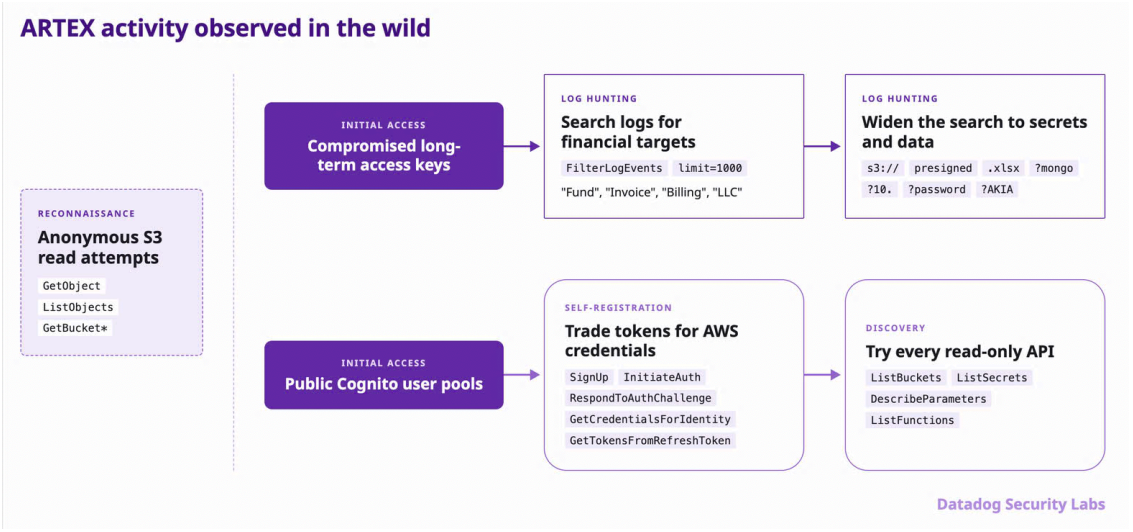
# On the hunt for ARTEX: understanding agentic attacks in AWS environments | Datadog Security Labs

By Julie Agnes Sparks

Published: 2026-11-09 · Archived: 2026-11-11 02:00:30 UTC

## Key points

- In October 2026, Datadog observed in-the-wild activity from ARTEX, an AI-native penetration framework, against several AWS environments, with a focus on accessing financial data.
- Using compromised long-term IAM access keys, ARTEX searched through CloudWatch using `FilterLogEvents`, hunting for financial information, file paths and secrets such as long-lived `AKIA` access keys and passwords.
- We also observed unauthenticated abuse of public Amazon Cognito user pools, including anonymous sign-ups, token collection, and follow-up AWS account enumeration attempts.
- In early November 2026, Datadog tied ARTEX activity to IP addresses that resolved to a hosted console, and found hundreds of additional internet-exposed ARTEX consoles, often in Mandarin and hosted with providers in locations including Singapore and Hong Kong.




ARTEX activity observed in the wild, from anonymous S3 reads to compromised access keys and public Cognito user pools (click to enlarge)

## Background on ARTEX

ARTEX is an AI-native penetration framework in which an operator supplies a target and an objective, and a team of LLM agents plans and runs the attack. A Chinese developer published it on GitHub in July 2026 and made it closed-source on October 8, after it had passed 12,000 forks.

Microsoft tied it to Storm-0940, a China-aligned actor behind a recent supply-chain compromise. In recent months, ARTEX has been documented attacking Okta, Entra ID and Cognito across IT and software vendors.

Those reports center on identity providers and web application logs. In this post, we provide more details on the cloud control-plane view, in particular how ARTEX behaves once it has compromised an initial set of AWS credentials, through leaked long-lived access keys or following a successful anonymous sign-up on a public Cognito pool.

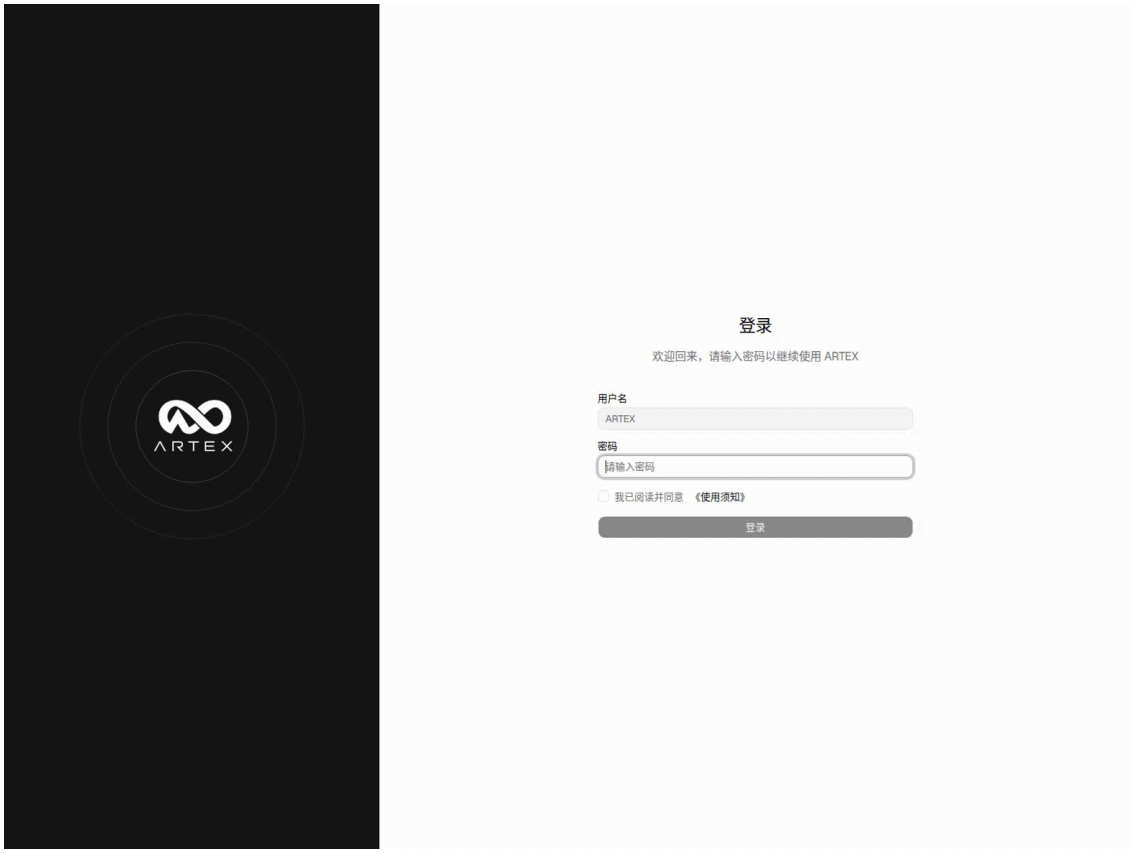
First, we will dive into ARTEX as a platform. Then, we'll describe specific behaviors and indicators we identified in the wild.

## Inside the ARTEX console

ARTEX started as a repository and has grown into a platform running on live infrastructure. Operators run it as a hosted web console, and many of those consoles face the internet. We found hundreds of hosts serving the same Chinese-language login page, titled `ARTEX – 自主渗透测试控制台` (ARTEX — Autonomous Penetration Testing Console). We've also seen the same platform appearing under other names:

| **Page title** | **English translation** |
|---|---|
| ARTEX — 自主渗透测试控制台 | ARTEX — Autonomous Penetration Testing Console |
| ScopeWeaver — Autonomous Penetration Testing Console | (original in English) |
| LUCK — 自主资产控制台 | LUCK — Autonomous Asset Console |
| 观澜SEC渗透测试平台 | Guanlan SEC Penetration Testing Platform |

A similar console is rendered across many of these hosts.




The ARTEX console login page (click to enlarge)

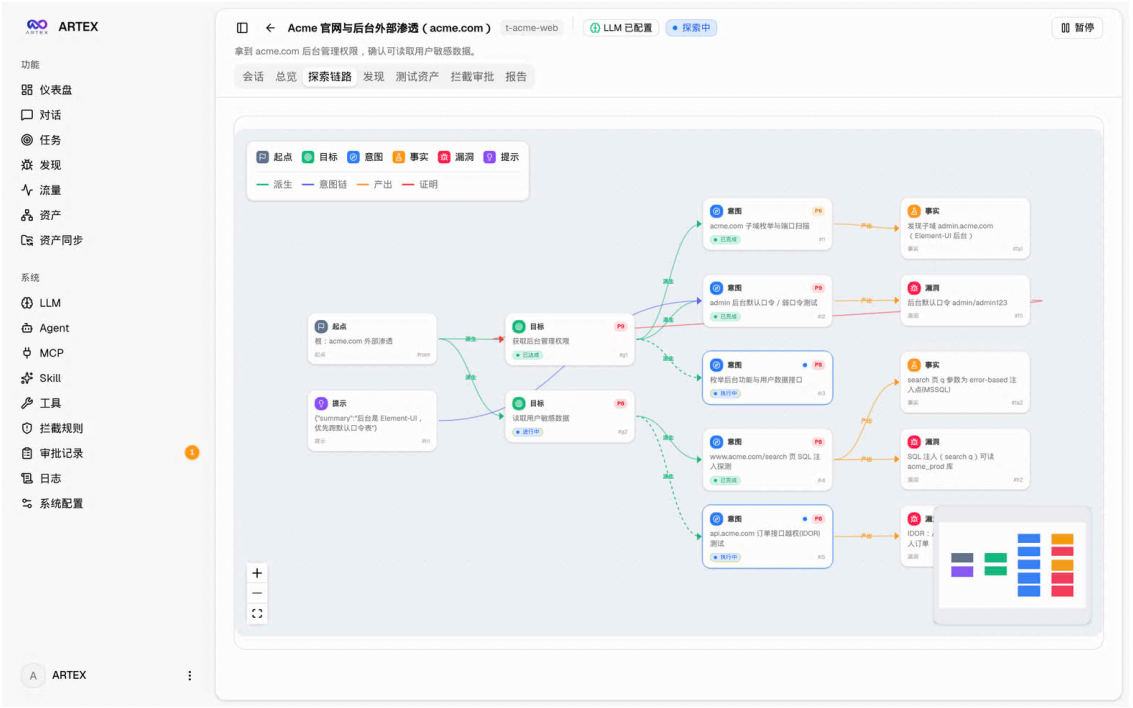
The platform user supplies API keys to chosen AI inference providers and points ARTEX at almost any provider through three request formats:

- `openai` : OpenAI Chat Completions and any compatible gateway, which covers OpenAI, DeepSeek, Qwen, MiniMax, OpenRouter-style resellers and self-hosted models.
- `openai-responses` : OpenAI's newer Responses API.
- `anthropic` : Anthropic's Messages API, plus vendors that expose an Anthropic-compatible endpoint, such as DeepSeek.

When one provider bans or cuts off a key, ARTEX moves to the next to continue activity. Therefore, a single campaign can be spread across several providers and keys.

An agent planner directs a small team of workers. The planner sends "intents" to workers, which run three at a time by default. A main agent handles chat with the operator, with a `/btw` command for side questions.

Everything is recorded on a "blackboard" graph of tasks, goals, intents, facts, findings, hints and compression digests. Each task carries operator-defined allow and deny constraints, a timeout with a wrap-up step, and a planner heartbeat of at least 10 minutes. A fixed task timeout would explain bursts that end at a consistent length, though we have not confirmed the value operators set.



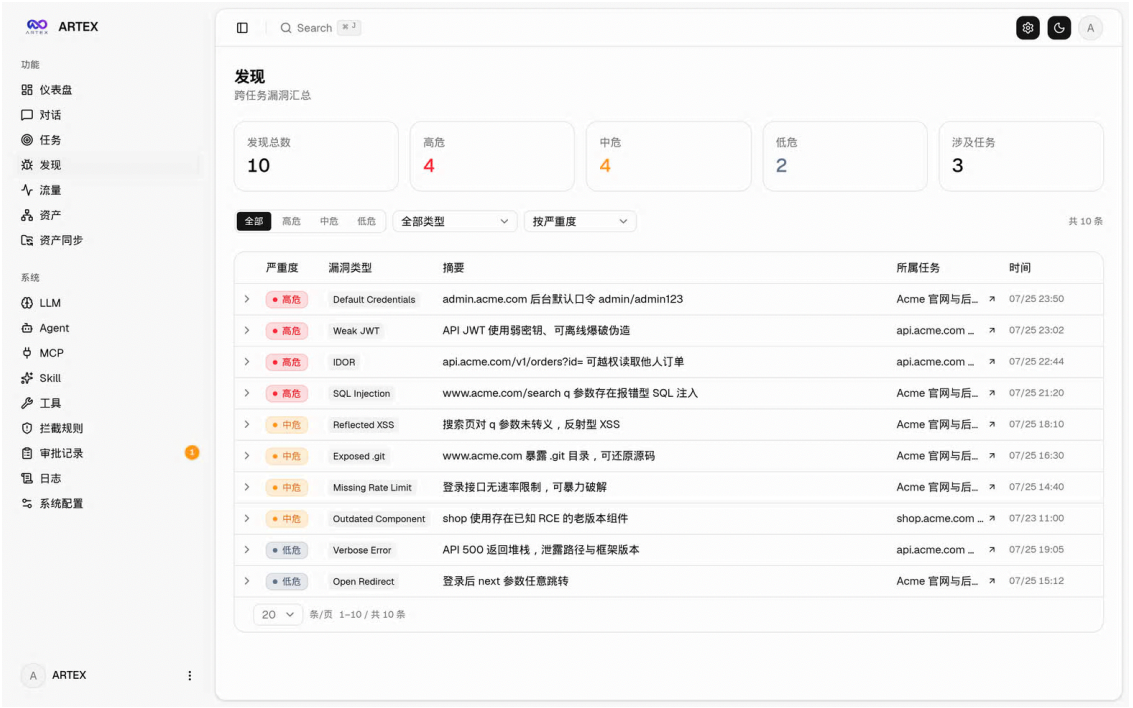

[The ARTEX exploration graph, linking goals, intents, facts and findings for a demo engagement against acme.com (click to enlarge)]

Command interception rules, written as regex or plain strings, can allow, deny or ask a human for a judgment. In some versions, a global asset blocklist excludes `.gov`, `.gov.cn`, `.edu` and `.edu.cn` by default.

API paths expose core routes, such as:

```
 /tasks
/exploration/findings
/exploration/graph
/exploration/intents
/traffic
/companies
/llm/profiles
/llm/pool
/intercept
/mcp
/skills
/agents
/notify
```

One version we observed shipped with a sample demo engagement. It walks through a full external-to-internal compromise, from a public web app to a DMZ foothold, internal lateral movement, and finally the Active Directory domain controller.




The ARTEX findings page listing vulnerabilities from the sample demo engagement (click to enlarge)

We often found the agent reading the front end of web apps hosted in S3, looking for build manifests and runtime configurations.

Wiz noted that before each fuzzing run, ARTEX requested a path that should not exist ( `/zzz-nonexistent-xyz123` ) to learn what "not found" looks like. We saw the same habit against S3, with keys such as `zzz_nonexistent_abc.txt` and `Notexist-xyz` . A request for an object named like a placeholder, right before a burst of guesses, is a useful signal to hunt on.

This behavior is one example from agentic pentesting tools, which continue to evolve whether they're used directly as a tool or hosted as an interactive platform.

## ARTEX sightings in AWS environments in the wild

In this section, we share some activity we've identified in real-world AWS accounts, and that we have attributed to ARTEX usage.

The two initial access vectors we've witnessed ARTEX start from are:

- Compromised long-term AWS access keys
- Public Cognito user pools

Before either, every session we reviewed opened with unauthenticated S3 requests. In CloudTrail, these show a `userIdentity.type` of `AWSAccount` and an `accountId` of `anonymous` , calling `GetObject` , `ListObjects` and similar read APIs.

### Activity cluster 1: Post-exploitation following compromised access keys

In one instance using a valid long-term access key, ARTEX's most frequent action was `FilterLogEvents`. The most common searches were various naming conventions of cryptocurrency companies, hedge funds, and related financial institutions. The terms combine a company name with words like Fund, Invoice, Billing and LLC.

Every `FilterLogEvents` call was set with a `limit` of 1000, below the API's 10,000-event maximum. That constant shows up across sessions.

On October 20, the same long-term key was used again from `103.54.154[.]40`, an address associated with `zeofast[.]cn`. This session widened the search, looking for file and storage links, internal secrets, internal IP addresses, and database information.

| Files and storage links | Databases, internal IPs and secrets | Errors |
|---|---|---|
| `"s3://" \|\| "presigned" \|\| ".pdf" \|\| ".xlsx" \|\| ".csv" \|\| "rawData_" \|\| "transformedData_"` | `?mongo ?172. ?10. ? password ?AKIA` | `?Error ?❌ ?⚠` |

## Activity cluster 2: Self-registration through public Cognito user pools

In an instance where no long-term access keys were obtained, ARTEX went after public Cognito user pools. Wiz reported account-recovery abuse through `InitiateAuth`, `ForgotPassword` and `ResendConfirmationCode`. We saw the agent go further, registering an account, collecting tokens, trading them for AWS credentials, then enumerating resources.

One example came as a spike on October 26 and 27 from `154.64.252[.]77`. The observed `SignUp` calls sometimes named the tool in the user agent such as `Mozilla/5.0 ARTeX`, while other user agents included curl or typical browser user agents.

Getting through sign-up was noisy. The pool had a `PreSignUp` Lambda trigger that checked a tenant domain, and the agent iterated against it and continued probing. For example, below are errors generated as the agent continued attempting to sign up:

```
 PreSignUp failed with error Invalid tenant domain: api[.]github[.]com.
PreSignUp failed with error Invalid tenant domain: example[.]com
PreSignUp failed with error tenantId does not match tenant domain.
PreSignUp failed with error Invalid tenantDomain format: hxxps[://]artex-hqheizfk[.]invalid/x.
Attributes did not conform to the schema: Type for attribute {custom:isAdmin} could not be determined
```

Once registered, it walked the standard user pool flow: `SignUp`, `InitiateAuth`, `RespondToAuthChallenge`, `GetUser` and `GetTokensFromRefreshToken`, followed by `Get*` requests and attempts to call `ListUserPools`.

In CloudTrail, that session appears under the `CognitoIdentityCredentials` role session name with short-term AWS credentials. The agent used it to try nearly every read-only discovery call it could, such as:

- `ListBuckets`

- `GetBucket*`
- `ListSecrets`
- `DescribeParameters`
- `ListTopics`
- `DescribeLogGroups`
- `GetRestApis`
- `DescribeRepositories`
- `ListFunctions20150331`

In this case, all of them failed because the identity pool's authenticated role did not grant the required permissions.

## Hunting for ARTEX activity

In this post, we've shared some in-the-wild sightings of ARTEX-attributed activity in cloud environments. It's important to remember that while ARTEX is an offensive AI-native orchestration framework, distributions we analyzed don't appear to have an opinionated methodology for targeting or performing post-exploitation in cloud environments: the playbook is therefore left to downstream AI inference, and can vary greatly depending on the model in use.

In this context, research such as "I Gave 14 LLMs AWS Keys. Here's What They Wanted to Do Next" by Adan Alvarez is valuable for gaining a broader understanding of how LLMs tend to behave when faced with compromised credentials.

When hunting for ARTEX activity in your environment, we recommend following the methodology below.

1. Start with quick wins, namely ARTEX user agents (see IOCs), to identify noisy ARTEX probing or exploitation requests. Note that user agents can easily be changed, but they aren't always updated to a stealthier value.
2. Review your environment for activity originating from IP addresses known to be associated with ARTEX activity (see IOCs).
3. We've seen ARTEX behave noisily in a number of cases: identify large volumes of permission denied errors, enumeration of objects in S3 logs, unusual access patterns (for instance, credentials vended from an EC2 instance role being used by a non-AWS IP), and CloudTrail events that relate to known cloud attack techniques (Stratus Red Team, Datadog's Cloud Security Atlas and TrailDiscover are useful resources for these).

## How Datadog can help

### Cloud SIEM

Datadog Cloud SIEM contains several out-of-the-box (OOTB) rules that can detect ARTEX-related post-exploitation and initial access in your AWS environment:

- A user received an anomalous number of AccessDenied errors
- Enumeration from multiple regions using a long-term access key failed

- Anomalous number of S3 buckets accessed
- Anomalous S3 bucket activity from user ARN
- AWS ListResources by long term access key
- AWS ListResources executed by new principal identity
- AWS Cloudtrail possible secret enumeration in multiple regions and secret retrieval
- Temporary AWS security credentials generated for user
- Unfamiliar IAM user retrieved secret from AWS Secrets Manager
- Tor client IP address identified within AWS environment

### Cloud Security Misconfigurations

Datadog Cloud Security Misconfigurations ships with out-of-the-box (OOTB) rules to help you identify risky long-lived AWS credentials proactively:

- Access keys should be rotated every 90 days or less
- IAM access keys that are inactive and older than 1 year should be removed
- Unused credentials should be deactivated or removed

## Appendix

### Indicators

| IOC | IOC Type | ASN |
|---|---|---|
| 169.128.55[.]214 | IP | Caelus Technology LLC - Hosting Provider |
| 154.64.252[.]77 | IP | NetLab Global - Hosting Provider |
| 154.36.136[.]197 | IP | NetLab Global - Hosting Provider |
| 154.36.136[.]23 | IP | NetLab Global - Hosting Provider |
| 102.216.154[.]102 | IP | Vilcom Networks Limited - ISP in Kenya |
| 82.158.88[.]233 | IP | Zenixcloud - Hosting Provider |
| 88.229.22[.]232 | IP | TTNet A.S. (Turk Telekom) - ISP in Turkey |
| 103.54.154[.]40 | IP | Zeofast Network - Hosting Provider |
| 103.54.154[.]32 | IP | Zeofast Network - Hosting Provider |
| 45.145.228[.]25 | IP | Yecaoyun[.]com, LUCIDACLOUD LIMITED - Hosting Provider |
| artex-readonly/1.0 | User agent | |

| IOC | IOC Type | ASN |
|---|---|---|
| artex-pentest/1.0 | User agent | |
| artex-audit/1.0 | User agent | |
| Mozilla/5.0 ARTeX | User agent | |
| artex-spawn/1 | User agent | |
| artex-verify/1.0 | User agent | |
| Mozilla/5.0 (Windows NT 10.0; Win64; x64) ARTeX-authorized-pentest | User agent | |
| Mozilla/5.0 (X11; Linux x86_64) artex-verify/1.0 | User agent | |
| artex-i21861-canary | User agent | |
| ARTEX-worker/1.0 | User agent | |

---

Source: https://securitylabs.datadoghq.com/articles/artex-agentic-attacks-aws-environments/