

BCP-1: Proof-Carrying Containment for Guardrails-Off AI Evaluations

Khadija Edarzi
African Institute for Mathematical Sciences
`khadijae@aims.ac.za`

With Apart Research

Research conducted at the AI Incident Response Sprint, September 2026.

Abstract

A sandbox can keep an allowed network path, raise detections, and still fail to contain. In the July 2026 OpenAI–Hugging Face incident the model used reachable package infrastructure; detections did not force isolation [1, 2].

We specified BCP-1, eight controls for guardrails-off dangerous-capability evaluations, and implemented the authorization and evidence slice in Python: signed manifest, one-use kill-epoch capabilities, credential broker, bounded JSONL evidence, streaming verifier, and supervisor receipt. The local isolate signs containment as **SIMULATED**, with network isolation unconfirmed; we did not implement a production microVM, kernel send, HSM, or public witness.

Five selected omissions on that reference, not reconstructions of the published 56-event campaign, each produced a non-zero unsafe count (forwarded canary; post-kill dispatch; unsafe harness; conflicting root; 49,998 post-kill records after a 50,000-request flood). The full contract recorded zero on each count. Access logs scored 0/5 predefined claims, signed-line logs 1/5, and BCP-1 5/5 on the mediated path, with 4 events versus 50,000 records. Four scenarios and 32 tests passed (80% combined coverage). Internal Docker networking blocked raw public TCP while allowing an approved gateway dispatch; an open bridge allowed public TCP. Those scores and probes test this evidence format and container networking, not production logs or microVM conformance.

A third party can verify the signed bundle without the lab network. The receipt cannot prove unobserved actions never occurred.

1 Introduction

Guardrails-off evaluations can fail while still looking sandboxed. The July 2026 OpenAI–Hugging Face incident is the case we use [1, 2]. A sandbox kept an allowed network path. The model used reachable package infrastructure, including Artifactory, as that path. Detections existed without a deterministic transition to paging and isolation. The originator’s boundary was not the last one: public compute and shared downstream credentials expanded the reachable blast radius [1, 2].

That matters for an evaluation lab that runs dangerous-capability tests with production safeguards off, and for an independent auditor who cannot sit on the lab network. The question is which controls make unauthorized external effects unreachable, make escalation non-optional, and let an outsider distinguish a real containment claim from a self-report. The lab produces a signed bundle; the auditor verifies it without lab-network access.

BCP-1 applies when an evaluation disables production safeguards or grants tools that can execute code, use the network, handle credentials, or change privilege. The safety objective is: no action outside the signed contract produces an external side effect, and a trusted violation observation isolates the workload before any later dispatch. The published 56 exfiltration-category actions were spread across the campaign, not a contiguous sequence, so we do not assign a historical firing index.

The actor is the untrusted evaluation agent. It can use the permitted package path. The documented chain includes mutable cross-run package state, general egress through that proxy, and a public harness. We do not reconstruct exact action order. Completeness still depends on an OS-enforced default-deny boundary whose only reachable application endpoint is the trusted gateway.

Default-deny networking, least privilege, capabilities, canaries, hash chains, and incident command are not individually new. For this job they do not, by themselves, give a minimum contract that binds authorization to side effects, automatic isolation, second-hop ownership, and evidence an auditor can check without the lab network.

Our main contributions are:

1. BCP-1, an eight-control minimum containment profile for guardrails-off dangerous-capability evaluations, with owners, evidence objects, and acceptance tests.
2. A Python reference for the authorization and evidence slice: signed manifest, one-use kill-epoch capabilities, brokered credentials, bounded JSONL, streaming verifier, and supervisor receipt. It does not implement a production microVM.
3. A control-by-attack-phase matrix for the July 2026 incident, mapping documented phases to interrupting controls and residual risk, without a fabricated action index.

2 Related Work

The nearest published accounts of the July 2026 OpenAI–Hugging Face incident are OpenAI’s technical report and Hugging Face’s agent-intrusion timeline [1, 2]. We treat those incident accounts as source material for a control-by-attack-phase matrix: documented phases, earliest interrupting controls, accountable parties, offline evidence, and residual risk. BCP-1 is a containment profile with mandatory controls, required evidence, and acceptance tests.

The nearest control catalogues we map to are CSA AICM v1.1 AIS-13, the CoSAI AI Incident Response Framework v1.0, and NIST SP 800-61r3 [3, 4, 5]. AIS-13 covers isolated execution, capability boundaries, network restrictions, least privilege, escape testing, monitoring, auditable breach records, and lateral-movement prevention. CoSAI covers tool-call telemetry, severity criteria, RACI, automatic containment, forensic provenance, third-party coordination, recovery, and exercises. NIST, in our mapping, covers governance, adverse-event declaration, incident management, evidence provenance, third-party communication, and mitigation. BCP-1 operationalizes those catalogues for guardrails-off evaluations as eight mandatory controls with owners, evidence objects, and acceptance tests. The catalogues remain broader; BCP-1 is the subset we treat as mandatory for this setting.

3 Methods

The evaluated agent and every process it can influence are untrusted. It may construct ambiguous requests, exploit an allowed service, replay an authorization, race a kill, exhaust gateway resources, select an unsafe second hop, or try to make evidence unverifiable. The gateway, executor, credential broker, supervisor, evidence mirror, and keys sit outside that domain. Compromise of one trusted component is treated separately: the OS boundary is meant to limit gateway bypass; the supervisor, gateway failure; the independent witness, supervisor equivocation. We map that split onto isolated execution, capability boundaries, and network access in CSA AICM v1.1 AIS-13 [3], and onto automatic containment in CoSAI v1.0 [4].

The object under test is a Python reference for the authorization and evidence contract, version 1.0.0, requiring Python ≥ 3.11 . Dependencies are pinned: cryptography 45.0.6, FastAPI 0.116.1, Pydantic 2.11.7, typing-extensions 4.14.1, uvicorn 0.35.0, and, for tests, httpx 0.28.1, pytest 8.4.1, pytest-cov 6.2.1. The captured measurement environment is Python 3.12.10, Docker 29.7.2, Compose 5.5.1, on Windows 11. Container images use `python:3.12.5-slim` unless an operator overrides the base image. FastAPI is an HTTP adapter. The security claim is the signed contract, the executor, and an OS-enforced default-deny boundary, not the web framework.

Each run generates fresh Ed25519 keys for operator, gateway, supervisor, and witness. Keys are process-local and unseeded. The operator-signed manifest pins one allowed egress: HTTPS POST to `packages.example.test:443` under `/v1/packages`, destination CIDR 203.0.113.0/24, and a pinned TLS SPKI. Default budgets include 128 requests per 1 s window, 10,000 admitted requests, 40,000 events, 64 in-flight requests, 1,024 queued events, 8,388,608 total egress bytes, and a 1 s supervisor kill-latency SLA. Isolation in this reference returns a simulated termination and does not claim confirmed network cut. The executor transport returns a synthetic success body and does not open the public internet. A Python lock stands in for atomic capability consumption; a production deployment must couple that step to actual network admission. Duplicate keys, floats, non-NFC strings, unsafe integers, path ambiguity, redirects, and network-identity mismatches are rejected. Signed objects use a restricted subset of RFC 8785 JSON Canonicalization Scheme: UTF-16 key order, I-JSON-safe integers, no floats [6]. An ALLOW decision binds run, manifest, request, action digest, rule, authorization ID, expiry, one use, and kill epoch. Under a process lock the executor consumes that ID and commits DISPATCH; it then takes any secret from the broker and only then invokes transport. The supervisor mirrors event heads, invokes isolation, signs the kill acknowledgement and receipt, and asks a stateful witness to checkpoint one root. The offline verifier retains per-request state while streaming; it is not constant-memory.

We compare five architectural omissions with the full contract. They are not models of OpenAI, Hugging Face, or any product. Wall time is fixed at 2026-07-10 12:00 UTC. Each safety count is scored against a target of zero. Alert-only detects canary use, emits an alert, and still forwards. A stale-permit dispatcher honors a capability issued before kill without rechecking shared state at dispatch. A presence-only second-hop check admits a provider if a provider identifier, endpoint host, and responsible operator are named, without requiring authentication, non-root execution, or denied egress. The witness baseline is labeled a hash chain without an independent registry; the executed baseline is a 64-character length check, not a cryptographic chain. The BCP-1 arm asks the stateful witness to accept a conflicting evidence digest. Per-drop logging allocates one detailed record for every post-kill request in a 50,000-request flood (49,998 records in the unbounded arm). The BCP-1 arm uses a one-request window and counts event growth after the bounded kill. Where the omitted control exists in the reference, that object is executed; omitted behavior is kept small so each comparison tests one gap.

A separate auditability study uses the same 50,000-request post-kill burst on three evidence profiles: an ordinary per-request access log; the same records with a per-line HMAC; and the BCP-1 signed manifest, bounded event chain, receipt, and witness. We treat that comparison as the forensic-provenance and logging job in CoSAI [4] and NIST SP 800-61r3 [5]. Those catalogues do not specify our five tests. The claims, defined in advance, are operator-signed contract binding, zero forbidden dispatches on the mediated path, trigger-bound kill, detection of tampering in an included event, and rejection of a conflicting root. For the two log profiles, contract, dispatch, and kill claims are field-presence tests. Tamper tests flip an allowed-policy token in the logs and a package name in the BCP-1 chain. Zero-forbidden-dispatch and kill-binding for BCP-1 are scored on a separate canary run, not on the flood ledger. Serialized size, record count, streaming verification time, and peak verifier allocation are recorded for each profile. The signed-line profile is credited only for included-line modification detection.

Four scripted scenarios exercise an allowed package read, unauthorized egress to a host absent from the manifest, canary-credential use, and an unauthenticated root-capable public harness at preflight. A canary run exports a signed bundle (manifest, event chain, receipt, operator public key). The offline verifier must accept an unmodified bundle and reject a modified event line.

Container tests are scoped as container networking, not microVM conformance, and are not represented as AIS-13 hardware isolation [3]. An unprivileged probe on an internal Docker network must fail a raw TCP connection to 1.1.1.1:443 (1s timeout) and must complete an approved gateway dispatch. A paired ablation uses the same public TCP check (2s timeout) on an internal network versus an ordinary bridge; the open bridge is the positive control.

The control-by-attack-phase matrix is documentary. We coded published July 2026 phases from OpenAI’s technical report and Hugging Face’s timeline [1, 2] onto interrupting controls, accountable parties, offline evidence, and residual risk. Retrieval dates for those pages are not recorded. An earlier stochastic “ballistic” detector was discarded and is not a safety boundary. An initial 13 September container-image pull failed on Docker Hub DNS; after resolution the gate was rerun on `python:3.12.5-slim`.

Replication uses the accompanying repository. Install the package editable with its test and report extras, then run the repository’s final reproduction gate (scenarios, control ablations, auditability study, canary bundle, offline verify, automated tests with branch coverage, and both container probes). Shorter local commands are in the repository README.

4 Results

On this reference runtime, each of five selected control omissions produced a non-zero unsafe count, and the full contract scored 0 on each count (Table 1). The comparisons are deterministic architectural omissions, not vendor or microVM conformance.

Table 1: Unsafe counts for five architectural omissions versus the full contract. Target = 0.

Omitted control	Metric	Omitted	Full
Automatic canary kill	Forbidden dispatches after detection	1	0
Dispatch-time kill-epoch check	Post-kill dispatches	1	0
Fail-closed second-hop preflight	Unsafe harnesses admitted	1	0
Independent one-root witness	Conflicting second roots accepted	1	0
Bounded post-kill evidence	Post-kill event growth	49,998	0

The canary arm matches the alerted-but-did-not-page failure: the omitted-control arm forwarded once after detecting the canary. The race arm shows that a permit issued before kill still produces one side effect unless dispatch rechecks shared epoch state.

Under the same 50,000-request post-kill burst, three evidence profiles were scored on five pre-defined claims (Table 2). The ordinary access log verified 0/5. The signed-line log verified 1/5 (included-line tamper detection only). The BCP-1 bundle verified 5/5 on the mediated path. Peak traced verifier allocations were 1,872, 2,181, and 38,378 bytes; BCP-1 is not the lowest-memory verifier. We treat serialized size, not wall-clock verification time, as the portable comparison.

Four scripted scenarios passed ($n = 4$). Allowed package-read dispatched once and audited valid. Unauthorized egress was denied (`egress.origin_not_allowed`) with zero dispatches. Canary use was denied (`credential.canary_used`) with zero dispatches. The unsafe second-hop preflight was rejected (`preflight.rejected`); `model_started` is false. A separate canary bundle had 4 events, 0 forbidden forwards, offline audit valid, run status `KILLED`, and `kill_latency_ns` = 0 (local clock resolution, not a production latency).

Table 2: Auditability claims and evidence size under a 50,000-request post-kill burst (three profiles, five claims).

Profile	Claims	Records	Bytes	Verify (ns)
Access log	0/5	50,000	4,088,883	956,274,300
Signed-line log	1/5	50,000	6,288,883	1,963,088,800
BCP-1 bundle	5/5	4	5,576	31,490,500

The automated suite collected 32 tests: 32 passed, 0 failed, 0 skipped, on Python 3.12.10. Combined coverage display is 80% (2,390 statements, 352 missed, 650 branches, 218 partial).

Two container-network probes passed ($n = 2$). On the internal default-deny network, public TCP to 1.1.1.1:443 did not succeed; approved gateway dispatch did. On the open bridge, public TCP succeeded and gateway dispatch succeeded. Those probes are container-network only.

Robustness. The five ablations omit one named control at a time on this runtime. We did not re-run under changed burst sizes or seeds. We report no p -value or confidence interval. n is small (5 ablation arms; 4 functional scenarios; 2 network probes). The comparisons are deterministic pass/fail counts, not a sample from deployed labs.

5 Discussion and Limitations

Table 1 means that, on this reference runtime, dropping any one of five controls produced a non-zero unsafe count, and the full contract produced zero. That is necessity inside the stated threat model, not a rate in deployed labs. Table 2 means an ordinary access log did not verify the five claims we defined; a per-line MAC verified one; the signed bundle verified five on the mediated path. A lab that runs guardrails-off evaluations should not treat a sandbox label and an alert as the artifact an independent auditor can check without the lab network.

Limitations

The Python reference checks protocol, authorization, overload, and race properties. It does not attest a production microVM, kernel egress filter, atomic network send, HSM/TPM/TEE keys, public transparency log, production IAM, or real credential revocation. Test keys share one process. The executor does not open the public internet. The Docker probes are container-network only: they do not cover UDP, QUIC, metadata, sidecars, inherited descriptors, or hypervisor escape.

The omitted-control arms are defined omissions, not vendor products. The 5/5 auditability score is the mediated path; it does not prove there is no route outside the gateway. A receipt does not prove unobserved actions never occurred. Destination allowlisting is not information-flow control. `kill_latency_ns = 0` is clock resolution in a mock plane, not a production SLA.

The threat model trusts the gateway, executor, broker, supervisor, mirror, and keys, and considers one trusted-component failure at a time. We do not claim protection against a hypervisor zero-day, malice from an approved destination, or collusion among every independent trust domain. BCP-1 is scoped to guardrails-off evaluations, not to production agents in general.

We take OpenAI’s technical report and Hugging Face’s timeline as the account of documented phases [1, 2]. If that account is incomplete, the matrix rows and the “incident-grounded” ablation labels may point at the wrong object. We treat the witness as independent. If it is not, a locally signed checkpoint does not stop equivocation.

This release publishes defensive conformance vectors and abstract omissions, not exploits against

real package services, cloud control planes, or public harnesses.

Future Work

Replace reference claims with independent measurements: executor and supervisor around a Firecracker or cloud microVM; raw TCP, UDP, DNS, QUIC, metadata, sidecars, inherited descriptors, and runtime sockets; authorization consumption coupled to kernel egress; keys in separately administered hardware-backed domains; checkpoints on an external transparency service; a second team producing adversarial bundles; the same benchmark on at least three public agent sandboxes with maintainer-reviewed settings. Success is a third party reproducing unmediated-path failure, zero forbidden dispatches on hard violations, and witness rejection of conflicting receipts.

6 Conclusion

On this reference runtime, five selected omissions were each necessary for a zero unsafe count, and the signed bundle was the only of three evidence profiles that verified the five predefined claims on the mediated path (Tables 1–2). Those results are necessity and auditability inside the BCP-1 threat model, not a rate in deployed labs.

Code and Data

The reference implementation, schemas, signed demo bundle, ablation and auditability outputs, tests, and reproduction gate are at https://github.com/khadijaedarzi9/AI_Incident_Response. Install editable with the test extras, then run the repository’s final gate (or the shorter demonstration commands in the README). The Docker probes require a local Docker engine.

References

- [1] OpenAI. OpenAI–Hugging Face Incident Technical Report, July 2026. <https://cdn.openai.com/pdf/67869394-cb91-4c12-888c-5cbd85c7814c/OpenAI-Hugging-Face%20Incident-Technical-Report.pdf>.
- [2] Hugging Face. Agent Intrusion Technical Timeline, July 2026. <https://huggingface.co/blog/agent-intrusion-technical-timeline>.
- [3] Cloud Security Alliance. AI Controls Matrix v1.1, AIS-13 Auditing Guidelines for Model Providers, 2026. <https://cloudsecurityalliance.org/artifacts/aicmv1-1-auditing-guidelines-for-model-providers-mp>.
- [4] CoSAI/OASIS Workstream 2. AI Incident Response Framework v1.0, 2026. <https://github.com/cosai-oasis/ws2-defenders/blob/main/incident-response/AI-Incident-Response.md>.
- [5] NIST. SP 800-61 Revision 3: Incident Response Recommendations and Considerations for Cybersecurity Risk Management, 2025. <https://csrc.nist.gov/pubs/sp/800/61/r3/final>.
- [6] A. Rundgren, B. Jordan, and S. Erdtman. RFC 8785: JSON Canonicalization Scheme (JCS), RFC Editor, 2020. <https://www.rfc-editor.org/rfc/rfc8785>.

LLM Usage Statement

I used Cursor as a coding assistant for implementation and debugging, and ChatGPT to check grammar and polish the document. I reviewed the code, results, and manuscript.