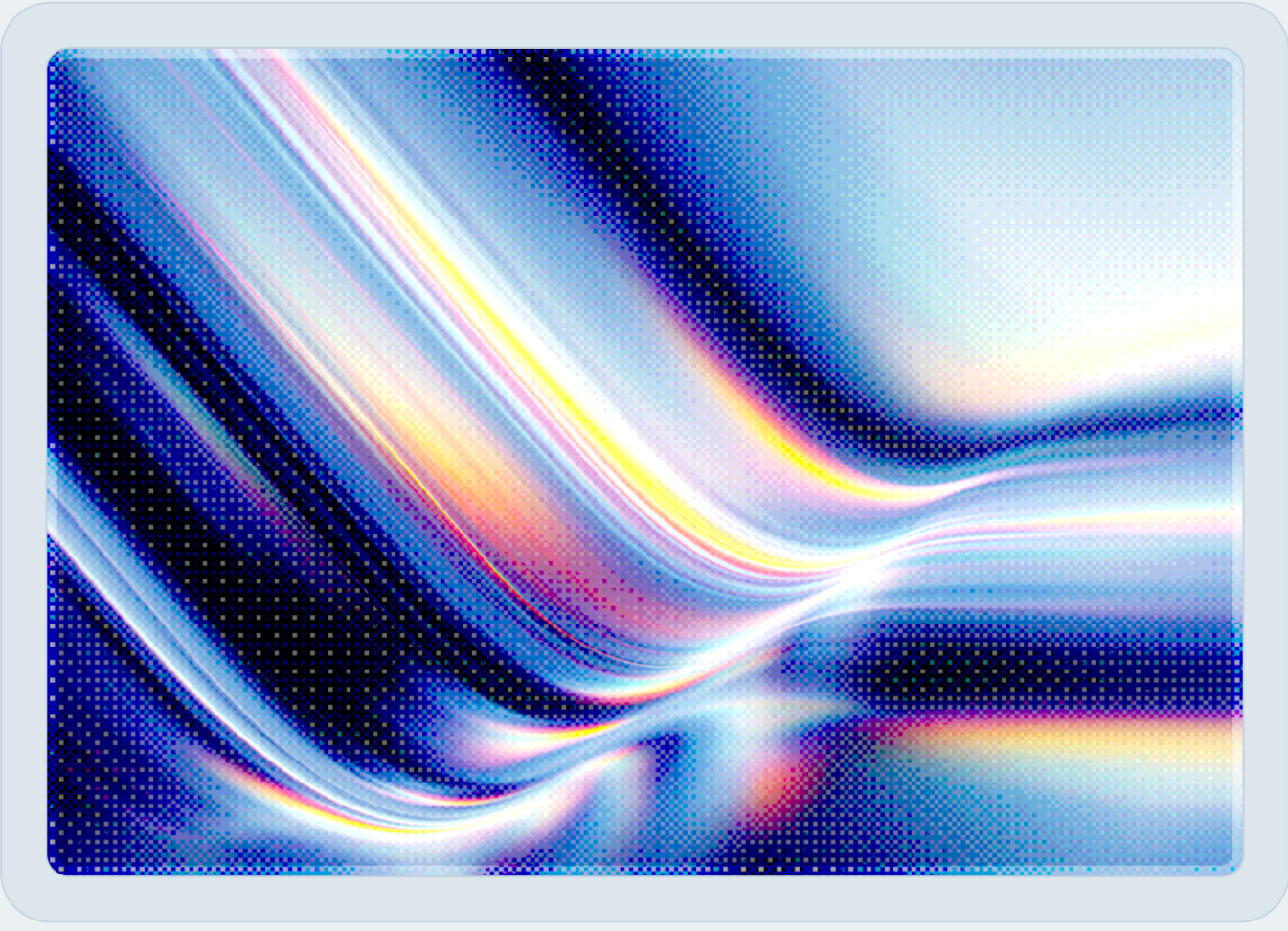
CSAI Foundation | Cloud Security Alliance

ARTEX-Class Agentic Intrusions Against Financial APIs: Security Implications and Guidance

Defending payment, disbursement, and data APIs against autonomous attack frameworks

2026-11-06

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

Over six weeks in late 2026, defenders saw open-source agentic penetration testing frameworks used in live campaigns against financial institutions in South Korea and India. Microsoft also reported such a framework in use by a China-aligned actor against hundreds of organizations, including financial services firms. ARTEX is the most widely reported of these frameworks. This note examines what those campaigns mean for the application programming interfaces (APIs) that financial institutions expose to customers, brokers, partners, and their own staff. Five points stand out.

The attack surface is the periphery. The publicly reported ARTEX intrusions against South Korean banks entered through peripheral, internet-facing services: a broker-facing loan inquiry service and an employee mobile work-support system. Core banking platforms were not the entry point [1][2].

Fraud now runs through the victim's own APIs. India's CERT-In reports that CRAFTY FALCON, an AI-assisted actor, calls victims' own payment and disbursement APIs to make unauthorized transfers. The actor tests these paths with micro-value transactions before moving larger sums [5][6]. CERT-In lists ARTEX among the actor's tools but does not say which tools were used at which stage of the fund-transfer chain.

Machine-speed operation compresses the remediation window. CERT-In reports that entities which had received restricted advisories and remediated were deliberately targeted again, in some cases after incomplete remediation [5]. Microsoft reports that some ARTEX-driven intrusions reached privileged cloud access in less than three hours [4].

Indicator blocking alone is insufficient. The attack infrastructure rotates IP addresses within hours to days [5]. As of November 4, 2026, scans found between roughly 1,465 and 1,645 internet-exposed ARTEX consoles, and Censys estimates about 815 of them are distinct deployments [7][8]. A backdoored ARTEX image on Docker Hub has been downloaded about 9,000 times [7]. The tooling is commoditized and widely dispersed.

Business-logic controls matter as much as perimeter controls. Of the defenses CERT-In recommends, CSA considers the following most relevant to API abuse: server-side transaction validation, signed API requests, app attestation, new-beneficiary cooling periods, and maker-checker approval for settlements. All of them operate in the transaction layer rather than at the network edge [5].

Background

ARTEX is an LLM-driven offensive security platform. A developer using the handle "Autumn-27" published it on July 26, 2026, under the AGPL-3.0 license [8]. Microsoft describes it as a multi-agent system written in Go with a Next.js frontend. It can use any AI model, supports five languages, and runs penetration tests from reconnaissance through data exfiltration with little human involvement [4]. Zscaler ThreatLabz describes the internal design. A planner splits an engagement into tasks, worker agents run tools and record findings, and a reporting component compiles the validated results. The model chooses each next action, and each tool's output feeds back into its next decision [2]. Zscaler also notes that the default ARTEX blocklist prevents targeting of government and university domains but places no restriction on financial institutions [2]. By October 8, 2026, after its misuse had become public, the developer had made the project closed-source [3]. Forks and copies are still in circulation, and at least one has been weaponized against the people who run it [7].

Three public reporting streams describe the campaigns covered in this note. The first comes from CrowdStrike, which reported on October 7 that an unattributed operator used ARTEX against South Korean financial organizations from late September to early October 2026. CrowdStrike assesses the operator as likely Chinese-speaking and financially motivated [1]. The operator's exposed infrastructure held ARTEX configuration files and Claude Code session histories. These showed DeepSeek v4.1-flash as the primary model backend, probably accessed through an API reseller, with GLM-5.3 and Grok 4.6 used in supplementary sessions [1][3]. Zscaler later estimated that personal data on about 67,000 individuals was exfiltrated, including names, phone numbers, annual income, and loan limits [2]. Press tallies of victims and records vary, and CrowdStrike states that the number of affected organizations is unconfirmed [1]. CSA's earlier research note on this campaign discusses attribution and the open questions in more detail [10].

Microsoft provides the second stream. Its October 21 report attributes ARTEX-driven activity against hundreds of organizations, financial services among them, to Storm-0940, a China-state-aligned actor. That activity included a supply-chain compromise of the ConnectWise Cloud Backup multi-tenant application [4]. The third stream is the one most directly about financial APIs. CERT-In issued advisory CIAD-2026-0058 on November 4 with a Critical rating. It names CRAFTY FALCON as an AI-assisted actor targeting non-banking financial companies, digital lenders, fintechs, payment aggregators, and virtual digital asset providers, and it lists ARTEX, PentAGI, Cairn, and CyberStrikeAI as tools used against affected entities [5]. The advisory also reports wallet-key theft and fund siphoning from cryptocurrency exchanges and virtual digital asset providers [5]. It updates an October 9 alert, which reported that attackers were stealing payment-gateway and partner-bank credentials and then calling fund-transfer APIs from whitelisted infrastructure. In doing so they bypassed KYC, beneficiary verification, and transaction limits [6].

Security Analysis

Why Agentic Frameworks Change the Economics of API Attacks

Most of the techniques used in these campaigns are not new. SQL injection, authentication bypass, credential theft, and abuse of exposed endpoints are long-standing problems, and most map to risks in the OWASP API Security Top 10 [9]. What has changed is how cheaply and persistently attackers can apply them. A planner-worker agent architecture can enumerate an institution's entire external footprint, probe every endpoint it finds, and keep going indefinitely. CERT-In describes this pattern as "continuous non-business-hours operation against a single target" [5]. Microsoft notes that ARTEX agents can attempt several breach paths at once and build a graph of the assets and vulnerabilities they discover [4]. As a result, a peripheral service that a human attacker might have skipped now gets the same scrutiny as a flagship mobile banking API, at low marginal cost to the operator.

This should change how defenders read their own API inventories. The South Korean intrusions reached a broker-facing loan inquiry service and an employee work-support system [1]. Zscaler describes these targets as customer inquiry portals and employee-facing mobile app backends outside core banking infrastructure [2]. Systems like these often hold sensitive customer data, yet they rarely receive the authentication rigor, rate limiting, and monitoring applied to primary transaction channels. CERT-In makes a related observation: internet-exposed UAT, staging, and test environments serve as footholds for pivoting into production [5]. Taken together, the evidence suggests that an agentic adversary searches the whole attack surface. The least-defended API therefore becomes the institution's effective security boundary.

From Data Theft to Direct Fund Movement

The South Korean campaign centered on theft of personal data, which the operator apparently intended to sell through Telegram channels [1][2]. The Indian campaign shows a more direct way to profit, and CERT-In describes a sequence that turns the institution's own transaction machinery against it. CERT-In does not say which tools were used at each stage. The techniques described here may therefore reflect AI-assisted human operation rather than autonomous agent execution. The actor reverse-engineers client applications to build tailored exploits. It then uses modified app packages to make API calls that the server accepts on the assumption that only the legitimate client would send them. It tests the unauthorized flows with transfers as small as one rupee before moving larger amounts through disbursement and merchant settlement APIs to mule accounts [5][7]. It also bypasses IP allow-listing by routing requests through whitelisted third-party fetcher pools, such as content-acceleration and app-review services, that can be induced to fetch arbitrary URLs [5].

This pattern matters because each step can look legitimate on its own. A one-rupee transfer to a new beneficiary is unremarkable. A settlement instruction from an allow-listed IP passes network controls, and a well-formed API call that appears to come from the official app passes schema validation. In CSA's assessment, the underlying weakness is architectural. The server trusts the client, the network location, or the credential instead of independently validating the business transaction. AI assistance plausibly lowers the cost of reverse-engineering an app and producing many tailored script variants, and so makes it easier to find and chain these trust assumptions. CERT-In reports large numbers of tailored scripts, with functionally identical variants refined in rapid succession. It describes this as a pattern consistent with AI-generated code and iterative AI-assisted development [5].

The table below maps the API abuse techniques reported in these campaigns to OWASP API Security Top 10 (2023) risk categories and to the transaction-layer controls that would interrupt them. The OWASP mappings are CSA's analytical inference, not the source advisories', and they are approximate. In particular, injection is no longer a standalone category in the 2023 list, so the injection row maps to the closest current categories.

Reported technique	Source	Related OWASP API risk (2023) [9]	Interrupting control
SQL injection and authentication bypass on peripheral portals	[2]	API2 Broken Authentication; API8 Security Misconfiguration (approximate; injection is not a standalone 2023 category)	MFA on any data-returning function; production-grade hardening for peripheral APIs
High-frequency sequential enumeration of login and search endpoints	[2]	API4 Unrestricted Resource Consumption; API1 Broken Object Level Authorization	Per-session record-access thresholds; server-side authorization; response field minimization
Modified client app packages issuing unauthorized calls	[5]	API6 Unrestricted Access to Sensitive Business Flows	App attestation; signed requests with nonces and timestamps; server-side business rule validation
Micro-value test transfers preceding large transfers	[5]	API6 Unrestricted Access to Sensitive Business Flows	New-beneficiary cooling periods; velocity and

Reported technique	Source	Related OWASP API risk (2023) [9]	Interrupting control
			amount-band anomaly detection
Merchant settlement API abuse	[5]	API5 Broken Function Level Authorization; API6 Unrestricted Access to Sensitive Business Flows	Maker-checker approval; reconciliation of all settlement instructions
IP allow-list bypass via shared fetcher pools	[5]	API7 Server Side Request Forgery (abuse of third-party fetch services); API8 Security Misconfiguration	Mutual TLS, signed requests, or scoped keys in place of IP allow-lists
Pivot from UAT/staging to production via shared secrets	[5]	API9 Improper Inventory Management	Full isolation of non-production; no shared credentials or network paths

Identity and Supply-Chain Amplification

The Storm-0940 activity shows a second route to financial data. Instead of attacking transaction APIs directly, the actor abused trusted machine identities. Microsoft reports that after the actor took control of the identities of several ConnectWise cloud applications, ARTEX-driven agents authenticated to customer tenants using the trust those applications already held. They then ran Microsoft Graph reconnaissance, changed application credentials, and modified privileged roles; in one case, a management application assigned itself Global Administrator [4]. Some of these ARTEX-driven intrusions reached privileged cloud access in less than three hours [4]. Microsoft observed sign-in attempts against more than 2,250 tenants. A related password-spray campaign produced more than 151,000 failed logons against 327 accounts and peaked at 2,900 failures per minute, although Microsoft observed no successful authentication from that spray [4]. Microsoft also cautions that a successful authentication does not by itself mean data was accessed [4].

The lesson for financial institutions is that third-party SaaS integrations with broad API permissions are effectively part of the institution's own API attack surface. Backup, monitoring, and productivity integrations often hold tenant-wide read or administrative scopes and authenticate with long-lived

application credentials. An agentic framework that compromises one such vendor can attempt access across that vendor's customer base faster than vendor and customer incident processes are typically able to coordinate. Microsoft's guidance reflects how difficult scoping becomes after the fact: it advises searching by application ID rather than display name, and preserving evidence before deleting enterprise applications [4].

Detection Under Machine-Speed Conditions

Indicator-based detection is struggling to keep pace. CERT-In reports infrastructure rotating within hours to days and explicitly warns against relying solely on known indicators [5]. Censys counted roughly 1,465 exposed ARTEX consoles, about 815 of which appear to be distinct deployments. That count likely understates the real number, because the tool can be run without public exposure [8]. The Hacker News separately reported 1,645 hosts running the ARTEX console as of November 4, 2026 [7], and the difference probably reflects methodology. Either figure shows wide deployment. Some deployments belong to authorized testers, however, so the presence of a console does not by itself establish malicious ownership [8].

Behavioral signals depend less on specific infrastructure, and the reporting points to several that track the agentic operating model rather than any particular tool. Zscaler highlights high-frequency, sequential enumeration against login and search endpoints, unexpected server-side script files that suggest web shells, and outbound transfers of structured personal data [2]. CERT-In adds bursts of near-identical script variants in web roots and temp directories, attacks timed for holidays and weekends, and interactive use of cloud management services such as AWS Systems Manager and Lambda to run commands [5]. Both Zscaler and CERT-In treat outbound calls from servers to LLM API endpoints as a meaningful signal: agentic tooling running on a compromised or attacker-staged host has to reach a model somewhere [2][5]. The October 9 CERT-In alert also reported log deletion and tampering with endpoint detection tools [6]. That finding argues for shipping API and cloud audit logs off-host in near real time.

Recommendations

Immediate Actions

Financial institutions should begin by building a complete inventory of internet-reachable APIs. The inventory should explicitly cover broker, partner, and employee-facing services, mobile app backends, and every UAT, staging, and test environment. Each entry needs an owner, a data classification, and a record

of which authentication and rate-limiting controls apply. Any API that returns customer financial data or initiates a fund movement should enforce server-side authorization on every object and function, and multi-factor authentication wherever users authenticate. Internet-exposed non-production environments should either be removed from public reach or hardened to production standards, and every credential and token they share with production should be rotated [5].

At the same time, payment and disbursement teams should implement the transaction-layer controls CERT-In names. These include flagging micro-value transfers to new beneficiaries that are followed by larger transfers, cooling periods for new beneficiaries, velocity and amount-band limits on all disbursement paths, and maker-checker approval with reconciliation for merchant settlement instructions [5]. Security operations should extend coverage and lower alert thresholds over weekends and holidays. After any confirmed intrusion, teams should assume persistence and carry out full forensic eradication instead of relying on point fixes [5]. Institutions that use ConnectWise Cloud Backup or similar multi-tenant integrations should follow Microsoft's published hunting and evidence-preservation guidance [4].

Short-Term Mitigations

Over the next one to three months, institutions should remove implicit client and network trust from their transaction APIs. In practice, this means signing requests with nonces and timestamps and using app attestation and certificate pinning to detect modified clients. Where shared third-party infrastructure is involved, IP allow-lists should give way to mutual TLS, signed requests, or scoped API keys [5]. Static, long-lived API tokens for partner banks and payment gateways should be replaced with short-lived credentials bound to specific workloads. Secret scanning should run across repositories and CI/CD pipelines. According to The Hacker News, CERT-In identifies stolen static tokens, plaintext credentials, and forged commits as common initial access paths [7].

Detection engineering should shift toward behavioral analytics that operate at machine speed. Useful sources include API gateway telemetry for sequential enumeration and per-session record-access anomalies, file-integrity monitoring on application deployment directories, cloud audit alerts for remote command execution and serverless code changes, and egress monitoring for server-originated calls to LLM API endpoints and reseller domains [2][5]. Deception assets such as canary credentials and decoy administrative endpoints are especially useful against agentic adversaries, because automated exploration is likely to touch them [5]. Third-party SaaS integrations with tenant-wide API scopes should be reviewed for least privilege, with alerts on credential additions and self-assigned roles [4].

Organizations that use agentic offensive tools for their own testing should obtain them only from verified sources, because unverified forks carry a real risk of compromise [7]. Every such engagement should run under written scope and time limits. These frameworks do not verify that an engagement is authorized; ARTEX's default blocklist covers only certain government and academic domains [2][5].

Strategic Considerations

Over the longer term, these campaigns suggest that the window between an API's exposure and its first adversarial probing is narrowing. Institutions should plan for this by continuously and automatically testing their own API estate from the outside, ideally with the same class of agentic tooling under controlled conditions, so that defenders find peripheral weaknesses before adversaries do. Governance of partner-facing and internal tools should apply the same lifecycle rigor as customer channels. In the South Korean intrusions, attackers reached partner-facing and employee-facing services as readily as customer channels.

Sector-level coordination matters as well. CERT-In shares restricted, TLP-marked indicators and detection rules with BFSI entities [5]. After the South Korean breaches, the Financial Services Commission and Financial Supervisory Service warned consumers about phishing and loan scams that could exploit the leaked data [3]. Both are examples of regulators responding to agentic intrusion activity. Institutions should take part in sector information-sharing bodies and set up abuse-reporting channels with the AI model providers and API resellers whose services appear in attacker infrastructure. They should also contribute to emerging norms for releasing dual-use autonomous offensive tooling, including whether default target blocklists should cover critical financial infrastructure [2].

CSA Resource Alignment

The most directly relevant CSA artifact is the research note *ARTEX: Open-Source Agentic Pentest Tool Turned on Korean Banks* [10], published on October 10, 2026. That note covers the South Korean campaign's attribution, infrastructure, and unresolved questions, and it recommends inventorying non-primary applications and monitoring egress to LLM APIs. The present note extends it in two ways. It incorporates the later Microsoft and CERT-In reporting, and it moves the focus from data exfiltration to the abuse of payment, disbursement, and settlement APIs for direct fund movement.

CSA's whitepaper *Autonomous Agentic AI Adversaries* [11] provides the conceptual framework this note relies on. It distinguishes AI-assisted from AI-autonomous operations and argues that agentic adversaries are already an operational reality. Its emphasis on replacing static API keys and shared service accounts, detecting LLM API calls from infrastructure, and moving beyond human-speed detection models applies directly to the CRAFTY FALCON and Storm-0940 activity described here. Its earlier discussion of CyberStrikeAI is also relevant, because CERT-In now names that framework alongside ARTEX [5].

For sector context, CSA's *The State of Cloud and AI for Financial Services 2026* [12] documents how quickly financial institutions themselves are adopting autonomous agents. CSA reported that 62% of surveyed organizations have already deployed AI agents, while many still lack adequate visibility and

controls [13]. This gap matters because the non-human identity and API governance weaknesses that complicate oversight of an institution's own agents also widen the surface available to adversarial agents.

At the control level, the AI Controls Matrix (AICM) v1.1 [14] provides a structured basis for the recommendations above. Identity and Access Management covers short-lived workload credentials and third-party application scopes. Application and Interface Security covers server-side validation and request signing. Threat and Vulnerability Management addresses rapid patching of internet-facing applications, and Logging and Monitoring supports tamper-resistant off-host audit trails. Teams threat-modeling their own exposure to agentic adversaries can use CSA's MAESTRO framework [15] to analyze how an autonomous attacker would chain trust assumptions across the layers of a financial API stack.

References

[1] CrowdStrike. "Unknown Threat Actor Uses AI-Driven ARTEX to Target South Korean Finance." CrowdStrike Blog, October 7, 2026.

[2] Hegde, R., and Nair, J. P. "Threat Actor Abuses Agentic AI Pentesting Framework to Target South Korean Financial Institutions." Zscaler ThreatLabz, October 14, 2026.

[3] Lakshmanan, R. "ARTEX AI Pentesting Tool Used in Data Theft Attacks on South Korean Financial Firms." The Hacker News, October 8, 2026.

[4] Microsoft Threat Intelligence. "AI-driven, large-scale nation-state campaign using ARTEX enables supply chain compromise." Microsoft Security Blog, October 21, 2026.

[5] CERT-In. "Escalation of AI-Driven Attacks on BFSI Entities by Threat Actor 'CRAFTY FALCON' (CIAD-2026-0058)." Indian Computer Emergency Response Team, November 4, 2026 (accessed November 6, 2026).

[6] MediaNama. "CERT-In flags an AI threat actor targeting BFSI payment APIs for unauthorized fund transfers." MediaNama, November 5, 2026.

[7] Lakshmanan, R. "CERT-In Flags AI-Driven Attacks as Backdoored ARTEX Exposes 62 Consoles to Root Access." The Hacker News, November 5, 2026.

[8] Kathir, M. "ARTEX AI Framework Linked to Bank Breaches and Cloud Attacks Found Exposed Across Hundreds of Servers." GBHackers, November 6, 2026.

[9] OWASP Foundation. "OWASP Top 10 API Security Risks – 2023." OWASP API Security Project, 2023.

[10] Cloud Security Alliance AI Safety Initiative. "ARTEX: Open-Source Agentic Pentest Tool Turned on Korean Banks." CSA Labs, October 10, 2026.

[11] Cloud Security Alliance AI Safety Initiative. "Autonomous Agentic AI Adversaries: Frontier Models Cross the Threshold from Tool to Actor." CSA Labs, June 24, 2026.

[12] Cloud Security Alliance. "The State of Cloud and AI for Financial Services 2026." Cloud Security Alliance, June 2026.

[13] Cloud Security Alliance. "Financial Services Industry Shifts from AI Adoption to Governance as Autonomous Systems Proliferate, Cloud Security Alliance Survey Finds." Cloud Security Alliance Press Release, June 9, 2026.

[14] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, June 22, 2026.

[15] Huang, K. "Agentic AI Threat Modeling Framework: MAESTRO." Cloud Security Alliance Blog, February 6, 2025.