CYBER
THREAT
ANALYSIS

Recorded Future®

Insikt Group®

February 19, 2026

2025 Cloud Threat Hunting and Defense Landscape

Cloud threat actors are continuing to diversify the ways they gain access to cloud environments, including by compromising third-party vendors and exploiting edge technologies to obtain initial cloud access.

Cloud ransomware is a growing issue, as evidenced by the inventive ways threat actors are abusing legitimate cloud services to destroy and encrypt victim data hosted in cloud environments.

All instances where threat actors penetrated beyond the edge of a cloud environment involved valid accounts and credential abuse.

Executive Summary

Insikt Group has observed continued trends of growth and increased activity of threat actors leveraging and exploiting cloud infrastructure to broaden the number of victims they target and infect. Recent reporting across the observed incidents shows that cloud-focused threats are converging on a few consistent patterns, which serve as the main sections of this report:

- Exploitation and Misconfiguration
- Cloud Abuse
- Cloud Ransomware
- Credential Abuse, Account Takeover, and Unauthorized Access
- Third-Party Compromise

Across cases, initial access frequently comes from vulnerable or misconfigured services exposed to the internet — including application delivery controllers, monitoring dashboards, email security gateways, and enterprise resource planning (ERP) platforms — as well as stolen or weakly governed credentials sourced from public leaks, compromised developer workstations, and socially engineered helpdesk workflows. Once inside a targeted environment, threat actors systematically pivot through hybrid identity and virtual private network (VPN) infrastructure, targeting directory-synchronized accounts, non-human and executive identities, and privileged cloud roles to gain tenant-wide administrative control.

Post-compromise activity is characterized by heavy use of built-in cloud and SaaS functionality: enumerating and exfiltrating data via native storage and backup services, destroying or encrypting cloud backups and snapshots for impact, manipulating static frontends and continuous integration/continuous deployment (CI/CD) pipelines to subvert trust in applications and repositories, and using mainstream platforms such as calendar services as covert command-and-control (C2) channels.

In comparison to its previous iteration, the majority of the events discussed in this report indicate that threat actors are engaging in similar threat behaviors; however, there are three specific trends that appear to have emerged since the most recent iteration:

- Cloud threat actors are registering their own legitimate cloud resources for use in attack chains.
- DDOS attacks are becoming less effective when targeting cloud environments, even in instances of record-breaking throughput, due to increased cloud-native capabilities for mitigating these threats.
- Cloud threat actors are increasingly diversifying the types of services that they target in victim environments during an attack chain, with a notable focus on LLM and other AI-powered services hosted in cloud environments.

The trends associated with abuse indicate a shift in threat actor perception, demonstrating that threat actors are exploring the broader benefits that compromised cloud services can provide.

Key Findings

- The threat of exploitation and misconfiguration grows as the number of cloud services and third-party technologies embedded at the perimeter of cloud environments grows.
- Threat actors are displaying increased interest in the abuse of compromised victim cloud LLM and machine learning (ML) services, both for profit and as a method of post-compromise during attack chains.
- Threat actors use trusted identities and cloud-native tools to directly execute their end goals, such as ransomware, disruption, data destruction, and data theft, without relying on traditional malware.
- Instead of deploying large numbers of ransomware binaries, threat actors use compromised accounts, roles, tokens, and keys to change encryption settings, rotate or destroy key material, or mass-modify stored data through cloud application programming interfaces (APIs) and consoles.
- Compromising a third party or supply chain provides attackers with inherited trust and permissions, enabling them to inject malicious code, access data, and modify configurations across multiple cloud tenants.

Table of Contents

Background

The rapid adoption and continuous evolution of cloud services have created a distinct and expanding attack space threat actors actively exploit. As organizations continue to move core systems and sensitive data from traditionally isolated, on-premises environments into internet-accessible cloud platforms, adversaries treat these deployments as a centralized collection of valuable data and often-weak security controls. They take advantage of widespread misconfigurations, uneven cloud security expertise, and the uniformity of major cloud and SaaS platforms, using publicly available discovery tools to scan at scale for exposed assets and insecure settings. At the same time, threat actors increasingly leverage cloud infrastructure for their own operations, using commercial services to host malware, maintain command-and-control, and stage data exfiltration. The same flexibility, reach, and obscured attribution that cloud services provide to legitimate users enables threat actors to blend malicious traffic with legitimate activity.

Defenders, meanwhile, operate in an environment where cloud computing is mature in terms of adoption, but still early in its practical use and secure implementation. Security teams must contend with rapidly changing product offerings, constant feature updates, and complex, distributed architectures that make it difficult to maintain complete visibility and consistent controls across regions, tenants, and services. Many organizations adopt cloud technologies with limited in-house expertise, unrealistic expectations, or suboptimal implementations, and they frequently report shortages of senior personnel capable of securely architecting and managing these environments. Skill and knowledge gaps, combined with the scale and constant change of cloud environments, result in configuration drift, unmonitored assets, and security blind spots. As a result, defenders face a growing burden: protecting highly available, globally distributed systems that deliver business value but simultaneously increase the organization's attack surface and the likelihood of successful compromise.

Methodology

This report identifies five main threats to cloud environments, each of which is explored in its respective section:

- Exploitation and Misconfiguration
- Cloud Abuse
- Cloud Ransomware
- Credential Abuse, Account Takeover, and Unauthorized Access
- Third-Party Compromise

Each section includes radar charts that measure the following attributes associated with a given threat. These determinations were made by Insikt Group by investigating instances where a threat vector was observed to answer the following questions:

- **Cost of Impact:** How much would this threat cost a victim in terms of monetary, reputational, and operational losses? In the radar chart, the higher the number, the higher the cost the victim can expect to receive monetarily, reputationally, operationally, or otherwise.
- **Commonality:** How often is this threat vector observed in attack chains against cloud environments in the wild? In the radar chart, the higher the number, the more likely a cloud defender is to observe this behavior in their own environment.
- **Evolution Potential**: What is the potential for threat actors to further "evolve" this attack vector in terms of new tools, attack methods, and tactics, techniques, and procedures (TTPs) that can be employed to achieve this threat vector? In the radar chart, the higher the number, the more likely it is that threat actors will be able to perform actions demonstrating this threat in ways previously unobserved, thereby complicating the behavior's detection.
- **Effort to Perform:** What are the technical and monetary costs associated with performing this threat vector? In the radar chart, the higher the number, the greater the barrier for an attacker to demonstrate this threat against a cloud environment, generally in terms of monetary cost or technical capability.

Threats To Cloud Environments

Exploitation and Misconfiguration

Key Takeaways

- Exploitation and misconfiguration remain key threats at the edge of cloud environments.
- Exploitation in cloud environments most commonly occurs at the perimeter against integrated third-party products.
- Misconfiguration remains a consistent way for threat actors to gain initial access to cloud environments.

Figure 1 illustrates and compares attributes associated with cloud abuse. A description of each attribute can be found in the **Methodology** section of this report.

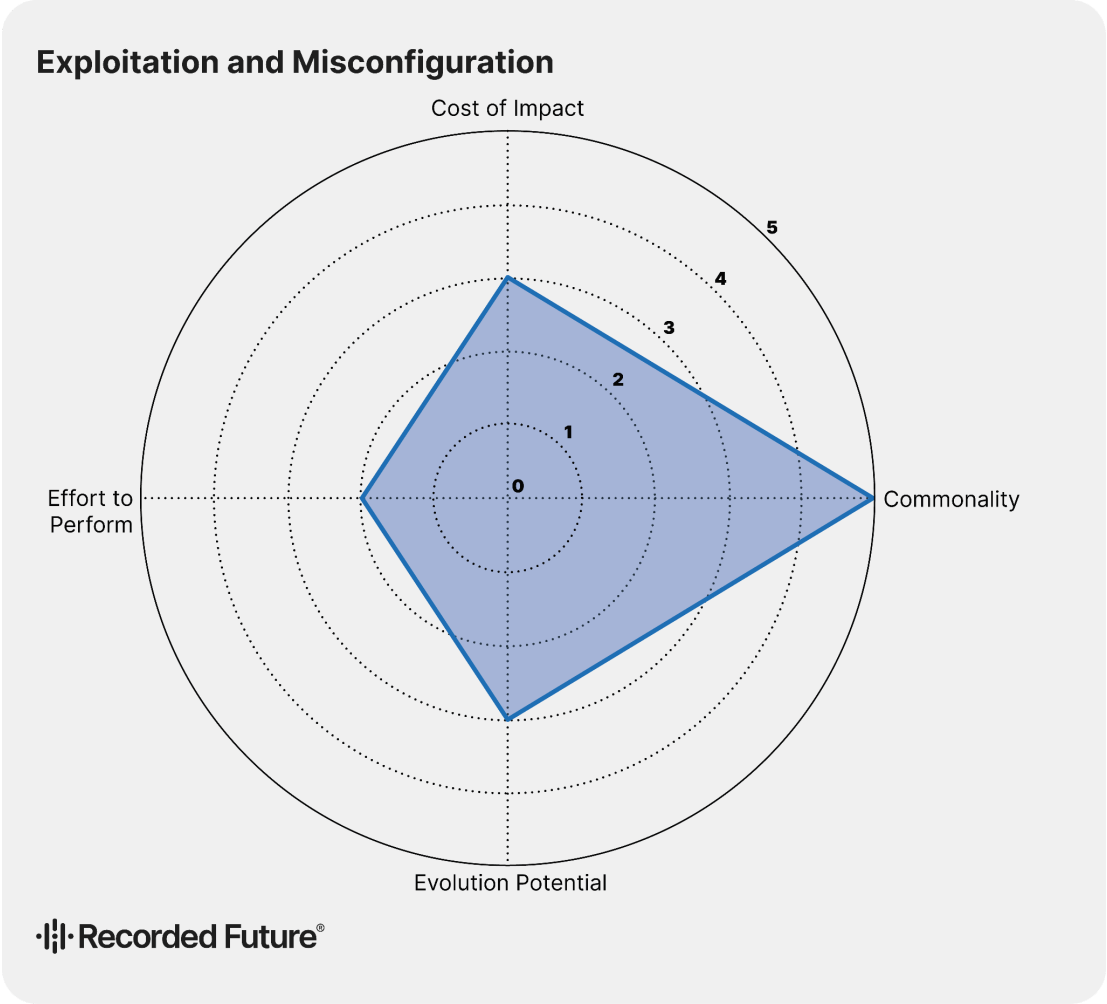

Figure 1: *Radar chart illustrating exploitation and misconfiguration as a threat vector (Source: Recorded Future)*

Cost of Impact: 3 (Moderate)

As previously determined by Insikt Group: "Successful exploitation of cloud infrastructure or the technologies embedded in it may result in multiple benefits to a threat actor, but may not directly translate to victim cost." This determination can also be applied to the risks posed by misconfigured cloud infrastructure, as discussed in this report. In both instances, while exploitation and misconfiguration often allow threat actors to gain initial access, additional internal access, or increased permissions within a cloud environment, the overall impact on a cloud environment is not explicitly tied to exploitability or a threat actor's ability to exploit misconfigurations.

Commonality: 5 (Very High)

Misconfigurations are a common risk in cloud environments, and threat research has consistently shown that they are frequently exploited by threat actors as an initial access vector. While not as commonly exploited for initial access, cloud environments inherit a myriad of vulnerability-related risks from third-party technologies often embedded within them.

Insikt Group additionally notes that multiple critical vulnerabilities were identified in cloud-native services within the past year; however, based on Insikt Group's research, these vulnerabilities are discovered and disclosed much less often than vulnerabilities in the technologies embedded in cloud infrastructure.

Evolution Potential: 3 (Moderate)

The risk presented by misconfigurations and vulnerability exploitation in cloud environments is bounded by the technologies and services implemented within it. As new technologies and services become available and existing ones undergo updates, new misconfiguration and exploitation risks may arise. At the same time, threat actors can only capitalize on these risks in the context of the vulnerable or misconfigured technology or service, limiting the overall evolution potential of this risk.

Effort to Perform: 2 (Low)

Identifying known misconfigurations and vulnerabilities externally is a straightforward process for threat actors. Numerous tools exist that allow threat actors to programmatically identify and, in some cases, automatically exploit these weaknesses. Vulnerability research is much more challenging than abusing and weaponizing open-source proof-of-concept (PoC) exploits; however, based on data collected by Recorded Future, scanning and exploitation attempts indicate that vulnerability and reconnaissance activity following a vulnerability disclosure is a significantly more likely threat to defenders in terms of volume.

Threat Summary

Data collected over the past year indicates that cloud misconfiguration and vulnerability exploitation continue to pose a critical risk to defenders, alongside common attack vectors that threat actors will exploit both within and outside cloud environments. These weaknesses enable threat actors to gain

access and establish false legitimacy in cloud environments, serving as a means to achieve their ultimate goals as they continually target the environment.

Misconfiguration is an issue that cloud providers, architects, and security experts continuously attempt to mitigate in cloud environments; however, this cycle of human misconfiguration remediation can also present misconfigurations. The human, operational, and financial costs associated with monitoring, identifying, and mitigating misconfigurations in cloud environments are high. Additionally, as cloud computing continues to grow, cloud-native service offerings and the infrastructure itself also continue to grow, leading to mitigation challenges created by an increased external attack surface. Defenders must remain aware of new iterations of cloud services to ensure that configuration best practices are followed.

Exploitation in cloud environments is not as commonly identified as instances of misconfigurations due to the managed nature of cloud service providers. However, threat reporting throughout the year demonstrates that exploitation occurs both within and at the perimeter of cloud environments.

Outlook

Threat actors are expected to continue focusing on cloud environments as attractive targets, leveraging a combination of configuration weaknesses and the exploitation of vulnerable public-facing technologies, particularly in the wake of high-profile vulnerability disclosures.

Based on current trends, exposure-driven risks, including configuration-related issues and unpatched or newly disclosed vulnerabilities, are likely to remain a consistent factor in cloud compromises, often enabling initial access while also impacting internal cloud services and supporting infrastructure.

As cloud adoption continues to accelerate, the volume and diversity of deployed services and technologies will also grow. This expansion increases the likelihood of both exploitable vulnerabilities and unintended exposures, reinforcing the importance of scalable security controls, timely patching, and continuous visibility across cloud environments.

Mitigations and Detections

Figure 2 illustrates a hypothetical attack chain involving misconfiguration and exploitation methods. Throughout this visual, Insikt Group has identified parts of the attack chain where defenders can most efficiently hunt for and mitigate behaviors associated with cloud misconfiguration and exploitation.

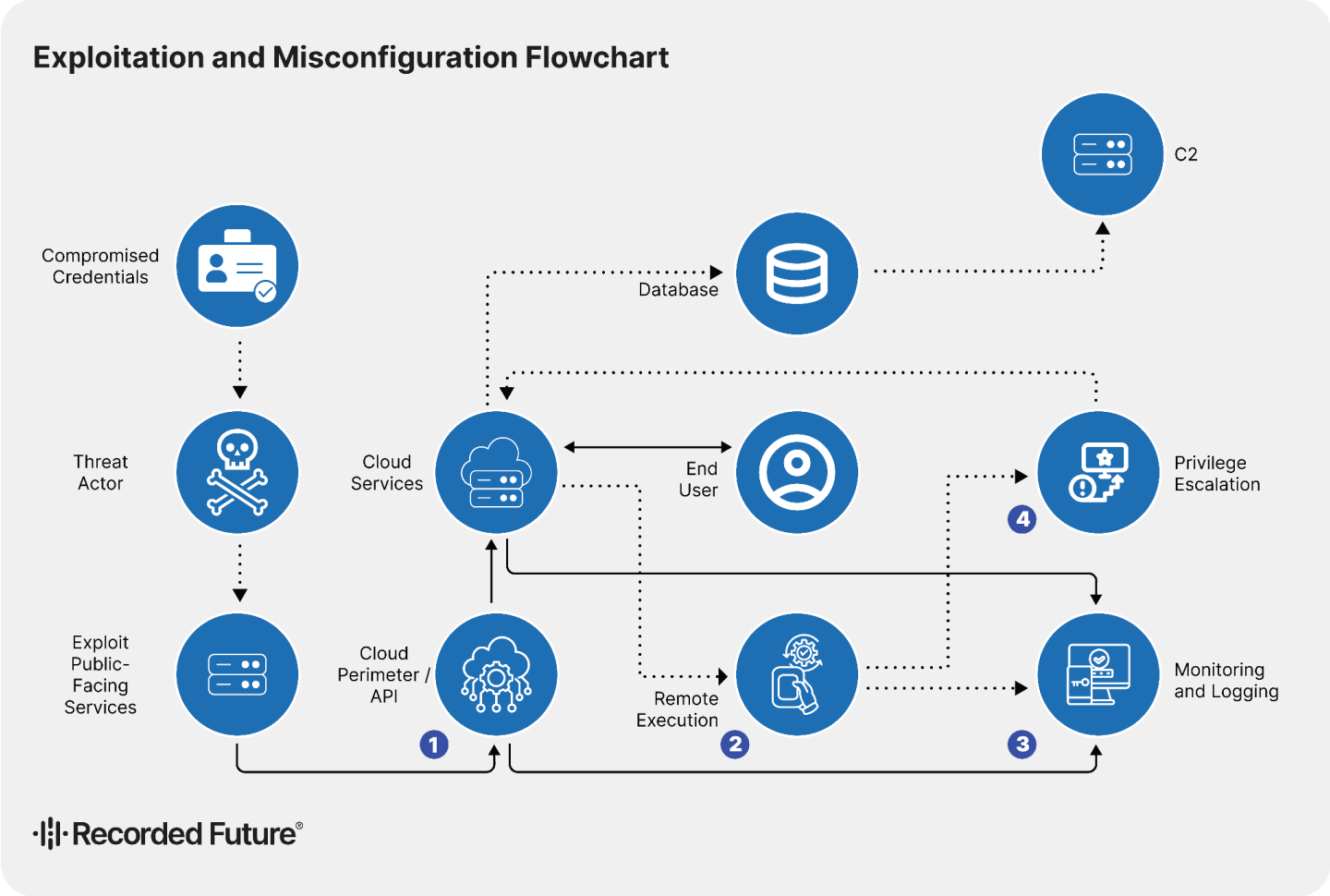

Figure 2: *Visual representation of potential cloud abuse attack vectors (Source: Recorded Future)*

① Exploit Attempts Targeting Public-Facing Cloud Services and Appliances

Unpatched or misconfigured internet-exposed services, such as application gateways, VPN portals, and web applications, are frequent targets for initial access. Successful exploitation can grant remote code execution, credential theft, and a foothold for lateral movement into cloud and on-premise resources.

Defenders can mitigate these threats in the following ways:

- Maintain an up-to-date inventory of public-facing services and prioritize rapid patching and hardening of edge systems.
- Restrict direct internet exposure of administrative interfaces by placing them behind a VPN, zero-trust network access (ZTNA), or IP allowlists.
- Use web application firewalls (WAFs) and reverse proxies to provide virtual patching, request validation, and detection of common exploit patterns.

- Analyze web and edge logs for scanning behavior, abnormal HTTP patterns, and indicators of exploit attempts, backed by intrusion detection systems (IDS) or intrusion prevention systems (IPS) where appropriate.

② Script-Based Post-Compromise Activity

After gaining a foothold, threat actors often rely on script-based execution, such as PowerShell and other interpreters, with obfuscation and in-memory techniques to download tools, disable defenses, and move laterally while avoiding obvious file-based indicators.

Defenders can mitigate these threats in the following ways:

- Restrict script execution and enforce the use of signed, approved scripts by applying application control to critical systems.
- Enable and centralize detailed script logging, such as script block logging and Antimalware Scan Interface (AMSI) output, for analysis and detection.
- Detect encoded, obfuscated, or unusually long command lines and scripts launched from unexpected parent processes.
- Use endpoint detection and response (EDR) to identify suspicious script behavior, including credential access, lateral movement, and tampering with security controls.

③ Unauthorized Remote Access Tools or Anomalous Binary Execution

Threat actors frequently deploy or repurpose remote access and management tools, as well as renamed or portable binaries, to maintain persistent and interactive access. These binaries often masquerade as legitimate activity and may use scheduled tasks or services for durability.

Defenders can mitigate these threats in the following ways:

- Monitor process execution for known RMM tools and binaries running from unusual or user-writable locations.
- Apply behavior-based EDR and application allowlisting to block unauthorized, unsigned, or renamed executables.
- Restrict installation and use of RMM and remote access software to approved tools and authorized administrators.
- Monitor for new or modified scheduled tasks, services, and startup entries referencing unfamiliar binaries or scripts.

④ Privilege Escalation through Identity Federation or Credential Manipulation

Cloud-focused campaigns increasingly abuse identity and federation mechanisms to escalate privileges and maintain long-lived access. By altering federation settings, abusing sync accounts, forging tokens, or creating backdoor identities, attackers can impersonate users, bypass certain multi-factor authentication (MFA) protections, and persist even after password changes.

Defenders can mitigate these threats in the following ways:

- Restrict and closely monitor who can modify federation settings, token configurations, and synchronization accounts, and require strong approvals for any changes.
- Enforce conditional access, MFA, or passkeys for all administrative and high-impact accounts and avoid legacy protocols that weaken these controls.
- Detect unusual token issuance, high-privilege app registrations, and the creation of new federated domains or privileged accounts using identity logs.
- Use identity threat detection capabilities to correlate identity, cloud, and directory events and surface unauthorized privilege elevation or token abuse.

Examples in the Wild

Insikt Group curated a list of events published within the past year that demonstrate the threats posed by exploitation and misconfiguration. These events are discussed below.

Threat Actors Abuse AzureHound for Post-Compromise Discovery in Microsoft Azure Environments

On October 24, 2025, Palo Alto Networks's Unit 42 published a report detailing how threat actors abuse AzureHound in Microsoft Azure environments. AzureHound is an open-source data collection tool initially developed by SpecterOps (@SpecterOps) for penetration testing within the BloodHound suite. Per Unit 42, threat actors, specifically the Iranian-linked group Curious Serpens, the suspected state-sponsored threat actor Void Blizzard, and the ransomware operator Storm-0501, repurpose AzureHound in post-compromise discovery phases to map Microsoft Entra ID (formerly Azure Active Directory) environments. These operations targeted Azure tenants across hybrid and multi-tenant environments as recently as August 2025, enabling threat actors to perform comprehensive discovery that facilitated lateral movement and privilege escalation.

Based on Unit 42's analysis, the infection chain begins when threat actors gain initial access to a target's Azure environment using stolen credentials or authentication tokens. Threat actors then use information-stealing malware, such as Raccoon Stealer and Redline, to acquire credentials and session tokens from the victim's browser. Using these stolen authentication artifacts, including usernames and passwords, refresh tokens, or JSON Web Tokens (JWTs), threat actors authenticate to the Azure environment. In some cases, they use multi-factor authentication (MFA) fatigue techniques to authenticate into the victim's Microsoft Entra ID environment. Once authenticated, threat actors use available tokens or service principal credentials to connect to Azure APIs.

After establishing access, threat actors deploy AzureHound to implement the following features and actions:

- Enumerate users, devices, and service principals to collect identity information from Microsoft Entra ID
- Extract ownership relationships between identities and resources to build a comprehensive identity graph

- Save enumerated user data to files like `users.json` for offline analysis and targeting of high-value accounts
- Filter users by job title or attributes to identify privileged roles such as “Global Administrator”
- Discover group memberships, roles, and role assignments to uncover potential privilege escalation paths
- Identify nested group structures and inherited permissions that may grant elevated access
- Enumerate app role assignments and key vault access policies to find excessive or misconfigured permissions
- List storage accounts and blob containers to locate data storage locations and assess exposure
- Collect storage account metadata, including names, endpoints, access control lists (ACLs), replication type, and custom domains
- Reveal network ACLs on storage accounts to identify allowed IP address ranges and trusted services
- Discover web, function, and logic applications, automation accounts, and Kubernetes clusters (AKS) to identify cloud services that may be misconfigured or vulnerable
- Enumerate container registries and automation pipelines to identify services that may enable code execution or lateral movement
- Identify tenants, subscriptions, management groups, resource groups, virtual machines (VMs), and key vaults to understand the cloud environment’s structure and target sensitive assets
- Bypass visibility by abusing Azure REST API calls, not recorded in Azure activity or resource logs by default
- Trigger test requests to Graph API, Azure REST API, and Microsoft identity platform endpoints using the default AzureHound user-agent string (for example, “azurehound/v2.6.0”)
- Export discovery output in JSON format for ingestion into BloodHound to visualize privilege escalation paths and relationships

Suspected Salt Typhoon Campaign Exploited Citrix NetScaler Gateway To Deliver SnappyBee Backdoor, Sample Available from Recorded Future Malware Intelligence

On October 20, 2025, cybersecurity firm Darktrace published a technical blog detailing a campaign against a European telecommunications organization. According to Darktrace, the campaign, which began in early July 2025, exploited a public-facing Citrix NetScaler Gateway appliance, likely CVE-2023-3519, for initial access and deployed SnappyBee (also known as Deed RAT), a variant of ShadowPad. CVE-2023-3519 is a critical remote code execution (RCE) vulnerability in Citrix Application Delivery Controller (ADC) and Citrix Gateway appliances. Darktrace attributes the campaign to the state-sponsored threat actor Salt Typhoon with moderate confidence.

After gaining access, the threat actor pivoted from the compromised NetScaler Gateway to internal Citrix Virtual Delivery Agent (VDA) hosts within the organization's Machine Creation Services (MCS) subnet. To obscure their infrastructure and complicate attribution, the threat actor routed their initial access through an endpoint associated with the SoftEther virtual private network (VPN) service. Once inside the network, the threat actor deployed SnappyBee as a dynamic-link library (DLL) file to multiple

internal Citrix VDA hosts, bundled alongside legitimate antivirus (AV) executables, such as Norton, Bkav, and IObit Malware Fighter, and executed it using DLL sideloading.

Microsoft's OneDrive File Picker Flaw Allows Web Applications Full Read Access to Users' Entire OneDrive

On May 28, 2025, Oasis Security published a technical blog detailing a flaw in Microsoft's OneDrive File Picker. File Picker is a web component that allows applications to access, upload, and download files from a user's OneDrive storage via OAuth authorization. File Picker (versions 6.0 to 7.2 using Implicit Flow and 8.0 using Microsoft Authentication Library) always requests the broad scopes `Files.Read.All`, `Files.ReadWrite.All`, and `offline_access`. As a result, even single-file upload consent silently grants apps full, persistent read and write access to the user's entire OneDrive.

According to Oasis Security, web applications such as ChatGPT, Slack, Trello, ClickUp, and others that support OneDrive file uploads may unintentionally receive full read access to a user's entire OneDrive account, even if a single file gets uploaded. Given the integration of OneDrive File Picker in numerous and diverse SaaS applications, the flaw is both widespread and trivially exploitable: a malicious app only needs to host the picker and then read the user's authorization token from `localStorage` or `sessionStorage` to enumerate or exfiltrate all files stored in the user's OneDrive account.

Oasis reported the flaw to Microsoft, which then notified vendors implementing OneDrive File Picker. At the time of writing, Microsoft has not yet issued a fix; however, per Oasis, Microsoft stated they are "considering future improvements, including more precise alignment between what OneDrive File Picker does and the access it requires."

The flaw stems from the File Picker's lack of fine-grained OAuth scopes. When a user authorizes a web application like ChatGPT or Slack to upload a file via OneDrive, the File Picker requests full read (and in some cases write) access to the entire OneDrive directory. This is due to the overly broad predefined scope permissions available in Microsoft Graph that cannot be customized for single-file access scenarios. Therefore, instead of exclusively granting access to a specific file, the OAuth tokens issued through the File Picker grant access beyond the selected file, extending to the user's entire OneDrive directory. These tokens often request the `offline_access` scope, allowing applications to retain access for extended periods beyond the user's session.

This design flaw increases the risk due to ambiguous consent prompts that suggest access remains limited to selected files; however, in reality, applications gain permissions without clear user awareness. In File Picker version 7.0, it requests both read and write permissions even in upload scenarios, and while version 8.0 delegates authentication to developers, it does not enforce narrower scopes. As a result, users may unknowingly expose sensitive data to third-party applications with far broader access than intended.

The second flaw stems from the insecure storage of sensitive authentication tokens, such as access and refresh tokens, particularly in older versions of the File Picker (6.0 to 7.2). These versions use the

"Implicit Flow" authorization method, which exposes access tokens via URL fragments and stores them as plaintext in the browser's `localStorage`, allowing malicious scripts or browser extensions to access them easily. In newer versions like 8.0, with authentication handled through Microsoft Authentication Library (MSAL), developers often store tokens in `sessionStorage` without encryption or safeguards, leading to significant security risks. These insecure storage practices allow threat actors with access to the browser context to easily hijack tokens and gain full read access to the OneDrive account.

Based on reporting associated with the flaw, threat actors could exploit this flaw in at least two ways:

- Malicious Web Pages: Threat actors could create a web page that uses the OneDrive File Picker and is embedded with malicious JavaScript. When users visit that page and authorize OneDrive File Picker, the attacker's JavaScript can retrieve the OAuth token from `localStorage` or `sessionStorage`, granting full read access to files stored in the user's OneDrive.
- Compromise Legitimate Web Pages: Threat actors could seek to identify vulnerable, legitimate web pages that implement OneDrive File Picker. For example, web pages vulnerable to cross-site scripting (XSS) could allow threat actors to inject malicious JavaScript into a trusted web application. Once the web page is compromised, threat actors can trigger the same OneDrive Picker Flow that waits for a user to consent to uploading at least one file, stealing their authorization token from `localStorage` or `sessionStorage`, and obtaining full read access to the user's OneDrive.

In either exploit scenario, minimal user interaction is required. Given the widespread adoption of the OneDrive File Picker across numerous collaboration and productivity platforms, the flaw is common, as any web app that embeds the OneDrive File Picker in upload mode is impacted.

WhoAMI Attack Exploits AMI Name Confusion to Compromise AWS Environments

On February 12, 2025, Datadog Security Labs published a write-up detailing the "WhoAMI" attack, which targets a name confusion vulnerability affecting Amazon Machine Images (AMIs) used in Amazon Web Services (AWS) environments. AMI is a pre-configured virtual machine template used to launch Elastic Compute Cloud (EC2) instances in AWS, containing the operating system, applications, and necessary configurations. EC2 is a scalable cloud computing service that provides virtual servers (instances) to run applications on AWS. According to Datadog Security Labs, which discovered the vulnerability in August 2024, threat actors can inject malicious AMIs into unsuspecting AWS accounts. Using the WhoAMI attack, threat actors can compromise a significant number of AWS environments, including those of major organizations. Notably, AWS itself had non-production systems vulnerable to this issue before implementing fixes. To mitigate this risk, AWS introduced the "Allowed AMIs" feature that allows users to restrict AMI selection to verified providers.

The vulnerability stems from improper filtering of AMI searches using the `ec2:DescribeImages` API. Many organizations and Infrastructure-as-Code (IaC) tools, such as Terraform, dynamically search for the latest AMI matching a given name pattern. If the search query omits the "owners" attribute, AWS returns a list of AMIs from trusted and untrusted sources. Threat actors can exploit this design flaw by publishing a malicious AMI with a name that mimics an official one with a more recent timestamp,

causing automated processes to select it. When used to launch EC2 instances, these malicious AMIs grant threat actors control over the victim's cloud environment, potentially leading to unauthorized access, data theft, or further network compromise. According to Datadog, the flaw is not limited to Terraform; it also affects AWS command-line interface (CLI) commands and various programming languages, including Python, Go, and Bash scripts.

DataDog shared the following exploitation steps to execute arbitrary code within a victim's AWS environment and gain unauthorized access to cloud resources:

1. Craft a custom AMI with embedded malicious payloads, such as a backdoor or data exfiltration mechanism.
2. Name the AMI similarly to a trusted one, ensuring it fits commonly used naming patterns (for example, `ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*`).
3. Publicly publish the AMI or share it with targeted AWS accounts.
4. Ensure it appears as the most recent AMI rather than the legitimate ones.
5. Trigger the victim to dynamically retrieve the AMI using a misconfigured script or IaC tool, such as Terraform, that fetches the latest AMI without specifying the owner.
6. Cause the victim to launch an EC2 instance using the compromised AMI, allowing the threat actor's code to execute within the victim's cloud environment.
7. Establish persistence and control over the instance, enabling further exploitation such as credential theft, lateral movement, or data exfiltration.

On February 13, 2025, Datadog released whoAMI-scanner, an open-source tool designed to identify and flag instances using untrusted AMIs. Based on the repository, whoAMI-scanner requires an AWS profile to scan, a region, and a list of trusted AMI provider accounts. If no parameters are provided, it defaults to predefined settings. Once provided, whoAMI-scanner loads the AWS configuration, retrieves the caller's AWS account identity, and determines which AWS regions to scan. For each region, it queries running EC2 instances and extracts the AMI ID associated with each instance. whoAMI-scanner then checks these AMIs against a database of known AWS accounts, categorizing them as verified, self-hosted, allowed (if AWS's "Allowed AMIs" feature is enabled), or unverified. If whoAMI-scanner finds a public AMI from an unknown or untrusted source, it flags the AMI as potentially malicious. whoAMI-scanner generates a summary report displaying the status of all analyzed AMIs and warns users about unverified instances. If specified, whoAMI-scanner also writes to a CSV file. Finally, whoAMI-scanner informs operators about the status of AWS's "Allowed AMIs" feature and recommends ways to secure their environments.

Coordinated Exploitation of Grafana CVE-2021-43798 in Cloud Environments

On October 2, 2025, GreyNoise published a report on a coordinated one-day surge of exploitation attempts against Grafana instances vulnerable to CVE-2021-43798, a path traversal flaw that enables arbitrary file reads on exposed systems. The activity, observed on September 28, 2025, involved 110 malicious IP addresses focusing on internet-accessible Grafana deployments in the United States, Slovakia, and Taiwan, highlighting how long-patched but widely deployed software in cloud and hybrid environments remains a high-value reconnaissance and initial access vector.

The attack chain centers on HTTP requests that exploit the CVE-2021-43798 path traversal vulnerability to read arbitrary files from Grafana servers. These servers, when internet-exposed, commonly reside on cloud infrastructure or act as monitoring frontends into cloud workloads. Successful exploitation can expose sensitive configuration or credential material, enabling follow-on actions such as service discovery, lateral movement into underlying cloud resources, or pivoting to other managed services. GreyNoise notes that Grafana vulnerabilities, including CVE-2021-43798, frequently appear in reconnaissance phases of multi-step exploit chains and large-scale server-side request forgery (SSRF) and exploit waves spanning diverse software ecosystems, underscoring their role as reusable building blocks in cloud-focused toolchains rather than isolated bugs.

The observed surge exhibited tightly aligned targeting and tooling: traffic from multiple countries followed a consistent destination ratio (roughly 3:1:1 across the US, Slovakia, and Taiwan), with converging Transmission Control Protocol (TCP) and HTTP fingerprints indicating that different infrastructure sets, mainly originating from Bangladesh, with smaller clusters in China and Germany, were likely executing shared tasking or reusing a standard exploit kit and target list. In a cloud context, this pattern reflects how adversaries operationalize "resurgent" vulnerabilities. Attackers will integrate older vulnerabilities, such as CVE-2021-43798, and newer Grafana flaws into automated scanner and exploit frameworks to continuously scan large swaths of public cloud address spaces for unpatched instances. They will then selectively escalate from arbitrary file read to broader compromise when high-value telemetry or credentials are discovered.

Exploitation of Barracuda ESG and Impact on Belgium's VSSE

Between February 26 and 27, 2025, Reuters and Recorded Future News reported that Belgium's state security service (VSSE) was allegedly compromised by China-linked hackers via Barracuda Networks's Email Security Gateway (ESG) appliance, prompting a federal judicial investigation and the agency's decision to stop using Barracuda services. This activity is linked to earlier campaigns in which the China-nexus group UNC4841 exploited Barracuda ESG zero-day vulnerabilities to conduct long-term espionage against government and other high-value targets.

The primary exploitation path used malicious email attachments to trigger remote command injection in the ESG attachment-scanning component (CVE-2023-2868) on affected on-premise ESG versions. Crafted `.tar` archives abused Perl's `qx` operator to execute arbitrary system commands on the gateway, which sits in front of mail servers and provides a privileged vantage point into email flows. UNC4841 then deployed custom malware, including SALTWATER (a trojanized Simple Mail Transfer Protocol [SMTP] module enabling command execution and tunneling), SEASPY (a backdoor masquerading as BarracudaMailService triggered by "magic packets"), and SEASIDE (a Lua module that turns SMTP HELO/EHLO data into reverse shells), to turn ESGs into persistent access and exfiltration nodes. Barracuda later disclosed follow-on exploitation of CVE-2023-7102 in the `Spreadsheet::ParseExcel` library, again via malicious Excel attachments, to reinstall updated SEASPY and SALTWATER variants after initial remediation.

Belgian reporting indicates that between 2021 and 2023, this Barracuda-focused attack chain enabled exfiltration of roughly 10% of VSSE's email traffic via an external mail server used for communication with ministries, police, prosecutors, and foreign partners. Although the VSSE's classified internal systems were reportedly not compromised, the same external server handled HR-related correspondence, exposing personal data for nearly half of VSSE's staff and applicants and forcing identity document renewals for affected personnel. The case illustrates how the compromise of a single email security gateway product, through the iterative exploitation of Barracuda ESG parsing vulnerabilities and the deployment of bespoke implants, directly translated into strategic intelligence collection and the exposure of sensitive personnel data for a European intelligence service.

Cloud Abuse

Key Takeaways

- Cloud environments are increasingly being exploited by threat actors as command-and-control (C2) infrastructure; these environments are used for exfiltration and hosting malicious assets.
- Threat actors demonstrated an evolution in compromised victim cloud abuse during 2025, abusing LLM and machine learning (ML) cloud services in attack chains.

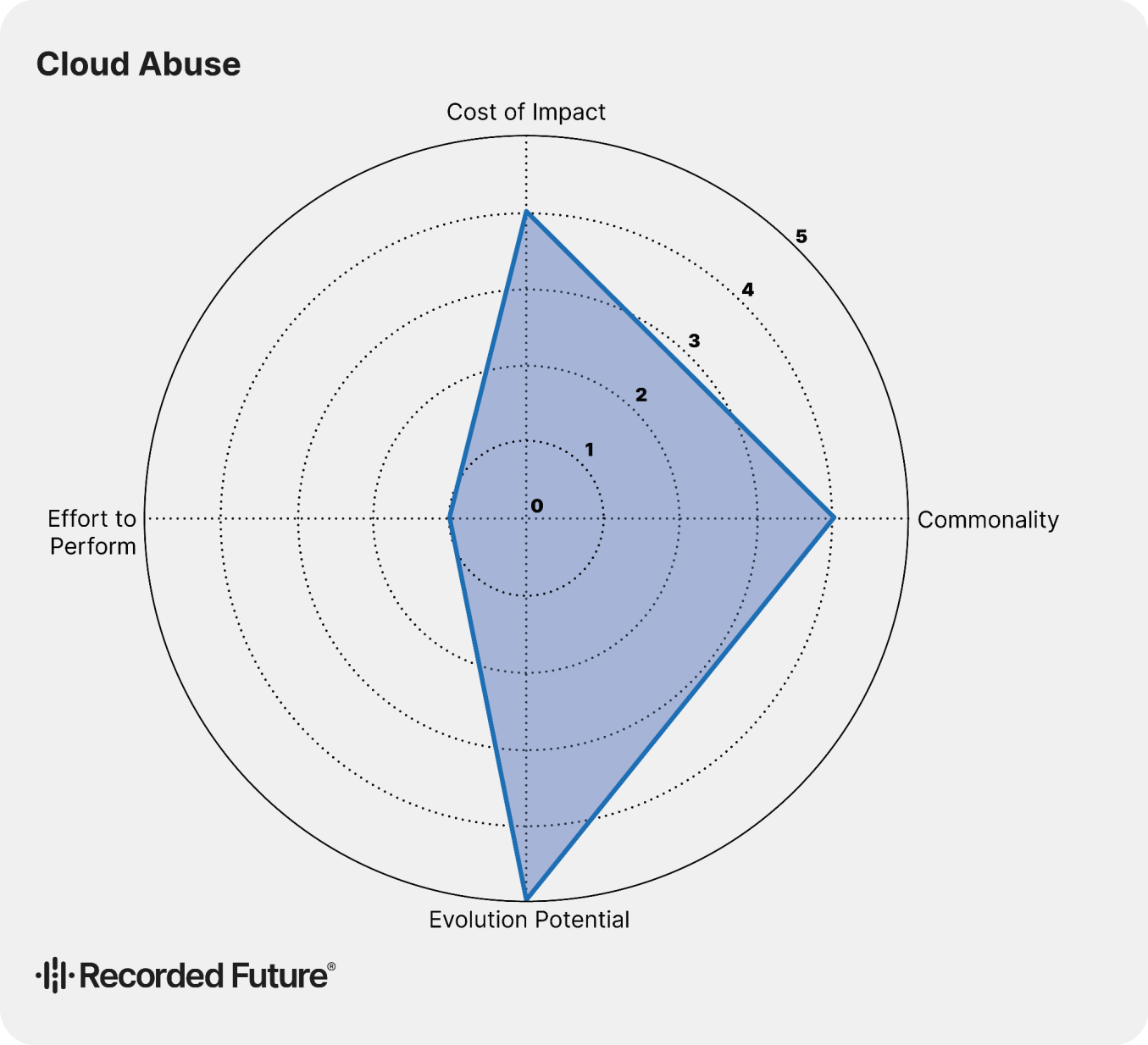

Figure 3: *Radar chart illustrating exploitation as a threat vector (Source: Recorded Future)*

Cost of Impact: 4 (High)

Cloud abuse, particularly the abuse of compromised cloud assets, can incur high monetary, operational, and reputational costs for cloud tenants. Evidenced by reports throughout the past year, which are presented later in this section, threat actors often exploit compromised cloud infrastructure to maintain legitimacy within a victim's environment and perform costly operations, for which the tenant is held responsible. Additionally, in instances where threat actors register managed cloud services and abuse them in malicious activities, the providers may assume adverse impacts on their reputation.

Commonality: 4 (High)

Cloud abuse is a broad threat that originates from both compromised legitimate infrastructure and infrastructure controlled by threat actors. Based on the evidence presented in this report, threat actors are identifying and developing new methods to exploit legitimate cloud services for activities such as malware hosting, C2 hosting, and data exfiltration.

Evolution Potential: 5 (Severe)

Cloud abuse events reported within the past year demonstrate that threat actors are increasingly abusing cloud services as part of their C2 infrastructure and to host and deliver malicious payloads. The methods threat actors employ to perform these actions have also evolved, demonstrating that threat actors are actively identifying new ways to abuse cloud services.

Effort to Perform: 1 (Minimal)

Threat actors can easily register legitimate cloud services, which can then be abused for malicious purposes. Many cloud service providers require only an email address and a payment method to register for their services. Additionally, compromised victim cloud infrastructure can be easily abused by a threat actor as a result of broader cloud environment compromise.

Threat Summary

Cloud abuse refers to the use of legitimate cloud services and infrastructure by a threat actor to perform malicious actions. This threat can generally be categorized into two main groups of activity: the abuse of threat actor-controlled cloud assets and the abuse of compromised victim cloud assets.

Within the past year, Insikt Group has noted that the abuse of compromised victim cloud assets is more prominently reported, with reporting, discussed further in the **Examples in the Wild** section, demonstrating that threat actors will commonly enroll in cloud services or develop cloud infrastructure with malicious intentions (mainly hosting malware, phishing websites or websites embedded with malicious artifacts, or C2 infrastructure, or deriving capabilities from cloud services, such as LLM services).

Insikt Group notes that, in contrast to previous iterations of this report, there has been little publicly accessible evidence within the past year, indicating either an uptick or persistence of common

compromised victim cloud abuse behaviors, such as cryptojacking and business email compromise (BEC), among others.

Outlook

Threat actors will likely continue to register and abuse cloud services and infrastructure to perform malicious actions.

As discussed earlier in this section, threat actors continue to identify new products and methods for abuse, demonstrating both their creativity in achieving malicious goals and the various cloud products they can procure and exploit to accomplish these goals. Threat actors recognize the anonymity these products provide, compared to similar traditional solutions, which make malicious traffic appear benign to most security products.

Threat actors will likely continue to abuse compromised cloud services during an attack chain rather than compute services that enable customized code execution, as abusing compromised victim cloud services can be financially lucrative or a means for threat actors to propagate during an attack. Additionally, over the past year, threat actors have demonstrated an interest in targeting cloud-based LLMs for abuse. This behavior is indicative of an evolution for both threat actor TTPs and victimology and signals that other service types may also be targeted in the future. It further indicates that threat actors prefer to abuse cloud resources rather than execute custom code in cloud environments to achieve malicious goals.

Mitigations and Detections

Because there are two distinct domains associated with cloud abuse, they are discussed separately below. **Figure 4** demonstrates a hypothetical attack chain associated with threat actor-registered cloud abuse, while **Figure 5** demonstrates a hypothetical attack chain associated with threat actor abuse of compromised victim cloud services. Throughout these visuals, Insikt Group has identified parts of the attack chain where defenders can most efficiently hunt for and mitigate behaviors associated with both domains of cloud abuse.

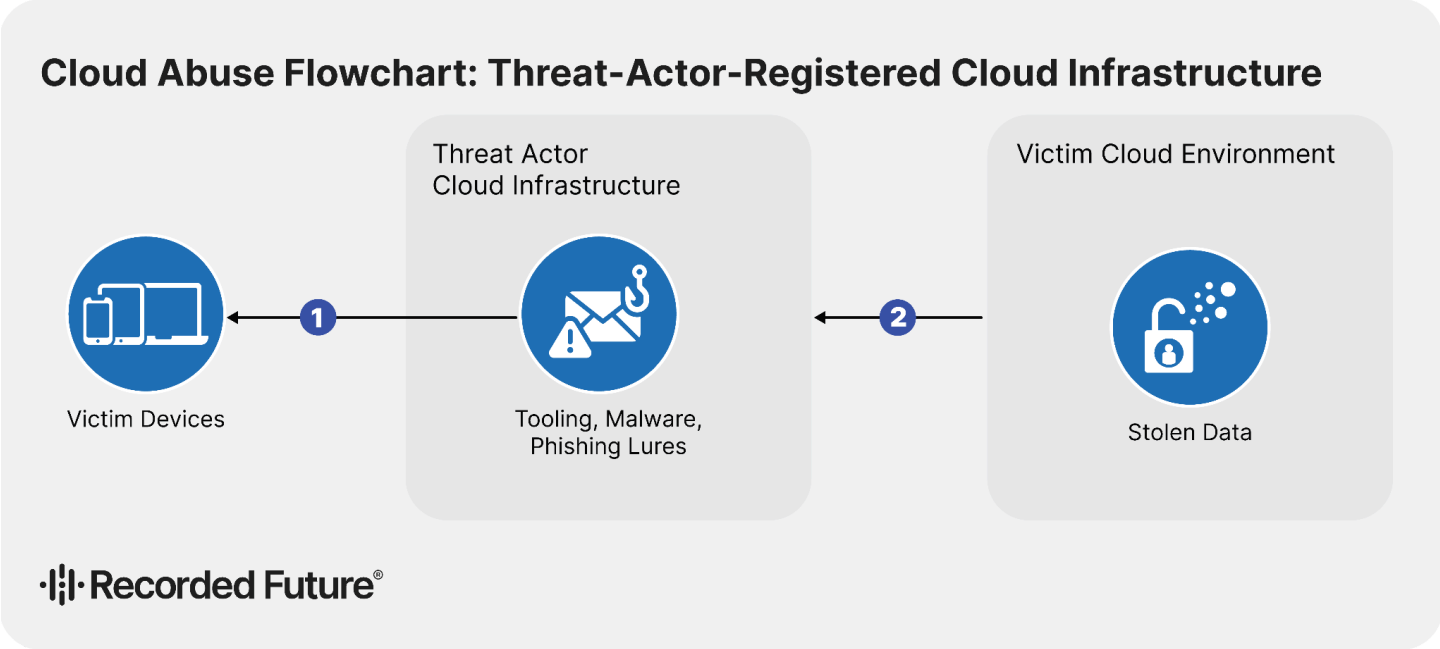

Figure 4: *Visual representation of potential threat actor-registered cloud abuse methods (Source: Recorded Future)*

① Cloud Infrastructure Abused for Malware Hosting

Threat actors will host malicious assets and tools within their own cloud infrastructure, allowing them to be loaded into victim environments during an attack. While these assets may not always be directly identifiable at the perimeter of a threat actor's infrastructure, some threat actors may expose web servers or file systems that contain the malicious assets, allowing them to be programmatically accessed.

Defenders can mitigate these threats in the following ways:

- Implement stateful packet analysis and identify requests that are typical of executable file downloads. This could include HTTP POST requests with common executable suffixes, or unexpected connections implementing standard file transfer protocols or remote-access protocols.
- Perform routine malware scans of local filesystems, particularly targeting directories where threat actors are known to stage malware, such as the temp and root directories.
- Employ IDS and IPS technologies and keep their detection capabilities up to date via indicator streaming and malware intelligence products.

② Exfiltration to Attacker-Controlled Cloud Locations

Over the past year, threat actors have consistently demonstrated the use of cloud services and infrastructure as targets for exfiltration during attacks. Similar to the threat discussed above, threat actors use these services for exfiltration because the traffic generated during exfiltration appears benign and can, therefore, blend into common network traffic patterns.

Although this exfiltration technique poses detection and mitigation issues due to the benign nature of this traffic, there are methods for detecting and mitigating this threat, which are included below:

- Monitor network activity for high-volume outbound POST requests to a source that is either unknown or untrusted.
- Compare telemetry destined for cloud infrastructure against net flow data for your environment and determine whether requests to specific cloud services are common; requests to uncommon services may be indicative of exfiltration.

In instances where threat actors are exfiltrating data from a cloud environment, they have been known to exploit cloud service functions to establish exfiltration channels rather than use traditional methods of exfiltration. To detect and mitigate this threat, the following actions may be taken:

- Identify the establishment of new, unexpected workflows and systems in cloud environments capable of backing up data to a remote cloud storage system. This activity may also be accompanied by the establishment of new service accounts and identity and access management (IAM) roles.
- Identify cloud accounts that attempt to access data without the requisite permissions.
- Identify the creation or modification of virtual private cloud (VPC) systems and cloud firewall technologies.

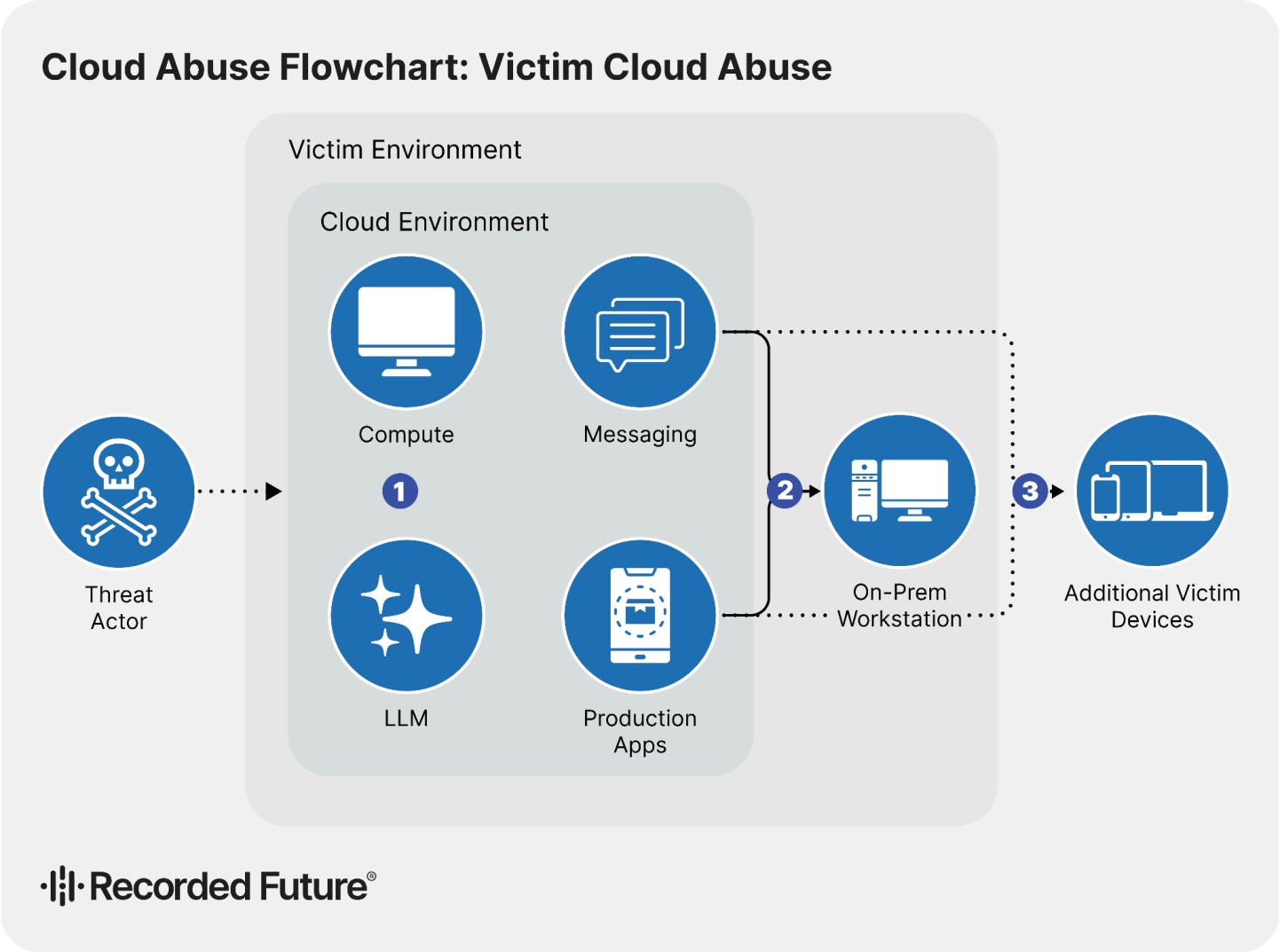

Figure 5: *Visual representation of potential compromised victim cloud abuse methods (Source: Recorded Future)*

① Cloud Service Abuse

Threat actors may abuse cloud services to perform resource-intensive operations without incurring the associated costs and to make malicious operations more difficult to attribute to themselves. Ways of detecting and mitigating these threats are discussed below:

- Set metrics-based alerts that trigger when a specific threshold is reached for certain resources. Alerts can be set for metrics such as memory use in compute instances and when a certain percentage of queries have been made from a total monthly allowance. Alerts should be designed after a baseline of expected activity is set, so that they can more accurately assist in identifying suspicious activity.
- Ensure that roles and permissions are applied to user accounts, adhering to the principle of least privilege, to ensure that, even in instances where cloud environments are breached, access to

restricted services is minimized, or that the permissions a threat actor has when accessing services are limited.
- Implement IAM logging and behavioral analysis services to identify unusual account behavior. Specifically, identify instances in which a cloud account is observed creating new cloud assets, using cloud services that are not frequently used, or modifying existing cloud assets.
- Monitor for unusual requests or interactions between cloud services, especially where no existing inter-service workflow exists.

② Abuse for On-Premises Pivot

Threat actors will exploit compromised cloud services to pivot into on-premises workstations and infrastructure, often by masquerading as legitimate cloud users or services, or by leveraging existing connections between the cloud environment and these assets. Methods to detect and mitigate this threat include the following:

- Monitor dedicated networking components, such as AWS Direct Connect and Azure ExpressRoute, for modifications and errors, especially those related to outbound traffic. Irregularities logged by these components may indicate malicious activity.
- Ensure that cloud networking components, such as VPCs and cloud firewalls, are properly configured.
- Ensure that authentication mechanisms, both to and from cloud environments, are properly configured.
- Monitor for the abuse of mailing systems and implement training solutions associated with business email compromise (BEC) attacks. Warn personnel of the dangers of trusting internal communications without first verifying that the communication is ordinary and expected and that any attachments do not include malicious content.

③ Abuse for External Propagation

Similar to the above, threat actors will also attempt to abuse cloud services to infect additional targets while masquerading as a member of the organization that owns a victimized cloud environment. Ways that this behavior can be detected and mitigated are discussed below:

- Monitor instances of cloud-based mailing services where accounts are creating outbound mail at an increased volume that is destined for unknown recipients.
- As mentioned above, identify the creation of unexpected cloud assets that can be made publicly available, such as files that are accessible to the public or webpages hosted on external-facing cloud infrastructure.

Examples in the Wild

Insikt Group curated a list of events published within the past year that demonstrate the threats posed by cloud abuse. These events are discussed below.

APT28's LameHug Infostealer Observed Abusing Qwen LLM During Post-Compromise Actions

On July 10, 2025, CERT-UA reported that the Russian state-sponsored threat group APT28 (also known as UAC-0001) had conducted spearphishing attacks against Ukrainian government agencies and entities in the defense and security sectors. The campaign involved the deployment of a new Python-based infostealer dubbed LameHug, representing the first publicly documented case of malware integrating an LLM to dynamically generate commands during runtime.

The campaign began with emails impersonating a government ministry and delivering ZIP archives named `Appendix.pdf.zip` or `Dodatok.pdf.zip`. These contained `.pif` executables disguised as PDFs, packed with PyInstaller to execute LameHug in memory. The malware used a hijacked email account *(boroda70@meta[.]ua*) to deliver payloads and leveraged compromised infrastructure, including the domain *stayathomeclasses[.]com* and IP address *144[.]126[.]202[.]227*, for C2 and exfiltration activities.

LameHug accessed Alibaba Cloud's Qwen 2.5-Coder-32B-Instruct LLM via Hugging Face API to generate system commands based on Base64-encoded prompts. This allowed it to evade static detection signatures and tailor its execution to victim environments in real time. Commands generated by the LLM performed reconnaissance using native Windows utilities, including `systeminfo`, `wmic`, `tasklist`, `ipconfig`, and `dsquery`. The collected data was stored locally at `%PROGRAMDATA%\info\info.txt`, and document files from user directories were staged for exfiltration.

The two LameHug variants used different exfiltration methods: one sent data via HTTP POST to a compromised website, while the other leveraged Secure File Transfer Protocol (SFTP) with hard-coded credentials. Both methods exfiltrated sensitive files like Office documents and system reconnaissance outputs. Notably, the malware did not establish persistence, aligning with "smash-and-grab" tactics focused on short-lived espionage operations.

Multi-Stage Cyberattack "Operation SalmonSlalom" Targets Industrial Organizations in APAC with FatalRAT Delivered via Youdao Cloud Notes and MyQcloud CDN

On February 24, 2025, cybersecurity firm Kaspersky published a report detailing Operation SalmonSlalom, a multi-stage campaign that delivers the FatalRAT backdoor to industrial organizations in the APAC region. The operation primarily targets Chinese-speaking users in the manufacturing, telecommunications, and energy sectors across Taiwan, Malaysia, China, Japan, the Philippines, and other countries in the Asia-Pacific region. The threat actors heavily rely on legitimate cloud services, specifically Youdao Cloud Notes and Tencent's MyQcloud content delivery network (CDN)/object storage, to host configuration data and payloads, enabling them to rotate their infrastructure and rapidly blend into regular traffic.

The attack chain begins with phishing emails and WeChat or Telegram messages that deliver ZIP archives masquerading as tax documents to employees at targeted industrial organizations. These

archives contain a packed first-stage loader (using packers such as UPX, AsProtect, or NSPack) that contacts Youdao Cloud Notes to obtain an updated list of URLs hosting the second-stage components, `Before.dll` (configurator) and `Fangao.dll` (loader). `Before.dll` fetches additional configuration from Youdao-hosted notes, including C2 and payload URLs, writes this data to a local configuration file, and sends basic system profiling data to the C2. `Fangao.dll` then reads this configuration, performs environment checks (it looks for system language and file paths, and ensures the time zone is set to "UTC+8") to ensure it is running on a victim system. If successful, it downloads and decrypts FatalRAT.

From there, the campaign uses a mix of GUI-driven Group Policy abuse and DLL sideloading to establish persistence and deliver FatalRAT from cloud infrastructure. `Fangao.dll` automates the Windows Group Policy Editor to add a logon script that launches a trojanized media player binary, thereby avoiding direct registry modification and reducing detection. It then leverages a legitimate driver utility (DriverAssistant) for DLL sideloading, placing a malicious loader DLL that contacts MyQcloud-hosted content to download FatalRAT. Once active, FatalRAT conducts reconnaissance, anti-VM checks, and establishes its own persistence, then supports keylogging, file exfiltration, command execution, Server Message Block (SMB) brute forcing for lateral movement, optional destructive actions (such as MBR overwrites), and deployment of remote access tools like UltraViewer and AnyDesk to extend interactive control over compromised industrial networks.

APT41 Cyber-Espionage Campaign Employs Multi-Stage Malware Framework TOUGHPROGRESS that Uses Google Calendar for C2 Communication

On May 29, 2025, Insikt Group published a Validated Intelligence Event (VIE) summarizing Google Threat Intelligence Group's (GTIG) analysis of a cyber-espionage campaign conducted by APT41, a Chinese state-sponsored threat actor. According to GTIG, the campaign, first observed in October 2024, uses a multi-stage malware framework called TOUGHPROGRESS that exploits Google Calendar as a covert channel for C2. The campaign targets global government organizations through a phishing scheme that uses a compromised government website.

Based on Mandiant's analysis, APT41 sends spearphishing emails containing a link to a ZIP archive named 出境海關申報清單.`zip`, hosted on a compromised government website. This archive includes a shortcut file (LNK) named 申報物品清單.`pdf.lnk`. It is disguised as a PDF using the double-file extension technique, along with a directory hosting seven image files. Among those seemingly benign image files, `6.jpg` contains an encrypted payload, and `7.jpg` is a DLL that decrypts and launches the payload. When a victim opens the LNK file, it executes the DLL, deletes itself to avoid detection, and opens a decoy PDF resembling a customs declaration document to divert suspicion.

The decrypted payload initiates the following three-stage execution process:

1. PLUSDROP: This stage decrypts and loads the next module into memory, enabling a stealthy, fileless progression.
2. PLUSINJECT: This stage hijacks a legitimate Windows process (`svchost.exe`) via process hollowing to execute the final payload, TOUGHPROGRESS.

3. TOUGHPROGRESS: This stage performs the core functionality on the compromised host. It employs advanced evasion techniques including register-based indirect calls, dynamic control-flow obfuscation, and 64-bit arithmetic overflows. It embeds shellcode within the `.pdata` section of its code, decrypts it with a hard-coded sixteen-byte XOR key, and decompresses it using the Lempel-Ziv NT version 1 (LZNT1) algorithm.

Once activated on a compromised system, TOUGHPROGRESS establishes threat actor-controlled C2 communication using Google Calendar. It creates zero-duration calendar events to exfiltrate data and polls the calendar for encrypted threat actor commands scheduled on July 30 and 31, 2023. TOUGHPROGRESS decrypts these commands using a two-layer XOR scheme that combines a static ten-byte key with a dynamic four-byte key generated for each message. After executing the commands, TOUGHPROGRESS encrypts the results and uploads them to new calendar events, establishing full-duplex communication over a legitimate platform.

According to the report, Mandiant and GTIG disrupted APT41's infrastructure by revoking access to the malicious Workspace projects and blocking associated URLs, such as `word[.]msapp[.]workers[.]dev`, `resource[.]infinityfreeapp[.]com`, and `hxxps://my5353[.]com/nWyTf`, through Google Safe Browsing.

ACR Stealer Employs Dead Drop Resolver to Obscure its C2 Infrastructures

On March 1, 2025, Trend Micro Research reported a surge in activity related to ACR Stealer, accompanied by a new TTP update. ACR Stealer is a malware-as-a-service (MaaS) information stealer operated by a threat actor named "SheldIO". According to Trend Micro Research, ACR Stealer uses advanced dead-drop resolvers to conceal its C2 infrastructure by leveraging legitimate platforms, such as Steam and Google Docs. Additionally, Trend Micro Research reported active infection in multiple countries, including the United States, Canada, Germany, France, the Czech Republic (Czechia), Brazil, Peru, Sweden, Finland, and Indonesia over the past 30 days.

Based on Trend Micro Research's analysis, threat actors distribute ACR Stealer through cracked software downloads and use Cloudflare Web Application Firewall (WAF) to protect their distribution domains from detection. Once executed, ACR Stealer performs the following actions on a victim's machine:

- Collects sensitive data from browsers, email clients, and cryptocurrency wallets
- Steals login credentials, financial information, system information, and personally identifiable information (PII)
- Captures clipboard data
- Harvests session tokens
- Encodes its C2 domains using Base64 to evade detection
- Exfiltrates stolen data over C2 channels using trusted web services to bypass security controls

Abuse of AWS LLM Services in Attack Chains Targeting Machine Learning Supply Chains

On February 27, 2025, Trend Micro published a report describing a multi-stage machine learning supply-chain attack that abuses cloud-based AI services, specifically Amazon SageMaker and Amazon Bedrock, to compromise LLM-driven applications. The source highlights that misconfigurations, over-privileged roles, and unvetted third-party ML components enable an attacker to move from a single exposed SageMaker notebook in a development account to complete control over Bedrock-based LLM workflows, guardrails, and retrieval-augmented generation (RAG)-backed knowledge bases in production.

The attack begins in the developer AWS account, where a SageMaker notebook instance with direct internet access and a default execution role is used by developers to install third-party extensions. The attacker publishes a malicious Python package that a developer installs and runs within that notebook, thereby opening a reverse shell to the attacker and granting them command execution on the ML workstation. From there, the attacker enumerates SageMaker and IAM resources via AWS CLI, discovers that the notebook's role has powerful SageMaker and `iam:PassRole` permissions, and uses `sagemaker:CreateNotebookInstance` plus `sagemaker:CreatePresignedNotebookInstanceUrl` to create a new notebook bound to an administrative role. With this privilege escalation, the attacker gains effective control of the developer account and leverages CloudTrail logs to identify and assume a cross-account Bedrock role in the production account via `sts:AssumeRole`, bridging directly into the LLM-serving environment.

Once in the production account, the attacker abuses Bedrock Guardrails and downstream automation to weaponize LLM interactions. First, they update a guardrail attached to a Bedrock conversational agent so that prompts containing words like "How" or "What" are blocked and replaced with a malicious block message that instructs users to download a PDF from an attacker-controlled URL, using the LLM chat interface to deliver malware through otherwise "safety" infrastructure.

In a closely related variant, the attacker targets a code-generation model whose outputs are executed by an AWS Lambda function; they poison the guardrail's block message with Python code that, when passed downstream and executed, exfiltrates the Lambda role's credentials to an attacker-controlled IP, granting the adversary complete control over the LLM application's business logic layer. With those cloud credentials, the attacker enumerates Bedrock agents, knowledge bases, and data sources, repoints the RAG knowledge base to an attacker-controlled Simple Storage Service (S3) bucket to poison LLM responses with tampered content, and finally uses S3 access (for example, via `aws s3 sync`) to exfiltrate sensitive inference data, including user prompts and proprietary information driving the LLM's behavior.

Cloud Ransomware

Key Takeaways

- Threat actors are actively identifying novel methods to perform cloud ransomware attacks.
- Threat actors must rely on native services to encrypt or destroy data during a cloud ransomware attack.

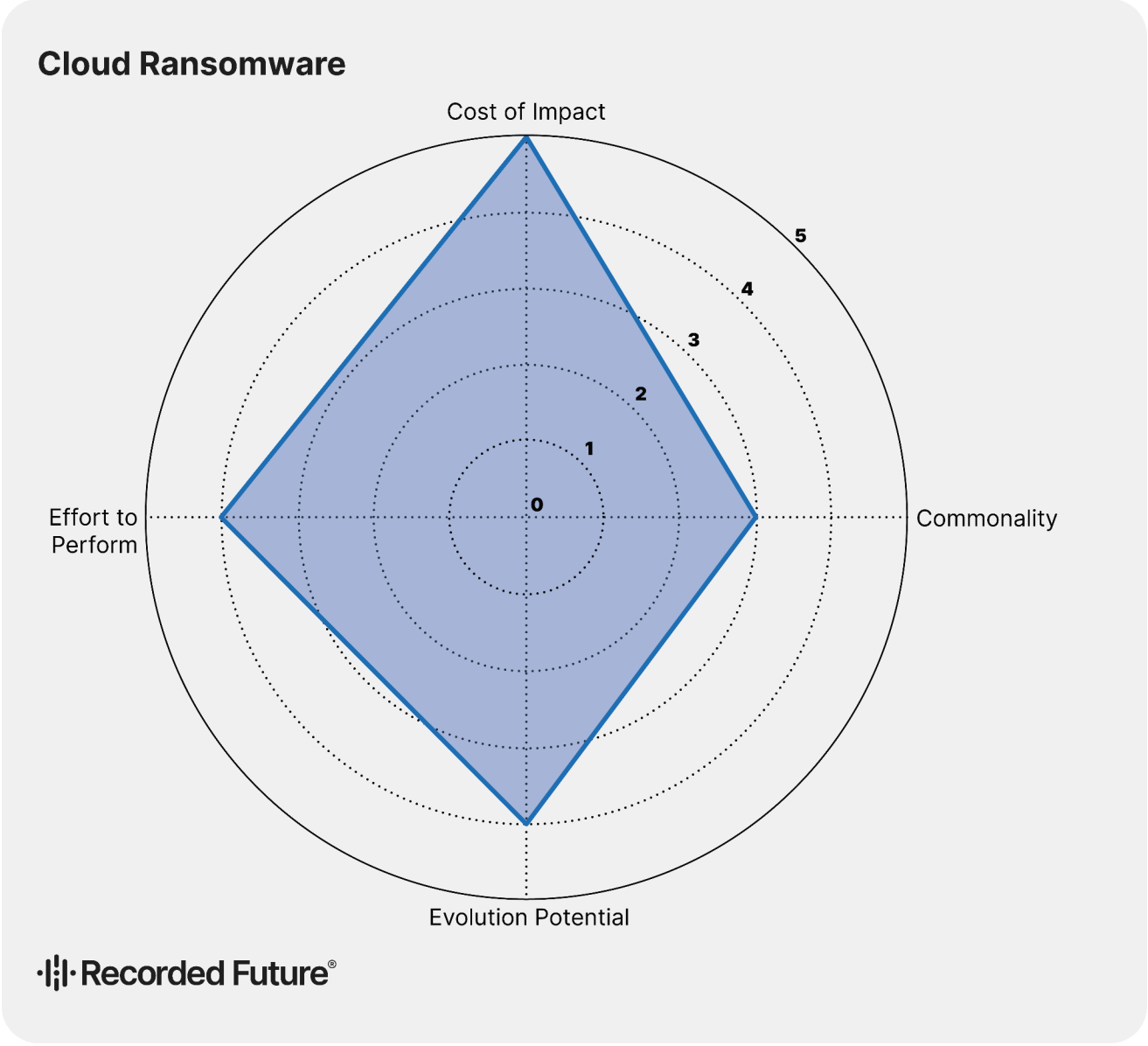

Figure 6: *Radar chart illustrating cloud ransomware as a threat vector (Source: Recorded Future)*

Cost of Impact: 5 (Severe)

Victims of cloud ransomware attacks incur high costs monetarily, operationally, and reputationally. Even when victims refuse to negotiate, critical data and systems are effectively rendered useless depending on the method used to deny access to these assets.

Commonality: 3 (Moderate)

Based on a comparison of cloud ransomware events observed throughout the past year to previous reporting, cloud ransomware events continue to be monitored periodically, but they are not represented in high volumes.

Evolution Potential: 4 (High)

Threat actors have demonstrated in the past year that they are continuously identifying novel methods to perform cloud ransomware attacks. While the majority of these attacks still rely on threat actors' use of native cloud services to perform the attacks, the methods they employ (and the theoretical methods threat researchers have reported) indicate that additional attack methods will likely surface in the future.

Effort to Perform: 4 (High)

Similar to Insikt Group's determinations in previous reporting, threat actors' reliance on cloud services to perform cloud ransomware operations ensures that an in-depth knowledge of cloud service provider (CSP) environments will remain necessary to perform cloud ransomware attacks.

Threat Summary

Cloud ransomware operations exploit trusted access to cloud environments to encrypt cloud assets or render them inaccessible; this includes object storage, virtual disks, databases, and backups. Instead of deploying large numbers of binaries, threat actors use compromised accounts, roles, tokens, and keys to change encryption settings, rotate or destroy key material, or mass-modify stored data through cloud APIs and consoles. This keeps activity primarily within typical administrative paths and forces defenders to rely on behavioral indicators, such as unusual spikes in encryption, snapshot changes, or bulk storage operations.

Outlook

Cloud ransomware will continue to rely on native cloud services to rapidly encrypt large volumes of cloud data or render them inaccessible.

Due to the managed nature of CSP environments, which were the only targets identified in cloud ransomware reporting during the past year, threat actors must rely on native services to encrypt or destroy data during a cloud ransomware campaign. Since these native services provide threat actors

with the functionality needed to perform destructive actions against targeted assets, threat actors will also continue to need privileged access within victim cloud environments to perform cloud ransomware attacks effectively.

Mitigations and Detections

Figure 7 illustrates a hypothetical cloud ransomware attack chain and identifies parts of the attack chain where defenders can most efficiently hunt for and mitigate behaviors associated with it.

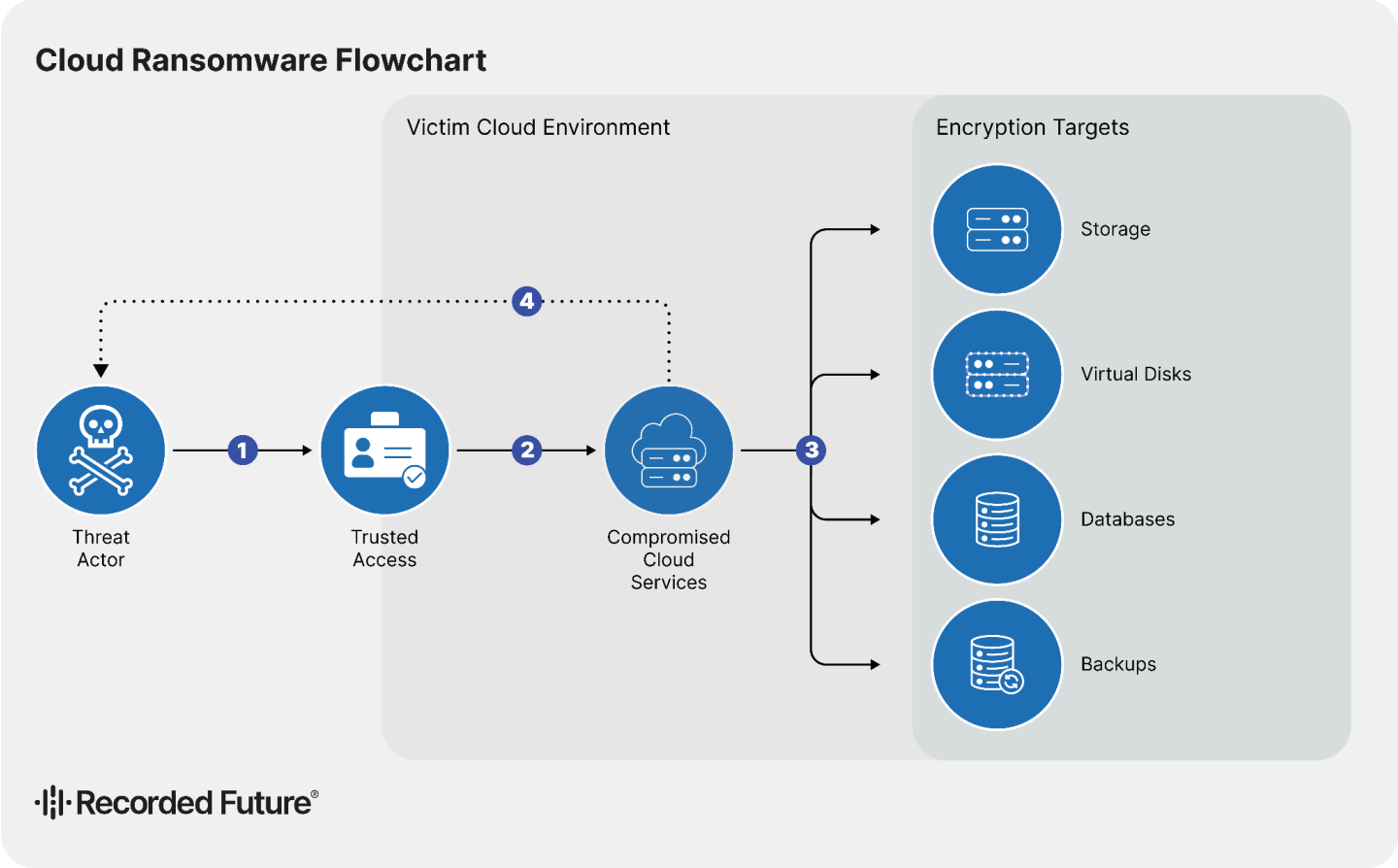

Figure 7: *Visual representation of potential cloud ransomware attack vectors (Source: Recorded Future)*

① Threat Actor Access to Victim Cloud Environment

In all events observed throughout the past year, threat actors had to first gain trusted access to the victim environment. This access, which grants the threat actor heightened privileges within the environment, is necessary to perform cloud ransomware attacks, since cloud ransomware attack chains rely on cloud service abuse for encryption, exfiltration, and persistence actions later.

Mitigation and hunting strategies for this behavior are discussed in the section titled **Credential Abuse, Account Takeover, and Unauthorized Access**.

② Discovery to Support Cloud Ransomware Activities

After initial access is established, threat actors will profile the cloud environment, performing discovery to identify assets for encryption and services that can be abused to support functions necessary for the cloud ransomware attack to succeed.

Common mitigation and threat hunting capabilities for this behavior include the following:

- Ensure that only tightly scoped roles have "list," "describe," and metadata-read permissions.
- Segment business-critical data and services into separate accounts, projects, or subscriptions so that compromise of a single identity does not enable broad discovery across the environment.
- Monitor cloud audit logs for broad or unusual enumeration patterns (such as bulk listing of storage, compute, identities, or security settings) and alert when they deviate from normal administrative baselines.
- Limit who can view or query security coverage (EDR, logging, backup, and monitoring deployments) and alert on attempts to inventory or probe those protections.
- Treat synchronization and integration components (identity sync, directory connectors, CI/CD systems, gateways) as high-value assets, restricting interactive access and monitoring them for discovery tooling or large-scale queries.
- Constrain service and automation accounts to narrowly defined scopes and non-interactive use, and detect mass directory or graph-style enumeration from any identity.
- Apply strong governance and approvals on any privilege escalation or "elevate access" mechanisms, and closely monitor for broad discovery activity that follows a privilege increase.
- Regularly review and reduce broad read-only or viewer roles that span many resources or tenants, replacing them with granular, task-specific permissions.

③ Cloud Service Abuse for Encryption or Denial of Access

Based on the events discussed later in this section, threat actors rely on native cloud services to conduct the action encryption of cloud assets or to deny users access to these assets.

Due to the fact that the methods Insikt Group observed only affected AWS and Azure environments, specific mitigation and hunting suggestions for each platform are shown below:

- Require S3 bucket versioning, Object Lock (preferably in compliance mode), and MFA Delete on business-critical buckets so that malicious encryption or deletion does not permanently deny access.
- Use AWS Organizations service control policies (SCPs) and S3 bucket policies to block use of SSE-C (server-side encryption with customer-provided keys) on critical data (deny requests with `x-amz-server-side-encryption-customer-algorithm`) and require SSE-S3 or SSE-KMS (Key Management Service) with organization-controlled keys.

- Enforce that S3 encryption uses AWS KMS customer-managed keys (CMKs) in tightly controlled accounts with key policies that application roles cannot modify, preventing attackers from swapping or monopolizing encryption keys.
- Configure immutable, cross-account S3 replication to backup buckets in a separate AWS account where production roles lack overwrite or delete permissions, preserving recovery options after in-place encryption or mass delete.
- Restrict `DeleteObject`, `DeleteBucket`, `PutBucketVersioning`, `PutObjectRetention`, and related destructive S3 permissions to a minimal set of break-glass roles, with MFA required for use.
- Monitor and alert on sudden changes to bucket encryption settings (especially bulk enabling of SSE-C or mass object re-encryption) as indicators of in-progress cloud-native ransomware.
- Detect and alert on high-volume S3 delete operations or complete bucket wipes within short time windows, treating these as access-denial events requiring immediate response.
- Ensure S3 bucket versioning and long-retention delete marker policies are enabled so that object deletions can be rolled back and do not equate to permanent data loss.
- Maintain offline or out-of-band backups (for example, periodic exports to storage outside AWS control) so that even if S3 data and its versions are encrypted or destroyed, recovery remains possible.

Azure:

- Restrict the use of `Microsoft.Authorization/elevateAccess/action` and require Privileged Identity Management with approvals and limited durations, so attackers cannot easily obtain subscription-wide Owner rights for destructive actions.
- Separate Azure Key Vault administration from storage and backup administration, ensuring the same identities cannot both control encryption keys and delete or reconfigure the storage protected by those keys.
- Enforce Key Vault soft delete and purge protection for all keys used by storage encryption scopes so that deleting a key cannot permanently prevent decryption.
- Apply Azure Policy to constrain encryption scopes so that storage accounts can only use keys from approved Key Vaults, and block ad hoc changes to key Uniform Resource Identifiers (URIs) or encryption scopes on production storage.
- Place critical backups, Recovery Services Vaults, and long-term retention stores in dedicated subscriptions or management groups with independent resource locks and immutability controls that production Owners cannot remove.
- Enable and monitor soft delete and retention for storage accounts, blobs, snapshots, and backup items so that deletion does not immediately equate to access denial.
- Alert on large-scale deletion of VM snapshots, Restore Point Collections, storage accounts, and backup containers, mainly when such activity follows privilege escalation or changes to encryption scopes or keys.
- Implement cross-region or cross-tenant backup strategies where restore rights are held by a separate, highly restricted set of identities, preventing the same compromised accounts from both encrypting production and destroying backups.

- Use Conditional Access, strong MFA, and device restrictions for administrative access to the Azure portal and management APIs, limiting attackers' ability to execute tenant-wide encryption and deletion operations once credentials are stolen.

④ Establishment of Exfiltration and Persistence Channels

Before encryption or deletion actions, threat actors will often attempt to establish channels between the victim environment and their own infrastructure that will be used to exfiltrate the data they are making inaccessible to support double extortion. Additionally, either before or after the encryption or deletion of data, threat actors will establish a persistence channel, which will be used to restore access to the compromised data and, in some instances, maintain communications with the victim for negotiation purposes.

Common mitigation and threat hunting capabilities for this behavior include the following:

- Enforce least-privilege read access to data stores so only narrowly scoped identities can download or list large volumes of sensitive data.
- Implement strict egress controls for workloads handling sensitive information so outbound traffic to the internet or untrusted networks is tightly filtered and observable.
- Prefer private or service-specific network endpoints for data access and disable direct public access so exfiltration must traverse monitored, controlled paths.
- Monitor for abnormal bulk download or data-transfer activity from cloud storage and databases, especially when initiated by identities or locations that do not typically perform such operations.
- Use policies to restrict the creation of new cross-account or cross-tenant data sharing, replication, or peering links, and require review and approval for any new trust or data-movement relationship.
- Rely on short-lived, automatically rotated credentials (tokens, role assumptions) instead of long-lived keys, and rapidly revoke or rotate any credentials suspected of compromise to disrupt ongoing exfiltration channels.
- Restrict who can create or modify federated identity and SSO configurations, and require approvals and logging for any new trust relationships to prevent attackers from creating identity-based persistence.
- Regularly audit identity providers for unexpected or unused federated domains, external trusts, or custom token rules, and remove any configuration not explicitly required for business purposes.
- Separate duties so no single administrator identity can both manage federation and trust configurations and administer high-value data resources, reducing the chance of durable, identity-agnostic persistence for ransomware operations.

Examples in the Wild

Insikt Group curated a list of events published during 2025 that demonstrate the threats posed by cloud ransomware attacks. These events are discussed below.

Breaking Down Observed S3 Ransomware Techniques in AWS

On November 18, 2025, Trend Micro's Trend Research team published a report examining how ransomware operators are shifting from traditional on-premises encryption malware to cloud-centric techniques that abuse native AWS capabilities, with a particular focus on real-world S3 ransomware behaviors. The report highlights that attackers increasingly rely on misconfigured permissions and stolen IAM credentials to gain write-level access to S3 buckets, enumerate business-critical storage, and then make data irrecoverable (without necessarily deploying conventional binaries).

One observed technique centers on the threat actor Codefinger, who used S3 server-side encryption with customer-provided keys (SSE-C) to ransom S3 data. After obtaining write-level access via leaked IAM role credentials or a compromised AWS account, the threat actor enumerates S3 buckets to find high-value targets lacking protective features such as versioning, object lock, and MFA Delete. They then initiate SSE-C by supplying a locally stored AES-256 key through the `x-amz-server-side-encryption-customer-algorithm` mechanism (via REST API/HTTP or AWS SDKs), causing AWS to encrypt all affected objects without ever retaining the key itself, only a hash-based method authentication code (HMAC) in CloudTrail that cannot be used for decryption. Once encryption is complete, a ransom note is placed in the bucket, leaving both the customer and AWS unable to restore access without the attacker's key.

The Bling Libra threat actor also leveraged stolen AWS credentials to access S3, exfiltrated bucket contents to attacker-controlled infrastructure, and then deleted the data or entire bucket, sometimes leaving ransom notes in the original bucket or in newly created, ransom-branded buckets. Trend Micro notes this exfiltration-and-deletion model is particularly likely in practice because it is operationally simple and makes recovery impossible if versioning or backups are absent. Consistent with these techniques, Trend Vision One telemetry has recorded CloudTrail-based behaviors associated with S3 ransomware campaigns, including S3 bucket enumeration, bulk download followed by deletion, ransom note uploads, and IAM policy or administrative changes that expand privileges over S3 resources, all of which contribute to effective cloud-native ransomware operations against AWS environments.

Storm-0501 Cloud-Based Ransomware in Hybrid Azure Environments

On August 27, 2025, Microsoft Threat Intelligence published a report detailing how the financially motivated threat actor Storm-0501 has shifted from traditional endpoint encryption to cloud-based ransomware operations targeting hybrid environments that integrate on-premises Active Directory, Microsoft Entra ID, and Azure resources. The report describes a campaign against a large multi-subsidiary enterprise where Storm-0501 used preexisting domain administrator access to move from fragmented on-premises coverage into multiple Entra ID tenants, gained Global Administrator rights, and then used only native cloud capabilities for data theft, destruction, and encryption before extorting the victim.

On-premises activity began from an already-compromised Active Directory domain where only part of the environment was protected by Microsoft Defender for Endpoint. Storm-0501 enumerated security coverage using commands such as `sc query sense` and `sc query windefend`, then used

Evil-WinRM from an unenrolled Entra Connect Sync server to laterally move and tunnel to additional systems. The threat actor performed DCSync against domain controllers to obtain credential material and then abused multiple Entra Connect Sync servers and their Directory Synchronization Accounts to enumerate identities and resources across more than one Entra ID tenant using AzureHound. They encountered blocked sign-in attempts when Conditional Access and MFA were enforced on privileged accounts.

Storm-0501 escalated privileges in the cloud by identifying a synced non-human account with the Global Administrator role that lacked MFA, resetting its on-premises password, and relying on Password-Hash Synchronization to propagate the new password to Entra ID, then registering attacker-controlled MFA to satisfy Conditional Access. Because Azure portal access required Microsoft Entra hybrid-joined devices, the actor moved laterally until successfully logging into the portal from a hybrid-joined server. With Global Administrator permissions, Storm-0501 created a malicious federated trust to an attacker-controlled tenant using AADInternals and a custom certificate, enabling the forging of Security Assertion Markup Language (SAML) tokens. This allowed the attacker to impersonate arbitrary users based on ImmutableId and inherit their roles, establishing durable, identity-agnostic persistence in the victim tenant.

From this Entra ID foothold, Storm-0501 elevated access in Azure by invoking `Microsoft.Authorization/elevateAccess/action` to gain User Access Administrator and then assigning themselves Owner across all subscriptions, enabling complete control of Azure Storage, virtual machines, and backup infrastructure. The actor discovered storage accounts, snapshots, Restore Point Collections, and Recovery Services Vault configurations; exposed internal storage accounts to the internet; retrieved storage access keys; and exfiltrated data at scale using AzCopy to attacker-controlled cloud storage. They then deleted VM snapshots, Restore Point Collections, storage accounts, and backup containers, removing resource locks and immutability policies as necessary. For residual storage accounts, they configured encryption scopes to use an attacker-controlled Azure Key Vault key before deleting that key, resulting in cloud-based encryption for impact. Following exfiltration and destructive actions, Storm-0501 used compromised identities to contact the victim via Microsoft Teams and deliver ransom demands.

Credential Abuse, Account Takeover, and Unauthorized Access

Key Takeaways

- Valid credentials remain a valuable resource for threat actors, both for gaining access to cloud environments and for performing additional actions post-compromise.
- Threat actors exploit a wide range of authentication methods, including username and password combinations, keys, and tokens, to gain unauthorized access.
- Threat actors continue to implement and develop capabilities for capturing and abusing MFA codes.

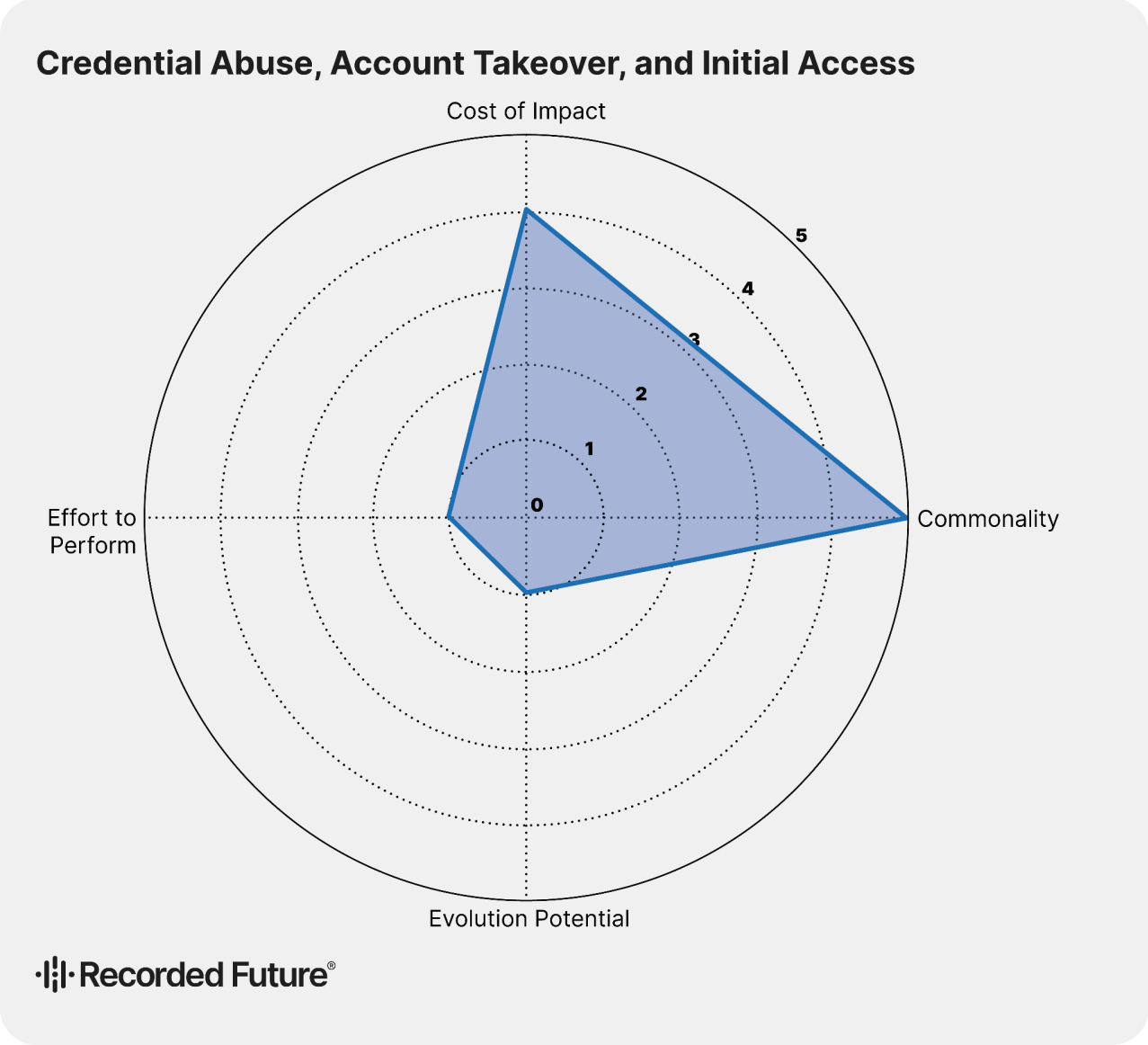

Figure 8: *Radar chart illustrating unauthorized access as a threat vector (Source: Recorded Future)*

Cost of Impact: 4 (High)

This threat involves threat actors gaining privileged access to cloud environments. While this privileged access does not inherently ensure that threat actors are going to be able to perform malicious actions within a cloud environment, the capabilities that this access affords a threat actor are potentially highly damaging.

Commonality: 5 (Severe)

Nearly all of the events discussed in this report include the abuse of valid credentials at some point in their attack chain. Due to the value of valid cloud authentication materials and their necessity to perform restricted access in cloud environments, these materials will likely be observed in most attacks against cloud environments.

Evolution Potential: 1 (Minimal)

The methods that allow unauthorized access are rigid. To perform these methods, threat actors must abuse authentication materials only in the way they are intended to be used; therefore, there is little potential for threat actors to innovate on these attack methods.

Effort to Perform: 1 (Minimal)

For the same reason that the complexity of this threat will likely never evolve, the effort to perform related attacks will similarly remain low. Threat actors have access to a myriad of solutions for gathering valid credentials, which can then be used directly for authentication to and within cloud environments.

Threat Summary

Threat actors must be able to gain access to an environment before they can pursue or achieve their overarching goals. Often, as is demonstrated by the events discussed in this section, threat actors will obtain and abuse legitimate credentials to gain this access.

One of the most common ways threat actors achieve unauthorized access is by abusing valid credentials associated with a trusted entity within the cloud environment. These valid credentials provide inherent permissions or grant access within the context of the victim's environment through account takeover. There are multiple ways that threat actors are capable of obtaining valid credentials: password spray and stuffing attacks, phishing and adversary-in-the-middle (AitM) or man-in-the-browser (MitB) attacks, and by purchasing valid credentials from an initial access broker (IAB), among other methods.

The term “valid credentials” is often used synonymously with username and password combinations; however, threat actors have demonstrated that other forms of authentication material can be used to gain access to or within cloud environments. Secret keys and tokens generated by and stored within cloud environments can also be used to authenticate with specific cloud services, allowing threat actors to collect data from the environment or perform otherwise unauthorized actions while masquerading as

a legitimate entity. Additionally, username and password combinations are often not enough for breaching a cloud environment, and threat actors must intercept MFA codes to gain access.

Valid credentials are also highly valued by threat actors, as they often facilitate account takeover. Once a threat actor has gained access to a valid account trusted by a victim's cloud environment, they can masquerade as that account, making their actions within the environment appear legitimate. Often, the only way for defenders to detect that an account has been compromised is through behavioral indicators, which may not immediately alert defenders to the active risk in their environment, whether at the perimeter of a cloud environment or internally.

Outlook

Threat actors will continue to implement stolen valid credentials materials to gain access to cloud environments for the foreseeable future. Additionally, threat actors will continue to execute campaigns for gathering valid credentials that can be used in future attacks against cloud environments. As mentioned above, valid accounts are highly valued by threat actors for their ability to grant access to cloud environments, as well as their capacity to mask malicious activities and represent them as benign events.

Mitigations and Detections

Figure 9 illustrates a hypothetical attack chain that includes unauthorized access. Throughout this visual, Insikt Group has identified parts of the attack chain where defenders can most efficiently hunt for and mitigate behaviors associated with cloud abuse.

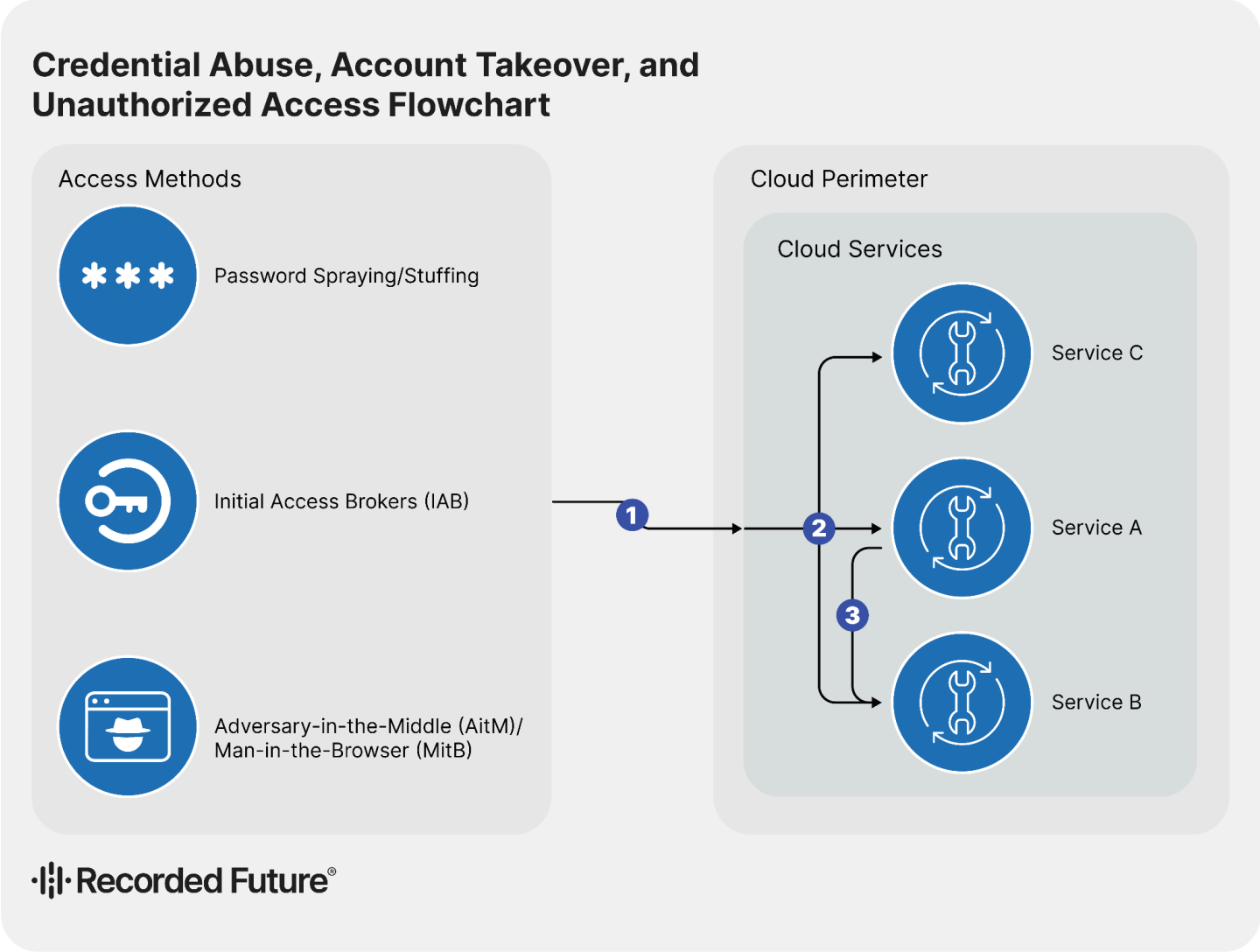

Figure 9: *Visual representation of potential unauthorized access attack vectors (Source: Recorded Future)*

① Detecting Unauthorized Access at the Edge

Threat actors will leverage multiple methods to gain unauthorized access to a cloud environment. As such, being able to detect and mitigate these threats at the edge of a cloud environment is paramount to the environment's security. Some methods for detecting unauthorized initial access to a cloud environment are described below:

- Monitor cloud firewall, network, and IAM logs for a high volume of login requests made rapidly. Additionally, be highly aware of instances where many incorrect requests are made, followed by a successful response, and then no other responses. This behavior is almost always indicative of password spraying or credential stuffing attacks.
- Compare cloud access logs against threat intelligence associated with known compromised credentials to determine if credentials known to be compromised were used to gain access to the cloud environment.

- Identify instances of impossible travel, which occurs when a cloud user authenticates from two locations simultaneously, from two physical locations that are far apart from each other.
- Compare the expected login behaviors associated with a user account with the most recent attempt; be mindful of the location of the attempt, the time of day, and what device was used for the attempt (if possible).

Additionally, these threats can be mitigated in the following ways:

- Ensure that MFA mechanisms are implemented instead of 2FA, requiring users to authenticate with more artifacts than their credentials and a code provided by an authentication application.
- Be mindful of phishing messaging and approach unexpected authentication requests with skepticism.
- Enable rate limiting on authentication attempts.
- Restrict what credentials can be used to perform API authentication.

② Unauthorized Access to Cloud Services and Resources

If a threat actor gains access to a cloud environment using compromised valid credentials, they will inherit the permissions granted to the account associated with those credentials. Threat actors will attempt to perform follow-on actions with these permissions after gaining access to a cloud environment, including determining the scope of their existing permissions. Some methods for hunting and mitigating actions threat actors can take at this stage are discussed below:

- Monitor API logs for a burst of requests made by a single user shortly after authentication associated with describing cloud services and assets. Since the threat actor is profiling the environment and testing their capabilities, many of these requests will fail, a log artifact that can also aid in identifying this behavior.
- Use behavioral analysis tools (some CSPs have this functionality built in natively) to identify accounts that access services and resources they do not commonly use.
- Identify attempts made by a cloud user to create additional cloud accounts, roles, or infrastructure (such as a VPC, compute instance, or container), which may indicate a threat actor attempting to establish persistence.
- Monitor and investigate compute instances that display abnormal hardware metrics, such as increased memory or GPU use statistics.
- Ensure that the principle of least privilege is applied when granting privileges to cloud accounts to prevent data and services from being unintentionally available to a compromised account.

③ Accessing Linked Cloud Resources

This threat is closely tied to the Unauthorized Access threat discussed above, and its mitigations are similar. After gaining access to a cloud environment, threat actors will often attempt to propagate throughout it, gaining access to as many resources and artifacts as possible. Importantly, threat actors

may also attempt to access parts of the cloud environment's infrastructure hosted in other data centers or entirely different cloud environments owned and managed by the same entity.

As stated above, the mitigations and hunting strategies previously discussed in this section are also effective for identifying this behavior. Specifically, ensure that the principle of least privilege is followed and that cloud access logs are retained across all cloud environments owned and managed by a single unifying entity.

Examples in the Wild

Insikt Group curated a list of events published within the past year that demonstrate the threats posed by unauthorized access, valid credential abuse, and account takeover. These events are discussed below.

Silk Typhoon Initial Access and Credential Abuse in IT Supply Chain Campaign

On March 5, 2025, Microsoft Threat Intelligence published research on Silk Typhoon, a Chinese state espionage threat actor, detailing how the group targets IT service providers, identity and privileged access management platforms, and other supply-chain infrastructure to gain initial access and then systematically abuses stolen credentials, keys, and cloud identities to conduct espionage against downstream customers. After compromising edge devices or third-party providers, Silk Typhoon pivots by using valid but stolen authentication materials, rather than relying solely on bespoke malware, to move into customer environments and access data aligned with China-based strategic interests, US government policy, and sensitive legal and law enforcement information.

Silk Typhoon's initial access phase combines exploitation of internet-facing vulnerabilities with direct credential abuse. The group rapidly operationalizes zero-day exploits against common edge appliances, including CVE-2025-0282 in Ivanti Pulse Connect VPN, CVE-2024-3400 in Palo Alto Networks GlobalProtect Gateway, CVE-2023-3519 in Citrix NetScaler ADC/Gateway, and the Microsoft Exchange "ProxyLogon" chain (CVE-2021-26855, CVE-2021-26857, CVE-2021-26858, CVE-2021-27065) to gain system-level access to public-facing services. In parallel, Silk Typhoon performs password spray attacks and other password abuse techniques, harvesting leaked corporate passwords from public repositories such as GitHub and using them to authenticate directly to corporate accounts. These intrusions primarily target IT providers, identity management, privileged access management, and remote monitoring and management solutions, where a successful compromise yields powerful credentials and trust relationships that can be leveraged across multiple customers.

Once initial footholds are established, Silk Typhoon shifts to systematically stealing and reusing authentication materials to deepen access and reach downstream tenants. In supply-chain intrusions since late 2024, the threat actor has stolen API keys and credentials from privileged access management, cloud application providers, and cloud data management companies, then used those keys to authenticate into downstream customer tenants as an administrator, perform reconnaissance, collect data, reset default admin accounts, deploy web shells, create additional users, and clear logs to hide activity.

Within victim environments, Silk Typhoon dumps Active Directory, steals passwords and secrets from key vaults, and targets AADConnect and Entra Connect servers to obtain privileges that span both on-premises and cloud identity planes. The group then abuses legitimate cloud identities by compromising or creating service principals and OAuth applications with administrative permissions, adding their own credentials to already-consented applications, compromising multi-tenant applications, and developing new Entra ID applications named to blend in with the environment. Using these trusted identities, Silk Typhoon accesses Microsoft Graph and Exchange Web Services APIs to exfiltrate large volumes of email, OneDrive, and SharePoint data across multiple tenants.

Scattered Spider Use of Valid Credentials in Logistics-Firm Intrusion

On June 27, 2025, Information Security Media Group published a ReliaQuest-sourced teardown describing how Scattered Spider (aka Octo-Tempest/UNC3944) compromised a logistics firm by targeting the CFO, abusing valid credentials, and manipulating MFA and privileged password vaults to achieve and re-attempt access.

Attackers began by targeting C-suite personnel and collecting publicly available identity data for the CFO, including their date of birth and the last four digits of their Social Security number. They used this information against the company's public-facing Oracle Cloud portal to confirm the CFO's employee number and attempted to authenticate with stolen CFO credentials; these initial login attempts failed due to enforced MFA.

Next, Scattered Spider impersonated the CFO in a phone call to the IT help desk, presenting a plausible scenario that leveraged the previously gathered identity details. Help-desk staff were convinced to reset both the MFA device and credentials tied to the CFO's account, effectively transferring control of a legitimate executive identity to the attackers and turning MFA into an enabler rather than a barrier. Using the now-compromised CFO account, the attackers accessed the environment via single sign-on and VMware Horizon virtual desktops, then used valid credentials to enumerate identities in Microsoft Entra ID, assign an Exchange administrator role to a compromised account, and access high-value services, including the company's CyberArk privileged access vault and Snowflake data warehouse.

Inside the CyberArk vault, the threat actor used vaulted secrets associated with approximately 1,400 accounts to move through the environment with legitimate credentials, ultimately dumping the `NTDS.DIT` Active Directory database and accessing mailboxes belonging to the CISO and other senior staff. Even after defenders fully ejected the threat actor roughly 62 hours after initial breach, Scattered Spider continued attempting to regain access by logging in with passwords cracked from the previously stolen privileged account hashes; these later attempts used valid (but now protected) credentials and triggered MFA prompts, preventing reentry but demonstrating repeated credential-based access attempts.

DPRK Abuse of Authentication in the ByBit Heist

On May 5, 2025, Elastic Security Labs published a report detailing how DPRK-linked TraderTraitor operators stole roughly 400,000 ETH from ByBit by abusing Safe{Wallet}'s AWS authentication. The report focuses on how attackers moved from a compromised Safe{Wallet} macOS developer

workstation to authenticated access in Safe{Wallet}'s AWS account, and then used that cloud identity to tamper with the Safe{Wallet} frontend that ByBit approvers trusted for transaction validation.

The attack chain began with the compromise of "Developer1" at Safe{Wallet} through a backdoored crypto-themed Python application on macOS that exploited unsafe `PyYAML` deserialization to run arbitrary code. The malware dropped a Mythic Poseidon agent and a Python stealer that systematically collected authentication materials and configuration details from the host, including Secure Shell (SSH) private keys, AWS access keys, and secret keys from `~/.aws/credentials`, environment-based session tokens, and macOS Keychain data.

Using the stolen AWS access key ID, secret key, and the active session token, the DPRK operators authenticated directly into Safe{Wallet}'s AWS account within the token's twelve-hour validity window. The cached session token, initially created by the victim via AWS mechanisms using GetSessionToken and AWS SSO, stored in standard AWS configuration locations, was replayed from the attacker's infrastructure using the AWS CLI or Boto3, which handles request signing and token use. The attackers operated entirely as the authenticated Safe{Wallet} user, with the MFA context already satisfied at session creation, and issued AWS API calls under that identity, including attempts to register additional MFA devices and to access storage resources.

With valid AWS authentication materials and an active user session, the operators then explored Safe{Wallet}'s cloud environment and accessed the S3 bucket hosting the statically built Safe{Wallet} Next.js frontend. From there, they downloaded, modified, and redeployed the application bundle so that when ByBit approvers loaded the dApp, transaction details displayed in the interface were silently altered while signatures and on-chain workflows still appeared normal.

Third-Party Compromise

Key Takeaways

- Threat actors increasingly reach cloud environments by compromising trusted third parties, such as SaaS, identity providers (IdPs), managed service providers (MSPs), CI/CD platforms, and package ecosystems, rather than targeting victims directly.
- Once a third-party or supply chain component is compromised, attackers inherit its trust and permissions, allowing them to push malicious code, access data, and modify configurations within multiple cloud tenants.
- The real risk is that malicious activity arrives "pre-trusted" through standard integrations and deployments, making it difficult to distinguish from legitimate vendor behavior without strong controls and monitoring on third-party access and code.

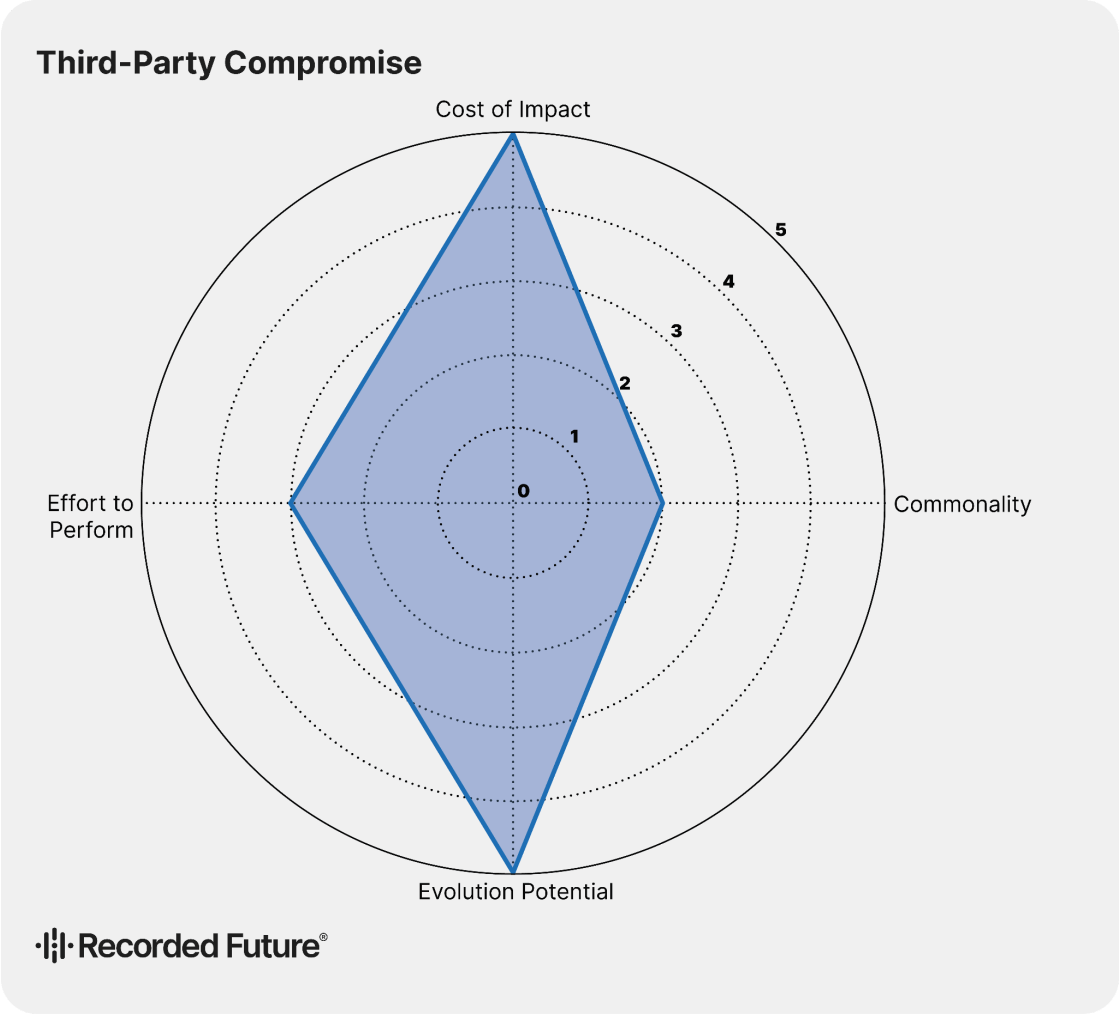

Figure 10: *Radar chart illustrating third-party compromise as a threat vector (Source: Recorded Future)*

Cost of Impact: 5 (Severe)

Abuse of services and compromise of a third-party product can have a severe impact on a victim and an organization, both monetarily and reputationally. Third-party compromise can result in the loss of infrastructure and information for the victim, but the potential reputational impact for the third party can lead to a loss of business and contracts.

Commonality: 2 (Low)

Third-party compromise and abuse have a relatively low commonality rating compared to other threats discussed in this report. While the attacks are not the most prevalent among those Insikt Group has observed, they are used when attackers need indirect, trusted access to reach a victim.

Evolution Potential: 5 (Severe)

As cloud adoption accelerates and organizations rely on an ever-growing ecosystem of vendors for hosting, administration, storage, and computation, the potential for third-party compromise will continue to increase and evolve. Each new integration, management service, or colocated data center relationship expands the attack surface and creates additional trust pathways threat actors can exploit. As a result, third-party risk is likely to become both more frequently abused and more technically sophisticated over time, with attackers combining cyber and, where feasible, physical avenues of attack to reach high-value cloud infrastructure through the least-secure entity.

Effort to Perform: 3 (Moderate)

The effort required to compromise third parties varies widely based on the organization and the attacker's objectives, resulting in a generally moderate overall effort. Well-resourced or highly regulated providers with mature physical and cybersecurity controls can demand significant planning, capability, and time to breach. In contrast, smaller or less mature vendors, legacy partners, or lightly monitored integrations may be compromised with comparatively simple techniques and commodity tooling.

Threat Summary

Third-party compromise and abuse in cloud environments describes threat activity that targets the external vendors, platforms, and components organizations rely on, rather than the organization directly. Instead of breaking into the victim's own accounts from the outset, threat actors compromise a connected identity provider, SaaS platform, MSP, CI/CD service, or other integrated provider and then abuse that trusted connection to reach cloud workloads and data. Because these third-party relationships are deeply embedded into authentication, monitoring, deployment, and business workflows, they often carry broad permissions and implicit trust that can be repurposed for access, persistence, and action on objectives inside victim cloud environments.

In the context of third-party risk (TPR) abuse, threat actors focus on the trust and permissions granted to external services and applications. Compromised vendor portals, support accounts, or management consoles can be used to issue legitimate-looking login sessions, API calls, and configuration changes

into the customer's cloud tenants. OAuth applications, marketplace integrations, and service accounts associated with third parties may be overprivileged or widely deployed across tenants, allowing an attacker who controls the third party to pivot into multiple customers simultaneously. From there, the threat actor can enumerate resources, manipulate identities and policies, or deploy further tooling while appearing as a standard, trusted integration.

Supply-chain compromise in cloud environments extends this model into the software and build pipelines that produce and deliver code to cloud-hosted systems. Threat actors may target source-code repositories, CI/CD platforms, artifact registries, or update mechanisms used by cloud-native applications and services. By tampering with build configurations or injecting malicious logic into dependencies, images, or update channels, they can ensure that compromised components are automatically built, signed, and deployed into production workloads. This gives them a covert and scalable foothold that inherits the permissions and reach of the affected services and applications across cloud accounts and regions.

Package abuse focuses specifically on public or internal package ecosystems that feed into cloud applications and automation. Threat actors publish or modify libraries, containers, or infrastructure-as-code modules in ways that cause them to be pulled into victim environments through normal development and deployment processes. Techniques such as typosquatting, dependency confusion, or hijacking of abandoned packages allow malicious code to enter build pipelines and runtime environments without direct intrusion into the target organization. Once those packages are executed in cloud workloads, they can access credentials, call cloud APIs, exfiltrate data, or stage additional payloads under the guise of standard application behavior, making third-party compromise, supply-chain compromise, and package abuse tightly interlinked risks for cloud-centric organizations.

Outlook

Threat actors will continue to prioritize third-party compromise and abuse as an efficient means of reaching multiple cloud environments through a single, trusted channel. TPR abuse, supply-chain compromise, and package abuse all allow threat actors to inherit the privileges, reach, and implicit trust granted to vendors, platforms, and components that are deeply woven into cloud identity, deployment, and operations. As organizations increase their reliance on SaaS, MSPs, CI/CD platforms, and external integrations, the potential impact from a single upstream compromise will continue to grow.

Because these attacks ride on top of legitimate integrations, signed artifacts, and expected package use, they will remain difficult to distinguish from regular activity within cloud environments. The combination of broad access, automatic propagation through pipelines, and the ability to masquerade as a trusted third party ensures that TPR abuse, supply-chain compromise, and package abuse will remain high-value strategies for threat actors seeking scalable, low-friction access to cloud workloads and data.

Mitigations and Detections

Figure 11 demonstrates a hypothetical attack chain associated with third-party risk attack methods and behaviors. Throughout this visual, Insikt Group identified parts of the attack chain where defenders can most efficiently hunt for and mitigate behaviors associated with cloud abuse.

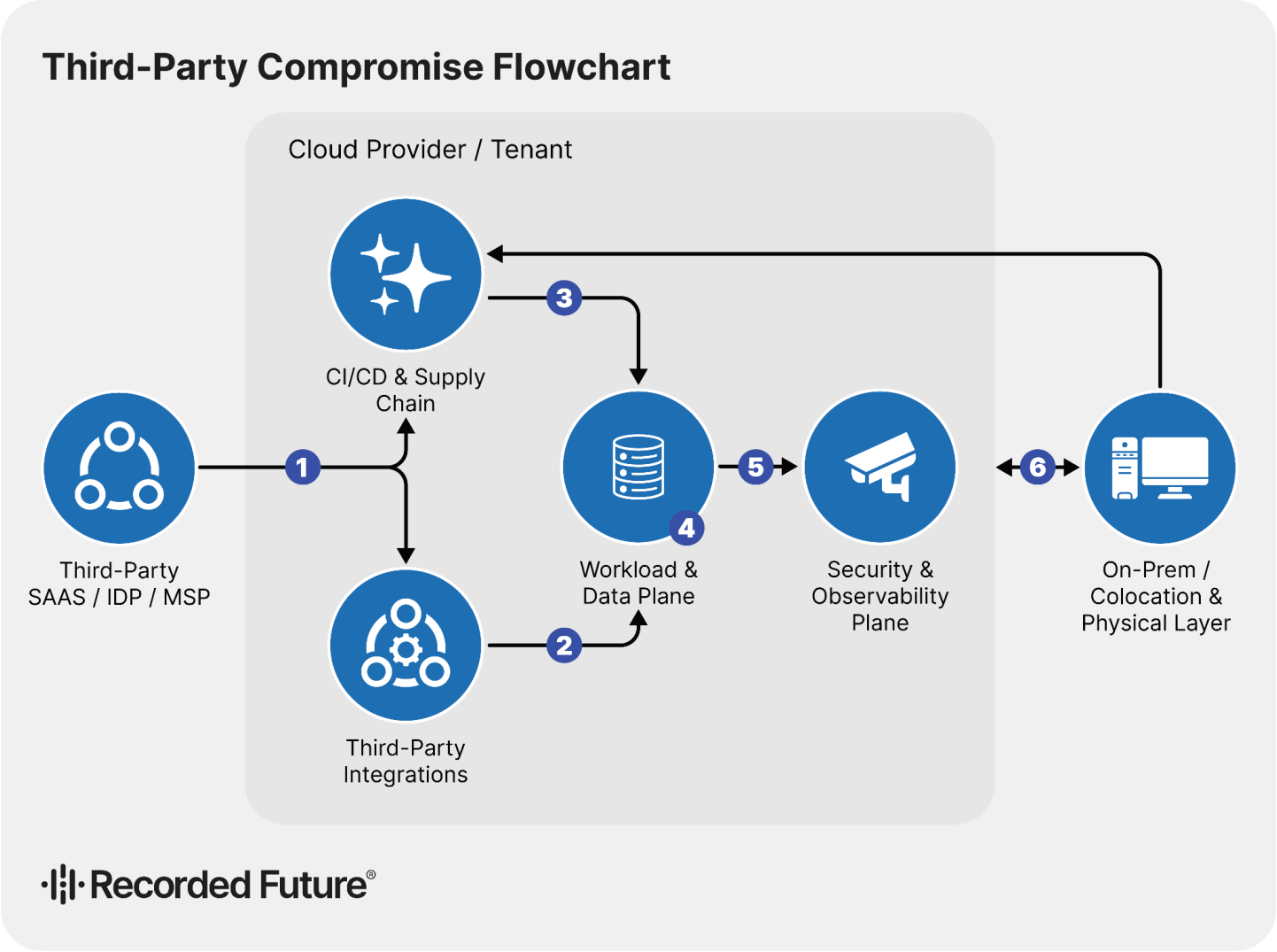

Figure 11: *Visual representation of potential third-party risk and detections (Source: Recorded Future)*

① Third-Party SaaS, IdP, and MSP Trust Relationships

Threat actors will target third-party SaaS, identity providers, and managed service providers (such as SSO platforms, remote monitoring tools, or IT service management systems) to gain indirect access to cloud tenants. Compromising these providers enables attackers to establish authenticated sessions, manipulate configurations, or introduce changes into customer environments through trusted channels.

Defenders can mitigate these threats in the following ways:

- Establish strict onboarding and risk assessment for third-party providers, including requirements for MFA, logging, incident notification service level agreements (SLAs), and clear scoping of their access into your cloud tenants.
- Monitor and regularly review all SSO, federation, and API-based trust relationships (such as SAML, OpenID Connect [OIDC], and vendor-specific integrations), paying close attention to newly added connections, newly granted scopes, or expanded permissions over time.
- Correlate vendor-originated activity (such as support logins, remote sessions, or automated remediation actions) in cloud and IdP logs, and alert on activity that occurs outside agreed maintenance windows, locations, or expected patterns.

② Third-Party Integrations Inside the Cloud Tenant

Threat actors will abuse over-privileged third-party integrations, such as OAuth applications, marketplace add-ons, service principals, and API keys, to perform operations directly within cloud environments. Once these integrations are compromised upstream, attackers can pivot into multiple tenants and conduct discovery, data access, or configuration changes under the guise of legitimate apps.

Defenders can mitigate these threats in the following ways:

- Implement a formal approval process for third-party cloud applications and service principals, requiring justification, scope review, and security validation before granting tenant-wide or high-privilege permissions.
- Continuously monitor for new app consents, increased permission scopes, or unusual usage of existing service principals, and alert when an integration begins accessing resources or performing actions outside its documented purpose.
- Enforce least privilege for all third-party integrations, limiting them to specific roles, resource groups, and APIs, and regularly re-certify access to ensure unused or unnecessary permissions are removed.

③ CI/CD Pipelines, Artifact Registries, and Package Abuse

Threat actors will exploit software supply chains by tampering with CI/CD pipelines, internal artifact registries, or package feeds, as well as public package ecosystems. By injecting malicious dependencies, images, or build steps, they can ensure compromised components are automatically built, signed, and deployed into cloud workloads as part of routine development and release processes.

Defenders can mitigate these threats in the following ways:

- Harden CI/CD platforms by enforcing MFA for pipeline administrators, restricting who can edit build configurations, and maintaining audit logs for all pipeline and credential changes.

- Require provenance and signing for artifacts (such as container images, binaries, and infrastructure-as-code templates), and enforce that production environments only run artifacts that pass signature and integrity verification.
- Limit the use of public package repositories to vetted sources, implement dependency and software composition analysis, and alert on the introduction of new, unreviewed dependencies, particularly when they resemble existing internal or popular packages (for example, typosquatting or dependency confusion patterns).

④ Workloads and Data Plane Executing Third-Party Components

Threat actors will leverage compromised third-party code running in cloud workloads, applications, containers, serverless functions, and background jobs to execute their objectives. Once deployed, these components can silently exfiltrate data, abuse workload identities to call cloud APIs, stage ransomware, or create covert channels that blend with normal application behavior.

Defenders can mitigate these threats in the following ways:

- Deploy runtime security as well as EDR and Cloud Workload Protection Platform (CWPP) capabilities on workloads to detect anomalous behavior originating from application processes, such as unexpected outbound connections, abnormal API calls, or access to sensitive secrets and data stores.
- Apply the principle of least privilege to workload identities and secrets, ensuring that even if a third-party component is compromised, it can only interact with a minimal set of resources and cannot directly manage keys, backups, or broader tenant configurations.
- Enforce network segmentation and egress controls for workloads, restricting outbound connectivity to approved destinations (such as known APIs, registries, and partner endpoints), and alert on connections to unknown IPs or domains from production workloads.

⑤ Security and Observability Plane: Vendor Telemetry and Monitoring

Threat actors will attempt to exploit or degrade the security and observability plane, including integrations with third-party security tools and monitoring services. By tampering with log forwarding, disabling agents, or abusing shared trust with security vendors, attackers may reduce visibility into their activity or generate misleading signals.

Defenders can mitigate these threats in the following ways:

- Centralize log collection from cloud services, IdPs, vendors, and on-prem infrastructure into hardened, controlled logging environments, and verify that vendor-provided logs are complete, timely, and correlated with internal telemetry.
- Monitor changes to logging configurations, security tool integrations, and agent deployments, and generate high-severity alerts when critical data sources are disabled, downgraded, or suddenly become unavailable without an approved change.

- Validate that detections and incident notifications from managed security providers are being delivered and actioned as expected, and periodically simulate vendor-originated incidents or test cases to confirm that integrations and response workflows function correctly.

⑥ Physical Data Centers, Colocation, and On-Premise Pivots

Threat actors may exploit physical security weaknesses at data centers, colocation facilities, or on-premises environments connected to the cloud via direct connect, VPN, or multiprotocol label switching (MPLS). By abusing contracted technicians, stolen badges, unsecured network closets, or poorly controlled smart-hands access, they can tamper with on-premise switches, cross-connects, or management consoles, creating a pivot into cloud environments or disrupting connectivity.

Defenders can mitigate these threats in the following ways:

- Enforce strong physical access controls at on-premise and colocation sites, including badge plus biometric access, visitor registration, escort policies for third-party technicians, and strict separation of customer cages or racks.
- Harden network and compute infrastructure in data centers by locking racks, securing console ports, enabling port security and network access control on switches, and monitoring for unauthorized devices or link changes on uplinks and cross-connects.
- Require cloud and data center providers to meet defined physical security standards and audit requirements, and ensure that connectivity paths between on-premise and cloud (such as direct connect circuits, VPN gateways, and SD-WAN appliances) are monitored for configuration changes, link disruptions, or unexpected routing behavior that could signal tampering.

Examples in the Wild

Insikt Group curated a list of events published during 2025 that demonstrate the threats posed by third-party compromise. These events are discussed below.

Third-Party Risk from Oracle E-Business Suite Zero-Day Data Theft Campaign

On October 28, 2025, SecurityWeek (Eduard Kovacs) described an extortion campaign in which Cl0p ransomware threat actors, linked by the security community to an unknown cluster of the FIN11 profit-driven threat group, exploited Oracle E-Business Suite (EBS) zero-day vulnerabilities to steal data from Oracle customers, including Schneider Electric, Emerson, Logitech, GlobalLogic, and Dartmouth College. These sources collectively depict a third-party risk scenario in which compromise of a shared Oracle EBS platform at the vendor layer cascades into multi-sector data breaches, with victim organizations emphasizing that the incidents affected Oracle-hosted or other third-party systems rather than their core IT or production environments.

The attack chain began with the exploitation of at least one Oracle EBS zero-day vulnerability, tracked as CVE-2025-61882, against Oracle EBS instances starting in early August 2025, enabling unauthorized access to EBS platforms used by multiple organizations. For GlobalLogic, BleepingComputer reports that threat actor activity in the Oracle environment occurred between July 10 and August 20, 2025, with

confirmed access to Oracle and data exfiltration on October 9, 2025, followed by notification to more than 10,000 current and former employees whose information was stored in the EBS deployment. GlobalLogic's disclosure states that the incident did not target or impact systems outside its Oracle platform and that the affected data was confined to HR information stored in Oracle. SecurityWeek notes that over 50 victims have been identified on Cl0p's leak website to date, and that the group is releasing stolen information in large archive sets and torrents, indicating systematic exfiltration from Oracle environments across multiple organizations.

GlobalLogic reports that attackers exploited an Oracle EBS zero-day vulnerability to steal personal information belonging to 10,471 employees, including HR-collected data such as names, addresses, phone numbers, email addresses, dates of birth, passport information, national or tax identifiers (including Social Security numbers), and bank account details. Dartmouth College has disclosed that an unauthorized threat actor accessed files from its Oracle EBS servers between August 9 and August 12, 2025. Subsequent review identified files containing names and Social Security numbers, as well as documents with financial account information, for at least 1,494 individuals notified to date. In both cases, the adversary's behavior involves exploiting the shared Oracle EBS platform, collecting sensitive identity and financial data stored there, and then using the Cl0p leak site as leverage in an extortion model focused on data theft rather than disrupting the victims' primary business systems.

Logitech reports that an unauthorized third party exploited a zero-day vulnerability in a third-party software platform to extract limited data about employees, consumers, customers, and suppliers from an internal IT system. The company states that no national ID numbers, credit card data, products, business operations, or manufacturing systems were affected. SecurityWeek links this activity to the same Oracle EBS campaign claimed by Cl0p, which has listed Logitech on its leak site and published 1.8 TB of alleged Logitech archives, alongside large Oracle-sourced data sets for Emerson (2.7 TB), Schneider Electric (116 GB), Harvard University, and Envoy Air, underscoring systemic third-party exposure across multiple sectors.

Malicious npm Package "@acitons/artifact" Targeting GitHub Actions via Typosquatted Dependency

On November 10, 2025, Veracode Threat Research published an analysis of a malicious npm package, "@acitons/artifact", that typosquats the legitimate GitHub Actions dependency "@actions/artifact" and targets GitHub-owned repositories using the GitHub Actions CI/CD platform. The report states that six malicious versions of this package included a post-install hook that downloaded and executed additional malware not detected by popular antivirus products at the time of analysis. Furthermore, the malicious versions were removed, while the package name itself remained on npm. Veracode assessed that the campaign intended to execute during builds of GitHub-owned repositories, exfiltrate build-environment tokens, and then use those tokens to publish new malicious artifacts on GitHub, highlighting software supply-chain risk consistent with OWASP Top 10 2025 "A03:2025 – Software Supply Chain Failures."

The attack chain begins with supply-chain compromise via a deceptive package.json that closely mirrors the legitimate "@actions/artifact" metadata, including the same homepage, repository URL, and description, but swaps the letters "ti" for "it" in the package name to form "@acitons/artifact". The

malicious versions (4.0.12 through 4.0.17) define a postinstall script that runs curl against a GitHub Gist URL under the user "jmasdg" to fetch a binary named `harness`, which is then executed on the GitHub Actions runner as part of npm installation. The blog notes that VirusTotal did not initially show popular antivirus vendors detecting the `harness` binary as malicious.

Analysis of the `harness` binary reveals that it is an obfuscated shell script compiled with the Shell Script Compiler, which includes a hard-coded expiration mechanism to prevent execution after 2025-11-06 UTC. A related sample (`tester`) expires one day later. On execution, `harness` re-runs itself under bash after setting an environment variable to alter its runtime behavior, then outputs, extracts, and executes an embedded Node.js package containing an obfuscated `verify.js` script. `verify.js` performs environment checks for GitHub Actions by examining `GITHUB_` environment variables and proceeds only when the repository owner is GitHub, exiting otherwise. The script writes an encrypted data blob (`env.enc`) after collecting environment data, including tokens available to the build job.

The payload obtains an AES encryption key from *hxxps://83hfhjasksn.hopto[.]org:443/kljkalsd/ajkl12389/slkj1n_189n*, described as a service used for resolving DNS names, encrypts the collected data, and exfiltrates the encrypted file to *hxxps://laughing-space-capybara-x5g6rjxq7jwvfp6q6-443.app.github[.]dev/sllkjdsss_user-dasd.txt*. The campaign appears to focus on GitHub's own repositories and a low-activity user account, "y8793hfiuashfjksdhfjsk", which may be used for testing purposes. Historical inspection also identified and blocked twelve versions of another malicious npm package, `8jfiesaf83`, earlier in the month. GitHub later provided a statement that the referenced packages were part of a tightly controlled red team exercise, asserting that GitHub systems and data were not at risk.

whoAMI Cloud Image Name Confusion Attack Against AWS AMIs

On February 12, 2025, Datadog Security Labs published research describing the "whoAMI" cloud image name confusion attack, a misconfiguration pattern in how many software projects retrieve Amazon Machine Image (AMI) IDs that allows attackers who publish specially crafted AMIs to gain code execution within vulnerable Amazon Web Services (AWS) accounts. The research identifies that this vulnerable pattern, retrieving AMIs by name via ec2:DescribeImages without constraining the owner and then using the most recent returned image, appears across private and open-source repositories; Datadog estimates that roughly 1% of organizations using AWS are affected and that, at scale, the technique enables access to thousands of AWS accounts. The team also confirmed that internal non-production AWS systems were susceptible to the same pattern before AWS fixed the issue through its Vulnerability Disclosure Program (VDP).

From the victim's perspective, the attack chain begins when code retrieves an AMI ID using the `ec2:DescribeImages` API with a name filter, omits the owner, owner-alias, or owner-id parameters, and then selects the most recent image from the results. The same AMI ID is subsequently used to create EC2 instances, launch templates, and deploy other infrastructure. Datadog illustrates this with a common Terraform data `aws_ami` pattern that sets `most_recent = true` and filters on `ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*` but does not specify an owner. This causes the AWS API call to return public community AMIs from any account, and Terraform

automatically chooses the newest AMI in that list. Similar anti-patterns appear in Go and other languages where code logic calls `DescribeImages` with only a name filter (for example, values matching `amzn2-ami-hvm-2.0.*-x86_64-gp2`) and then passes the first returned `ImageId` directly into `RunInstances`.

An attacker exploiting this configuration publishes or clones an AMI whose name matches the victim's search pattern and ensures it is newer than any legitimate images. Then, the attacker either makes the AMI public or privately shares it with the targeted AWS account. Datadog's example uses a malicious AMI named `ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-whoAMI`, which will be selected whenever vulnerable Terraform or API code resolves "latest Ubuntu Focal" via name-only search. When such code runs, the EC2 instance is launched from the attacker's AMI rather than the intended provider's image, allowing the attacker to execute remote code on the victim's instance and enabling any subsequent post-exploitation activity supported by the backdoor embedded in the AMI. To demonstrate this safely, Datadog created an AMI with a C2 backdoor, shared it privately with a controlled "victim" account, and showed that the vulnerable pattern reliably instantiated the attacker's AMI.

The vulnerable AMI-lookup pattern is not limited to Terraform. It appears in multiple ecosystems, including Python, Go, Java, Pulumi, and Bash scripts that use the AWS CLI, whenever DescribeImages is filtered only by name and the "most recent" image is selected. Datadog reports that roughly 1% of monitored organizations exhibited this anti-pattern, affecting thousands of distinct AWS accounts. After validating the attack in a controlled environment, Datadog collaborated with AWS to test whether internal AWS systems would consume benign "doppelgänger" AMIs with the prefix `amzn2-ami-hvm-2.0`; tens of thousands of SharedSnapshotVolumeCreated events showed that AWS services were indeed pulling these AMIs, indicating that some internal, non-production environments were using insecure AMI retrieval logic before the issue was remediated. The research also highlights a vulnerable AWS-maintained tool, aws-simple-ec2-cli, whose hard-coded wildcard AMI name patterns (such as `amzn2-ami-hvm-2.?.????????.?-*-gp2`) and lack of owner checks in getDescribeImagesInputs allowed a cloned AMI `amzn2-ami-hvm-2.0.20241014.0-RESEARCH-x86_64-gp2` (AMI ID `ami-08b96120f4ae90485`) to be selected over the official image until the tool was updated.

General Mitigations and Recommendations

While specific mitigation strategies have been discussed for each of the cloud threats outlined in this report, Insikt Group identified general mitigation strategies that can be implemented based on the events discussed in this report. The following threat hunting and mitigation strategies can be implemented in any cloud environment to increase the environment's security posture:

Vulnerability and Third-Party Exposure Management

- Enforce aggressive patching and configuration management for internet-exposed cloud and hybrid services, treating vendor advisories for critical CVEs as triggers for patching products that front cloud workloads or identity infrastructure.
- Enforce strict vendor and third-party risk controls for shared cloud platforms (for example, Oracle EBS, Barracuda ESG appliances, cloud email gateways), including contractual security requirements, patch service level agreements (SLA), segmentation of hosted data, and a clear understanding of whose logging and detection covers the shared environment.
- Minimize sensitive data stored in third-party cloud platforms to the minimum required for business operations, and segment HR, financial, and identity data so that compromise of a single hosted system does not automatically expose full identity and banking datasets across the organization.

Network and Perimeter Controls

- Minimize direct internet exposure of management and monitoring planes by placing them behind cloud VPNs, private access tools, or zero-trust products and restricting access to internal cloud infrastructure.
- Enforce strong egress controls on cloud workloads and user devices, explicitly scrutinizing or restricting traffic to general-purpose SaaS platforms that can act as covert C2 or dead-drop channels.
- Deploy granular network controls around cloud storage and backup services (S3, Azure Storage, Recovery Services Vaults, snapshots, EBS volumes) to prevent broad access, and require elevated workflows for operations such as exposing storage to the internet, retrieving access keys, or deleting backups and restore points.
- Ensure DDoS protection is enabled and correctly scoped for all internet-facing cloud endpoints, and regularly exercise operational readiness through simulated large-scale volumetric attacks to confirm that workloads remain available when targeted by IoT botnets and similar infrastructure.

IAM Hardening

- Implement the principle of least privilege in cloud IAM for both human and non-human cloud accounts, explicitly removing broad roles and powerful permissions, such as global administrator and owner roles, from day-to-day and developer accounts.

- Require strong, enforceable MFA on all privileged and directory-synchronized accounts and prohibit any high-privileged account from operating without MFA, including service accounts where password-hash synchronization is used.
- Harden MFA and help desk workflows so that high-value identities (such as C-suite accounts, cloud administrators, and other root accounts) cannot have MFA or passwords reset based solely on static personal data, and ensure help desk tooling clearly flags privileged accounts and enforces additional verification steps before changing authentication factors.
- Restrict cross-tenant and cross-account trust relationships in cloud identity platforms, allowing sts:AssumeRole, SAML federation, and multi-tenant OAuth apps only where explicitly required and reviewing them regularly for abuse patterns similar to those used in Storm-0501 and Silk Typhoon campaigns.
- Implement strong governance for service principals and OAuth applications in cloud identity (for example, Entra ID), including reviewing app consents, monitoring for new high-privilege apps or credentials added to existing apps, and validating that multi-tenant applications are necessary for system operation.
- Constrain session-token lifetime and monitor for unusual use of cached tokens from atypical locations, especially where an attacker could replay a developer's existing authenticated session to access AWS or other cloud control planes without re-prompting MFA.
- Implement robust processes for account deletion and access revocation to prevent accounts associated with former staff or contractors from being reused as entry points to cloud tenants or repositories. Periodically verify that no shared or orphaned admin credentials remain in an environment.
- Monitor privileged access vaults and key stores (for example, CyberArk, cloud key vaults, PAM platforms), and alert on actions such as bulk secret access, high-volume credential export, and new access grants that could support lateral movement into cloud services at scale.

Software Supply Chain and CI/CD hardening

- Treat all API keys, cloud access keys, and session tokens as high-value secrets. Scan public and internal repositories for exposed secret or sensitive keys, rotate any discovered keys, and ensure developer tools do not leave long-lived credentials in predictable file system locations or plaintext configuration files.
- Restrict cloud image selection workflows by always constraining AMI lookups to trusted owners and accounts, and use platform controls such as AWS "Allowed AMIs" plus periodic scanning (for example, with tooling similar to whoAMI-scanner) to detect and prohibit workloads booted from unverified or public AMIs.
- Review Infrastructure-as-Code, scripts, and software development kit (SDK) usage for unsafe AMI lookup patterns and enforce coding standards that require explicit owner filters for cloud images across Terraform, CLI tooling, and automation code.
- Isolate and harden cloud ML and LLM services by separating developer and production accounts, restricting internet egress from notebooks, limiting execution roles to only required actions, and preventing developer notebooks from assuming administrative or cross-account roles in production LLM environments.

- Treat LLM guardrails, agents, and retrieval-augmented generation knowledge bases as sensitive configuration surfaces. Protect these resources with change controls, monitor for modifications to guardrail block messages or knowledge base data sources, and require review for any changes that could inject malicious download links or executable code into user-visible responses.
- Protect cloud-hosted static frontends and application bundles (for example, S3-backed web apps) with strict change control, code-signing, or integrity checks. Monitor for unexpected changes to objects or buckets that present user interfaces.
- Strengthen security around continuous integration and continuous delivery or development (CI/CD) environments (for example, GitHub Actions) by scrutinizing dependency declarations for typosquatted packages, pinning versions where feasible, and implementing automated checks to block or flag suspicious packages that add postinstall hooks or fetch binaries from untrusted domains.
- Restrict CI/CD job tokens and other ephemeral credentials to the narrowest required scopes and lifetimes, and monitor pipeline hosts for outbound connections to unusual domains or encrypted data uploads that could indicate token exfiltration from build environments.
- Protect corporate and cloud-provider code repositories (for example, GitHub) with strong MFA, access reviews for former employees and contractors, and backup/archival strategies that allow rapid restoration if a destructive actor wipes repositories or branches.

Monitoring, Detection, and Response

- Monitor for identity operations indicative of cloud ransomware playbooks, such as elevation of access (for example, Azure elevateAccess), mass assignment of owner or equivalent roles across subscriptions, creation of new federated trusts, and large-scale storage enumeration and key retrieval.
- Ensure logging is comprehensive and retained for identity, control-plane, and data-plane actions across cloud providers. Integrate those logs with analytic workflows capable of detecting credential abuse, cross-tenant pivots, and mass exfiltration.

Endpoint and Administrative Workstation Hardening

- Harden developer and administrative endpoints that bridge into cloud control planes by prohibiting cracked or pirated software, enforcing endpoint protection and application control, and treating any system with cloud access keys or SSO tokens as a high-value asset requiring stronger monitoring and restriction.

Backup and Recovery Hardening

- Use cloud-native protections, such as immutability (for example, read-only permissions), locks, and retention policies, on backup containers, snapshots, and vaults. This ensures that even highly privileged roles cannot easily delete or alter recovery data in a single step during ransomware or destructive cloud attacks.

The most common CSPs offer native cloud services that enable the mitigation strategies discussed above, including environment scanning for potential misconfigurations, baseline account behavior monitoring, network traffic monitoring, role-based access control or IAM suites, and data protection. However, in cloud environments where there are fewer managed cloud services and more responsibility for defense is more greatly assumed by the cloud defenders and architects, ensuring that clear policies and baselines are specified for the creation, management, and upkeep of an organization's cloud infrastructure is paramount.

Outlook

The events discussed in this report from 2025 indicate that threat actors have a growing understanding of both cloud environments and the benefits that they provide as part of an attack chain. Events related to cloud ransomware, threat actor abuse of cloud LLM and ML services, and the abuse of various cloud services for C2 infrastructure all support this claim and further demonstrate that legitimate cloud resources can be abused to achieve malicious aims. Threat actors will almost certainly continue to identify additional cloud platforms and products that can be abused to perform malicious actions. Similarly, threat actors will continue to leverage cloud-native services as methods in attack chains against cloud environments, specifically since these methods are the most effective option for threat actors to disrupt and steal from cloud environments.

To perform these actions, threat actors must continue to gain unauthorized access, both at the perimeter of cloud environments and within them. The most common ways this access will be granted, both externally and internally, will remain misconfiguration and valid credential abuse, in that order. However, threat actors are demonstrating that exploiting products embedded at the perimeter of cloud environments, or in instances of private cloud, is a viable option for cloud compromise. As the adoption rates of cloud computing continue to increase, the volume of new products and services will also rise, leading to an increase in potential vulnerabilities and misconfigurations. Threat actors will continue to exploit these weaknesses as a result. Furthermore, as more programs and services written with AI-driven code emerge in cloud ecosystems, the number of vulnerabilities from unchecked code will also increase, providing avenues for threat actors to exploit cloud users and services.

Finally, data theft remains the most significant outcome of attacks on cloud environments; however, threat actors are increasingly pursuing additional outcomes, including coercion techniques in the cloud and various forms of resource hijacking. As such, the overall dwell time associated with attacks against cloud environments may increase, as threat actors may require additional time and resources for an attack chain to reach fruition, or to maximize the benefit of their attacks. While this dwell time may provide defenders more time to identify and remediate malicious actions within their environments, it also provides attackers with more opportunity to identify sensitive information. The longer these entities can maintain access, the greater the risk of compromise for the entire cloud environment.

Recorded Future reporting contains expressions of likelihood or probability consistent with US Intelligence Community Directive (ICD) 203: Analytic Standards (published January 2, 2015). Recorded Future reporting also uses confidence level standards employed by the US Intelligence Community to assess the quality and quantity of the source information supporting our analytic judgments.

About Insikt Group®

Recorded Future's Insikt Group, the company's threat research division, comprises analysts and security researchers with deep government, law enforcement, military, and intelligence agency experience. Their mission is to produce intelligence that reduces risk for customers, enables tangible outcomes, and prevents business disruption.

About Recorded Future®

Recorded Future is the world's largest intelligence company. The Recorded Future Intelligence Operations Platform provides the most complete coverage across adversaries, infrastructure, and targets. By combining precise, AI-driven analytics with the Intelligence Graph® populated by specialized threat data, Recorded Future enables cyber teams to see the complete picture, act with confidence, and get ahead of threats that matter before they impact your business. Headquartered in Boston with offices around the world, Recorded Future works with more than 1,900 businesses and government organizations across 80 countries.

Learn more at recordedfuture.com