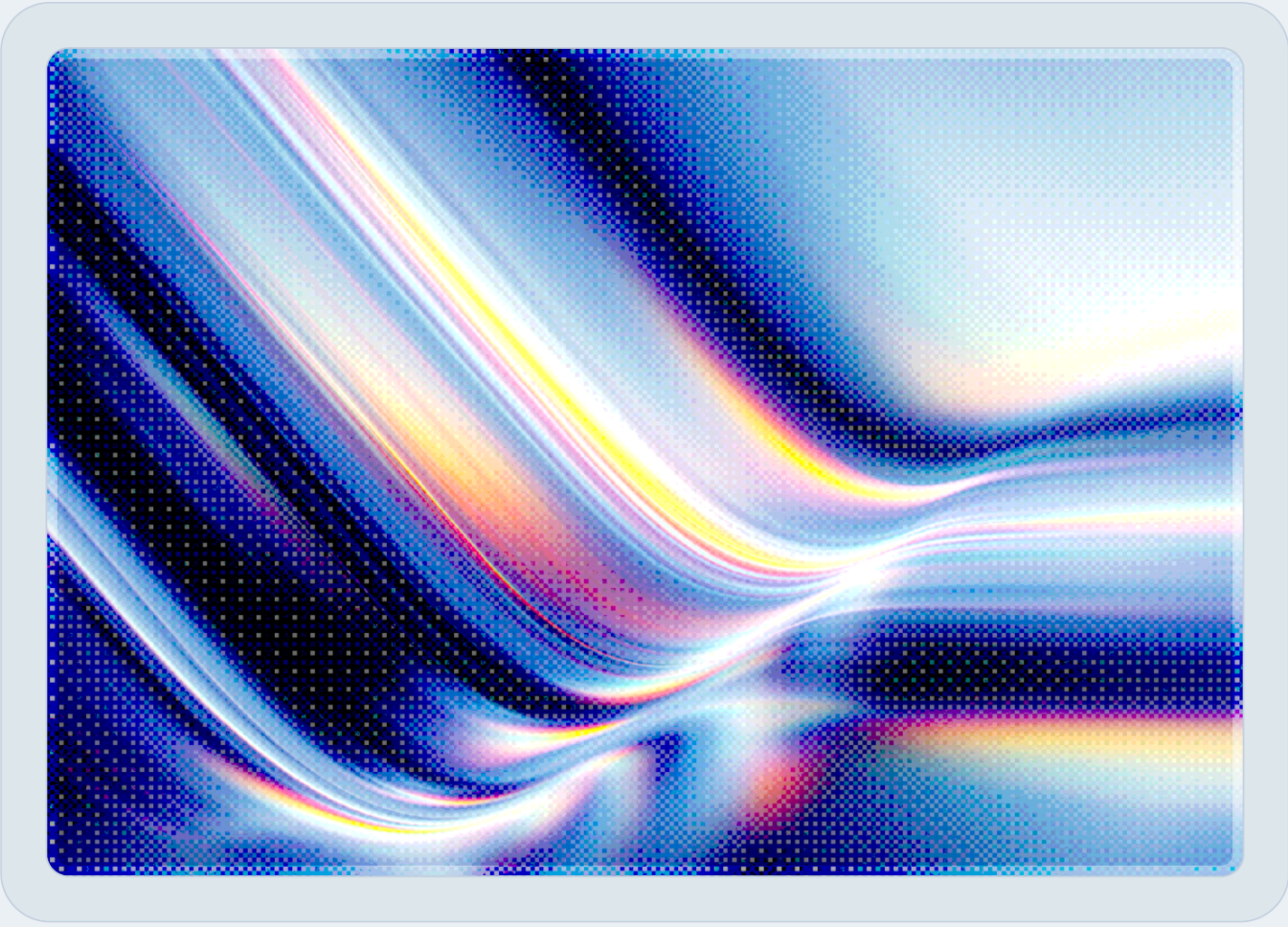
CSAI Foundation | Cloud Security Alliance

Trusted by Default: Qubole Cross-Account Role Abuse

The Dormant Vendor-Trust Problem in Cloud Data and AI Platforms

2026-11-10

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

Between October 14 and October 26, 2026, a threat actor used IAM credentials belonging to the AWS account that the Qubole data lake platform uses to reach its customers, and assumed the cross-account integration roles those customers had created for Qubole [1]. Wiz, working with Sentra and Mandiant, reported that every role assumption attributed to the actor included a valid External ID and succeeded, and that most of the roles assumed were otherwise unused, with the attacker's requests being the only vendor role assumptions seen in a majority of affected environments that month [1]. Once inside, the actor enumerated storage, compute, database, and key-management resources, harvested secrets from AWS Secrets Manager, SSM Parameter Store, and Lambda environment variables, and downloaded S3 contents, in some cases whole buckets [1][2]. The harvested material included AWS access keys, Microsoft Entra ID service principal secrets, and Okta OAuth client credentials, which the actor validated with the open-source TruffleHog tool [1][2].

The central lesson is that the trust relationship itself was the vulnerability. Wiz's analysis identifies no AWS vulnerability in the attack chain; customer roles behaved exactly as their trust policies specified once the vendor's own credentials were compromised [1]. The External ID offered no protection in this scenario, because AWS states that it does not treat the External ID as a secret [4]. That control prevents confused-deputy misuse by *other* AWS accounts, but it does not stop someone who holds the vendor's own credentials. The incident also shows that dormant integrations carry live risk: roles created for a platform that was no longer actively used still granted read access to secrets and data [1][3].

Two further findings shape the defensive response. According to Wiz, a single network condition would likely have helped, since an `aws:SourceIp` condition on the trust policy alone would have blocked every assumption it observed [1]. Logging gaps, meanwhile, hide exfiltration. CloudTrail management events record role assumption and bucket listing but not object reads, so S3 data events or server access logs are needed to establish what left the account [1].

Organizations that ever integrated Qubole should treat this as a potential compromise and hunt for it immediately. They should review CloudTrail from October 1, 2026 [1], and where suspicious role assumptions are found, revoke sessions and rotate every secret the integration role could read. More broadly, every organization using AWS should inventory all IAM roles that trust external accounts, because this pattern is not specific to Qubole.

Background

Qubole is a cloud-native data lake platform that orchestrates big-data and machine learning workloads such as Spark, Hive, and Presto across customer cloud accounts. Idera, Inc. acquired it in October 2020 [6]. Like many data, analytics, and AI platforms, Qubole operates inside its customers' AWS environments instead of hosting their data itself. It moved away from asking customers for long-lived access keys in favor of IAM roles in 2015, describing roles as a security best practice because customers "no longer need to provide access and secret keys" to the platform [7]. Under this model, the customer creates an IAM role whose trust policy names Qubole's AWS account as a permitted principal and requires a matching External ID on each `sts:AssumeRole` request. Qubole's documentation acknowledges that the resulting role can reach S3 buckets that "may contain sensitive data," and it offers a dual-role pattern to fence off the most sensitive buckets [8].

This architecture is the AWS-recommended way to grant third parties access. AWS guidance directs account owners to specify the third party's account ID as the trust policy principal and an External ID as a condition, and it notes that the External ID exists primarily to address the confused deputy problem [4]. A confused deputy attack occurs when one customer of a multi-tenant vendor tricks the vendor into acting on another customer's resources. The External ID ensures the vendor acts only for the customer whose ID it presents [4][5]. The same AWS documentation is explicit about what the External ID is not: "AWS does not treat the external ID as a secret," and it can be seen by anyone with permission to view the role [4]. The design assumes that the vendor's own account and credentials remain uncompromised. That assumption failed in October 2026.

According to Wiz, the attacker began on October 14 by assuming roles from a Mullvad VPN exit node using one IAM user in Qubole's account (`AIDA3W7DIFLYOLTK63SFS`), shifted most activity to an OVH-hosted virtual server the following day, and from October 18 also used a second IAM user (`AIDAJYRRH5Y3A7AAJM2HW`) that additionally carries legitimate platform traffic [1]. Role session names followed a `quiam-` prefix pattern such as `quiam-enum` and `quiam-scope-full` [1]. Activity ended on October 26. Sentra first identified the suspicious activity across several customer environments and traced stolen files back to roles that trusted Qubole [1][3]. CSO Online reported that Sentra put the number of affected Qubole customers in the dozens, and that Sentra observed more than 100 passwords and keys read within minutes along with thousands of additional credential records in downloaded files [2]. How the attacker obtained Qubole's IAM credentials has not been publicly disclosed, and no public statement from Qubole or Idera describing the root cause or remediation was available as of this writing [1][2][3].

Wiz also documented a separate actor that, between October 8 and 11, tested customer-owned AWS access keys believed to have been created for Qubole integrations, testing all of them within a single hour on October 8 and compromising a few environments [1]. Wiz found no link between that activity and the role-abuse campaign [1]. Wiz further identified a bootable Linux image, `sso_agent.iso`, hosted on the actor's OVH infrastructure. The image is designed to be booted remotely on a victim machine, beacons for shell commands on TCP port 23458, and reboots after a 90-minute timer to destroy evidence. Wiz has not identified any deployment of it [1]. The campaign has not been attributed to a known group, and Wiz characterized the targeting as opportunistic [1][2].

Security Analysis

The incident is best understood not as a failure of any single control but as a demonstration of what the standard cross-account pattern was never designed to do. The subsections below examine why correctly configured controls still failed, how dormancy amplified the exposure, why data and AI platforms concentrate this risk, and which observability gaps limited the response.

Why the Controls Worked as Designed and Still Failed

The incident is instructive precisely because every control functioned as specified. The trust policy correctly restricted role assumption to Qubole's account, and the External ID condition was correctly satisfied. AWS returned temporary credentials scoped to the role's permission policy, as it should. Each element of the cross-account model answers the question "Is this request coming from the vendor on behalf of this customer?" None of them answers "Is this request coming from the vendor's legitimate infrastructure, for a legitimate business purpose, today?" Once an adversary held valid vendor credentials, the system could not distinguish between the vendor and the attacker. The External ID did not change this, because it is a value the vendor stores alongside each customer's role ARN and that is logged in plaintext in CloudTrail [1][4]. How the attacker obtained the External IDs has not been disclosed. The most plausible explanation is access to the vendor's own systems, where those values are stored alongside each customer's role ARN.

Customers may be tempted to treat the External ID as a second authentication factor, but it was designed as an anti-confusion token, and AWS states that it does not treat the value as a secret [4]. The distinction matters for risk assessments. A customer who reasons that the vendor's credentials *plus* a per-customer ID constitute two factors is mistaken. Both values are held by the same party, so a compromise of that party's credential store would likely yield both. The table below summarizes what each layer of the standard pattern actually protects against.

Control	Threat it addresses	Effective against compromised vendor credentials?
Trust policy principal (vendor account `:root`)	Arbitrary AWS accounts assuming the role	No. Any principal in the vendor account qualifies.
`sts:ExternalId` condition	Confused deputy: vendor tricked into acting for the wrong customer	No. The vendor stores the value; AWS does not treat it as secret [4].
Permission policy scope	Overbroad vendor actions	Partially. It limits blast radius but does not prevent access.
Specific principal ARN instead of `:root`	Misuse by other identities in the vendor account	Partially. It helps only if the compromised identity is a different one.
`aws:SourceIp` / `aws:SourceVpce` condition	Use of vendor credentials from unexpected networks	Yes, for this campaign. Wiz states it would have blocked every observed assumption [1].
Anomaly alerting on role assumption	New networks, dormant roles suddenly active	Yes, as a detective control, if logs are collected and monitored.

The network condition deserves a caveat despite its strong showing in this campaign. It depends on the vendor publishing stable egress ranges, and an attacker operating from inside AWS or the vendor's own infrastructure could evade it. The fact that the second IAM user also carried legitimate platform traffic illustrates that the line between legitimate and illegitimate use is not always a network boundary. Source-IP pinning raises the bar, but it does not substitute for least privilege and monitoring.

Dormancy as the Force Multiplier

The most consequential finding may be that most of the assumed roles were not otherwise in use [1][3]. Integration roles are often created during a proof of concept or onboarding, granted broad read access so that the platform "just works," and then forgotten when the organization migrates to another tool or lets the contract lapse. The trust policy has no expiry, and nothing in AWS forces a review. As a result, the role remains a standing invitation to the vendor's account long after anyone in the customer organization

remembers it exists. Dormancy also undermines detection in practice. Although any activity on a long-unused role is itself a strong anomaly signal, these roles are rarely covered by alerting, and their owners are unlikely to be watching them.

This pattern directly parallels the OAuth-token compromises CSA analyzed earlier in 2026. In that series, durable, unmonitored trust granted to SaaS integrations such as Salesloft Drift, Gainsight, and Klue, including a dormant test credential, gave attackers access to downstream customer Salesforce data without exploiting Salesforce itself [9]. The Qubole campaign moves the same failure mode from SaaS-to-SaaS OAuth grants down into infrastructure-level IAM trust. Here the "connected app" is an IAM role, and the "token" is the vendor's ability to call `AssumeRole`.

Why Data and AI Platforms Concentrate the Risk

Data lake, analytics, MLOps, and AI platforms are structurally prone to this pattern because their function requires read access to large volumes of data, and often to credentials for other data sources. Qubole's own documentation acknowledges that its role can reach buckets that "may contain sensitive data" [8], and in this campaign the actor found secrets reachable through Lambda, Secrets Manager, and Parameter Store [1]. The attacker's playbook targeted exactly this combination. After enumeration, the actor went straight to Lambda configuration, Secrets Manager, and Parameter Store, then validated harvested credentials against AWS, Entra ID, and Okta [1]. The theft of identity-provider client secrets is significant because it may extend the compromise beyond AWS into enterprise identity systems, where the blast radius is harder to bound.

As organizations deploy AI workloads, they are adding similar integrations. Model training platforms, feature stores, vector database services, evaluation tools, and agent frameworks commonly request cross-account roles or equivalent delegated access to training data and inference infrastructure. Each such integration inherits the same structural weakness: the customer's security depends on the vendor's credential hygiene, and the customer typically has little visibility into it. CSA assesses that, absent changes to how these trust relationships are governed, the Qubole campaign is likely to be repeated against other platforms that hold standing cross-account access to many customers. That assessment is an inference, not a reported fact.

Detection Gaps

Two observability limitations shaped this incident. First, CloudTrail management events capture the `AssumeRole` call and `ListObjects` activity but not `GetObject` reads. Organizations without S3 data events or server access logs enabled therefore cannot determine which objects left their environment [1]. Second, rotating a compromised credential does not invalidate temporary role sessions

already issued. Responders must explicitly revoke active sessions for the affected roles, which AWS supports through a deny policy keyed on token issue time [1][10]. Both gaps are common in practice and should be addressed before the next incident.

Recommendations

The recommendations that follow are organized by time horizon. Immediate actions focus on organizations with a known Qubole relationship, short-term mitigations apply to any organization that grants external accounts access through IAM roles, and strategic considerations address the governance and platform defaults that allowed dormant trust to persist.

Immediate Actions

Any organization that has ever integrated Qubole should hunt in CloudTrail from October 1, 2026 onward, across every account that contains a role trusting Qubole's AWS account (`805246085872`) [1]. Priority searches are `AssumeRole` events from that account where the source IP is not AWS infrastructure, any activity by principal `AIDA3W7DIFLYOLTK63SFS` , `AssumeRole` by `AIDAJYRRH5Y3A7AAJM2HW` from non-AWS addresses, session names beginning with `quiam-` , and any activity from the indicator IP addresses Wiz published [1]. Because the Mullvad addresses are shared VPN exit nodes, a match on IP alone is not conclusive [1]. Responders should also check identity provider sign-in logs for TruffleHog user agents paired with the indicator IPs, since harvested Entra ID and Okta credentials were tested there [1][2].

Where a suspicious session is found, teams should treat every secret, parameter, and Lambda environment variable that the role could read as compromised and rotate it. They should also revoke active sessions for the role, because credential rotation alone does not end sessions already issued [1][10]. If the integration is no longer in use, the role's trust policy should be removed or the role deleted outright. Organizations that are still active Qubole customers should coordinate with the vendor on External ID rotation and on the published network ranges from which legitimate assumptions originate.

Short-Term Mitigations

Within the next 30 days, every organization using AWS should inventory all IAM roles whose trust policies allow principals outside its AWS Organization. IAM Access Analyzer's external access findings are designed for exactly this purpose [11]. For each role, owners should establish whether the vendor

relationship is still active, when the role was last assumed, and who in the organization owns it. Unowned or unused roles should be removed. Those that remain should be hardened along the lines summarized in the table below, which pairs each measure with its implementation approach.

Hardening measure	Implementation note
Narrow the principal	Replace the vendor's account `:root` with the specific vendor role or user ARN where the vendor supports it [1].
Pin the network	Add `aws:SourceIp` or `aws:SourceVpce` conditions using vendor-published egress ranges [1].
Reduce permissions	Remove Secrets Manager, SSM, Lambda configuration, and account-wide S3 read unless the platform demonstrably needs them [1].
Separate sensitive data	Use patterns such as Qubole's dual-role model so the vendor-assumable role cannot reach the most sensitive buckets [8].
Close logging gaps	Run an organization-wide, all-region CloudTrail trail and enable S3 data events or server access logs on sensitive buckets [1].
Alert on anomalies	Alert on vendor role assumptions from new networks, VPN or hosting providers, and roles with no recent activity [1].

These measures are complementary rather than alternatives. Network pinning and principal narrowing reduce the chance that stolen vendor credentials can be used at all, while permission reduction and data separation limit what an attacker can reach if they are. Logging and alerting ensure that any remaining misuse is visible and can be scoped.

Strategic Considerations

The deeper lesson concerns third-party governance, not IAM syntax. Cross-account roles should be governed as non-human identities with a lifecycle that includes an owner, a business justification, a review date, and a decommissioning trigger tied to contract termination. In this campaign, roles survived after customers stopped using the platform [1][3], indicating that vendor offboarding did not include removing the trust relationship. Procurement and offboarding processes should include it explicitly. Vendor risk assessments should also ask concrete questions. Does the provider use long-lived IAM user keys or short-lived workload credentials to assume customer roles? Does it publish egress ranges so

customers can pin trust policies? Does it monitor its own CloudTrail for cross-account assumptions from unexpected networks? Wiz recommends all three practices for SaaS providers [1], and they are reasonable baseline expectations for any platform holding standing access to customer environments.

Cloud providers and platform vendors also have a role to play. The current model makes the secure configuration opt-in and leaves dormant trust indefinitely valid. Features such as default trust expiry, prompts to recertify unused cross-account roles, and vendor-attested network bindings would shift the default. Until such changes arrive, customers should assume that any role trusting an external account is only as secure as that external party's least-protected credential, and should size permissions accordingly.

CSA Resource Alignment

CSA's research note *ShinyHunters' OAuth Pivot: A Year of SaaS Supply-Chain Breaches* is the closest precedent for this incident [9]. It traced three successive compromises in which attackers reached customer Salesforce data through durable, overlooked OAuth trust held by third-party vendors, and identified dormant credentials and an absence of monitoring for already-authorized access as root causes. The Qubole campaign reproduces that structure at the IAM layer. That note's recommendations on inventorying third-party grants, enforcing least privilege on integrations, and monitoring authorized-but-anomalous sessions apply directly to cross-account AWS roles, and readers should treat the two analyses as companion pieces.

CSA's *Confronting Shadow Access Risks: Considerations for Zero Trust and Artificial Intelligence Deployments* addresses the broader category this incident exemplifies [12]. Shadow access refers to unintended, unmonitored, or forgotten access paths that accumulate across cloud environments, particularly as AI and data pipelines proliferate. A dormant integration role that still trusts a vendor account is a textbook example. The paper's emphasis on continuous verification of access and on applying Zero Trust principles to non-human identities supports the network-pinning and anomaly-alerting measures recommended above. CSA's *Zero Trust Principles and Guidance for Identity and Access Management (IAM)* extends this foundation to identity governance more generally [15], and its Zero Trust approach to IAM is directly relevant to the principal-narrowing, network-pinning, continuous-verification, and identity-lifecycle measures this note recommends.

The *AI Controls Matrix (AICM) v1.1* provides the control framework for operationalizing these recommendations [13]. Its Identity and Access Management domain covers least privilege, periodic access review, and deprovisioning of identities, including those granted to third parties. Its Supply Chain Management, Transparency, and Accountability domain addresses third-party risk assessment and

contractual security obligations. Its Logging and Monitoring domain covers the audit-log coverage gaps this incident exposed. Organizations should map each external-trust IAM role to AICM IAM controls and require vendors to evidence the corresponding controls, for example through STAR assessments. Finally, CSA's *Roles and Responsibilities of Third Party Security Services* offers guidance on delineating customer and provider responsibilities in cloud environments [14]. That delineation is often unclear in cross-account integrations, where neither party clearly owns the decision to retire the trust relationship.

References

[1] Morrow, Z., Johnstone, S., and Read, B. (Wiz). "Trusted by Default: Qubole Cross-Account Access Abused in the Wild." Wiz Blog, November 9, 2026.

[2] Sharma, S. "Qubole supply chain attack exposed AWS, Entra ID, and Okta credentials." CSO Online, November 10, 2026.

[3] RuntimeWire Staff. "Sentra and Wiz trace AWS data theft to customer roles that trusted Qubole." RuntimeWire, November 9, 2026.

[4] Amazon Web Services. "Access to AWS accounts owned by third parties." AWS Identity and Access Management User Guide, accessed November 10, 2026.

[5] Amazon Web Services. "The confused deputy problem." AWS Identity and Access Management User Guide, accessed November 10, 2026.

[6] Idera, Inc. "Idera Inc. Acquires Qubole Adding Award Winning Data Lake Platform to Growing Database Tools Portfolio." Business Wire, October 13, 2020.

[7] Qubole. "Announcing Support for AWS IAM Roles." Qubole Blog, September 3, 2015 (updated May 1, 2024).

[8] Qubole. "Choosing between a Cross-account IAM Role and Dual IAM Roles." Qubole Data Service Documentation, accessed November 10, 2026.

[9] Cloud Security Alliance. "ShinyHunters' OAuth Pivot: A Year of SaaS Supply-Chain Breaches." CSA AI Safety Initiative, July 16, 2026.

[10] Amazon Web Services. "Revoke IAM role temporary security credentials." AWS Identity and Access Management User Guide, accessed November 10, 2026.

[11] Amazon Web Services. "Using AWS Identity and Access Management Access Analyzer." AWS Identity and Access Management User Guide, accessed November 10, 2026.

[12] Cloud Security Alliance. "Confronting Shadow Access Risks: Considerations for Zero Trust and Artificial Intelligence Deployments." Cloud Security Alliance, 2024.

[13] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance.

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

[14] Cloud Security Alliance. "Roles and Responsibilities of Third Party Security Services." Cloud Security Alliance.

[15] Cloud Security Alliance. "Zero Trust Principles and Guidance for Identity and Access Management (IAM)." Cloud Security Alliance.