INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-30 14:01 UTC

Supply Chain Attacks Exploit Non-Human Identities Amid Identity Governance Gaps

GOVERNANCE | HIGH

SCC Item ID	SCC-GOV-2026-0041
Type	Governance
Severity	HIGH
Affected Products	Organizations broadly relying on third-party integrations, external service accounts, API tokens, OAuth grants, and machine/non-human identities across all sectors
Published	2026-05-28
Discovery Source	Gemini

Executive Summary

Threat actors are increasingly compromising organizations by exploiting non-human identities, service accounts, API tokens, OAuth grants, and CI/CD pipeline credentials, through trusted third-party relationships rather than direct system attacks. Because identity governance programs have historically focused on human workforce accounts, NHIs are frequently over-permissioned, long-lived, unrotated, and unmonitored, creating a persistent vulnerability attackers exploit for initial access and lateral movement. Any organization relying on cloud services, third-party integrations, or automated pipelines carries material exposure; a single compromised supplier credential can provide broad access to downstream customer environments.

Technical Analysis

This governance item describes a structural attack pattern rather than a discrete CVE. Threat actors target non-human identities (NHIs), including OAuth tokens (T1528), service account credentials (T1078.004), application access tokens (T1550.001), and secrets embedded in pipelines or repositories (T1552.001), to achieve initial access via trusted relationships (T1199) or software supply chain compromise (T1195), then perform account manipulation (T1098) to persist. Relevant CWEs include CWE-522 (Insufficiently Protected Credentials), CWE-613 (Insufficient Session Expiration), CWE-306 (Missing Authentication for Critical Function), CWE-250 (Execution with Unnecessary Privileges), and CWE-269 (Improper Privilege Management). No CVE applies; this is a systemic governance gap. Root causes include the absence of NHI inventory, no token expiration enforcement, over-permissioned service accounts, no behavioral anomaly detection on machine identities, and inadequate vendor access review cycles. Cloud-native architectures and AI agent deployments accelerate NHI proliferation faster than governance programs adapt. There is no single patch; remediation

requires inventory, lifecycle controls, least-privilege enforcement, and continuous monitoring.

Action Checklist

1. Step 1: Containment, Audit all active OAuth grants, API tokens, and service account credentials with access to production systems. Revoke any token or service account credential that cannot be attributed to a current, documented integration. Prioritize internet-facing and cloud-hosted environments. Reference NIST AC-2 (Account Management) and CIS 5.1 (Establish and Maintain an Inventory of Accounts).
2. Step 2: Detection, Query identity provider logs, cloud IAM audit logs (AWS CloudTrail, Azure AD sign-in logs, GCP Audit Logs), and CI/CD pipeline logs for service account authentications from unexpected source IPs, unusual hours, or anomalous API call volumes. Look for OAuth token usage from new geographic regions or clients. Alert on any NHI authenticating outside its documented integration pattern. Reference NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs).
3. Step 3: Remediation, Implement token expiration on all long-lived API keys and OAuth tokens; replace non-expiring tokens with short-lived credentials using a secrets manager (e.g., HashiCorp Vault, AWS Secrets Manager). Apply least-privilege scoping to all service accounts, remove permissions not required for the documented function. Reference NIST AC-6 (Least Privilege), NIST IA-5 (Authenticator Management), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), and D3FEND countermeasures D3-CRO (Credential Rotation) and D3-CH (Credential Hardening).
4. Step 4: Recovery, Validate that all revoked credentials have been replaced with scoped, expiring equivalents and that dependent integrations remain functional. Re-run access reviews for all third-party vendor service accounts against the principle of least privilege (NIST AC-6). Enable anomaly alerting on NHI behavior baselines in your SIEM or CASB. Confirm CI/CD pipelines are not storing secrets in plaintext environment variables or repository code. Reference NIST AU-6 (Audit Record Review).
5. Step 5: Post-Incident, Extend your Identity Governance and Administration (IGA) program scope to formally include NHIs. Establish NHI inventory as a documented control, mapped to CIS 1.1 (Enterprise Asset Inventory) and CIS 5.1 (Account Inventory). Implement a vendor access review cycle (minimum quarterly) covering all third-party service accounts and OAuth grants, referencing NIST AC-20 (Use of External Systems) and NIST AC-2 (Account Management). Document NHI lifecycle policies covering provisioning, rotation schedules, and deprovisioning triggers.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to CISO and legal/privacy counsel if forensic review of NHI activity logs reveals evidence of data exfiltration (e.g., bulk 's3:GetObject', 'storage.objects.get', or database query API calls by a compromised service account), access to systems storing PII, PHI, or PCI-scoped data (triggering breach notification obligations under GDPR Art. 33, HIPAA §164.410, or applicable state laws), or if a compromised CI/CD credential had write access to production code repositories (indicating potential software supply chain integrity breach requiring customer notification).

Recovery Notes	After credential rotation and scope reduction are confirmed, maintain elevated monitoring on all NHI principals for a minimum of 30 days — adversaries with prior access to CI/CD pipelines may have implanted backdoors in build artifacts, container images, or infrastructure-as-code templates that survive credential rotation. Validate the integrity of container images built during the suspected compromise window by re-pulling from registry and comparing SHA256 digests against pre-incident baselines, or rebuilding from source with clean credentials. Continue weekly NHI access reviews for 90 days post-incident before reverting to the standard quarterly cycle, and confirm that all monitoring baselines have been recalculated using post-remediation activity to avoid alert fatigue from the remediation changes themselves.
Forensic Artifacts	AWS CloudTrail: 'AssumeRoleWithWebIdentity', 'GetCallerIdentity', 'CreateAccessKey', and high-volume data plane events (s3:GetObject, dynamodb:Scan) attributed to NHI principals — these establish the adversary's access timeline and data exposure scope from supply chain credential abuse | Azure AD Sign-In Logs and Audit Logs: service principal authentication records with 'ipAddress', 'userAgent', 'resourceDisplayName', and 'conditionalAccessStatus' fields — OAuth token reuse by an adversary typically shows the legitimate grant scope being accessed from a foreign IP with a non-vendor user-agent string, bypassing CA policies that apply only to human accounts | CI/CD pipeline execution logs (GitHub Actions workflow run logs at .github/workflows/*.yml, GitLab job trace logs): secret injection events, OIDC token exchange requests, and any 'run:' steps invoking curl, wget, or base64 against external hosts — indicative of a compromised pipeline exfiltrating secrets or staging payloads during build execution | IdP System Logs (Okta: 'app.oauth2.token.grant.implicit', 'app.oauth2.as.token.grant'; Auth0: 'fapi' and 'sapi' log events): OAuth grant issuance and token refresh events showing scope, client_id, and grant_type — a compromised third-party integration reusing a refresh token will show continuous token refresh activity from an unexpected client_id or redirect_uri | Secrets manager access logs (AWS Secrets Manager CloudTrail events 'GetSecretValue' by non-approved principals; HashiCorp Vault audit log showing lease issuance and secret read by unexpected entity IDs): anomalous secrets retrieval by NHI principals outside their documented pipeline execution windows is a high-confidence indicator of credential harvesting during the supply chain compromise

Per-Action IR Details

Step 1: Containment — Audit all active OAuth grants, API tokens, and service account credentials with access to production systems. Revoke any token or service account credential that cannot be attributed to a current, documented integration. Prioritize internet-facing and cloud-hosted environments. Reference NIST AC-2 (Account Management) and CIS 5.1 (Establish and Maintain an Inventory of Accounts).

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolate compromised identities before eradication to prevent lateral movement via trusted third-party credential chains

Controls: NIST AC-2 (Account Management), NIST AC-20 (Use of External Systems), NIST IR-4 (Incident Handling), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: For teams without a CASB or IGA platform: in AWS, run 'aws iam list-users' and 'aws iam list-roles' combined with 'aws iam generate-credential-report' (CSV export) to enumerate all access keys and their last-used timestamps; flag any key unused >30 days or with no associated CloudTrail principal tag. In Azure, use 'az ad app list --all' and 'az ad sp list --all' piped to jq to enumerate service principals and app registrations; cross-reference against a manually maintained integration register. For GitHub Actions or GitLab CI, grep all workflow YAML files for hardcoded tokens using: grep -rE '(ghp_|glpat-|AKIA|xox[baprs]-)[A-Za-z0-9]' .github/ or equivalent pipeline config directory.

Evidence: BEFORE revoking any credential, capture: (1) AWS CloudTrail 'ListBuckets', 'AssumeRole', 'GetCallerIdentity', and 'CreateAccessKey' events tied to the service account principal for the prior 90 days; (2) Azure

AD sign-in logs filtered on servicePrincipalId showing resource access, IP address, and userAgent fields; (3) GCP Cloud Audit Logs (Admin Activity and Data Access) for the affected service account email, specifically 'iam.serviceAccounts.actAs' and 'storage.objects.list' calls; (4) OAuth grant records from your IdP (Okta System Log event type 'app.oauth2.token.grant', Azure AD 'Add app role assignment to service principal') showing grant scope and consenting user; (5) CI/CD pipeline execution logs (GitHub Actions workflow run logs, GitLab job logs) showing which secrets were injected and from which environment at time of suspected compromise.

Step 2: Detection — Query identity provider logs, cloud IAM audit logs (AWS CloudTrail, Azure AD sign-in logs, GCP Audit Logs), and CI/CD pipeline logs for service account authentications from unexpected source IPs, unusual hours, or anomalous API call volumes. Look for OAuth token usage from new geographic regions or clients. Alert on any NHI authenticating outside its documented integration pattern. Reference NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS 8.2 (Collect Audit Logs).

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate NHI authentication anomalies across IdP, cloud IAM, and pipeline logs to distinguish legitimate automation from adversary use of stolen credentials

Controls: NIST AU-2 (Event Logging), 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)

Compensating: Without a SIEM, use the AWS CLI to detect anomalous NHI activity: 'aws cloudtrail lookup-events --lookup-attributes AttributeKey=Username,AttributeValue= --start-time ' and filter for sourceIPAddress values outside the known integration CIDR. For Azure, use the Microsoft Graph API or Azure CLI: 'az monitor activity-log list --query "[?contains(caller, '@')]" --offset 72h' filtered on service principal names. For GCP, use 'gcloud logging read "resource.type=service_account AND protoPayload.authenticationInfo.principalEmail=" --limit 500 --format json'. For OAuth anomaly detection without a CASB, enable Okta ThreatInsight or pull Okta System Log via API filtering on 'app.oauth2.as.token.grant' events and manually pivot on 'client.geographicalContext.country' and 'client.ipAddress' fields. Use the open-source Sigma rule 'aws_sts_assumerole_by_unknown_principal.yml' for offline log analysis against exported CloudTrail JSON.

Evidence: Capture before alerting thresholds are tuned: (1) AWS CloudTrail events 'ConsoleLogin', 'AssumeRoleWithWebIdentity', and 'GetSessionToken' for all NHI principals, noting 'userAgent' strings — adversaries abusing stolen tokens frequently present non-standard or scripted user agents (e.g., 'python-requests/2.x', 'curl/7.x') inconsistent with the vendor's documented integration SDK; (2) Azure AD sign-in logs with fields 'appDisplayName', 'resourceDisplayName', 'ipAddress', 'conditionalAccessStatus', and 'riskDetail' — supply chain token abuse often bypasses Conditional Access because the token was issued legitimately before compromise; (3) Okta System Log events 'policy.evaluate_sign_on' and 'user.session.access_admin_app' for service accounts, particularly those with 'outcome.result: ALLOW' and no MFA step-up (NHIs typically exempt from MFA policies); (4) GitHub Actions audit log entries for 'workflows.completed_workflow_run' and 'secret_scanning.secrets_received' to identify pipeline secret exposure events; (5) Network flow logs (VPC Flow Logs, NSG Flow Logs) showing outbound connections from integration hosts to non-approved external IPs during pipeline execution windows.

Step 3: Eradication — Enforce token expiration on all long-lived API keys and OAuth tokens; replace non-expiring tokens with short-lived credentials using a secrets manager (e.g., HashiCorp Vault, AWS Secrets Manager). Apply least-privilege scoping to all service accounts — remove permissions not required for the documented function. Reference NIST AC-6 (Least Privilege), NIST IA-5 (Authenticator Management), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), and D3FEND countermeasures D3-CRO (Credential Rotation) and D3-CH (Credential Hardening).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove the adversary's persistent access mechanism (long-lived NHI credentials) and eliminate the structural condition (over-permissioning) that made supply chain lateral movement possible

Controls: NIST AC-6 (Least Privilege), NIST IA-5 (Authenticator Management), NIST CM-7 (Least Functionality), NIST SA-9 (External System Services), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 6.5 (Require MFA for Administrative Access)

Compensating: Without HashiCorp Vault enterprise, deploy HashiCorp Vault open-source (free) on a hardened internal host and configure the AWS Secrets Engine to issue dynamic, time-bound IAM credentials (TTL 1 hour) in place of static access keys. For GitHub Actions, migrate from static repository secrets to OIDC federation: configure 'permissions: id-token: write' in workflow YAML and use 'aws-actions/configure-aws-credentials@v4' with 'role-to-assume' — this eliminates stored AWS credentials entirely. For permission scoping without a CIEM tool, run AWS IAM Access Analyzer's policy generation feature (free, built-in) against CloudTrail to produce a least-privilege policy from actual API call history: 'aws accessanalyzer start-policy-generation --policy-generation-details principalArn='. For Azure service principals, use 'az role assignment list --assignee --all' to enumerate all role assignments and manually remove any role broader than the documented function.

Evidence: Before rotating credentials, preserve: (1) A complete snapshot of the current IAM policy document attached to each affected service account or role (AWS: 'aws iam get-role-policy', 'aws iam list-attached-role-policies'; Azure: 'az role assignment list'; GCP: 'gcloud projects get-iam-policy') — this establishes the over-permissioned baseline for the post-incident report and any regulatory disclosure; (2) OAuth token introspection records showing granted scopes at time of compromise (call the IdP introspection endpoint while the token is still active — this data is lost after revocation); (3) AWS CloudTrail 'DeleteAccessKey' and 'CreateAccessKey' events timestamped during the incident window to establish adversary credential lifecycle if the attacker generated their own access keys using a compromised role; (4) CI/CD pipeline secret audit export (GitHub: Settings > Security > Secret scanning alerts; GitLab: Security > Vulnerability Report) showing any secrets detected in committed code before rotation.

Step 4: Recovery — Validate that all revoked credentials have been replaced with scoped, expiring equivalents and that dependent integrations remain functional. Re-run access reviews for all third-party vendor service accounts against the principle of least privilege (NIST AC-6). Enable anomaly alerting on NHI behavior baselines in your SIEM or CASB. Confirm CI/CD pipelines are not storing secrets in plaintext environment variables or repository code. Reference D3-LAM (Local Account Monitoring) and D3-UAP (User Account Permissions).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore trusted integration functionality using verified, scoped credentials and confirm monitoring coverage is active before returning NHI-dependent workloads to production

Controls: NIST AC-6 (Least Privilege), NIST AC-20 (Use of External Systems), NIST CA-7 (Continuous Monitoring), NIST SI-4 (System Monitoring), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without a SIEM or CASB for NHI baseline alerting, use AWS CloudWatch Metric Filters to create alarms on service account API call volume deviations: create a metric filter on CloudTrail logs for a specific IAM principal and set a CloudWatch Alarm triggering SNS email when call count exceeds 2x the 7-day rolling average. For GitHub Actions, enable 'required reviewers' on environment secrets so any pipeline job requesting production credentials requires manual approval. To detect plaintext secrets in pipelines without a paid scanning tool, run truffleHog (open-source, free) against all pipeline config repositories: 'trufflehog git file://. --only-verified' — this detects high-entropy strings and known secret patterns in git history, not just the current HEAD. For ongoing NHI monitoring without EDR, deploy osquery with the 'aws_instance_metadata' table query scheduled hourly to detect credential endpoint access from unexpected processes.

Evidence: Before declaring recovery complete, collect and retain: (1) Integration smoke test results showing each replaced credential authenticates successfully and only accesses its documented resources — failures here indicate either over-scoping correction broke functionality (document the gap) or the vendor integration is broader than documented; (2) A re-run of the IAM credential report (AWS) or service principal audit (Azure/GCP) confirming no legacy static keys persist for affected principals; (3) CI/CD pipeline execution logs from the first post-recovery run of each affected pipeline, confirming no 'secret not found' errors that would indicate missed credential dependencies; (4) SIEM or CloudWatch alert configuration export confirming NHI behavioral baselines are codified and alert thresholds are active before workloads return to production.

Step 5: Post-Incident — Extend your Identity Governance and Administration (IGA) program scope to formally include NHIs. Establish NHI inventory as a documented control, mapped to CIS 1.1 (Enterprise Asset Inventory) and CIS 5.1 (Account Inventory). Implement a vendor access review cycle (minimum quarterly)

covering all third-party service accounts and OAuth grants, referencing NIST AC-20 (Use of External Systems) and NIST AC-2 (Account Management). Document NHI lifecycle policies covering provisioning, rotation schedules, and deprovisioning triggers.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: document lessons learned and drive structural IGA program changes to close the NHI governance gap that enabled the supply chain attack vector

Controls: NIST AC-2 (Account Management), NIST AC-20 (Use of External Systems), NIST IR-4 (Incident Handling), NIST PM-9 (Risk Management Strategy), NIST SA-9 (External System Services), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Without an IGA platform (SailPoint, Saviynt, etc.), build an NHI inventory using a shared spreadsheet or low-cost tool: for AWS, schedule a monthly Lambda (free tier) that calls 'aws iam generate-credential-report' and exports to S3, then sends a diff report via SNS email highlighting new or changed service accounts. For OAuth grant inventory without a CASB, use Google Workspace Admin SDK (free) 'admin.directory.tokens.list' or Microsoft Graph 'GET /servicePrincipals/{id}/oauth2PermissionGrants' on a scheduled basis and export to CSV for quarterly review. For the vendor access review process, use a documented Google Form or equivalent to require each third-party vendor to re-attest their service account usage quarterly; non-responses trigger automatic revocation under the policy. Publish a YARA rule or Sigma rule to your detection stack covering the specific NHI abuse patterns observed in this incident to operationalize threat intelligence from the event.

Evidence: Artifacts to retain for the post-incident review and to inform the updated IGA program: (1) The full timeline of NHI credential activity extracted from CloudTrail, Azure AD, and GCP Audit Logs covering the period from initial compromise through containment — this becomes the forensic basis for the lessons-learned report and any regulatory notification; (2) The before/after permission delta for each affected service account documenting the over-permissioned state at time of compromise versus the least-privilege state post-remediation; (3) The vendor integration register as it existed at time of discovery versus the authoritative register produced during Step 1 audit — the delta identifies undocumented integrations that represent ongoing risk; (4) Any threat intelligence indicators (source IPs, user-agent strings, anomalous API call sequences) extracted from the incident logs, formatted as STIX 2.1 or MISP objects for sharing with ISACs or internal threat intel platforms; (5) The IGA program gap analysis document produced from this incident, formally mapping NHI lifecycle controls to NIST AC-2, AC-20, and SA-9 as the authoritative record for future audit or regulatory review.

Detection Guidance

Detection focuses on behavioral anomalies in machine identity activity rather than signature-based IOCs, as NHI abuse leaves no malware footprint. Key log sources: cloud IAM audit logs (AWS CloudTrail, Azure AD sign-in/audit logs, GCP Cloud Audit Logs), CI/CD platform logs (GitHub Actions, GitLab, Jenkins), secrets manager access logs, and API gateway access logs.

Behavioral indicators to hunt:

- Service account or API token authenticating from a source IP not associated with its documented integration host

- OAuth token used outside its registered application or from a new user agent string

- Service account calling APIs or accessing resources outside its documented functional scope

- Long-lived tokens (creation date older than 90 days) with recent high-volume API activity

- CI/CD pipeline credentials used interactively (as opposed to automated pipeline context)

- Sudden spike in service account authentication failures followed by success (credential stuffing against NHI)

- New OAuth grants or service account key creations not associated with a change ticket

Query approach (conceptual, adapt to your SIEM):

- Join IAM audit logs on service account principals; filter for source IPs not in the known integration IP allowlist

- Alert on OAuth token issuances where the requesting application ID does not match the approved application registry

- Baseline NHI API call rates per identity; alert on deviations exceeding 2-3 standard deviations

Reference NIST SI-4 (System Monitoring), AU-6 (Audit Record Review), and D3FEND guidance on machine identity monitoring.

Framework Mappings

MITRE-ATTACK

- **T1199** — Trusted Relationship
- **T1078.004** — Cloud Accounts
- **T1550.001** — Application Access Token
- **T1098** — Account Manipulation
- **T1195** — Supply Chain Compromise
- **T1552.001** — Credentials In Files
- **T1528** — Steal Application Access Token

NIST-800-53R5

- **SA-9** — External System Services
- **SR-2** — Supply Chain Risk Management Plan
- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **IA-5** — Authenticator Management
- **IA-2** — Identification and Authentication (Organizational Users)
- **AC-6** — Least Privilege
- **SC-13** — Cryptographic Protection
- **SI-4** — System Monitoring

OWASP-TOP10-2021

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

CIS-V8

- **5.2** — Use Unique Passwords
- **6.3** — Require MFA for Externally-Exposed Applications
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts

- **6.8** — Define and Maintain Role-Based Access Control
- **15.1** — Establish and Maintain an Inventory of Service Providers
- **8.2** — Collect Audit Logs

HIPAA-SECURITY

- **164.308(a)(5)(ii)(D)** — Password Management
- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC9.2** — Manages risks associated with vendors and business partners

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain
- **A.5.23** — Information security for use of cloud services

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1199	Trusted Relationship	Initial-Access
T1078.004	Cloud Accounts	Defense-Evasion
T1550.001	Application Access Token	Defense-Evasion
T1098	Account Manipulation	Persistence
T1195	Supply Chain Compromise	Initial-Access
T1552.001	Credentials In Files	Credential-Access
T1528	Steal Application Access Token	Credential-Access

Sources

Source	URL	Tier
Non■Human Identities Are a Growing AI Security Risk — Here's Why	https://blog.lastpass.com/posts/non-human-identities	T3

Source	URL	Tier
Third-Party Cyber Risk: Why Vendor Attacks Are Rising and How ...	https://www.cybermaxx.com/resources/third-party-cyber-risk-why-vend...	T3
10 Cyber Security Trends For 2026 - SentinelOne	https://www.sentinelone.com/cybersecurity-101/cybersecurity/cyber-s...	T3
Emerging Trends in Cybersecurity: A Holistic View on Current ... - PMC	https://pmc.ncbi.nlm.nih.gov/articles/PMC11073482/	T1
Non-human identities: Agentic AI's new frontier of cybersecurity risk	https://www.weforum.org/stories/2025/10/non-human-identities-ai-cyb...	T3

Generated 2026-05-30 14:01 UTC by TJS Security Command Center