The Machine That Hacks Back:

Autonomous API Penetration Testing

Kurtis Shelton

- Principal AI Research Engineer at NetSPI where he founded the AI/ML penetration testing service
- Specializes in a "purple" security approach, combining defensive strategies with adversarial tactics to identify vulnerabilities in AI systems
- Transitioned from biochemistry and bioinformatics to focus on AI/ML security
- Expert in adversarial machine learning, researching AI system vulnerabilities and developing innovative defenses
- Committed to advancing secure AI frameworks through knowledge sharing and collaboration in the evolving field of AI security

The Machine that Hacks Back

- Agentic AI for endpoint reconnaissance & verification
- From unknown endpoints to reproducible lab test case in minutes
- Safe, scoped, auditable

Why Endpoints? Why now?

Ubiquity
- Endpoints = everything users and systems interact with (APIs, routes, IOT services, etc.)

Drift
- Contracts change faster than defenders update specs

Sparse Documentation
- Attacker's advantage

Opportunity
- Automation closes the gap

Scope & Ethics

Lab only: authorized targets, no production exploits

Guardrails:

- Signed scope tokens
- Rate limits
- Audit logs & approvals

Human-in-the-loop escalation

```
policy lab-default {
  allow_hosts = ["127.0.0.1","localhost"]
  max_rps = 3
  max_requests = 500
  verify_only = true
  require_manual_approval = true
  allowed_methods = ["GET","HEAD","POST"]
}
```

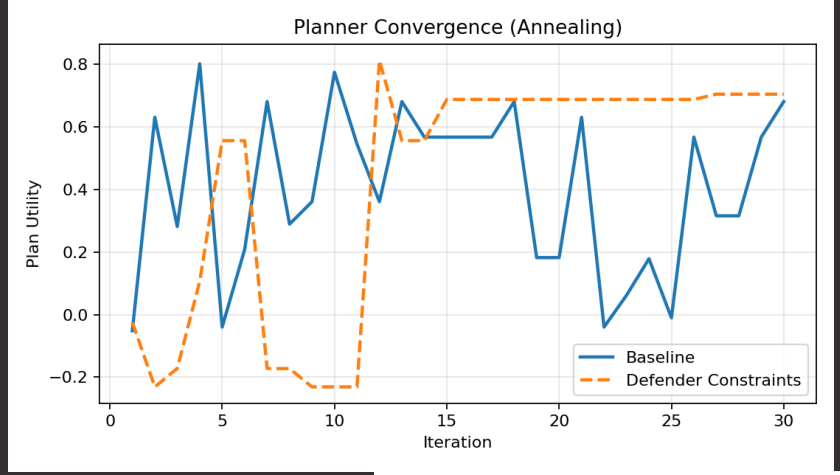
`configs/killswitch.flag -> "1" (abort all)`

High-Level Architecture

- Orchestration: planner/optimizer
- Discovery: spider + fingerprinting
- Contract inference: fuzz-diff + Bayesian inference
- Tooling adapters: MCP sandboxed harnesses
- Verifier: ensemble evaluator + provenance
- Safety: policy DSL, rate limiter, audit trail

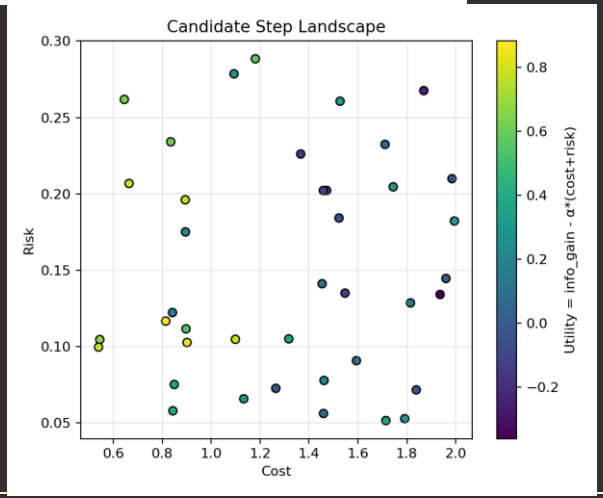
Orchestration “Secret Sauce”

- Meta-Planner (simulated annealing, cost-aware utility)
- Self-Play Robustness: defender injects constraints (rate limits, auth shifts)
- Plans scored by: info gain - cost -risk
- Visual trace of planner convergence



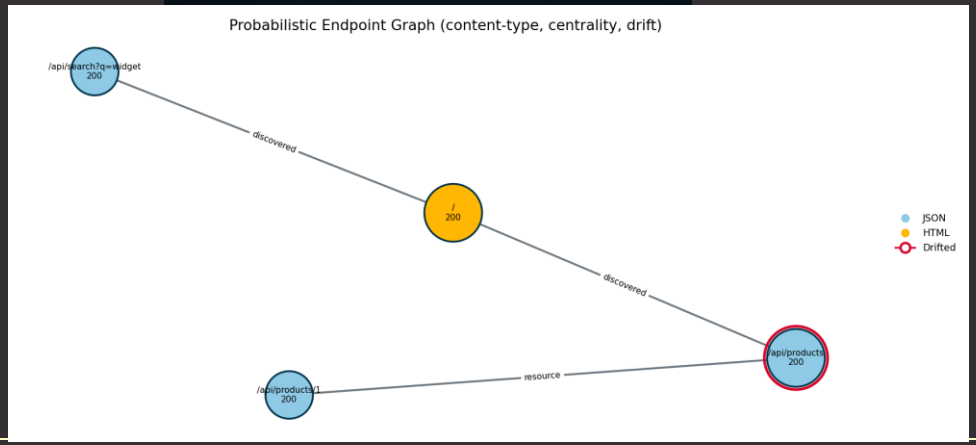
Flowchart:

- A[Input: Base URL + Policy] --> B[Safety: Scope/Rate/Kill]
- B --> C[Discovery: HEAD/GET + Fingerprints]
- C --> D[PEG: Probabilistic Endpoint Graph]
- D --> E[Inference: Schema Hints]
- E --> F[Planner: Annealing, Cost-Aware]
- F --> G[Adapters (MCP): HTTP/Nuclei/Burp]
- G --> H[Verifier: Ensemble + Counterfactual]
- H --> I[Evidence: Redacted + Provenance]
- I --> J[Audit Trail]
- D --> K[Drift Detector]



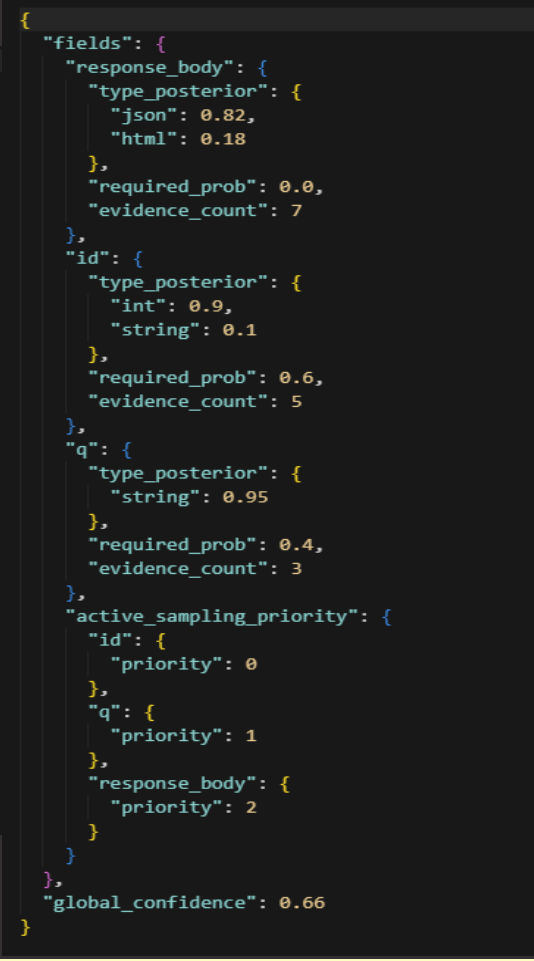
Discovery “Secret Sauce”

- Probabilistic Endpoint Graph (PEG):nodes = endpoints, edges = inferred relations
- Graph algorithms rank “high-value” endpoints
- Multi-modal fingerprints: headers +timing + TLS + content shape
- Drift detection with fingerprint similarity thresholds



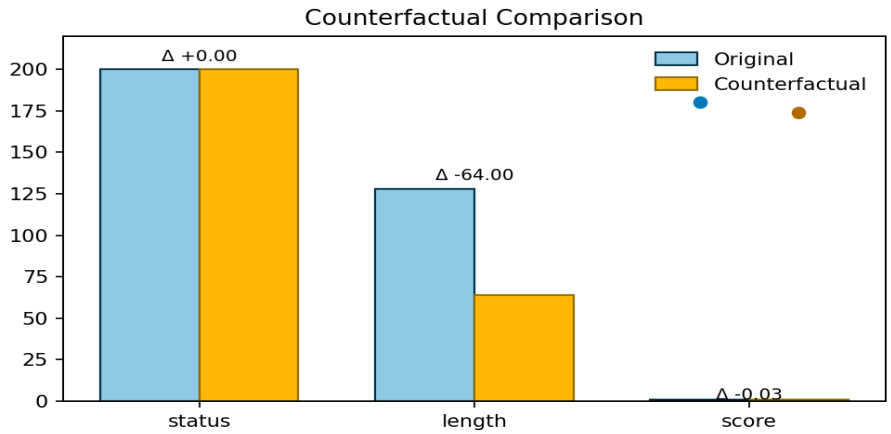
Contract Inference “Secret Sauce”

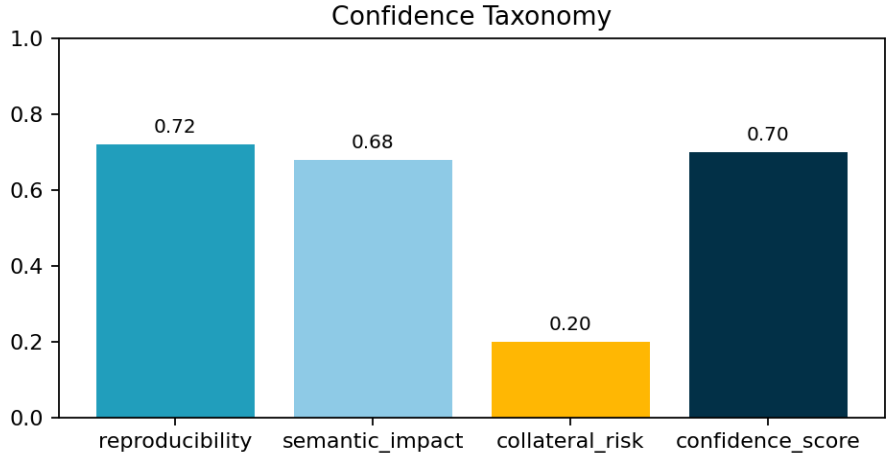
- Bayesian posterior over param types& requiredness
- Active sampling: query high-entropy params first
- Semantic response clustering(embedding space)
- Counterfactual reasoning > map params to behaviors



Tooling Adapters (MCP)

- Formal adapter contracts: typed schemes
- Sandboxed execution with resource caps
- Differential mode: run tool twice > output delta only
- Evidence minimization: redact PII, store structured facts



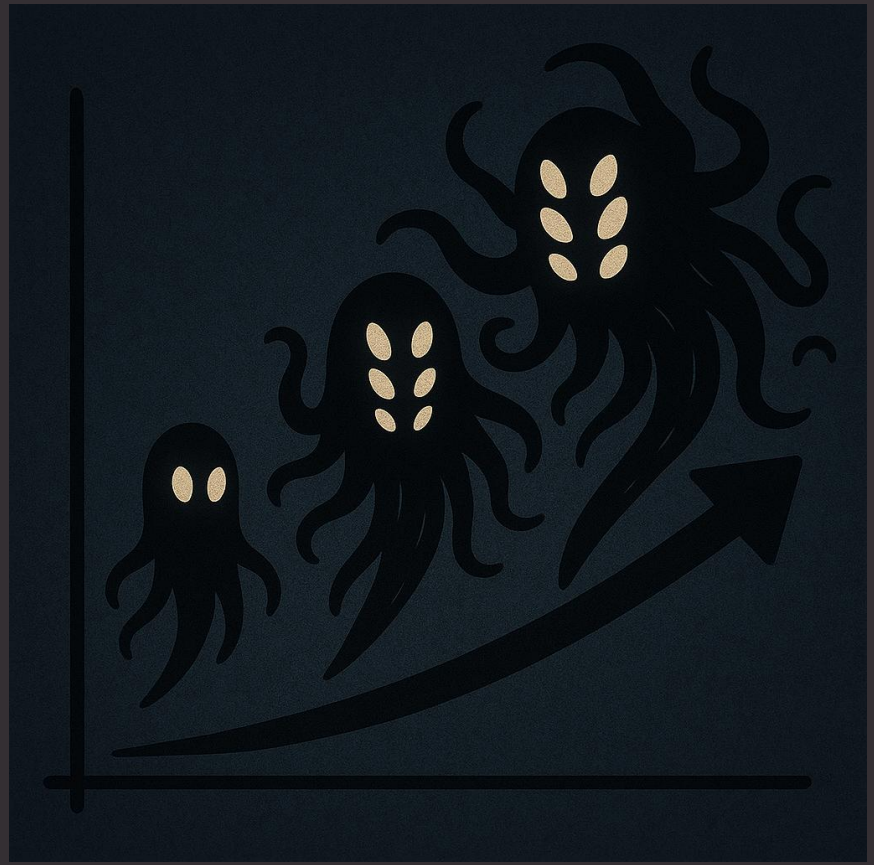


Verifier “Secret Sauce”

- Ensemble evaluators (statistical, rule-based, small SFT)
- Counterfactual validation (reissue benign variants)
- Multi-axis confidence taxonomy: reproducibility, semantic impact, collateral risk
- Provenance chains: cryptographically hashed artifacts

Safety & Governance

- Policy DSL: declarative rules(hosts, rates, approvals
- Capability token prove scope
- Kill switches + runtime monitors
- Red-Team Emulator: inject WAFs, 5xxspikes, latency to test robustness



Cross-Cutting Novelty

- Drift detection & continual learning loop
- Explainable rationales: each plan step documents “why”
- Privacy-aware evidence retention (structural facts + hashes)
- CI/CD integration: test pipeline against lab apps nightly

Takeaways

- Endpoints evolve faster than docs; automation closes the gap
- Novel pipeline: planner optimization, PEG, Bayesian inference, provenance evidence
- Safety first: guardrails, policies, red-team emulator
- Replicate in your lab: skeleton repo+ policy DSL + safe adapters

Q&A

- https://github.com/Spookalicious/Offensive_AI_CON_2025_Framework/tree/main