Prancer is autonomous penetration testing.

It proves what an attacker can actually do.

An operator authorizes a named scope. SwarmHack discovers, exploits, and publishes only evidence-graded findings.

Discover
Map hosts, apps, APIs, identities, and services in scope.

Exploit
Prove the path with live, bounded attacks.

Prove & fix
Evidence-graded findings, then retest the close.

Certify
A badge for that scope, bound to a UUID and hash.

WHO IT IS FOR

Security leadership
Show what an attacker could actually take — not a ranked list of unproven CVEs.

Security architects
One pentest that chains web, API, identity, cloud, AI agents, and the edge.

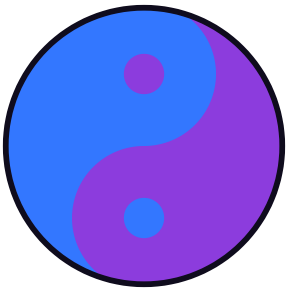
Operators and pentesters
Named scope, time box, kill switch, replayable plan, and a hash to file.

Platform security
The same class of proof in CI and on a schedule, so a fix is re-tested.

TWO NATURES OF ATTACK — ONE PLATFORM

YIN · CORE

Deterministic
Plays by the rulebook. Versioned PAC files, GOAP A* plans that replay. No model required. Air-gap ready.



YANG · EXPLORE

Predictive
Has never seen a rulebook. Open-weight models, opt-in and self-hosted. Cannot publish Critical without captured output.

You can leave the predictive engine off. You cannot turn the evidence gate off.

HOW PRANCER WORKS

Autonomous planning and exploitation inside a deterministic, evidence-driven methodology.

A scanner executes a list. An autonomous pentest pursues an objective.

SwarmHack agents choose tools from live observation, share state, and re-plan when an action fails.

01 Port Discovery

Map reachable hosts, ports, and protocols before payload selection.

02 Recon

Fingerprint services, stacks, identities, and reachable control points.

03 Discovery

Generate and rank candidate attack paths by reachability and context.

04 Exploitation

Attempt safe, bounded exploitation and capture target-side proof.

05 Post-exploit

Reuse credentials, escalate privileges, pivot, and move laterally.

06 Collection

Grade evidence, deduplicate findings, map paths, and build reports.

Evidence doctrine

- **Exploited** — Captured target output. Only this grade can be Critical.
- **Observed** — Condition confirmed; exploit not proven.
- **Attack Path Identified** — Reachable chain proven; exploit not executed.
- **Simulated** — Modeled only; never presented as exploit proof.

Authorization & safety

- Signed envelope with named targets and expiry ≤24h.
- Kill switch and fail-closed scope enforcement.
- Default-deny destructive classes; reversible probes by default.

DETERMINISTIC EXECUTION
Repeatable plans, audit replay, bounded primitives.

+

ADAPTIVE REASONING
Open-weight models, opt-in. Observe, re-plan, pivot.

WHAT PRANCER ATTACKS: WEB APPLICATIONS

Show the attacks the platform can execute, not vague coverage claims.

ATTACK TYPE	WHAT IT MEANS	
01 **Injection**	SQL, NoSQL, command injection	PARTIAL
02 **Remote Code Execution**	OS command execution and web application RCE	✓
03 **Server-Side Template Injection**	SSTI payload injection and execution	✓
04 **Cross-Site Scripting**	Reflected, stored, and DOM XSS	✓
05 **Server-Side Request Forgery**	SSRF and metadata access abuse	✓
06 **XML External Entity**	XXE and XXE-to-SSRF paths	✓
07 **Insecure Deserialization**	Java, PHP, Python, and .NET gadget-chain probing	✓
08 **File Upload Attacks**	Malicious upload and web-shell style abuse	✓
09 **Path Traversal**	LFI, directory traversal, and unauthorized file access	✓
10 **HTTP Request Smuggling**	CL.TE and TE.CL smuggling over raw TCP	✓
11 **Open Redirect**	Redirect abuse and token theft paths	✓
12 **CORS Misconfiguration**	Origin reflection and credential leakage	✓
13 **CSRF**	Authenticated workflow abuse	✓
14 **Security Misconfiguration**	Dangerous methods, insecure headers, exposed admin paths	✓
15 **Vulnerable Components**	Known-CVE exploitation against exposed software	✓

WHY THIS MATTERS

- A scanner may report a weakness.
- Prancer shows whether it can actually be exploited.
- Evidence and attack paths help teams fix what matters first.

Mapped to OWASP Top 10 · CWE · CVE/CVSS

WHAT PRANCER ATTACKS: APIS

Prancer tests API behavior, authorization, business logic, and exploit paths.

API ATTACK TYPE	WHAT IT MEANS	
Broken Object-Level Authorization (BOLA / IDOR)	Access to objects or records that should be off limits	✓
Broken Authentication	Weak login, session, or token handling	✓
Broken Object Property Authorization	Unauthorized property exposure or modification	✓
Resource Consumption Abuse	API calls that can exhaust resources or create denial of service	PARTIAL
Broken Function-Level Authorization	Access to privileged or admin functions	✓
Sensitive Business Flow Abuse	Multi-step workflow abuse and business-logic attacks	✓
API Server-Side Request Forgery	API-driven SSRF and backend request abuse	✓
Security Misconfiguration	Insecure API settings, headers, or deployment choices	✓
Shadow or Outdated APIs	Hidden, forgotten, or old endpoints	✓
Unsafe API Consumption	Trust issues with downstream APIs and third-party services	PARTIAL
JWT Attacks	alg:none, algorithm confusion, and claim tampering	✓
Mass Assignment	Unexpected or unauthorized property updates	✓
GraphQL Attacks	Schema support and testing	✓
REST API Fuzzing	Malformed and context-aware payload testing	✓
Multi-Step Authenticated Testing	Role-aware testing across real user flows	✓

Recognizable standards

- OWASP API Security Top 10 2023
- OWASP Top 10
- CWE / CVE / CVSS

What buyers should know

- Checks real exploit paths, not only schema mistakes.
- Supports REST, GraphQL, OpenAPI, Postman, and authenticated workflows.
- Shows which API issues can lead to real impact.

WHAT PRANCER ATTACKS: IDENTITY, NETWORK, CLOUD

Start anonymous, produce credentials, re-enumerate with access, then exploit and move laterally.

IDENTITY & ACTIVE DIRECTORY		
Enumeration	Anonymous LDAP/SMB enumeration of users, SPNs, trusts, and policy	✓
Password Spraying	Lockout-aware Kerbrute and LDAP spray; never brute force	✓
Kerberos Attacks	AS-REP roasting and Kerberoasting	✓
Credential Dumping	SAM/LSA secret extraction, LAPS and gMSA password reads	✓
Pass-the-Hash / NTLM Relay	Native DCERPC lateral movement; PrinterBug and PetitPotam coercion	✓
DCSync	Directory replication abuse for the krbtgt hash	✓
ADCS / Delegation / ACL Abuse	ADCS ESC1–ESC15, delegation incl. RBCD, shadow credentials, GPO	✓
Ticket Forgery / SCCM / Entra ID Hybrid	Golden/Silver/Diamond/Sapphire tickets; SCCM takeover; PRT replay	✓

NETWORK & SERVICES		
Host and Port Discovery	Reachable hosts, open ports, and exposed services	✓
Service Fingerprinting	Identify versions, stacks, and likely attack paths	✓
SMB Attacks	Null sessions, signing weaknesses, and file-share abuse	✓
SSH / FTP / WinRM	Bounded default-credential testing; SSH and WinRM lateral execution	✓
MSSQL and Service Access	Database access and command execution paths	✓
TLS and Certificate Analysis	Protocol, cipher, and certificate weaknesses	✓
Pivoting and Segmentation Validation	Can an attacker cross trust boundaries?	✓

CLOUD, CONTAINERS & CI/CD		
Cloud Metadata Abuse	IMDSv2 credential harvest and metadata access	✓
IAM Privilege Escalation	Role and permission abuse	✓
IAM Trust-Chain Attacks	Assume-role and trust-chain pivoting	✓
Secrets Access	Secrets Manager reads, strictly read-then-report	✓
Sensitive Cloud Data Access	Storage and database exposure validation	✓
Container Escape	Docker Engine escape proven by a marker container, then removed	✓
Kubernetes Exposure	Anonymous kubelet (10250) audit and privileged-pod exposure	✓
CI/CD Compromise	Jenkins script-console RCE, GitHub Actions runners, pipeline secrets	✓

WHY THIS PAGE IS IMPORTANT

These are the attack paths that turn one foothold into broader compromise.

BEYOND TRADITIONAL PENTESTING

Prancer validates security controls, attacker movement, and connected systems.

SECURITY CONTROL VALIDATION		
ZTNA / Private Application Access	Can protected internal apps still be reached?	✓
SWG and URL Filtering	Can web controls be bypassed?	✓
CASB Restrictions	Are SaaS and tenant controls enforced?	✓
DLP Controls	Can sensitive data still leave the environment?	✓
Firewall / FWaaS	Do network and application controls block the path?	✓
DNS Security and C2 Controls	Can attacker callbacks still work?	✓
MFA and Identity Controls	Do step-up controls stop the attacker?	✓
GenAI Egress Controls	Can LLM or GenAI channels leak sensitive data?	✓
Network Segmentation	Do trust boundaries really hold?	✓

POST-EXPLOIT & ATTACK CHAIN		
Privilege Escalation	Can low access become admin or root access?	✓
Credential Reuse	Can found credentials unlock more systems?	✓
Lateral Movement	Can compromise spread across hosts and identities?	✓
Persistence Paths	Can the attacker maintain access?	✓
Sensitive Data Collection	Can important data be reached and read?	✓
Crown-Jewel Targeting	Can the attacker reach critical systems or assets?	✓
Attack-Path Chaining	Can separate weaknesses become one real path?	✓
Safe Proof of Impact	Can impact be shown without destructive action?	✓

ROBOSEC, IOT & CYBER-PHYSICAL		
MQTT Broker Attacks	Authentication bypass and topic abuse	✓
ROS2 / DDS Attacks	Service discovery and unauthorized commands	✓
Unsigned OTA Updates	Firmware promotion, OTA update, and downgrade-path risk	✓
Firmware Analysis	Embedded secrets and hardcoded keys	✓
Fleet API IDOR	Cross-device or cross-tenant control abuse	✓
BLE / RF / Zigbee	Wireless replay and unencrypted control paths	✓
UART / JTAG / USB	Local hardware and debug-interface exposure	✓
Robot Admin Interfaces	Default credentials and privilege escalation	✓
Device Integrity / Attestation	Can identity and integrity be trusted?	✓

Not default SKU — engagement / SOW. OT: IT/OT boundary and path validation; broad ICS protocol coverage not claimed.

RECON → EXPLOIT → ESCALATE → PIVOT → COLLECT → PROVE

WHAT PRANCER ATTACKS: AI AGENTS & NON-HUMAN IDENTITIES

Discover agents and machine identities read-only, then prove what they can reach with SwarmHack.

AGENTLESS NHI & AI AGENT DISCOVERY		
Agentless, Read-Only Collection	Reviewed allowlist of read-only API calls; nothing to install	✓
AI Agent & Shadow-AI Inventory	Agents, MCP tool servers, model endpoints, and their identities	✓
AI Egress Visibility	Firewall-seen traffic to AI providers and MCP servers, tied to a host	✓
Identity Resolution	Same identity across systems merged into one entity, with evidence	✓
Effective Permissions	Deny rules, boundaries, org policies, PIM, and federation evaluated	✓
Blast Radius & Ownership	Reach through chains of trust; owner inferred with confidence	✓
Deterministic Risk Scoring	Toxic-combination rules, mapped to OWASP NHI and Agentic-AI	✓
Choke Points, Minimum Cuts & What-If	Smallest permission change that removes the most risk	✓

DISCOVERY CONNECTORS		
AWS	IAM, Organizations, Identity Center, secrets, Bedrock, and AgentCore	✓
Microsoft Azure	RBAC and PIM, managed identities, Key Vault, AKS, AI Foundry agents	✓
Google Cloud	IAM, deny policies, service accounts, workload federation, Vertex AI	✓
FortiGate	Admin and API tokens, VPN and auth secrets, hardening, AI traffic	✓
Entra ID / Microsoft 365	Applications, service principals, OAuth apps, and tokens	✓
Okta / GitHub	Applications, tokens, OAuth apps, repositories, and workflows	✓
OpenAI / Anthropic Admin APIs	Projects, API keys, and service accounts	✓

AI AGENT ATTACKS (SWARMHACK)		
AI Gateway & Infrastructure	Unauthenticated model access, credential leakage, budget exhaustion	✓
MCP Protocol Attacks	Tool poisoning, confused deputy, cross-audience tokens, manifest drift	✓
Agentic Prompt Injection	Carrier-channel mapping and filter evasion, proven by tool execution	✓
Tool-Call Hijacking	Agent tools driven with controlled arguments; depth-2 chains	✓
AI Runtime & Sandbox	Interpreter execution proofs and sandbox-separation testing	✓
Memory Poisoning	Planted directives that survive restart and drive a later action	✓
Multi-Agent (A2A) Attacks	Cascade across agent handoffs, role swap, cross-tenant memory leaks	✓
Canary-Oracle Grading	Exploited only when the oracle fires; echo is never graded as execution	✓

Google Cloud connector replay-tested; FortiAuthenticator connector in development. Secrets recorded as presence only, never values.

DISCOVERY FEEDS ATTACK VALIDATION

Discovery findings hand off to SwarmHack, which proves what an agent or identity can actually reach.

AI PENTESTING: AGENTS, MCP, A2A, LLMS & RAG

● GENERALLY AVAILABLE

Pentest AI agents, MCP servers, A2A systems, LLMs, and RAG with the same proof-first doctrine.

WHAT AI PENTESTING TESTS		
AI Agents	Tool misuse, excessive agency, and goal hijacking across multi-step workflows	✓
Agent Identities	Privilege escalation through agent identities	✓
MCP Servers	Tool poisoning, unauthorized tool invocation, and over-scoped tools	✓
MCP Credentials & Secrets	Credential and secret exposure through MCP servers	✓
A2A (Agent-to-Agent)	Agent trust abuse, message injection and impersonation, delegation-chain abuse	✓
LLM Prompt Attacks	Direct and indirect prompt injection, and jailbreaking	✓
LLM Safety & Leakage	Safety and harmful-content bypass, system-prompt leakage, and data leakage	✓
RAG Systems	Retrieval poisoning, tenant boundary abuse, and knowledge-base data extraction	✓

REAL ATTACK CHAINS & AI SECURITY FRAMEWORKS		
Content-Layer Chain	One email to a self-propagating agent worm (EchoLeak CVE-2025-32711, Morris II)	✓
Protocol-Layer Chain	Poisoned MCP tool to client code execution (CVE-2025-6514, CVE-2025-9611)	✓
Infrastructure-Layer Chain	One request to vector-database RCE, embedding inversion, and index poisoning	✓
Step-Level Grading	Each chain step graded Exploited, Proven, or Control-blocked	✓
OWASP Top 10 for LLM Apps	Prompt injection, sensitive information disclosure, excessive agency, and more	✓
OWASP Top 10 for Agentic Apps	Agent goal hijacking, tool misuse, identity and privilege abuse, memory poisoning	✓
NIST AI RMF / ISO/IEC 42001 / EU AI Act	Adversarial-testing evidence to support risk, AI management, and robustness needs	✓

PROOF-FIRST BY CONSTRUCTION		
Reproducible Evidence	Prompts, tool calls, and responses that reproduce each finding, so you can retest	✓
Framework-Mapped Findings	Every finding mapped to the relevant framework category, with evidence	✓
Runs on Your Infrastructure	All exploitation runs inside your environment; findings delivered to you	✓
Model Choice	Deterministic-only, or frontier, private, open-weight, sovereign models you approve	✓
Authorization Envelope	Signed: named targets, 24-hour maximum expiry, tester identity, contract reference	✓
Kill Switch	Process-global; halts all exploitation the instant an operation steps out of scope	✓
Read-Then-Report	Bounded, side-effect-free proofs: marker round-trips, not destructive payloads	✓
Evidence-Gated Reporting	Models may expand what is tried; captured target evidence decides what is proven	✓

Exploited means a real, reproduced attack. For pre-production AI behavioral certification and assurance, ask us about Gadriel.

GENERALLY AVAILABLE

SwarmHack AI Pentesting is testing customer AI systems today. Pairs with AI agent discovery on page 7.

PROOF, REPORTING, DEPLOYMENT, AND USE CASES

Prancer is designed to prove exploitability and help teams act on it.

EXPLOITED
Captured target output proves the exploit worked.

OBSERVED
The condition is real, but exploitation was not completed.

ATTACK PATH IDENTIFIED
The reachable chain is proven, even if the exploit was gated.

SIMULATED
Modeled only; not presented as exploit proof.

Outputs & reporting

- OCSF 1.1.0 JSON with MITRE ATT&CK mapping
- Interactive HTML attack graphs
- Markdown reports
- DOT / Cytoscape / Neo4j graph exports
- Technical findings with evidence and CVSS v3.1
- Executive summaries
- Compliance rollups
- Retest and validation output

Deployment & operations

- SaaS, private cloud, on-premises, and air-gap
- On-demand and scheduled runs
- CI/CD integration
- Workflow and ticketing integration
- Signed authorization and scoped runs
- Global kill switch
- Default non-destructive probes
- Explicit approval for destructive classes

BEST-FIT USE CASES

- Web and API
- Internal network and AD
- Cloud and container
- Security-control effectiveness
- Robotics and IoT
- Continuous autonomous pentesting

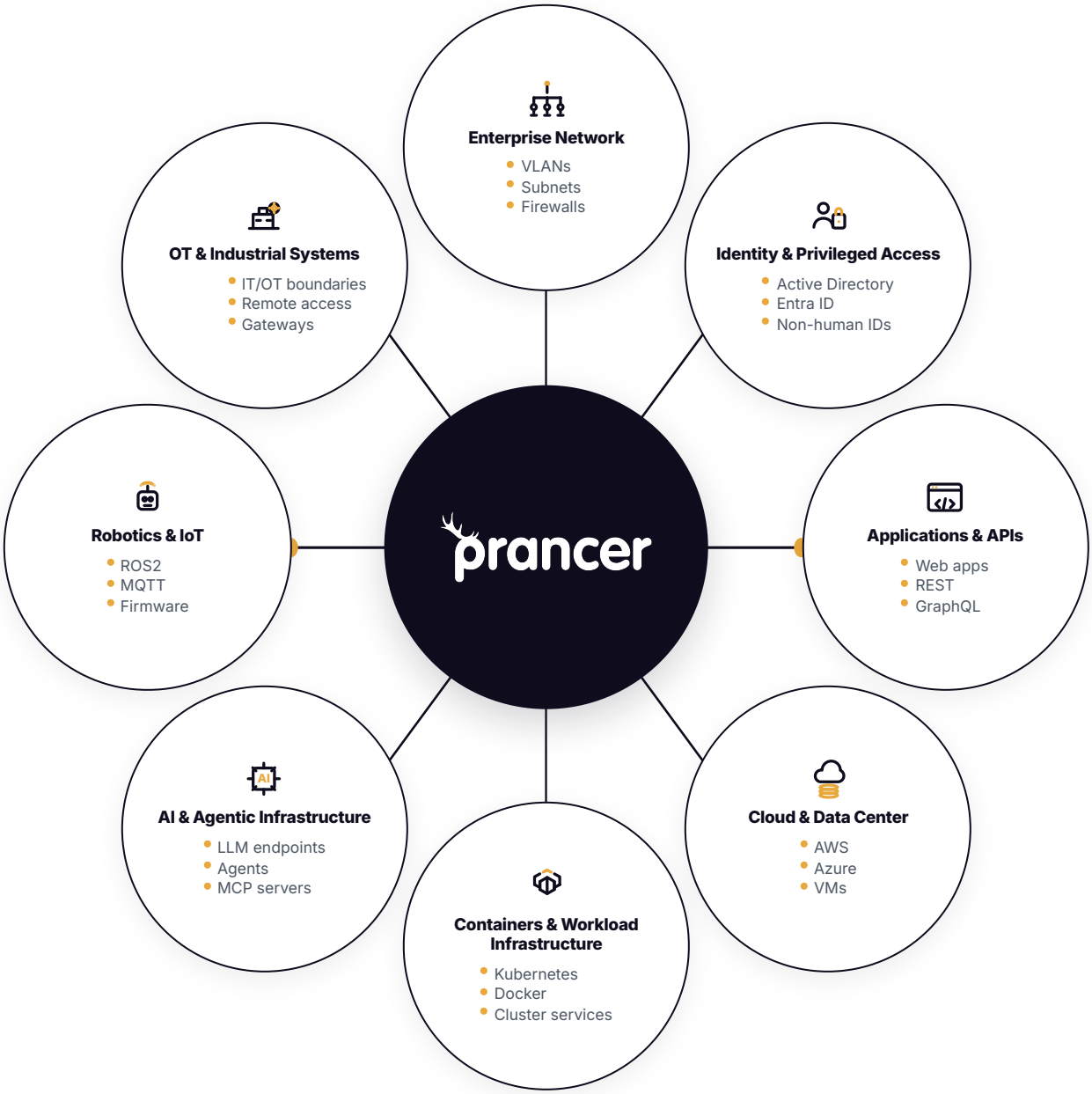
Compliance rollups **PCI DSS 4.0 · NIST CSF 2.0 · OWASP Top 10 · SOC 2 · HIPAA · ISO 27001:2022 · DORA · NIS2** — finding mappings, not vendor certificates

SEE WHAT AN ATTACKER CAN ACTUALLY EXPLOIT.

Learn more at **prancer.io**

WHAT PRANCER ATTACKS

One platform across the estate. Network to industrial.



NETWORK → IDENTITY → APPLICATIONS → CLOUD → WORKLOADS → AI → ROBOTICS → INDUSTRIAL