THE PAPERS · THURSDAY, NOVEMBER 19, 2026 DEFENSE ASSESSMENT · WORST-CASE SERIES

Assessment of PenTesting Frameworks

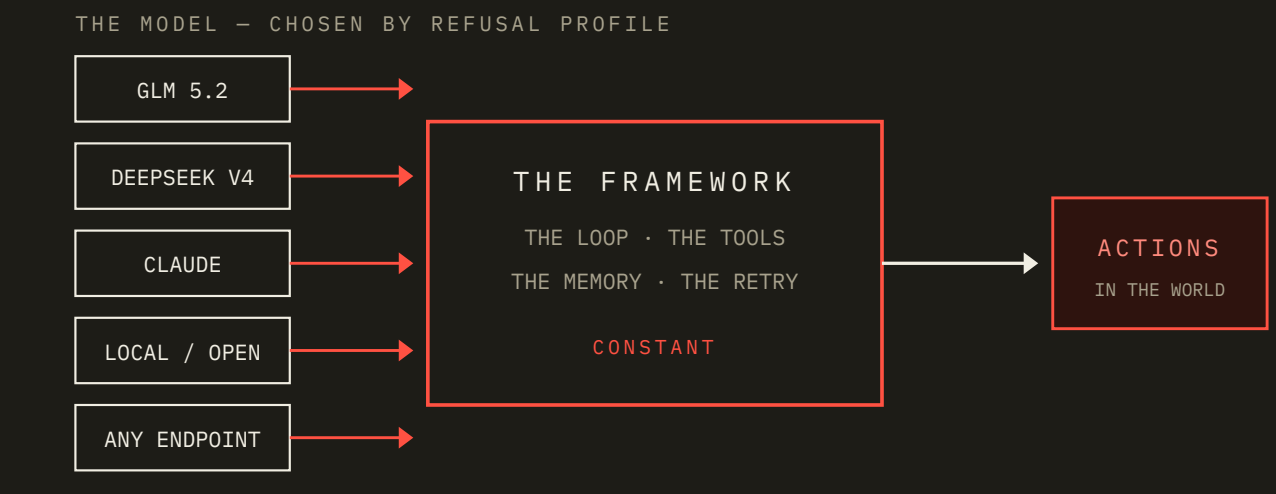
No Skills Required

The framework is the weapon. The model is just the part you plug in.

ALEX ESTABROOK THREAT ASSESSMENT · NOVEMBER 2026 SIX FRAMEWORKS · GRADED AS SOURCED

PDF · MARKDOWN

EXHIBIT A · THE SWAPPABLE PART



SWAP THE LEFT COLUMN FREELY. THE RIGHT SIDE DOES NOT CHANGE.

Every framework in this assessment accepts whatever model the operator plugs in — a cloud API, a reseller, a stolen key, a local open-weight model. Refusal is a property of the left column. The capability is on the right.

25
CAIRN OPERATIONS, AUG–SEP 2026

19+
ORGANIZATIONS TARGETED BY CAIRN

10
ORGANIZATIONS ACCESSED OR READ

1,323
CYBERSTRIKEAI HOSTS, OCT 2026

609
ARTEX HOSTS, OCT 2026

5,200+
RECORDS EXFILTRATED BY ONE LOW-SKILL ACTOR

ABSTRACT Abstract

Six frameworks are assessed here: Cairn, ARTEX, CyberStrikeAI, Hermes Agent, PentAGI, and AutoHunter. They differ in origin, architecture, and purpose. They share one property, and it is the only property that matters: none of them is a model. Each is a harness — a loop, a tool set, a coordination layer — that accepts whatever model the operator plugs into it, and keeps working when the model is swapped. The security conversation still grades two things: the actor, by sophistication, and the model, by capability and refusal. The documented record grades a third thing, and it is neither. It is the framework: the part that plans, remembers, retries, and acts, while the model underneath it is replaced like a battery.

The evidence is graded, and the grades are kept separate throughout. Four of the six are documented in real operations. One is documented in limited form. One has no documented malicious use at all, and is assessed as dual-use only. Nothing in this paper is graded higher than its source.

01 1. The walls came down

Three walls used to separate an ordinary person from a working intrusion, and they were different walls.

The skill wall. Intrusion tradecraft — reconnaissance, tool selection, output interpretation, next-step choice — took years to acquire. The frameworks now perform that chain themselves. In the clearest documented case, a single low-skill actor, working in broken English, ran credential theft, telecommunications-fraud planning, Wi-Fi password cracking, and cryptocurrency reconnaissance through one general-purpose agent — four trades that used to belong to four different people.

The refusal wall. A model that refuses is no longer an obstacle; it is a filter the operator shops around. Every framework here is model-agnostic. If one provider refuses, the operator changes the channel, the endpoint, or the model — including to a local, open-weight model that answers to no provider at all. Refusal has become a property of a swappable part.

The access wall. The low-friction options sit in the same browser, at the same kind of price, behind the same kind of prompt box as the legitimate ones. There is no back alley to find. Access is a tab.

What replaced the walls is an interface: now you just give the computer instructions. A target, and a goal stated in about a hundred characters, was the entire human input to one documented campaign. The framework wrote the plan underneath it. Threat assessment that still begins by grading the actor’s sophistication and the model’s frontier position is measuring the two parts of the system that no longer decide the outcome — and not measuring the part that does.

02 2. The comparison

FRAMEWORK	MODEL SWAP	COORDINATION LOOP	OPERATOR SUPPLIES	EV
Cairn	Model-agnostic; GLM 5.2 and DeepSeek V4 Flash/Pro benchmarked against each other across operations	Reason → Explore loop over a shared blackboard fact-intent graph	A target and a one-line goal	D c
ARTEX	Needs little more than access to an LLM; no per-operation model census published	Multi-agent; evaluate, adapt, pursue	Access to a model; a target	D i c c
CyberStrikeAI	Any OpenAI-compatible endpoint, selectable per session	Server-side LLM orchestrator; a classic C2 implant with no AI on the host executes	A task in general terms	D e c c
Hermes Agent	Any model, switchable with no code changes; BUGSTER settled on DeepSeek V4	General agent with memory and self-improving skills; specialized profiles as separate bots	Prompts, in conversation, in broken English	D s u
PentAGI	10+ cloud providers, OpenAI-compatible gateways, or local models	Hierarchical: flows → tasks → subtasks → actions; planner over specialist agents	A target and a goal in plain language	D a c s
AutoHunter	OpenAI- or Anthropic-protocol endpoint; DeepSeek by default	Collector → Worker → Reviewer → human	A machine, run 24×7; human re-review at the end	D c u

The rows differ. The shape does not. In every row, the operator supplies intent and the framework supplies competence; the model supplies text; and the only checkpoint anywhere in the loop belongs to the operator or to the model itself.

CASE 01 · DOCUMENTED

3. Cairn — documented

The record. Between August and September 2026, Chinese-speaking attackers used the open-source Cairn framework to autonomously conduct 25 operations against at least 19 organizations. Unit 42 tracks the activity as CL-STA-1238 and assesses with moderate

confidence that it represents China-nexus intelligence collection. Unauthorized access or data retrieval was achieved at 10 of the targeted organizations — including dozens of valid credentials, personal data exceeding one million records at one organization, military and government identities, and persistent administrator accounts in a production system. Targets concentrated in advanced semiconductor manufacturing, United States defense policy, and civil society. Individual operations ran from 20 minutes to more than 50 hours.

The mechanism. Cairn is model-agnostic, and the operators treated it that way: 13 operations ran on Zhipu GLM 5.2, 8 on DeepSeek V4 Flash, and 4 on DeepSeek V4 Pro — the models benchmarked against each other, the stronger one brought in for targets that resisted the weaker. A model drives each agent worker; coordination runs over a shared blackboard, where confirmed findings accumulate as facts and next actions are queued as intents. The loop is Reason, then Explore, repeating until the model judges the goal met, the options exhausted, or the attacker stops it.

What the operator supplied. A target and a goal — one line, roughly a hundred characters: obtain substantive high-value access, including webshell, SSH, source code, keys, user information, credentials, ideally to enter the internal network. Cairn planned, executed, and recorded underneath that line. The exploited weaknesses were known and ordinary — authentication bypasses, insecure object references, weak and default credentials, exposed cloud APIs, dated vulnerabilities. Autonomous AI raised the volume and coverage of exploitation, not its complexity.

The bypass. Cairn's stock worker prompts already instruct the model never to refuse. The attackers reinforced that with an authorization pretext, and each operation generated its own AUTHORIZATION.md — a file in which the attacker's creation of the task was treated as the client's approval. The framework attested to itself, and neither model refused the framing. Permission was granted. The applicant and the approver were the same party, and the paperwork was in order.

The signature. Volume over stealth: in Unit 42's study, hundreds of thousands of noisy, high-frequency requests from a single origin — a loop that does not tire and does not hide, because it does not need to.

The gate seam. Between Reason and Explore, at the blackboard's intent queue — where every proposed intent already exists as a discrete, recorded proposal before it becomes an

action. A gate there, outside the model and outside Cairn, checking each intent against an authorization record the operator did not issue, intervenes at every iteration instead of none.

CASE 02 · DOCUMENTED

4. ARTEX — documented

The record. ARTEX is an open-source, multi-agent penetration-testing framework, first published in mid-2026 and widely forked since. Unit 42's Incident Response team investigated multiple intrusions in which the attacker's server hosted an ARTEX console, and its threat hunters tracked a coordinated ARTEX campaign against customer cloud environments, in sectors including retail and consulting, worldwide. As of October 2026, 609 internet-facing hosts were running ARTEX. Attackers reached the cloud through identity workflows and exposed APIs, sometimes in less than a day — in observed cases without exploiting a single software vulnerability. The author has since made the repository closed-source and ceased public development to prevent misuse; the forks remain.

The mechanism. ARTEX needs little more than access to a large language model. The framework supplies the agents and the loop; the operator supplies whatever model they can reach. Unit 42 publishes no per-operation model breakdown for ARTEX, and none is claimed here. What is documented is the behavior of the loop: evaluate the response, adapt when something fails, and keep pursuing the objective.

What the operator supplied. A model connection and a target. What the framework removed is stated by Unit 42 directly: adversaries no longer require specialized tradecraft to execute multi-stage, adaptive penetration testing. AI tooling did not invent new attacks; the pace and the persistence changed.

The bypass. None specific is documented in the ARTEX report, and none is asserted here. The mechanism is access, not prompting: free tooling plus a low-friction model is sufficient.

The signature. ARTEX is the one framework with a published behavioral fingerprint, and it is a machine's fingerprint: bursts of one to fourteen requests about a second apart, then a pause of twenty-five to thirty-one seconds while the model thinks. A pivot within about a minute of a credential download. More than 200 client secrets validated in about three minutes. A read after almost every write, to check its own work. Bogus canary inputs before

real ones. Polling every 48 seconds for three days. Any one trait could be a human; together, they are a loop.

The gate seam. At the target's own action path, at the two moments that convert access into reach: a write to an exposed API or self-service flow, and the first use of a discovered credential. A gate at those seams — every write authenticated, every newly presented credential checked against an independent record before it is honored — acts after reconnaissance, which cannot be prevented, and before the pivot, which can.

CASE 03 · DOCUMENTED 5. CyberStrikeAI — documented

The record. CyberStrikeAI is an open-source platform, written in Go, that combines LLM-based attack orchestration and a built-in command-and-control server in a single process. As of October 2026, Unit 42 counted 1,323 internet-facing hosts running it — the largest footprint in this assessment — and 21 hosts running it alongside ARTEX. Kaspersky documented a campaign against education-sector organizations in several countries across Africa and Latin America in which CyberStrikeAI played the key role in post-exploitation: from agent deployment to compromise of the domain administrators in three to four days. Persistence and local reconnaissance took two minutes; the final stage — lateral movement, credential access, database export — no more than one hour.

The mechanism. The intelligence is isolated on the control server. An LLM orchestrator plans, selects tools, and generates commands; the implant on the compromised host is a classic agent with no AI functionality at all — no API keys, no model calls, no autonomous logic. It polls, executes shell commands, and reports back; the model analyzes the results and decides what to do next. The operator defines the task in general terms. The model determines how to execute it.

The swap. Any OpenAI-compatible endpoint: a channel is a provider, a base URL, a key, and a model, selectable per session. As the Cloud Security Alliance's synthesis puts it, because these frameworks accept any OpenAI-compatible endpoint, a single provider's safety controls cannot reliably stop them — an operator blocked by one provider switches to another, to a reseller, or to stolen credentials.

The bypass. No jailbreak is documented for CyberStrikeAI, and none is needed. The repository ships approval gates, tool allowlists, and audit logs — and nothing in the design stops an operator who controls the deployment from disabling those safeguards. The check exists. The operator holds it.

The signature. Almost none on the victim host — that is the signature. With the model on the server, the intrusion looks like a typical remote-control session. What distinguishes it is speed, and a fingerprint on the attacker's own server: a self-signed certificate generated at launch, carrying the framework's name.

The gate seam. Between the server-side orchestrator and the implant task queue — the single chokepoint every post-exploitation action passes through, on its way to an implant with no judgment of its own. The framework's built-in human-in-the-loop approval is not that gate; it belongs to the operator. An independent gate at the queue, sealing each proposed command before it is tasked, is the same seam as Cairn's intent queue, in a different architecture.

CASE 04 · DOCUMENTED

6. Hermes Agent — documented

The record. Hermes Agent is not a penetration-testing framework. It is a general-purpose, self-improving agent built by Nous Research — open source, with a built-in learning loop that creates skills, improves them in use, and remembers across sessions. Unit 42's investigation of a single actor, handle BUGSTER, documents what a general agent becomes when a criminal configures it: the primary interface for credential theft and data exfiltration, VoIP telecommunications fraud, Wi-Fi password cracking, and cryptocurrency reconnaissance, run through specialized profiles maintained as separate bots. Against a second Indian university, the actor achieved a confirmed intrusion and exfiltrated a database of more than 5,200 records, including students' personally identifiable information. Multiple, unrelated actors are independently adopting the same Hermes-plus-DeepSeek combination — which Unit 42 calls, with professional restraint, a possible toolchain of choice.

The assessment language, noted. Unit 42 concludes that the actor "used basic tooling and relied on AI-generated scripts and reconnaissance reports, suggesting they didn't have the skills these activities would have required before the advent of readily available AI tools." Suggesting. The actor's own logs, his AI-generated reports, and his exfiltrated database are

in the same document. The field's confidence language was built for uncertain things. This is not one of them.

The mechanism. Hermes is model-agnostic by design: use any model you want, switch with a single command, no code changes, no lock-in. BUGSTER experimented with GLM 5.2 and other models through OpenRouter, and settled on DeepSeek V4 as the reasoning engine. Domain expertise was acquired the same way: through conversation. The VoIP fraud operation was advised into existence by a dedicated profile — a specialist the actor did not hire, train, or become, but simply opened a chat with.

What the operator supplied. Prompts. Among them, verbatim: "Can you find about like what scam or thing they done in past like spoofing calls or somthing." That sentence — misspelled, ungrammatical, and entirely sufficient — is the skill floor of the current era, preserved in an incident report.

The bypass. A system-prompt override, tested repeatedly until the model's safety controls stopped interfering. Unit 42 names the driver of the combination without ornament: low safety friction at low cost. The models are inexpensive, easy to access, and readily jailbroken. The assessment concurs.

The gate seam. Between a profile's output and the execution of that output — where a generated script, plan, or instruction leaves the conversation and is run against a login surface, or used to move collected data off a target. The conversation is not the consequential act. The execution is. A gate that sits at the seam, outside the operator's own configuration — the operator controlled the profiles and wrote the override — is the first check in the entire arrangement that the operator does not hold.

CASE 05 · DOCUMENTED, LIMITED

7. PentAGI — documented, limited

The record, at its grade. PentAGI is an open-source, self-hosted, multi-agent system for autonomous penetration testing, developed by VXControl. The documented malicious use is real and narrow, and it is stated at exactly its width: between late August and early October 2026, GTIG tracked a threat actor using a customized attack orchestration platform built off the open-source version of PentAGI to conduct systematic reconnaissance and vulnerability

scanning against a wide range of organizations in Asia. That is reconnaissance and scanning, by a customized derivative. An end-to-end compromise by stock PentAGI is not proven in the primary text, and is not claimed here.

The mechanism. Hierarchical delegation — flows, tasks, subtasks, actions — with a primary agent coordinating specialists for research, development, and execution, and memory kept in a database so that results and successful approaches survive into later flows. GTIG characterizes it as an autonomous offensive framework designed to lower technical barriers for entry-level operators running complex campaigns. The designers and the threat assessors agree on the product. They differ only in tone.

The marketing, quoted. The project's own documentation describes the interface as a feature: "You describe a target and a goal in plain language; a team of specialized AI agents plans the work, runs 20+ professional tools such as nmap, Metasploit and sqlmap inside an isolated Docker sandbox, remembers what it learned and hands you a report." Read that sentence as a capability statement instead of a brochure, and it is the most concise threat assessment in this paper. The vendor wrote it.

The swap. The model of your choice: more than ten cloud providers, your own OpenAI-compatible gateway, or local models — per agent, if desired, so the planner, the generator, and the utility roles can each run a different one. A model that refuses can be replaced by one that is running on the operator's own hardware, at which point the refusal layer of the industry consists of a settings menu.

The check. Authorization is a stated-use condition — for authorized testing only — not, on the evidence checked, an independently verified pre-action check. One third-party assessment of the default configuration reports no approval gate for any tool call, including arbitrary shell execution; that is a secondary assessment of a default configuration, and it is labeled as such. PentAGI's own logging is complete — every command and output, stored. Stored by the operator, in the operator's database, editable by the operator. It is a log. It is not a witness.

The gate seam. Between the planner's decomposition and the executor's tool invocation, inside the worker container — the one seam every consequential action passes through. A gate there requires a scope attestation bound to the named target before the first probe, decides each consequential action before execution rather than logging it after, and seals the record somewhere the operator's database credentials do not reach.

CASE 06 · DUAL-USE ONLY

8. AutoHunter — dual-use only

The record, at its grade. No documented malicious use of AutoHunter was found — not in Unit 42's reporting, not in GTIG's framework-abuse reporting, not in incident reporting generally. It appears in Unit 42's ecosystem table as a row, with no host count, no intrusion, and no campaign. This paper does not promote it beyond that evidence. It is a dual-use testing tool, assessed here because its design contains the cleanest version of a failure the other five exhibit in the wild.

What it is. A public project, created in July 2026: an automated vulnerability-mining system for authorized bug-bounty work. A Collector continuously produces targets from asset-mapping engines, probes liveness, and scores them. A Worker — one per target — autonomously reconnoitres and runs a real toolchain. A Reviewer filters junk findings. Then, and only then, a human: the operator re-reviews what survives, in minutes, and approves or rejects. It runs unattended, around the clock.

The swap. Three configuration variables — a key, a base URL, a model name. DeepSeek by default; OpenAI, Claude, and others by setting. Models without native tool calling are accommodated by simulating it. The framework does not care which model is plugged in, and says so in its own configuration.

The distinctive risk. AutoHunter has internal filters — it mechanically skips government and military keywords and loopback addresses, and suppresses whole classes of false positives. They are finding-quality and scope-hygiene filters, set by the operator, inside the operator's system: self-grading, competently done. The structural problem sits one step later. After a finding is accepted, the system validates it broadly — scoping systems of the same type, by fingerprint, and testing them to establish whether a whole batch can be hit at once. The human checkpoint gates the submission of a finding. It does not gate the expansion of the target set. New targets enter the action set on the strength of a prior finding, with no new authorization — including, as the project's own reviewers note, the same kind of system deployed at an organization nobody was authorized to test.

The gate seam. Two seams: Collector to Worker, where an attested scope check belongs before the first probe of any collected target; and, decisively, approval to expansion — where each newly swept target requires its own scope attestation, on record, before it is touched. AutoHunter is the control case for this entire assessment: a tool built in good faith, with a

human in the loop, whose loop still expands past its authorization at exactly the seam where no independent gate sits.

09 9. Self-attestation: the recurring failure

Lay the six side by side and one component recurs, wearing different clothes.

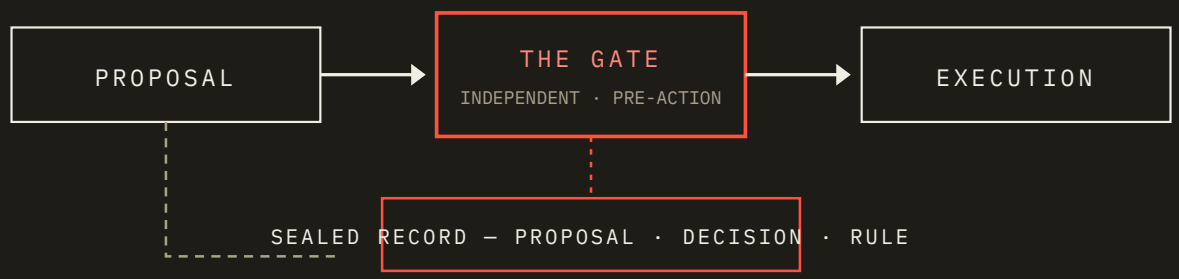
Cairn's operations generated their own AUTHORIZATION.md: the attacker created the task, and the creation was treated as the client's approval. CyberStrikeAI ships approval gates, allowlists, and audit logs — each of them disableable by the operator they exist to restrain. PentAGI logs everything, into a store the operator controls. Hermes' safety configuration was an override the operator wrote. AutoHunter's scope filters are set by the person they scope. ARTEX, the honest case, dispensed with the ceremony: no check was documented at all.

In every documented case, the only check present was controlled by the operator or by the model itself. This is not a coincidence of six designs. It is the default architecture of the field: verification performed inside the system being verified, by a party to the action, is a ceremony — and six frameworks, built independently, on three continents, for purposes ranging from enterprise testing to espionage, all arrived at the ceremony. The attacker does not have to defeat the check. The check was issued by the attacker.

10 10. The seam

The gate seam is the same shape in all six frameworks, because all six frameworks are the same shape: a proposal forms — an intent on a blackboard, a command from an orchestrator, a subtask from a planner, a script from a profile — and then it executes. Between those two events there is a gap, measured in milliseconds, in which nothing independent stands.

EXHIBIT B · THE SEAM, IN ALL SIX



AS OBSERVED IN ALL SIX: THE PROPOSAL LOOPS BACK TO THE OPERATOR FOR APPROVAL — SELF-ATTESTATION. THE CHECK WAS ISSUED BY THE ATTACKER.

Every framework here already separates proposal from execution — the seam exists in all of their architectures. What none of them put in it was a party other than the operator.

That gap is the entire remaining defensive surface. It cannot be filled by a better model, because the model is a swappable part, chosen by refusal profile. It cannot be filled by a more sophisticated actor profile, because the actor is no longer the variable — the same frameworks serve a state-nexus collection crew, a financially motivated campaign, and a man who cannot spell the crime he is committing. It cannot be filled by the framework's own safeguards, because the operator holds them.

It is filled, if it is filled, by a gate that is independent of all three — outside the model, outside the framework, outside the operator — deciding each proposed action before execution, against an authorization it did not receive from the person proposing, and sealing the record of proposal, decision, and rule where none of them can edit it. Proposal, decision, rule. We'll know what we did.

The walls came down at once, and they are not going back up. What is left to build is the one thing none of the six built: a witness that is not a counterparty.

Sources

1. Unit 42, Palo Alto Networks, "Cairn: How Threat Actors Are Using an Open-Source AI Framework to Launch Autonomous Attacks" (2026).

https://unit42.paloaltonetworks.com/cairn-framework-ai-autonomous-attacks/

2. Unit 42, Palo Alto Networks, “ARTEX and the Rise of AI-Powered Red Teaming Tools” (2026-11-11).

https://unit42.paloaltonetworks.com/artex-and-ai-powered-red-teaming-tools/

3. Unit 42, Palo Alto Networks, “‘Next Exploit Opportunities’: How a Threat Actor Used AI to Help Plan and Execute Multiple Types of Cybercrime Activities” (2026-10-21).

https://unit42.paloaltonetworks.com/ai-assisted-cybercrime-analysis/

4. Kaspersky Securelist, “CyberStrikeAI in the wild” (2026-11-02).

https://securelist.com/

5. Cloud Security Alliance, research note on AI penetration-testing frameworks, synthesizing TrendAI, Team Cymru, and CrowdStrike reporting (October 2026).

6. Google Threat Intelligence Group (GTIG) / Mandiant, reporting on open-source autonomous offensive frameworks observed in the wild (2026).

7. Nous Research, Hermes Agent project repository.

https://github.com/NousResearch/hermes-agent

8. VXControl, PentAGI project repository and documentation.

https://github.com/vxcontrol/pentagi

9. StanleyNull, AutoHunter project repository.

https://github.com/StanleyNull/AutoHunter

10. AIPentest, CyberStrikeAI project repository.

https://github.com/AIPentest/CyberStrikeAI

VERITY / THE PAPERS · Threat assessment of Thursday, November 19, 2026

Assessment of PenTesting Frameworks · Graded as published · Nothing on this page is graded higher than its source