Fugue

The Engineer's Handbook on Cloud Security

A primer on securing your cloud infrastructure and demonstrating compliance

FIRST EDITION

Contents

Introduction 3

You own the security of your cloud 3

Misconfiguration: The #1 cause of cloud-based data breaches 3

Distributed teams, cloud security, and COVID-19 4

Why Cloud Security is Different 5

How cloud upended traditional IT security 5

How hackers changed strategy with cloud 7

Zombie infrastructure: The invisible cloud risk 7

The Nature of the Cloud Misconfiguration Problem 8

The causes of cloud misconfiguration 8

The frequency of cloud misconfiguration 9

The types of cloud misconfigurations and critical incidents 9

Cloud compliance and security policy 10

Assuring the compliance of cloud environments 10

What compliance misses in the cloud 10

Policy-as-code for cloud infrastructure 11

DevSecOps for cloud infrastructure 12

Managing the cloud misconfiguration problem 12

The high cost of managing cloud misconfiguration 12

The “shift left” on cloud security 13

Using automated remediation for cloud misconfiguration 14

Measuring the effectiveness of your cloud security (MTTR) 14

Climbing the cloud security mountain 15

Fugue

Introduction

You own the security of your cloud

The cloud has radically changed the IT security landscape. The vulnerability surface is different. Attackers operate differently. And how you go about keeping your data safe needs to be different.

Unlike the data center, the cloud is 100% software-defined. Everything is knowable, and everything is programmable. That wasn't possible before.

And because misconfiguration is the number one cause of cloud-based data breaches, cloud security is now a software engineering problem, not a traditional security analysis one.

It's a problem tailor-made for engineers to tackle. Compliance officers can hand you a book of policies. Security analysts can tell you about vulnerabilities they're seeing. And analysis tools may be able to tell you if you were hacked—but only if those tools are really good and you're really good at using them.

All of these have their place, but you simply can't count on being able to stop cloud misconfiguration attacks in progress. But if you prevent misconfiguration, you can keep these breaches from happening in the first place.

Prevention is the operative word here. And only the engineers building and modifying their cloud infrastructure have the means to prevent misconfiguration.

You can apply the same programming concepts to cloud security as you do your applications and infrastructure. Do this well, and you can eliminate misconfiguration and prove compliance at any time, all of the time.

You'll also eliminate the big mess of tedious, repetitive tasks that we think of when it comes to security and compliance.

If your competitors are still mired in that big mess of tedious, repetitive tasks, you can move faster than they can. Not only will you reduce serious risks to your organization, you'll turn security and compliance into a competitive advantage.

Our objective with this handbook is to help you think differently about cloud security.

You'll better understand the nature of the threats and think more critically about how to prevent them.

Let's go.

Misconfiguration: The #1 cause of cloud-based data breaches

Cloud security is configuration security.

Cloud operations are focused on the configuration of cloud resources, including security-sensitive resources such as networks, security groups, and access policies for databases and object storage. And you need to make sure the configuration of your cloud resources are correct and secure on day one, and that they stay that way on day two and beyond.

Industry analysts call this *Cloud Security Posture Management* (CSPM). And this is what cloud customers tend to get wrong all the time, often with devastating consequences. Many of the data breaches that make the headlines are the result of cloud misconfiguration attacks.

"There is more risk from cloud infrastructure misconfiguration than from workload compromise."
— Market Guide for Cloud Workload Protection Platforms, Gartner

"Cloud technology moves rapidly, making oversight a complex task."
— The National Security Agency, Mitigating Cloud Vulnerabilities

"Nearly all successful attacks on cloud services are the result of customer misconfiguration, mismanagement and mistakes."
— Neil MacDonald, Gartner

"I'm seeing a lot of cloud configuration errors in the real world—and it's scaring the hell out of me."
— David Linthicum, InfoWorld

"Skilled or well-funded hacker groups are employing automation to discover and exploit misconfigured cloud assets within hours of their deployment."
— John Breeden II, CSO Online

We tend to focus a lot on avoiding misconfiguration for cloud resources such as object storage services (e.g., Amazon S3; Azure Blob) and virtual networks (e.g., AWS VPC; Azure VNet), and that's absolutely critical.

But it's also important to recognize that cloud security is also all about identity. In the cloud, many services connect to each other via API calls, requiring Identity and Access Management (IAM) services for security rather than IP based network rules, firewalls, etc.

For instance, a connection from a Lambda to an S3 bucket is accomplished using a policy attached to a role that the Lambda takes on—its service identity. IAM and similar services are complex and feature rich, and it's easy to be overly permissive just to get things to work, which means that overly-permissive (and often dangerous) IAM configurations are the norm.

But because cloud IAM services are created and managed with configuration, cloud security is still all about configuration—and avoiding misconfiguration.

Misconception: "We're using the latest threat detection and firewall security tools in our cloud environment, so our sensitive data is secure."

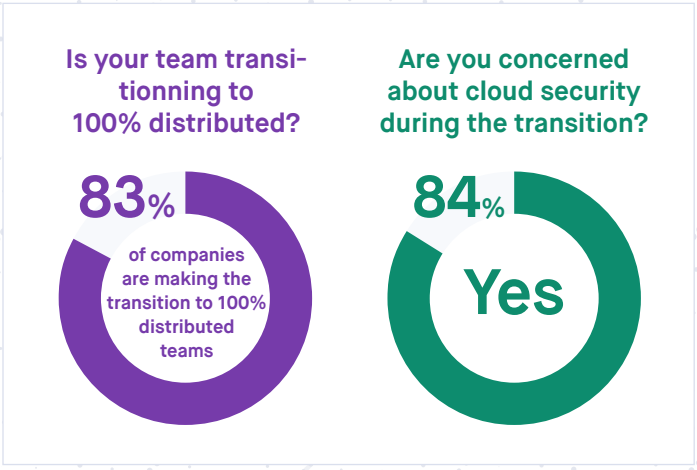
Reality: Traditional security tools can't prevent, detect, or stop advanced cloud misconfiguration exploits. You need effective Cloud Security Posture Management for that."

Distributed teams, cloud security, and COVID-19

Since the COVID-19 crisis began, just about every employer has transitioned to 100% work-from-home for every team that can do so. Even for those that were already operating with some form of distributed team arrangement, this shift has likely caused significant disruption.

This transition likely resulted in new cloud vulnerabilities, and it's better to find and eliminate them before attackers do. You likely need to update protocols to accommodate new access patterns from employees who previously worked at the office.

And with team members experiencing additional pressures at home, they're understandably stressed and mistakes are more likely.



But this crisis has only highlighted the fundamental need to maintain secure access to cloud environments. Here's where to start:

1. Secure all devices

Ensure all corporate devices used to access and manage cloud environments are under corporate control. With modern public cloud services, we can manage and protect the physical devices used by distributed teams. Use a mobile device management (MDM) tool and make sure all devices that are used for work have disk encryption turned on.

Enforce strict password requirements and make sure team members are locking their screens and set policies that encrypt (brick) the device, making it unusable after a defined amount of attempted logins. This mitigates brute force attacks.

Extend your organization's antivirus solution to any other devices that employees are using while working from home. Adopt or clarify policies regarding the use of personal devices to access company data or accounts, and prohibit it if you can't ensure their security.

2. Lock down your access policies

If you don't already have a formalized access policy for your accessing and managing your cloud environments, now's the time to create one.

Use Virtual Private Networks (VPNs) to enforce secure communications to critical network spaces (e.g., AWS VPC or Azure VNET). Make VPN access available or required so that the team can access company resources even if they are on a less trusted Wifi network.

Engineers are prone to creating new security group rules or IP whitelists so that they can access shared team resources in the cloud. Frequent audits can certify that virtual machines or other cloud infrastructure haven't been put at additional risk. Oversee the creation bastion hosts, lock down source IP ranges, and monitor for unrestricted SSH access (e.g., 0.0.0.0/0 on Port 22).

In the public cloud (e.g., AWS; Azure; GCP), Identity and Access Management (IAM) acts as a pervasive network. Follow the principle of least permission. Make IAM changes a part of your change management process. And make use of privileged identity and session management tools.

3. Get visibility and control over your cloud environment

Humans make mistakes, and it's unrealistic to expect everyone on your team to follow policies to the tee at all times. That's even more the case when it comes to cloud security, because the threat surface is complex and dynamic, and the rulesets can be expansive.

You need to understand the full configuration state of your cloud environment. Don't rely on spreadsheets or infrastructure-as-code files to tell you what's running and how it's configured.

Set up notifications so you know when configuration changes are made to security-critical resources (e.g., IAM; security groups; object storage; databases). Quickly identify and remediate dangerous misconfigurations when they occur. We'll have a lot more on this later.

With vigilance, communication, and understanding, your team can manage these transitions and ensure your cloud assets stay secure..

4. Tune your processes with security in mind

Leverage teleconference tools to conduct configuration checks and peer reviews of code. With a newly distributed team, these peer reviews may take on increased importance. Consider requiring more than one person to review pull requests in your code repository before merging.

Expand the use of your existing ticketing system for internal tech requests. Remote team members are going to need changes to their devices and cloud environments, and anyone should be able to open a trackable ticket from any internet connected device to get a request moving.

Why Cloud Security is Different

How cloud upended traditional IT security

The cloud represents the most disruptive trend in enterprise IT in decades. It's understandable for security professionals to feel a loss of control over the cloud, and for engineers to be frustrated with security and compliance processes that don't work the way they work.

By understanding how the cloud has changed IT, we can begin to change how we secure it.

The Shared Responsibility Model

While the cloud has introduced new kinds of security risks, we should recognize that the cloud has relieved us of some really big security responsibilities, primarily physical data center infrastructure.

The Shared Security Model of Cloud dictates that Cloud Service Providers (CSPs) such as AWS, Azure, and Google Cloud Platform are responsible for the security of the physical infrastructure. CSP customers (that's you!) are responsible for the secure use of cloud resources.

But many misunderstand the Shared Responsibility Model and assume their CSP is responsible for more than they actually are. And this misconception can bring great risk. Take the time to fully understand your responsibilities as the cloud customer.

Further Reading:
The Shared Responsibility Model (AWS)

Developers are making decisions about infrastructure

One of the most profound changes with the cloud is that developers are building and modifying their own cloud infrastructure. In doing so, they're making their own infrastructure and security decisions—and then changing them, constantly.

Because the cloud is self-service and infrastructure resources are available on-demand via Application Programming Interfaces (APIs), developers tend to move fast and automate, sidestepping traditional security gatekeepers.

When developers spin up cloud environments for their applications, they're also defining the security of their infrastructure through configuration (and re-defining it daily).

Data Center	Cloud
Physical infrastructure	Software-defined infrastructure
Generally static	Highly dynamic
Manually deployed, configured, and maintained	Deployed, configured, and maintained via APIs • Manually, or... • Automated with Infrastructure as code (Terraform: Cloudformation)

So, even if developers get the security of their cloud infrastructure correct on day one, they may have introduced a misconfiguration vulnerability on day two (or hour two). No human can memorize thousands of rules.

There's more kinds of cloud infrastructure... and more of it

In certain respects, security in the datacenter is easier to manage. You have your network, firewalls, and servers on racks. The cloud has those too, in virtualized form. But with cloud has come a flurry of new kinds of infrastructure resources, like serverless and containers.

AWS alone has introduced hundreds of new kinds of services over the past few years. Even once familiar things like networks and firewalls can appear alien to the data center engineer. All require new and different approaches to security.

And due to the on-demand and elastic nature of the cloud, there's more infrastructure resources to track and secure. Enterprise cloud teams may be running dozens of environments across multiple regions and accounts, and each may involve tens of thousands of resources, all individually configured and accessible via APIs.

These cloud resources interact with each other and require their own identity and access control (IAM) permissions. Microservice architectures only compound this challenge.

Traditional security doesn't help much with cloud security

By now, you've probably concluded that many of the security tools that worked in the data center aren't of much use in the cloud. Security has traditionally been focused on the "perimeter" and endpoints. In the cloud, there really is no perimeter (if there ever was one).

For instance, application security still matters in the cloud, but network monitoring tools that rely on spans or taps to inspect traffic don't because CSPs don't provide direct network access. Data doesn't typically traverse TCP/IP networks in cloud environments, and neither do cloud data exfiltration events.

Of course, this doesn't mean you should ditch every tool and method you relied upon for security, but it's good to understand which still apply, where, and how—and which are useless.

Your cloud can be more secure than your data center ever could be

In other respects, security can be easier in the cloud than in the data center. Because the cloud is fully programmable and can be automated, the security of your cloud is also programmable and can be automated. In the midst of all this cloud chaos lies opportunity!

You can, and should, continuously monitor for cloud resource misconfiguration and drift from your approved and provisioned configuration baseline. You can protect critical resources and sensitive data using automated remediation. We'll dig into that topic more later.

And before developers provision or update their cloud infrastructure, they can run automated tests to validate that their infrastructure-as-code and dev environments comply with security policies, just like they do with their application code. This lets developers know earlier on if there are problems they need to fix, and it helps them move faster and keep innovating.

And these same validation tools can be used to include cloud security as part of automated deployment processes (i.e. CI/CD). We'll cover "Shift Left" more later.

And proving compliance can be a lot easier in the cloud

There's also good news for cloud engineers who are responsible for making sure their cloud infrastructure adheres to industry compliance controls and internal policies.

Manual cloud infrastructure audits can be incredibly costly, error-prone, and time-consuming, and they're usually obsolete before they're even completed. If the cloud is programmable and can be automated, compliance audits and reports can be automated, saving everyone time and avoiding mistakes.

How hackers changed strategy with cloud

If you're running a workload in the cloud, take a moment to look at the activity logs for your public-facing resources. There's bad guys there, and they're probing your cloud infrastructure looking for misconfigurations they can exploit.

A lot has changed with how hackers go about stealing data or otherwise damaging an organization. The traditional approach involves picking an organization to target, and then searching for vulnerabilities to exploit.

A high-profile example is the infamous Sony hack. A certain country was unhappy about a particular movie, and they set about to attack the media company that produced it. Using a mix of backdoors, proxy tools, and malware, the attackers were able to destroy assets and publish sensitive data on the Internet. This is called an Advanced Persistent Attack, or APT.

Of course, this still happens and it must be guarded against. But another, more broadly dangerous kind of threat has emerged with cloud computing. Malicious actors now use automation tools to scan the entire internet searching for cloud misconfigurations, such as unrestricted SSH access (e.g., 0.0.0.0/0 on Port 22), orphaned and unpatched compute instances, and many others.

They don't have to look too hard, and what they get back is essentially a long shopping list of cloud environments they can access. It's relatively trivial to discover who owns these environments, so the attacker goes shopping.

Hacker Strategy Before Cloud	Hacker Strategy in the Cloud
Step One: Pick your target	**Step One:** Search for vulnerabilities
Step Two: Search for vulnerabilities	**Step Two:** Pick your target

You need to ask yourself some important questions when assessing your cloud security posture, and these questions must go well beyond whether or not your environment is passing compliance audits.

Within minutes of adding a new endpoint to the internet, a potential attacker has scanned it. A single cloud misconfiguration can put a target on your organization's back and put your data at risk.

Assume for a moment that an attacker finds one of these vulnerabilities and gains access to your environment. What kind of damage could they do? Would it be easy for them to discover other cloud resources and where you store your sensitive data? Could they leverage Identity and Access Management (IAM) misconfigurations—such as overly permissive settings—to gain access to additional resources and data? Might they be able to extract that data into their own cloud account without detection?

Chances are you're not going to like the answers. Take swift action to close these gaps in your cloud security posture when you find them. But also recognize that cloud configuration "drift" happens all the time, so you need to stay vigilant. A cloud environment that's free of misconfiguration on Day 1 may not be on Day 2 (or hour 2).

If you rely too much on manual prevention and remediation of cloud misconfiguration, you're too slow to counter these automated threats, because it only takes minutes for an attacker to go from accessing your environment to making off with your data. Look for ways to automate the prevention of misconfiguration in CI/CD, and the detection and remediation of misconfiguration when it does occur for security-critical resources in production. We'll cover how to do both later.

Finally, make sure your penetration testers (pentests) include cloud misconfiguration vulnerabilities. There's a good chance they currently don't factor in these kinds of attack vectors.

Zombie infrastructure: The invisible cloud risk

We already discussed how easy it is to create cloud infrastructure resources. It's more difficult to destroy them completely. Not surprisingly, cloud customers wind up running — and paying for — far more cloud infrastructure than they actually use.

While some of those untracked resources may still serve legitimate business uses (which itself is concerning if you aren't tracking them), much of it is "orphaned infrastructure" — idle cloud resources in our environment that serve no business purpose.

Orphaned cloud infrastructure has been long recognized as a cost problem, but few recognize them as a security problem. These costly orphans can transform into dangerous zombies that invite malicious actors into your cloud environment, and they've played a key role in recent, major cloud breaches.

By definition, zombie cloud resources are:

- not tracked in your management tools
- not scanned for misconfiguration vulnerabilities
- not patched with the latest security updates
- not checked for policy compliance
- not cycled out via immutable infrastructure practices

You should assume the presence of zombie resources in your environment, and make eliminating them a priority. You'll save you money while improving your security posture. (One great thing about cloud security and compliance done right is that it usually also saves you money).

Here's some tips on how to effectively eliminate orphaned/zombie cloud infrastructure:

Establish visibility into your environment

Cloud APIs make it possible to ascertain the state of everything you have running and discover resources you may have been missing. An accurate visual diagram of your cloud environments is more useful than lists and spreadsheets because humans are visual, and orphaned resources are easily exposed when environments are visualized.

Enforce resource tags and effective tagging conventions

Using tags is one of the best ways to help you track and manage your cloud resources, and many cloud management tools rely on them. Any untagged cloud resource should be immediately suspect and targeted for termination (or properly tagged if it's needed). Establish tagging conventions that are scalable and human-readable, and enforce their proper use.

Adopt infrastructure-as-code and CI/CD pipelines

If you're operating at some scale, you'll want to adopt infrastructure as code and an automated CI/CD pipeline like Jenkins or AWS CodePipeline. It will be easier to keep track of what you're doing and identify resources created or modified outside of your pipeline.

Include dev environments in your cloud security plan

Dev environments can pose a security risk to production infrastructure and data. A lot can be learned about production accounts by gaining access to dev accounts, and hackers can leverage IAM misconfigurations to execute cross-account attacks.

The Nature of the Cloud Misconfiguration Problem

The causes of cloud misconfiguration

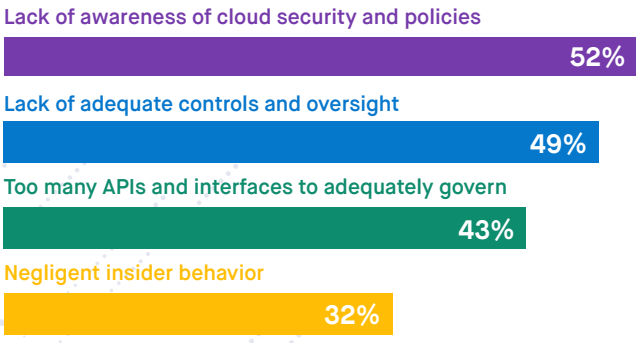
In this age of agile development and CI/CD, the only constant with cloud infrastructure environments is change, and every change brings risk of misconfiguration.

According to Gartner, through 2023 at least 99% of cloud security failures will be the customer's fault. That 1% seems like a hedge considering cloud misconfiguration is how cloud security failures happen, and misconfiguration is 100% the result of human error.

But why do engineers make these critical mistakes so frequently?

Lack of awareness of cloud security and policies was named the top cause of cloud misconfiguration. Compile all of your compliance rules and internal security policies together and you probably have something as thick as War and Peace. No human can memorize all of that.

Nor shouldn't be expected to. But 49% say their organizations lack adequate controls and oversight to prevent cloud misconfiguration mistakes. And part of the reason for that is there are too many APIs and cloud interfaces to adequately govern. For cloud infrastructure "guardrails" to be effective, they need to account for every change made using any interface.



Misconception: ***"Every change to production goes through our automation pipeline"***

If your team has adopted a CI/CD pipeline that's subject to controls and approval processes, it's not uncommon to believe that all changes to your cloud environment go through it. But engineers still make changes to production manually using the cloud console—most often by creating a new security group (i.e., firewall) rule to perform some maintenance and then forgetting to "close the door" once they're finished.

Even teams managing the most highly secure and regulated cloud environments are surprised at just how much change occurs outside of the normal deployment pipeline. Use CI/CD if you can, but always monitor for changes to your environment so you can quickly identify and remediate problems.

The frequency of cloud misconfiguration

Cloud misconfiguration vulnerabilities are different from application and operating system vulnerabilities in that they keep popping up even after you've fixed them. You likely have controls in place in your development pipeline to make sure developers don't deploy known application or OS vulnerabilities to production. And once they're there, it's generally a solved problem (at least for those specific application and OS vulnerabilities).

Cloud misconfiguration is different. It's commonplace to see the same misconfiguration vulnerability appear over and over again. A security group rule allowing for unrestricted SSH access (e.g., 0.0.0.0/0 on Port 22) is just one example of the kind of misconfigurations that occur on a daily basis with at-scale cloud environments. We use this example because most engineers are familiar with it (and have likely committed this egregious act at some point in their career).

Because cloud infrastructure is so flexible and elastic, and we can change it at will using APIs, we tend to change it a lot. That's a good thing, because we're constantly innovating and improving our applications and need to constantly modify our infrastructure to support that innovation. But when we don't have automated controls in place to guard against misconfiguration, we can expect a lot of misconfiguration to get introduced into our environment.

In the survey we conducted in March of 2020, we asked about the frequency of cloud misconfiguration engineers were experiencing in their environment. It's really high.

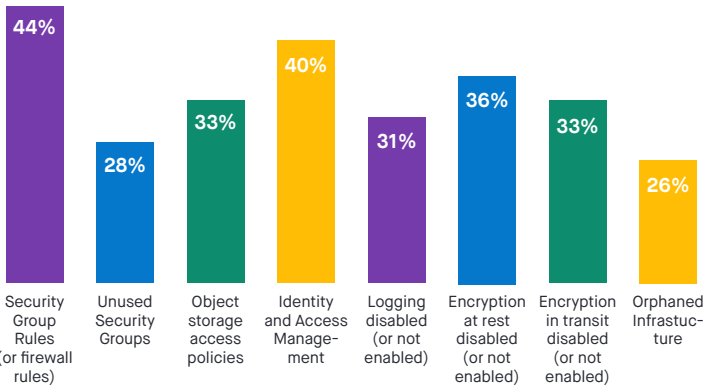
Cloud Misconfiguration Incidents per Day		
	1-10	24%
	10-50	18%
	50-100	19%
	100-250	13%
	250-500	13%
	500-1,000	7%
	More than 1,000	3%
	Don't kow	3%

The types of cloud misconfigurations and critical incidents

I've used a common example of cloud misconfiguration - security group rules allowing for unrestricted SSH access (e.g., 0.0.0.0/0 on Port 22) throughout this handbook, but admittedly at the risk of oversimplifying the problem.

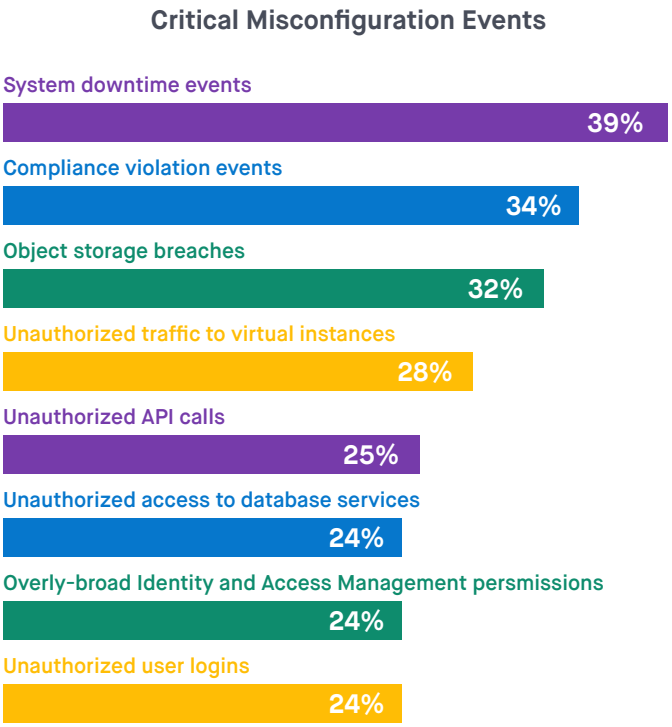
There's a lot more kinds of cloud infrastructure than there was in the data center, and all of those resources are highly configurable - and misconfiguarable. Take into account the kinds of resources and the ways they can be combined together to support applications, and the configuration possibilities are effectively infinite.

Cloud misconfigurations by type



Keep in mind that the security of resources such as object storage and IAM services can get extremely complex fast, and it's important to thoroughly understand your use cases and think critically about how these services need to be configured securely.

Cloud misconfiguration can lead to a broad spectrum of critical incidents. While data breaches are the most serious, and compliance violations can bring hefty fines, misconfiguration can also take down your application, costing your company for every minute your team scrambles to identify the issue and restore service.



Cloud compliance and security policy

Assuring the compliance of cloud environments

Just about every organization using the cloud needs to adhere to one or more industry compliance controls. Examples include HIPAA for healthcare data, PCI for financial services data, and SOC 2 for any organization that holds, stores, or processes customer data in the cloud, such as Software as a Service companies. There's also NIST 800-53, ISO 27001, GDPR, and California's answer to GDPR, the CCPA. There's others.

And for those not compelled by a particular industry regulations, there's the CIS Benchmark. It's a good idea to adhere to the CIS Benchmark controls even if you don't have to.

Each of these compliance families cover a lot more areas of your business than your cloud infrastructure environment, but they all have something to say about it and how you use it. And there's a lot of overlap between compliance families. For instance, every compliance control I just mentioned states that data at rest should be encrypted. But only [INSERT COMPLIANCE CONTROL] says you need to [INSERT UNIQUE RULE].

Fun Fact: When Fugue customers initially select which compliance families they'd like to apply to their cloud environment, the most popular selection by a country mile is "all of the above". We think that's a good sign, because while no compliance standard can protect you against every misconfiguration vulnerability (see the next chapter), it's best to cover as many of your bases as you can.

In many cases, compliance requirements are quite vague when it comes to cloud infrastructure. "Hand-wavy" even. This is actually a good thing, because overly-prescriptive rules can reduce your flexibility. The lack of specificity affords us the flexibility to bring disparate IT systems and processes into compliance.

There are people or teams at your organization whose primary job is compliance. But we can't expect compliance officers to understand the complexities and nuances of our cloud infrastructure environments and how compliance applies to them

You'll need to work with your compliance team to understand what's an acceptable level of compliance risk as it applies to your specific use case. Chances are, you'll need waivers for specific rules that aren't viable considering application requirements or or applicable when considering your use case .

Just like engineers own the security of their cloud environment, you also own the compliance of it. You'll save yourself a lot of time and headaches if you take command of this responsibility — and automate as much of the compliance processes as you can. Compliance officers and management will thank you too.

Additional Resources

AWS resources on cloud compliance

Microsoft Azure resources on cloud compliance

Google Cloud Platform resources on cloud compliance

What compliance misses in the cloud

Bringing your cloud environment into compliance with one or more of the common standards is essential, but it's not enough.

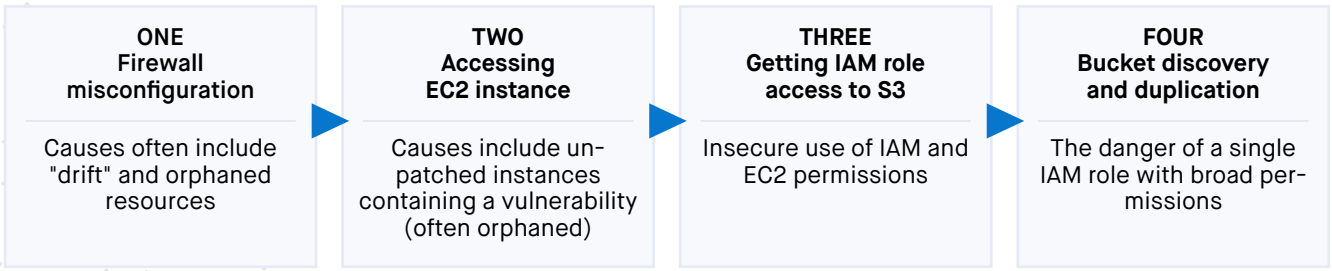
Many dangerous cloud misconfigurations—like the kinds we're seeing more and more in high-profile

breaches—are ones that are not typically recognized as vulnerabilities by security teams. And the reason they aren't is that industry compliance standards don't consider them to be policy violations.

They're also exceedingly common in enterprise cloud environments. Some examples of cloud misconfigurations that compliance frameworks miss are:

- Identity and Access Management (IAM) misconfigurations that can provide bad actors, including malicious insiders, with the ability to move laterally and discover resources to exploit
- Object storage policy misconfigurations that can be exploited in order to take data exfiltration actions
- Network security group rule misconfigurations that can enable malicious access via Elasticsearch, MongoDB, and other services

Here's an example of a cloud attack that begins with orphaned "zombie" infrastructure and proceeds to exploit a series of misconfigurations not typically addressed by compliance frameworks.



Modern cloud attacks involve multiple vulnerabilities exploited to gain access to a cloud environment, move laterally, discover resources, and extract data. All without detection using traditional security analysis tools.

Additional Resources

The Capital One cloud misconfiguration breach in 2019 involved a series of vulnerabilities that would not have been flagged by common industry compliance standards. Learn what they were and how this exploit was executed: A Technical Analysis of the Capital One Cloud Misconfiguration Breach

The Fugue Best Practices is a set of rules that address common and dangerous cloud misconfigurations that are not covered by common industry compliance frameworks. They include steps on how to identify them in your environment and remediate them.

Policy-as-code for cloud infrastructure

We recognize that enterprise-scale cloud infrastructure environments are too vast, complex, and dynamic to effectively govern using manual auditing practices. And all of the industry compliance standards and internal policies we're required to maintain involve hundreds, if not thousands, of individual rules. No human can be expected to memorize all of that.

The only way we can ensure our cloud environment stays in compliance with all required policies and prove it to management and auditors at any time is by using policy-as-code.

Policy-as-code empowers engineers to own the security and compliance of their cloud infrastructure. It can be peer-reviewed and itself audited. It's repeatable, testable, shareable, and scalable to any cloud environment of any size. It can also be used to validate cloud environments on a frequent basis, which is necessary considering how fast we're changing them.

Just like programming languages express logical functions as code, policy-as-code allows you to express your required security posture as code. Programming languages use compilers and interpreters to provide developers with feedback on whether or not their code is functionally correct. Policy-as-code evaluations provide developers with feedback on whether or not their cloud infrastructure adheres to policy.

And when policy is expressed as code, everyone, from the security and compliance teams to the developers, are all on the same page speaking the same "language". And engineers can use policy-as-code across the software development life cycle (SDLC) by validating infrastructure at design time, in CI/CD when deploying infrastructure and changes, and for running production environments.

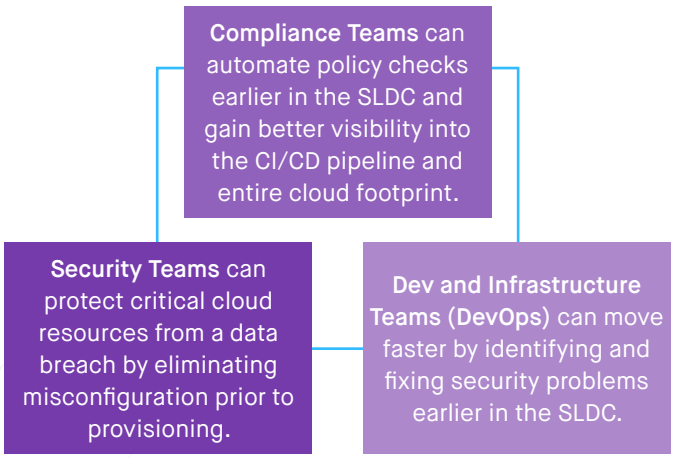
DevSecOps for cloud infrastructure

Just as DevOps is the application of software engineering to ops (to oversimplify here), DevSecOps is the application of software engineering to security. Policy-as-code is a key element of DevSecOps.

This means that the security team needs some development capabilities, or the engineering (i.e. DevOps) team needs to better understand cloud security and compliance in order to program it with policy-as-code. Delivering applications is now highly automated through CI/CD pipelines, infrastructure-as-code, and automated deployments into cloud. Security, therefore, has to be automated using policy-as-code to keep pace.

With DevSecOps for cloud security, teams can go fast and innovate, decreasing risk, and help bridge the traditional divide between teams.

Bridging the Divide with DevSecOps



Misconception: "We're not ready for policy-as-code because we haven't yet adopted infrastructure-as-code."

Reality: "Using infrastructure-as-code is not a prerequisite to using policy-as-code. In fact, you need policy-as-code because your teams are using a variety of methods to create and modify cloud infrastructure, and you need a common method of validating that the infrastructure they're building and managing doesn't violate policy."

Additional Resources

Open Policy Agent (OPA) is a popular open source standard for policy-as-code that can evaluate any JSON output, which makes it extremely powerful and flexible for a wide variety of simple to sophisticated cloud use cases. There's a robust OPA ecosystem and community and it's easier to find engineers that know OPA. You'll also avoid vendor lock-in and capability constraints that come with proprietary policy-as-code frameworks. Learn more about Open Policy Agent.

Regula is a tool that uses OPA to evaluate Terraform infrastructure-as-code. Learn more about Regula.

Managing the cloud misconfiguration problem

The high cost of managing cloud misconfiguration

When your team is attempting to manage dozens, if not hundreds, of cloud misconfigurations per day, it's easy to see why this problem has become such a burden for cloud engineering and security teams. The cost of managing cloud misconfiguration is almost entirely due to the engineering resources we invest in managing it manually, day in and day out.

But the cloud is purpose-built for automation. API-driven infrastructure, continuous integration, provisioning, instance configuration, and monitoring are among the many cloud-based tasks that have been automated. Show me a common manual repetitive task in the cloud, and I guarantee you it's been automated.

That is, with one glaring exception: cloud security. Cloud infrastructure needs to be provisioned according to a set of security or compliance policies, and once running, it must not drift out of compliance. Both of these tasks remain largely manual, slow, and error-prone. And it comes with a high cost.

Managing cloud misconfiguration typically involves the following mostly manual tasks:

- Review cloud misconfiguration alerts to identify critical events that require remediation (78% take five minutes or longer per incident)
- Remediate misconfigured resources (63% take 15 minutes or longer per incident)
- Producing reports on each misconfiguration (68% take 15 minutes or longer per incident)

Factor in the frequency of misconfiguration and we have an expensive game of whack-a-mole.

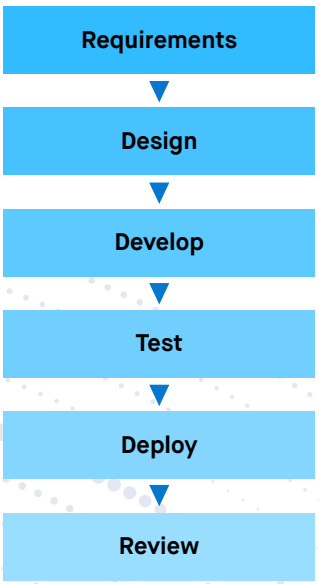
Hours per week invested in managing cloud misconfiguration

We've found a prime candidate for automation!

You can read more about the manual processes involved in The Day in the Life of a Cloud Misconfiguration.

The "shift left" on cloud security

In every software development lifecycle (SDLC), there are similar steps. They are sometimes described in Agile as:



These are a loop that iterates, and in agile methodology, they iterate quickly—perhaps every two or three weeks.

Shifting left is moving a function that used to take place only in later phases to earlier phases. The reason this is attractive is that as we move to the right through the lifecycle, we've committed more resources and time, so making changes becomes more expensive and slows down the process.

In the cases of compliance and security, they're often implemented as a gate during the test phase, and it's common for them to cause rework in design, development, and testing to continue due to problems found during security and compliance testing.

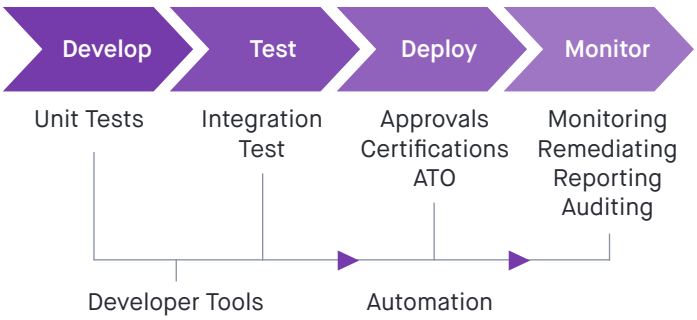
If we could shift cloud compliance and security into the design and develop phases, we'd go faster, make better systems, and turn those functions into highway builders rather than toll booth operators. To shift compliance and security left, they need to be automated and fit naturally into the development process.

A good place to start shifting compliance left is the development environments. To do this, you'll need tools that can scan dev environments for compliance issues and provide feedback to the developers on any issues.

Developers move really fast, so it's important that the tool be able to move with them. Daily or weekly scans/reports aren't good enough - the teams need the ability to run the tool themselves on whatever frequency they need.

This approach can be applied throughout the SDLC.

Shift left on cloud infrastructure security



1. **Development Phase:** Perform security unit tests in the dev/sandbox environments where you are building your infrastructure-as-code.
2. **Testing Phase:** Perform security integration tests in the test/stage environment to catch new security flaws that weren't found during unit testing.
3. **Production Phase:** Baseline the production system's infrastructure and constantly monitor for drift and enable auto-remediation for critical resources.

Misconception: "Validating infrastructure as code is sufficient for Shifting Left on cloud security."

Reality: Empowering engineers with tools to validate their infrastructure-as-code files is a good thing that will save a lot of time and headaches. But it's a mistake to assume this alone ensures secure cloud infrastructure.

First, It's rare to have a single infrastructure-as-code file define an entire cloud environment. Rather, they're often broken down into many, more manageable pieces. Second, IaC files don't include every configuration attribute for every cloud resource they define, and a lot of default configurations will result. Finally, IaC languages are dynamic and must be resolved at runtime.

The only way to truly know your infrastructure is secure and complies with policy is to validate the running cloud environment. Do this for dev and staging environments, before deploying changes to production.

Using automated remediation for cloud misconfiguration

A vast majority of cloud teams are remediating cloud misconfigurations manually when they're identified, but a few have adopted automated remediation. Attackers use automation to detect and exploit misconfigurations, often within minutes, and there's a growing recognition that automated remediation is necessary to protect security-critical resources and sensitive data.

But the adoption of security automation faces stiff resistance from application and ops teams, and for good reason. Automatically remediating security issues brings significant risk of business disruption due to sudden application downtime events and other breaking changes.

Effective security automation requires vigilance to address risks posed by false positives and target remediation states that differ from application requirements, which happens when different teams define them, or they drift over time as infrastructure evolves. Establishing trust is the biggest challenge in gaining buy in for automated remediation.

There are essentially two methods of automated remediation for cloud misconfiguration, but the implementation details of each can vary.

The first method uses automation scripts that are triggered by an alert event. When an automation script is triggered, it executes a configuration change via the cloud APIs. Typically these scripts or workflows leverage serverless functions, such as AWS Lambda or Azure Functions, to execute changes to the environment. Remediation actions change the configuration to a predefined "safe" configuration when triggered. These automation scripts are often referred to as bots, cloud bots, or "lambdas".

The second method enforces established infrastructure configuration baselines—"snapshots" of provisioned cloud resource configurations that serve as the target configuration state for remediation events. Baseline enforcement is also triggered by an event such as an alert, or when configuration drift is detected by subsequent snapshots. Remediation actions restore the resource to the established baseline configuration and don't require target configuration states to be predefined. This method is called configuration baselining and baseline enforcement, and it makes resource configurations effectively "self-healing" in the event of misconfiguration.

Automation scripts are easier to get started with, but they run the risk of destructive changes because target configurations can break the business function of the application. There's also maintenance burdens. A single script designed to remediate a specific AWS Security Group misconfiguration can exceed 300 lines of code or require bespoke customization using a workflow. Rinse and repeat for every misconfiguration event you need to protect against, and scale that to multiple environments, regions, and cloud accounts.

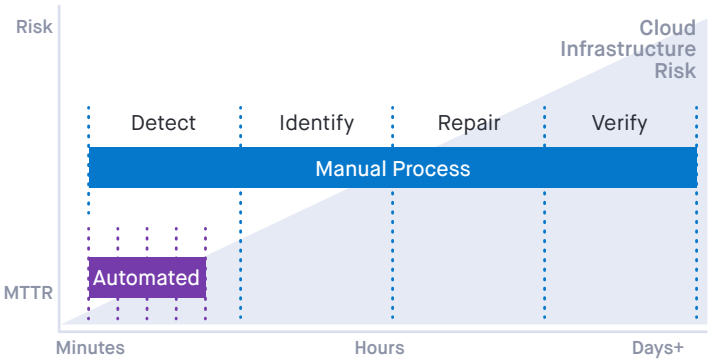
Baseline enforcement is more challenging to implement without third-party tooling, but it eliminates the risk of unintended destructive changes because the target configuration state for remediation is the established and provisioned configuration prior to misconfiguration. Because this method requires no development and maintenance of separate automation scripts, the operational burden is low.

Measuring the effectiveness of your cloud security (MTTR)

Mean Time to Remediation (MTTR) is the key security metric for measuring your management of cloud misconfiguration risk. Threats to cloud infrastructure are highly automated, constantly probing for attack

vectors to exploit. The longer cloud misconfigurations go unaddressed, the greater the risk of a major security incident.

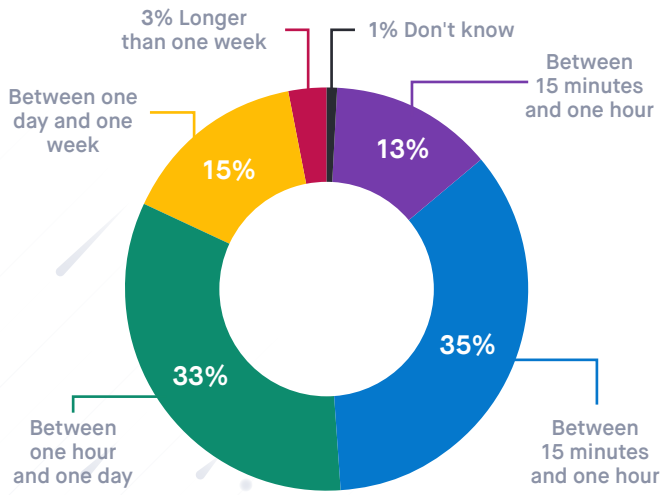
Manual vs. Automated



If you're manually remediating cloud misconfiguration, your MTTR is probably hours or days, which is unacceptably high for security-critical resources. And if you don't know what your MTTR is, that's not a good sign.

Most cloud teams have an MTTR for cloud misconfiguration that's beyond a safe level.

Mean time to remediation for cloud misconfiguration



To determine your MTTR, you need to identify who "owns" the management of cloud misconfiguration. Who's monitoring and identifying cloud risks? Who's responsible for remediating misconfigurations?

Document your process and understand how long it takes, on average, to go from the moment a misconfiguration occurs to the time it's been verified as remediated.

Then set an MTTR goal. It should be measured in minutes for security-critical resources. Are there inefficiencies and unnecessary delays in your process?

These are prime candidates for automated remediation.

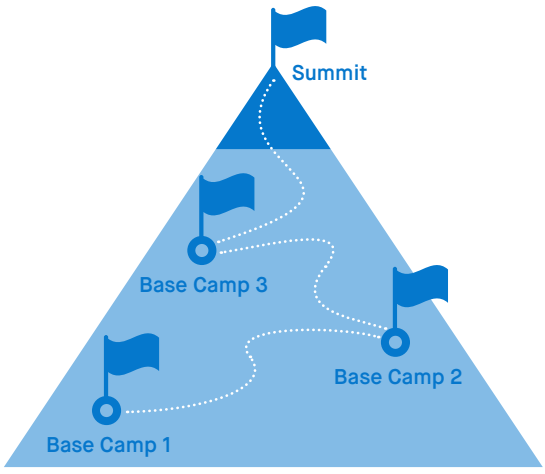
Climbing the cloud security mountain

Many enterprise-scale organizations already have enterprise-scale cloud environments running, and have for quite some time. And it's all configured differently today than it was yesterday.

Most likely can't account for 20% or more of what they have running in their cloud. Security teams and auditors are missing things. And the compliance standards they're applying aren't addressing some of the most dangerous vulnerabilities.

We've covered a lot in this handbook, and you can't do it all all at once. But there's a clear path for "climbing the cloud security mountain" that will get you to the top faster, and with fewer headaches. You should easily identify which base camp your team is in right off the bat. And it's quite common for different cloud teams at the same organization to be in different base camps.

Climbing the cloud security mountain



Base Camp 1: Discovery and posture reporting

To use the mountain climbing analogy, your first stop is Base Camp 1. This stage is all about discovering all of the resources you currently have running, identifying misconfiguration vulnerabilities, establishing your security compliance posture, and working to bring everything into compliance. When everyone who needs compliance reports is getting them automatically, (without manual audits), you're ready to head off to Base Camp 2.

Most cloud teams are at Base Camp 1 (or trying to get there). If you're new to the cloud, or need to secure an existing cloud environment and bring it into compliance, start here.

Base Camp 2: Configuration change control

Now that you've brought your cloud into compliance, your focus turns to making sure it stays that way in the face of constant change. You want to know when things change (i.e. detect for configuration drift), and whether that change violates policy or introduces misconfiguration vulnerabilities. When that happens, a ticket is created and these misconfigurations are remediated manually as quickly as possible. All of this is automatically included in your regular reports.

Many cloud teams are doing some of this, such as using a scan and alert tool to tell them about specific issues in their cloud environment. They may have a toe in Base Camp 2, they're still likely struggling to get into Base Camp 1.

Base Camp 3: Shift Left on cloud security

Once you've got systems in place to identify policy violations and misconfigurations when they occur in production environments, it's time to "Shift Left" and prevent these issues earlier in the software development life cycle (SDLC). Developers get tools to validate their infrastructure-as-code, and feedback that helps them fix these issues. Policy checks are performed in CI/CD pipelines to prevent issues from being introduced to production.

Many teams are in Base Camp 3 when it comes to application and OS security, but when it comes to cloud infrastructure security, few have made this "Shift Left".

Summit: The peak of cloud security

Teams at the summit are using automated remediation to fix critical cloud misconfiguration issues without the need for human intervention. They can prove cloud compliance at any time without having to drop whatever it is they're working on. Management and executives trust that their sensitive data is kept safe, and that this no longer slows application teams down. Cloud security and compliance is now a competitive advantage.

Very few cloud teams that have reached the summit.

About Fugue

Cloud Infrastructure Security & Compliance

Fugue is a Software-as-a-Service product that puts engineers in command of cloud infrastructure security.

Prove cloud security and compliance

Always know the security and compliance posture of your cloud environment and prove it to management and auditors at any time, all of the time.

Build security into cloud development

Empower developers to find and fix security issues in development environments and infrastructure-as-code --- before they deploy to production.

Stay safe by eliminating cloud misconfiguration

Detect configuration drift and automatically remediate unapproved changes made to security-critical resources --- without scripts or the risk of breaking changes.

Fugue supports CIS Foundations Benchmarks, CIS Controls/SANS Top 20,GDPR, HIPAA, ISO 27001, NIST 800-53, PCI, SOC 2 and your custom policies. The Fugue Best Practices Framework protects your cloud infrastructure against advanced misconfiguration attacks.

Fugue supports Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. Fugues's API can be used to integrate cloud seucrity in CI/CD pipelines.

Customers such as A T&T, SAP NS2, and A+E Networks trust Fugue to protect their cloud environments.

To learn more, visit www.fugue.co

Fugue Headquarters
47 East All Saints St.
Frederick, MD 21701
(844)-463-8483
hello@fugue.co