
 TECHNICAL WHITEPAPER · ISSUE 1

Autonomous Agentic Red Teaming

Preliminary validation and post-remediation verification

Subject: Kolega's agentic penetration testing platform, operated as a service for adversarial validation

Audience: security architects, application security leads, penetration testing managers and risk officers

AUTHORS

John Pellew
john@kolega.ai

Faizan Raza
faizan@kolega.ai

ISSUED BY



COMPANION DOCUMENTS

Rings of Risk (working draft, 22 Oct 2026)
RealVuln Benchmark Comparative Report v3.1
Continuous Architectural Resilience Framework v1.0

ISSUED

23 October 2026
Preliminary validation

READ FIRST

The status of these claims

This is a preliminary validation report, and the word is doing real work. Of the six claims below, two are validated against published evidence and one is demonstrated by a study others can review. The remaining three describe a method that is built and being measured, not a benchmarked result. No client engagement results appear here, and nothing in this document claims performance on any bank's production estate. Section 8 sets out what would falsify each claim and what evidence will settle it.

SUMMARY

Executive summary

Static scanning tells you whether the code contains a weakness. That is necessary, but it is not the question a security committee is actually asking, which is whether an adversary can reach a protected outcome from somewhere they could realistically start.

The gap between those two questions is where assurance programmes tend to come apart, because answering the second one means running something. Scanners do not run anything.

The Red Team Agent is our attempt to close that gap. It is an agentic penetration-testing capability, and a deliberately aggressive one. It maps an application's ingress points and identity positions, forms hypotheses about multi-step routes between them, then attempts those routes under authorization in an isolated environment. When a route works, it produces a reproducible proof of concept and a Rings of Risk finding report. After remediation it re-executes the same path, which is how we establish that a path is actually closed rather than closed on paper.

TABLE 1 Claim status. Read this before any other table.

CLAIM	STATUS	BASIS
1. A security-specialized scanner can detect a high proportion of real vulnerabilities, including access-control flaws	■■■■ Validated	RealVuln v3.1: F3 84.4 strict, 90.2% recall across 140 repositories and 4,138 labelled vulnerabilities; 87.9% recall achievable on broken access control by the best configuration.
2. Multi-flaw routes to serious outcomes can be found by systematic analysis of real applications	■■■□ Demonstrated	Rings of Risk study: 246 source-supported route records across 15 open-source applications, 178 labelled B1 Critical, and 54 exact-ID joined path records across four applications.
3. An agent can execute a multi-step chain and confirm it, rather than proposing it	■■□□ In validation	Method implemented and documented (§4). Adjudicated case set and measured confirmation rate pending (§8.2).
4. A generated proof of concept is reproducible by a third party	■■□□ In validation	Artefact specification and reproducibility protocol defined (§5, §8.2); measured reproducibility pending.
5. Post-remediation closure can be determined by re-execution rather than by assertion	■□□□ Method defined	Closure-state taxonomy and verification loop defined (§6); measured closure-detection accuracy pending.
6. Results generalise to a bank's production estate at scale	□□□□ Not claimed	Provisionally contradicted for general-purpose scoring at scale: recall on 263k-816k-line systems measured at 1.77%. Any claim must be earned per engagement.

Where that leaves us: the detection layer is benchmarked and strong. The reasoning layer has been demonstrated on real code by a study others can review. The execution and closure layers are engineered but not yet measured in public. Read this as a specification of what the capability does and how it will be judged, not as a performance claim.

The confirmation gap

Two separate bodies of evidence frame the problem this capability exists to solve.

Detection is incomplete, and predictable in where it fails. In RealVuln v3.1, rule-based scanners found none of the 232 broken-access-control findings and none of the 173 missing-authentication findings. Semgrep missed 3,728 of 4,138 labelled vulnerabilities. The best general-purpose AI configuration still missed 755. A programme that leans on a single detection layer ends up holding an inventory of what was visible, not of what is reachable.

Source analysis can find routes. It cannot prove them. The Rings of Risk study traced 246 source-supported routes across 15 applications, 178 of them labelled B1 Critical, and joined 54 paths across four. The study's own conclusion is the part that matters: those routes are candidates, and "none is an observed or demonstrated attack, and the applications were not run." Source review can show you where a handoff sits. It cannot show that the credential, token or session state that handoff needs will actually line up when the code runs.

WHAT THE GAP ACTUALLY IS

A route stays a hypothesis until something attempts it. Better pattern matching does not close that gap, and neither do more findings. What closes it is **authorized execution of the composed path, with evidence attached**. That is the narrow job of an agentic red team, and it is why we run this as a service instead of shipping it as a scanner.

Design principles

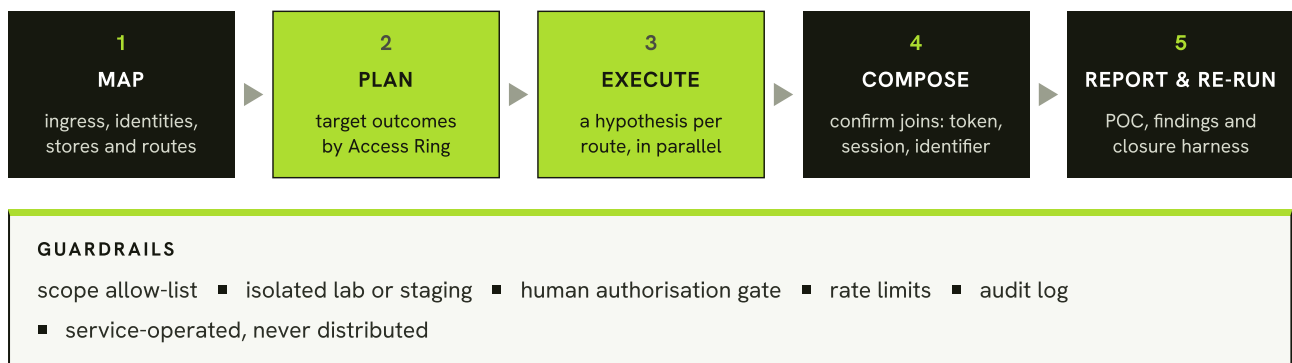
- 1 **Adversarial parity.** If adversaries use agents to compose paths, the defence has to be tested by an agent doing the same thing at a similar tempo. Quarterly testing by humans cannot cover the space an agent covers continuously.

- 2 **Chain-first, not finding-first.** The unit of work is a route between positions, expressed in Access Rings and consequence bands. A finding that confers no position gets recorded, but it is not prioritised as though it were an exploit.
- 3 **Evidence over scores.** A confirmed chain produces a reproducible artefact, not a confidence rating. If a third party cannot re-run it, we do not call it a result.
- 4 **Prove closure, do not assume it.** The same artefact that confirms a path is the instrument that verifies its removal. Remediation is complete when the path fails, not when the ticket closes.
- 5 **Operate, do not ship.** The capability attempts exploitation, on purpose. Unattended self-service would be an unacceptable risk model for that, so we operate it as a service (Section 9).

SECTION 3

Architecture

FIGURE 1 The five-stage pipeline and its guardrails



Every confirmed chain leaves behind a reproducible artefact and a harness that can be re-run, which is what post-remediation verification is built on.

Every stage sits inside the same guardrail layer: scope allow-list, isolation, human authorization for each action, limits on rate and blast radius, and a full audit log of every action taken.

Map and model. The agent builds a working model of the target: ingress points, authentication surfaces, identity tiers, data stores and the routes between them. This is the same vocabulary the Rings of Risk framework uses, so an automated result and a manually reviewed map can be compared directly.

Plan. Rather than sweeping for findings, the planner picks target outcomes (an AR1 or AR2 position, a B1 consequence band) and works backwards to plausible starting positions. That inversion is what makes the search tractable. Outcomes are few and known; findings are many and ambiguous.

Execute. Each worker agent pursues a single hypothesis along a single route, in parallel and with narrow intent. A worker that cannot advance reports the blocking condition (a missing credential, a role mismatch, a rejected join), and that report is itself useful evidence, because a rejected join is what separates a candidate path from a real one.

Compose. This is where most automated approaches stop, and it is where this one earns its keep: **reconciling preconditions across steps.** A token issued at step one has to be accepted at step three. A session established at step two has to carry the role that step four assumes. Chains that fail that check are recorded as rejected, with the reason.

Confirm, report and re-run. A successful route is minimised, evidenced, and packaged as a proof of concept together with the closure harness that will re-test it later.

WHERE THIS CAME FROM

The approach builds on a compilation of current and evolving practices as found in today's agentic hacking tools which are used to exploit the Rings of Risk attack chains. The platform gives us the planner-and-worker agent structure, linked asset and finding graphs, human review of actions, and a retest capability.

Our methods include: **reconciling preconditions across steps, generating a proof of concept per chain, and producing Rings of Risk reporting for engineering and for executives.** Kolega operates the capability and does not distribute it. This document describes behaviour and method, not source code.

■ SECTION 4

The execution method

Execution runs through six stages. The discipline sits in the exit criteria. **A stage that cannot meet its criterion ends the route; it does not get downgraded into something that sounds plausible.**

TABLE 2 Execution stages and exit criteria

STAGE	WHAT HAPPENS	EXIT CRITERION
1. Enumerate	Ingress, identities, sessions, tokens and data stores mapped; scope reconciled against the written authorization.	Every target in scope is represented; anything out of scope is excluded and logged.
2. Hypothesise	Routes proposed from observed weaknesses: which finding supplies which position, and what the next step needs.	Each hypothesis names its starting ring, required preconditions and intended outcome.
3. Attempt	The route is executed step by step against the isolated target, with each action authorized and logged.	Either the outcome is reached, or a specific step is recorded as blocked with the observed reason.
4. Reconcile	Preconditions across steps are checked against what each prior step actually produced.	Every join is satisfied by an observed artefact, not an assumed one.
5. Minimise	Redundant steps removed; the shortest route that still reaches the outcome is retained.	No remaining step can be removed without breaking the outcome.
6. Package	Proof of concept, evidence bundle, chain record and closure harness produced.	Artefact re-runs successfully from a clean state in a second execution.

What counts as confirmed

A route is reported as a confirmed chain only when all five conditions hold. Anything short of that is reported as a candidate, and labelled as one in every downstream artefact.

C1	Reproducible	The route succeeds twice from a clean starting state, not once.
C2	Minimal	No step can be removed without the outcome becoming unreachable.
C3	Condition-stated	Starting ring, credentials, victim action and configuration are recorded explicitly.
C4	Evidenced	Requests, responses, relevant logs and timestamps attached, with hashes.
C5	Non-destructive	No data modified beyond the minimum needed to demonstrate the outcome; cleanup recorded.

TABLE 3 Chain record schema (fields present in every confirmed chain)

FIELD	CONTENT	PURPOSE
Identifier	Stable ID, target, campaign and date	Traceability across reports and re-runs
Ordered steps	Each step with its finding reference and artefact	Makes the chain auditable and minimisable
Entry / terminal ring	AR5-AR1 at start and at outcome	States what the chain actually confers
Consequence band	B1-B4 for the terminal outcome	Separates harm from position
Preconditions	Credentials, roles, victim actions, configuration	Allows a reader to judge applicability
Reach ring	RR1-RR4	Distinguishes chokepoint steps from incidental ones
Evidence refs	Hashes of the evidence bundle	Integrity and reproducibility
POC ref / closure state	Artefact link; current closure state	Connects detection to verification

■ SECTION 5

Proof-of-concept generation

A finding tells an engineer where to look. A proof of concept tells them what to fix and, later, whether they fixed it.

Every confirmed chain is therefore packaged as a reproducible artefact rather than a narrative. The package contains, at minimum:

- **An ordered, re-runnable sequence:** the exact requests or script, parameterised for the target environment, with the expected response at each step.
- **Preconditions:** the accounts, roles, tokens and configuration assumed, and how to establish them in a test environment.
- **Expected versus observed:** what a non-vulnerable system would do at the decisive step, and what this system did.
- **Evidence bundle:** raw responses, relevant log excerpts, timestamps and hashes.
- **Cleanup and safety notes:** what the run touched, and how to return the environment to its prior state.
- **A pass condition:** the precise observation that means "closed" on a re-run, so closure is a measurement rather than an opinion.

That last item is the one that changes the economics. Because every chain arrives with a pass condition a machine can check, post-remediation verification is not another penetration test. It is a targeted re-execution of a known path: faster, cheaper, and directly comparable with the original result.

■ SECTION 6

Post-remediation verification

The verification loop is simple to state and easy to fake, so the taxonomy is deliberately unforgiving. A path that no longer works is not automatically closed, because a fix can block one route and leave another open to the same outcome.

TABLE 4 Closure states

STATE	DEFINITION	OPERATIONAL CONSEQUENCE
■ Closed	The re-run fails at the same decisive step, and no alternative known route to the same outcome succeeds.	Chain may be retired; closure record issued with evidence.
▣ Partially closed	The original path is blocked, but another known route reaches the same outcome.	Outcome remains reachable; reopen against the alternative route.
■ Still reachable	The re-run succeeds unchanged.	The fix did not address the path; escalate to the outcome owner.
◆ Displaced	The original path is blocked, but re-execution reveals a different route to the same or a deeper ring.	Record as a new chain; the fix changed the route, not the exposure.
■ Regressed	The path was previously closed and now succeeds again.	Control failure; treat as an incident of the assurance process itself.

Two design decisions follow. **Verification runs against the outcome rather than the finding**, so a partial fix shows up as partial instead of as success. And **the closure harness stays with the chain record**, which leaves the institution with a growing regression suite of paths that must never reopen.

Over time that suite is probably the most useful thing the programme produces: **a testable definition of what has to stay unreachable.**

WHAT "VERIFIED" DOES AND DOES NOT MEAN

A *closed* verdict means every known route to that outcome failed when re-run. It does not mean the outcome is unreachable by a route nobody has found yet. Every closure statement in a Red Team Agent report says so explicitly, and we hand over the full chain and route inventory so the claim can be tested.

■ SECTION 7

Reporting

Each campaign produces two documents from the same evidence, because the two audiences need different things from it.

TABLE 5 Report structure

FOR ENGINEERING Engineering finding report Per confirmed chain: ordered steps, affected components and line references, precondition set, reproduction instructions, evidence bundle, remediation guidance at the step that breaks the chain, and the pass condition for closure. DECISION IT SUPPORTS What to fix, in what order, and how the fixer will know it worked.	FOR THE EXECUTIVE Executive Rings of Risk report Outcome profiles by consequence band; positions reached; confirmed routes per outcome; chokepoints; known-route closure sets; residual AR1/AR2 reach; the campaign's rejected candidates with reasons. DECISION IT SUPPORTS Where the institution stands, what one fix would buy, and what remains reachable.
--	---

Both reports include rejected routes with their reasons. A candidate rejected because a required credential is never issued is evidence of a control doing its job, and leaving it out of the report would make the confirmed findings harder to trust, since the reader could not tell whether any candidate had ever been rejected.

SECTION 8

Preliminary validation

This section sets out what is established, what is being measured, and what would prove the capability wrong. It is meant to be checkable rather than reassuring.

8.1 What is already established

The detection layer is benchmarked. RealVuln v3.1 evaluates 35 scanner runs on 140 repositories with 4,138 labelled vulnerabilities and 280 false-positive traps, under a published manifest and open scoring code. The security-specialized scanner family that underpins our detection ranks first (F3 84.4 strict, 90.2% recall), and the benchmark reports where it is beaten: on broken access control, the best general-purpose configuration achieved 87.9% against DevSec Max's 66.4%. These are the numbers an engineering team should use to size the detection layer.

The reasoning layer is demonstrated on real code. The Rings of Risk study produced 246 source-supported route records and 54 exact-ID joined paths across 15 applications, with rejected joins documented alongside accepted

ones. It is a working draft, and it is peer-reviewable in the ordinary sense: the ledgers, the classification rules and the audit data are published.

8.2 What is being measured, and how

The adjudicated case set is the obvious instrument, and the Rings of Risk study names it as its own next evidence step. Take the six selectively hardened forks from that corpus, where a documented fix exists against a known baseline, and ask the agent to do three things: find the routes the manual study found, confirm them by execution, and determine closure after the documented fix is applied. Because the manual study already recorded what should be found and what should remain, the case set has a ground truth that does not come from the tool being tested.

TABLE 6 Validation metrics and their falsification conditions

METRIC	DEFINITION	RESULT THAT WOULD FALSIFY THE CLAIM
Route rediscovery	Share of manually recorded routes the agent independently finds.	Low rediscovery on routes already known to be findable.
Confirmation precision	Share of reported confirmed chains that an independent operator reproduces.	The majority of "confirmed" chains fail independent reproduction.
False-path rejection	Share of impossible joins correctly rejected, with reason.	Joins that cannot work are reported as confirmed chains.
POC reproducibility	Share of POCs that run unmodified in a clean environment.	Artefacts require an operator's tacit knowledge to work.
Closure-state accuracy	Agreement between reported closure state and an independent re-test.	Partial closures reported as closed.
Time-to-verified-close	Median elapsed time from confirmation to a failing re-run.	No material improvement over the institution's current cycle.

WHAT THIS DOCUMENT DOES NOT CLAIM

No client campaign results are cited, because none of them are being published at this stage. Nothing here claims that the agent finds all reachable routes, that it replaces human penetration testers, or that a clean campaign means an application is secure. The output is a set of confirmed chains and their closure states. That is a **bounded, evidenced statement about what was found and re-tested, and nothing wider.**

■ SECTION 9

Operating constraints and rules of engagement

The capability attempts exploitation. That single fact drives every constraint below, and it is why the capability is not offered for client self-service.

Written authorization precedes any action

Scope, targets, time windows, excluded systems and stop conditions are agreed in writing; the scope allow-list is enforced technically, not by instruction.

Isolated environments only

Staging, a designated lab, or a replica. **Production is not a target for this capability**, and requests to run against production are declined.

Human authorization gate

Actions with meaningful blast radius require operator approval before execution, and the approval is recorded against the action.

Rate and blast-radius limits

Concurrency and request budgets are bounded per target so that discovery cannot become a denial-of-service event.

Complete action audit log

Every request the agent makes is attributable to a step, a chain and an operator decision.

Data minimisation

Where a chain exposes data, the evidence captures only what is needed to demonstrate the outcome; bulk extraction is out of scope by design.

Service-operated, not distributed

The capability runs under Kolega's operational control with trained operators, both because of the risk profile and because aggressive autonomous exploitation requires judgement that a shipped binary cannot supply.

Why this is a service and not a product

Clients who want continuous, self-directed visibility across their own code have a different need, and it takes a different instrument. The two are complementary and easy to confuse.

TABLE 7 Two capabilities, two purposes

	KOLEGA DEVSEC MAX	RED TEAM AGENT
Purpose	Self-curated scanning, remediation workflow and compliance evidence (SOC 2, ISO 27001)	Adversarial validation of chains and proof of closure
Mode	Client-operated platform, integrated with the delivery pipeline	Kolega-operated continuous penetration testing service
Activity	Read-only analysis of source and configuration	Active exploitation in an authorized isolated environment
Output	Findings, triage, remediation tracking, control evidence	Confirmed chains, POCs, engineering and executive Rings of Risk reports
Cadence	Per merge and continuous	Scheduled campaigns plus post-fix verification runs
Availability	Licensed for client self-use	Service only: not distributed, not self-service

SECTION 10

Limitations

Execution and closure not yet benchmarked

The validation protocol in §8.2 is the remedy; until it reports, claims 3–5 in Table 1 remain open.

Coverage is bounded by what is modelled	Routes through components outside the mapped scope (third-party SaaS internals, network-layer controls, physical or human factors) are not covered.
Victim-dependent routes need conditions	Chains requiring a privileged user to open attacker content are confirmed only where that condition can be established; otherwise they remain conditional candidates, however plausible.
Agent non-determinism	As with any LLM-driven system, two campaigns may explore differently. Results are a lower bound on what is reachable, not an upper bound.
Scale remains open	The 1.77% recall measured on 263k–816k-line systems is a general-purpose scanning result, not a Red Team Agent result; whether agentic execution degrades comparably at enterprise scale is an open question that only engagement evidence can answer.
No substitute for human judgement	The service is designed to extend a penetration testing function, not to replace the tester's judgement about business impact.

■ APPENDIX

Definitions

Access Ring (AR)

Privilege tier, numbered from the core.

AR5 Outside	AR4 Standard user	AR3 Delegated authority	AR2 Control plane	AR1 Crown jewels
----------------	----------------------	----------------------------	----------------------	---------------------

Consequence band

B1 Critical, B2 Major, B3 Moderate, B4 Limited: the harm of an outcome, independent of the attacker's position.

Reach Ring (RR)

RR1 Critical reach, RR2 Major reach, RR3 Chain-dependent, RR4 Bounded reach: a finding's role in reaching an outcome.

Candidate route	A route supported by analysis but not executed. Not evidence of exploitability.
Confirmed chain	A route meeting all five confirmation criteria (C1–C5) with a reproducible artefact.
Closure harness	The retained, re-runnable artefact and pass condition used to verify post-remediation closure.
Known-route closure set	The smallest set of fixes that interrupts every known route to a given outcome.

SOURCES

Rings of Risk: Consequence and Reachability in Application-Code Vulnerability Chains, working draft, 22 October 2026, github.com/kolega-ai/rings-of-risk.

RealVuln Benchmark Comparative Report, Python & TypeScript/JavaScript stratum, v3.1.0, data generated 11 September 2026, realvuln.com.

Disclaimer. This document is provided for information and is not legal, regulatory or investment advice. It describes a security-testing capability that performs active exploitation; it must be used only against systems the operator owns or is expressly authorized in writing to test, within an agreed scope and in an isolated environment. Kolega Labs provides this capability as an operated service and develops the scanning products referenced in the companion documents; readers should weigh the corresponding commercial interest accordingly. Statements about future validation are intentions, not commitments, and results may differ.

© 2026 Kolega Labs. Technical whitepaper, Issue 1, 23 October 2026.