# Prashant Kulkarni

925-678-0849 kulkarniprashants@gmail.com LinkedIn GitHub Google Scholar

## Summary

Security Architect with 20+ years in enterprise security, specializing in AI control, mechanistic interpretability, and adversarial defensive architectures for agentic AI systems. Published work on multi-turn manipulation attacks and prompt injection defenses. Current research on latent adversarial detection via activation trajectory analysis and micro-protocols for safe deployment of untrusted AI agents.

## Selected Research

**Latent Adversarial Detection (LAD):** arXiv:2604.28129

- Detecting adversarial intent from LLM internal activations using mechanistic interpretability; discovered "adversarial restlessness"—multi-turn attacks produce distinctive cumulative activation drift in the residual stream
- Six scalar trajectory features (drift magnitude, cosine similarity, cumulative drift, acceleration) with XGBoost probe achieve 93.8–97.2% conversation detection at 0.5–3.5% FP rate across 4 model families (Gemma, Mistral, Qwen, Llama; 24B–70B)
- Designed synthetic multi-turn adversarial conversation dataset with matched clean/injected pairs for probe training; generation pipeline covers trust-building, pivot, and escalation phases
- Validated on 797 held-out conversations; targeting EMNLP 2026 submission

**SWE-bench-IPI: Indirect Prompt Injection Benchmark for Coding Agents:** GitHub

- Designing benchmark for evaluating coding agent resilience to indirect prompt injection; built on SWE-bench Verified (500 real-world Python bug fixes across 12 repositories)
- 13 attack surfaces (issue text, code comments, docstrings, config files, etc.) across 4 stealth levels with MITRE ATT&CK-based payloads; multi-agent evaluation harness covering Claude Code, Codex CLI, Gemini CLI, and SWE-agent
- Deterministic token-matching evaluation (no LLM-as-judge); measures both attack success rate and utility degradation under attack

**Micro-Protocols for AI Control:** *Astra Fellow, Redwood Research* *Jan 2026 – Mar 2026*

- Designing composable security protocols for agentic AI deployments: Destructive Editing, Trusted Summary, and Untrusted Plan micro-protocols that enable graduated response to adversarial behavior
- Developed Sanitizer model architecture that isolates trusted verifiers from adversarial content—preventing context contamination in multi-model oversight systems
- Building evaluation framework with tiered threshold system mapping monitor scores to action types (Verify/Edit/Replace); Pareto frontier analysis for safety vs. usefulness trade-offs
- Evaluating frontier open-weight models (Kimi-K2.5) for cyber capability risks
- Participating in AI research sabotage detection study—auditing codebases for adversarial modifications by AI systems, informing protocol design for detecting goal subversion

**Capability-Constrained Control (C3):** GitHub

- Developing data-flow constraint enforcement library for AI coding agents with taint tracking, resource provenance analysis, and capability grants
- Dual-gate defense architecture (pre-check + runtime enforcement) integrated with Inspect AI eval framework
- Evaluated on ControlArena-based scenarios across credential exfiltration, malware download, and privilege escalation categories

**CausalArmor: Indirect Prompt Injection Defense:** GitHub

- Open-source Python library (PyPI) implementing causal attribution defense against indirect prompt injection in tool-using LLM agents (arXiv:2602.07918)
- Uses leave-one-out causal attribution via proxy model to detect whether agent actions are driven by user intent or poisoned tool output; sanitizes and regenerates when attack is detected
- Benchmarked on AgentDojo (11,322 scenarios), achieving 18–24pp reduction in attack success rate; supports vLLM, OpenAI, Anthropic, Gemini, and LiteLLM

**LLM Sandbox Escape Evaluation:** GitHub

- Built AI-vs-AI adversarial testing framework: Gemini Pro attacker vs. Gemini Flash victim agent with MCP tool integration
- Tested 54 attack prompts across 5 categories—100% blocked by defense-in-depth architecture; key finding: agent reasoning alone insufficient, server-side validation caught attacks that bypassed AI-level refusal
- Implemented secure MCP server with command whitelist, path validation, and input sanitization—reusable benchmark for containment evaluation

**Sandbox Containment Evaluation (SPAR):** GitHub

- Leading empirical study on LLM agent containment resilience using UK AISI Inspect framework; testing privilege escalation and escape attempts against gVisor, Firecracker, and restricted Python sandboxes
- Developing model-agnostic benchmarking toolkit quantifying real-world containment risk for agentic deployments

**Automated AI Red Teaming: Models and Agentic Systems:** agentic-tool-misuse | agent-tools-attack-vectors

- **Model-level red teaming:** Building automated pipelines with DeepTeam, NVIDIA Garak, and Promptfoo to probe base LLMs for OWASP LLM Top 10 and NIST AI RMF categories—jailbreaks, harmful content, bias, hallucination, and prompt leaking
- **Agentic red teaming:** Designing attack harnesses for tool-using agents targeting indirect prompt injection, tool misuse, goal redirection, memory poisoning, and cross-context data leakage; released as agentic-tool-misuse and agent-tools-attack-vectors, evaluated on AgentDojo and ControlArena-style scenarios
- **LLM-driven prompt synthesis:** Using a generator model to produce creative single-turn attacks (role-play, obfuscation, encoding) and multi-turn conversational strategies (trust-building, topic pivots, gradual escalation)
- Extending these frameworks with custom attack strategies and domain-specific vulnerability templates; integrating Garak probes and Promptfoo assertions into regression suites to track robustness across model and policy updates

**Multi-Turn Manipulation Defense:** *IEEE CAI 2025 (Accepted)*

- Published "Temporal Context Awareness: A Defense Framework Against Multi-turn Manipulation Attacks on Large Language Models"—demonstrated vulnerability patterns where LLMs succumb to gradual trust-building; designed detection mechanisms for context poisoning and escalation attacks

## Open Source Security Research

| | | |
|---|---|---|
| **latent-adversarial-detection** | Activation-level adversarial intent detection via mechanistic interpretability | GitHub |
| **causal-armor** | Causal attribution defense against indirect prompt injection (PyPI) | GitHub |
| **swe-bench-ipi** | Indirect prompt injection benchmark for coding agents (SWE-bench Verified) | GitHub |
| **agentic-tool-misuse** | Detection system for tool misuse patterns in LLM coding agents | GitHub |
| **agent-tools-attack-vectors** | Tool misuse and capability abuse patterns in agents | GitHub |
| **mcp_security_test_agent** | MCP protocol security testing framework | GitHub |
| **guardrails-ai** | Production guardrails for LLM input/output validation | GitHub |
| **weak-to-strong-gen** | Empirical study on scaling thresholds for reasoning supervision | GitHub |

## Professional Experience

**AI Security Research Engineer** Oct 2019 – Present

*Google Cloud* *Mountain View, CA*

- Lead security research for enterprise GenAI deployments: threat modeling for LLM systems, adversarial robustness testing, and secure architecture design for agentic applications
- Developed multi-agent attack demonstrations for memory poisoning, goal misalignment, and cross-context data leakage—directly informing Google Cloud's Model Armor product
- Conducted red team evaluations using DeepTeam (OWASP Top 10, NIST AI RMF) and ART frameworks; tested 40+ vulnerability types including agentic attacks (authority spoofing, goal redirection, context injection)
- Published defensive disclosures: "Method to Isolate Tenancies for LLMs" and "Adaptive LLM DoS Detection and Prevention"

**Security Architect** Sep 2018 – Sep 2019

*Ripple* *San Francisco, CA*

- Led cloud security architecture for blockchain payment infrastructure (AWS/GCP); designed IAM policies, AppSec pipelines, and secure deployment workflows

*Additional 15+ years enterprise security experience (2001–2018) available upon request*

## Teaching & Curriculum Development

**Adjunct Faculty – Trustworthy Machine Learning** Fall 2025

*UCLA Extension*

Designed and taught course covering adversarial robustness, differential privacy, federated learning, and LLM security evaluation

**Teaching Assistant – Applied Generative AI** Fall 2025

*University of Chicago*

Provided technical guidance on multi-agent coordination, A2A protocol, and MCP tool security

## Education

**University of Chicago** – M.S. Applied Data Science 2025

*Relevant coursework:* Generative AI, Multi-Agent Systems, Advanced ML/AI (Transformers), Computer Vision, MLOps

**Savitribai Phule Pune University** – B.E. Electronics 1999

## Technical Skills

| | |
|---|---|
| **Agent Security** | MCP protocol, LangGraph, A2A, Google ADK |
| **Adversarial ML** | Prompt injection, jailbreaking, multi-turn attacks, NVIDIA Garak, PyRIT, ART |
| **Mech. Interp.** | Residual stream probing, activation trajectory analysis, GemmaScope SAE, XGBoost probes |
| **Eval Frameworks** | UK AISI Inspect, Control Arena (specifically BashArena), DeepEval, custom adversarial benchmarks |
| **Privacy/Safety** | Differential Privacy, Federated Learning (Flower), TEEs, Constitutional AI |
| **Infrastructure** | Google Cloud (Vertex AI, GKE), Docker, gVisor, Firecracker, Terraform |
| **Languages** | Python, R, Java |

## Selected Publications

Kulkarni, P., Namer, A. "Temporal Context Awareness: A Defense Framework Against Multi-turn Manipulation Attacks on Large Language Models." *IEEE CAI 2025* (Accepted)

Habler, I., Huang, K., Narajala, V., Kulkarni, P. "Building A Secure Agentic AI Application Leveraging A2A Protocol."

Chowdhary, A., Kulkarni, P. *Google Cloud Professional Security Certification Guide.* Packt.

Defensive disclosures: LLM Tenancy Isolation (TDCommons-6596), LLM DoS Detection (TDCommons-6642)

**Certifications:** Google Cloud Professional Security Engineer — GIAC Cloud Security Automation (GCSA)