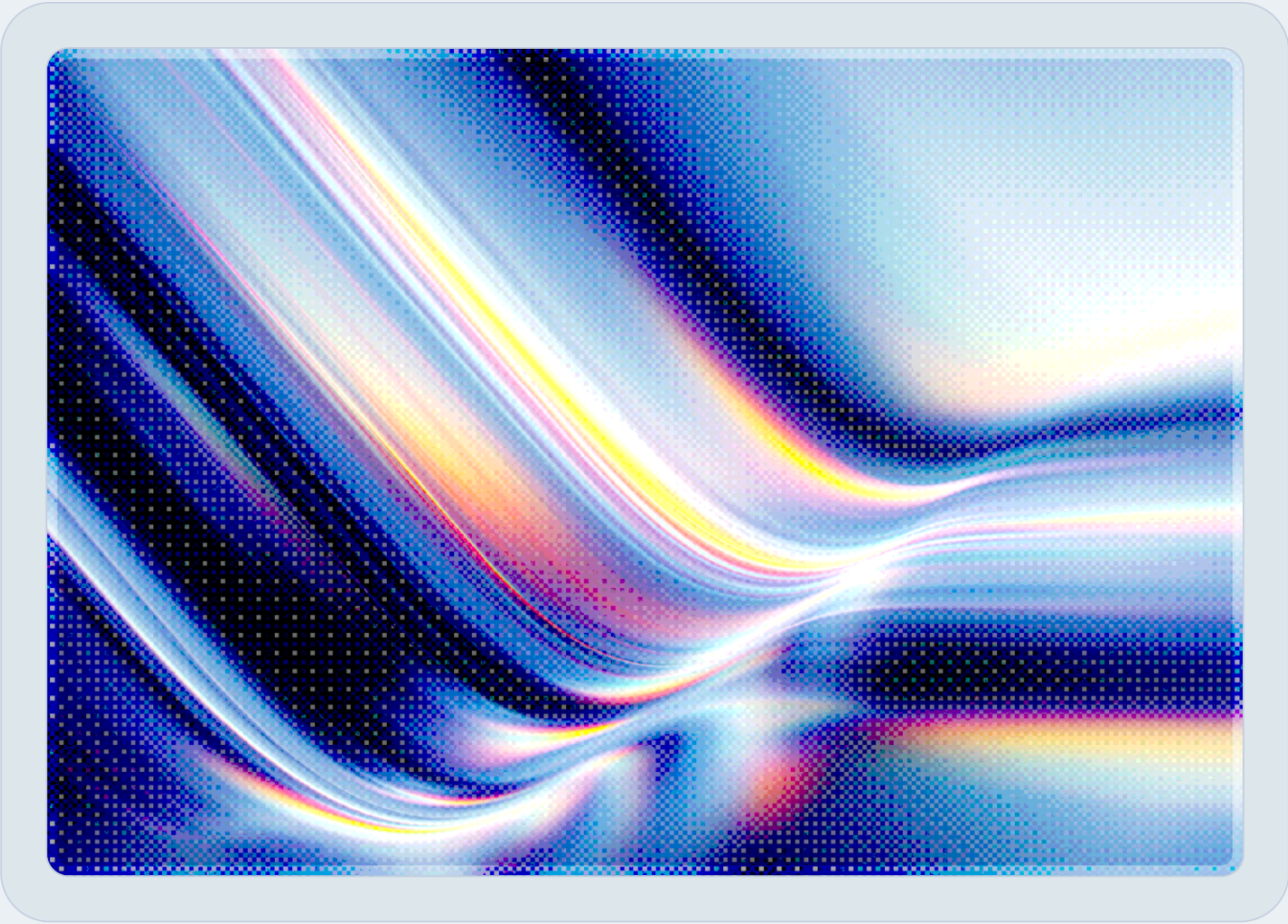
CSAI Foundation | Cloud Security Alliance

AntiFox in the Wild: Forkable Agentic Frameworks Reach AWS Secrets

Security implications of open-source agentic pentesting platforms used against IIS servers, Secrets Manager, S3, and Amazon Bedrock

2026-11-12

AI-assisted Rapid Research



CSAI

CSA 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

Arctic Wolf Labs reported on November 10, 2026, that self-hosted instances of AntiFox, an open-source agentic penetration testing platform, were linked to confirmed malicious activity. That activity included compromise of internet-facing IIS applications and abuse of valid AWS credentials [1]. In one intrusion, the operator cycled through several AWS identities and issued hundreds of API calls across more than a dozen services in about 30 minutes. Most calls were denied, but a `GetSecretValue` request against a production secret in AWS Secrets Manager succeeded [1][2]. The AntiFox README states that the project is secondary development of ARTEX [3]. ARTEX is the agentic framework Microsoft tied one week earlier to automated harvesting of secrets from GitHub [4]. The case therefore shows that a published offensive agent can be forked, rebranded, and redeployed by third parties, and the AntiFox repository predates Microsoft's report by almost four weeks.

The first lesson for defenders is that a burst of denied requests should be investigated for successful calls before it is dismissed. The value of an agentic or scripted enumeration sweep lies in the few calls that succeed, so a sequence should not be discounted because most calls returned `AccessDenied` [1][5]. The second lesson is that long-lived credentials remain the core weakness. In the IIS intrusions the operator harvested `web.config` connection strings, while in the AWS intrusions it used valid IAM credentials of undetermined origin. Prior reporting on related tooling shows that agents can exploit exposed secrets faster than scheduled rotation can respond [1][4][6].

Two further points shape the response. Amazon Bedrock appeared in the enumeration scope, but no model invocation has been reported. Its presence still warrants monitoring, because stolen cloud credentials have previously been used to run AI workloads at the victim's expense [1][7]. Finally, the activity left actionable fingerprints. These include Boto3/Botocore 1.43.125 user agents from unfamiliar hosting networks, one source IP rotating across several identities, cross-service List/Get/Describe storms, and `w3wp.exe` spawning shells, together with the published IP addresses and the TCP/8787 console [1]. These signals are most reliable in combination, since individual elements also occur in legitimate automation.

Background

AntiFox describes itself as an "AI autonomous penetration testing platform." Given a target, a multi-agent system made up of a planner, workers, and a main agent runs reconnaissance, attack-surface discovery, exploit verification, finding capture, and report generation [3]. The project is written as a Go backend with an embedded Next.js front end and PostgreSQL storage. It keeps a global asset graph alongside a per-task exploration graph. It supports MCP and skill extensions and a pgvector knowledge base, and it accepts Anthropic or OpenAI API keys as its reasoning backend [3]. The repository is licensed under AGPL-3.0 and serves its console on port 8787 by default. Its disclaimer restricts use to personal learning and local verification and prohibits attacks on online systems [3]. When The Hacker News covered the project on November 11, the repository had 474 stars and 251 forks [2]. By November 12, those counts had risen to 579 stars and 323 forks [3].

The framework's lineage matters. The AntiFox README says the project began as secondary development of ARTEX starting from ARTEX v0.1.1 [3]. The AntiFox repository was created on October 9, 2026 [3]. On November 5, 2026, Microsoft Threat Intelligence reported that threat actors were using ARTEX to search public GitHub repositories for exposed secrets and to abuse them against cloud services. In one case, an agent went from a GitHub search to Azure Key Vault data in about five minutes [4][6]. Arctic Wolf's own advisory does not mention ARTEX and describes AntiFox only as a Chinese-language, self-hosted platform [1]. The ARTEX link therefore rests on the maintainer's own statement, not on attribution by the incident responders.

Independent testing helps explain why frameworks in this family are effective against exposed credentials. GitGuardian, which analyzed about 45 agentic attack frameworks, found that only four shipped a working integrated secrets scanner. In a controlled test, however, an ARTEX build with no scanner found eight of nine planted credentials using ordinary `git` and `grep` commands. It began searching for secrets within 6 to 13 seconds, even when asked only to assess a repository's security posture [5].

Arctic Wolf observed attacker activity coming from Alibaba Cloud hosts that exposed AntiFox interfaces. The interface title was "AntiFox – 自主渗透测试控制台" ("autonomous penetration testing console"), and two source addresses were published: 47.79.4.33 and 8.208.101.210 [1]. The advisory does not name a threat actor, and none of the reporting reviewed for this note attributes the activity to a specific group. AWS stated that its services "were not compromised and operated as designed." It said the activity involved credentials taken from customers' compromised applications [8]. Nothing in the available reporting contradicts that statement, which places the defensive burden on customer-side identity, secrets, and logging practices.

Security Analysis

The Observed Attack Chain

Arctic Wolf documented two entry paths that converge on the same cloud objective. In the first, an internet-facing IIS application was compromised through a vulnerability that has not been identified. The IIS worker process `w3wp.exe` then wrote an ASPX web shell, with filenames matching the pattern `pwn*.aspx`, and ran operating system commands through that same process [1]. The operator ran `whoami`, enumerated the application directory, and checked permissions with `icacls` to confirm the web root was writable. Next, the operator used PowerShell and .NET configuration classes to parse `web.config`, extracting connection strings and application settings. Those credentials were used in an attempted login to an internal production database [1][2]. Configuration and log files were copied into a web-accessible upload folder under names such as `cfg*.txt`, `wc*.txt`, and `el*.txt` for retrieval [1].

In the second path, the operator started with a valid AWS credential obtained by undetermined means. Some secondary analysts speculate that public repositories or earlier breaches were the source, but that has not been confirmed [1][9]. Arctic Wolf suggests correlating web-shell activity with later AWS use in case a compromised web application stored AWS credentials, but that is a hunting hypothesis rather than an observed link between the two paths [1].

Once inside AWS, the activity was broad, fast, and low-cost. That profile is consistent with the agentic tooling exposed on the source hosts, although the reporting does not separate agent-issued calls from operator-run scripts [1]. The operator switched among IAM users and at least one assumed EC2 role. It issued List, Get, and Describe calls across IAM, EC2, S3, Lambda, Secrets Manager, KMS, RDS, ECS, EKS, Redshift, Athena, Glue, Systems Manager, API Gateway, CloudWatch, CloudFormation, DynamoDB, and Amazon Bedrock [1][2]. The table below summarizes the API activity Arctic Wolf reported, grouped as in the source; `ListKeys` is a KMS API but appears under identity discovery in Arctic Wolf's reporting, and the ECR request was reported separately from the 18 enumerated services.

Objective	Reported API calls	Outcome reported
Identity and permission discovery	`GetCallerIdentity`, `GetUser`, `ListUsers`, `ListRoles`, `ListPolicies`, `ListKeys`, ECR `GetAuthorizationToken` (attempted)	Mostly denied; some discovery succeeded

Objective	Reported API calls	Outcome reported
Secret access	`ListSecrets` → `GetSecretValue`	Production secret retrieved in at least one case
Storage access and tampering	`ListBuckets`, `ListObjects`, `ListObjectVersions`, `HeadObject`, `GetObject`, `CopyObject`, `DeleteObject`	Object operations performed by a valid identity in a separate case
Broad service enumeration	List/Get/Describe across 18 services, including Bedrock	Predominantly `AccessDenied`

Source: Arctic Wolf Labs [1]; The Hacker News [2].

Why the Agentic Pattern Changes the Detection Problem

Conventional alerting tends to treat a high volume of `AccessDenied` responses as noise from a misconfigured script or a failed attack. The AntiFox intrusions illustrate why that assumption is risky. Whether the calls were issued by an agent or by a script, an operator who can cheaply try hundreds of calls against several identities needs only one overly broad policy to reach a production secret. The surrounding denials are a side effect of the search strategy, not evidence that it failed [1][5]. This matches the behavior described in Microsoft's ARTEX reporting, where agents checked what each key could reach and corrected and retried failed calls within roughly 28 to 80 seconds [4][6]. It also matches the observation in prior CSA analysis that systematic enumeration and small-variation retries are among the clearest behavioral signs of a goal-seeking agent [10].

The practical consequence is a much shorter window for defenders. In the reported cases, discovery that would typically take a human operator considerably longer was completed in roughly 30 minutes or less [1][4]. That leaves little room for manual triage. Scheduled rotation remains useful for limiting the lifetime of exposures that have not yet been discovered, but it cannot contain an intrusion already in progress.

The S3 activity shows that the impact goes beyond confidentiality. The reported sequence of `CopyObject` followed by `DeleteObject` is consistent with staging or exfiltration followed by removal [1]. Because `CopyObject` operates within S3, whether the copy amounts to exfiltration depends on whether the destination bucket is attacker-controlled. Arctic Wolf did not describe the

affected buckets or the operator's intent, so destructive or extortion motives remain an inference rather than a finding. Without S3 data events in CloudTrail, versioning, or Object Lock, an organization may be unable to reconstruct which objects were read or removed [1][11].

The Amazon Bedrock Dimension

Bedrock's place in the enumeration list deserves careful wording. None of the primary or secondary sources reviewed reports a Bedrock model invocation, a configuration change, or any other Bedrock call beyond enumeration [1][2][8]. The defensible conclusion is that the tooling treated Bedrock as one more service whose access it should test. Even so, Bedrock is a meaningful target for two reasons. First, stolen AWS credentials with Bedrock invocation rights have been abused since at least 2024 to run large language models at the victim's expense. Sysdig documented this "LLMjacking" pattern after credentials were stolen through a vulnerable web application [7]. Second, CSA's prior analysis found that stolen AI compute has shifted from resale toward serving as the reasoning engine for offensive automation [12].

An agentic framework that needs an LLM backend to operate, as AntiFox does [3], has a clear incentive to find and use the victim's own model access. Doing so lowers its operating cost and removes one point where the operator could be attributed. This is a plausible escalation path, not an observed one. It does make Bedrock invocation logging and least-privilege policies for `bedrock:InvokeModel` relevant to this threat [13].

Forkability as a Threat Multiplier

AntiFox illustrates a structural problem that goes beyond any single tool. A published, AGPL-licensed offensive agent framework can be forked, localized, and redeployed by third parties on commodity cloud hosting [1][3]. The AntiFox repository was created on October 9, 2026, almost four weeks before Microsoft's ARTEX report [3][4]. Defenders therefore should not expect public disclosure of a parent framework to precede in-the-wild use of its derivatives. The maintainer's disclaimer does not control what self-hosted deployments do, and fork-and-rebrand cycles weaken defenses that depend on tool names or code signatures.

The more durable detection surface is behavioral. Cloud API telemetry, endpoint process lineage, and infrastructure characteristics persist regardless of what a tool is called. GitGuardian's finding that agents without a dedicated secrets scanner still locate credentials with general-purpose tools points the same way [5]. Defenders should assume capability parity across forks and focus on the credential and logging conditions the agents exploit.

Recommendations

Immediate Actions

Organizations running workloads on AWS should first hunt for the published indicators. These are API activity from 47.79.4.33 or 8.208.101.210, Boto3/Botocore 1.43.125 user agents from unfamiliar hosting networks, and any external exposure of an "AntiFox" console on TCP/8787 [1]. Next, query CloudTrail for successful `GetSecretValue`, `GetObject`, `CopyObject`, and `DeleteObject` calls that occur inside bursts of denied requests. The sequences `ListSecrets` → `GetSecretValue` and `GetCallerIdentity` → `ListBuckets` from a new source deserve priority review [1][5][14].

On Windows web estates, alert on `w3wp.exe` spawning `cmd.exe`, `powershell.exe`, `whoami.exe`, or `icacls.exe`. Also look for new `.aspx` files, particularly `pwn*.aspx`, written by the worker process, and for PowerShell that references `WebConfigurationManager`, `ConnectionStrings`, or `AppSettings` [1]. Where any match is found, revoke active sessions and access keys for the affected identities. Review their full CloudTrail history before issuing replacements, and rotate every credential exposed in configuration files, logs, backups, or environment variables [1][5].

Short-Term Mitigations

Over the following weeks, the priority is to remove the kinds of static credentials this activity relied on. Replace long-term IAM user access keys in application configuration with role-based temporary credentials [8][15]. Require IMDSv2 with a restrictive hop limit on EC2 to reduce credential theft from compromised hosts [1][16]. Scope application identities so that one compromised workload cannot list every secret or bucket in an account [1]. Enable CloudTrail S3 data events, S3 versioning, and Object Lock on critical buckets so that object-level tampering can be reconstructed and reversed [1][11].

Detection coverage should be extended in parallel. Turn on Amazon GuardDuty and review its IAM credential-access, discovery, and EC2 credential-exfiltration findings [8][17]. For Bedrock, enable model invocation logging and restrict `bedrock:InvokeModel` to the specific roles that need it. These steps make misuse of model access more visible and limit which compromised identities can invoke models [13]. Finally, add detections for a single source IP using several keys, users, or roles within a short window, and for rapid List/Get/Describe sequences that span unrelated services [1].

© 2026 Cloud Security Alliance. AI-assisted Rapid Research.

Strategic Considerations

The AntiFox case supports a broader shift in how organizations model their exposure to agentic adversaries. CSA recommends that detection and response operate at machine speed. CSA's agent-operated intrusion guidance calls for automated containment for high-confidence, multi-signal detections and prompt revocation of keys and sessions [10]. Rotation policies built around 30- to 90-day cycles do not contain an exposed secret that an agent can exploit in minutes. Event-driven revocation and federated, short-lived workload identities should be the target state for production systems [6].

Security teams should also assume that any offensive framework drawing public attention may already exist, or will reappear, under new names. Detection content should therefore be anchored to behavior and telemetry, not to tool signatures. Finally, the IIS intrusions began with an unidentified vulnerability in an internet-facing application. Arctic Wolf recommends prioritizing upload, file-conversion, and document-processing services that accept user-controlled content for patching and WAF coverage [1]. The speed of post-foothold activity raises the cost of every unpatched or unmonitored web tier.

CSA Resource Alignment

CSA's research note *Five Minutes from Search to Key Vault: Agentic Secret Harvesting* (November 2026) analyzed the ARTEX activity reported by Microsoft and is the most directly relevant prior work [6]. Its central argument is that agentic exploitation outpaces scheduled rotation, so exposed secrets must be treated as compromised and revoked by event-driven automation. AntiFox extends that concern from public repositories to secrets held inside compromised web servers and cloud accounts. If the maintainer's account of its ARTEX origins is accurate, it indicates that tooling from the same lineage is being used against AWS environments as well as the Azure case Microsoft described.

Fingerprinting Agent-Operated Intrusions: Detection and Response Guidance (October 2026) supplies the detection model this note applies [10]. Its four-layer approach covers timing and tempo, behavior and decision structure, on-host artifacts, and infrastructure and model dependency. That approach maps closely onto the Arctic Wolf indicators: a 30-minute API storm, systematic cross-service enumeration, web shell and configuration-harvest artifacts, and Alibaba-hosted consoles for a framework that depends on an LLM backend. The note's warning that fingerprints alone should not support attribution applies here as well, because legitimate automation can produce similar CloudTrail patterns.

LLMjacking Evolves: Stolen AI Compute as Attack Infrastructure (June 2026) frames the Bedrock dimension [12]. It documents how stolen model access moved from a resale commodity to a reasoning engine for offensive pipelines. That is the escalation path that makes Bedrock enumeration by an LLM-

dependent framework worth monitoring, even though no invocation has been reported in the AntiFox incidents.

For control implementation, the CSA AI Controls Matrix (AICM) v1.1 provides the relevant control objectives [18]. Its Identity and Access Management domain covers least privilege and credential lifecycle for workload identities. Its Logging and Monitoring and Threat and Vulnerability Management domains cover the CloudTrail, GuardDuty, and web-tier patching practices recommended above. CSA's *Defining Non-Human Identity* offers useful vocabulary for the IAM users, assumed roles, and application credentials that the agents abused [19].

References

[1] Arctic Wolf Labs. "2026-11-10: Antifox Agentic Pentesting Platform Used to Steal AWS Credentials and Move Laterally in Compromised Environments." Arctic Wolf wolf-tools (GitHub), November 10, 2026.

[2] Lakshmanan, Ravie. "AntiFox AI Pentesting Platform Linked to Attacks That Accessed AWS Production Secrets." The Hacker News, November 11, 2026.

[3] AntiFox-AI. "AntiFox: AI Autonomous Penetration Testing Platform (README)." GitHub, accessed November 12, 2026.

[4] Microsoft Threat Intelligence. "Threat actors use agentic AI tooling to find and abuse exposed secrets." Microsoft Security Blog, November 5, 2026.

[5] McDaniel, Dwayne. "AntiFox Targeting AWS Credentials: The Latest Wave of AI Agent Attacks." GitGuardian Blog, November 11, 2026.

[6] Cloud Security Alliance AI Safety Initiative. "Five Minutes from Search to Key Vault: Agentic Secret Harvesting." CSA Labs, November 9, 2026.

[7] Sysdig Threat Research Team. "LLMjacking: Stolen Cloud Credentials Used in New AI Attack." Sysdig, May 2024.

[8] Zwets, Berry. "Attackers use AntiFox AI penetration testing tool against AWS environments." Techzine Global, November 11, 2026.

[9] Aviatrix Threat Research Center. "AntiFox AI Pentesting Platform Exploited to Steal AWS Production Secrets (2026)." Aviatrix, November 11, 2026.

[10] Cloud Security Alliance AI Safety Initiative. "Fingerprinting Agent-Operated Intrusions: Detection and Response Guidance." CSA Labs, October 22, 2026.

[11] Amazon Web Services. "Locking objects with Object Lock." Amazon S3 User Guide, accessed November 12, 2026.

[12] Cloud Security Alliance AI Safety Initiative. "LLMjacking Evolves: Stolen AI Compute as Attack Infrastructure." CSA Labs, June 18, 2026.

[13] Amazon Web Services. "Monitor model invocation using CloudWatch Logs and Amazon S3." Amazon Bedrock User Guide, accessed November 12, 2026.

© 2026 Cloud Security Alliance. AI-assisted Rapid Research.

[14] Amazon Web Services. "Log AWS Secrets Manager events with AWS CloudTrail." AWS Secrets Manager User Guide, accessed November 12, 2026.

[15] Amazon Web Services. "Temporary security credentials in IAM." AWS IAM User Guide, accessed November 12, 2026.

[16] Amazon Web Services. "Configure the instance metadata service options." Amazon EC2 User Guide, accessed November 12, 2026.

[17] Amazon Web Services. "GuardDuty IAM finding types." Amazon GuardDuty User Guide, accessed November 12, 2026.

[18] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, accessed November 12, 2026.

[19] Cloud Security Alliance. "Defining Non-Human Identity." Cloud Security Alliance, accessed November 12, 2026.