# Kien

**AI Security | Platform Engineering | Cloud Architecture**

Email: lkien201@gmail.com Portfolio: https://kiendev.space

## PROFILE

Computer Science graduate from Arizona State University with experience across cybersecurity engineering, DevOps/platform delivery, secure AI workflows, Kubernetes operations, CI/CD, observability, and solution architecture. I work well on systems where reliability, security, and product delivery overlap: model-backed features with real tool permissions, cloud services that need clean release paths, and security findings that need practical remediation.

## CORE STRENGTHS

- **Secure AI systems:** LLM gateway design, tool governance, RAG boundaries, eval checkpoints, trace review, policy checks, audit logging, rollback planning.
- **Platform engineering:** Kubernetes delivery, CI/CD gates, deployment hygiene, environment separation, service handoff, observability, incident runbooks.
- **Security engineering:** application/API review, access-control analysis, threat modeling, hardening plans, evidence handling, remediation workflow.
- **Solution architecture:** service boundaries, identity flows, data movement, control registers, architecture decision records, rollout planning.

## EXPERIENCE

### Cybersecurity Engineer - Twilio

**2024 - 2025**

- Reviewed authentication, authorization, data exposure, and abuse-case paths through the lens of how services behave in production.
- Helped convert security observations into practical guardrails, runbooks, and fix language instead of leaving teams with vague risk statements.
- Worked with telemetry, service ownership, incident context, and release realities while keeping private details out of public artifacts.
- Kept security communication bounded: what is affected, what is proven, what is not claimed, and what should change.

### DevOps Engineer - Dyson

**2023 - 2024**

- Worked on CI/CD and environment workflows with attention to release promotion, repeatability, rollback, and operational readiness.
- Supported container and Kubernetes-style operations with service visibility, practical runbooks, and ownership boundaries.
- Improved the quality of troubleshooting signals across application and infrastructure layers so issues were easier to isolate.
- Built documentation and handoff notes around real operator questions: what changed, what failed, who owns it, and how to recover.

### Vulnerability Researcher - Bug Bounty and HackerOne Programs

**2024 - Present**

- Performs authorized vulnerability research focused on access-control boundaries, mobile/API behavior, auth state transitions, and proof packages that can survive triage.
- Builds attacker/victim/object matrices for authorization testing instead of stopping at one-off suspicious responses.
- Separates local-only behavior, expected denial, backend impact, and reportable exposure before writing a claim.
- Turns raw testing into triage-ready evidence: request pairs, denied controls, impact narrative, remediation notes, and clear non-claims.
- Keeps public portfolio material sanitized: no target secrets, private reports, unsafe technical detail, or customer data.

## SELECTED WORK

### AI Security Control Plane

Production-oriented design for routing model traffic through identity-aware policy, retrieval checks, tool permissions, eval gates, logs, and replayable incidents.

### Kubernetes Delivery Platform

Delivery model for services that need predictable builds, Kubernetes readiness, progressive rollout, observability, ownership, and rollback instructions.

### Pentest Evidence Workflows

Report-building workflow for scoped testing: request pairs, object-boundary checks, denied controls, impact notes, false-positive reduction, and remediation language.

### MLOps and LLMOps Observability

Operating model for model-backed products: trace fields, evaluation checkpoints, prompt and retrieval release notes, cost visibility, latency budgets, and incident review.

## EDUCATION

### Bachelor of Science in Computer Science - Arizona State University

**2019 - 2023**

Software engineering, data structures, algorithms, systems thinking, and maintainable production software.

## TECHNICAL SKILLS

**AI / LLM:** OpenAI API, RAG, agents, prompt risk review, evals, trace review, LLMOps, tool governance.

**Cloud / DevOps:** Kubernetes, Docker, CI/CD, observability, infrastructure-as-code patterns, SLOs, release and rollback.

**Security:** threat modeling, OWASP, pentest evidence, access control, API security, hardening, HackerOne.

**Architecture:** solution design, IAM, service boundaries, runbooks, ADRs, risk registers, stakeholder handoff.

## SECURITY BOUNDARY

Security work is authorized, scoped, and sanitized. Public examples do not include private target details, secrets, customer data, unsafe reproduction detail, or confidential implementation information.