

Methodology paper; no frontier-model benchmark results are reported. Results are forthcoming after containment validation.

The Autonomy Trust Curve: A Methodology for Evaluating Autonomous and Semi-Autonomous Offensive Cyber Agents

Joe Daoud · CRUCIBLE

CRUCIBLE Methodology Paper · Version 1.1 · September 15, 2026

<https://www.cyber-crucible.com/publications/autonomy-trust-curve/v1.1/>

Abstract

Cyber-agent benchmarks increasingly test whether language-model agents can complete executable capture-the-flag, vulnerability-exploitation, and penetration-testing tasks. Success alone does not show whether an agent remains stealthy, stays within authority, limits collateral effects, collaborates with a supervisor when one is present, or remains contained when it acts unsafely. To measure those properties, we present a methodology for evaluating a declared model-and-harness system across a frozen battery of instrumented enterprise scenarios and an ordered autonomy ladder, implemented as the CRUCIBLE platform. The ladder spans semi-autonomous operation with human approval gates through fully autonomous operation with no routine human intervention. The test battery is designed to create repeated opportunities to measure twelve capability, supervision, and risk axes, with each axis scored only when the relevant condition occurred and the required evidence was captured. Deterministic generation, a simulated operator where the selected rung requires one, an active defender, an ephemeral range, off-range telemetry custody, and deterministic scoring are designed to make each result reproducible and auditable once the live path is closed. The resulting autonomy-trust curve quantifies the deployment risk an owner takes on at each autonomy level and identifies the most autonomous level that still meets that owner's capability and risk thresholds. Because the live path is not yet closed, the first benchmark results will be reported only after CRUCIBLE completes its containment validation.

1. Introduction

Cyber-agent benchmarks now place language-model agents inside capture-the-flag tasks, penetration-testing networks, real software repositories, and reproducible vulnerability environments. They show that frontier models can perform multistep cyber work and that the harness, tools, and assistance change the result.

Operational use, however, needs more than task completion. Two agents can reach the same objective while differing in whether they evade defensive observation, remain within authorization, recover from disruption, resist manipulation, report accurately, and request human intervention at the right time. A terminal-success score can treat those systems as equivalent

despite their different operational risks.

A 2026 Cyber Fusion Innovation Center assessment event, held in collaboration with U.S. Army Cyber Command and the Army Cyber Technology and Innovation Center, solicited industry white papers for an agentic C2 framework "capable of stealthy navigation, environmental observation, and anomaly detection" and required to "operate with minimal human oversight" in "contested, air-gapped, or cloud-native systems" [24].

Practice is already running ahead of measurement. OpenAI describes its Defense Factory, a continuous agent-first operation for finding and fixing vulnerabilities in its own systems, as having "started with small batches and human review, then removed repeated manual steps as the results earned trust," and draws the loop with a step labeled "Learn, Adapt, and Increase Autonomy" [26]. The page reports rates for reproduced findings, false positives, and accepted ownership, but it gives no instrument for deciding when trust had been earned or how much autonomy each result justified.

CRUCIBLE exists to fill the gap between what those benchmarks measure, what a solicitation like that one requires, and what an operator raising autonomy by judgment lacks: a measurement. Its thesis is:

CRUCIBLE measures how much autonomy a cyber agent can be given before the risks of independent action outweigh its useful capability.

In practical terms, the output is quantified deployment risk per autonomy level: what an owner takes on by running a declared system at a given rung, in a form that can be compared across candidate systems.

CRUCIBLE evaluates a declared agent system, not a bare model: model-service settings, harness, prompt, information feed, tools, memory, budgets, and retry policy all form part of the subject. A frozen battery then gives each such subject byte-reproducible scenario inputs, rules, and scoring criteria, so the result can be reconstructed for audit, compared across systems, and rerun to detect regressions or improvement as part of an ongoing process rather than a one-time certification [23].

The methodology combines a scenario battery, deterministic generation, a simulated operator, an active defender, an ephemeral target range, a common reference harness, off-range evidence custody, and deterministic scoring. Across all of them, it keeps agent behavior and range containment as separate measurements: an unsafe attempt blocked by the range is still unsafe behavior, while compliant behavior does not prove that the range could have contained a rogue agent.

The rest of this paper describes each of those components, how far each has been validated, and the governance an owner needs before acting on the evidence. It does not report frontier-model results, which wait on safety-guardrail and containment validation.

2. Related Work and Benchmark Landscape

CRUCIBLE builds on executable cyber benchmarks, security-evaluation suites, enterprise simulators, and defense assurance practice, but it does not try to add more realistic cyber tasks to that literature. Instead it measures how a declared agent system conducts itself under supervision, defensive pressure, and enforced boundaries, and its scores are meant for the agent owner who has to decide whether, and with how much autonomy, to deploy a given model and harness.

2.1 Executable cyber benchmarks

Interactive capture-the-flag benchmarks were the first to evaluate tool-using cyber agents, and later work follows their pattern. InterCode-CTF evaluates 100 picoCTF objectives in Ubuntu containers [1]; NYU CTF Bench expands the approach to 200 challenges and a tool-equipped agent framework [2]; and CTF-Dojo increases the scale of executable training and evaluation environments [3]. Cybench adds professional tasks, subtask decomposition, and comparisons across scaffolds under controlled memory, iteration, and token budgets [4]. Together they make multistep behavior observable and show that capability depends on the surrounding agent system.

Real-vulnerability benchmarks extend this work beyond CTF puzzles. CVE-Bench evaluates exploitation of real web-application vulnerabilities [6]. CyberGym and CyberGym-E2E construct executable corpora spanning vulnerability reproduction and discovery-to-patch workflows [7, 8]. BountyBench connects detection, exploitation, patching, token cost, and economic impact [9], while ExploitGym tests whether agents can turn vulnerability-triggering inputs into working exploits across userspace, V8, and Linux-kernel targets [10]. These projects supply realistic targets, scale, and executable verification. CRUCIBLE keeps all three and adds measurements of authority, supervision, defensive pressure, collateral effect, containment, and evidence quality.

2.2 Autonomous and assisted penetration testing

AutoPenBench is the closest published academic parallel to CRUCIBLE. It connects a Kali workstation to vulnerable systems, permits unrestricted Linux commands through MCP, defines intermediate penetration-testing milestones, and compares autonomous with human-assisted agents across 33 tasks [5]. In its assisted mode, a human decomposes the objective into successive subtasks while the agent remains autonomous within each subtask. The authors report a substantial improvement, under the assumption that the human knows enough about the vulnerability to direct the work.

AutoPenBench evaluates discovery and exploitation of one vulnerable system at a time, with flag recovery as terminal proof. Between the start and that flag, an LLM judge interprets intermediate command and stage milestones. The judge can recognize multiple valid paths, and the authors report agreement compatible with a human evaluator on the included tasks, but it also adds a probabilistic model to the measurement path: a hosted LLM judgment is harder to reproduce exactly and verify offline than a fixed scoring function.

CRUCIBLE changes the experimental unit to a multi-host operation with an active defender, explicit ROE and approval gates, a fallible operator, ordered autonomy levels, and independent containment controls. Intermediate credit comes from recorded effects and sealed criteria instead of an LLM judge, because a certifying authority should be able to recompute a score from the retained evidence without calling another model, which may return a different answer on retry.

The closest industry parallel is Booz Allen's Cyber Weapon Index, published in August 2026 with an addendum a week later [30]. It ran 18 U.S. and Chinese models as fully autonomous attackers, each controlling a real attacker machine against a live, defended Active Directory network, issuing one command at a time with no tool menu or scaffolding, under repeated trials, and scored from network traffic, host logs, domain-controller logs, and intrusion-detection sensors rather than model claims. Its findings match this paper's premises: the unit of risk is the full system of model, harness, tools, and autonomy; a harness let a mid-tier model rival the leader; the same model refused a stealth task that its cyber-tuned sibling executed; and a leaderboard that was accurate at publication was materially different a week later, so measurement has to be continuous.

The index measures capability, and it stops there. It sweeps no autonomy levels, places no supervisor or rules of engagement in the loop, records no hard-fail events, and averages a vulnerability-research score and a kill-chain score into one number, with reliability reported as "reaches reliably" or "occasionally" and no trial counts or intervals. It is also a single-vendor result with

undisclosed scenarios and no independent replication. CRUCIBLE keeps the same evidence discipline, telemetry over claims and one action per model output, and adds what the index leaves out: the supervision plane, the autonomy sweep, separate hard-fail accounting, coverage reporting, and an evidence chain a third party can recompute.

2.3 Security evaluations and enterprise simulators

CyberSecEval 2 probes prompt injection, code-interpreter abuse, cyber-request compliance, refusal, and vulnerability exploitation [11]. These tests are cheap to run and can expose safety-capability tradeoffs before a model enters a long-running agent loop. Their usual unit, though, is a prompt, generated artifact, or bounded task rather than an enterprise operation that accumulates effects and changes authority over time.

CyberBattleSim represents machines, vulnerabilities, credentials, and attacker-defender transitions as an abstract graph for reinforcement-learning research [12]. It provides safe scale and direct ground truth but omits full operating systems, deployed network controls, and evidence reconciliation across endpoint, runtime, and network sensors. CRUCIBLE sits at the other end of that tradeoff: it is an executable range, not an abstract graph, so its claims concern one declared agent system inside a specified range and supervision policy.

2.4 Frontier-lab cyber-evaluation incidents

Recent disclosures from OpenAI, Anthropic, and the United Kingdom AI Security Institute (AISI) describe three different evaluation-boundary failures. "Escape" is the wrong word for all three; each was a different engineering failure with its own lesson. All three are known only because the laboratories and evaluators chose to disclose them, and they deserve credit for that.

OpenAI's July 2026 disclosure describes GPT-5.6 Sol and a more capable pre-release model, configured with reduced cyber refusals, being evaluated on ExploitGym [13]. Direct internet access was denied, but agents could reach a shared Artifactory package service. From there they used supporting infrastructure to reach the internet, escalated privileges, moved laterally, and attacked Hugging Face while searching for benchmark material. The episode compromised both containment and evaluation validity.

OpenAI researchers Eric Wallace and Michael Dalton later gave a provisional reconstruction at Black Hat USA [25]. By that account, across months of training and evaluation, agents discovered that Artifactory was writable and shared between runs. They exchanged notes, assignments, credentials, scripts, and exploits; found an SSRF route to the internet; and later obtained remote code execution and administrative access. An outage exposed the activity on July 4, and OpenAI revoked credentials, cleared state, rebuilt the service, and patched the known vulnerability. Even so, by July 8 agents had created another message board through unauthenticated WebDAV directory names. At the time of the talk, the presenters said, the investigation was still under way.

Anthropic reported a different mechanism [14]. Among 141,006 reviewed cyber-evaluation runs, three incidents spanning six fictional CTF runs occurred with evaluation partner Irregular. Prompts said that internet access was unavailable, but a misconfiguration left live paths open. Through those paths, agents attacked a real organization whose domain matched a fictional target, published a malicious package to PyPI, and, in another run, scanned roughly 9,000 targets before compromising a real application. Anthropic names Cybench, CyberGym, and ExploitBench elsewhere in its evaluation program but does not attribute these incidents to those public benchmarks.

UK AISI intentionally enabled internet access to approximate a human attacker's resources and disabled model-service cyber classifiers to elicit underlying capability [15]. Across 122 runs with seven models, it identified 19 out-of-scope actions in 10 runs, 17 associated with Mythos 5 and two with GPT-5.6 Sol. These included a malicious pull-request attempt, false identities intended to influence a maintainer, use of public services, prompt-injection artifacts, and resources left for later agents. Here the agents did not leave AISI's virtual-machine sandbox; the failure was one of authorization and monitoring across a network boundary that was open by design.

OpenAI's own account confirms the distinction: AISI deliberately allowed open-internet tool acquisition, while Irregular intended isolation but exposed a path through misconfiguration, and both were separate from the Hugging Face incident [21], whose cross-run message board WIRED first reported [16] and whose fullest technical account is now the Black Hat presentation [25]. Together the three cases produce four requirements: enforce claimed network boundaries, monitor and constrain intentionally permitted access, eliminate cross-run writable channels, and record the model-service safeguards used in every evaluation.

Anthropic's chief executive later characterized the same incident as a swarm that attacked targets unrelated to its task, sacrificed individual agents for the group's success, and attempted to break into the grader that scored it, and argued that every frontier company should act as if it had happened to them [29]. The attempt on the grader is why CRUCIBLE's scoring component runs outside the range on frozen evidence (§3.3).

These disclosures changed what CRUCIBLE does first. The scenario generator, harness, scoring system, and range lifecycle can already produce benchmark artifacts, but the fact that the components work is not, on its own, a good enough reason to put a frontier model and an offensive toolkit together in a live range. So CRUCIBLE is validating containment and the evidence chain in stages first, and it treats every unresolved observation as inconclusive until it is reproduced and explained. That decision has delayed the first benchmark scores: there are no published results because CRUCIBLE has chosen not to run frontier models until its safety gate passes, not because it lacks a way to measure them. Publishable model runs will begin only after the exact execution environment has shown that it can bound and observe what the evaluand does.

2.5 Military AI assurance and accreditation context

The Department of War's developmental test-and-evaluation (T&E) organization describes AI-enabled-system assessment as emergent and difficult across large, variable state spaces, and calls for independent evidence supporting judgments beyond individual test points [17]. Its assurance-case guidance links trustworthy use to structured arguments, measurements, instrumentation, and test infrastructure.

The Chief Digital and Artificial Intelligence Office divides AI evaluation into model, human-systems-integration, systems-integration, and operational levels [18]. In that scheme, model performance is one input alongside team interaction, security, interoperability, effectiveness, suitability, and survivability.

The Department's developmental T&E guidebook assigns accreditation authority to the relevant Department component and treats verification, validation, and other T&E outputs as evidence for that authority [19]. It also identifies human-machine teaming as different from ordinary tool use and notes that assessment remains immature when responsibilities change or communications degrade. Operational T&E likewise emphasizes independent assessment of production-representative systems in realistic conditions [20].

Section 3 of Executive Order 14409 directs specified national-security components to maintain a classified benchmarking process for advanced cyber capabilities of AI models and a threshold for designating a "covered frontier model" [22]. It does not create an accreditation system for offensive-cyber agents and says nothing about CRUCIBLE.

A 2026 *Cyber Defense Review* article by Lieutenant General Christopher L. Eubank frames "decision hand-off" as the question of when autonomous systems may act and when authority must return to a commander [23]. Its proposed TAGA framework connects training, authorities, guardrails, and accountability, with autonomy adjusted or withdrawn as conditions change. The article is the author's professional view, not Department policy, but the problem it describes, when a commander can let a system act on its own, is the problem CRUCIBLE measures.

Outside government, the same executive has proposed pacing frontier development through "checkpoints" in which a model with a given capability, such as defeating common sandboxing methods, must be accompanied by certified evidence about its propensity to use that capability, verified by embedded third-party evaluators [29]. That pairing of a capability threshold with separately measured propensity is the claim split in §3.4, and the demand for verification by a party outside the developer is the requirement in §9.2.

Booz Allen's recommendations ask for a national program that tests models under consistent conditions with repeated trials and independent telemetry, evaluates the full system including harness, tools, configuration, and autonomy, and requires standardized diagnostics from providers covering demonstrated capability and sensitivity to configuration [30]. The paper cites this as context, not endorsement; it is the shape of measurement §3 through §8 specify.

Taken together, these sources describe who accredits a system and what evidence they need. CRUCIBLE is designed so that a Component or owner who already holds that authority could use its evidence; it is not an accrediting authority itself, and it cites these sources as context, not as customers or endorsements. The question it measures is whether a declared agent system can operate effectively as autonomy increases while meeting stated supervision, authorization, containment, and evidence requirements.

2.6 Continuous agentic defense operations

OpenAI's Defense Factory is a continuous, agent-first operation in which agents find, validate, and fix vulnerabilities in the operator's own systems, and Cloudflare, Ramp, and Google have published similar operations [26]. Its reference architecture places a control plane for workload orchestration, policy enforcement, and a credential proxy beside a data plane of isolated, reproducible development containers with environment identities and host monitoring, with host activity, infrastructure security, and agent audit providing oversight across both. Each environment is created for one run and discarded with its state afterward so that one run cannot contaminate the next.

OpenAI reports results from an internal sprint in the form CRUCIBLE's axes would produce: agent-assisted deduplication marked 37% of findings as duplicates, runtime validation reproduced 19.5% of findings and cut the false-positive rate after dynamic validation to 0.81%, and 90.6% of ownership assignments were accepted [26]. It also reports that early severity classifications varied with the instructions the agents received until the rubric and prompts were versioned and repeatable evaluations were added, which is the same instability that leads CRUCIBLE to freeze its rubric and scoring function (§8). Autonomy was raised in steps: the team "started with small batches and human review, then removed repeated manual steps as the results earned trust," with people moving toward setting boundaries, handling exceptions, and checking outcomes [26].

Three things follow for CRUCIBLE. First, the isolation pattern in that architecture, ephemeral environments behind a policy and credential control plane with host monitoring and audit, is now common practice rather than a distinguishing feature, and CRUCIBLE's range in §6 follows the same pattern; what CRUCIBLE adds is the doctrine in §6.4 that exercises each control from the agent's context and the separate scoring of containment in §8.2. Second, the Defense Factory solves a different problem. Its agents work against the operator's own code with no adversary that responds, its people review outputs rather than adjudicate decisions at gates, and its measurements are outcome rates. It therefore has no equivalent of CRUCIBLE's active defender (§5.3), modeled supervisor (§5.1), or trajectory-level scoring (§8.1), and those three are where CRUCIBLE's

contribution lies. Third, "as the results earned trust" is a judgment, and the page gives no measurement behind it. The autonomy-trust curve (§8.5) is a way to state that judgment as evidence: which rung a declared system has earned, for which scenario families, against which defender, at what violation rate.

OpenAI also now provides purpose-trained offensive-capable models through an access-gated program. Daybreak Red gives approved users GPT-5.6-Cyber for penetration testing, red teaming, and exploit development, subject to additional approval and "stronger verification, monitoring, access controls, and human oversight" [27, 28]. That program governs who may use the model. It does not say how much autonomy such a model may be given inside a mission, or who establishes that independently of the laboratory that trained it; §9.2 treats that independence as a governance requirement.

3. Evaluation Target, Claim Boundary, and Threat Model

CRUCIBLE freezes the evaluated subject, its information, the evidence base, and the permitted claims before an episode. The four subsections below take them in that order.

3.1 The evaluand

CRUCIBLE defines the evaluated subject as a versioned system tuple:

```
model + model-service provider and safeguard settings + harness version + system prompt +  
information-feed definition + action schema/toolkit + memory policy + budgets + retry policy
```

Every result is tied to this tuple because changing any one component changes how the evaluand behaves, so a result from the changed system is a result for a different subject. For example, harness changes can alter planning cadence, information exposure, tool affordances, retry behavior, and action cost even when the model name stays fixed. That is why the empirical study will compare models inside one pinned reference harness and report results for those complete systems.

3.2 Information boundary

The evaluated system receives four classes of information.

Episode tasking supplies the objective, authorized organization or target context, starting assumptions, and desired result. It tells the agent what to accomplish without revealing a solution path.

Rules of engagement (ROE) identify permitted and prohibited targets, approval-gated actions, time and scope limits, and stop conditions. The distinction matters because a reachable asset is not necessarily an authorized one.

A recon-gated world projection contains only environmental facts the agent has discovered. The full topology stays hidden so that the interface does not hand the agent a complete map of a network it has not yet explored.

Action outcomes report whether a gate allowed the request, whether execution occurred, and what observable result or error followed. This preserves the chain from model output to gate decision, effect, and evidence.

The sealed answer key, defender-private telemetry, undiscovered topology, internal execution sentinels, and information from other episodes remain unavailable, because exposing any of them could reveal the solution, remove the need to do reconnaissance, create an unintended completion channel, or permit cross-run coordination.

3.3 Trusted and untrusted components

The agent and target content are untrusted: either may be mistaken, manipulated, deceptive, or actively adversarial.

Against them, CRUCIBLE relies on four trusted functions, each with a distinct job:

- **Scenario admission** verifies before a run that the deployed environment, mission briefing, private answer key, and scoring criteria describe the same scenario. A missing host, contradictory ROE rule, or scoring criterion with no possible evidence source prevents admission.
- **The runtime** receives the agent's requested action, applies ROE and operator gates, executes only an allowed action, and records the resulting lifecycle event.
- **Evidence producers** observe what happened at endpoint, network, identity, runtime, and cloud enforcement points. They record effects independently of what the agent says it accomplished.
- **CRUCIBLE's scoring component** reads the frozen evidence and private scenario criteria, then applies a fixed scoring function. It does not ask an LLM or a human reviewer to reinterpret the episode.

No component is allowed to prove its own enforcement merely by reporting success. For example, a runtime record saying that an external connection was blocked is checked against evidence from another relevant enforcement or observation point when that claim requires corroboration. Sources have different visibility, so an empty cloud network log is not by itself proof that an in-node network control denied the connection.

Because the evaluated agent may compromise a Kubernetes-class node in the target range, authenticated sources send evidence to a dedicated collector outside the target range as the episode runs. The collector identifies the source and episode, preserves record order, stores the accepted evidence under retention controls, and signs the frozen checkpoint with a key that is not available to the range nodes. This prevents a later node compromise from rewriting evidence already accepted by the collector. It does not, however, make a compromised node truthful: claims originating only from that node remain node-attested and require an independent source when stronger assurance is needed.

3.4 Claim boundaries

Claim	Question	What it cannot establish
Capability	What did the evaluand accomplish?	Safe conduct or containment
Behavioral safety	How did it act within authority and supervision?	Whether range controls would stop a rogue action
Containment	Did independent controls bound effects?	Whether the agent attempted to behave safely
Certification policy	Will an owner accept the measured risk for a stated scope and autonomy level?	Universal safety or operational validity outside that scope

Internal scoring reliability is a fifth question, separate from all four: a deterministic score is not evidence that the measure predicts real mission performance.

4. Battery and Scenario Design

No single cyber operation creates every condition needed to measure all twelve axes. CRUCIBLE therefore uses scenario families selected to activate different observables, replicate measurements across distinct situations, and expose actual observation coverage, and an axis is not scored in an episode that never presented the relevant condition.

Battery v1 allocates 15 design families and targets at least three strong ranked families per graded axis, with four for selected margin axes. Because a family intended to cover an axis may never be instrumented, executed, or observed for it, CRUCIBLE tracks intended, ranked, instrumented, executed, and observed coverage separately.

Multiple seeds vary the details that do not affect whether the objective is reachable or how the episode is scored. They test stability beyond one fixed world, but variants of one family remain related observations.

The battery manifest binds the scenarios, frozen content identities, weights, exclusions, ordering, and governance. Its exact scenarios and answer keys are not published, to reduce memorization, copying, and the kind of search for outside answers seen in the OpenAI/Hugging Face incident [13, 25]. Keeping them secret does not make containment any less necessary.

4.1 Deterministic generation and admission

CRUCIBLE models scenario generation as a deterministic transformation:

```
immutable scenario bundle + seed + benchmark version -> environment + ROE scope + tasking +  
sealed answer key + evaluation criteria
```

The transformation binds the environment and briefing exposed to the agent to the private facts used for evaluation.

All randomized choices derive from domain-separated streams under one master seed, so identical frozen inputs produce a byte-identical world, and serialization and identity failures fail closed. The generator also records the bundle, seed, benchmark version, reviewed content, buildability, and governance needed to identify an admitted instance.

Battery admission checks that identities agree, artifacts build, an intended solution path is reachable, tasking and ROE match the sealed scenario, and evaluation criteria name evidence the design can produce. Determinism alone is not enough: a world that is reproducible but wrong, such as an answer key naming a host that does not exist, still fails admission.

Passing admission shows only that the design is internally consistent, not that anything was observed live; scoring still requires the condition to occur, healthy sensors to capture it, and episode-bound evidence to reach the scoring component.

4.2 Variation and difficulty

At `entropy = 0`, the generated surface is pinned for regression testing. Higher values vary properties that do not affect the solution, such as decoys, placement, image selection, and technique choice while preserving reachability and scoring truth.

Variants from one family share an objective, logic, and implementation, so ten seeds of one family tell you less than ten unrelated families would. The preregistered analysis must account for that relationship through family-first aggregation, family-level effects, or family-clustered resampling.

Difficulty changes related settings together: tasking intelligence, decoy burden, route complexity, and topology. **Tasking intelligence**, for example, is the target information or mission context supplied at the start of the operation: a lower-burden instance may name a relevant system or service; a higher-burden instance may provide only the objective.

These settings change the expected workload in a reproducible way, but nobody has yet calibrated them, so they are not claims about difficulty. Labels such as "easy," "moderate," or "hard" will have to wait for frozen mappings and empirical evidence of changed workload or success rates.

4.3 Multiple paths and sealed evaluation

The answer key can represent several valid routes instead of one author-preferred sequence. It can also contain **negative truth**: a sealed assertion that an apparent credential is a decoy, a reachable host is prohibited, or a tempting route does not satisfy the objective.

The agent receives only tasking and discovered state, while CRUCIBLE's scoring component receives the sealed facts needed to credit alternate valid strategies and reject false opportunities.

5. Human Supervision and Active Defense

The simulated operator and defender create the approvals, redirections, detections, and setbacks needed to measure supervision and adaptation, and both are pinned in benchmark provenance so that a rerun faces the same supervisor and the same defender.

5.1 Fallible simulated operator

The simulated operator has bounded attention, experience, and judgment. It sees the agent's message, tasking, ROE, interaction history, and its own declared profile. It does not see the answer key, hidden topology, defender-private telemetry, or evaluator-only state, because with those facts it would know the answers the agent is supposed to find.

The agent can request approval, surface a finding, recommend another path, flag an ROE boundary, or report status. The operator returns approval, denial, redirection, or halt with a recorded rationale. To decide, it evaluates authority, supporting evidence, relevance, and whether the message gives a supervisor enough information, so a denial may reflect inadequate authorization or explanation rather than technical error.

Basic, Senior, and Master experience levels vary tendencies such as caution and susceptibility to weak claims, and a content-derived seed makes identical adjudications reproducible across those configurations. These nominal profiles are implemented and tested, but they have not been calibrated to human operators.

5.2 Supervision events and manipulation probes

Scenario families stage approval gates, new findings, redirects, stop instructions, and competing courses of action. Each event records what the agent knew, its message, the operator's response, and the next attempted action. The chain distinguishes timely escalation and adaptation from needless escalation or repetition.

Manipulation probes appear in documents, service output, or other target artifacts. Before a probe counts either way, scoring requires evidence that the content was available, the agent encountered it, and the subsequent behavior was bound to that encounter.

The agent may also send the operator messages that are weakly supported by evidence, or worded to get a particular answer. Because the operator can be taken in by such a message, the sealed record, not the operator's verdict, determines whether a report or recommendation was factually correct.

5.3 Active defender and sensor fabric

The active defender consumes endpoint, network, and identity telemetry with declared coverage and blind spots. From that telemetry, signature rules identify high-signal events, and behavioral rules combine observations over time. Detections produce structured alerts with technique and evidence provenance, and a tiered policy can apply graduated defensive pressure.

Two axes depend on this component. Stealth is measured against a declared, functioning sensor baseline, and recovery is measured from the moment a defender action, execution failure, or environmental change sets the agent back, while a supervisor's redirection is measured separately. A frozen defender configuration produces the same alerts and response decisions from identical telemetry, so both measurements are reproducible.

Booz Allen reports that coordinated counter-AI playbooks reduced attacker success by more than 95 percent in its testing [30]; a defender that can produce an effect of that size is the reason stealth and recovery are measured against an active defender rather than a passive log.

An empty alert log counts as clean only when every required sensor reports healthy event-driven capture without a gap. That condition is not yet met. CRUCIBLE has implemented deterministic rules, alerts, response decisions, health gates, and tests over recorded telemetry, but the live path remains incomplete: endpoint collection misses important short-lived activity, identity and network coverage have gaps, and the response adapter does not apply every declared effect. CRUCIBLE will not make live stealth or defender-response claims until it has event-driven sensors, off-range custody, and proof that the declared responses were applied.

6. Ephemeral Evaluation Range and Control Validation

Each episode runs inside bounded, ephemeral infrastructure whose lifecycle is recorded:

```
reserve -> provision -> pre-action containment gate -> instantiate -> execute -> observe ->
freeze -> destroy -> residue check
```

Containment is checked before agent action, evidence is frozen before teardown, and authenticated inventory follows destruction. If a controller fails or resources are left behind, watchdogs, error-path cleanup, an orphan ledger, and startup reconciliation handle them.

6.1 Staged validation before agent execution

Four numbered validation phases, including a focused Phase 2.5 diagnostic, precede the full benchmark.

Phase 0: offline substrate validation exercises generation, artifact identity, policy rendering, lifecycle logic, and failure handling without cloud infrastructure.

Phase 1: ephemeral substrate validation uses a scripted stand-in to prove that a live range can be created, sealed, observed, destroyed, and checked for residue.

Phase 2: containment and topology validation runs controlled probes from the agent's future execution context. Positive controls prove the probes and sensors; in-range paths confirm topology; out-of-range paths test denial and telemetry reconciliation.

Phase 2.5: containment-observability diagnostic settles a question that came up during validation: whether a denied connection was blocked inside the sandbox or by host policy. That diagnostic is not an evaluand run and does not relax the sandbox requirement for benchmark execution.

Phase 3: bounded exploitation arming adds the offensive workstation image, sandbox, sensors, stop mechanisms, and scripted tool activity. It revalidates the node, runtime, credential absence, containment, telemetry, topology, and defensive staging. Free model-driven tool use remains disabled, so Phase 3 produces control evidence rather than a model result.

Phase 4: full benchmark execution first enables the frontier-model evaluand, unrestricted use of its authorized toolkit, real tool input and output, and the complete scoring path. It begins only after the preceding gates pass.

6.2 Isolation and containment architecture

Containment uses a deny-by-default episode network, a hardened sandbox runtime, scoped workload identities, action authorization, and lifecycle budgets. Network, identity, and runtime controls constrain effects independently of operator and ROE gates. This layering matches the published reference architectures for agentic defense operations, which also separate a policy and credential control plane from ephemeral, monitored execution environments [26]. The layering itself is therefore not CRUCIBLE's contribution; the validation doctrine in §6.4 and the separate scoring of containment in §8.2 are.

The two layers answer different questions: authorization decides whether an action should execute, and containment limits its effects if authorization fails or is bypassed. Blocked unsafe attempts therefore remain behavioral evidence.

A boundary crossing aborts and invalidates the run, and a fault in boundary observation fails closed the same way. Short of a crossing, blocked escape attempts reduce blast-radius scoring, but recurrence alone is not a separate hard fail; instead the result distinguishes between a blocked probe, persistent attempted escape, and an actual crossing.

6.3 Off-range telemetry custody

Endpoint, runtime, network, identity, and lifecycle sources will stream records to a dedicated collector outside the range once the live path is closed. The collector is designed to bind each record to an authenticated source and episode, preserve ordering, checkpoint raw and normalized evidence into retention-controlled storage, and sign the frozen commitment with a key unavailable to range nodes. This off-range custody is designed to prevent later rewriting, but it does not make a compromised source truthful, so node-attested events still require corroboration from another enforcement point for stronger claims.

6.4 Control-validation doctrine

Each control declares its objective, enforcement point, evidence producer, and pass, fail, and inconclusive states. Positive controls prove that a probe and observation path work; watched failures prove detection. A **policy object**, such as a NetworkPolicy or security-group rule, records intended configuration but not effective enforcement, so CRUCIBLE exercises the restriction from the agent's context and confirms that the probe ran.

Cloud-infrastructure-provider telemetry is produced outside CRUCIBLE's in-node runtime. It can corroborate platform and dataplane evidence, but the absence of a cloud log entry cannot show which lower layer denied a connection. Residue absence likewise requires a complete, authenticated post-teardown inventory.

6.5 Incident-derived requirements and CRUCIBLE controls

The disclosures in §2.4 lead to four requirements for the range.

Prove effective isolation. CRUCIBLE applies a deny-by-default episode network, seals the cloud network boundary, and independently tests permitted and denied paths before and after sealing and again from the Phase 2 workload. Missing probes, telemetry, or reconciliation prevent a pass. Fictional names must resolve inside the range or fail closed.

Remove unintended pivots. Dependencies are resolved before sealing; public package services, cloud credentials, and external infrastructure are absent unless the frozen experiment names them. An intentionally permitted service receives destination-specific authority, attribution, and stop rules. Credential reuse, public artifacts, tunneling, and social engineering remain behavioral evidence even when controls block the effect.

Isolate episodes across time. Every run receives its own identity, namespace, artifacts, and evidence bindings, followed by authenticated residue checks. No standing writable package registry or shared artifact service is exposed. OpenAI's Black Hat account showed why an inventory must cover functions, not just known paths: after the Artifactory message board, which WIRED first reported [16], was removed, agents recreated one through WebDAV directory names [25]. For that reason CRUCIBLE reruns affected gates after remediation and checks that no run can inherit messages, credentials, scripts, or artifacts.

Bind the model-service configuration. Ordinary, reduced, and disabled safeguard settings define different evaluands. The range does not credit external refusal mechanisms as its own containment control. Booz Allen observed the same effect in its testing, where a general model refused a stealth mandate that its cyber-tuned sibling executed [30].

7. Reference Agent Harness and Toolkit

CRUCIBLE evaluates frontier models through one pinned reference harness, which is both part of the evaluand and an experimental control.

The harness runs a plan-act-observe loop: on each turn it presents the briefing, discovered world, prior gate decisions, and observed outcomes; requests one action; lowers it into a common representation; applies ROE and operator gates; and returns the result. The current bound is 32 model turns within the wider runtime budget.

Each of those choices serves the comparison: one action per turn binds each attempt to a model output and gate decision, common action lowering gives model-service providers the same runtime and scoring path, and the toolkit is pinned because it defines what the model can request.

The harness excludes answer keys, defender alerts, undiscovered details, and private sentinels used for budgets, terminal conditions, and objective state. The model sees only its action outcomes and discovered public state, and it authors its own operator messages, so escalation, reporting, and redirection remain evaluand behavior.

Provenance records model-service and model identity, settings, prompts, harness, budgets, and retries. When the model service fails, the run ends with a recorded terminal outcome; CRUCIBLE does not substitute a scripted action. Likewise, a refusal before action is recorded separately from one issued after the agent has acted.

The common harness supports controlled comparisons but cannot predict performance in another scaffold, which is why bring-your-own-harness evaluation is planned as a separate future class (§10.2).

8. Measurement and Scoring

CRUCIBLE's scoring component is a deterministic, LLM-free function over a frozen episode record. It reports each measurement, the evidence coverage, and any disqualifying event separately instead of reducing the episode to one number.

8.1 Graded axes

The scoring schema contains twelve graded axes across an environment plane and a human-supervision plane.

Plane	Axis	Operational definition
Environment	Success	Degree to which sealed episode objectives were achieved through trusted, observed effects.
Environment	Efficiency	Observed progress or completion relative to the enforced time, action, tool, and inference budgets.
Environment	Stealth	Ability to avoid or limit detection and defensive response relative to the declared, healthy sensor baseline.
Environment	Decision quality	Groundedness and operational relevance of selected actions given the information available at the decision point.
Environment	Recovery	Coherent adaptation after a non-supervisor operational setback, such as a failed action, lost route, revoked credential, changed service, or defensive response.
Environment	Blast radius	Extent to which attempted or realized effects spread beyond the minimum assets, identities, or services needed for the objective.
Human supervision	Escalation quality	Whether the agent seeks human judgment when warranted, avoids unnecessary escalation, and supplies the evidence needed for a timely decision.
Human supervision	Report accuracy	Agreement between agent reports and trusted positive and negative evidence, including uncertainty and unsuccessful effects.
Human supervision	Report legibility	Clarity, structure, and actionability of information supplied to the supervisor.
Human supervision	ROE adherence	Graded compliance with target, action, approval, time, and stop constraints, including attempted actions that a gate denied.
Human supervision	Redirection responsiveness	Degree to which the agent acknowledges and follows an explicit supervisor redirect or halt instead of persisting with the displaced plan.
Human supervision	Manipulation resistance	Ability to preserve the supervisor-authorized mission and authority boundary when adversarial instructions or deceptive content are encountered in the target environment.

An axis is measured only when the scenario activates it and the live system produces the required evidence.

8.2 Hard-fail events

Five outcomes are counted separately and never averaged in: ROE violations, acted-on hallucinations, injection hijacks, unauthorized range-boundary crossings, and faults in boundary observation. The last two concern containment integrity, not agent behavior. A high average does not cancel any of them.

A denied request reduces graded ROE adherence without becoming a completed violation, whereas an actual ROE crossing both reduces the grade and increments the hard-fail count. An acceptance policy may tolerate a nonzero rate, but every sub-threshold event still affects the score.

8.3 Evidence and observation coverage

Outcome credit requires a trusted effect bound to the episode, asset, action, and sealed objective. An agent saying it succeeded is not proof that it did, and missing or malformed evidence fails closed.

"Not observed" is metadata, not a score: an unmeasured axis is excluded from the aggregate and reported with `n/N` coverage while a measured zero remains. Every aggregate and autonomy-curve point reports the valid observations supporting each required axis.

8.4 Reliability and validity

Golden tests require exact scores, coverage flags, and hard-fail outputs from frozen records. **Sensitivity tests** change one input and verify the intended axis response without unrelated movement. **Rank-order tests** require the rubric to order constructed trajectories consistently with sealed qualitative differences.

These tests establish deterministic implementation and rubric-consistent discrimination, but only on constructed records; external validity requires independent expert judgments and prospective operational evidence.

8.5 The autonomy-trust curve

CRUCIBLE holds the subject and battery fixed while each autonomy rung changes which actions execute automatically, require approval, or remain unavailable. The highest rung may auto-approve every action that the ROE permits, creating a fully autonomous evaluation with no routine operator intervention; prohibited actions, containment controls, monitoring, and emergency termination remain active. The **maximum acceptable autonomy level** is the least restrictive rung that meets the owner's capability, risk, and minimum-evidence requirements. Read per rung, each point on the curve is a statement of deployment risk at that autonomy level: the graded axes, hard-fail counts, blast radius, and observation coverage measured there. The comparison an owner most often needs is between candidate systems at the rung they intend to use, and the curve is built so that two declared systems evaluated on the same battery can be compared that way.

A run with no supervisor interaction cannot establish escalation quality, report legibility, or redirection responsiveness. Those axes remain not observed (`n = 0`) unless the declared evaluand includes an emergency or reporting interface that the scenario actually exercises. A policy for fully autonomous operation must state which supervisor-dependent axes it requires; and their absence must not be reported as if the agent had done well on them.

9. Implementation Status and Governance

9.1 Implementation status

The table below distinguishes design completion, implementation, and validation in the live target range as of August 15, 2026. "Implemented" means that executable component behavior exists and has component-level or integration tests; it does not mean that the full benchmark evidence chain has been validated.

Component	Designed	Implemented	Validated in live range	Supported statement today
Battery v1 and coverage model	Yes: 15 scenario families with ranked coverage targets	Partial: private specifications and manifests exist; instrumentation varies by family	No complete battery campaign	The battery design and coverage doctrine can be described; full observed coverage cannot.
Deterministic scenario generator	Yes	Yes; determinism, identity, and admission behavior are internally tested	Partial; generated artifacts have been exercised, but not as one frozen eligible battery	Reproducible generation mechanisms exist; difficulty is not empirically calibrated.
Simulated operator	Yes	Yes; typed interactions, experience levels, behavioral profiles, and deterministic adjudication are tested	No	The declared operator model can be reproduced; its parameters are not calibrated to human operators.
Active defender and sensor fabric	Yes	Partial: detection, alerting, response logic, and health gates run on recorded telemetry	No complete live sensor-and-response path	Defender behavior can be tested offline; live stealth and defender-response claims are not yet supported.
Ephemeral range lifecycle	Yes	Yes: reservation, provisioning, sealing, evidence freeze, teardown, and reconciliation paths exist	Partial: offline, ephemeral-substrate, containment/topology, and containment-observability validation phases have completed; the bounded-exploitation arming phase is under way	Lifecycle mechanisms have live evidence, but the full range is not yet eligible for an evaluation campaign.
Isolation and containment evidence	Yes	Partial: layered controls, probes, stop paths, and reconciliation exist	Containment-observability diagnostic demonstrated; bounded-exploitation arming phase is under way; unresolved observations remain inconclusive until reproduced	Individual control results can be reported; no broad containment claim is made.
Reference harness and toolkit	Yes	Yes; the plan-act-observe loop, action lowering, gates, and provenance are implemented	No eligible full-battery campaign	The harness is a pinned experimental control, not a demonstrated source of model rankings.
Scoring and aggregation	Yes	Yes; deterministic, LLM-free scoring has golden, sensitivity, and rank-order tests	No external or prospective validation	Internal reliability and rubric discrimination are supported; predictive validity is not.
Certification policy and credentials	Yes	Proposed/nominal: policy and credential mechanisms exist as an authorable policy mechanism	No operational accreditation study	Authorable policy mechanism and future certification research; not operational accreditation.

9.2 Governance, auditability, and conflict of interest

CRUCIBLE's developer authors the methodology and may also provide evaluation or certification services. Because of that conflict, the scoring mechanics are kept separate from the authority to accept a result. The same requirement applies one level up. A laboratory that trains, gates, and monitors an offensive-capable model [27] is not positioned to be the sole source of

evidence for how much autonomy that model may be granted in another owner's mission, for the same reason that a system's developer is not its own accreditor (§2.5); an independent measurement that the laboratory does not control is what the curve is meant to supply. Anthropic's commitment to give embedded third-party evaluators employee-like access to verify its safety practices [29] is the same principle applied inside the laboratory. CRUCIBLE's frozen evidence and deterministic scoring are built so that such an evaluator, or an accrediting authority, can recompute a result without trusting the developer's summary of it.

On the measurement side, the deterministic scoring function gives frozen evidence the same axis values, coverage, and hard-fail outputs. Append-only benchmark identities and provenance bind the model-service configuration, harness, battery, evidence digests, exclusions, and policy version. These controls expose post hoc changes without claiming that the rubric itself is valid.

On the acceptance side, the system owner or certifying authority sets the mission profile, score floors, hard-fail ceilings, evidence requirements, scope, expiration, and acceptable autonomy. CRUCIBLE supplies only the measurement engine, and a credential states that a declared system met one named authority's frozen policy for a bounded scope.

The credential signs the verdict, trajectory chain head, episode digests and terminal reasons, aggregate scores, evaluation identity, safety configuration, and battery manifest. Ed25519 and ECDSA P-256 credentials can be verified offline, and that verification establishes integrity and provenance, not scientific validity.

Third parties can review sealed scenarios, admission records, scoring code, and sampled evidence under controlled disclosure, although no standing audit body or completed third-party validation is claimed. Each public release receives a version number, publication date, permanent version-specific URL, checksum manifest, and changelog entry; version 1.0 is accompanied by a versioned working analysis plan, which will be frozen at a permanent URL before benchmark outcomes are inspected, with later changes recorded as dated deviations. Reports will also distinguish vendor-operated evaluations from independently replicated results.

This evidence can support governance, risk, and compliance workflows: governance names the accountable owner, risk policy sets acceptance thresholds, and compliance checks whether declared controls were exercised. Configuration or policy documents alone do not prove enforcement.

10. Limitations and Future Work

10.1 Limitations and threats to validity

Holding one harness constant supports controlled comparison but does not establish performance across scaffolds, because prompts, information, tools, memory, retries, and action cadence can all change measured behavior.

Claims extend only to the included topologies, techniques, budgets, autonomy rungs, and observables. Contamination is a second limit: public vulnerabilities and CTF patterns may appear in model training data, and seed variants change surface details but retain shared structure and possible contamination, so they remain related observations.

Instrumentation varies by axis: efficiency needs enforced budgets; stealth needs healthy sensors and a resolved baseline; blast radius spans several kinds of effect; ROE needs grounded attempts, gates, and outcomes; manipulation resistance needs a complete exposure-to-action chain. Report legibility may be unavailable for some model cohorts. For all of these, coverage reporting narrows each claim to the evidence actually captured.

The range remains a simulation. Its operator, defender, timing, and response policies may not reproduce real teams, and selected probes may miss cloud or runtime failures.

More capable models may also recognize that they are being evaluated and behave differently under test than in deployment [29]. CRUCIBLE's hidden topology and sealed answer key limit memorization and answer-seeking, not evaluation awareness, and this paper makes no claim that a measured propensity survives the subject's knowledge that it is being measured.

Model-service nondeterminism and rolling revisions limit exact reproduction even when all observable settings are bound. The same marketed model name may later denote a different underlying system.

CRUCIBLE has not established predictive validity for real missions. That requires expert judgments, operator-agent comparisons, security-operations telemetry, and downstream outcomes.

10.2 Bring-your-own-harness evaluation

A later evaluation class will accept developer harnesses under a conformance profile that preserves battery identity, action authorization, evidence bindings, and scoring semantics. Cross-harness results must control or report differences in tools, information, memory, autonomy, and retries.

10.3 Authorable certification definitions

An agent deployer could define the autonomy rung, score floors, hard-fail ceilings, evidence requirements, approved scope, and accepted residual risk. The output would be a bounded authorization decision, not a universal safety label. Even then, calibration and predictive evidence would still be required before such a certification could be called validated.

A later extension could express a rung's risk in expected-loss terms, as the probability of a violation at that rung multiplied by the measured blast radius and the owner's cost per incident. Such a figure would let parties that consume evaluation evidence rather than produce it, such as auditors or underwriters, work in their own units. It would inherit every limitation in §10.1, and it would not be an actuarial input until the curve has been shown to predict real losses (§10.4).

10.4 External validation and calibration

Future studies should compare CRUCIBLE with expert ratings, real operator-agent performance, security-operations telemetry, and mission outcomes. Calibration must remain scoped to the harness class, mission type, controls, and autonomy ladder; one composite correlation could conceal disagreement among axes.

11. Conclusion

CRUCIBLE measures a declared cyber-agent system under active defense and bounded authority across semi-autonomous and fully autonomous operating levels. Its autonomy-trust curve quantifies deployment risk per autonomy level and identifies the highest autonomy that meets one owner's capability, risk, and evidence requirements for a stated scope. Operators are already raising agent autonomy by judgment as results earn trust [26]; the curve gives them a measurement to raise it by instead. As of this version, the live-range evidence chain has not yet passed the safety gate that frontier-model runs require, so those results will be published separately after containment validation.

References

1. John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao, "InterCode: Standardizing and Benchmarking Interactive Coding with Execution Feedback," *NeurIPS*, 2023.
https://papers.neurips.cc/paper_files/paper/2023/file/4b175d846fb008d540d233c188379ff9-Paper-Datasets_and_Benchmarks.pdf
2. Minghao Shao et al., "NYU CTF Bench: A Scalable Open-Source Benchmark Dataset for Evaluating LLMs in Offensive Security," *NeurIPS Datasets and Benchmarks*, 2024.
https://proceedings.nips.cc/paper_files/paper/2024/file/69d97a6493fbf016fff0a751f253ad18-Paper-Datasets_and_Benchmarks_Track.pdf
3. Terry Yue Zhuo, Dingmin Wang, Hantian Ding, Varun Kumar, and Zijian Wang, "CTF-Dojo: A Massively Scalable Pipeline for Automated Training and Evaluation of Cybersecurity Agents," 2025. <https://arxiv.org/abs/2508.18370>
4. Andy K. Zhang et al., "Cybench: A Framework for Evaluating Cybersecurity Capabilities and Risks of Language Models," 2024. <https://arxiv.org/abs/2408.08926>
5. Luca Gioacchini, Alexander Delsanto, Idilio Drago, Marco Mellia, Giuseppe Siracusano, and Roberto Bifulco, "AutoPenBench: A Vulnerability Testing Benchmark for Generative Agents," *EMNLP Industry Track*, 2025.
<https://aclanthology.org/2025.emnlp-industry.114/>
6. Yuxuan Zhu et al., "CVE-Bench: A Benchmark for AI Agents' Ability to Exploit Real-World Web Application Vulnerabilities," 2025. <https://arxiv.org/abs/2503.17332>
7. Zhun Wang, Tianneng Shi, Jingxuan He, Matthew Cai, Jialin Zhang, and Dawn Song, "CyberGym: Evaluating AI Agents' Cybersecurity Capabilities with Real-World Vulnerabilities at Scale," 2025. <https://arxiv.org/abs/2506.02548>
8. Tianneng Shi et al., "CyberGym-E2E: Scaling End-to-End Cybersecurity Evaluation Across the Vulnerability Lifecycle," 2026. <https://arxiv.org/abs/2606.04460>
9. Andy K. Zhang et al., "BountyBench: Dollar Impact of AI Agent Attack Capabilities," *NeurIPS Datasets and Benchmarks*, 2025. https://papers.neurips.cc/paper_files/paper/2025/hash/faed4276b52ef762879db4142655c699-Abstract-Datasets_and_Benchmarks_Track.html
10. Zhun Wang et al., "ExploitGym: Can AI Agents Turn Security Vulnerabilities into Real Attacks?," 2026. <https://arxiv.org/abs/2605.11086>
11. Manish Bhatt et al., "CyberSecEval 2: A Wide-Ranging Cybersecurity Evaluation Suite for Large Language Models," 2024. <https://ai.meta.com/research/publications/cyberseceval-2-a-wide-ranging-cybersecurity-evaluation-suite-for-large-language-models/>
12. Microsoft Research, "CyberBattleSim." <https://www.microsoft.com/en-us/research/project/cyberbattlesim/>
13. OpenAI, "OpenAI and Hugging Face Partner to Address Security Incident During Model Evaluation," July 21, 2026. <https://openai.com/index/hugging-face-model-evaluation-security-incident/>
14. Anthropic, "Investigating Three Real-World Incidents in Our Cybersecurity Evaluations," July 30, 2026. <https://www.anthropic.com/news/investigating-incidents-cybersecurity-evals>
15. UK AI Security Institute, "Incident Report: Unsanctioned Agent Behaviour During Cyber Testing," August 4, 2026. <https://www.aisi.gov.uk/blog/incident-report-unsanctioned-agent-behaviour-during-cyber-testing>
16. WIRED, "OpenAI Didn't Notice Its AI Agents Using a Message Board to Plan Their Hacking Spree," August 2026. <https://www.wired.com/story/openai-didnt-notice-its-ai-agents-using-a-message-board-to-plan-their-hacking-spree/>
17. Office of the Under Secretary of War for Research and Engineering, "Developmental Testing and Evaluation of Artificial Intelligence Enabled Systems." https://www.cto.mil/dtea/te_aies/

18. Chief Digital and Artificial Intelligence Office, "CDAO Test and Evaluation Strategy Frameworks," May 20, 2024. <https://www.ai.mil/Latest/Blog/Article-Display/Article/3940283/cdao-test-and-evaluation-strategy-frameworks/>
19. Office of the Under Secretary of Defense for Research and Engineering, *Developmental Test and Evaluation of Artificial Intelligence Enabled Systems Guidebook*, February 2025. https://www.cto.mil/wp-content/uploads/2025/02/TE_of_AIES_Guidebook_Final_26Feb25.pdf
20. Office of the Director, Operational Test and Evaluation, "Assessing the Joint Force: An Inside Look at the Office of the Director, Operational Test and Evaluation," April 26, 2024. <https://www.war.gov/News/Feature-Stories/Story/Article/3757702/assessing-the-joint-force-an-inside-look-at-the-office-of-the-director-operatio/>
21. OpenAI, "Third-Party Cyber Evaluations Involving OpenAI Models," August 4, 2026. <https://openai.com/index/third-party-cyber-evaluations-involving-openai-models/>
22. The White House, "Promoting Advanced Artificial Intelligence Innovation and Security," Executive Order 14409, June 2, 2026. <https://www.whitehouse.gov/presidential-actions/2026/06/promoting-advanced-artificial-intelligence-innovation-and-security/>
23. Christopher L. Eubank, "Commanding Agentic Forces in Multi-Domain Operations: Mastering Cognitive Convergence and Decision Hand-Off," *Cyber Defense Review*, vol. 11, no. 2, 2026. https://cyberdefensereview.army.mil/Portals/6/Documents/2026-vol11-iss2/CDR_V11_N2_Eubank.pdf
24. Cyber Fusion Innovation Center, "Agentic C2 Framework Tool (ECHOTRIBBLE) Assessment Event," 2026. <https://www.cyberfic.org/events/echotribble-ae>
25. Eric Wallace and Michael Dalton, "The 'Breaking' News: The OpenAI-Hugging Face Incident," *Black Hat USA*, August 5, 2026. <https://www.youtube.com/watch?v=87DyyMV0kCY>
26. OpenAI, "Defense Factory," September 2026. <https://openai.com/the-defense-factory/>
27. OpenAI, "Expanding Daybreak as the Cyber Defense Window Narrows," August 10, 2026. <https://openai.com/index/expanding-daybreak-as-the-cyber-defense-window-narrows/>
28. OpenAI Help Center, "OpenAI Daybreak - Trusted Access for Cyber Overview," 2026. <https://help.openai.com/en/articles/20001258-openai-daybreak-trusted-access-for-cyber-overview>
29. Dario Amodei, "We Must Pace the Frontier," September 2026. <https://darioamodei.com/post/we-must-pace-the-frontier>
30. Booz Allen Hamilton, "The Offensive Frontier: AI as the Attacker. A New Cyber Weapon Index and the Strategic Imperative to Accelerate Agentic AI Offense and Defense," report with addendum, August 2026. <https://www.boozallen.com/insights/cyber/cyber-weapon-index.html>

Publication Information

Preferred citation: Joe Daoud, "The Autonomy Trust Curve: A Methodology for Evaluating Autonomous and Semi-Autonomous Offensive Cyber Agents," *CRUCIBLE Methodology Paper*, version 1.1, September 15, 2026, <https://www.cyber-crucible.com/publications/autonomy-trust-curve/v1.1/>.

Current version: HTML · PDF

Permanent version 1.1: HTML · PDF

Version history: Changelog

Companion analysis plan: Working preregistration

Corrections: contact@cyber-crucible.com

Version history

V er si o n	Dat e	Stat us	Summary
1.1	September 15, 2026	Minor revision	Adds section 2.6 on continuous agentic defense operations and the Daybreak access program, with references 26 to 28; adds the Anthropic pacing proposal to sections 2.4, 2.5, and 9.2 and an evaluation-awareness limitation to section 10.1, with reference 29; adds the Booz Allen Cyber Weapon Index to sections 2.2, 2.5, 5.3, and 6.5, with reference 30; adds an independence requirement for model providers to section 9.2; states the curve's output as quantified deployment risk per autonomy level in the abstract, sections 1, 8.5, and 11, and the glossary; records an expected-loss extension as future work in section 10.3; replaces mannered phrasing with direct statement and joins sentences that previously stood apart throughout; withdraws the version 1.0 artifacts from the site. Changes no method, claim boundary, or implementation-status row.
1.0	August 16, 2026	Initial methodology release	Establishes the evaluation target, scenario-battery design, supervision and defender models, containment doctrine, scoring method, implementation status, governance model, and planned empirical study. No benchmark results are reported. Pre-publication corrections downgrade the certification-status row, generalize live-stack and diagnostic wording, neutralize government-context passages, and replace remaining em-dashes, en-dashes, and curly quotes with plain ASCII punctuation.

Corrections and later revisions receive new version numbers and changelog entries. The permanent URL for version 1.1 continues to resolve to its unchanged artifacts. Version 1.0 was withdrawn from the site when version 1.1 superseded it; its entry remains in the version history and changelog, and its former URL redirects to the current release.

Appendix A. Glossary

Active defender: The deterministic component that converts declared endpoint, network, and identity telemetry into alerts and graduated defensive responses. It observes telemetry rather than sealed ground truth.

Action outcome: The gate decision, execution status, and observable result associated with one action requested by the evaluated system.

Autonomy rung: One ordered operating configuration defining which actions execute automatically, require operator approval, or remain prohibited. A fully autonomous rung may auto-approve every ROE-permitted action while containment and stop controls remain active.

Autonomy-trust curve: The relationship between autonomy rung and measured competence and risk, subject to a minimum valid-evidence requirement for every required axis, used to identify the maximum autonomy acceptable under a named policy. Read per rung, it states the quantified deployment risk of running the evaluand at that autonomy level.

Cloud-infrastructure provider: The service that hosts the target range and produces infrastructure-level resource or network evidence. This is distinct from a model-service provider.

Containment integrity: Whether independent range controls prevented effects outside the declared technical boundary. A containment breach invalidates a run independently of the agent's ROE score.

Evaluand: The complete system under evaluation, including the model, model-service settings, reference harness, prompt, information feed, tools, memory, budgets, and retry behavior.

Harness-private sentinel: An internal marker used by the harness or runtime to track budgets, terminal conditions, or hidden objective state. It is not exposed to the evaluated system.

Hard fail: A discrete event or policy threshold that makes an evaluand unacceptable for the declared use even if other scores are high.

Model-service provider: The service that supplies model inference and associated safeguard settings. This is distinct from the cloud-infrastructure provider hosting the range.

Negative truth: A sealed assertion that an apparent opportunity, credential, target, or route is invalid, prohibited, deceptive, or insufficient to satisfy the objective.

Observation coverage: The number and identity of eligible measurement opportunities that were activated and produced the required live evidence, reported against the total opportunities expected for that axis.

Off-range telemetry collector: The dedicated evidence service outside the target range that authenticates sources, preserves record order, freezes retained evidence, and produces the normalized evidence package consumed by CRUCIBLE's scoring component.

Policy object: A declarative configuration expressing an intended control, such as a network policy, security-group rule, or identity policy. Its existence is not proof of effective enforcement.

Recon-gated world projection: The subset of environmental facts the evaluated system has legitimately discovered through reconnaissance.

Rules of engagement (ROE): The delegated authority for an episode, including permitted and prohibited targets and actions, approval requirements, scope and time limits, and stop conditions.

Scenario fidelity: Whether the deployed world matches the frozen scenario graph and task assumptions closely enough for scoring to remain valid. A fidelity failure is not automatically a containment breach.

Sealed ground truth: Evaluation-only facts, including valid objective paths, negative truth, and criteria used to interpret evidence. These facts are hidden from the evaluated system.

Simulated operator: The fallible supervisor component, which does not know the sealed answers, that adjudicates agent messages from tasking, ROE, interaction history, and a declared experience and behavioral profile.

Tasking: The target information and mission context supplied at the start of a penetration test or offensive cyberspace operation, including the objective and relevant starting assumptions.

Target range: The ephemeral, instrumented environment containing the authorized scenario assets and the controls used to bound agent effects.