Google Cloud

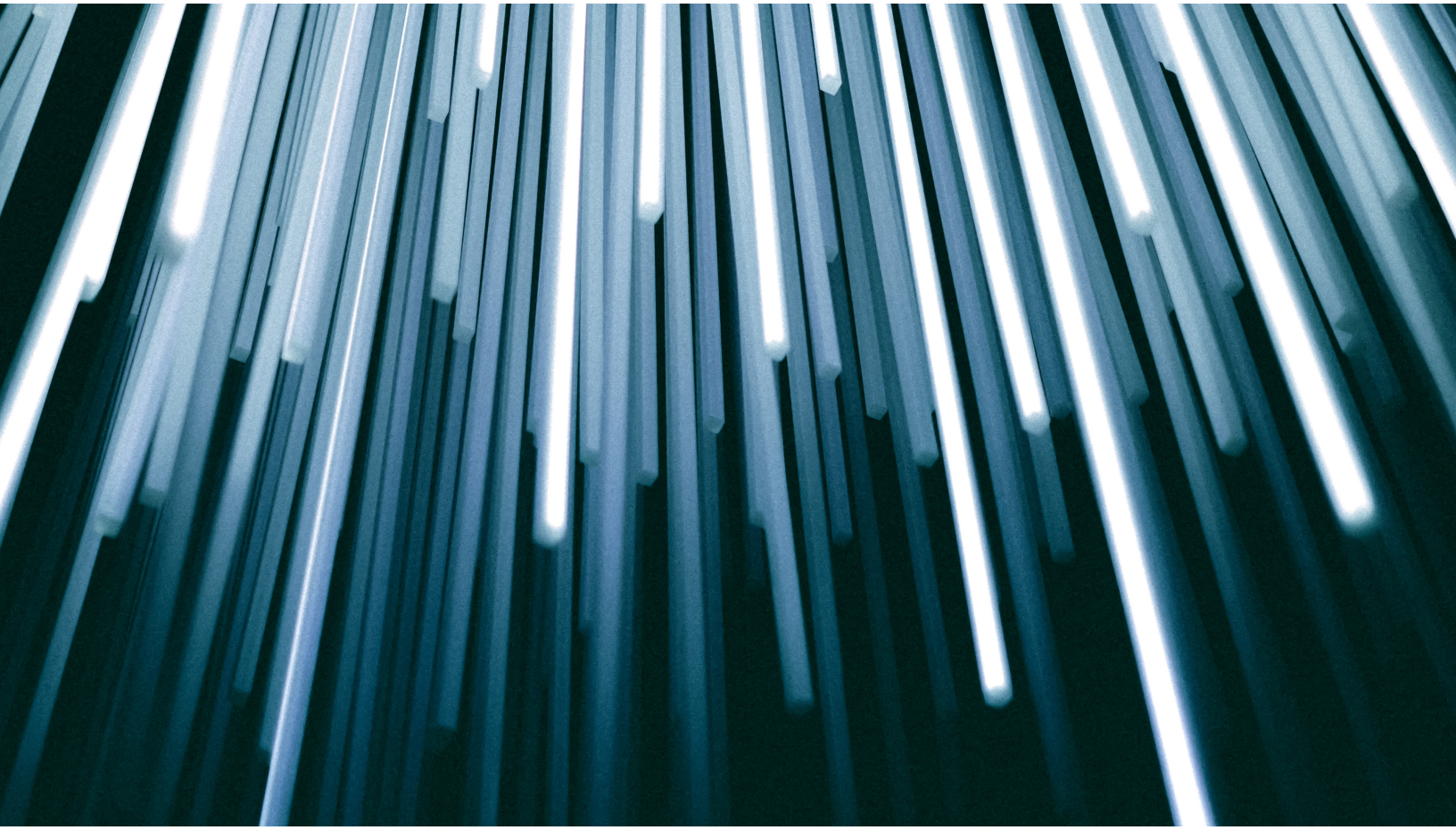
Threat Horizons

April 2023 Threat Horizons Report

Office of the CISO

April 2023

Table of contents



Mission statement

The Google Cloud Threat Horizons Report brings decision-makers strategic intelligence on threats to cloud enterprise users and the best original cloud-relevant research and security recommendations from throughout Google's intelligence and security teams.

Letter from the Editor

Strategic Perspective—Government-Backed Hackers Likely to Look to Criminals for Inspiration Targeting Cloud

When one thinks about the proliferation of military or espionage technology, the typical image is of "spin off," where once-futuristic fighter jets and spy gadgets gradually become available for public use and commercialized. Sometimes there's an intermediary step, where non-state groups including organized crime are early adopters and have a mismatch with the defenses of their targets stuck in a previous generation of civil conflict. The spread of the AK47 as a rifle meant to spread Communist revolution is one example, which was soon adopted by a variety of state- and non-state organizations. More recently we can see the adoption of unmanned aerial systems, or drones, that not too long ago were the sole province of the world's most advanced military and intelligence organizations, but now deliver drugs to prisoners or groceries to your home.

The process works in the other direction, too. Japanese defense industries are perhaps the best example of this over recent decades, taking in aircraft and other systems shared by their American allies and enhancing them beyond the capabilities of similar models in the U.S. by adding radar systems or advanced electronics developed by Japanese industry. This can be an effective mechanism to cut development costs or speed rapid prototyping (as with the famed F-117 Nighthawk, the first truly stealth plane) or to tap into the advanced software capabilities of Silicon Valley through the Defense Innovation Unit.

The use of technology in military conflict and intelligence competition means state-sponsored cyber threat actors seek options from which to build ever-growing replacements for their cyber armories, including adapting tools from the cybercriminal underground, ingesting published vulnerability research, or augmenting exploits used in the wild to supplement the more bespoke tools they develop in-house. Noting the undeniable value of cloud services, for improving enterprise defenses, bolstering cybersecurity capabilities, and improving resilience to a host of cyber threats (and the resulting volume of enterprises leveraging cloud capabilities), it is only prudent to consider that state-sponsored cyber threat actors may steal from the playbooks of cyber criminals to target such systems:

(Letter from the editor, cont'd.)

- **"Trusted" source of malware:** Broadly speaking, cyber threat actors have to defeat two kinds of systems when planning an operation that targets your organization: the people making singular decisions about whether and how to interact with the attackers, and the technical systems which automate decision-making across the network at large scale. Cybercriminals and nation-states alike have long known the value of compromising legitimate websites for use in continuing operations against the suppliers, partners, and competitors that are their ultimate target. A familiar domain name disarms many of the natural defenses we all have when viewing a suspicious email, and the degree to which it is trusted will often be hard-coded into security systems screening for spam or malware

  Cloud providers are useful targets for these kind of operations, either as hosts for malware or providing the infrastructure for command-and-control (C2). Indeed, an independent study by Netskope indicates that malware delivery via cloud services has steadily increased to make up nearly half of all malware downloads they observed in 2022, probably reflecting both the trusted state of those domains by other organizations and the sheer scale of operations originating from them overall.

  Nation-states are paying attention: in May 2022, Google's Threat Analysis Group disrupted an operation by APT29, previously attributed by the US and UK Governments to the Russian Foreign Intelligence Service (SVR), that was controlling compromised websites by means of C2 on Google Drive to deliver ISO files containing a malicious DLL. Google disabled the relevant accounts, and developed signatures to identify similar tooling, protect users, and prevent the group–and future imitators–from abusing Google's services. It's the kind of operation we expect to see attempted more often in the future, including by nation-states keen to bypass traditional security controls.

- **The Importance of Identity:** Our research has shown that the most common vector used to compromise any network, including cloud instances is to take over an account's credentials directly: either because there is no password, as with some default configurations, or because a credential has been leaked or recycled or is generally so weak as to be guessable. This is especially problematic for accounts with unnecessarily powerful IAM permissions, which can be used to move laterally, start new instances for cryptomining and other malicious activities, or to create new Service Accounts for maintaining persistence.

  This is overwhelmingly a cybercriminal phenomenon today. However,for cyber operations sponsored by nation-states for espionage purposes where persistence is paramount, the utility of taking over legitimate accounts or starting new Service Accounts for persistence in a compromised environment will be prioritized. Additionally, appearing to be an American or other Western citizen can be helpful for complicating attribution of follow-on operations, or limiting the collection of Western intelligence agencies.

(Letter from the editor, cont'd.)

Nation-state cyber threat groups will not only emphasize persistence in their operations, but can tap an array of tools–human intelligence, analysis, signals collection, diplomatic intelligence–to go after their targets over a long period of time. Finding a recycled password from a compromised commercial service being used by an account administrator in the cloud becomes more feasible with large, sustained operational investments.

- **The Convenience of Being a Non-state Actor:** Many of today's most troubling cyber threats originated as criminal- or hacktivist techniques that were later turned to geopolitical advantage: hack-and-leaks of private emails by public officials, doxxing of political opponents' records held by commercial companies, targeted deployment of ransomware, swarming social media propaganda, pastebins, even the recruitment of insiders via forum postings all originated or were popularized by non-state actors, typically with ideologically vice profit-seeking motivations.

  Eventually this evolved from the use of non-state hacking groups by the state, as in the case of the Syrian Electronic Army's targeting of the West that landed members on the FBI's Most Wanted List, into the use of entirely false hacktivist groups as a front for nation-state activity, as when the Russian military posed as the Cyber Caliphate to disable operations at a French TV station. North Korea and Iran have similar histories of using non-state hacking groups as cover for their intelligence operations or pressing real hackers into state service. Russia and, increasingly, China have each cultivated ecosystems that allow cybercrime to fester so long as it is focused on targets outside their own countries.

As the next article in our Report points out, because of the security of the GCP platform most compromises in the cloud are simply from lack of passwords, poor password strength, reused and leaked credentials, or straightforwardly misconfigured software: all techniques within the grasp of even the most unsophisticated actors. An actual state-sponsored APT group, however, would have greater persistence and means of maintaining and quietly expanding access over long periods of time, possibly coupled with more resilient command-and-control infrastructure compared to most non-state groups.

Posing as a non-state actor in those circumstances would be a viable cover, complicating attribution or at least deflecting geopolitical blowback to the apparent proxy instead of themselves, while permitting states with a history of working with criminal groups to perhaps simply purchase the access they need for their operations. Cloud-based enterprises can reduce this risk with the same basic security blocking-and-tackling they use to counter less targeted, criminal threats every day, ideally coupled with cyber threat intelligence to help with speedy and authoritative attribution in the event they are targeted this way.

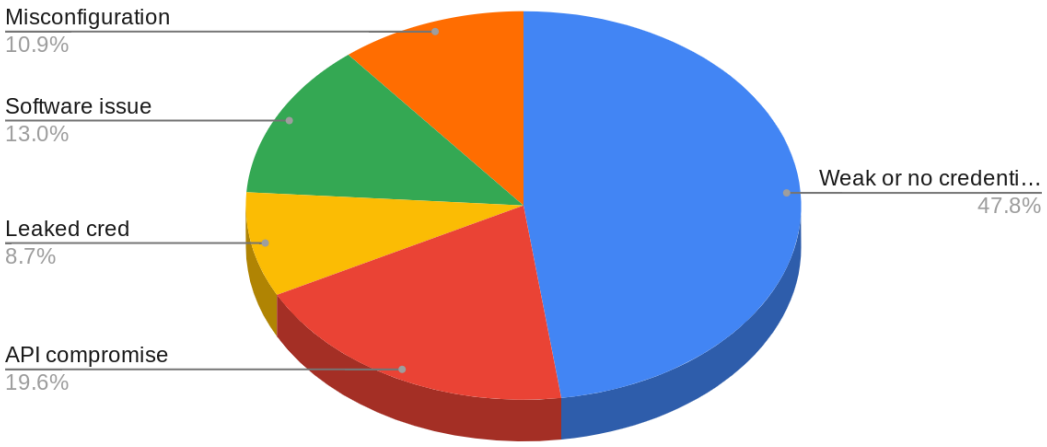
Christopher Porter is the Head of Threat Intelligence for Google Cloud.

Credentials, API Issues Continue to Lead Compromise Factors

Compromise Factors Remain On Track in Q4

Weak passwords accounted for nearly half of observed incidents in the fourth quarter of 2022, a continuation of the activity pattern tracked in the prior three month period. In addition, the rise in API compromise in Q3 maintained course, being a factor in nearly 1/5th of incidents. A slight decrease in misconfiguration and software issues provided some positive news for the quarter, with the lost share taken up in part by an increase in the use of leaked credentials. The continued focus by threat actors on leveraging credential-based access remains a key concern point in 2023 (see article 7 of this report for a deeper dive) and should be of top priority for organizations to address.

Cloud Compromise Factors (Q4 2022)



(Credentials, cont'd.)

Case Study: Ransomware in the Cloud

Threat actors often use ransomware in the cloud to extort companies in a different manner than traditional on-premises environments, threatening to release or delete data rather than simply encrypt it. This variation on ransomware approach ties directly to the benefit cloud usage provides in prompt reconstitution of encrypted resources. In September 2022, an attacker compromised a small number of insecurely configured Google Cloud SQL database instances, corrupted the data, and left behind a ransom note. The compromise was due to poor hygiene and lack of basic control implementation, not a specific bug in GCP or Cloud SQL. The attacker left a ransom note demanding payment in Bitcoin and threatening that if the ransom was not paid, they would leak the stolen data to darkweb marketplaces. The note included a specific Tor .onion address and instructions for payment.

```
### UPDATE `Z_README_TO_
RECOVER`.`RECOVER_YOUR_DATA`

### @1='All your data is a backed
up. You must pay 0.2 BTC to [Bitcoin
address] 48 hours for recover it.
After 48 hours expiration we will
sell all your data on dark markets
and the database dump will be
dropped from our server!'
```

Ransom note left on insecurely configured Cloud SQL instances

Affected MySQL instances created a window of opportunity for the attacker due to misconfigurations (public IP addresses configured to allow inbound and outbound traffic from any IPv4 address and port) and likely weak passwords or no password at all. While ransomware impacts can be mitigated through cloud usage in different ways, threat actors continue to evolve their techniques to target this environment and it should remain a key concern for organizations.

HOODOO Uses Public Tooling, Google Workspace to Target Taiwanese Media

In October 2022, Google's Threat Analysis Group (TAG) disrupted a campaign from HOODOO, a Chinese government-backed attacker also known as APT41, that targeted a Taiwanese media organization by sending phishing emails that contained links to a password-protected file hosted in Drive. The payload was an open source red teaming tool called "Google Command and Control" (GC2). Written in Go, the tool gets commands from Google Sheets, likely to obfuscate the malicious activity, and exfiltrates data to Google Drive. After installation on the victim machine, the malware queries Google Sheets to obtain attacker commands. In addition to exfiltration via Drive, GC2 enables the attacker to download additional files from Drive onto the victim system. HOODOO previously used GC2 in July 2022 to target an Italian job search website.

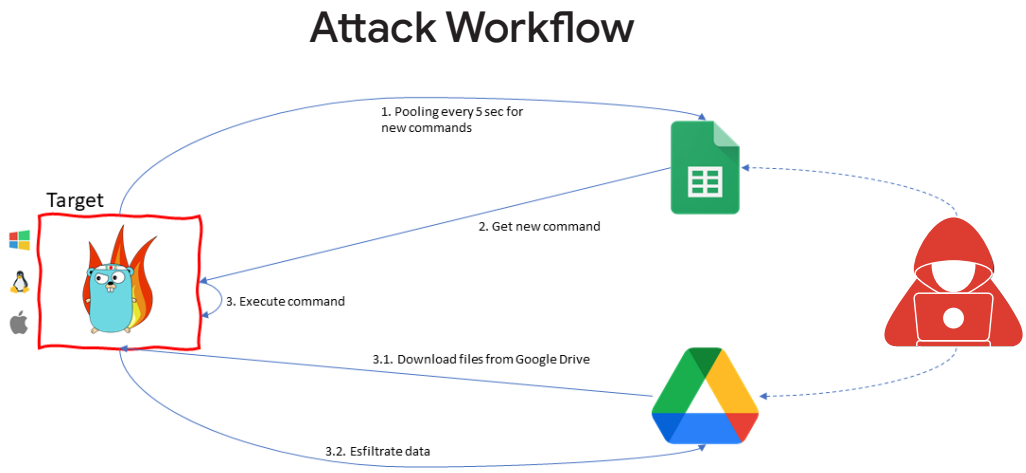


Figure 1. Publicly Available GC2 Workflow Documentation (source: GitHub)

(HOODOO Uses Public Tooling, cont'd.)

These incidents highlighted a few key threat trends by China-affiliated threat actors. First, as opposed to developing their own custom tools, Chinese APT groups are increasingly using publicly available tooling such as Cobalt Strike and other "pentest" software available for purchase or on sites like Github. HOODOO's use of GC2 is an example of this trend. Second, the proliferation of tools written in the Go programming language has continued to expand, likely due to the flexibility of the Go language and its convenience for adding and removing module components. Finally, the targeting of Taiwanese media illustrates the continued overlap of public sector threat actors targeting private sector organizations with limited government ties.

Compromised Customer Websites Hosted on IT Service Providers' Infrastructure

Issue Description

As of February 2023, Google discovered 14 compromised customer websites hosted on Google Cloud having bi-directional communication to malware–which was originally identified using 2022 VirusTotal (VT) data. All the sites were set up through IT Service Providers (ITSP), who had resold GC services to the 14 clients. Each site hosted malicious files, had external malware communicating with it, and had its domain embedded into the source code of various external malware. We also confirmed the suspicious nature of these sites using independent security tools.

Google Cloud ITSP customers can prevent such types of environment misuse by:

- properly segregating clients, and limiting all-client access granted to individual admins, within multi-tenant environments;
- logging and monitoring for potential compromises on endpoint servers exposed to clients as well as on common ITSP infrastructure;
- and patching vulnerabilities found using vulnerability scans to prevent threats via common ITSP infrastructure.

To identify compromised sites, we looked for malware first submitted to VT in 2022 having actual Cloud Service Provider "interactions" rather than just benign "one way communications". We were not looking for malware that just checked whether it had Internet connectivity or was conducting port scans, but only activity with bidirectional communication we associate with potentially malicious behavior.

(Compromised Customer Websites, cont'd.)

We found 38 domains meeting our overall criteria, and among these, identified 14 customer sites that were live as of early February 2023. When we further explored potential patterns of compromise, we found that all 14 sites were related to ITSPs. The ITSPs purchased the Google Cloud services, and resold them—helping these 14 clients set up domains—and acquire Google Cloud IP addresses. We also confirmed these 14 sites' suspicious nature by searching for their domains and corresponding IPs in the independent AlienVault and Shodan security platforms.

Although we cannot comment on the root cause for these compromises, we recommend that ITSPs using Google Cloud should consider the following suggestions to mitigate compromises or hinder proliferation to other environments hosted by ITSPs.

Suggested Mitigations for Google Cloud ITSP Customers

1. ITSPs should ensure role based access, with segregation of duties, so that one internal company admin does not have access to all of an ITSP's clients. If such an admin were to be compromised, not all ITSP clients would be at risk. Such admins should also use a hardware second factor in the form of security keys, for authentication.
2. ITSPs should provide guidance to clients on how to securely set up websites, web applications, and domains in their resold assets. Clients should be recommended to follow best practices such as the OWASP Top 10 guidelines (using proper authentication protocols, like MFA, when needed; protecting against injection attacks, such as by validating user-supplied data for suspicious content; and analogous controls) to secure websites and web applications. They should also be reminded to be vigilant against employee phishing attacks, which can prevent domain administration portal credentials from being stolen–that could lead to domain misuse.
3. Where ITSPs employ multi-tenancy by hosting multiple customers in a single organization, design isolation boundaries between tenants. Consider segregating customers into isolated projects or folders, isolated VPC networks, and isolated VPC Service Control (VPCSC) perimeters. VPCSC could help to mitigate the threat of a compromise in one customer perimeter spreading through Google services to resources in a separate customer perimeter.

Cloud-Hosted Encrypted ZIP Files Evading Detection

Research by Google Cloud and Mandiant into threat actor use of Google Drive for malware hosting highlighted threat actors storing malware in Google Drive as encrypted ZIP files, likely in an effort to evade detection. For example, in Q4 2022 Mandiant observed a campaign distributing URSNIF malware, a well-established general intrusion software with history as a banking bot, by hosting the URSNIF binary on Google Drive. The threat actors used a phishing email to convince victims to download a password protected ZIP file hosting the malicious content which was then installed on their machine.

An expansion of this technique was seen later in Q4 2022 by threat actors using DICELOADER malware, another general intrusion malware which could be used for many different purposes. In this campaign, Mandiant identified phishing emails using a malicious Google Drive link that downloaded a ZIP file that contained a LNK file. The LNK file then subsequently downloaded and installed a Trojanized Zoom MSI installer, which led to the eventual DICELOADER infection. This campaign appeared to target the financial services sector based on the phishing emails found by Mandiant. By separating the malware binary from the downloaded ZIP file, the actors even further obfuscated their malicious intention from the Google Drive download. Google took multiple steps to halt this activity at the time, as well as implementing additional detective capabilities to spot and stop similar malicious use of Google Drive in the future.

These techniques highlight the risk posed by threat actors using cloud services to host malicious content and their continued development of evasion techniques to avoid detection, moving from encrypted ZIP files containing malware to encrypted ZIP files linking to trojanized legitimate installers. This continued evolution suggests that organizations need to carefully monitor downloads from even legitimate-seeming sites. While Google Cloud continues to implement a host of controls for all customers in order to detect and prevent this activity, targeted organizations should also consider using tools like Chronicle detection engine to identify and stop malware activity faster.

Customer Challenges and Solutions When Security Patching Google Kubernetes Engine

Cloud customers are increasingly running their workloads in Kubernetes clusters due to the availability, flexibility, and security they provide. But like other IT assets, these clusters need to be routinely patched, too, to keep their features current and to install security and bug fixes. Reviewing 2021-2022 data, Google has found that Google Kubernetes Engine (GKE) customers sometimes delay security patching their clusters, often from concern that patching might inadvertently interrupt production operations. Yet delaying security patching introduces its own challenges–as a number of vulnerabilities can be identified in unpatched GKE environments over time.

GKE customers can maintain workload availability and security patching currency by configuring and orchestrating Kubernetes environments to speed up patching while maintaining business continuity; specify appropriate maintenance windows and maintenance exclusions windows to align with business needs; and use several notification and scanning services to find vulnerabilities, and plan for security patch installations.

GKE customers are concerned with tradeoffs between Kubernetes cluster availability and security patching. Customer workloads need to run during production timeframes–especially during key periods such as Black Friday, New Year's Eve, etc. Security patching might impact this availability, as systems can be down during patching–and there might even be outages after patching. (For example, across any IT environment, patches could inadvertently even introduce new bugs). Across a number of conversations between GKE teams and GKE customers during 2021-22, customers provided the following main reasons why they prolonged or delayed security patching, as they balanced security and availability. (Solutions to these concerns are described at the bottom of this article).

- Customers of applications that need to maintain "state"—such as via sessions or similar mechanisms—worried that unexpected patching would break the states (for example by recycling a node-hosted load balancer or web server

(Customer Challenges and Solutions, cont'd.)

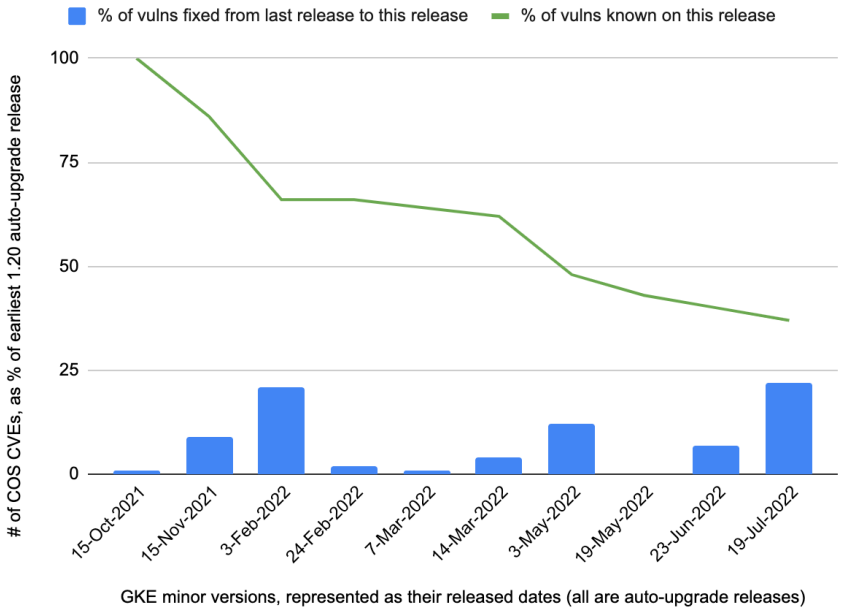
containing a pinned session), undermining their applications' execution. (This issue can be addressed by a Pod Disruption Budget and Termination Grace Policy, as explained at the bottom of the article).

- Similar to the "stateful" customers, were other clients executing batch and AI/ML applications. They worried that patching might interrupt work like ML "training", as not-yet-completed workloads could be restarted during patching. Although for such clients, non-interruption, rather than patching per se, was the aim. If the cluster nodes (i.e., the virtual machines making up Kubernetes clusters where customers run their workloads) can finish without interruption, they could be terminated without patching. These workloads were, in effect, 'one-time' jobs. (Patching concerns for such workloads can be mitigated via maintenance exclusion configurations, as below).
- Customers delayed security upgrades because they worried that such upgrades might also bring unanticipated API changes, which might undermine their applications' functionality. Yet, APIs do not change when security upgrades occurs within a minor version (the version of the Kubernetes cluster's overall operating environment); and updates can be configured to only upgrade the current minor version, and not upgrade to a newer minor version (i.e., the scope of the updates can be controlled). Customers were not always aware of this configuration option, however.
- For customers with very large node fleets, patching could take more time, and possibly maintain a weak security posture longer. The default GKE node patching method is surge upgrade, with a default node-update parameter, maxSurge, of 1. Customers can modify this parameter to make patching go faster, but the default value creates a longer patching process, as only one node can be updated at a time. The environment will, consequently, run with vulnerable nodes longer.

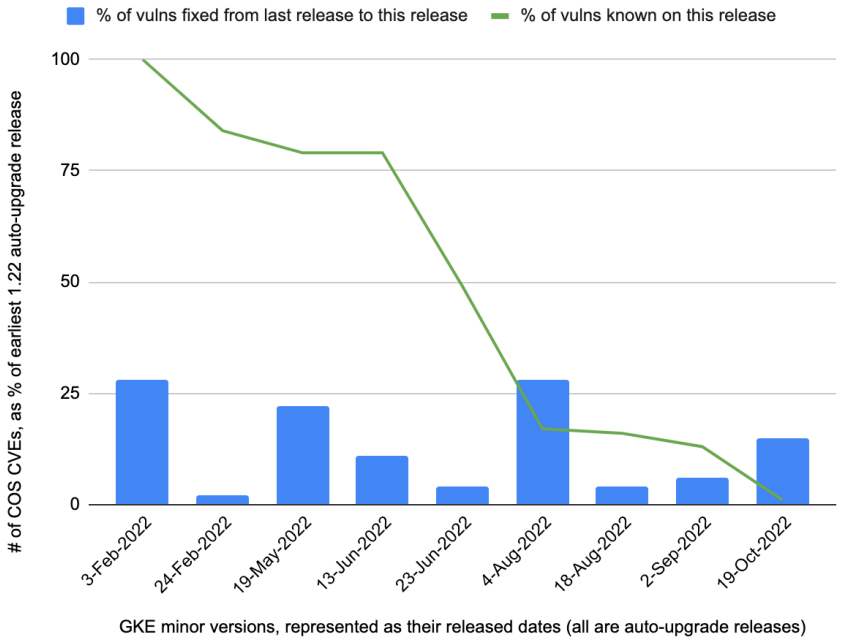
Prolonging or delaying security patching has its own challenges, however. Clusters running on older GKE patch releases had 2X, 3X, or more open CVEs compared to those on newer patch releases. Effectively, unmanaged risk accumulates over time. We looked at a year's worth of node patching data, ending in October 2022. Customers were running several GKE minor versions and various Container-Optimized Operating Systems (COS, the actual OS image deployed within the minor version) in this timeframe. The relative number of High and Moderate COS CVEs in one of the earlier GKE minor versions, 1.20, as well as one of the later minor versions, 1.22, in this timeframe, is illustrated below.

(Customer Challenges and Solutions, cont'd.)

COS CVEs in each 1.20 auto-upgrade release, as % of COS CVEs in earliest 1.20 auto-upgrade release (i.e. from Oct 15, 2021)



COS CVEs in each 1.22 auto-upgrade release, as % of COS CVEs in earliest 1.22 auto-upgrade release (i.e. from Feb 3, 2022)



(Customer Challenges and Solutions, cont'd.)

The graphs show the patch lifecycle over time when using the "auto-upgrade" process, wherein Google automatically deploys patches in customer clusters within the same minor version at a regular cadence. The graphs show relative CVE percentages, i.e. the CVE count for a given auto-upgrade release, as compared to the CVE count for the first auto-upgrade release in the same minor version. As seen, the percentages generally decreased by 2X, 3X, or higher multiples, as clusters were being patched towards more current releases–as less CVEs were found in more current releases.

The suggestions below explain how customers can achieve availability and security patching in a more unified manner.

Solutions for balancing availability and security patching within GKE

1. Ensure that your clusters are in a release channel, which are Google-managed rollout policies allowing customers to choose the appropriate, relevant upgrade path for them. And the channels automatically maintain customer clusters with new Kubernetes features and security and bug fixes. Of the three available release channels (Rapid, Regular, and Stable), Google recommends the Regular channel for a useful balance between security patch speed and availability. Still–security fixes are automatically published and patched in all three channels in due time.
2. Use regularly-occurring maintenance windows (specifying when environment upgrades are allowed), of proper duration, to ensure that nodes are patched when production processing permits. (However, note–short maintenance windows may be insufficient for large clusters, to fully complete patching in one cycle).
3. Use maintenance exclusion windows (MEW, specifying when environment upgrades are prohibited), to prevent patching from occurring during non-interruptible work periods. The timeframe for the exclusion, as well as its scope, can be specified. For example, a given minor version can continue to be updated, but migrating to a newer minor version–where, for instance, API functionality could change–can be excluded via a MEW.
4. The default node pool patching method is surge upgrade, as before. If using surge upgrade for very large clusters, consider increasing the maxSurge parameter beyond the default of 1. More than one node will be patched simultaneously, allowing for faster node fleet patching. If the deployed application(s) can also withstand some disruption during patching, increase the maxUnavailable parameter beyond its default 0. This might take down some production nodes–which by definition, should be tolerable–but will also finish the patching faster.

(Customer Challenges and Solutions, cont'd.)

5. For stateful and similar workloads, set a Pod Disruption Budget (PDB). Pods are subtasks of a workload running within a node. Setting an appropriate PDB will ensure that for session-based and similar applications, the "minimum available" pods specified in the Budget will continue executing–while the patching is occurring. The Termination Grace Period can be increased if necessary (the default is 30 second) to ensure that as pods get shutdown during the patching process a sufficient amount of time is allotted for a graceful shutdown of workload tasks, if needed.
6. For additional production workload availability, customers should set up regional clusters rather than zonal clusters when creating their GKE environment. While a zonal cluster is being patched, access to the Kubernetes API is not available, which can impact production applications dependent upon it. Regional clusters don't have this limitation, however. The zonal cluster has only one control plane, the software infrastructure managing the Kubernetes cluster. While the zonal cluster is patched, the Kubernetes API cannot be used, as the API's manager (the control plane) is unavailable. Regional clusters, in contrast, have control plane redundancy of three–and when they are patched, one control plane will be unavailable, while the others will be available. The available control planes will, among other functionality, keep the Kubernetes API operating.
7. Use the security posture dashboard (SPD), currently in Public Preview, to find various security concerns with your clusters. Through the SPD, vulnerability scans looking for workload misconfigurations, or CVEs in the OS images used on nodes, can be initiated–and the results provided in a dashboard. Vulnerability scan results are also placed into Cloud Logging for additional reportability.
8. Utilize various notification services for additional security awareness regarding your clusters. View deprecation insights and recommendations to determine which APIs or other GKE features will be changing in the future, including suggestions how to mitigate the changes within your environment. Subscribe to the GKE pub/sub to receive security bulletins and patch release information tailored to the specific cluster versions you're running. Both notifications can be used for planning and change management purposes.

The low hanging fruit: leaked service account keys and the impact to your organization

Leaked, or inadvertently shared, service account credentials continue to be one of the leading factors of abuse on Google Cloud. The Cloud Security Alliance's (CSA) 2022 "Top Threats to Cloud Computing - Pandemic Eleven" report identified "Insufficient Identity, Credentials, Access, and Key Management" as the #1 security issue facing cloud customers. In 42% of leaked key incidents detected by our abuse systems, customers did not take action after Google attempted to contact the project owner, so the key remained vulnerable to abuse. While there are many instances of new accounts or developers testing code exposing service account keys, our teams have observed compromises distributed across varying sizes and maturities of organizations. Organizations seeking to reduce and mitigate the risk must take into account key management, principle of least privilege with scoped IAM policies, policies and controls to prevent keys from leaking, as well as continually scanning and monitoring for keys that have already been exposed.

Attackers Shifting Tactics to Conceal API Calls

Our teams fighting fraud and abuse on the platform have observed attackers continually changing their defense evasion techniques to conceal the origin of API calls. Attackers have attempted to use Tor nodes, open proxies, other compromised cloud instances, and even other cloud service providers. Oftentimes attackers don't know what IAM permissions or resources are associated with a service account and instead try to maximize fraudulent profits by utilizing automation tools to spin up cloud resources such as high cpu instances as quickly as possible and across several Google Cloud zones before they're detected and instances shutdown. However, depending on the IAM permissions granted, an attacker that has discovered a service account key could cause more detrimental harm to customers.

(The low hanging fruit, cont'd.)

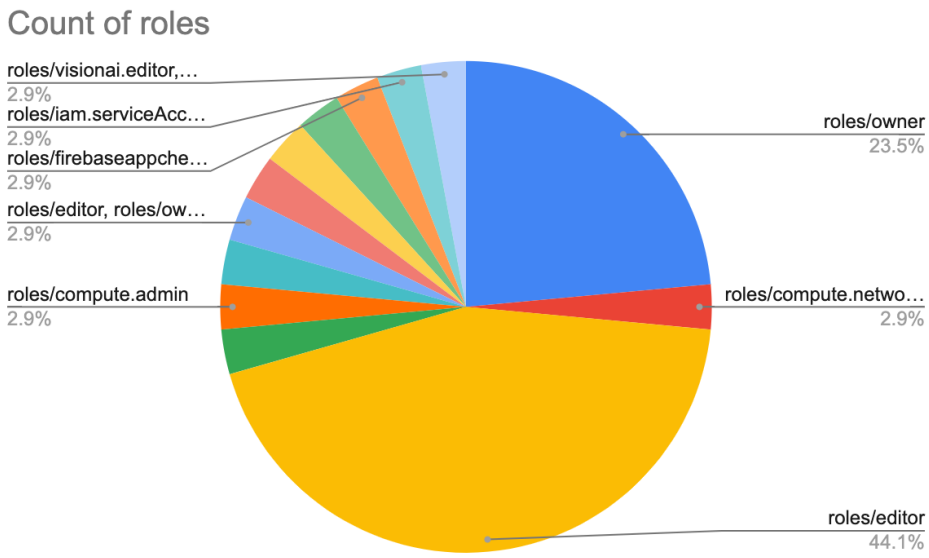


Figure 1. Snapshot distribution of IAM roles of confirmed compromised service account keys.

IAM roles control who (identity) has what access (role) for which resource. Taking a closer look at IAM roles and permissions associated with leaked and compromised service account keys, our abuse data surfaces that 67.6% of keys in confirmed incidents of customer compromise used basic IAM roles (23.5% had Owner roles, and 44.1% had Editor roles). These basic roles increase an organization's risk by including thousands of permissions across all Google Cloud services. Palo Alto's Unit42 Cloud Threat Research reported "99% of cloud users, roles, services, and resources were granted excessive permissions." Among Unit 42's observations, AWS's AdminstratorAccess and Azure's Owner were among the top 5 most used policies, similar to what our Fraud and Abuse teams have reported with IAM roles assigned on Google Cloud.

Hardcoded credentials checked into code repositories

One of the most common situations observed when keys are discovered leaked includes a developer downloading a service account key, which is a RSA private/public key pair that grants long-lived access, and checking code into a public code repository with the key hardcoded. Similarly, there are instances of developers checking in code into private repositories which at some point were updated to become public and therefore exposing keys that currently exist or have at any point in the repository's history. Bad actors are continually scanning for this low hanging fruit to spin up cloud resources. As noted in the January 2023 Threat Horizons Report, one of the most targeted software in Q3 2022 was IT automation

(The low hanging fruit, cont'd.)

software Jenkins which is used in CI/CD infrastructure. What is less obvious is that these keys, among other credentials, are also found in an organization's commit history and even CI/CD logs such as when they're passed as a command line argument going unnoticed for an extended period of time. IBM's Cost of Data breach 2022 found stolen or compromised credentials to be the most common cause of a breach, 19% of breaches, and the longest life cycle of 243 days to identify. A developer scanning the Python Package Index (PyPi) recently identified 57 valid AWS keys, including from Amazon themselves, and found the oldest active key to be uploaded 10 years ago.

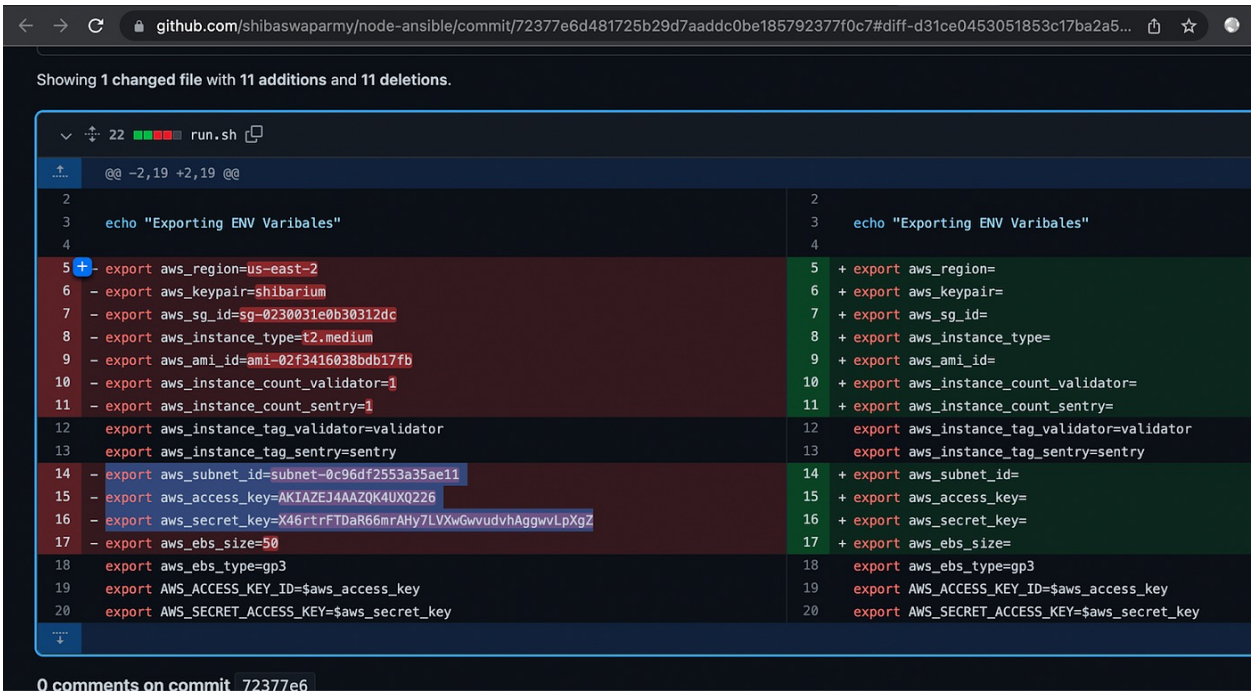


Figure 2. Image of a Github repo with an AWS key found in the commit history of an Organization from the PingSafe blog.

(The low hanging fruit, cont'd.)

Mandiant has observed service account keys being used in other concerning ways such as insider risk, cross-cloud breaches, as well as reconnaissance in Google Workspace. In one such example a disgruntled employee downloaded AWS service account keys ahead of a termination and later used those keys to delete data from a production database in Amazon's Relational Database Service.

In another instance, an attacker was able to breach two clouds with one service account key. The attacker scanned a public code repository and discovered a hard coded AWS service account key and using that credential they were able to gain access to a customer's AWS instance which in turn hosted an internal code repository. Next the attacker was able to discover a hard coded Azure credential within the internally hosted code repository and use that key to gain access to the customer's Azure environment.

Access to service account credentials can also allow attackers to span across services such as when a GCP service account is granted domain wide delegation authority to an organization's Google Workspace environment. In one instance, a developer had their laptop compromised and an attacker gained access to the service account key which was downloaded and had been granted domain-wide delegation with an API scope that could access Gmail and Drive. The attacker was able to perform reconnaissance within Workspace with access to that organization's emails and files.

Leaked service account keys, and persistent credentials in general, expose organizations to risk and can lead to serious consequences. Though keys are predominantly found in code repositories they can be harvested in several other locations such as within archive files, emails, slack messages, pastebin, previous leaks/disclosures, open storage buckets, and more. Organizations should reduce their reliance on long-term credentials and where this isn't feasible, employ the strategy of defense in depth in conjunction with other practices such as the principle of least privilege to reduce and mitigate the risk of leaked keys. There are several mitigation strategies below organizations could leverage in Google Cloud and their own environment to strengthen their security posture and reduce their exposure.

Mitigations

Assess the need for Service Accounts and follow best practices

- Assess whether service accounts with long-lived credentials are needed - most scenarios don't require a downloaded service account key.
  - » For local development, use your user credentials to create and authenticate with application default credentials or service account impersonation.
  - » For production workloads, assess which authentication method is appropriate.
- Create service account keys only when you have a scenario that cannot be met with the previous options, and follow best practices for Service accounts and best practices for managing Service Account keys.

Keep an inventory and audit usage

- Introduce a naming convention when creating service accounts with a name or description that is indicative of the purpose such as `<team_name>-<resources_accessed>-<purpose>`.
- Audit who has access to the service account and how they're being used.
  - » For service account usage, use the IAM Policy Analyzer to check which identities can use or impersonate a service account.
  - » For service account key usage, you can monitor authentication activities with Activity Analyzer.
- Create a policy to disable accounts not used in a while and revoke employee access during offboarding.
- Monitor Audit Logs and leverage pre-built queries to identify anomalous behavior with service accounts.

Have a plan in place when keys are exposed and test your assumptions

- Create or update playbooks and run tabletop exercises within your organization to respond to a leak by disabling, rotating keys, and revoking access.
- Document the process of rotating keys (what will break, what are the dependencies, who are the owners). CACAO Security Playbooks is a community standard that helps organizations create such playbooks in a structured and standardized manner.
- Partner with your red teams to test and validate your assumptions of preventing, detecting, and remediating exposed credentials.
- Assure all essential contacts are up to date so the right stakeholders receive important Google Cloud notifications and action them in a timely manner.

(Mitigations, cont'd.)

Enforce Organizational policies, scope IAM permissions and use IAM recommender

- Consider applying organization policies broadly across your organization, particularly `iam.automaticIamGrantsForDefaultServiceAccounts`, constraints/iam.serviceAccountKeyExpiryHours and `iam.disableServiceAccountKeyCreation`. These policies help prevent the proliferation of unmanaged keys and the default Editor role granted to default service accounts. Apply these policies as high in your hierarchy as possible (the organization node) and grant limited project-level policy exceptions only for well-vetted exceptions that must use a service account key.
- Don't use Basic roles such as Owner, Editor, and Viewer. `Use the IAM recommender to evaluate permissions and identify which permissions could be removed.

Prevent keys from being checked in and monitor for exposed keys

- Introduce automation into your build system such as pre-commit checks to prevent keys from ever being checked in. Cloud Build can create build triggers and there are also open source projects such as git-secrets which use Git hooks.
- Use Security Command Center's Anomaly detection to view findings for leaked service account credentials. Google Cloud's DLP offering can also detect credentials and secrets from GCP, AWS, and Azure.
- Consider open source tools such as Trufflehog and ScoutSuite to identify exposed keys or open storage buckets to assess your organization's security posture. BFG Repo Cleaner can help organizations cleanse git commit history including removing credentials.

Google Cloud