Chinese-nexus actor's Cairn AI agents found a zero-day on their own

by CJ Moses | on 20 NOV 2026 | in Announcements, Artificial Intelligence, Security, Identity, & Compliance, Thought Leadership | Permalink | Comments | Share

Amazon Threat Intelligence is sharing findings from an investigation into a Chinese-nexus threat actor using autonomous AI agents to conduct hacking operations with minimal human involvement. The actor gave the agents a simple goal: "obtain shell access." During the campaign, the agents found a zero-day in an open-source application and used it to gain full control of a server. These findings show that agentic offensive operations are already happening, and defenders should prepare for the speed and persistence these tools enable.

Discovery

On September 10, 2026, Amazon Threat Intelligence identified a server with a publicly hosted instance of Cairn, an open-source autonomous AI penetration-testing framework, running without authentication. Because the operator left the server publicly accessible, the operational history of the platform was visible to anyone on the internet, including project configurations, actor prompts, and AI output. The operator wrote prompts in Simplified Chinese, and the timestamps in executed scripts were configured to China Standard Time.

Analysis of the instance revealed a campaign targeting IP addresses associated with universities in East Asia outside mainland China over an approximately 48-hour period beginning September 8, 2026.

Cairn and an overview of this campaign

Cairn was built as a research and problem-solving engine for defenders. Operating it is simple. Given a starting point and an objective in plain language, it coordinates multiple AI agents to find a path to that goal. That same ease of use makes it attractive to threat actors. In this campaign, the actor provided a list of targeted IP addresses with variations of the goal to "obtain shell access." Based on those inputs, the AI agents planned, executed, and adjusted their approach as they learned, without the operator choosing each next step.

The campaign included several operations that varied widely in complexity and outcome. The following operations show both the persistence and the capability of autonomous AI agents.

Persistence against a hardened target

The most extensive operation targeted a Redis (remote dictionary server) instance at one university that was configured to allow access without authentication. Redis is a database that keeps frequently used data in memory so applications can retrieve it quickly, which can make it a valuable foothold inside a network. Over roughly 16 hours and 148 sequential actions, the agents worked through a long list of known techniques for taking control of an open Redis server. These included abusing Redis's replication feature to load unauthorized code, escaping its scripting sandbox, writing to its configuration files, and corrupting its memory. The agents also attempted Log4Shell injection against another service running on the same system.

When those attempts failed, the agents mapped the internal Docker network topology through a server-side request forgery (SSRF) vulnerability in a co-located Qdrant instance and launched credential sprays against PostgreSQL, SSH, and several other services. None of these attempts succeeded.

A separate operation ran for more than 6 hours and targeted a university GPU server, which is a group of high-performance computers built for computing-intensive research. The agents first built a list of about 80 real usernames by researching the university's public presence, including faculty directories, lab websites, archived web pages, and academic papers. Then, they tested more than 30,000 credential combinations against the server's remote login (SSH), and, on other systems across the university network, an admin dashboard, a network storage device, a VPN server, and the university's campus portal. They never found a working login.

Neither operation succeeded, yet together they show that the agents kept adapting for nearly a full day without human direction.

Zero-day discovery

Against a target running an open-source medical imaging AI application, the outcome was different. The agents identified the application's framework, mapped its API endpoints, and found an upload endpoint with a filename parameter. They tested that parameter for path traversal, and the parameter was vulnerable. In about 11 minutes, the agents went from mapping the application's API to arbitrarily writing files on the server.

In a later session, the agents used the same vulnerability to write a Python reverse shell payload to sitecustomize.py, a file Python automatically imports at interpreter startup. The agents then triggered a training API endpoint to spawn a new Python process. The new process imported the payload and established a reverse shell running as root.

At the time of discovery, this path traversal vulnerability in the medical imaging application did not correspond to any assigned security advisory and was unpatched. Amazon Threat Intelligence assesses with high confidence that it was a zero-day based on the absence of any corresponding fix, issue, advisory, or prior mention in vulnerability databases for the affected open-source project.

Subsequent use

In another session, the threat actor stepped in. After the agents discovered the zero-day, the actor then provided the agents with a proxy that tunneled into a second university network. The server that the proxy connected to was running the same vulnerable software, and we infer the actor had used the zero-day to compromise it and establish the tunnel. The actor directed the agents to use the tunnel to target other machines in that network.

Attribution

Amazon Threat Intelligence assesses with moderate confidence that this is a Chinese-nexus threat actor based on the Simplified Chinese prompts, the China Standard Time script timestamps, and the targeting of universities likely of strategic interest to China.

Actions taken by AWS

Amazon Threat Intelligence notified the project's maintainers of the vulnerability, and they released a fix and a GitHub Security Advisory.

The affected organizations were non-AWS customers that AWS had no direct path to notify. Instead, AWS worked with a government partner to notify the organizations about the activity.

Adoption is growing

Gambit Security was one of the first organizations in the industry to document the use of Cairn by a threat group. Its research describes a separate, financially motivated actor using Cairn to target online retailers. Together, these cases show that this autonomous AI agent framework is being adopted in both espionage and criminal operations. Additional details are available in Gambit Security's report.

What this means for defenders

The operational tempo in these cases compresses the window available for defensive response. An agent that discovers and exploits a zero-day in minutes completes its activity faster than many security operations centers can manually triage an alert. The agents that failed against the hardened Redis target ran continuously for 16 hours without breaks, sleep, or working-hour constraints. To keep pace, organizations should prioritize high-confidence detections that trigger automated response, with humans overseeing outcomes rather than reviewing every individual action.

Behind the scenes, AWS active defense systems use global honeypots (decoy systems that attract threat actors) and network traffic analysis to identify actors that are scanning for vulnerable systems, and then help restrict that activity at the network level. These protections are built into the AWS network at no extra cost to customers.

Indicators of compromise

The following addresses appear in records dated September 8 to 10, 2026.

142[.]4.98.145

142[.]4.107.34

If you have questions about this post, contact AWS Support or start a new thread on AWS re:Post.



CJ Moses

CJ Moses is the CISO of Amazon Integrated Security. In his role, CJ leads security engineering and operations across Amazon. His mission is to enable Amazon businesses by making the benefits of

security the path of least resistance. CJ joined Amazon in December 2007, holding various roles including Consumer CISO, and most recently AWS CISO, before becoming CISO of Amazon Integrated Security September of 2023.

Prior to joining Amazon, CJ led the technical analysis of computer and network intrusion efforts at the Federal Bureau of Investigation's Cyber Division. CJ also served as a Special Agent with the Air Force Office of Special Investigations (AFOSI). CJ led several computer intrusion investigations seen as foundational to the security industry today.

CJ holds degrees in Computer Science and Criminal Justice, and is an active SRO GT America GT2 race car driver.

TAGS: Security Blog

Like Share

Comments

Log in to comment