hoop.dev

TECHNICAL GUIDE

FRAMEWORK **OWASP GenAI Data Security Risks**

OWASP GenAI Compliance for Humans and AI Agents

The OWASP GenAI Data Security framework defines 16 risk categories. This guide connects those risks to Hoop.dev controls that secure your data and improve engineering productivity.

- 11 of 16 OWASP risks addressed through a single gateway
- Covers humans, AI agents, and non-human identities
- Change the risk profile of data-in-transit with data masking and in-session guardrails

16	4	4	3
OWASP risk categories	Direct control matches	Contributing controls	Adjacent references

THE CORE PROBLEM

Restricting AI context doesn't solve the problem, it just delays the solution

Data leakage, credential sprawl, and uncontrolled SQL generation all existed before generative AI. AI agents just make these failures happen faster. Most teams respond by locking down access. That slows engineers and frustrates security teams without fixing the root cause. The OWASP GenAI framework calls for stronger controls at the data layer. Hoop.dev satisfies those requirements with in-session governance at the infrastructure boundary.

1 Requester

An engineer, AI agent, or pipeline needs production data. Today this means a slow approval chain or no controls at all.

2 In-Session Control

Hoop.dev intercepts at the protocol layer. Masking, firewall rules, and JIT reviews happen inline. No credentials stored. Risk profiles change on the fly.

3 Compliant by Default

Access granted in seconds. PII redacted. Commands logged. Audit trail generated. No bottleneck. No blind spot.

DIRECT CONTROL: WHERE HOOP ELIMINATES THE BOTTLENECK

DSGAI01

Sensitive Data Leakage

OWASP requires structured masking on inputs, outputs, and logs. Hoop masks 150+ PII types at the protocol layer in real time. Data is redacted before any human or NHI sees it. No separate review step needed.

DSGAI02

Agent Identity & Credential Exposure

OWASP recommends JIT access for AI agents. Hoop does exactly this. Tokens are scoped to a task and revoked when it ends. The stateless gateway stores no credentials. It deploys in a day and eliminates NHI sprawl.

DSGAI12

Unsafe LLM-to-SQL Gateways

AI copilots can generate raw SQL against wide schemas. Hoop intercepts at the command level. Firewall rules block dangerous patterns. Masking redacts sensitive columns. Risk profiles change per session with no redeployment.

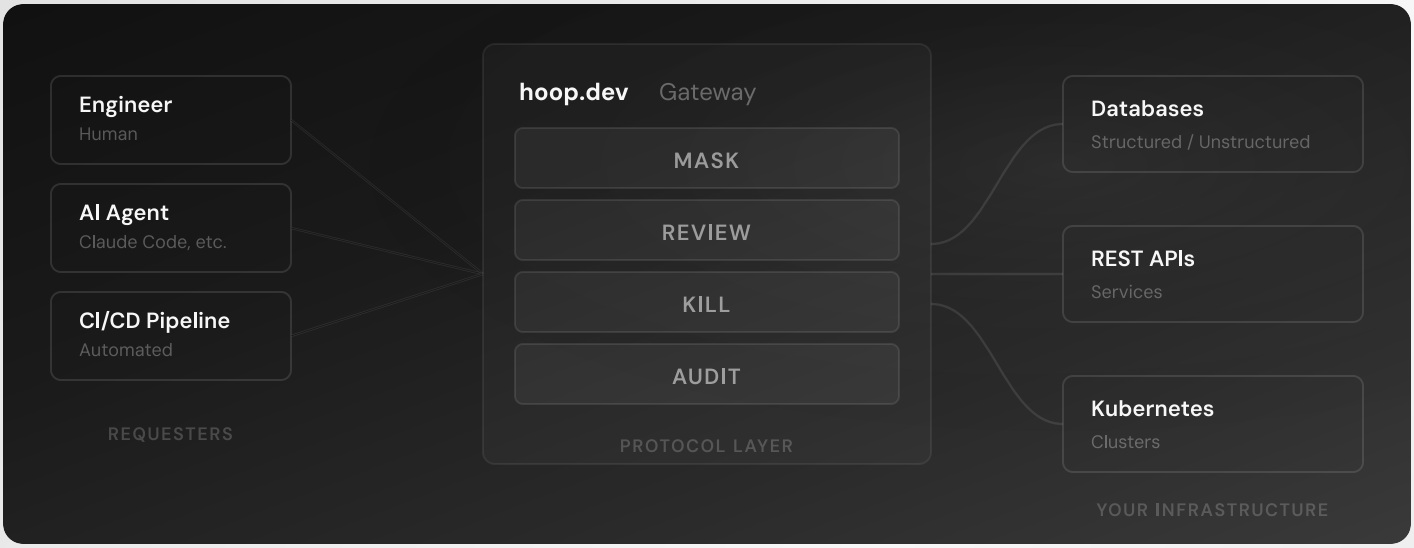
DSGAI08

Non-Compliance & Regulatory Violations

OWASP requires records of processing activities. Hoop records every session: who accessed what, when, which commands ran, and what data came back. This generates audit evidence for SOC 2, PCI DSS, HIPAA, and GDPR automatically.

IN-SESSION CONTROL FOR EVERY IDENTITY

One gateway for humans and non-human identities, down to the command level



CONTRIBUTING CONTROL: PART OF THE MITIGATION STACK

DSGAI03

Shadow AI & Unsanctioned Data Flows

Hoop doesn't control which AI tools people use. It controls what data those tools can reach. If access goes through Hoop, the data is already redacted before it hits any unsanctioned tool.

DSGAI06

Tool, Plugin & Agent Data Exchange

AI agents call tools that touch databases and clusters. Hoop sits at that boundary. It limits what data comes back through masking and logs every interaction at the command level.

DSGAI07

Data Governance & Lifecycle

Group-based RBAC, JIT reviews, and time-bounded sessions ensure the right identity gets the right data for the right duration. Risk profiles adjust on the fly with no redeployment.

DSGAI15

Over-Broad Context & Prompt Over-Sharing

AI systems pull production data to improve answers. Hoop's masking redacts PII before it leaves the database layer. Data minimization happens upstream, inline, before the model ever sees it.

QUICK REFERENCE

OWASP GenAI risk-to-control mapping

RISK ID	RISK NAME	HOOP CONTROL	TIER
DSGAI01	Sensitive Data Leakage	AI data masking (150+ PII types)	DIRECT
DSGAI02	Agent Identity & Credential Exposure	JIT access, stateless gateway	DIRECT
DSGAI12	Unsafe LLM-to-SQL Gateways	Command interception, firewall policies	DIRECT
DSGAI08	Non-Compliance & Regulatory Violations	Immutable session recording	DIRECT
DSGAI03	Shadow AI & Unsanctioned Data Flows	Data minimization at boundary	CONTRIBUTING
DSGAI06	Tool & Agent Data Exchange Risks	Context minimization, observability	CONTRIBUTING
DSGAI07	Data Governance & Lifecycle	RBAC, JIT reviews, time-bounded sessions	CONTRIBUTING
DSGAI15	Over-Broad Context Windows	Upstream masking at infra layer	CONTRIBUTING
DSGAI11	Cross-Context Conversation Bleed	Per-user session isolation	ADJACENT
DSGAI14	Excessive Telemetry Leakage	Masking applies to audit logs	ADJACENT
DSGAI16	Endpoint Assistant Overreach	Access gate for all requesters	ADJACENT

The key insight: Compliant access should be the fastest path, not a bottleneck. Every OWASP GenAI risk traces back to what humans and NHIs can reach. In-session controls at the protocol layer let you manage what every identity can see and do. You can change the risk profile on the fly.

CONTROL GLOSSARY

How Hoop.dev gateway controls map to OWASP GenAI framework language

MASK → **AI Data Masking**

Redacts 150+ PII types in real time at the protocol layer. Data is filtered before any human or NHI sees it. Maps to DSGAI01 and DSGAI15.

REVIEW → **JIT Access Reviews**

Time-bounded sessions with approval gates. No standing credentials. Tokens scoped to a task and revoked when done. Maps to DSGAI02 and DSGAI07.

KILL → **Firewall Policies & Command Interception**

Blocks dangerous query patterns. Controls outbound traffic. Risk profiles change per session with no redeployment. Maps to DSGAI12 and DSGAI06.

AUDIT → **Immutable Session Recording**

Full request/response traceability for every human and NHI session. Generates audit evidence for SOC 2, PCI DSS, HIPAA, and GDPR. Maps to DSGAI08.