INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-05 08:19 UTC

# AWS IAM Credential Exposure Enables Automated SES-Based Phishing and BEC Campaigns

**THREAT CAMPAIGN** | **HIGH** | CVSS 7.5

| | |
|---|---|
| **SCC Item ID** | SCC-CAM-2026-0272 |
| **Type** | Threat Campaign |
| **Severity** | HIGH |
| **CVSS Base Score** | 7.5 |
| **Affected Products** | Amazon Web Services, Amazon SES (all regions), AWS IAM (access key credentials); DocuSign (impersonated brand) |
| **Published** | 2026-05-04T16:03:28 |
| **Discovery Source** | Rss |

## Executive Summary

Threat actors are harvesting exposed AWS IAM access keys from public code repositories, then using those keys to send phishing and business email compromise emails through Amazon's own Simple Email Service (SES). Because SES is a legitimate AWS service, these emails pass all standard email authentication checks, SPF, DKIM, and DMARC, making them indistinguishable from trusted senders at the email gateway layer. Organizations using AWS SES or those whose employees receive DocuSign-branded communications face elevated phishing risk that cannot be blocked by conventional email filtering alone.

## Technical Analysis

Attack chain: Automated secret-scanning tools (notably TruffleHog) harvest plaintext AWS IAM access keys from public GitHub repositories and misconfigured cloud storage. Actors then call SES APIs directly using those keys to dispatch high-volume phishing and BEC campaigns. Because SES sends from AWS-controlled infrastructure, outbound messages inherit valid SPF alignment, DKIM signatures, and DMARC pass status, bypassing authentication-based filtering at the recipient's mail gateway. No CVE is assigned; the vulnerability is a credential hygiene and cloud configuration failure, not a software defect. Relevant CWEs: CWE-522 (Insufficiently Protected Credentials), CWE-312 (Cleartext Storage of Sensitive Information), CWE-287 (Improper Authentication), CWE-359 (Exposure of Private Information). MITRE ATT&CK coverage includes T1552.001 (Credentials in Files), T1528 (Steal Application Access Token), T1078.004 (Valid Accounts: Cloud

Accounts), T1566.002 (Spearphishing Link), T1586.002 (Compromise Email Accounts), T1534 (Internal Spearphishing), T1071.003 (Application Layer Protocol: Mail), T1114 (Email Collection), and T1598 (Phishing for Information). Impersonated brands include DocuSign. Kaspersky (Securelist) documented measurable volume increases in this activity. Source: https://securelist.com/amazon-ses-phishing-and-bec-attacks/119623/

## Action Checklist

1. Step 1: Containment. Audit all AWS IAM access keys with SES:SendEmail or SES:SendRawEmail permissions immediately. Revoke any keys that appear in public repositories (search GitHub, GitLab, and internal repos using TruffleHog or git-secrets). Disable SES sending for any identity you cannot verify as actively in-use. Console path: IAM > Access Keys > Last Used column identifies stale or anomalous keys.
2. Step 2: Detection. Query AWS CloudTrail for SES API calls (SendEmail, SendRawEmail, SendBulkEmail) from IAM principals that do not match expected sending applications or IP ranges. Alert on SES API calls originating from unexpected AWS regions, unfamiliar IP addresses, or access keys with no recent prior SES activity. Also review SES sending statistics in the SES console for unexplained volume spikes. Log source: CloudTrail (eventSource: ses.amazonaws.com). Cross-reference IAM Access Analyzer findings for public key exposure.
3. Step 3: Eradication. Rotate all exposed or suspect IAM access keys immediately using the IAM console (IAM > Users > Security Credentials > Create Access Key, then deactivate and delete the old key). Apply least-privilege IAM policies: SES sending permissions should be scoped to specific identities and regions, not granted broadly. Enable AWS Secrets Manager or SSM Parameter Store for application credential storage; remove hardcoded credentials from all codebases and CI/CD configurations.
4. Step 4: Recovery. After key rotation, confirm SES sending volume returns to baseline using SES sending statistics and CloudTrail. Verify no unauthorized SES sending identities (domains/email addresses) remain verified in the SES console (SES > Verified Identities). Re-scan repositories with TruffleHog post-remediation to confirm no residual key exposure. Notify downstream email recipients or partners if BEC activity is confirmed.
5. Step 5: Post-Incident. Implement pre-commit hooks (git-secrets, detect-secrets) to block credential commits at the developer workstation level. Configure IAM policies to restrict SES permissions to dedicated service accounts; human users should not hold SES API keys. Establish periodic automated secret scanning of all repositories as a standing security control. Brief security awareness program on DocuSign impersonation and BEC indicators for finance and HR teams most frequently targeted by BEC.

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | IMMEDIATE |
| **Escalation Criteria** | Escalate to CISO, Legal, and external IR retainer immediately if CloudTrail confirms SES sends were delivered to customers, partners, or employees containing financial wire-transfer instructions or credential-harvesting links (BEC confirmed), or if the exposed IAM key had permissions beyond SES — particularly sts:AssumeRole, iam:*, or s3:GetObject on buckets containing PII — which would trigger breach notification obligations under GDPR, CCPA, or state-level statutes. |

| | |
|---|---|
| **Recovery Notes** | After key rotation and SES identity cleanup, maintain elevated CloudTrail and SES sending statistics monitoring for a minimum of 14 days to detect any secondary keys or roles the attacker may have provisioned during the dwell period that were not identified in the initial audit. Verify that all application workloads dependent on the rotated key have resumed normal operation by comparing SES sending volume and bounce/complaint rates against the 30-day pre-incident baseline. If BEC activity is confirmed, coordinate with recipient organizations' security teams to identify and retract any fraudulent payment instructions, and preserve all CloudTrail SES event records with recipient lists for potential law enforcement referral. |
| **Forensic Artifacts** | AWS CloudTrail S3 logs filtered on eventSource=ses.amazonaws.com for the exposed AKID — reveals full timeline of SendEmail/SendRawEmail/SendBulkEmail calls including sourceIPAddress, userAgent, recipient addresses (requestParameters.destinations), and message subjects, directly attributing the phishing campaign volume to the compromised credential \| SES Sending Statistics DataPoints (aws ses get-send-statistics) — per-15-minute DeliveryAttempts, Bounces, Complaints, and Rejects during the incident window, establishing total campaign scale and identifying the precise onset timestamp when unauthorized sending began \| IAM Credential Report and GetAccessKeyLastUsed records for the exposed AKID — documents the key creation date, the last legitimate use, and the first anomalous use, defining the attacker's dwell time window and the geographic/service context of the compromise \| SES Verified Identities list (aws ses list-identities) captured at time of detection — identifies any sending domains or email addresses the attacker added to the account to expand the phishing infrastructure beyond the victim organization's legitimate SES identity \| Git repository commit history and TruffleHog scan output showing the specific commit, file path, line number, and timestamp of the AKID exposure — establishes root cause, exposure duration (time from commit to key revocation), and the developer account responsible for the commit for post-incident process remediation |

### Per-Action IR Details

**Step 1: Containment — Audit all AWS IAM access keys with SES:SendEmail or SES:SendRawEmail permissions immediately. Revoke any keys that appear in public repositories (search GitHub, GitLab, and internal repos using TruffleHog or git-secrets). Disable SES sending for any identity you cannot verify as actively in-use. Console path: IAM > Access Keys > Last Used column identifies stale or anomalous keys.**

**NIST Phase:** Containment

**Reference:** NIST 800-61r3 §3.3 — Containment Strategy: stop ongoing harm by isolating the compromised credential before additional SES-based phishing volume is generated

**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 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

**Compensating:** Without a CSPM tool, run TruffleHog directly against all repos: `trufflehog git https://github.com/your-org/repo --only-verified`. Cross-reference discovered key IDs against IAM using AWS CLI: `aws iam list-access-keys --user-name ` and `aws iam get-access-key-last-used --access-key-id `. To immediately halt SES abuse without revoking the key (for evidence preservation), use `aws ses put-account-sending-attributes --no-sending-enabled` or scope a deny policy: `aws iam put-user-policy --user-name --policy-name BlockSES --policy-document '{"Version":"2012-10-17","Statement":[{"Effect":"Deny","Action":"ses:*","Resource":"*"}]}'`.

**Evidence:** Before revoking any key, capture: (1) full CloudTrail event history for the suspect AKID — run `aws cloudtrail lookup-events --lookup-attributes AttributeKey=AccessKeyId,AttributeValue= --start-time ` and export to JSON; (2) SES sending statistics snapshot from `aws ses get-send-statistics` showing per-15-minute DataPoints for volume anomalies; (3) IAM credential report (`aws iam generate-credential-report && aws iam get-credential-report`) documenting key creation date, last used date/region/service for all keys; (4) SES verified identities list (`aws ses list-identities`) to record any domains or addresses the attacker may have added; (5) screenshot or export of IAM

Access Analyzer external access findings to document the public exposure event.

**Step 2: Detection — Query AWS CloudTrail for SES API calls (SendEmail, SendRawEmail, SendBulkEmail) from IAM principals that do not match expected sending applications or IP ranges. Alert on SES API calls originating from unexpected AWS regions, unfamiliar IP addresses, or access keys with no recent prior SES activity. Also review SES sending statistics in the SES console for unexplained volume spikes. Log source: CloudTrail (eventSource: ses.amazonaws.com). Cross-reference IAM Access Analyzer findings for public key exposure.**

**NIST Phase:** Detection Analysis

**Reference:** NIST 800-61r3 §3.2 — Detection and Analysis: correlate CloudTrail SES telemetry against IAM principal baselines to determine scope of credential misuse and identify all sending activity attributable to the exposed key

**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), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs)

**Compensating:** Without a SIEM, run this AWS CLI query to extract all SES API calls from CloudTrail logs stored in S3: `aws s3 cp s3:/// ./ct-logs/ --recursive --include '*.json.gz'` then decompress and grep: `zcat *.json.gz | python3 -c "import sys,json; [print(json.dumps(e)) for l in sys.stdin for e in json.loads(l).get('Records',[]) if e.get('eventSource')=='ses.amazonaws.com'" > ses_events.json`. Alternatively, use AWS CloudTrail Insights (no additional cost if already enabled) to surface anomalous SES API call rates. For a free Sigma-based approach, apply the Sigma rule 'aws_cloudtrail_ses_sendmail_unusual_caller.yml' (available in the SigmaHQ repository) using the sigmac converter against CloudWatch Logs Insights.

**Evidence:** Collect before pivoting to eradication: (1) CloudTrail JSON records for all `ses.amazonaws.com` events scoped to the suspect AKID, including `sourceIPAddress`, `userAgent`, `requestParameters.destinations`, and `responseElements` fields — these reveal recipient addresses targeted in the BEC campaign; (2) SES sending statistics (`aws ses get-send-statistics`) DataPoints showing timestamps correlated to the key compromise window; (3) CloudTrail `ConsoleLogin` and `GetSessionToken` events for the same principal to determine if the attacker also achieved console access beyond API abuse; (4) IAM Access Analyzer findings export (`aws accessanalyzer list-findings`) identifying which public repository triggered the external access finding; (5) SES event destinations and configuration sets (`aws ses list-configuration-sets`) to identify if the attacker created new routing rules to suppress bounce notifications.

**Step 3: Eradication — Rotate all exposed or suspect IAM access keys immediately using the IAM console (IAM > Users > Security Credentials > Create Access Key, then deactivate and delete the old key). Apply least-privilege IAM policies — SES sending permissions should be scoped to specific identities and regions, not granted broadly. Enable AWS Secrets Manager or SSM Parameter Store for application credential storage; remove hardcoded credentials from all codebases and CI/CD configurations.**

**NIST Phase:** Eradication

**Reference:** NIST 800-61r3 §3.4 — Eradication: remove the attacker's means of access (the exposed AKID) and eliminate the structural conditions (hardcoded credentials, overly broad SES permissions) that enabled the campaign

**Controls:** NIST IA-5 (Authenticator Management), NIST AC-6 (Least Privilege), NIST SI-2 (Flaw Remediation), NIST CM-6 (Configuration Settings), CIS 5.2 (Use Unique Passwords), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

**Compensating:** Rotate the exposed key via CLI without console access: `aws iam create-access-key --user-name ` (capture new key), then `aws iam update-access-key --access-key-id --status Inactive`, update the application credential reference, verify application health, then `aws iam delete-access-key --access-key-id `. Apply a restrictive inline SES policy scoped to a single verified sending identity and region: `aws iam put-user-policy --user-name --policy-name SES-Least-Privilege --policy-document '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Action":["ses:SendEmail","ses:SendRawEmail"],"Resource":"arn:aws:ses:us-east-1::identity/yourdomain.com"}]}'`. For secrets migration without AWS Secrets Manager cost, use AWS SSM Parameter Store SecureString (free tier) and update application environment variables or config files to reference the parameter ARN.

**Evidence:** Before deleting the old key, preserve: (1) complete IAM policy document attached to the compromised user/role at time of incident (`aws iam list-attached-user-policies` and `aws iam get-policy-version`) to document the

over-privileged SES scope that enabled mass sending; (2) git log or repository commit history showing the commit that introduced the hardcoded AKID — capture commit hash, author, timestamp, and the specific file path for the incident record; (3) final CloudTrail `DeleteAccessKey` event record confirming the AKID was destroyed, with timestamp, to establish the eradication completion timestamp for the incident timeline; (4) SES verified identities list post-eradication (`aws ses list-identities`) to confirm no attacker-added sending domains remain.

**Step 4: Recovery — After key rotation, confirm SES sending volume returns to baseline using SES sending statistics and CloudTrail. Verify no unauthorized SES sending identities (domains/email addresses) remain verified in the SES console (SES > Verified Identities). Re-scan repositories with TruffleHog post-remediation to confirm no residual key exposure. Notify downstream email recipients or partners if BEC activity is confirmed.**

**NIST Phase:** Recovery

**Reference:** NIST 800-61r3 §3.5 — Recovery: restore normal SES sending operations only after confirming the attacker's foothold is fully removed and monitoring confirms no residual unauthorized activity

**Controls:** NIST IR-4 (Incident Handling), NIST IR-6 (Incident Reporting), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.2 (Establish and Maintain a Remediation Process)

**Compensating:** Establish a 72-hour monitoring window post-key rotation: schedule an hourly cron job that runs `aws ses get-send-statistics --output json | python3 -c "import sys,json; data=json.load(sys.stdin); [print(dp) for dp in data['SendDataPoints'] if dp['DeliveryAttempts']>0]" >> ses_baseline_monitor.log` and compare output against the pre-incident daily average. For BEC notification triage without a dedicated IR communication tool, query CloudTrail `requestParameters.destinations` fields from the SES send events to extract the full recipient list, deduplicate, and draft targeted partner/customer notifications. Re-run TruffleHog in verified-only mode to reduce noise: `trufflehog git https://github.com/your-org/repo --only-verified --json >> tufflehog_post_remediation.json`.

**Evidence:** Collect to close the recovery phase: (1) SES sending statistics export showing return to pre-incident volume baseline, with timestamps, as the formal recovery completion record; (2) exported list of all currently verified SES identities post-cleanup (`aws ses list-identities --identity-type Domain` and `--identity-type EmailAddress`) to confirm attacker-added identities were removed; (3) TruffleHog post-remediation scan output confirming zero verified findings for the previously exposed AKID pattern; (4) CloudTrail events for `ses:DeleteIdentity` actions confirming removal of any unauthorized verified sending identities added by the attacker.

**Step 5: Post-Incident — Address the structural control gap: implement pre-commit hooks (git-secrets, detect-secrets) to block credential commits at the developer workstation level. Enforce IAM policies that deny SES permissions to human users; sending should be service-account-only. Establish periodic automated secret scanning of all repositories as a standing security control. Brief security awareness program on DocuSign impersonation and BEC indicators for finance and HR teams most frequently targeted by BEC.**

**NIST Phase:** Post Incident

**Reference:** NIST 800-61r3 §4 — Post-Incident Activity: conduct lessons-learned review and implement systemic controls that prevent IAM credential exposure from recurring as the root cause of future SES-based phishing campaigns

**Controls:** NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-2 (Flaw Remediation), NIST SA-15 (Development Process, Standards, and Tools), NIST AT-2 (Literacy Training and Awareness), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

**Compensating:** Install detect-secrets as a pre-commit hook at zero cost: `pip install detect-secrets && detect-secrets scan > .secrets.baseline && pre-commit install` with the following `.pre-commit-config.yaml` entry: `- repo: https://github.com/Yelp/detect-secrets, rev: v1.4.0, hooks: [{id: detect-secrets, args: ['--baseline', '.secrets.baseline']}]`. Enforce the human-user SES deny at the AWS Organizations SCP level (free): create a Service Control Policy with `{"Effect":"Deny","Action":"ses:*","Resource":"*","Condition":{"StringNotEquals":{"aws:PrincipalType":"Service"}}}` applied to all OUs. For the BEC awareness brief, use CISA's free 'Phishing Guidance: Stopping the Attack Cycle at Phase One' resource and include concrete DocuSign impersonation indicators: sender domains not matching

docusign.com, envelope IDs that do not validate at docusign.com/signing/emailtools, and SES-sourced emails with `amazonses.com` in the Return-Path header.

**Evidence:** Document for the lessons-learned record and future detection engineering: (1) the original git commit hash and repository path where the AKID was exposed, as the root-cause artifact; (2) IAM policy document showing the overly broad SES permissions that existed at time of incident, to justify the least-privilege remediation; (3) CloudTrail event records spanning the full attacker dwell time (first unauthorized SES call to key deactivation) to quantify campaign duration and total emails sent; (4) SES sending statistics DataPoints for the incident window documenting total delivery attempts attributable to the attacker, for breach notification scoping if PII was present in BEC lure content.

## Detection Guidance

Primary detection surface is AWS CloudTrail. Query for eventSource = 'ses.amazonaws.com' with eventNames SendEmail, SendRawEmail, or SendBulkEmail. Filter for: (1) API calls from IAM principals not associated with known sending applications, (2) calls originating from IP addresses outside your known application infrastructure, (3) calls using access keys with a 'Last Used' timestamp inconsistent with normal application behavior, (4) sudden increases in SES send volume relative to 30-day baseline. Secondary: use IAM Access Analyzer to surface any access keys exposed in public S3 buckets or public repository integrations. On the email recipient side, SOC teams should note that these emails will pass SPF, DKIM, and DMARC; detection must rely on content analysis, link analysis, sender reputation at the sending IP level (not domain), and behavioral anomaly detection (e.g., DocuSign-branded email not originating from docusign.com or docusign.net). SIEM teams can create a rule: SES API call volume per IAM principal exceeds [baseline + 2 standard deviations] within a 1-hour window. Organizations lacking a 30-day SES sending baseline should establish one before deploying this alerting rule to avoid false positives.

## Indicators of Compromise

| Type | Value | Context | Confidence |
|---|---|---|---|
| DOMAIN | `docusign.com (impersonated)` | DocuSign brand impersonated in phishing emails sent via compromised SES accounts; legitimate domain used as lure — do not block, use for content-based detection | MEDIUM |
| URL | `https://securelist.com/amazon-ses-phishing-and-bec-attacks/119623/` | Kaspersky Securelist primary research source documenting this campaign — not a malicious IOC, reference only | HIGH |

## Framework Mappings

**MITRE-ATTACK**

- **T1114** — Email Collection
- **T1528** — Steal Application Access Token
- **T1566.002** — Spearphishing Link
- **T1586.002** — Email Accounts

- **T1534** — Internal Spearphishing
- **T1071.003** — Mail Protocols
- **T1598** — Phishing for Information
- **T1078.004** — Cloud Accounts
- **T1552.001** — Credentials In Files

**NIST-800-53R5**

- **AT-2** — Literacy Training and Awareness
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection
- **IA-5** — Authenticator Management
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-8** — Identification and Authentication (Non-Organizational Users)

**OWASP-TOP10-2021**

- **A04:2021** — Insecure Design
- **A07:2021** — Identification and Authentication Failures

**CIS-V8**

- **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
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks

**HIPAA-SECURITY**

- **164.308(a)(5)(ii)(D)** — Password Management
- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(5)(i)** — Security Awareness and Training

**SOC2-TSC**

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

**ISO-27001-2022**

- **A.5.34** — Privacy and protection of personal information
- **A.5.23** — Information security for use of cloud services

## MITRE ATT&CK Mapping

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1114 | Email Collection | Collection |
| T1528 | Steal Application Access Token | Credential-Access |
| T1566.002 | Spearphishing Link | Initial-Access |
| T1586.002 | Email Accounts | Resource-Development |
| T1534 | Internal Spearphishing | Lateral-Movement |
| T1071.003 | Mail Protocols | Command-And-Control |
| T1598 | Phishing for Information | Reconnaissance |
| T1078.004 | Cloud Accounts | Defense-Evasion |
| T1552.001 | Credentials In Files | Credential-Access |

## Sources

| Source | URL | Tier |
|---|---|---|
| **Security News** | https://www.bleepingcomputer.com/news/security/amazon-ses-increasin... | T3 |
| **Amazon SES increasingly abused in phishing to evade detection** | https://www.reddit.com/r/cybersecurity/comments/1t48hy1/amazon_ses_... | T3 |
| **Phishing campaigns and BEC attacks through Amazon SES** | https://securelist.com/amazon-ses-phishing-and-bec-attacks/119623/ | T3 |
| **Attackers Exploit Amazon SES to Send Authenticated Phishing Emails** | https://gbhackers.com/attackers-exploit-amazon-ses/ | T3 |
| **Amazon SES Phishing and BEC Attacks Leverage Leaked AWS IAM ...** | https://www.technadu.com/amazon-ses-phishing-and-bec-attacks-levera... | T3 |

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-05-05 08:19 UTC by TJS Security Command Center