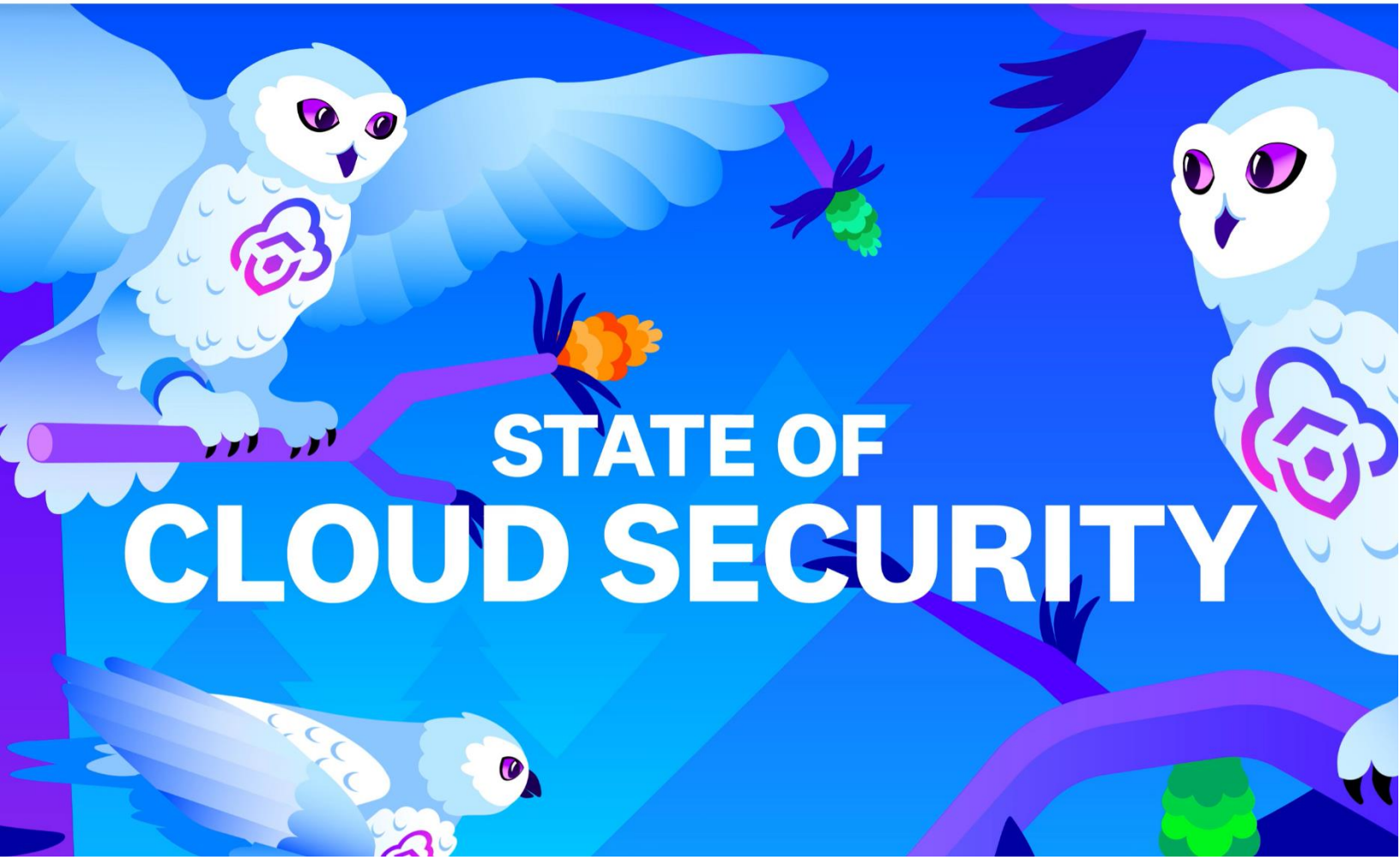
DATADOG



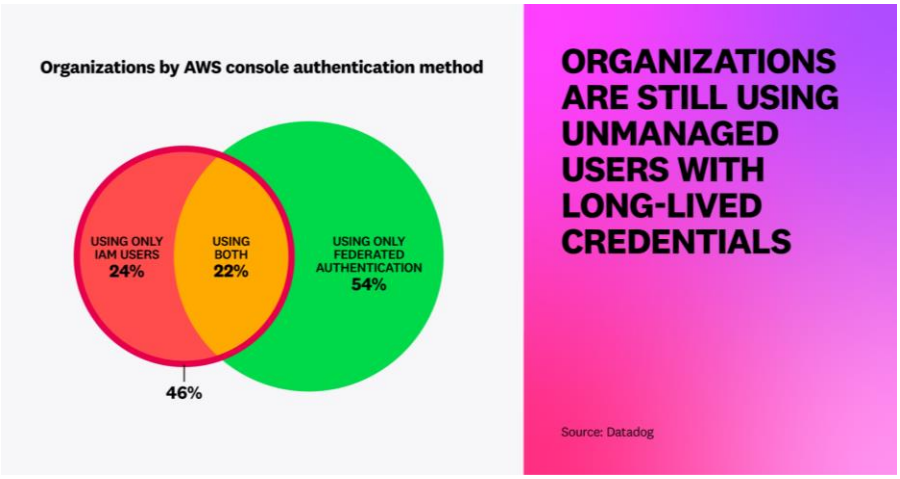
Attackers have continued to adapt their techniques to an increasingly cloud-native landscape, with new threats targeting the different cloud platforms constantly emerging. As threats evolve, understanding the major sources of risk and patterns of attacker behavior in cloud environments is critical.

For the 2024 edition of our State of Cloud Security study, we analyzed security posture data from a sample of thousands of organizations that use AWS, Azure, or Google Cloud. Our findings suggest that adoption of secure configurations in cloud environments continues to improve, thanks to greater awareness and better enforcement of secure defaults. Still, risky or overly privileged credentials remain a major entry point for attackers. This risk can be heightened by common misconfigurations across elements of cloud infrastructure, including compute and storage instances, managed Kubernetes distributions, and third-party integrations with SaaS providers.

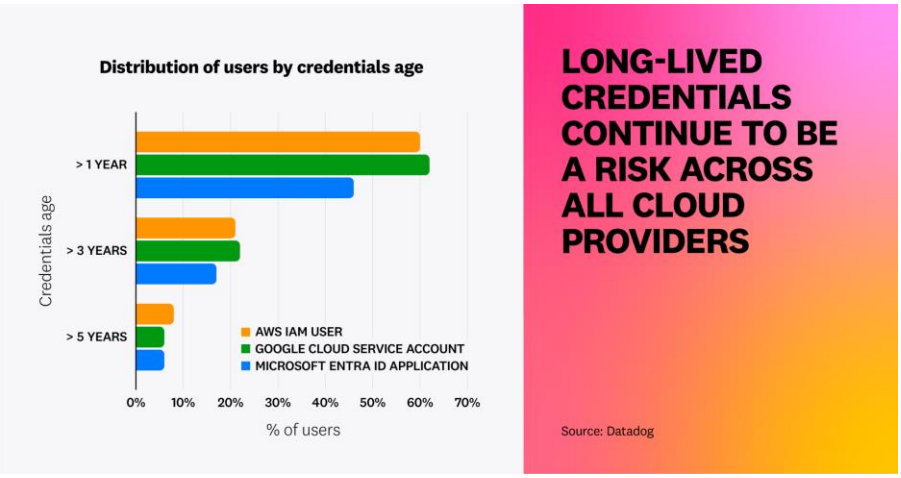
Fact 1: Long-lived cloud credentials continue to be a major risk

Long-lived cloud credentials pose a major security risk, as they never expire and frequently get leaked in source code, container images, build logs, and application artifacts. Past research has shown that they are the most common cause of publicly documented cloud security breaches. For this report, we analyzed how organizations leverage legacy versus modern authentication methods to authenticate humans and applications.

For humans, we found that most organizations leverage some form of federated authentication—i.e., using a centralized identity to grant users access to multiple systems—to access the AWS console (for instance, through AWS IAM Identity Center or Okta). However, almost half (46 percent) also use IAM users (a form of long-lived credential), and one in four *only* use IAM users. This shows that, although organizations increasingly use centralized identity management, unmanaged users with long-lived credentials remain popular.



We also determined that, in addition to being widespread, long-lived cloud credentials are often old and even unused. In AWS, 60 percent of IAM users have an active access key older than one year. Over half of these users have credentials that have been unused for over 90 days, hinting at risky credentials that could trivially be removed. Google Cloud service accounts and Microsoft Entra ID applications follow a similar pattern, with 62 percent and 46 percent, respectively, having active long-lived credentials older than one year.

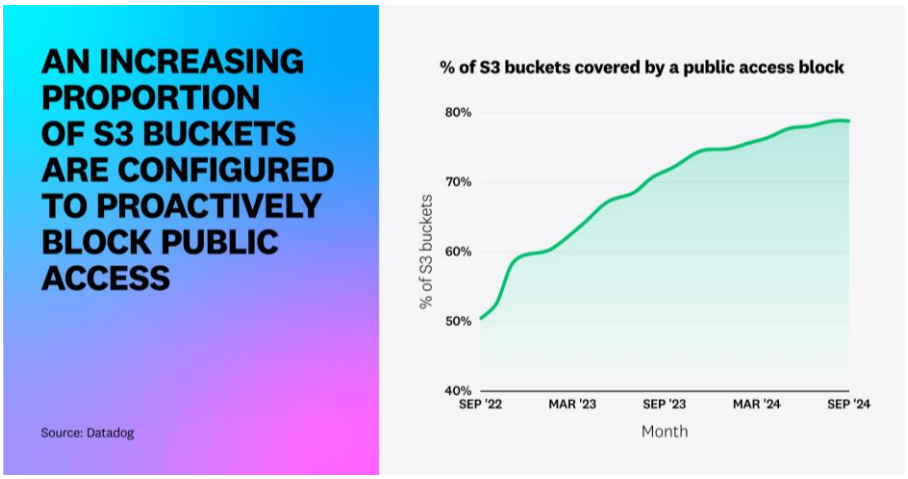


This data demonstrates that it's unrealistic to expect that long-lived credentials can be securely managed. Instead, organizations should leverage mechanisms that provide time-bound, temporary credentials. For workloads, this can be achieved with IAM roles for EC2 instances or EKS Pod Identity in AWS, Managed Identities in Azure, and service accounts attached to workloads for Google Cloud. For humans, the most effective solution is to centralize identity management using a solution like AWS IAM Identity Center, Okta, or Microsoft Entra ID, and to avoid the use of individual cloud users for each employee, which is highly inefficient and risky.

Fact 2: Adoption of public access blocks in cloud storage services is rapidly increasing

Modern security practices embrace "paved roads" that make it easy to do the right thing, and "guardrails" to prevent failures and ensure that human mistakes don't turn into data breaches. Public storage buckets have historically been responsible for a large number of high-profile data breaches. Today, there are modern controls—which fall into the category of guardrails—to ensure a bucket cannot be made public by mistake.

First, we identified that 1.48 percent of AWS S3 buckets are effectively public, similar to the 1.5 percent figure from 2023. Meanwhile, 79 percent of S3 buckets are covered by an account-wide or bucket-specific S3 Public Access Block, up from 73 percent a year ago. This increase is likely caused by wider awareness of the issue, and the fact that AWS proactively blocks public access for newly created S3 buckets as of April 2023.



On the Azure side, 2.6 percent of Azure blob storage containers are effectively public, down from 5 percent a year ago. Relatedly, over two in five (42 percent) Azure blob storage containers are in a storage account that proactively blocks public access. This is likely due to Microsoft proactively blocking public access on storage accounts that were created after November 2023, making them secure by default.

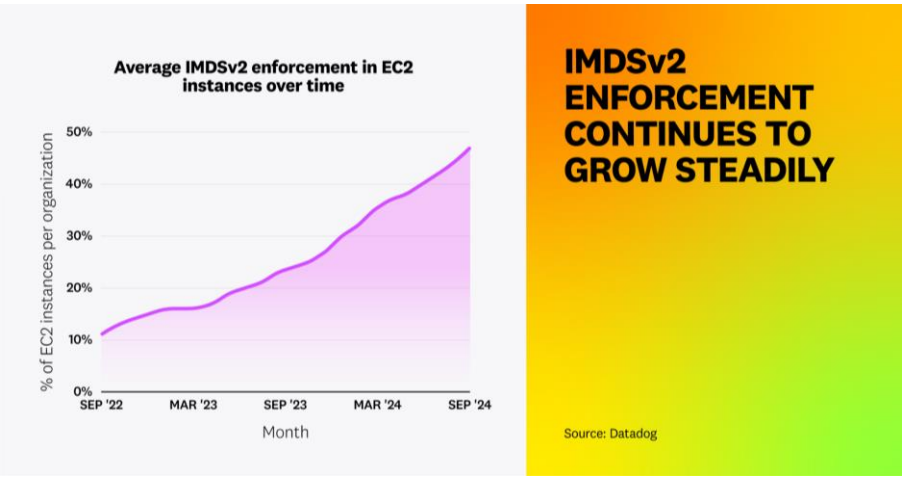
We can see that, when they exist, secure-by-default mechanisms from cloud providers are extremely powerful to remediate a whole class of vulnerabilities or misconfigurations.

To ensure that you don't mistakenly expose S3 buckets, it's recommended to turn on S3 Public Access Block at the account level and protect the configuration with a service control policy (SCP). In Azure, blocking public access in a storage account configuration allows you to make sure that no blob storage container in this storage account can inadvertently become public. Although there are legitimate use cases for public storage buckets, common uses such as static web assets should typically use a content delivery network (CDN) such as Amazon CloudFront, as this is usually a more efficient and cheaper solution.

Fact 3: Less than half of EC2 instances enforce IMDSv2, but adoption is growing fast

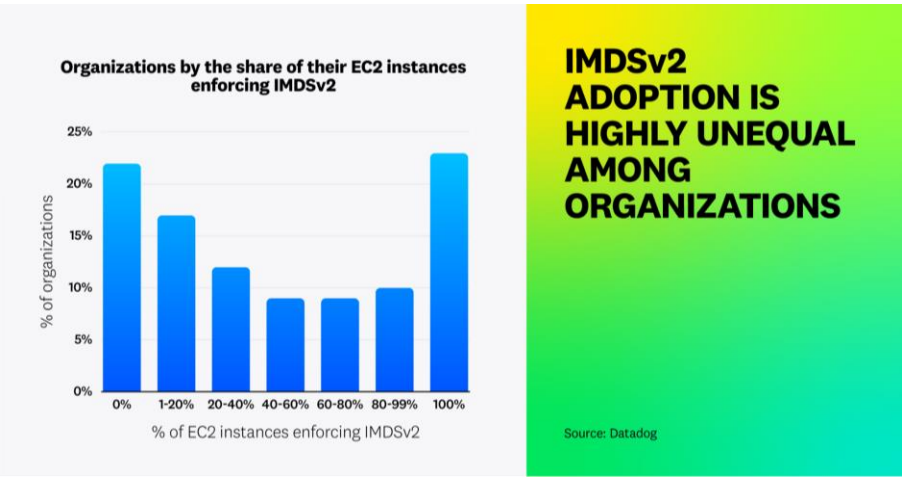
IMDSv2 is a critical AWS security mechanism to block credential theft in EC2 instances, which has led to a number of high-profile data breaches. While Azure and Google Cloud enforce analogous mechanisms to IMDSv2 by default, IMDSv2 historically had to be manually enforced on individual EC2 instances.

We're glad to report that IMDSv2 adoption is growing. On average, organizations enforce IMDSv2 on 47 percent of their EC2 instances, up from 25 percent a year ago. Overall, 32 percent of all EC2 instances have IMDSv2 enforced. However, instances created recently are by far more securely configured: 42 percent of EC2 running instances launched in the two weeks preceding our data collection period for this report had IMDSv2 enforced, versus only 10 percent for those created more than a year ago.

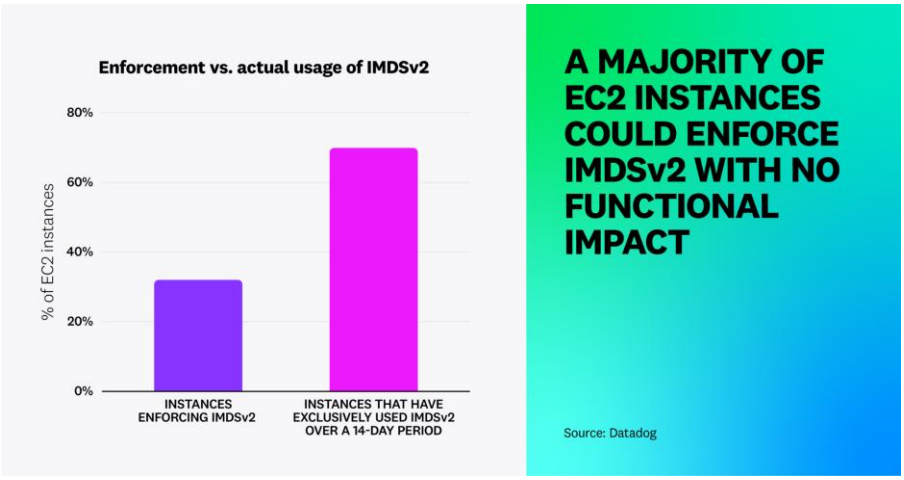


This positive increase is caused by a few factors. First, organizations are more widely aware of this issue today. In addition, in late 2022 AWS released a mechanism to enforce IMDSv2 by default on a per-AMI basis and turned it on for the now-popular Amazon Linux 2023 distribution. Then, in March 2024, AWS released a new region-wide setting that allows users to enforce IMDSv2 by default for all future instances created in a given region.

That said, we also identified that IMDSv2 adoption is highly unequal between organizations: 22 percent don't enforce IMDSv2 on *any* of their EC2 instances, while 23 percent enforce it on all of their instances. Overall, fewer than one in three organizations (33 percent) enforce IMDSv2 on more than 80 percent of their instances.



This data shows that while IMDSv2 enforcement has sharply increased, most instances still remain vulnerable. However, even when IMDSv2 is not enforced on an EC2 instance, individual applications and processes on that instance can use it. By looking at CloudTrail logs, we identified that although only 32 percent of EC2 instances have IMDSv2 enforced, 70 percent had exclusively used IMDSv2 in the past two weeks, meaning they could have enforced it with no functional impact. This shows a disconnect between enforcement and actual usage.



Region-level controls, such as enforcing IMDSv2 for new instances by default, are helpful for automatically implementing secure configurations, but they are insufficient, especially for long-lived EC2 instances that don't get recreated. Organizations should enforce IMDSv2 on all of their instances, use secure-by-default mechanisms such as AMI's "enforce IMDSv2" flag, and enforce IMDSv2 in all future instances within a region. It's also advisable to take extra care to monitor usage of credentials retrieved through the insecure IMDSv1 service. See also AWS's guide on migrating to IMDSv2 and Slack's journey.

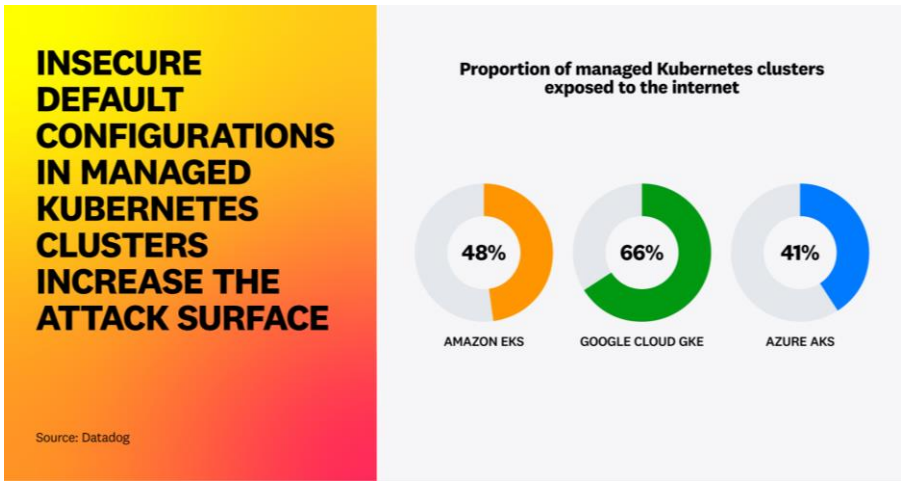
"Seeing improvement in IMDSv2 enforcement shows that the most impactful change continues to be AWS implementing more secure defaults within its AMIs and compute services. Many companies are also recognizing the critical importance of enforcing IMDSv2 usage on the internet edges of their environments. However, it's likely we'll soon reach a plateau in IMDSv2 enforcement. In 2024 and onward, companies will increasingly adopt data perimeter guardrails, which can help mitigate the impact of credential theft when IMDSv2 is not enforced."

Houston Hopkins
Cloud Security Veteran and fwd:cloudsec organizer

Fact 4: Securing managed Kubernetes distributions requires non-default, cloud-specific tuning

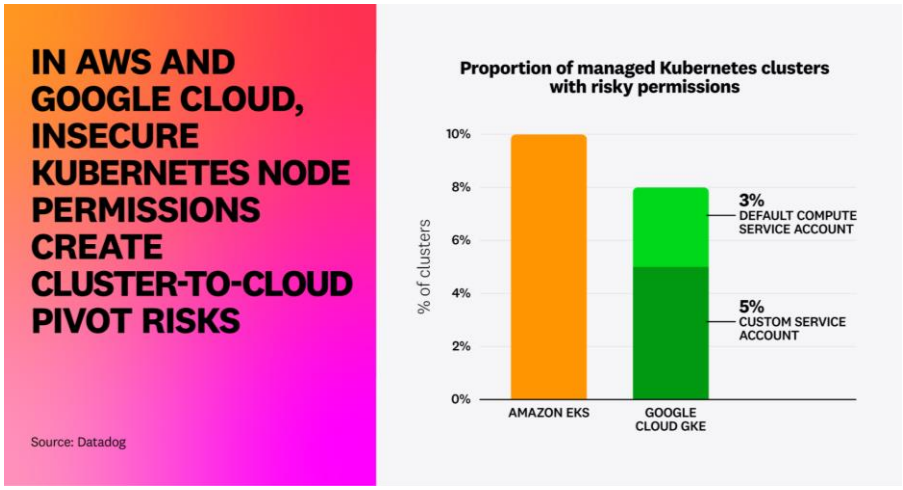
Managed Kubernetes services, such as EKS or GKE, are highly popular in cloud environments, as they allow teams to focus on running application workloads instead of managing complex Kubernetes control plane components such as [etcd](https://www.datadoghq.com/blog/etcd-key-metrics/). Although popular, the default configuration of these clusters often lacks security. This can be problematic, as these clusters are intrinsically running in cloud environments; compromising a managed cluster opens up a number of possibilities for an attacker to pivot to the underlying cloud infrastructure.

First, we've identified that a large number of managed Kubernetes clusters expose their managed API server to the internet. This accounts for almost half of Amazon EKS clusters, 41 percent of Azure AKS clusters, and two in three Google Cloud GKE clusters.



In addition, over one in four EKS clusters don't have audit logs enabled. This makes it impossible to understand activities performed inside the cluster, such as creating new pods or accessing secrets. Low adoption can be explained by the fact that these audit logs are not enabled by default and need to be explicitly turned on.

Finally—and even more importantly—we analyzed the IAM roles attached to EKS worker nodes and determined that 10 percent of clusters have a dangerous node role that has full administrator access, allows for privilege escalation, has overly permissive data access (e.g., all S3 buckets), or allows for lateral movement across all workloads in the account. In Google Cloud, 8 percent of GKE clusters have a privileged service account—3 percent through the use of the default compute service account with the unrestricted `cloud-platform` scope, and 5 percent through customer-managed service accounts.



In both cases, this is problematic—by default, a single compromised pod could steal worker node credentials from the instance metadata service and impersonate it against the AWS or Google Cloud API.

Organizations should not expose Kubernetes clusters to the internet. Instead, place them on a private network, or use modern zero-trust mechanisms such as Google Cloud's Identity-Aware Proxy (IAP), which is fully compatible with GKE. Enabling Kubernetes audit logs—at least for the API server—is also essential for cluster security. Finally, take extra care when assigning cloud permissions to worker nodes, which often run dozens of applications, and make sure that individual applications cannot access cloud credentials from their worker node.

While managed Kubernetes distributions are a great way of reducing complexity, they're also a good illustration that cloud resources that don't have security mechanisms turned on or enforced by default often end up with suboptimal security.

"Using Managed Kubernetes services remains a key recommendation, as allowing cloud providers to manage the Kubernetes control plane reduces attack surface. However, organizations still need to secure their clusters with key controls, such as restricting access to the cluster endpoint and configuring audit logs. Managed services make it easier to adopt Kubernetes, but they also hide the complexity of securing these environments. The low adoption rates of these essential security controls raises concern about deeper risks in these Kubernetes clusters. The increase in cluster-to-cloud attacks by threat actors like TeamTNT and SCARLETEEL underscore the urgent need for stronger cluster security."

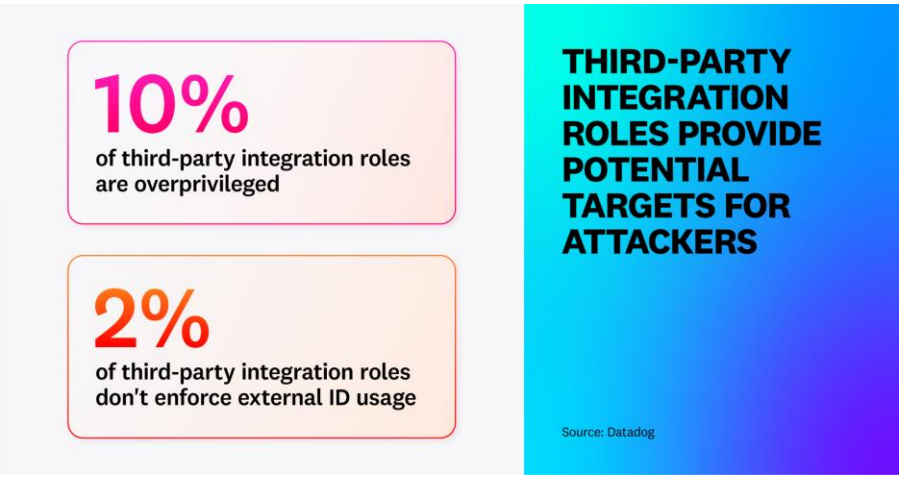
Rami McCarthy
Staff Cloud Security Engineer

Fact 5: Insecure IAM roles for third-party integrations leave AWS accounts at risk of exposure

As public cloud usage is growing, more and more vendors have started to integrate with their customers' AWS accounts—for instance, to monitor their cloud infrastructure or collect logs. In this situation, customers typically delegate access to their cloud environment through an IAM role that trusts the provider's verified AWS account. By design, this creates a cloud supply chain risk: If the provider's AWS account is compromised, it's likely that an attacker can access the same data as the provider can.

For this fact, we reviewed IAM roles trusting known AWS accounts that correspond to SaaS providers. We found that on average, an organization deploys 10.2 third-party integration roles (median 3), linked to on average 2.4 distinct vendors (median 2).

We then looked at two types of common weaknesses in these integration roles. We found that 10 percent of third-party integrations are dangerously overprivileged, allowing the vendor to access all data in the account or to take over the whole AWS account. We also identified that 2 percent of third-party integration roles don't enforce the use of External IDs; this allows an attacker to compromise them through a "confused deputy" attack.



This shows that protecting against third-party risk is just as essential in cloud environments as it is in software development. It's recommended to build an inventory of the third-party integrations you trust in your cloud account and make sure that you remove roles when you stop using them as a vendor. Then, it's critical to grant minimum permissions to your cloud environment. As an example, a vendor monitoring the status of your EC2 instances should not be able to read data from your S3 buckets. Finally, make sure to follow vendor instructions on enforcing External IDs.

Fact 6: Most cloud incidents are caused by compromised cloud credentials

As part of our security research activities, we proactively look for attacker activity in our worldwide telemetry, such as AWS CloudTrail logs, Azure activity, and Entra ID activity logs.

This allows us to uncover attack patterns and understand what threat actors are up to across a large set of organizations.

First, we found that **most cloud incidents are caused by attackers compromising identities**, both from humans (passwords) and applications.

In AWS, leaked access keys are a highly popular and efficient initial access vector. Attackers frequently leverage credentials scanners such as Trufflehog to identify them in source code repositories and version control system (VCS) history. Once they gain an initial foodhold, attacker behavior often follows a number of consistent patterns that we've identified:

- **Pivoting from access keys to console access**: Using the GetFederationToken API call, it's possible to generate an AWS console sign-in link. This is useful for attackers who don't want to use the AWS CLI and want to persist in the environment they've breached even if the compromised access key is disabled.

- **Enumerating core services**: This allows an attacker to quickly gain situational awareness and understand the value of the account they've just compromised. In particular, we've witnessed attackers performing in-depth enumeration of Amazon SES to understand email sending limits in the account, then using this access as a way to send phishing or spam emails, either directly or by reselling access on the secondary market.

- **Reselling access on the secondary market**: When an attacker compromises an AWS account, their next step is to determine how to make money from it. One way is to sell their access, either to the compromised account or to specific services. We've witnessed reselling access to the whole account, to Amazon SES only (for spam/phishing sending purposes), or to Amazon Bedrock.

- **Crypto mining:** In some cases, we've seen attackers run crypto miners directly in compromised workloads. In other cases, we've witnessed attackers creating EC2 instances in unused AWS regions, or even creating a new Elastic Container Service (ECS) cluster exclusively for crypto mining purposes. When the account was too restricted, we've also seen attackers open AWS support cases to increase quotas and even attempt to upgrade to a Business support plan.

We've observed similar patterns of behavior across other cloud platforms as well. In Microsoft 365:

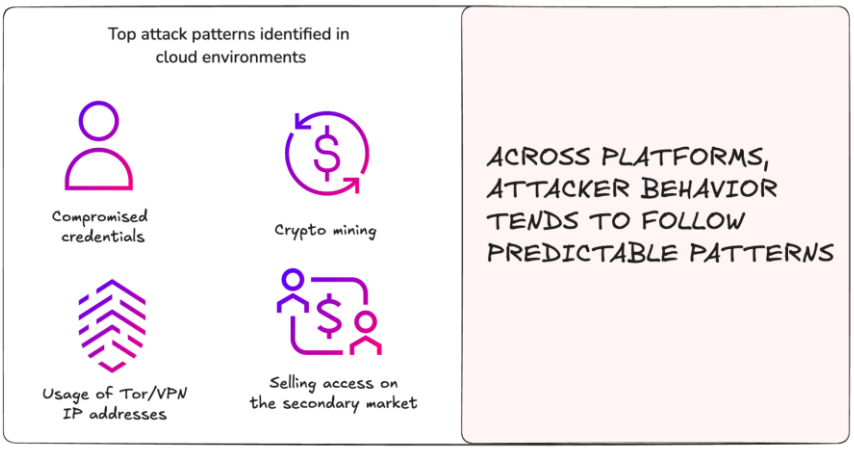
- **Initial access** is most often achieved through credential stuffing, but also malicious OAuth applications and adversary-in-the-middle (AitM) attacks. The two latter techniques allow an adversary to bypass all or some forms of multi-factor authentication (MFA).

- **Malicious OAuth applications** are not only used for initial access but also for email exfiltration. After compromising an identity, we've witnessed a number of attackers granting email access to a malicious application such as `eM Client` and `PERFECTDATA`, sometimes also actively disabling tenant-level settings that restrict usage of third-party OAuth applications.

- **Inbox rules** are frequently used to hide emails from the victim—for instance, so that they don't see an email from IT support about activity on their account.

- We've also seen a number of cases where an attacker attempted to **spread laterally** by sharing malicious attachments or OneNote notebooks company-wide, and sending internal phishing emails. These attacks can also spread externally to customers and clients of the company.

While our telemetry for Google Cloud is more recent, we have identified several attackers that could easily be caught through their usage of VPN networks, residential proxy providers, or Tor exit nodes. Google Cloud's threat intelligence confirms that attackers also launch crypto miners there.



This data shows that it's critical to secure identities with strong authentication mechanisms such as FIDO, limit long-lived credentials, and actively monitor changes to APIs that attackers commonly use.

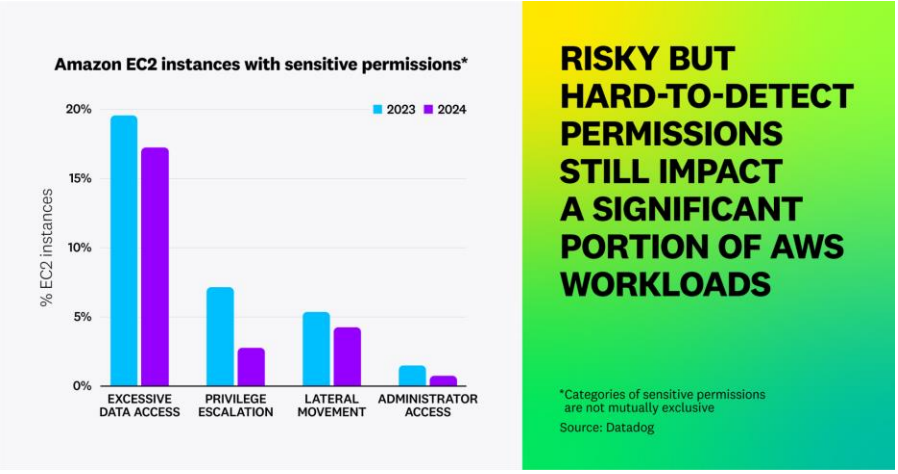
Fact 7: Many cloud workloads are excessively privileged or running in inappropriate places

Workloads running in cloud environments commonly need to access cloud resources. For this purpose, the recommended approach is to leverage mechanisms that assign a platform identity to the workload, such as IAM roles for EC2 instances or service accounts for Google Cloud virtual machines (VMs). However, assigning overprivileged permissions to such workloads can create substantial risk. Any attacker who compromises the workload—for instance, by exploiting an application-level vulnerability—would be able to steal the associated credentials and access the cloud environment.

In AWS, we found that while less than 1 percent of EC2 instances have administrator privileges, over 18 percent are overprivileged. Among these:

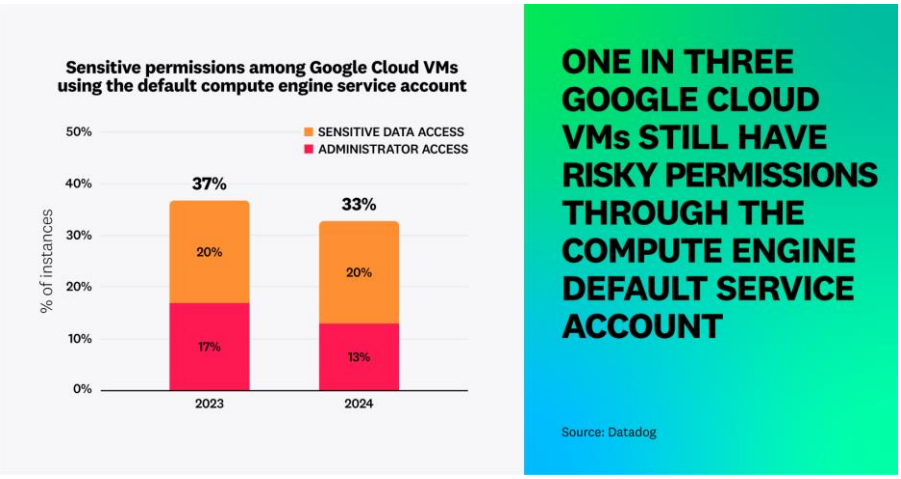
- 4.3 percent have risky permissions that allow lateral movement in the account, such as connecting to other instances using SSM Sessions Manager
- 2.8 percent allow an attacker to gain full administrative access to the account by privilege escalation, such as permissions to create a new IAM user with administrator privileges
- 17.6 percent have excessive data access, such as listing and accessing data from all S3 buckets in the account.

While these numbers remain high, we noticed a decrease in the prevalence of these insecure configurations compared to a year ago.



We also found that at least 6 percent of organizations run EC2 instances in their AWS Organization management account. This is considered a highly risky practice, as this AWS account can by design access any child account in the organization. We believe this finding is caused by organizations turning on AWS Organizations in their current production environment and running infrastructure in the management account, effectively turning it into a risky pivot point. This is why AWS recommends that users avoid deploying workloads in this account when using AWS Organizations, an important best practice to adopt when using this popular tool.

In Google Cloud, we found that 13 percent of VMs have privileged “editor” permissions on the project they run in, through the use of the default compute service account with the unrestricted `cloud-platform` scope. In addition, another 20 percent have full read access to Google Cloud Storage (GCS) through the same mechanism—so in total, over one in three Google Cloud VMs (33 percent) have sensitive permissions to a project. Although this is a small decrease from 2023 (down from 37 percent), these default permissions remain a widespread, systematic issue that requires additional awareness.



Organizations using Google Cloud should enable the “Disable automatic IAM grants for default service accounts” organization policy and ensure that VMs use a non-default service account.

Managing IAM permissions for cloud workloads is not an easy task. Administrator access is not the only risk to monitor—organizations should also be wary of sensitive permissions that allow a user to access sensitive data or escalate privileges. Because cloud workloads are a common entry point for attackers, it’s critical to ensure permissions on these resources are as limited as possible.

Methodology

Findings are based on data collected in September 2024.

Population

For this report, we analyzed the cloud security posture of a sample of thousands of organizations. Data in this report has come from customers of both Datadog Infrastructure Monitoring, Datadog Logs, and Datadog Cloud Security Management (CSM).