INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-13 18:25 UTC

ShinyHunters Supply Chain Attack: Anodot Token Theft Enables Downstream Snowflake, S3, and Kinesis Compromise

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0175
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Anodot (SaaS analytics platform, version unspecified); Snowflake (cloud data platform); Amazon S3; Amazon Kinesis; Zendesk (Rockstar support instance); Rockstar Games (Grand Theft Auto Online, Red Dead Online analytics data)
Published	2026-04-13T16:08:10
Discovery Source	Rss

Executive Summary

ShinyHunters, a prolific financially motivated threat group, stole authentication tokens from Anodot, a SaaS analytics integrator, and used those tokens to access connected cloud environments, including Snowflake data warehouses, Amazon S3 buckets, and Amazon Kinesis streams, belonging to Anodot customers. Rockstar Games is a confirmed victim; the attackers accessed internal analytics data covering game economy metrics and player telemetry, and have issued ransom demands while leaking portions of the data publicly. The broader campaign claims dozens of victims, meaning any enterprise using Anodot as a data integration layer faces direct exposure risk regardless of their own security posture.

Technical Analysis

ShinyHunters executed a supply chain attack via Anodot, a SaaS anomaly detection and analytics platform. The root cause is insufficiently protected credential storage at the integrator layer (CWE-522) combined with improper access control (CWE-284) and improper authentication (CWE-287). No CVE has been assigned. Stolen tokens, likely long-lived OAuth or API tokens stored by Anodot to maintain persistent integrations, were used to authenticate directly into downstream customer environments via trusted relationship abuse (MITRE T1199). From there, attackers executed T1528 (Steal Application Access Token), T1078.004 (Valid Accounts: Cloud Accounts), and T1530 (Data from Cloud Storage) against Snowflake instances, S3 buckets, and Kinesis data streams. Exfiltration occurred via T1567.002 (Exfiltration to Cloud Storage). The attack pattern is

structurally identical to the 2024 Snowflake credential theft campaign attributed to UNC5537, where stolen credentials to a third-party integrator enabled mass downstream compromise without direct victim infrastructure breach. No patch is available from Anodot at time of reporting; remediation requires token revocation and re-issuance through affected cloud providers.

Action Checklist

1. Step 1: Containment. Within 4 hours, audit all active Anodot integration tokens across your Snowflake, AWS S3, and AWS Kinesis environments. Revoke any token that Anodot holds or that was provisioned for Anodot integrations. Disable Anodot's OAuth or API access at the cloud provider level (Snowflake account admin console, AWS IAM, AWS KMS) pending confirmation of scope from Anodot.
2. Step 2: Detection. Query Snowflake query history and access logs for authentication events using service accounts or OAuth tokens associated with Anodot. In AWS CloudTrail, filter for S3 GetObject, PutObject, ListBucket, and Kinesis GetRecords events from IAM roles or access keys tied to Anodot integrations. Look for access from unusual IP ranges or at atypical hours. Cross-reference against ShinyHunters-associated IOCs if released by threat intelligence providers.
3. Step 3: Eradication. Rotate all cloud credentials (Snowflake service account passwords, AWS IAM access keys, OAuth tokens) that Anodot held or could have accessed. Re-provision integrations only after Anodot confirms remediation of their token storage controls. Apply least-privilege scoping to any replacement credentials, restrict to minimum required S3 prefixes, Kinesis streams, and Snowflake schemas.
4. Step 4: Recovery. After re-issuing credentials, validate Anodot integrations resume correctly in a controlled test before restoring production data flows. Monitor Snowflake, S3, and Kinesis access logs for 30 days post-rotation for any residual access attempts using revoked tokens or previously observed IP patterns. Confirm no unauthorized Snowflake shares or S3 bucket policies were created during the compromise window.
5. Step 5: Post-Incident. This attack exposes a systemic control gap: third-party SaaS integrators holding persistent, high-privilege tokens to cloud data environments without commensurate monitoring. Conduct a full third-party token audit across all SaaS integrations. Implement token expiry policies and enforce time-limited credentials for all integrator access. Add SaaS integrator access to your continuous access review cycle. Map all integrators against the data they can reach and classify exposure by data sensitivity.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to executive leadership, legal, and external counsel immediately if forensic review of Snowflake ACCESS_HISTORY or S3 server access logs confirms that ShinyHunters accessed schemas or buckets containing PII, PHI, financial records, or game player data subject to GDPR, CCPA, or state breach notification statutes — the ransom demand and confirmed data leakage by ShinyHunters create active regulatory and reputational exposure requiring breach counsel involvement within hours, not days.

Recovery Notes	Post-rotation, validate that all replacement Anodot IAM credentials are scoped with explicit S3 prefix conditions, Kinesis stream ARN restrictions, and Snowflake schema-level grants — not wildcard bucket or warehouse access — before re-enabling production data flows. Monitor Snowflake DATA_SHARING_USAGE, S3 bucket replication configurations, and Kinesis consumer registrations continuously for 30 days to detect any ShinyHunters persistence mechanisms (unauthorized shares, replication rules, or registered consumers) that survive credential rotation. Obtain Anodot's formal written attestation of their token storage remediation and request a copy of their third-party penetration test or SOC 2 Type II report before re-provisioning integration access, given this compromise originated in their platform.
Forensic Artifacts	Snowflake ACCOUNT_USAGE.ACCESS_HISTORY and LOGIN_HISTORY (90-day retention): filtered on Anodot OAuth client IDs and service account usernames — provides the authoritative record of which Snowflake queries were executed under the stolen token, which schemas and tables were read, and whether any data was exfiltrated via SELECT into stage or external location | AWS CloudTrail S3 data events (GetObject, PutObject, ListBucket) and Kinesis data events (GetRecords, GetShardIterator, RegisterStreamConsumer) for Anodot-associated IAM roles: reveals the exact objects accessed, byte volumes transferred, and whether ShinyHunters registered persistent Kinesis consumers or configured S3 replication for ongoing exfiltration beyond the initial access window | AWS IAM CreateAccessKey and PutUserPolicy CloudTrail events under Anodot-associated principals during the compromise window: identifies whether ShinyHunters attempted to escalate privileges or create secondary persistence credentials beyond the stolen Anodot token, a known ShinyHunters TTPs pattern in prior supply chain campaigns | S3 server access logs (bucket-level, separate from CloudTrail) for all buckets accessible to Anodot: captures byte-count-level object read activity including partial GETs, providing exfiltration volume estimates for data classification and breach notification scope assessment — CloudTrail alone does not capture object sizes for GetObject events | Anodot platform-side audit logs (must be requested directly from Anodot): the upstream evidence source showing which customer OAuth tokens were accessed, exported, or queried from Anodot's token storage infrastructure — this is the primary artifact for establishing the initial compromise timeline before any downstream Snowflake or AWS access occurred and is not available through AWS or Snowflake native logging

Per-Action IR Details

Step 1: Containment — Immediately audit all active Anodot integration tokens across your Snowflake, AWS S3, and AWS Kinesis environments. Revoke any token that Anodot holds or that was provisioned for Anodot integrations. Disable Anodot's OAuth or API access at the cloud provider level (Snowflake account admin console, AWS IAM, AWS KMS) pending confirmation of scope from Anodot.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy (CSF RS function: execute IR plan, categorize, contain, mitigate)

Controls: NIST IR-4 (Incident Handling), NIST AC-2 (Account Management), NIST AC-17 (Remote Access), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Export the full AWS IAM credential report via `aws iam generate-credential-report && aws iam get-credential-report --output text --query Content | base64 -d` and grep for any access key or role with 'anodot' in the associated tag, description, or policy name. In Snowflake, run `SHOW INTEGRATIONS;` and `SELECT * FROM SNOWFLAKE.ACCOUNT_USAGE.ACCESS_HISTORY WHERE USER_NAME ILIKE '%anodot%' AND QUERY_START_TIME >= DATEADD(day, -90, CURRENT_TIMESTAMP());` to enumerate OAuth integrations and recent activity before revoking. For AWS KMS, run `aws kms list-grants --key-id ` for any CMK used by S3 buckets or Kinesis streams accessible to Anodot, then revoke grants tied to the Anodot IAM principal using `aws kms

revoke-grant`.

Evidence: Before revoking tokens, snapshot all evidence of Anodot-associated credential usage: (1) Snowflake ACCOUNT_USAGE.ACCESS_HISTORY and LOGIN_HISTORY tables filtered to Anodot service account usernames and OAuth client IDs — export to immutable storage before revoking so revocation does not obscure pre-existing access records; (2) AWS CloudTrail entries for AssumeRole, GetSessionToken, and CreateToken API calls originating from Anodot IAM roles or access keys across all regions; (3) AWS IAM credential report timestamped at the moment of discovery to establish the last-used date for each Anodot-associated access key; (4) Snowflake SHOW INTEGRATIONS output and OAuth token metadata — captures the OAuth client ID ShinyHunters would have stolen or reused; (5) Current S3 bucket policies and Kinesis stream resource-based policies for any bucket/stream accessible to Anodot, preserving the pre-revocation permission state as forensic baseline.

Step 2: Detection — Query Snowflake query history and access logs for authentication events using service accounts or OAuth tokens associated with Anodot. In AWS CloudTrail, filter for S3 GetObject, PutObject, ListBucket, and Kinesis GetRecords events from IAM roles or access keys tied to Anodot integrations. Look for access from unusual IP ranges or at atypical hours. Cross-reference against ShinyHunters-associated IOCs if released by threat intelligence providers.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis (CSF DE function: monitor, detect, analyze, correlate, triage adverse events)

Controls: NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST SI-4 (System Monitoring), CIS 8.2 (Collect Audit Logs), MITRE ATT&CK T1078.004 (Valid Accounts: Cloud Accounts) — ShinyHunters lateral movement via stolen OAuth/API tokens into Snowflake and AWS

Compensating: For teams without SIEM, use AWS CloudTrail Lake or Athena to query raw CloudTrail logs stored in S3. Run: `SELECT eventTime, userIdentity.arn, sourceIPAddress, eventName, requestParameters FROM cloudtrail_logs WHERE eventName IN ('GetObject','PutObject','ListBucket','GetRecords','GetShardIterator') AND userIdentity.arn LIKE '%anodot%' AND eventTime > '2024-01-01T00:00:00Z' ORDER BY eventTime DESC;` — adjust the date to your Anodot integration provisioning date. For Snowflake, use the free ACCOUNT_USAGE schema (90-day retention): `SELECT query_id, user_name, role_name, query_text, start_time, end_time, client_application_id FROM SNOWFLAKE.ACCOUNT_USAGE.QUERY_HISTORY WHERE user_name ILIKE '%anodot%' OR client_application_id ILIKE '%anodot%' ORDER BY start_time DESC;`. Cross-reference source IPs against ShinyHunters infrastructure — check free feeds from abuse.ch and Feodo Tracker if ShinyHunters-specific IOCs are not yet available from your TI provider.

Evidence: Collect before analysis to prevent log rotation or expiry overwriting key records: (1) Snowflake ACCOUNT_USAGE.LOGIN_HISTORY for the full 90-day retention window, filtering on Anodot service accounts and OAuth client IDs — specifically look for login events from IPs not matching Anodot's documented egress ranges, which would indicate ShinyHunters reusing stolen tokens from a different infrastructure; (2) AWS CloudTrail management events for IAM actions (CreateAccessKey, AttachRolePolicy, PutBucketPolicy) taken under Anodot-associated principals — ShinyHunters may have attempted privilege escalation or persistence beyond initial data access; (3) Kinesis GetRecords and GetShardIterator events in CloudTrail showing which stream shards were consumed and the volume of records retrieved — this establishes the data exfiltration scope for Kinesis-sourced telemetry; (4) S3 server access logs (separate from CloudTrail) for affected buckets, which capture object-level GET requests including byte counts — critical for quantifying what game economy or player telemetry data Rockstar-class victims lost; (5) Anodot platform audit logs (request from Anodot directly) showing which customer OAuth tokens were accessed or exported from their system — this is the upstream evidence of the initial token theft before downstream cloud access occurred.

Step 3: Eradication — Rotate all cloud credentials (Snowflake service account passwords, AWS IAM access keys, OAuth tokens) that Anodot held or could have accessed. Re-provision integrations only after Anodot confirms remediation of their token storage controls. Apply least-privilege scoping to any replacement credentials — restrict to minimum required S3 prefixes, Kinesis streams, and Snowflake schemas.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication (CSF RS function: remove threat from environment, verify eradication)

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST AC-2 (Account Management), NIST AC-6 (Least Privilege), CIS 5.2 (Use Unique Passwords), CIS 7.2 (Establish and Maintain a Remediation Process), MITRE ATT&CK T1098 (Account Manipulation) — adversary may have created secondary persistence credentials during the access window

Compensating: Before rotation, enumerate all IAM entities that have had access to Anodot-associated resources using AWS IAM Access Analyzer: `aws accessanalyzer list-findings --analyzer-arn ` — this surfaces any external principal grants that ShinyHunters may have added for persistence. For Snowflake, run `SHOW GRANTS TO USER ;` and `SELECT * FROM SNOWFLAKE.ACCOUNT_USAGE.GRANTS_TO_USERS WHERE GRANTEE_NAME ILIKE '%anodot%';` to detect any role escalations added during the compromise window. Rotate AWS IAM access keys with a zero-downtime two-key approach: create new key, update Anodot's configuration (post-remediation confirmation only), then deactivate and delete old key using `aws iam update-access-key --access-key-id --status Inactive` followed by `aws iam delete-access-key`. For replacement S3 scoping, use prefix-level condition in IAM policy: `"Condition": {"StringLike": {"s3:prefix": ["anodot/ingest/*"]}}` rather than bucket-wide access.

Evidence: Before rotating credentials, capture: (1) AWS IAM Access Advisor data for each Anodot-associated role (`aws iam get-service-last-accessed-details`) — identifies which AWS services the compromised credential actually touched beyond S3 and Kinesis, revealing if ShinyHunters pivoted to additional services (e.g., Glue, Athena, DynamoDB) that are not yet in scope; (2) Snowflake ACCOUNT_USAGE.GRANTS_TO_ROLES and GRANTS_TO_USERS for the incident window — any new role grants or privilege escalations added under the Anodot identity during compromise indicate adversary persistence that must be removed before credentials are rotated; (3) AWS CloudTrail CreateAccessKey and UpdateAccessKey events under all Anodot-associated IAM users — ShinyHunters may have created additional access keys for persistence that survive a single-key rotation; (4) S3 bucket ACL and bucket policy snapshots for every bucket accessible to Anodot — confirm no anonymous read grants or cross-account access policies were inserted during the compromise window before those buckets are reconnected to re-provisioned integrations.

Step 4: Recovery — After re-issuing credentials, validate Anodot integrations resume correctly in a controlled test before restoring production data flows. Monitor Snowflake, S3, and Kinesis access logs for 30 days post-rotation for any residual access attempts using revoked tokens or previously observed IP patterns. Confirm no unauthorized Snowflake shares or S3 bucket policies were created during the compromise window.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery (CSF RC function: execute recovery plan, restore systems, verify integrity, communicate)

Controls: NIST IR-4 (Incident Handling), NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), MITRE ATT&CK T1537 (Transfer Data to Cloud Account) — validate no unauthorized Snowflake data shares were created to adversary-controlled Snowflake accounts, a persistence/exfiltration technique consistent with this attack pattern

Compensating: Automate 30-day post-rotation monitoring without a SIEM by creating an AWS CloudWatch Metric Filter on CloudTrail logs that alerts on any authentication using the specific revoked Anodot IAM access key IDs: use `aws cloudwatch put-metric-alarm` targeting a filter on `errorCode = 'InvalidClientTokenId'` combined with the old key ID. For Snowflake data share monitoring, run weekly: `SHOW SHARES;` and `SELECT * FROM SNOWFLAKE.ACCOUNT_USAGE.DATA_SHARING_USAGE WHERE START_TIME >= ;` to detect any unauthorized shares created during or after the compromise window. For S3 bucket policy drift, use AWS Config with the `s3-bucket-policy-not-more-permissive` managed rule and set a weekly evaluation; free-tier Config is sufficient for this check. For Kinesis, verify no new consumer applications were registered during the compromise window with `aws kinesis list-stream-consumers --stream-arn `.

Evidence: Establish a recovery validation baseline by capturing: (1) Snowflake SHOW SHARES output and DATA_SHARING_USAGE records immediately post-rotation — any Snowflake share created between the estimated compromise start date and the rotation date with an external account ID not matching known business partners represents data exfiltration infrastructure that must be removed; (2) S3 bucket replication configuration for all affected

buckets (`aws s3api get-bucket-replication`) — ShinyHunters may have configured cross-account or cross-region replication to adversary-controlled buckets as a persistent exfiltration channel that survives credential rotation; (3) Kinesis enhanced fan-out consumer registrations (`aws kinesis list-stream-consumers`) — an adversary-registered consumer application would continue receiving real-time stream data even after the original access key is rotated if the consumer registration itself is not revoked; (4) AWS CloudTrail for the 30-day monitoring window specifically looking for `401 Unauthorized` or `403 Forbidden` responses from the revoked Anodot key IDs — any attempt using revoked credentials indicates ShinyHunters is retrying access and the token was used in additional infrastructure not yet identified.

Step 5: Post-Incident — This attack exposes a systemic control gap: third-party SaaS integrators holding persistent, high-privilege tokens to cloud data environments without commensurate monitoring. Conduct a full third-party token audit across all SaaS integrations. Implement token expiry policies and enforce time-limited credentials for all integrator access. Add SaaS integrator access to your continuous access review cycle. Map all integrators against the data they can reach and classify exposure by data sensitivity.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity (CSF GV/ID functions: lessons learned, update policies, improve detection, share intelligence)

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), NIST SA-9 (External System Services), NIST AC-2 (Account Management), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 6.1 (Establish an Access Granting Process), CIS 6.2 (Establish an Access Revoking Process), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Build a free third-party token inventory using AWS IAM: run `aws iam list-users` and `aws iam list-roles`, then for each entity run `aws iam list-attached-user-policies` and `aws iam list-role-policies` — tag any IAM principal whose description, name, or tags reference a SaaS vendor name. Generate a CSV mapping each third-party IAM principal to the S3 bucket prefixes, Kinesis streams, and Snowflake schemas it can access, then classify each data asset by sensitivity. For token expiry enforcement without enterprise tooling, use AWS IAM credential report automation: schedule a weekly Lambda (free tier) or cron-driven `aws iam generate-credential-report` that flags any access key older than 90 days associated with a third-party integration and pages the team via SNS. In Snowflake, enforce OAuth token lifetime limits in the security integration definition: `ALTER SECURITY INTEGRATION anodot_oauth SET OAUTH_ACCESS_TOKEN_VALIDITY = 3600;` — this prevents long-lived tokens of the type ShinyHunters stole from providing extended unauthorized access.

Evidence: The post-incident artifact record that must be preserved for lessons-learned and potential regulatory notification includes: (1) The complete timeline reconstructed from Snowflake ACCESS_HISTORY, CloudTrail, and S3 server access logs showing the earliest evidence of anomalous Anodot token use — this establishes the breach window for any applicable breach notification obligations if PII or regulated data transited the compromised Snowflake schemas or S3 buckets; (2) The data classification inventory for every S3 bucket prefix, Snowflake schema, and Kinesis stream that the compromised Anodot tokens had read access to — required to scope regulatory exposure (GDPR, CCPA, state breach notification laws) if customer or employee PII was reachable; (3) Anodot's written remediation confirmation and any shared forensic findings about how the tokens were stored and compromised on their side — critical for determining whether the root cause was inadequate secrets management (e.g., tokens stored in plaintext configuration files or version control) versus a platform-level breach; (4) The pre- and post-incident IAM policy comparison for all Anodot-associated principals showing the privilege reduction applied during eradication, serving as documented evidence of control improvement for audit purposes.

Detection Guidance

Snowflake: Query SNOWFLAKE.ACCOUNT_USAGE.LOGIN_HISTORY and QUERY_HISTORY for authentication events from service accounts or OAuth clients associated with Anodot during the suspected compromise window. Flag any access from IPs not in your known Anodot service IP range. AWS CloudTrail: Filter for S3 and Kinesis API calls (GetObject, ListBucket, GetRecords, DescribeStream) made by IAM

principals provisioned for Anodot. Look for high-volume GetObject sequences or cross-region access inconsistent with Anodot's normal operating region. IAM: Check for any new IAM roles, policies, or access keys created by Anodot-associated principals, a potential indicator of persistence establishment. Behavioral indicator: access to Snowflake schemas or S3 prefixes outside Anodot's normal operational scope (e.g., accessing tables or buckets Anodot has no business reason to read). SIEM correlation: join Anodot integration activity logs against CloudTrail and Snowflake audit logs on principal identifier to surface any lateral movement beyond expected data paths.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://www.bleepingcomputer.com/news/security/stolen-rockstar-games-analytics-data-leaked-by-extortion-gang/`	BleepingComputer reporting on ShinyHunters Anodot/Rockstar campaign — source article, not a malicious IOC	HIGH
URL	`https://hackread.com/shinyhunters-rockstar-games-snowflake-breach-anodot/`	HackRead reporting confirming ShinyHunters claim of Snowflake breach via Anodot — source article, not a malicious IOC	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1078.004** — Cloud Accounts
- **T1530** — Data from Cloud Storage
- **T1528** — Steal Application Access Token
- **T1199** — Trusted Relationship
- **T1567.002** — Exfiltration to Cloud Storage
- **T1657** — Financial Theft
- **T1552.001** — Credentials In Files

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A04:2021** — Insecure Design
- **A07:2021** — Identification and Authentication Failures

NIST-800-53R5

- **AC-3** — Access Enforcement
- **IA-5** — Authenticator Management
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-8** — Identification and Authentication (Non-Organizational Users)
- **CP-9** — System Backup

- **IR-4** — Incident Handling
- **SR-2** — Supply Chain Risk Management Plan
- **SI-4** — System Monitoring

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **5.2** — Use Unique Passwords
- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **15.1** — Establish and Maintain an Inventory of Service Providers
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets
- **CC9.2** — Manages risks associated with vendors and business partners

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.308(a)(5)(ii)(D)** — Password Management
- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(7)(ii)(A)** — Data Backup Plan

NIST-CSF-2

- **RS.MI-01** — Incidents are contained
- **GV.SC-01** — Cybersecurity supply chain risk management program
- **DE.CM-01** — Networks and network services are monitored

ISO-27001-2022

- **A.5.29** — Information security during disruption
- **A.5.34** — Privacy and protection of personal information
- **A.5.21** — Managing information security in the ICT supply chain
- **A.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1078.004	Cloud Accounts	Defense-Evasion
T1530	Data from Cloud Storage	Collection

Technique ID	Technique Name	Tactic
T1528	Steal Application Access Token	Credential-Access
T1199	Trusted Relationship	Initial-Access
T1567.002	Exfiltration to Cloud Storage	Exfiltration
T1657	Financial Theft	Impact
T1552.001	Credentials In Files	Credential-Access

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/stolen-rockstar-game...	T3
Rockstar Games says hack will have 'no impact' - The Verge	https://www.theverge.com/games/910815/rockstar-games-says-hack-will...	T2
ShinyHunters Claims Rockstar Games Snowflake Breach via Anodot	https://hackread.com/shinyhunters-rockstar-games-snowflake-breach-a...	T3
Rockstar Games Caught Up In Anodot Cybersecurity Compromise ...	https://www.mmorpg.com/news/rockstar-games-caught-up-in-anodot-cybe.. .	T3
Rockstar confirms breach as hackers issue ransom threat - MSN	https://www.msn.com/en-gb/news/insight/rockstar-confirms-breach-as-...	T3

Generated 2026-04-13 18:25 UTC by TJS Security Command Center