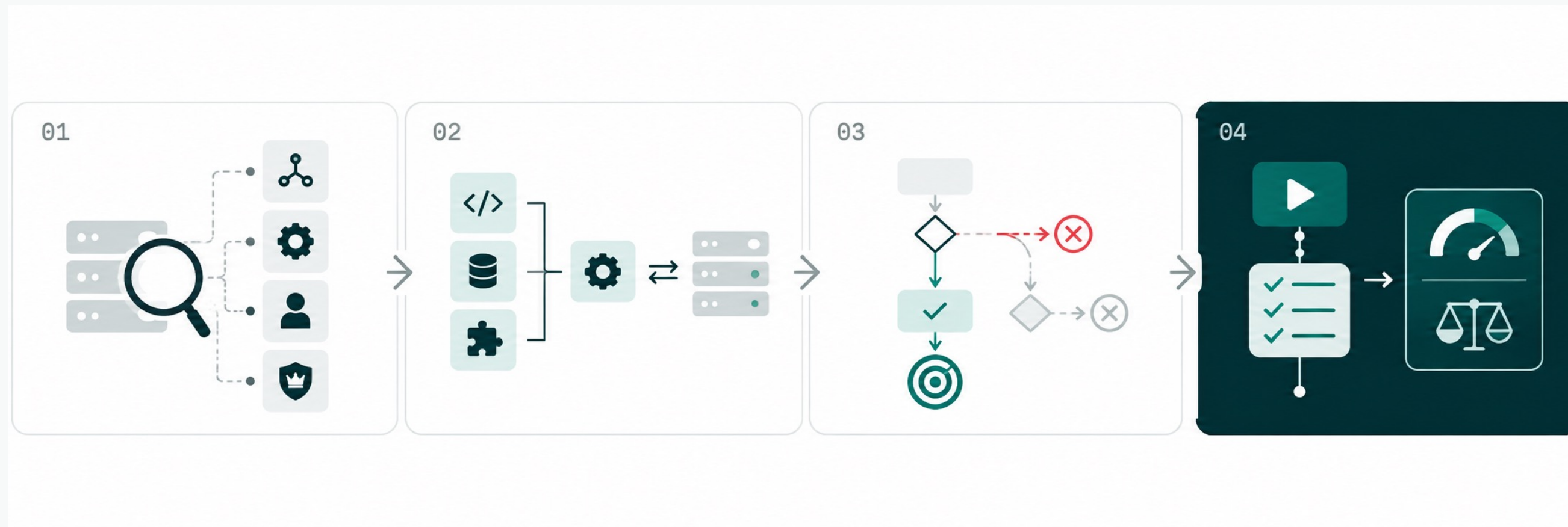
WHAT THE LIFT TELLS US

Threat discovery is the binding constraint on modern CVEs.

Once the agent knows what it is looking at, the remaining work — locate the artifact, pick the primitive, verify — is largely tractable. On legacy misconfiguration suites the gain is smaller because the artifact is the threat, so discovery and repair are one step.

This is a perception-versus-action decomposition, and it is the one the rest of the program inherits.

NeuroPlan: review the plan before it touches the host



96%

PDDL domain validation

88%

Plan returned

76%

End to End

The two pre-execution stages are [safe refusals](#) : the solver halts with a structured outcome before any change reaches the host.

Reviewability vs Cost vs Accuracy

SOLVER	DISTINGUISHING PROPERTY	OPERATING POINT
Reflexion	Highest single-shot accuracy	90% pass@1
ReAct	Best recovery under retry	94% pass@5
Plan-and-Solve	Lowest token cost per success	66.6% · ~38× cheaper
Basic	Fixed pipeline reference	50.4% pass@1
NeuroPlan	Pre-execution plan reviewability	76% end-to-end

Each solver is best against a different deployment property, so the right choice depends on which property the deployment requires. Removing the threat briefing collapses performance — which is the same perception-versus-action split, seen from the solver side.

Cross-host reasoning, and red/blue self-play

01

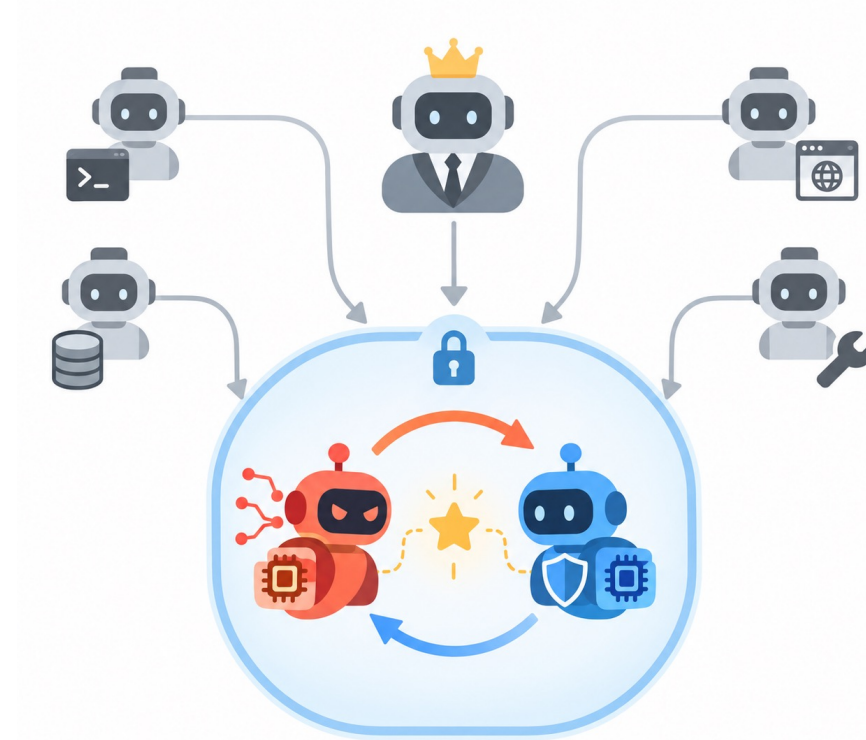
Multi-host and cross-host reasoning



And availability measured under load – latency and error-rate SLOs.

02

Reinforcement-learning self-play, red and blue



The open question is whether a defender trained this way generalises to threats its attacker never found.

THE THROUGH-LINE

Building Secure Scalable Agents and Harnesses For System Remediation

01 · survey

A closed-loop task model, and
seven named gaps.

02 · benchmark

313 dual-objective scenarios on
running services.

03 · planner & RL

Reviewable plans, and policies
that co-adapt.

04 · system

Safety as something you
evaluate.

THANK YOU

Questions

CO-AUTHORS

Webster C. Elumelu · Emmanuelli El-Mahmoud

ADVISOR

Dr. Erika Leal Department of Computer Science, Baylor University ·
Director of Research and Development, Central Texas Cyber Range

LAB

Cybearsecurity Lab · Central Texas Cyber Range, Waco, Texas

ARTIFACTS, OPEN LICENCE

labackdoor.com/publications

github.com/CybearesSecurityLab/sysrepair-bench

github.com/CybearesSecurityLab/neurosymbolic-sysrepair

github.com/AK0rojo/autopatch

-
- Microarchitectural Weird Machines
 - AI for Cybersecurity
 - Malware U