Cloud Multi-Account Policy Enforcement

GIAC (GSNA) Gold Certification

Author: Andy Huang, tokidoki2@gmail.com
Advisor: *Lenny Zeltser*

Accepted: *7/1/2021*

Abstract

Many enterprises have adopted a shift left approach for development and project implementation in the cloud. As part of this effort, many development teams have been given administrative powers to create, deploy and maintain cloud resources and access rights. This has created a multi-cloud account structure in many organizations, leaving each cloud account under the supervision and control of different teams. This can lead to accidental misconfigurations which leads to vulnerability exposures. This paper looks at the implementation of a central policy enforcement area to allow best practices in access control to be applied uniformly across the organization.

1. Centralized Development

1.1. Shift Left Model

The shift left movement began a few years ago with initiatives to move all aspects of a project into the development cycle. Business units and development teams are empowered to take control of product development and implementation. This approach is widespread in online products and software as a service offering, where a product is considered a web-based application that requires multiple layers of technology to function. In traditional development frameworks, each layer has a distinct engineering discipline to accomplish different tasks. For example, the system administration team provides systems and configurations for the applications to function on, and database teams maintain and provide database persistence layers for the data. The development team develops the application, and hands to the build team or QA team to test and deploy into the environment.

The introduction of a shift left model encourages the development team to take on many of the traditional roles of the implementation model. The development team can remove the transfer process to increase speed to market, given that they have full control of the cycle. This complete overall approach from the development team is made possible by cloud technologies, in which entire environments, including systems, network, storage, and databases, are provisioned via web user interfaces and API calls. Business units are encouraged to create multiple environments for testing, development and production. The development team is in charge of the entire development cycle of development, as well as testing, deployment and production maintenance.

1.2. Multiple cloud accounts

Since the development teams are encouraged to create and work in a dedicated environment, many have created multiple accounts to isolate themselves from other projects. An individual cloud account provides all services needed to launch a product, from networking to systems to databases. This isolation provides many advantages, including isolation and blast radius reduction from other projects. Many teams have created multiple accounts for a single project, choosing to additionally isolate

environments such as production, testing, development, and multiple end-to-end partner testing areas for one product. This approach results in a sprawl of accounts and topologies within an organization, each created based on interpretations and practices of differing teams.

One of the largest cloud services providers, Amazon Web Services, provided a well- architected model which proposes the DevOps model as the final operating model for many cloud projects. This model encourages multi-account creation due to its flexibility and other benefits such as isolation of critical data and cost control. Security control is also an advantage, as individual accounts limit security blast radius and aids in security control. (Optimizing AWS Environments, nd.) However, the ease with which any development team can create a cloud account means that a centralized security team is often not consulted before account creation. The central security team typically does not have the bandwidth to handle the number of requests to look at every account creation and run in danger of becoming a bottleneck. Many companies utilize pre-flight checks before cloud project implementation; however, these can become academic exercises where the development team proposes the structure for review to receive security approval. It is unknown and often untracked whether the team which created the environment followed through with the approved structure.

1.3. Loss of visibility

Each development team creating their own accounts and cloud environments leads to a loss of consistent policy and controls across accounts. The company now trusts that each development team has the knowledge and understands security policies when initializing an account. While many developer teams pursue best practices with the best intentions, it is not guaranteed that all development teams will ultimately create consistent accounts and environments that abide by all controls. The matter of identity and access controls can be particularly troublesome, as authentication and authorization controls and assignments usually take secondary importance given timelines and project success.

A distributed model of cloud account and environment creation also creates many blind spots for the company. The security team is ultimately responsible for controlling

access and is often called upon to implement compliance controls across the enterprise. Central teams that must ensure the proper identity of hundreds of accounts, with tens or hundreds of accounts created or destroyed daily, increases their overhead significantly. The timing of the implementation is also essential to ensure that there is no gap of time between environment creation and control implementation. The team needs to implement identity controls at account creation across a large distributed landscape of accounts and do it consistently and at scale.

2. Policy Enforcement

2.1. Singular accounts

There are many ways cloud providers provide security controls and enforcement mechanisms. In Amazon Web Services, a common practice is to utilize the Identity and Access Management module to maintain access rights for administrative or programmatic access. Many resources provide resource-based controls for additional controls; for example, S3 bucket policies provide fine-grained control to secure each bucket within an account. The Identity and Access policies work in conjunction with these resource policies to ensure proper authentication and authorization for changes. Therefore, identity policies provide the action and act on the user working with the bucket, while resource policies act on the S3 buckets themselves. (AWS IAM Users Guide, n.d)

Many companies turn to Cloud Security Posture Management or CPSM toolsets and vendors to assist in monitoring. These toolsets and third-party security tools come with a cost, often calculated per account or resources under monitoring or protection. When there are hundreds or even thousands of accounts to protect, the cost can be high. In addition, many tools monitor resource security policies and may not monitor or prevent lax identity policies. In the context of a shift left approach, the development team is often granted administrative access to create accounts for rapid development iteration, and administrators can freely create users with additional policies to change permissions on resources.

2.2. Organization best practices

In order to maintain consistency and standard policies across accounts, policy controls should be implemented in a broad base manner from a centralized area; however, the implementation should be scalable so that best practices are not dependent on individual account teams to implement. A development team can also easily misconfigure resources since they are not the subject matter expert in the cloud security area. Administrative access gives the users the ability to disable controls while troubleshooting, leading to accidental exposure.

Two of the more common misconfigurations involve inbound IP access control and S3 bucket configurations. Internal back-office applications are maintained in many companies and are accessible only by internal resources and IP address ranges. Due to the public nature of AWS, an account is inherently in the Amazon cloud provider space and not an on-premise resource; any interface is usually accessed via an internet IP address space. A common practice is to secure inbound access by restricting the inbound IP ranges to a set of IP ranges exposed by the company, such as the outbound IPs of the Virtual Private Network devices or proxy systems. The ability to define and implement a public inbound group is vital to secure this perimeter.

S3 bucket configurations have also been troublesome with security policies and enforcement. Publically exposed S3 buckets continue to be an issue in the industry. Compliance rules and laws may also mandate strict policies on S3 buckets, disallowing specific actions to be taken on data and the storage area bucket itself. A data evidence application may have strict policies and approval procedures around the destruction of data or the bucket in the account. In a recent survey of cloud misconfiguration incidents, broad identity access permissions and object storage breaches continue to be a significant source of problems. (Fugue Cloud Security Report, 2020)

In the shift left development and production support model, the account administrators may overwrite security and compliance configuration policies. Organizational control over these policies would allow the central team to specify and enforce policies across accounts, preventing accidental or malicious efforts to overcome corporate security practices and enforcement.

3. Centralized Options

Amazon Web Services introduced the concept of Organizations to address the maintenance and billing of multi-account implementations within an enterprise. This concept introduced centralized billing and enforcement of policy controls across the company. Other major cloud providers also introduced similar concepts, as multi-account and isolation become standard practices. In Microsoft Azure, this concept is called "Management Groups", while in Google Cloud, it is also called "Organizations". This paper will focus on AWS Organizations only. Since security policy enforcement is critical to protecting the company, the paper focuses on AWS Organizations' Service Control Policies, a part of AWS Organizations framework. It allows for overall policy implementation and identity policy enforcement across the business.

3.1. AWS Organizations

AWS Organizations organizes a company or enterprise in a tree root structure, with multiple organization units under the root and accounts under each organization unit. This setup allows for more fine-grain billing and visibility from the root level. For example, a centralized cloud or security team can adjust various policy controls and templates and match particular accounts within an organization unit. Policy assignments apply to a node in the hierarchy and flow down into the branch and leaves beneath it.

Organization concepts address the previous issue of lack of centralized control over multi-account spawn and run away policy enforcement gaps. In addition, since this is a natively offered service from AWS, it reduces costs by not relying on third-party tools for policy enforcement. Many of the organizations utilizing AWS likely have Organizations implemented due to the ability to do centralized cost and billing management. The need for an enterprise to break down and charge each business unit their share of AWS costs likely means that it already has accounts and business units arranged in organizational units to take advantage of policy enforcement.

Since policy enforcement overlaps the responsibilities between a cloud engineering team, security team, and compliance teams, one team may have to take the lead in defining and implementing policy controls across the enterprise. Such collaboration

would be easier to maintain than coordination between development teams and different contacts and policy sources.

3.2. Service Control Policies

Service Control Policies within the AWS Organization framework allow for maximum actions a user can take within an account. Interestingly, these policies set limits on the actions that any account administrator can delegate to users and roles in the account. However, Service Control Policies do not affect resource-based policies; they control a user's actions rather than resource permissions. This configuration is similar to Identity and Management policies within each account. Furthermore, service control policies do not grant permissions to any particular users; instead, they restrict and guard against maximum actions a user can enact. (AWS Org Service Control Policies n.d) Controlling specific actions across an organization enforces and maintains the secure state, preventing unintended changes.

4. Analysis and Experimentation

4.1. Policy enforcement and overrides

To validate the ability to control actions centrally utilizing AWS Organizations and gauge the effectiveness of preventing account administrators from provisioning extra permissions to user three accounts in AWS was created. Two of the accounts will serve as business unit organizational accounts. The primary account serves as the organization management account and is where the organization is formed and adjusted. One of the organizational accounts is added into an organizational unit while the other is not.

Three use cases will be used as part of the experiment, a policy for denying the delete of AWS S3 buckets, a policy to control any user or groups that can modify bucket policies, and a policy to limit the adjustment of security groups. Each use case represents scenarios where compliance team and security may have to lock down access to users to control their actions. The policy should override administrator permission which allows the adjustment of those actions. The use cases represent actions that a centralized team may need to implement to prevent accidental granting of permissions that may be too wide for the user base.

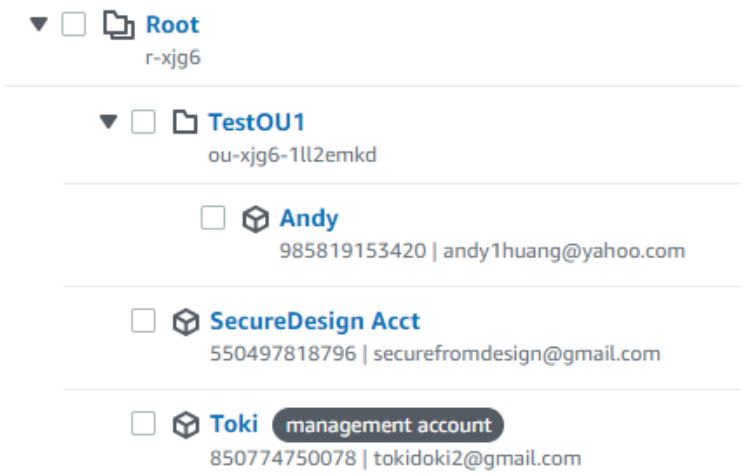
The effectiveness of organizational enforcement of the policies at account set up and whether an account administrator can override organizational policies will be evaluated to see if policy enforcement is reactive or proactive. Any gaps in enforcement time frames will be noted since a delay in enforcement will provide a vulnerability gap.

4.2. Organization and Account

An account can be created from the organization page, or an account can be retroactively added to the organization. After the account is established, the organization administrators can add the account to the proper organization unit in order to implement any policies. This area can potentially dissuade an account from joining the organization; however, an enterprise can mandate this course of action for any account to be paid for by the company. In addition, further service control policies can be added to deny an account the ability to leave the Organization. Service Control Policies are built independently of accounts and can be applied to any particular account or particular organizational unit.

Figure 1

Management account Organization Unit Structure



Note. Test OU is an organization unit, and contains accounts.

4.3. S3 bucket delete deny

In figure 2, this policy is applied to the account to disallow the delete action of S3 buckets that are created within the account. This action was initiated as a Service Control Policy in the Organization. As part of compliance requirements, buckets with evidence and other data are prohibited from being deleted. This policy could also serve as protection against accidental deletion of the bucket from the account in the production area. The policy is very similar to an Identify and Management policy but set to deny state for all resources applied to all accounts.

Figure 2

IAM policy to disallow the ability to delete a S3 bucket

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3Delete",
      "Effect": "Deny",
      "Action": [
        "s3:DeleteBucket"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

When a bucket is deleted, either programmatically or via the console, an Access Denied message is immediately presented, as follows:

Figure 3

Access Denied response during a bucket remove command

```
aws s3 rb s3://s3-scp-policytest2-ah1

remove_bucket failed: s3://s3-scp-policytest2-ah1 An error occurred
(AccessDenied) when calling the DeleteBucket operation: Access
Denied
```

Since Service Control Policies cannot maintain resource policies, it cannot switch a S3 bucket from public access to blocking public access. Access control of particular resource buckets is part of the resource policy mechanism. Other audit tools will have to be implemented in order to audit whether a bucket has been allowed public access or not. In this and other use cases, the Service Control Policy acts as global identity policy control to limit actions that a user can do for the resource. In the delete bucket example, this policy enforced the deny delete bucket action against anyone attempting to delete a bucket via the AWS console or programmatic access. When the option to delete is disallowed, the error that is produced does not specifically cite why the action is denied, this leads to user confusion. However, it is clear that the deny action can be applied across multiple accounts or perhaps all accounts to enforce the deny delete action.

4.4. S3 bucket Public Access

The following use case shows how the modification action of buckets permission access can be locked down to a particular user. While this use case is keyed on S3 buckets, the ability to control and limit actions to groups or roles can be applied to other resources and functionalities. One of the common issues with buckets is when they are accidentally exposed or switched to public access. There may be many reasons why a bucket is made public; however, buckets created for private access must remain private in the enterprise. In a multi-account scenario utilizing Service Control Policies, a global policy can be enacted to restrict the changing of the bucket to public access. From a compliance perspective, there may be only a single user or a single group of users that are allowed to initiate or make changes to bucket public access policies. This requires two necessary groups to approve the change and provide additional checks and balances for this configuration action.

The user can be further restricted to having only programmatic access, thereby disallowing anyone logging in as the user on the console from making these changes. The user can also be given only specific S3 roles to limit the ability to interact only with S3 buckets. In this example, the s3ControlUser user is granted s3admin access from an organizational standpoint. The Service Control Policy grants explicit permission to this user to control public/private permission access. Any other users within an account, even

if they have s3admin role, cannot adjust this particular permission. Therefore, in this case, AWS Service Control Policy provides the ability to restrict specific actions and control to a particular group of users. This policy can be applied across the accounts that require additional approval for the management of the public access block for the S3 bucket.

Figure 3

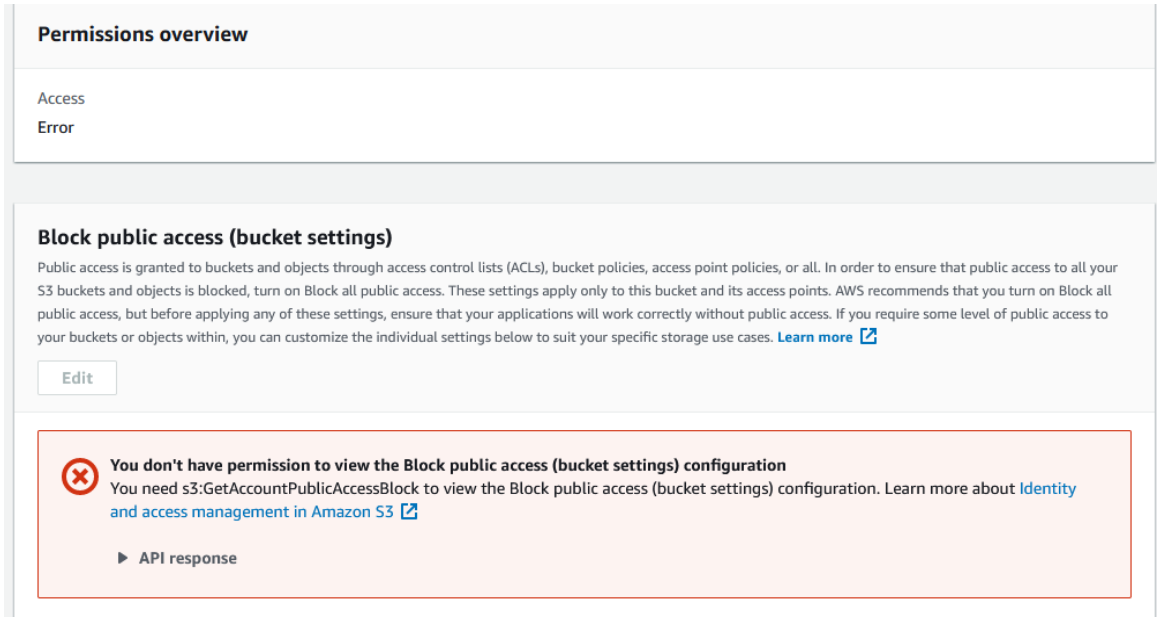
Policy code restricting policy control to one user.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictS3PublicAccessModify",
      "Effect": "Deny",
      "Action": [
        "s3:PutBucketPublicAccessBlock",
        "s3:GetBucketPublicAccessBlock"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:username": [
            "s3ControlUser"
          ]
        }
      }
    }
  ]
}
```

The policy above applies a Deny action to all users and restricts the ability to change or read the Public Access area of any buckets. The action is immediate, resulting in error messages when a user, even one with account root user access, can view the permissions of a S3 bucket. This result can be seen in Figure 4 below from the AWS console interface.

Figure 4

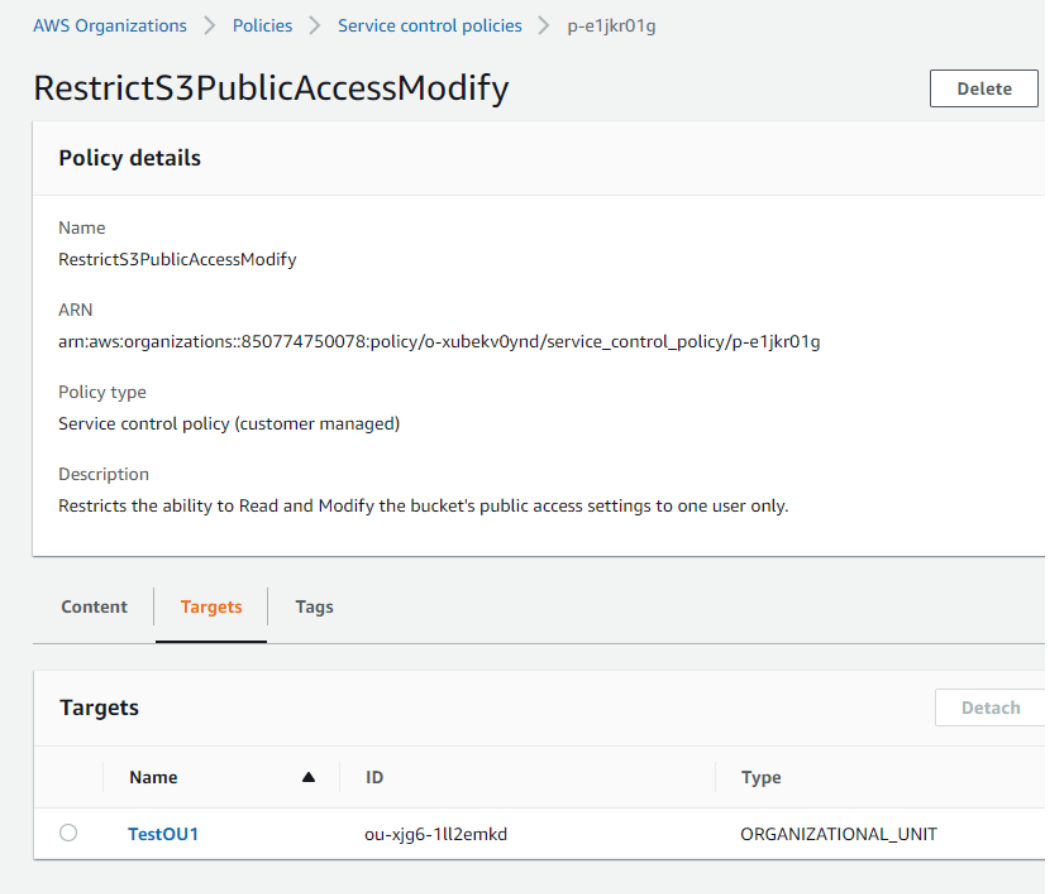
AWS console response denying configuration



In this case, both PutBucketPublicAccessBlock and GetBucketAccessBlock controls whether a bucket can be publically accessible. Therefore once permission is set at bucket creation, it remains in place guaranteed by the organization Service Control Policy's ability to prevent any changes to the permission. This ability prevents a drift situation where a bucket is created and later "drifts" from private access to the public. It also allows proper audit of changes, given that only a specific group of authorized or adjudicated users across the organization are allowed to make changes. The policy is implemented on the centralized management account of the organization and is hence outside of the account where the bucket is located.

Figure 5

AWS Console Organization SCP restricting abilty to change S3 public access



Many cloud monitoring tools react to changes in the security posture of cloud resources. In this case, the change is proactively prevented from happening. While this may seem heavy-handed from an enterprise standpoint, certain highly security-sensitive items, such as publically accessible S3 buckets may warrant this overall proactive deny measure.

4.5. Security Group Ingress restriction

Due to privacy laws, many countries have enacted laws and regulations specifying where access can be initiated to work on or maintain the hosting environment. For example, regulations such as GDPR restrict data and tasks to be performed by EU citizens in European Union locations. Similarly, United States tax regulations require customer consent to process US tax data beyond US shores ((Bhikha et al., 2009).

Therefore, a firm hiring many certified tax accountants may choose to enforce these legal restrictions where the accountants work. To abide by these regulations, access is typically granted to certain source IP addresses within geographic locations. Virtual Private Networks are also used to allow preliminary checks of endpoints to ensure they are located within a specific area. In an AWS organization, an organizational unit can encompass all accounts that need to abide by this type of regulation or rule. A global policy can then be set up to restrict access or actions to specific known IP addresses or subnet ranges, such as those stemming from a Virtual Private Network connection. This setup ensures that teams accessing the resources are logged into a VPN connection or using proper endpoints such as bastion jump systems to access and maintain the resources in the account. Further security controls can be adjusted and maintained upstream with endpoint security controls on the jump point to ensure the endpoints are adequately protected. A Service Control Policy that restricts IP or IP address ranges from tampering or accessing a particular account can be implemented below in the Organization.

Figure 6

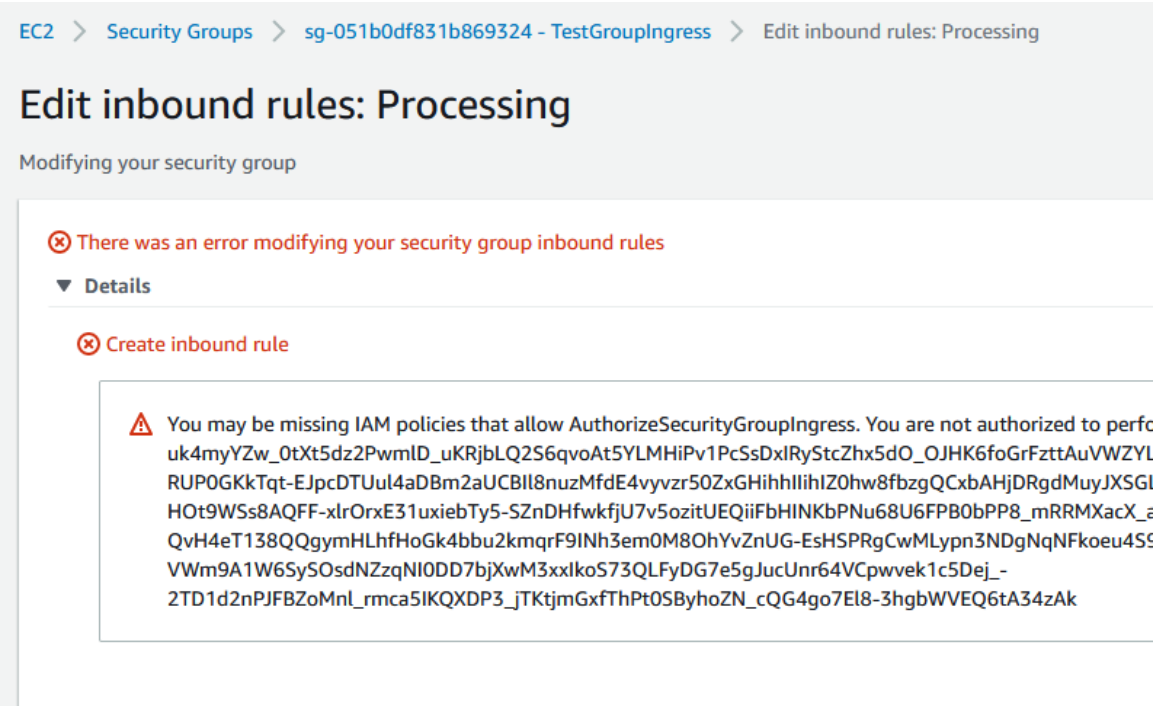
Policy code to restrict entry to one IP address

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictEC2SGIngressModify",
      "Effect": "Deny",
      "Action": [
        "ec2:AuthorizeSecurityGroupIngress"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "75.80.183.86/32"
          ]
        }
      }
    }
  ]
}
```

In the example, the policy restricts changes to the security groups, which defines and allows ingress ports and IP ranges within the account. The Source IP condition mandates that only systems with the Source IP address are allowed to modify the security group for ingress controls. While this does not restrict an "everywhere" ingress rule from being implemented, it provides protection on where the change or the addition can be done. Additional conditions can be used to restrict user groups or user roles to provide additional controls and processes. For example, any source IP outside of this particular IP address, 75.80.183.86, would be prohibited from creating or modifying any ingress security groups.

Figure 7

AWS console response showing deny in attempt to modify inbound rules for a Security Group



This policy control helps the centralized and compliance team maintain security policies and best practices and override any potential changes that can happen after an account is live in production. This use case shows that while organization Service Control Policy cannot control resource policy of a security group, any actions taken by any user or any endpoint on the policy can be restricted. Like controlling S3 bucket modifications,

modifying inbound IP security groups can be tracked, controlled, limited to particular groups or endpoints.

4.6. Drift Detection and Organization Enforcement

Since AWS Organization Service Control Policies control and enforce actions rather than the attributes of the resources, there would be no concern due to drift of attribute. From a drift detection standpoint, once a policy is implemented, the actions prohibit and prevent any changes. When implemented, these policies prevent drift from occurring in the first place. Manual changes of policies and accounts should be done at the top- level organization maintenance account level, by an approved team. This provides secondary checks and balances and can be set within a process that runs through proper change management control. Additionally, Service Control Policy prevents the drift in critical processes by denying permissions to those actions that can cause drifting. A policy to deny leaving the organization can also be implemented to prevent development teams from leaving the organization from the account. Any changes to policies and account membership would needed to be discussed with the cloud or security team, allowing discussion and promoting cross-check. The policy below declares that no resources can initiate a leave organization action, and therefore any attempts to leave the Organization would be met with an access denied error.

Figure 8

Policy code disallowing an account to leave th Organization and denied response

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyLeaveOrg",
      "Effect": "Deny",
      "Action": [
        "organizations:LeaveOrganization"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```
> aws organizations leave-organization

An error occurred (AccessDeniedException) when calling the
LeaveOrganization operation: You don't have permissions to access
this resource.
```

The above examples show how account Identity and Access Management policies can be enforced globally across multiple accounts and categorized into different organizational units. The action control perspective differs from resource controls which cloud security posture management tools and vendors may be utilizing. However, given that there may be gaps in timing with those tools, this global approach provides additional protection to those controls. Auditing and scanning are reactive tools in that they actively detect and identify misconfigurations after the fact; conversely, service Control Policies are proactive tools, allowing the enterprise to initiate and define best practices before and guard against accidental deviations from those practices.

WS also provides the AWS Control Tower service within the AWS Organizations framework, which combines AWS organizations, single sign on and service catalog to fully manage all accounts and organization units within AWS. The guardrails are written at higher level language and rule set and uses an interpretative engine to define preventive and detective controls. AWS Control Tower will re-interpret rules via a rules engine into Service Control Policies to implement within the organization.

5. Considerations

5.1. Proactive Controls

In a distributed environment, where different teams create and roll accounts to accomplish development goals, guard rails are critical in order to prevent accidental misconfiguration of any aspect of the account. Organizational application of security controls allows teams to control many aspects of their account while enforcing general company best practices and controls. Identity and Access Management misconfigurations accounted for 40% of the misconfiguration types in a recent report from Fugue. (Fugue Cloud Security Report, 2020) Some development teams are not well versed in identity

and permissions setup and can easily grant a higher level of access to particular users or roles. With the type of access, resource protections can be overridden without full understanding of the impact of the change. Even with toolsets that audit and detect changes, those actions are reactive controls. With Service Control Policies from AWS Organizations, the policies become proactive controls to deny and limit those actions.

5.2. Planning

The power to roll out global policies can be detrimental without careful planning; development projects that legitimately require interfacing with outside customers on the internet cannot be placed in an organizational unit where inbound access is restricted. For example, a project with multiple internet partners, for example, internal supply chain networks with multiple partners, cannot be in an organizational unit where the project teams cannot adjust external inbound IP ranges. The centralized teams in charge of the organization consider when setting up organizational units and policy applications. The amount of process and procedures also needs to be monitored, otherwise, there can be a risk of overload and bottleneck situations. Like all centralized controlled resources and teams, balance is needed in a shift left environment to maintain a sense of optimization and freedom while enforcing secure practices.

6. Conclusion

The arrival of shift left methodology to product and project development has yield faster time to market for crops and projects. Unfortunately, this is done at the expense of specialization and a deep understanding of particular areas in the technology stack. While understanding is usually not needed to launch a product or project, knowing nuisances can avoid misconfigurations and enforce best practices. Given that a development team will never be able to understand all the nuisances of the entire technology stack, having established guard rails to prevent them from accidentally misconfiguring resources would be invaluable.

Identity and Access Management provides such a mechanism, but the guard rail implementations can be inconsistent with many independent accounts spanning across the enterprise. In addition, compliance and regulation may dictate certain practices that

cannot be overlooked. Therefore an organization with a central area of enforcement and control is essential. AWS organization's Service Control Policy mechanism provides the ability to spread and enforce standard best practices in access management across the enterprise into many isolated accounts.

References

Optimizing AWS Environments. (n.d). Optimizing Your AWS Environment using Multiple Accounts, AWS Whitepaper. Retrieved May 10, 2021, from https://docs.aws.amazon.com/whitepapers/latest/organizing-your-aws-environment/organizing-your-aws-environment.pdf#organizing-your-aws-environment

AWS IAM User Guide. (n.d) AWS Identity and Access Management, User Guide. Retrieved May 10, 20201, from https://docs.aws.amazon.com/IAM/latest/UserGuide/iam-ug.pdf#introduction

Cloud Custodian Support Guide. (n.d) Cloud Custodian Lambda Support. Retrieved May 10, 2021, from https://cloudcustodian.io/docs/aws/policy/lambda.html

AWS Org (n.d). AWS Organizations Features. Retrieved May 10, 2021, from https://aws.amazon.com/organizations/features/

AWS Org Service Control Policies (n.d) Service Control Policies. Retrieved May 10, 2021 from https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scps.html

Fugue Cloud Security Report (2020) The State of Cloud Security 2020. Retrieved May 10, 2021, from The State of Cloud Security 2020 Report: Understanding Misconfiguration Risk (fugue.co)

Bhikha, N., Eldridge, P., Dolan, M., Snow, D., Michnay, R., & Miller, J. (2009, May 1). *New Tax Preparer Rules for Disclosure and Use of Return Information*. The Tax Adviser. https://www.thetaxadviser.com/issues/2009/may/newtaxpreparerrulesfordisclosureanduseofreturninformation.html.