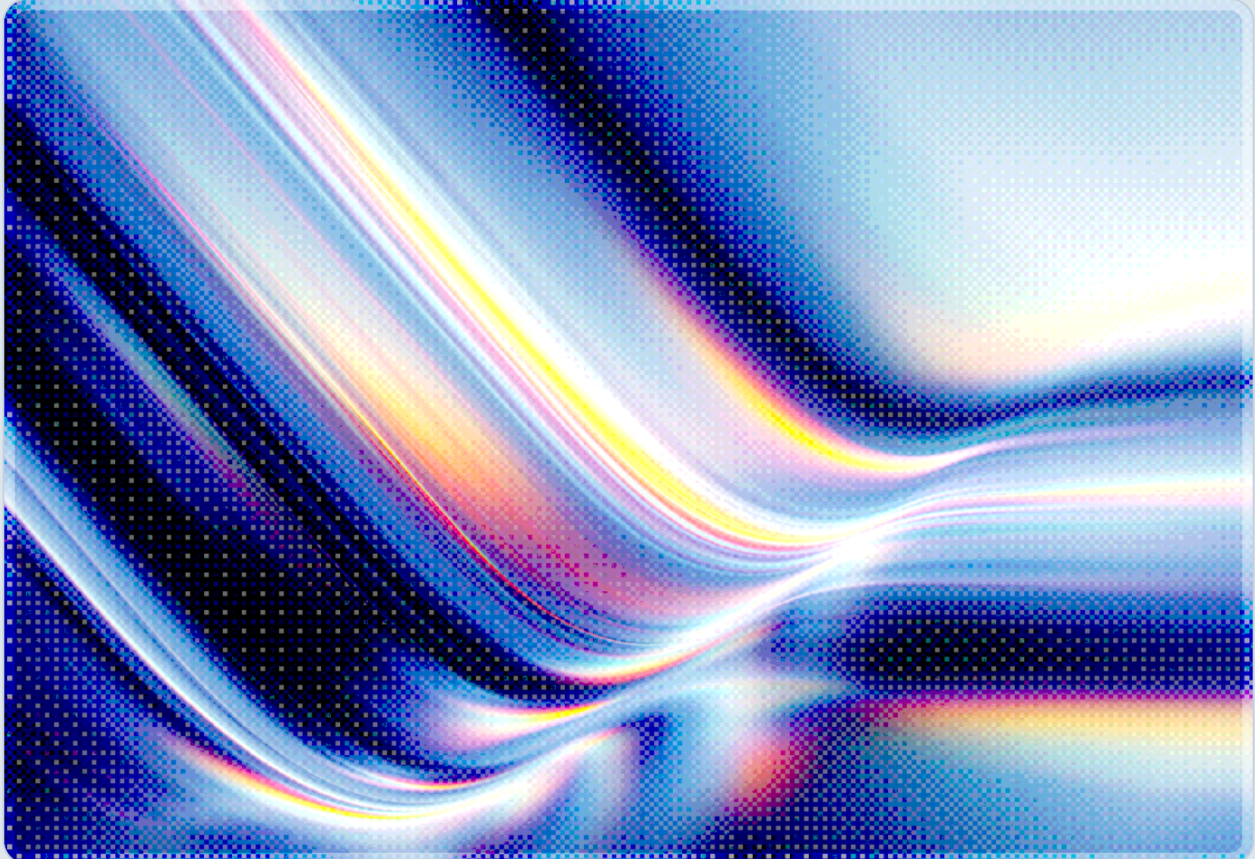


Fifty Dollars per Breach: Detecting Autonomous Agent Intrusion Campaigns

Security Implications and Guidance After the Cairn Operation

2026-10-21

 AI-assisted Rapid Research



CSAI

cloud
security
alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

On October 19, 2026, Reuters and the security firm NorForge disclosed that a Chinese-speaking operator had used Cairn, an open-source multi-agent platform, to attempt intrusions against more than 1,200 websites and to break into sites belonging to more than 100 companies, including Deere, SharkNinja, and Cummins [1][2]. NorForge's principal threat researcher told Reuters that he roughly estimated the operator's model spend at about US\$5,000 in tokens, or about \$50 per successful intrusion [1]. NorForge's own report gives a separate estimate of about \$5 per task when failed and early-terminated attempts are counted, and it warns that this figure carries considerable uncertainty [2]. The disclosure follows a September 2026 Gambit Security report in which a similar agent stack, with Cairn as its exploitation engine, compromised at least 27 companies and stole more than 600,000 unexpired payment card records [3].

- **The cost floor has moved.** Gambit measured a mean of \$25.46 per completed scan [3], NorForge estimated about \$5 per task [2], and a NorForge researcher estimated roughly \$50 per successful intrusion [1]. These measures differ in unit, but all put a breach in the tens of dollars, which makes mid-sized and long-tail web properties worth attacking.
- **Human supervision has become minimal.** In the NorForge-reported campaign, operator tasking ran to fewer than 100 Chinese characters, and successful intrusions averaged just under 23 hours of largely autonomous agent work [1][2]. Gambit's operator typed 1,951 prompts across 260 sessions, only a few per target [3].
- **Secondary assets feature prominently.** Two of three named victims described a standalone support site and a vendor-operated website, and NorForge documented agents pivoting from a hardened main application to a related control panel [1][2][4].
- **Each layer has its own detection opportunity.** Sustained machine-paced probing, uploader web shells and polyglot files, payment-page script tampering, and bulk database changes all leave observable traces. Because agents reached working access in under a day [1][3], defenders need to correlate these signals within hours, not days.
- **Exposed agent harnesses leak sensitive data, whoever runs them.** The exposed Cairn panels held hundreds of target secrets, 11 percent of which were verified as valid, and some exposed panels appeared to belong to legitimate testing operations [2][4].

Background

Cairn is an open-source, AGPL-licensed Python project published on GitHub that describes itself as a general-purpose AI state-space search engine, validated first on autonomous penetration testing [5]. Each task is a "project" with an origin and a predefined goal. Stateless worker agents post discoveries ("facts") and planned actions ("intents") to a shared blackboard, which a human operator can seed with hints, and a project graph records every branch, including failed paths [2]. Cairn's author, whose social media handle Reuters gives as lēlxia0, stated on X that the platform "was not designed for hacking attacks" and that harmful use is not the author's intention [1]. The project should not be confused with an unrelated AI malware-analysis tool of the same name released by Cisco Talos [6].

NorForge found the operation by searching Shodan and Censys for internet-exposed Cairn control panels. Beginning around October 2, it identified more than 20 panels that it judged likely malicious and grouped them into three clusters [1][2]. The largest, which NorForge designates AIA0001, is a Chinese-language cluster of 14 linked panels running more than 1,100 attack projects against online shops in the United States, Canada, Japan, and Australia. Targets were pre-validated with Playwright and selected from a traffic range of roughly 200,000 to 1.5 million (units not specified by NorForge) [2]. Every AIA0001 task ran on a worker labeled `claudecode_deepseek-v4.1-flash`. NorForge infers that this could mean a Claude Code harness driving DeepSeek's V4.1 Flash model, but it does not confirm the model identity [2]. The second cluster, AIA0002, is English-language, spans 10 panels, and pursued credit card and cryptocurrency theft through a worker labeled `pi_openrouter_deepseek`. A third single-panel cluster was rated malicious with low confidence [2].

Reuters reviewed one operator's records and identified more than 100 breached companies, but it could not establish how much data, if any, was taken [1]. The named victims described limited impact. Deere reported remediation and no indication that sensitive information was accessed, SharkNinja said the intrusion hit a standalone product-support site and was contained, and Cummins said it was investigating a possible attack on a website run by a third-party vendor [1]. NorForge itself did not name victims. It observed that all clusters appeared to have financial motives and that there was no obvious state-sponsored activity [2]. Chinese-language tasking alone does not establish the operator's nationality or any state backing.

The Cairn disclosure is the second public view into the same class of operation in as many months. Gambit Security obtained access to an attacker's staging server and reconstructed a campaign, active since at least July 2026, that combined three open-source agents. Strix handled vulnerability discovery, Cairn handled autonomous exploitation toward objectives such as a shell or administrative access, and Hermes handled orchestration and post-exploitation [3]. Gambit reports that Hermes ran on Anthropic's Opus

4.6 after newer models refused the operator's requests, that Strix ran on GLM 5.2 and later DeepSeek V4 Pro, and that the Cairn component ran on DeepSeek V4.1 Flash, consistent with the worker label NorForge later observed in AIA0001 [3][6]. The table below compares the two disclosures.

Dimension	Gambit Security (Sept. 22, 2026) [3]	NorForge AIA0001 (Oct. 19, 2026) [1][2]
Agent stack	Strix (discovery; GLM 5.2, then DeepSeek V4 Pro), Cairn (exploitation; DeepSeek V4.1 Flash), Hermes (orchestration; Opus 4.6)	Cairn with Claude Code-labeled worker; label indicates DeepSeek V4.1 Flash (unconfirmed; consistent with Gambit's findings)
Scale	105 attack projects in six days; at least 27 companies compromised	1,100+ projects across 14 panels; 1,200+ sites attempted; 100+ companies breached
Reported cost	\$25.46 mean per completed scan (\$3.13–\$79.31)	~\$50 per successful hack (researcher estimate) [1]; ~\$5 per task [2]
Time to access	Usually less than a day, often a few hours	Just under 23 hours on average for successful projects
Objective	Payment card theft and skimmer deployment	Code execution and an uploader web shell, likely for persistent access
Observed impact	600,000+ card records from two companies; skimmers confirmed on 19 named victims, with 100+ further sites carrying a related skimmer (source counts vary) [3][6]; 180 tables dropped at one retailer	One live uploader shell confirmed; data exfiltration not established

Security Analysis

The Economics Change Who Gets Targeted

Defenders have long relied on an implicit cost model: skilled intrusion work is scarce, so attackers concentrate it on high-value targets, and smaller or secondary properties benefit from obscurity. These campaigns suggest that model no longer holds. When an exploitation attempt costs a few dollars and a success costs tens of dollars, an operator can sweep a ranked list of mid-traffic retailers and accept a success rate of about one in ten, which is the average NorForge estimated across the harnesses it observed [2]. Gambit observed the same logic in target selection. Its operator filtered traffic-ranking lists toward shops running custom code, on the presumption that bespoke applications would be more vulnerable than major hosted platforms [3]. These campaigns suggest that the "too small to bother with" tier of web estates, including regional storefronts, campaign microsites, support portals, and vendor-hosted pages, increasingly carries risk proportional to its exposure rather than its perceived value. NorForge cautions that its success figures are not a universal AI hacking success rate, because the clusters pursued different goals and are not directly comparable [2][4].

Agents Behave Differently from Both Scanners and Human Operators

The behavioral profile in these reports sits between that of a commodity scanner and that of a human penetration tester, and that position matters for detection. Unlike a scanner, the agents persist. Successful AIA0001 projects averaged roughly 140 facts and just under 23 hours of work, and one ran for more than five days [2]. Unlike most human operators, they persist through extended trial and error without fatigue. In one Reuters-described case, agents found no weakness in a large company's public application, used public records to locate related infrastructure, found a control panel with a subtle weakness in how it processed data, and spent hours iterating before the server executed their commands [1]. NorForge's confirmed compromise of a hardware-parts vendor followed a similar multi-step chain. By the agent's own account, which NorForge's verification did not independently confirm, it combined unauthorized writes to a memcached session store, unsafe PHP object deserialization, and a gadget chain in legacy Zend Framework components [2][4]. The tooling NorForge observed during a brief window of exposure fits this pattern. It included OWASP Amass and dnsrecon for asset discovery, ffuf and Nmap for enumeration, PHPGGC deserialization gadget chains, headless Playwright and Chromium, and a proof-of-concept lab for the StyleSmuggler (CVE-2026-75650) Magento report-poisoning-to-code-execution chain [2].

For a security operations center, this profile suggests that per-request signatures will catch some tooling, but the stronger signal is temporal and structural. A single source, or a rotating pool of sources, sustains hundreds of distinct, adaptive requests against one property over many hours. It expands from the primary hostname to subdomains and adjacent panels, and it submits clusters of near-identical payload variants against a single parameter. Gambit's operator routed traffic through commercial proxy services, including IPRoyal, 711proxy, and 1024proxy, providers known for residential IP pools [3]. That practice weakens IP reputation as a control and makes session-level and target-level correlation more important than source blocking.

Post-Exploitation Leaves Concrete Artifacts

Both campaigns produced durable artifacts that standard controls can detect when they are deployed and monitored. The AIA0001 objective was an uploader web shell rather than a command shell, presumably to maintain lasting access to shop and payment systems [1][2]. At one large U.S. retailer, an unauthenticated upload endpoint accepted a polyglot file that combined an image signature with PHP code and stored it in a web-accessible directory. Only one of several load-balanced backends actually executed the file as PHP, which illustrates how inconsistent server configuration can determine whether a write becomes code execution [4]. Gambit documented a wide range of skimmer-injection techniques, which are summarized in the table below with corresponding detection points [3].

Observed technique [3]	Primary detection point
Loader appended to legitimate JavaScript libraries, original file timestamps restored	Content-hash file integrity monitoring (not modification time)
Foreign script tags on checkout pages pointing to lookalike CDN domains	Payment-page script inventory, Content Security Policy violation reports
Loader hidden inside analytics tag blocks, padded with whitespace	Tag-manager change auditing and rendered-DOM monitoring
Payloads in compromised storage buckets behind a CDN, or in poisoned server-side caches	Object-storage write logging and cache-content validation
Payloads stored in database content fields or injected into container init steps	Database change auditing and deployment-manifest drift detection

Observed technique [3]	Primary detection point
Cron jobs in application log directories that re-inject removed skimmers	Scheduled-task inventory and alerting on cron entries outside approved paths

The Gambit campaign also demonstrates a destructive failure mode that defenders should plan for even when the attacker's motive is theft. The operator's playbook instructed agents to wipe source card fields after extraction, and an agent-generated cleanup routine matched table names so broadly that it dropped 180 tables at one retailer, including backup tables [3]. Agent-driven intrusions may therefore cause collateral data loss that no human operator intended, which raises the importance of immutable, separately credentialed backups.

Exposed Offensive Harnesses Are a Two-Way Risk

NorForge's view into the Cairn operation was made possible by operators leaving control panels open to the internet, and that exposure cuts both ways. NorForge reported that the panels allowed project control functions, including starting, stopping, redirecting, and injecting data into running operations, although it did not test whether these worked without authorization [2]. Its Trufflehog scan of the collected data found hundreds of secrets belonging to targets, 11 percent of which verified as valid [2]. NorForge also observed exposed panels that appeared to belong to legitimate demos or penetration-testing firms, and it noted that running real customer tests on an unsecured panel could place those clients at risk [2][4]. Organizations that commission agent-driven penetration tests, or run them internally, should therefore treat the agent harness as sensitive infrastructure holding credentials and vulnerability findings for their own estate.

Recommendations

Immediate Actions

Organizations operating e-commerce, support, or vendor-hosted web properties should first establish whether they appear in the disclosed activity. NorForge has published redacted indicators and offers unredacted lists to bona fide researchers on request, and Gambit's report includes an IOC table of domains, IPs, and URLs for blocking and retrospective search in DNS, proxy, and web server logs [2][3]. Security teams should then hunt web roots and upload directories for recently created files, especially image-extension files containing PHP or other server-side code, and should preserve evidence before cleanup. NorForge-derived guidance stresses that an unexpected upload should trigger investigation of

access and persistence, not just a component patch [4]. For Magento and other PHP commerce stacks, teams should prioritize review of the StyleSmuggler chain, memcached instances reachable with write access, PHP deserialization paths, and legacy Zend Framework components named in the reporting [2]. Finally, any organization that has used AI-driven penetration-testing platforms should verify that no harness console is internet-reachable without authentication and should rotate credentials that such tools have handled.

Short-Term Mitigations

Over the following weeks, the priority should shift to making the artifacts described above reliably visible. Payment pages warrant particular attention. PCI DSS v4.0 requirements 6.4.3 and 11.6.1, which became mandatory on March 31, 2025, already require an authorized inventory of payment-page scripts and a mechanism to detect unauthorized changes to page content and security headers [7]. Smaller merchants should note that the PCI Security Standards Council removed these requirements from Self-Assessment Questionnaire A and replaced them with an eligibility criterion that the merchant's site is not susceptible to script attacks [8]. Because that population overlaps with the mid-sized and long-tail shops these campaigns targeted, SAQ A merchants should confirm the basis for that attestation rather than assume script monitoring is out of scope. The Gambit techniques show why that monitoring must examine rendered content and file hashes rather than relying on file timestamps or a single injection location [3].

Hardening and detection should advance alongside payment-page controls. Web servers should be configured so that upload directories never execute code, and that configuration should be verified consistently across every backend behind a load balancer, given the single-backend execution NorForge-derived reporting describes [4]. SOC detection content should add session-level analytics for sustained, adaptive probing of a single property across many hours, along with expansion from a primary site to adjacent subdomains and administrative panels. Response playbooks should assume an intruder can move from first probe to working access within a day [1][3]. Backups should be immutable and isolated from any credentials reachable by web applications, so that broad agent cleanup routines cannot destroy recovery points [3].

Strategic Considerations

The longer-term lesson is that attack surface management must be priced against an adversary whose marginal cost per attempt is close to zero. Asset inventories should explicitly include standalone microsites, support portals, and third-party-operated properties that carry the organization's brand, since two of the three named victims described exactly those assets [1]. Vendor contracts and third-party risk reviews should address patch cadence, upload handling, and incident notification for vendor-hosted web properties. Detection and response tempo should be benchmarked against agent timelines measured

in hours. Gambit argues that thresholds and on-call processes calibrated for human attackers may be too slow and that recovery planning should define a minimum viable business rather than stopping at a database restore [3].

Organizations should also expect defensive agents to become necessary for parity, and they should govern those agents, and any commissioned offensive agents, with the same identity, logging, and exposure controls they apply to production systems. Model providers have intervened against earlier abuse of agentic coding tools, as Anthropic did in its GTG-1002 disclosure [9]. However, both the AIA0001 worker label and Gambit's model assignments indicate that operators can substitute lower-cost alternative models, including open-weight ones, so defenders should not treat provider-side enforcement as a primary control [2][3].

CSA Resource Alignment

CSA's most directly relevant published work is *Hugging Face's Autonomous AI Agent Breach* (July 2026), which CSA described as the first publicly disclosed production breach driven end to end by an autonomous agent [10]. That incident involved an agent operating inside infrastructure after initial access, while Cairn operations show agents conducting the external phases of reconnaissance, exploitation, and web shell placement at scale. Together, they illustrate the full intrusion lifecycle now available to operators with modest resources. The Hugging Face report's emphasis on runtime visibility, distinguishing agent activity from human activity, and machine-speed response applies equally to the internet-facing detection problem described here.

The *Agentic AI Red Teaming Guide* is relevant mainly to organizations that deploy or commission agentic testing systems [11]. Its methods for testing agent authorization, tool misuse, and memory manipulation apply directly to harnesses like Cairn, and the exposed-panel findings show why those systems need governance, scoping, and protection of the sensitive findings they accumulate. For the offensive use of agents itself, *Using AI for Offensive Security* covers AI agents across reconnaissance, scanning, vulnerability analysis, and exploitation, the same phases Cairn automates [12]. It can help defenders apply agentic tooling to find the deserialization, upload, and authentication-bypass weaknesses that Cairn agents exploited before adversaries do.

CSA's MAESTRO agentic AI threat modeling framework offers a layered method for reasoning about both the attacker's agent stack and an organization's own agent deployments [13]. It can help teams model how orchestration, tool access, and shared memory structures such as Cairn's blackboard create both capability and exposure. The AI Controls Matrix (AICM) v1.1 provides the control baseline for operationalizing the recommendations above [14]. Its Threat and Vulnerability Management and

Application and Interface Security domains cover exposure and upload handling, Logging and Monitoring covers session-level detection and payment-page integrity, Supply Chain Management, Transparency, and Accountability covers vendor-hosted properties, and Identity and Access Management covers the credentials held by offensive and defensive agent harnesses.

References

- [1] Satter, R. (Reuters). "[How one hacker used AI agents and a few thousand dollars to break into scores of sites.](#)" Reuters via CP24, October 19, 2026.
- [2] Fagerland, S. "[Inside Cairn: Open Windows into massive AI-Driven Cyber Operations.](#)" NorForge, October 19, 2026.
- [3] Sela, E. "[AI Agents Are Hacking Online Retailers for \\$25 a Company.](#)" Gambit Security, September 22, 2026.
- [4] Smith, B. "[Cairn AI Attacks: Exposed Panels, Webshells and Secrets.](#)" GridinSoft Blog, October 20, 2026.
- [5] oritera. "[Cairn.](#)" GitHub, accessed October 21, 2026.
- [6] Toulas, B. "[Malicious AI agents steal 600K credit cards, infect 100+ sites with skimmers.](#)" BleepingComputer, September 23, 2026.
- [7] PCI Security Standards Council. "[New Information Supplement: Payment Page Security and Preventing E-Skimming.](#)" PCI SSC Blog, accessed October 21, 2026.
- [8] PCI Security Standards Council. "[Important Updates Announced for Merchants Validating to Self-Assessment Questionnaire A.](#)" PCI SSC Blog, accessed October 21, 2026.
- [9] Anthropic. "[Disrupting the first reported AI-orchestrated cyber espionage campaign.](#)" Anthropic, November 2025.
- [10] Cloud Security Alliance. "[Hugging Face's Autonomous AI Agent Breach.](#)" CSA, July 2026.
- [11] Cloud Security Alliance. "[Agentic AI Red Teaming Guide.](#)" CSA, May 2025.
- [12] Cloud Security Alliance. "[Using AI for Offensive Security.](#)" CSA, 2024.
- [13] Cloud Security Alliance. "[Agentic AI Threat Modeling Framework: MAESTRO.](#)" CSA, February 2025.
- [14] Cloud Security Alliance. "[AI Controls Matrix \(AICM\) v1.1.](#)" CSA, 2025.