INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-06 13:17 UTC

Deep-Dive Ransomware Activity for Instant Threat Intelligence: European Commission Suffers Data Breach

DATA BREACH | HIGH | CVSS 9.1

SCC Item ID	SCC-DBR-2026-0081
Type	Data Breach
Severity	HIGH
CVSS Base Score	9.1
Affected Products	European Commission internal cloud infrastructure; 42 internal clients; 29 EU entities; AWS services accessed via compromised API key; Trivy (open-source container scanning tool, supply-chain vector)
Published	2026-04-04
Discovery Source	Gemini

Executive Summary

A threat actor attributed by CERT-EU as TeamPCP breached European Commission cloud infrastructure by exploiting a compromised AWS API key obtained through a supply-chain attack targeting Trivy, an open-source container scanning tool used in CI/CD pipelines. Approximately 92 GB of compressed sensitive data was exfiltrated, affecting 42 internal clients and 29 EU entities. Organizations using Trivy or similar open-source scanning tools in cloud-connected pipelines face direct exposure if cloud credentials are embedded in scanning workflows.

Technical Analysis

Initial access vector: supply-chain compromise of Trivy (open-source container and artifact vulnerability scanner, maintained by Aqua Security), used to harvest AWS API keys embedded in CI/CD or scanning pipeline configurations. The compromised credential (CWE-798: Hard-coded/embedded credentials; CWE-522: Insufficiently Protected Credentials) enabled direct access to AWS-hosted Commission infrastructure. Lateral movement occurred via valid cloud account abuse (MITRE T1078.004: Valid Accounts, Cloud Accounts). Data collection targeted cloud storage (T1530: Data from Cloud Storage), and exfiltration was conducted via cloud services (T1567.002: Exfiltration to Cloud Storage). The supply-chain entry point maps to T1195.001 (Compromise Software Dependencies and Development Tools). Credential harvesting from cloud metadata or config files maps to T1552.005 (Cloud Instance Metadata API). No CVE has been assigned to this incident. The

qualitative severity rating of 'high' is assigned editorially based on impact scope (42 internal clients, 29 EU entities affected) and access scope (authenticated cloud credential compromise). Attribution confirmed by CERT-EU. No patch is available for Trivy as a discrete vulnerability; the risk is architectural: secrets exposure in scanning workflows.

Action Checklist

1. Step 1: Containment, Immediately audit all AWS IAM keys, service account tokens, and cloud credentials accessible to Trivy or any container/artifact scanning tool in your CI/CD pipeline. Rotate any key that has been exposed to a scanning tool's execution environment. Revoke and regenerate compromised credentials before investigating scope. Consult AWS IAM Access Analyzer and CloudTrail to identify actions taken with any suspect key.
2. Step 2: Detection, Review AWS CloudTrail logs for API calls originating from unexpected source IPs or service principals associated with Trivy execution environments. Query for anomalous S3 GetObject, ListBuckets, or IAM ListKeys activity from CI/CD runner IPs. Search pipeline logs for Trivy execution events occurring with outbound connections to endpoints outside Aqua Security official registries, public container registries (Docker Hub, ECR, GCR), and known artifact repositories in your environment. Look for large-volume data transfer events (S3 or equivalent) not associated with known automation jobs. IOC patterns: unexpected AWS API calls from scanner host IPs, credential use outside pipeline execution windows, bulk object enumeration or download events.
3. Step 3: Eradication, Remove all hardcoded or file-embedded AWS credentials from Trivy configurations, CI/CD pipeline definitions, and container image build contexts. Replace static API keys with short-lived credentials using IAM roles for EC2/ECS/Lambda or OIDC-based federation for CI/CD runners (GitHub Actions, GitLab CI, etc.). Apply least-privilege IAM policies to any identity used by scanning tools. Configure scanning tools with no data-plane read access to production storage. Update Trivy to the latest verified release from the official Aqua Security repository and validate image integrity via published checksums.
4. Step 4: Recovery and Monitoring, Validate that all rotated credentials are no longer referenced in any pipeline config, Dockerfile, or environment variable store. Confirm CloudTrail logging is active across all regions. Enable AWS GuardDuty and review findings for credential exfiltration indicators (Finding: UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration). Monitor for re-use of old credentials as a canary: if rotated keys generate new API calls, assume persistent access and escalate to full incident response. Verify Trivy execution environments are isolated from production data stores.
5. Step 5: Post-Incident, Conduct a secrets scanning sweep across all repositories and pipeline definitions using tools such as Gitleaks, TruffleHog, or AWS Secrets Manager audit features. Implement a secrets management solution (AWS Secrets Manager, HashiCorp Vault) as a systemic control. Establish a supply-chain risk policy requiring integrity verification (checksums, signed releases) for all open-source tools integrated into pipelines with cloud access. Map control gaps to NIST SP 800-161 (Supply Chain Risk Management) and NIST SP 800-53 controls SA-12, IA-5, and SC-28. Review open-source dependency inventory for tools with similar credential-adjacent access patterns.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to full enterprise incident response and engage legal/privacy counsel immediately if CloudTrail confirms S3 GetObject volume consistent with the 92 GB exfiltration baseline, if old rotated keys generate any post-rotation API success events indicating persistent access, or if data classification of exfiltrated S3 buckets confirms PII belonging to EU data subjects — triggering mandatory GDPR Article 33 notification to supervisory authorities within 72 hours.
Recovery Notes	Post-containment, monitor CloudTrail and S3 access logs daily for a minimum of 90 days for any re-enumeration of previously accessed bucket names, IAM discovery calls from new source IPs, or GuardDuty findings of type `UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS` — all of which would indicate TeamPCP retained knowledge from the initial reconnaissance and is re-entering with separately obtained credentials. Verify that OIDC-based federation or IAM role assumption is confirmed active for all CI/CD runners (GitHub Actions, GitLab CI) by reviewing the `AssumeRoleWithWebIdentity` event presence in CloudTrail for scanner job executions, confirming no static key fallback is occurring. Given that 29 EU entities were affected, maintain incident documentation in a format suitable for GDPR supervisory authority review for a minimum of 3 years per standard regulatory retention expectations.
Forensic Artifacts	AWS CloudTrail S3 data plane logs (must be separately enabled per bucket): look for `REST.GET.OBJECT` and `REST.GET.BUCKET` events attributed to the Trivy IAM key ID with a cumulative response size approaching 92 GB, originating from source IPs outside known CI/CD runner egress ranges — this is the primary exfiltration evidence trail left by TeamPCP's bulk data collection. | CI/CD pipeline execution logs (GitHub Actions workflow run logs at `.github/workflows/` level, or GitLab CI job traces under Project > CI/CD > Jobs): capture the exact Trivy scan job execution timestamps, environment variable context passed to the Trivy process, and any anomalous post-scan subprocess spawning or outbound network connections that co-occur with the scanner invocation — these logs establish the supply-chain injection point. | Git repository commit history for credential exposure: run `git log --all --full-diff -p -- '*.yml' '*.yaml' '**/Dockerfile' '**/.env' | grep -E 'AWS_|AKIA|SECRET'` across all repos to recover any AWS key IDs that were committed and deleted from HEAD but persist in object history — establishes the full exposure window predating the TeamPCP breach. | AWS IAM credential report (`aws iam generate-credential-report`) exported at time of discovery: captures access key IDs, creation dates, last-used timestamps, and last-used services for all IAM users — the delta between Trivy key last-used date and known pipeline execution windows identifies unauthorized out-of-band usage by the threat actor. | Trivy binary hash and provenance on each affected CI/CD runner: capture `sha256sum $(which trivy)` and compare against the official Aqua Security published checksums for the installed version — a mismatch confirms the supply-chain vector involved a tampered binary capable of harvesting and exfiltrating credentials at scan time, versus a configuration-only exposure of a legitimate binary.

Per-Action IR Details

Step 1: Containment — Immediately audit all AWS IAM keys, service account tokens, and cloud credentials accessible to Trivy or any container/artifact scanning tool in your CI/CD pipeline. Rotate any key that has been exposed to a scanning tool's execution environment. Revoke and regenerate compromised credentials before investigating scope. Consult AWS IAM Access Analyzer and CloudTrail to identify actions taken with any suspect key.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST IR-4 (Incident Handling), NIST AC-2 (Account Management), NIST IA-5 (Authenticator Management), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Export all IAM users and access keys using: `aws iam generate-credential-report && aws iam get-credential-report --output text --query Content | base64 -d > iam_credential_report.csv`. Cross-reference key LastUsedDate fields against Trivy execution windows in CI/CD pipeline logs. For teams without AWS Security Hub, run `aws cloudtrail lookup-events --lookup-attributes AttributeKey=Username,AttributeValue= --start-time --output json > cloudtrail_suspect.json` to enumerate actions taken by the compromised key before rotation. Revoke immediately with `aws iam delete-access-key --access-key-id --user-name `.

Evidence: Before rotating credentials, capture: (1) AWS CloudTrail event history for the specific IAM access key ID associated with Trivy's execution environment — export via `aws cloudtrail lookup-events --lookup-attributes AttributeKey=AccessKeyId,AttributeValue= --output json`; (2) IAM credential report showing key creation date, last rotation, and last-used service/region — this establishes the exposure window for TeamPCP's lateral movement; (3) Trivy config files (trivy.yaml, .trivyignore, CI/CD YAML definitions such as .github/workflows/*.yml or .gitlab-ci.yml) to confirm how credentials were passed into the scanner's execution context (env vars, mounted secrets, hardcoded values); (4) AWS Access Analyzer findings active at time of discovery, which may surface cross-account or external resource policies the compromised key accessed.

Step 2: Detection — Review AWS CloudTrail logs for API calls originating from unexpected source IPs or service principals associated with Trivy execution environments. Query for anomalous S3 GetObject, ListBuckets, or IAM ListKeys activity from CI/CD runner IPs. Search pipeline logs for Trivy execution events co-occurring with outbound connections to non-standard endpoints. Look for large-volume data transfer events (S3 or equivalent) not associated with known automation jobs. IOC patterns: unexpected AWS API calls from scanner host IPs, credential use outside pipeline execution windows, bulk object enumeration or download events.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

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), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Query CloudTrail without a SIEM using AWS CLI: `aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=ListBuckets --start-time 2025-01-01 --output json | jq '.Events[] | {Time: .EventTime, User: .Username, IP: .CloudTrailEvent | fromjson | .sourceIPAddress}'`. Pivot to S3 data plane logs (must be enabled separately in S3 bucket logging) and grep for GetObject volume spikes: `grep -E 'REST.GET.OBJECT' s3_access.log | awk '{print $1, $2, $8, $9}' | sort | uniq -c | sort -rn | head -50`. For pipeline-level correlation, parse GitHub Actions or GitLab CI job logs to extract Trivy execution timestamps and cross-reference against CloudTrail `sourceIPAddress` matching runner egress IPs. Use the free Sigma rule `aws_cloudtrail_iam_enumeration.yml` (SigmaHQ repository) to structure manual queries if forwarding to a log aggregator.

Evidence: Capture before analysis: (1) S3 server access logs from all buckets accessible to the compromised Trivy IAM key — look for `REST.GET.OBJECT` and `REST.GET.BUCKET` events totaling toward the 92 GB exfiltration volume, with `sourceIPAddress` values not matching known CI/CD runner egress ranges; (2) CloudTrail `ListBuckets`, `GetBucketLocation`, `GetObject`, `ListObjects`, and `IAM:ListAccessKeys` events attributed to the compromised key ID — these map to MITRE ATT&CK T1530 (Data from Cloud Storage) and T1087.004 (Cloud Account Discovery); (3) VPC Flow Logs or AWS Network Firewall logs showing outbound connections from CI/CD runner subnets to non-whitelisted external IPs during or immediately after Trivy scan job execution; (4) CI/CD pipeline execution logs (GitHub Actions workflow run logs, GitLab CI job traces) showing Trivy invocation timestamps and any anomalous post-scan subprocess or network activity; (5) AWS GuardDuty finding `UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS` if GuardDuty was active — this finding type specifically flags credential use from IPs external to AWS infrastructure.

Step 3: Eradication — Remove all hardcoded or file-embedded AWS credentials from Trivy configurations, CI/CD pipeline definitions, and container image build contexts. Replace static API keys with short-lived credentials using IAM roles for EC2/ECS/Lambda or OIDC-based federation for CI/CD runners (GitHub Actions, GitLab CI, etc.). Apply least-privilege IAM policies to any identity used by scanning tools — scanning

tools require no data-plane read access to production storage. Update Trivy to the latest verified release from the official Aqua Security repository and validate image integrity via published checksums.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST IA-5 (Authenticator Management), NIST CM-6 (Configuration Settings), NIST SA-12 (Supply Chain Risk Management), CIS 4.6 (Securely Manage Enterprise Assets and Software), CIS 4.7 (Manage Default Accounts on Enterprise Assets and Software), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Run Gitleaks across all repositories to locate embedded credentials: `gitleaks detect --source=./repo --report-format json --report-path gitleaks_findings.json`. For Docker image layers, use `docker history --no-trunc ` and `dive ` (free tool) to inspect each layer for embedded env vars or credential files baked into build context. Verify Trivy binary integrity without an enterprise tool: download the official Aqua Security release checksum file from `https://github.com/aquasecurity/trivy/releases` and validate with `sha256sum -c trivy__checksums.txt`. Enforce least-privilege by generating a minimal IAM policy using the AWS IAM Policy Simulator against only the actions Trivy legitimately requires (ECR image pull: `ecr:GetAuthorizationToken`, `ecr:BatchGetImage`) and apply with `aws iam put-user-policy`.

Evidence: Before eradication actions, preserve: (1) Snapshot of all CI/CD pipeline definition files (.github/workflows/, .gitlab-ci.yml, Jenkinsfile, Dockerfile) in their pre-remediation state as forensic evidence of how credentials were exposed to Trivy's execution context — this establishes root cause for the supply-chain entry vector; (2) Git history (`git log --all --full-history -- '**/*.yml' '**/*.yaml' '**/Dockerfile'`) to determine when credentials were first introduced and whether they appear in historical commits that persist even after deletion from HEAD; (3) Trivy binary hash from the affected CI/CD runner compared against the official Aqua Security published checksum to confirm whether the supply-chain compromise involved a tampered Trivy binary or simply credential harvesting from its execution environment.

Step 4: Recovery — Validate that all rotated credentials are no longer referenced in any pipeline config, Dockerfile, or environment variable store. Enable AWS GuardDuty and review findings for credential exfiltration indicators (Finding: UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration). Confirm CloudTrail logging is active across all regions. Monitor for re-use of old credentials as a canary — if rotated keys generate new API calls, assume persistent access and escalate to full incident response. Verify Trivy execution environments are isolated from production data stores.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-9 (Protection of Audit Information), NIST AU-11 (Audit Record Retention), NIST SC-28 (Protection of Information at Rest), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 8.2 (Collect Audit Logs)

Compensating: Implement rotated key canary monitoring without GuardDuty by creating a CloudWatch metric filter on CloudTrail: filter for `errorCode = InvalidClientTokenId` on the old key ID — any hit after rotation confirms external actor retry attempts. CLI setup: `aws cloudwatch put-metric-alarm --alarm-name OldKeyReuse --metric-name ErrorCount --namespace CloudTrailMetrics --statistic Sum --period 300 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --alarm-actions `. Validate environment variable stores using `printenv | grep -iE 'aws|key|secret|token'` on each CI/CD runner and confirm no old key IDs persist. Use `aws s3api get-bucket-logging --bucket ` to verify S3 access logging is re-enabled post-recovery on all buckets the compromised key could have accessed.

Evidence: Document for recovery validation: (1) CloudTrail evidence that the rotated (old) key ID generates zero successful API calls after rotation — any `ConsoleLogin` or service API success events post-rotation indicate the actor obtained additional persistent credentials beyond the originally identified key; (2) AWS Config snapshot confirming IAM policy changes applied during eradication are consistent across all regions and accounts in scope — TeamPCP's access spanned 42 internal clients, so multi-account scope must be verified; (3) S3 access log baseline for the 30 days post-recovery to detect re-enumeration attempts against previously accessed buckets, which would indicate the actor retained knowledge of bucket names and is probing with new credentials.

Step 5: Post-Incident — Conduct a secrets scanning sweep across all repositories and pipeline definitions using tools such as Gitleaks, TruffleHog, or AWS Secrets Manager audit features. Implement a secrets management solution (AWS Secrets Manager, HashiCorp Vault) as a systemic control. Establish a supply-chain risk policy requiring integrity verification (checksums, signed releases) for all open-source tools integrated into pipelines with cloud access. Map control gaps to NIST SP 800-161 (Supply Chain Risk Management) and NIST SP 800-53 SA-12, IA-5, and SC-28 controls. Review open-source dependency inventory for tools with similar credential-adjacent access patterns.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SA-12 (Supply Chain Risk Management), NIST IA-5 (Authenticator Management), NIST SC-28 (Protection of Information at Rest), NIST SI-2 (Flaw Remediation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Run TruffleHog across all Git history (not just HEAD) to catch credentials that were committed and subsequently deleted: `trufflehog git file://./repo --only-verified --json > trufflehog_results.json`. Build a lightweight open-source tool inventory by querying CI/CD pipeline definitions with: `grep -rE 'trivy|snyk|grype|anchore|syft|cosign|crane' .github/workflows/ .gitlab-ci.yml Jenkinsfile 2>/dev/null` — any scanner tool with AWS env var access is in scope for the same supply-chain risk profile as Trivy. For SBOM-based supply-chain controls without enterprise tooling, generate a CycloneDX SBOM for each pipeline tool using `syft -o cyclonedx-json` and store as a signed artifact in your repository for integrity baselining.

Evidence: Preserve for lessons-learned and potential regulatory reporting: (1) Complete Git log of all repositories showing when Trivy or equivalent scanner tooling was introduced into pipelines with AWS credential access — establishes the breach window for the 92 GB exfiltration and supports GDPR Article 33 breach notification timelines applicable to EU entities affected; (2) Inventory of all 29 EU entities and 42 internal clients confirmed in scope, with data classification for each to determine PII/personal data exposure requiring supervisory authority notification under GDPR; (3) Final AWS CloudTrail export covering the full suspected compromise window showing all API actions taken by TeamPCP via the compromised key — this is the evidentiary record for regulators, legal counsel, and law enforcement referral if pursued; (4) Pre- and post-remediation IAM policy comparison showing least-privilege gap that permitted Trivy's key to perform S3 enumeration and bulk download — documents the control failure for the post-incident review and supports SA-12 gap assessment.

Detection Guidance

Primary log sources: AWS CloudTrail (all regions), CI/CD pipeline execution logs, container registry access logs. Key behavioral indicators: (1) AWS API calls (especially s3:GetObject, s3:ListBucket, iam:ListAccessKeys, sts:GetCallerIdentity) originating from scanner host IPs outside scheduled pipeline windows; (2) bulk data enumeration or download events not correlated with known automation jobs; (3) API key usage from IP addresses inconsistent with your CI/CD runner infrastructure; (4) Trivy process making outbound connections to endpoints outside Aqua Security official registries, public container registries, or known artifact repositories during or after scans. CloudTrail query focus: filter by eventName in [ListBuckets, GetObject, ListAccessKeys, AssumeRole] where userAgent contains 'trivy' or source IP matches scanner hosts, flagging any calls outside expected execution windows. AWS GuardDuty findings to prioritize: UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.InsideAWS and Exfiltration:S3/ObjectRead.Unusual. Monitor the CERT-EU security advisory database for IOC updates.

Indicators of Compromise

Type	Value	Context	Confidence
ACTOR	`TeamPCP`	Threat actor attributed to this breach by CERT-EU	HIGH

Framework Mappings

MITRE-ATTACK

- **T1552.005** — Cloud Instance Metadata API
- **T1530** — Data from Cloud Storage
- **T1078.004** — Cloud Accounts
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1567.002** — Exfiltration to Cloud Storage

OWASP-TOP10-2021

- **A04:2021** — Insecure Design
- **A07:2021** — Identification and Authentication Failures
- **A06:2021** — Vulnerable and Outdated Components

NIST-800-53R5

- **IA-5** — Authenticator Management
- **SA-4** — Acquisition Process
- **SA-9** — External System Services
- **CP-9** — System Backup
- **IR-4** — Incident Handling

CIS-V8

- **5.2** — Use Unique Passwords
- **16.4** — Establish and Manage an Inventory of Third-Party Software Components
- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.3** — Require MFA for Externally-Exposed Applications

HIPAA-SECURITY

- **164.308(a)(5)(ii)(D)** — Password Management
- **164.308(a)(7)(ii)(A)** — Data Backup Plan
- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(6)(ii)** — Response and Reporting

ISO-27001-2022

- **A.8.28** — Secure coding

- **A.5.29** — Information security during disruption
- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information
- **A.5.23** — Information security for use of cloud services

NIST-CSF-2

- **RS.MI-01** — Incidents are contained
- **RS.CO-03** — Recovery activities and progress communicated

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC7.4** — Responds to identified security incidents

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1552.005	Cloud Instance Metadata API	Credential-Access
T1530	Data from Cloud Storage	Collection
T1078.004	Cloud Accounts	Defense-Evasion
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1567.002	Exfiltration to Cloud Storage	Exfiltration

Sources

Source	URL	Tier
European Commission cloud breach: a supply-chain compromise	https://cert.europa.eu/blog/european-commission-cloud-breach-trivy-...	T1
European Commission Confirms Data Breach Linked to Trivy Supply ...	https://www.securityweek.com/european-commission-confirms-data-brea...	T3
European Commission breached after hackers poisoned open ...	https://thenextweb.com/news/european-commission-breach-trivy-supply...	T3
European Commission breach exposed data of 30 EU entities ...	https://securityaffairs.com/190333/security/european-commission-bre...	T3

Source	URL	Tier
European Commission Data Breach Linked to Trivy Supply Chain ...	https://www.reddit.com/r/pwnhub/comments/1scdwmh/european_commissio...	T3

Generated 2026-04-06 13:17 UTC by TJS Security Command Center