AI Attack Surface Map

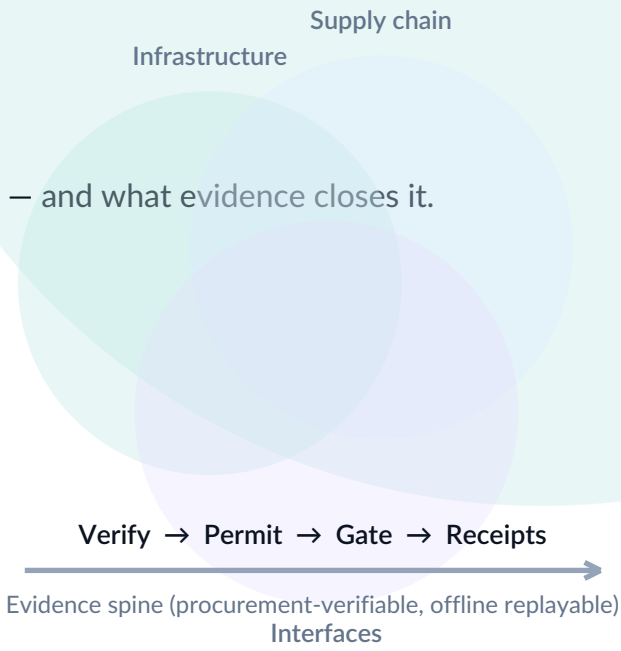
Infrastructure • Supply chain • Interfaces

Evidence spine

Verify → Permit → Gate → Receipts

Receipts, not promises.

A procurement-grade map of where GenAI breaks — and what evidence closes it.



Meridian Verity Group • meridianverity.com
Version 2.0 • March 2026

The three places GenAI breaks

GenAI security failures cluster into three surfaces: the execution environment (infrastructure), the artifacts you pull in (supply chain), and the user + API entry ramps (interfaces). This map translates each surface into reviewer checklists and evidence requirements — so teams can verify offline and fail closed when proof is missing.

Infrastructure

What fails

- Over-privileged IAM + weak auth/authz/logging on new AI resources (RAG/MCP/tool runners).
- Secret sprawl in pipelines (keys, PII, code).

What MVG enforces

- Gate side effects at egress, dispatch, promotion.
- Short-lived scoped permits + replayable receipts.

Supply chain

What fails

- Poisoned models/data/deps pulled from registries or mirrors.
- Unverified orgs/accounts and unsafe artifacts evading scanners.

What MVG enforces

- Pin + sign artifacts (digest + provenance) before promotion.
- Receipts bind policy version + artifact identity + outcome.

Interfaces

What fails

- Direct/indirect prompt injection driving tool misuse.
- Data exfiltration via overly broad tool/data access.

What MVG enforces

- Capability-scoped tools/data + deny on uncertainty.
- Reason-coded PASS/FAIL/HOLD for reviewers.

What makes MVG different

Most vendors observe. MVG proves and blocks: deterministic verifier, portable conformance pack, and stable reason codes — so reviewers can replay decisions offline.

Procurement translation

Replace narrative assurances with evidence your reviewers can replay offline. If evidence is missing, stale, or unverifiable: HOLD — and side effects remain blocked.

Ask for	Deterministic verifier + portable conformance pack + receipt schema (offline runnable).
Verify	Policy pinning, artifact identity, permit scope/expiry, non-bypassable gates.
Decide	PASS / FAIL / HOLD with stable reason codes (no free-text narratives).

Surface 1: Infrastructure

Execution environments mix established cloud services with emerging AI components like MCP servers and RAG pipelines. Misconfigurations and mismanaged credentials can open initial access, credential theft, and discovery paths.

Common failure modes

- Over-privileged IAM roles and public-facing services expanding blast radius.
- Long-lived credentials and exposed secrets inside clusters and pipelines.
- Insufficient authentication, authorization, and logging around new AI resources.

MVG control points

Verify	Prove component identity + policy freshness/continuity before execution.
Permit	Mint short-lived, scoped permits for each side effect (tool call, write, egress).
Gate	Enforce at non-bypassable control points (dispatch, egress, promotion).
Receipt	Emit canonical bytes with stable reason codes for every decision.

Stable reason codes (examples)

`INF-AUTH-001`	authn/z missing or weak
`INF-SECRET-002`	secret exposure / long-lived creds
`INF-GATE-003`	side effect blocked (no permit)

Reviewer checklist

Inventory	List MCP servers, RAG stores, model endpoints, and data connectors.
Authn/z	Short-lived credentials; least privilege by default; deny on uncertainty.
Logging	Immutable decision receipts; replay verification without vendor access.
RAG	Secret scanning + data classification; block poisoned entries from influencing actions.
Tests	Deterministic PASS/FAIL/HOLD vectors for initial access and credential misuse.

Surface 2: Supply chain

Supply chains include training datasets, pre-trained models, prompts, and third-party AI libraries. Organizations can unknowingly pull malicious or poisoned artifacts into production.

Common failure modes

- Poisoned training data or models introduced through third-party sources.
- Unverified organizations/accounts on public repositories leading to compromised models.
- Dependencies that evade scanners or exploit risky serialization formats.

MVG control points

Policy pinned	Explicit allowlist for registries, orgs, versions, and hashes.
Signed artifacts	Sign models, prompts, policy packs, and build outputs (digest + provenance).
Promotion gate	Block deployment unless provenance checks and policy evaluation PASS.
Receipts	Bind artifact identity, policy version, attestation inputs, and outcome.

Stable reason codes (examples)

`SUP-PROV-001`	provenance missing / untrusted
`SUP-PIN-002`	floating tag / hash not pinned
`SUP-GATE-003`	promotion blocked (policy FAIL)

Reviewer checklist

Provenance	Show where the artifact came from (registry + org + digest) and who signed it.
Pinning	Deny floating tags; require explicit versions and hashes.
Attest	Attach SBOM/model metadata and verification outputs to the receipt.
Drills	Run a revoke + re-verify drill to prove compromised artifacts fail closed.

Surface 3: Interfaces

Interfaces are the entry points and business logic that connect users to models and supporting services. Because user input is unpredictable, prompt-level attacks can blend into normal traffic and trigger execution or exfiltration.

Common failure modes

- Direct and indirect prompt injections that hijack model behavior.
- Execution of harmful code embedded in artifacts or retrieved context.
- Data exfiltration from connected services when tool/data access is too broad.

MVG control points

Capability scoping	Tool and data access are explicit, least-privilege, and reason-coded.
Non-bypassable gates	All side effects go through a gate that requires a valid permit.
Deterministic outcomes	Model outputs are not actions; actions require evidence + policy PASS.
Receipts	Each tool call is logged with inputs, policy decision, and stable reason code.

Stable reason codes (examples)

`IFACE-CAP-001`	capability out of scope
`IFACE-INJ-002`	prompt injection detected
`IFACE-EXFIL-003`	exfil attempt blocked

Reviewer checklist

Tool map	Enumerate tools + data connectors; show allowed operations and denial modes.
Injection tests	Provide red-team prompts and expected HOLD/FAIL outcomes.
Exfil guard	Demonstrate sensitive sources cannot be retrieved without permit.
Replay	Re-run verification offline and reproduce the same PASS/FAIL/HOLD outcomes.

Procurement-ready evidence pack

Convert the attack surface map into reviewable artifacts: a deterministic verifier, portable conformance packs, and receipts with stable reason codes. Taxonomy source: Datadog AI Security Best Practices eBook.

Your RFP question	Evidence artifact
How do you prevent side effects when the model is uncertain?	Fail-closed gate + scoped, expiring permits. No permit, no action.
Can our team verify without vendor access?	Offline verifier + deterministic replay of receipts and test vectors.
How do you control model and dependency updates?	Signed promotion receipts; pinned versions/hashes; provenance checks.
How do you prove prompt-injection resistance?	Red-team vectors with expected PASS/FAIL/HOLD + stable reason codes.

Two-minute reviewer flow

- Download the Conformance Pack (portable, mirror-safe).
- Run the offline verifier to replay receipts and test vectors.
- Read PASS/FAIL/HOLD outcomes and stable reason codes.
- Spot-check the highest-risk control points (egress, dispatch, promotion).

```
$ mvg verify pack ./conformance-pack.tgz
$ mvg verify receipts ./receipts/ --policy ./policy.json
$ mvg report --format summary
```

Receipt snapshot (canonical, reason-coded)

```
{
  "receipt_version": "1",
  "policy_id": "mvg/policy/2026-03-01",
  "artifact_digest": "sha256:...",
  "decision": "HOLD",
  "reason_code": "IFACE-CAP-001",
  "permit_expiry": "2026-03-01T12:34:56Z",
  "signed_by": "mvg-witness-2of3"
}
```