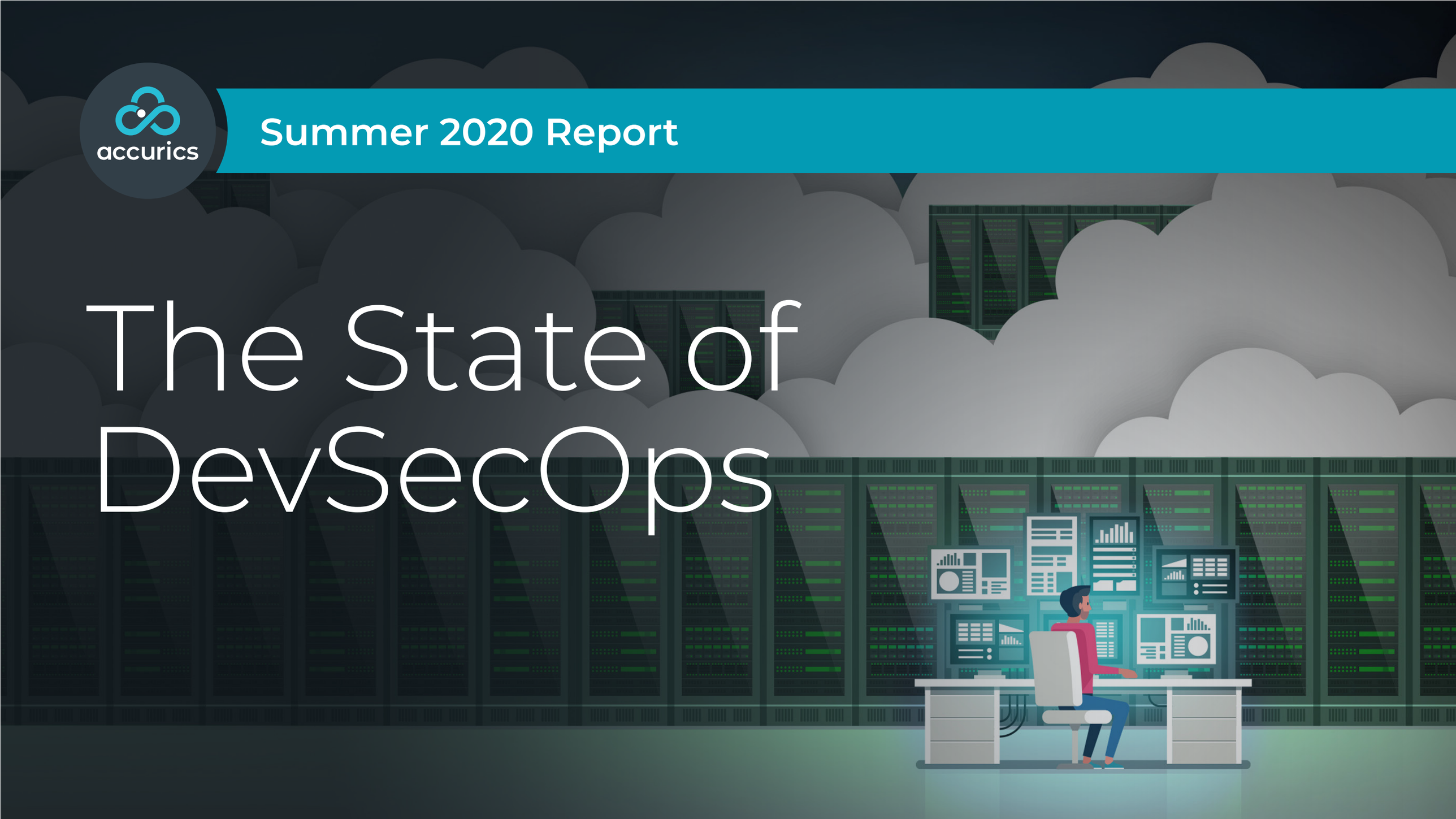
accurics

Summer 2020 Report

The State of DevSecOps



Contents

Foreword

The adoption of cloud native infrastructure such as serverless, containers, and service mesh are enabling organizations to deliver new innovations to market. Unfortunately, over 30 billion records have been exposed as a result of cloud infrastructure misconfigurations over the last two years and the velocity of cloud breaches continue to increase.

It is now more important than ever to understand cloud infrastructure configuration practices that are creating exposures. In this report, we analyze top issues, glean lessons from past cloud breaches, and recommend DevSecOps practices that organizations should consider as they embrace cloud native infrastructure. I hope you find the information in this report informative and actionable. Please reach out to me if you have any feedback.

Om Moolchandani

Om Moolchandani

CTO @ Accurics

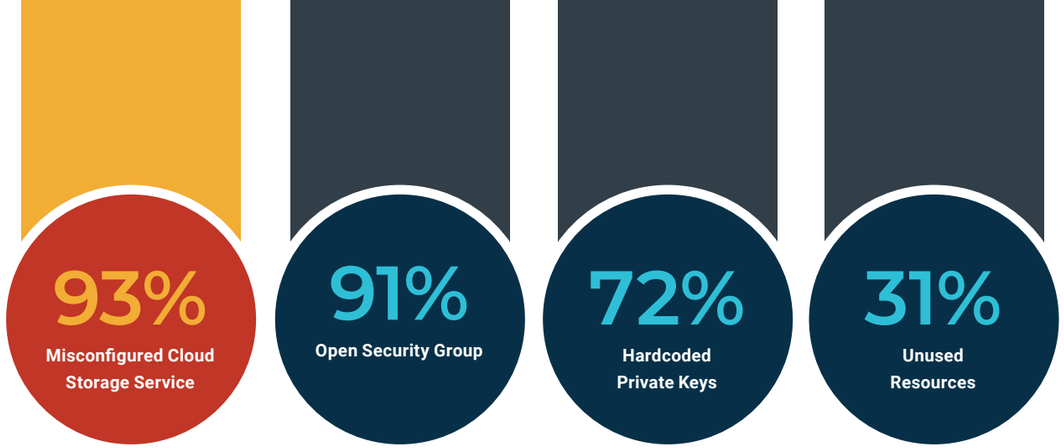
Missing the Fundamentals

New types of simple cloud misconfigurations are emerging

Since the Spring 2020 edition of the State of DevSecOps report, the usual slew of cloud data breaches were observed. Cloud storage services were compromised that led to **845 GB of information** from at least eight popular dating apps containing highly personal and sensitive data being exposed. The personally identifiable information of more than 99,000 customers of a global nutrition and fitness brand, **V Shred**, may have been exposed due to an unsecured cloud storage service. Last but not least, 409 GB of data containing financial and personally-identifiable information (PII) connected to India's largest payment application, **Bharat Interface for Money (BHIM)**, was exfiltrated.

Cloud breaches have been pretty rampant over the last two years and the latest research revealed that this trend will likely increase in velocity and scale. Misconfigured cloud storage services were commonplace in 93% of cloud deployments that were analyzed. Most deployments also had at least one network exposure where a security group was left wide open. These two practices alone have been at the center of over 200 breaches that exposed 30 billion records in the past two years.

Beyond the usual suspects, researchers also noticed some emerging practices that are creating exposures. Despite the availability of popular tools such as HashiCorp Vault and AWS Key Management Service (KMS), hardcoded private keys were found in 72% of deployments. Specifically, one in two deployments had unprotected credentials stored in container configuration files, which is worrisome given that **84% of organizations are using containers.** These keys and credentials could be used by unauthorized users to gain access to sensitive cloud resources.



Another observation was that 31% of organizations have unused resources. Further investigation revealed that the primary reason for this stems from the fact that resources are added to a default Virtual Private Cloud (VPC) upon creation if a scope is not defined, rather than an authorized VPC. Aside from the cost implications, unused resources may go undetected during security assessments, creating potential exposures.

Key Learning

While the severity of the recent breaches may suggest that cloud infrastructure security is not top of mind, that is likely not the case. Organizations are leveraging a variety of security capabilities built into cloud platforms as well as implementing a number of third party cloud security tools. However, most security controls are implemented in runtime to detect exposures. To make matters worse, research from earlier this year shows that only 4% of issues that are detected are actually addressed. The only way to reduce such exposures is to detect and resolve policy violations earlier in the development lifecycle and ensure that cloud native infrastructure is provisioned securely to being with. As organizations embrace Infrastructure as Code (IaC) to define and manage cloud native infrastructure, it becomes possible to codify policy checks **(Policy as Code)** into development pipelines.

Back to contents >>

Going Beyond Policy Checks

High severity cloud misconfigurations are not being addressed

While it is important to ensure that best practices are implemented across cloud native infrastructure during development, developers can quickly become inundated with alerts about policy violations. For example, if a number of hardcoded private keys are discovered in a deployment, the risk from each key will need to be assessed. If a key is unused, it does not pose immediate risk compared to one that is used. The time required to manually assess risk creates a window of opportunity for attackers. Automated threat modeling using frameworks such as the **MITRE ATT&CK® Matrix for Enterprise** and the **Common Attack Pattern Enumeration and Classification** (CAPEC) are essential.

Accurics researchers investigated the three most common threats plaguing cloud deployments. As stated earlier, most organizations had hardcoded keys in their configurations. However, 41% of organizations had one or more hardcoded keys with high privileges that were used to provision compute resources; a breach involving these keys will expose all resources associated with them.

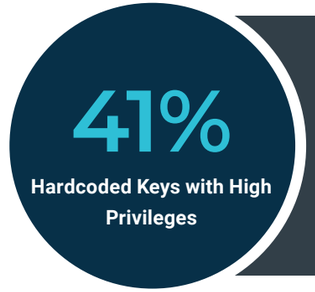
Overly permissive IAM policies threatened the majority of deployments that were analyzed. While there might have been legitimate reasons for the elevated privileges for a particular cloud resource, most organizations failed to assess the downstream impact of the elevated privileges on other resources that were using the policies. In 89% of deployments analyzed, the policies were being used by one or more resources that are highly sensitive; to remediate the issue, the privileges should not be increased or a separate IAM policy must be applied to those resources.

Network exposures resulting from misconfigured routing rules posed the greatest risk to all organizations. In typical cloud architectures, an application resides in a subnet which is connected to an internet gateway. Sensitive resources used by the application such as databases are hosted in private subnets and a route is created between the public and private subnets to enable communication. In 100% of deployments analyzed, a route existed that exposed the private subnet to the internet. This is particularly challenging for organizations to detect with simple policy based checks which only detect if a subnet is exposed to the internet but do not assess if the subnet contains highly sensitive resources such as databases.

Prominent attacks have occurred as a result of these misconfigurations

Given the severity of the risk posed by the top three threats, researchers studied recent cloud breaches to determine whether attackers were taking advantage of these issues. Research revealed that a combination of these threats created **breach paths** that were exploited in three prominent breaches: **CenturyLink**, **Imperva**, and **Capital One**.

In September 2019, a breach was discovered at **CenturyLink** that exposed 2.8 million customer records consisting of name, email address, phone number, home address, CenturyLink account number, notification logs, and conversation logs. An analysis of the breach revealed that there was a cloud **network misconfiguration** that exposed a MongoDB database. A non-admin account with **overly permissive IAM** policies was able to access the database due to the network misconfiguration. Lastly, the data within it was exposed since the **database was not encrypted**. Most concerning is the fact that the breach went undetected for approximately 10 months.



Back to contents >>

CenturyLink Breach Path

Imperva suffered a similar fate in August 2019 where a database snapshot containing emails, hashed and salted passwords, and some customers' API keys and TLS keys was exposed. While the exposure of emails and passwords is concerning, the exposed API and TLS keys could be leveraged by attackers to break companies' encryption and access corporate applications directly. An analysis of the breach revealed that a test cloud environment was created and a compute resource was misconfigured which exposed it to the internet (**network misconfiguration**). That compute instance contained a **hardcoded API key** which was discovered by the attackers and used to access the database. The API and TLS keys were not hashed (**best practice violation**) which ultimately put Imperva's customers at risk. This breach also went undetected for approximately 10 months.

Imperva Breach Path

Just prior to the Imperva breach, one of the largest cloud breaches to-date was discovered at Capital One in July 2019. It affected 100 million individuals in the United States and approximately 6 million in Canada. Data exposed included approximately 140,000 Social Security Numbers (SSN), 80,000 bank account numbers on U.S. consumers, and 1 million Social Insurance Numbers (SIN) for Canadian credit card customers. Analysis by the Federal Bureau of Investigation revealed that a vulnerability in a cloud compute resource was exploited which provided the attacker with a set of AWS access keys that were associated with an **IAM role with excessive permissions**. The attacker was able to discover and access a cloud storage service in the environment containing the unencrypted data (**policy violation**).

Capital One Breach Path

Key Learning

Policy as Code should be implemented to ensure that obvious best practices are employed such as encrypting databases, rotating access keys, and implementing multi-factor authentication. However, automated threat modeling is also necessary to determine if changes such as privilege increases and route changes create breach paths in a cloud deployment. As a result, organizations must augment Policy as Code with **Security as Code** when infrastructure is defined during development (Infrastructure as Code).

Back to contents >>

The Inconvenient Truth

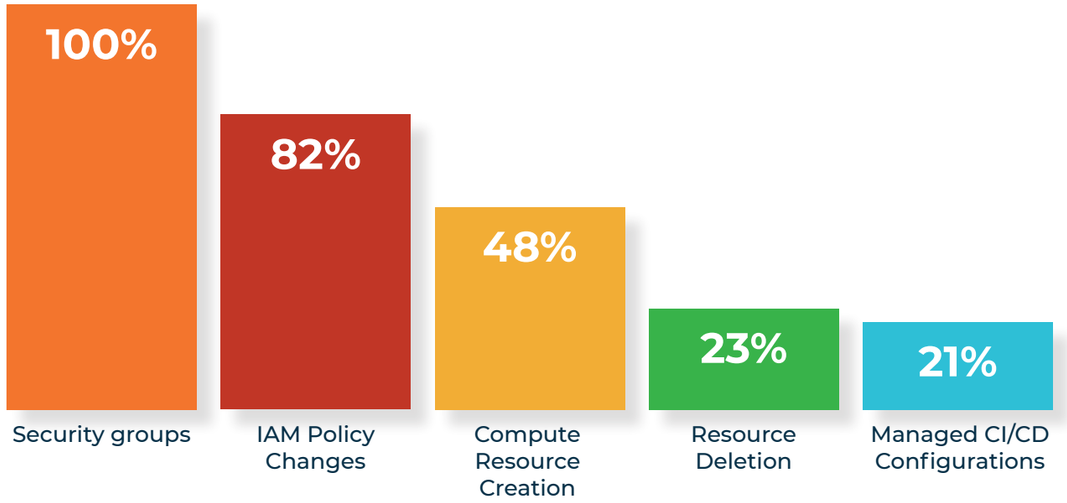
Cloud risk posture eventually drifts from a secure state

Earlier this year, Accurics' researchers determined that 90% of organizations do allow users to make changes to cloud native infrastructure in runtime. That number remains largely unchanged today. Below is a list of the top cloud infrastructure changes (drifts) that occur in runtime:

1. **Security groups** were created or modified in 100% of deployments
2. **IAM policy changes** were made in 82% of deployments
3. **Compute resource creation** occurred in 48% of deployments (examples: VPCs and EC2s)
4. **Resource deletion** occurred in 23% of deployments (provisioned through Infrastructure as Code and then deleted without updating the code)
5. **Managed CI/CD** platform configurations are being managed entirely in runtime (21% of deployments) since it is a new capability and Infrastructure as Code does not exist yet to define during development (example: AWS CodeBuild)

There is a strong correlation between the top types of cloud infrastructure drifts and the risks that create serious exposures. This implies that even if organizations exercise strong security hygiene when cloud native infrastructure is initially defined, drifts in runtime will create exposures.

Figure: Top Cloud Infrastructure Drifts



Key Learning

Cloud native infrastructure must be secured through its lifecycle: from the moment it is defined and throughout runtime. For some organizations, the cloud native infrastructure is defined in runtime through scripts, while others have adopted Infrastructure as Code (IaC) and it is defined during development. Regardless of the approach, cloud native infrastructure should be assessed for risk and issues should be mitigated **before it is provisioned** to ensure a secure initial posture. Once the infrastructure has been provisioned, it should be **continuously monitored** and any resource or configuration changes should be automatically assessed for risk (**Drift as Code**). If the change is legitimate, the original configuration scripts or IaC must be updated to reflect the change so that it is not lost if the infrastructure is redeployed. If the change introduces risk, the change must be reverted. This practice of eliminating risk posture drift is known as **Immutable Security**.

Back to contents >>

The Yin and Yang of Security

Remediation must also be codified into development pipelines

While most organizations will agree with the need for embedding security across cloud native infrastructure throughout the development lifecycle, it cannot come at the cost of agility. Detection and remediation of risks form the yin and yang of security; for agility to be maintained, both aspects must be codified into development pipelines. Policy as Code, Security as Code, and Drift as Code enable risk detection at the speed of DevOps. Unfortunately, the remediation of risks remains largely a manual process and only 6% of issues are addressed.

An emerging practice known as Remediation as Code is enabling organizations to resolve issues at the speed of DevOps. Once a risk is detected, the code to fix the issue is automatically generated and sent to the developer. The developer needs to simply review and accept the change. The results of this approach have been extremely encouraging: 80% of risks were addressed. The remaining 20% of risks could not be addressed without input from the developers, although the research and development team is making progress towards a solution. A future edition of the report will deep dive on the results from the new approach.

The next level of maturity for Remediation as Code involves automatically applying code during the build and deploy phases to override misconfigurations. While there are inherent dangers in automation, organizations appear to have an appetite to implement this for certain high risk misconfigurations. Examples include misconfigurations that create cloud storage service, cloud database service, and network exposures. This approach paves the path towards self-healing cloud native infrastructure. As more organizations adopt this approach, an analysis of the results will be detailed in future editions of this report.

Key Learning

Automated detection of risks paired with a manual approach to remediation is leading to alert fatigue and issues are not being addressed. Codifying remediation into development pipelines (**Remediation as Code**) is the only scalable way to ensure that automatically detected risks are addressed.

Back to contents >>

Recommended Best Practices

The key takeaway from the research is that the nature of cloud native infrastructure demands that security must be codified into development pipelines and enforced throughout the lifecycle. There are four distinct requirements for security to be effective:

Policy as Code:

New types of simple misconfigurations across cloud native infrastructure are emerging; policy guardrails must be embedded throughout the development lifecycle.

Security as Code:

High severity misconfigurations are not being addressed and have been exploited by attackers; automated threat modeling across cloud native infrastructure must be employed throughout the development lifecycle.

Drift as Code:

Change to cloud native infrastructure in runtime ultimately causes risk posture to drift; resource and configuration changes from a secure baseline must be detected and assessed for risk.

Remediation as Code:

Automated detection of risks across cloud native infrastructure paired with a manual approach to remediation is leading to alert fatigue and issues are not being addressed; remediation must be codified into development pipelines.

About Accurics

Accurics enables immutable security for immutable infrastructure so that organizations can embrace the latest cloud native technologies with confidence. The platform seamlessly scans Infrastructure as Code to detect and resolve risks before cloud infrastructure is provisioned. It subsequently monitors the deployed infrastructure for changes that introduce risks and enables organizations to revert to a secure posture. With Accurics, organizations can ensure compliance, governance, and security across their full cloud native stack.

Learn more at www.accurics.com

Methodology

The research was conducted by analyzing cloud native infrastructure across hundreds of deployments across Accurics customers and community users. The Accurics research team consists of elite security analysts, data scientists, and data engineers with deep security expertise.

Back to contents >>