A Holistic Approach to AWS Organization Management

Insomnihack, 2024

nexthink

Our Mission: To Delight People at Work

We are a software company focused on helping IT to shape smart and productive workplaces. We bring clarity to your IT department through a unique combination of real-time analytics, automations, and employee feedback. We think IT is an ocean of untapped potential, they just need the right solutions. And that's where we come in.

1000+ Nexthinkers (11 countries)

60 Nationalities

400+ R&D

1000+ Clients

15+ Million Cloud Endpoints

10 AWS Regions

Processing Trillions of Events Daily

nexthink

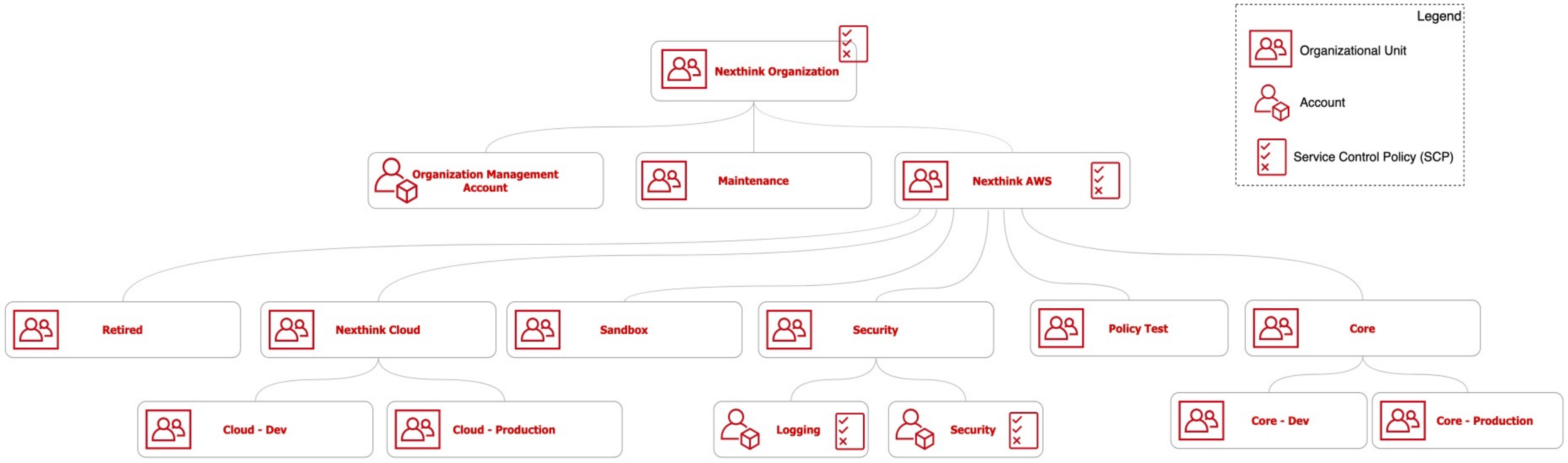
aws sts get-caller-identity

- Bogdan Nicorici (@0xboogy)
- Ex
  - UNIX / Linux Sysadmin
  - Penetration tester
  - CTF Player
- Cloud Security Architect @ Nexthink
- Security enthusiast
- Automation addict
- I love writing code & building security tools

Agenda

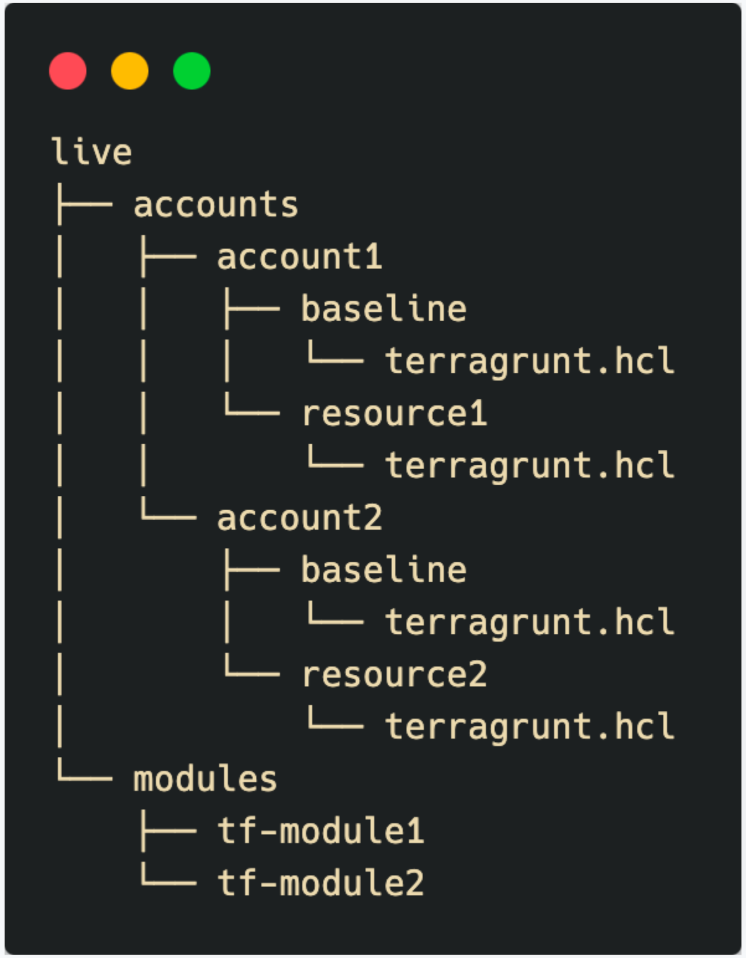
- AWS Organization
  - Structure
  - Services
  - SCPs
- Access and User Management
  - Permanent & Temporary
  - IaC
- Log Centralization
- Backups (Ransomware)
- Configuration Hardening
- Takeaways

AWS Organization



- Fully managed using IaC (terraform + terragrunt)
  - Fast and reproducible deployments, easy roll back, code review and approval for changes, etc.
- 70+ AWS accounts
- GitHub Repository owned by Security (engineering teams can contribute)
- Most Critical Account – Organization Management Account
  - Block all access to this account
  - Delegate all the services to dedicated accounts, ex: Security, etc.
  - Default administrator role created in each member account: OrganizationAccountAccessRole

AWS Organization



```
live
├── accounts
|   ├── account1
|   |   ├── baseline
|   |   |   └── terragrunt.hcl
|   |   └── resource1
|   |       └── terragrunt.hcl
|   └── account2
|       ├── baseline
|       |   └── terragrunt.hcl
|       └── resource2
|           └── terragrunt.hcl
└── modules
    ├── tf-module1
    └── tf-module2
```

- A folder per account and per component
- Each component has its own terraform state
- Terragrunt can manage dependencies
- Possible to apply all resources for an account in parallel
- Fast deployment ("smaller plans")
- Limit blast radius

AWS Organization

- A folder per account and per component
- Each component has its own terraform state
- Terragrunt can manage dependencies
- Possible to apply all resources for an account in parallel
- Fast deployment (“smaller plans”)
- Limit blast radius

Security Team deploys only the following resources in the organization:

- IAM Users
- IAM Roles (cross-account, IdP (OIDC), integrations)
- Account baseline (hardening, global configuration)
- IP restriction policies, permission boundaries, etc.
- SCPs

AWS Organization - Services

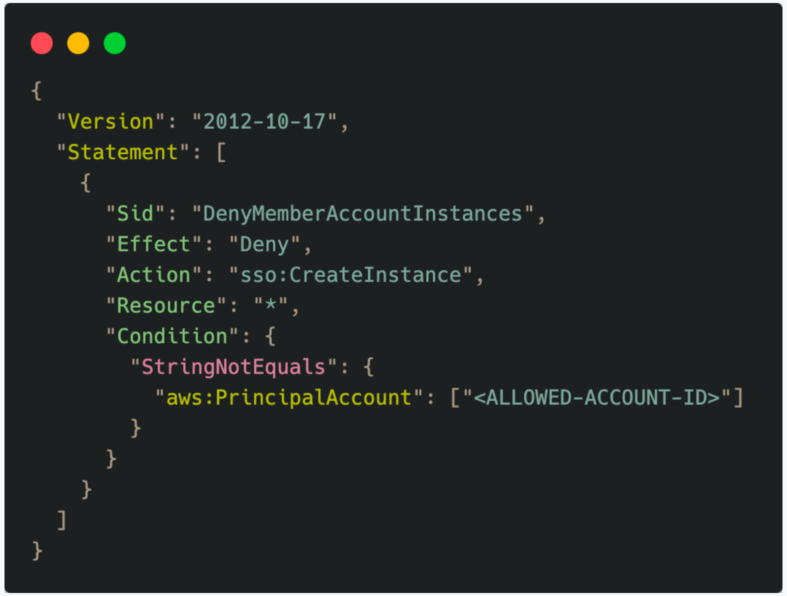
- Delegated Administrator
  - CloudTrail
  - GuardDuty
  - Security Hub
  - Detective
  - IAM Access Analyzer

AWS services that you can use with AWS Organizations

AWS Organization- Services

- Delegated Administrator
  - CloudTrail
  - GuardDuty
  - Security Hub
  - Detective
  - IAM Access Analyzer

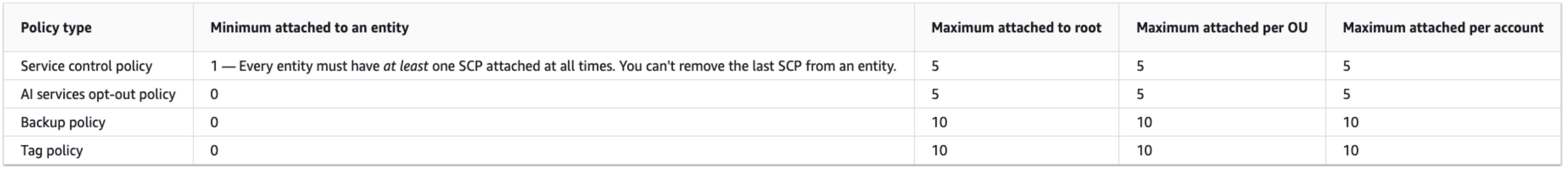
- IAM Identity Center (SSO)
  - One organization instance possible
  - Regional service
  - Possible to have an instance per account
  - Administrator delegation is possible
  - Use SCP to control instance creation



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyMemberAccountInstances",
      "Effect": "Deny",
      "Action": "sso:CreateInstance",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": ["<ALLOWED-ACCOUNT-ID>"]
        }
      }
    }
  ]
}
```

AWS Organization SCPs

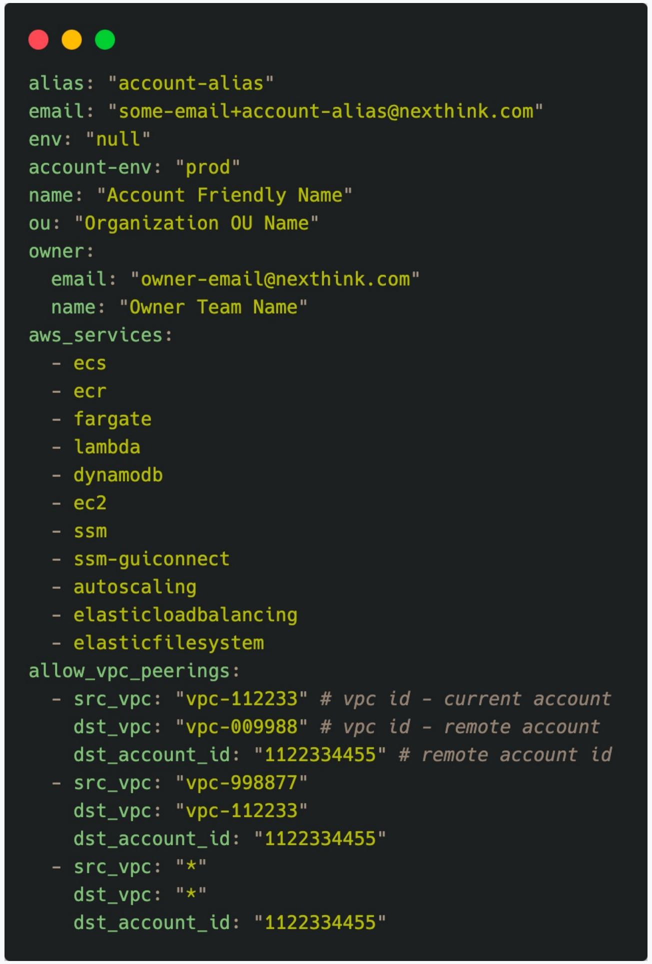
- What is an SCP 🤔
  - Think of a GPO for active directory applied to an OU or an AD object
- Globally applied SCPs
  - Block root account usage
  - Block creation of root access keys
  - Block dangerous actions
    - Account leaves organization
    - Create IAM users and keys outside of `IAM Bastion` account
    - Remove S3 public access block
    - Prevent disabling/deleting specific services, components
  - Ensure sensitive IAM roles cannot be tampered with
  - Ensure Lambda function URL requires IAM authentication
  - Region allow-list per environment
- Be mindful about SCP quotas



Policy type	Minimum attached to an entity	Maximum attached to root	Maximum attached per OU	Maximum attached per account
Service control policy	1 — Every entity must have *at least* one SCP attached at all times. You can't remove the last SCP from an entity.	5	5	5
AI services opt-out policy	0	5	5	5
Backup policy	0	10	10	10
Tag policy	0	10	10	10

Organization SCPs

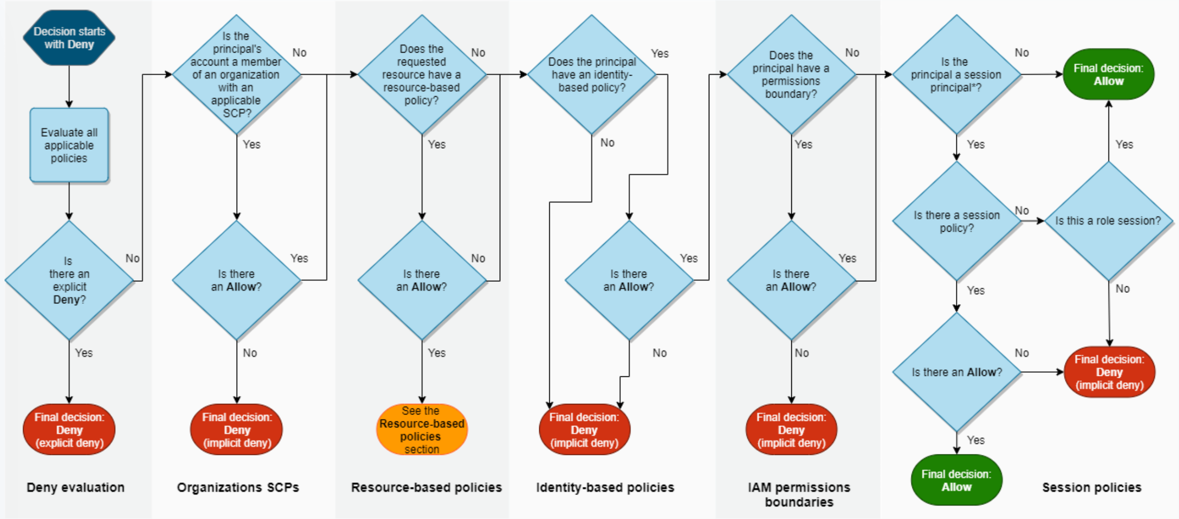
- Per account
  - AWS service allow-list
  - VPC peering allow-list
- Services per account
  - Useful for hardening and, compliance is a bonus
  - Reduces attack surface and cost
  - Can be tricky when services have dependencies
    - Ex: if you only want to use “ec2:*”
      - ec2messages:*
      - autoscaling:*
      - imagebuilder:*
      - ec2-instance-connect:*
- Configuration abstraction as yaml
  - Easy to read, update and understand
  - Single pane of glass for account configuration
  - Easier to track and understand changes
  - Easy to parse (CLI: **yq = jq but for yaml**)



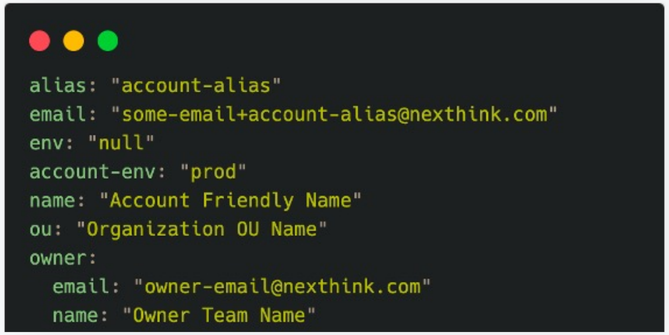
```yaml
alias: "account-alias"
email: "some-email+account-alias@nexthink.com"
env: "null"
account-env: "prod"
name: "Account Friendly Name"
ou: "Organization OU Name"
owner:
  email: "owner-email@nexthink.com"
  name: "Owner Team Name"
aws_services:
  - ecs
  - ecr
  - fargate
  - lambda
  - dynamodb
  - ec2
  - ssm
  - ssm-guiconnect
  - autoscaling
  - elasticloadbalancing
  - elasticfilesystem
allow_vpc_peerings:
  - src_vpc: "vpc-112233" # vpc id - current account
    dst_vpc: "vpc-009988" # vpc id - remote account
    dst_account_id: "1122334455" # remote account id
  - src_vpc: "vpc-998877"
    dst_vpc: "vpc-112233"
    dst_account_id: "1122334455"
  - src_vpc: "*"
    dst_vpc: "*"
    dst_account_id: "1122334455"
```

Organization SCPs

- SCPs – Deny vs Allow
- Tag accounts
  - security alerts (SIEM, lookup tables…)
  - billing
  - criticality based on environment, owner, workloads


Decision starts with Deny
Evaluate all applicable policies
Is there an explicit Deny?
No
Yes
Final decision: Deny (explicit deny)
Deny evaluation
Is the principal's account a member of an organization with an applicable SCP?
No
Yes
Is there an Allow?
Yes
No
Final decision: Deny (implicit deny)
Organizations SCPs
Does the requested resource have a resource-based policy?
No
Yes
Is there an Allow?
No
Yes
See the Resource-based policies section
Resource-based policies
Does the principal have an identity-based policy?
Yes
No
Is there an Allow?
Yes
No
Final decision: Deny (implicit deny)
Identity-based policies
Does the principal have a permissions boundary?
No
Yes
Is there an Allow?
Yes
No
Final decision: Deny (implicit deny)
IAM permissions boundaries
Is the principal a session principal*?
No
Yes
Final decision: Allow
Is there a session policy?
No
Yes
Is this a role session?
Yes
No
Is there an Allow?
No
Yes
Final decision: Deny (implicit deny)
Final decision: Allow
Session policies

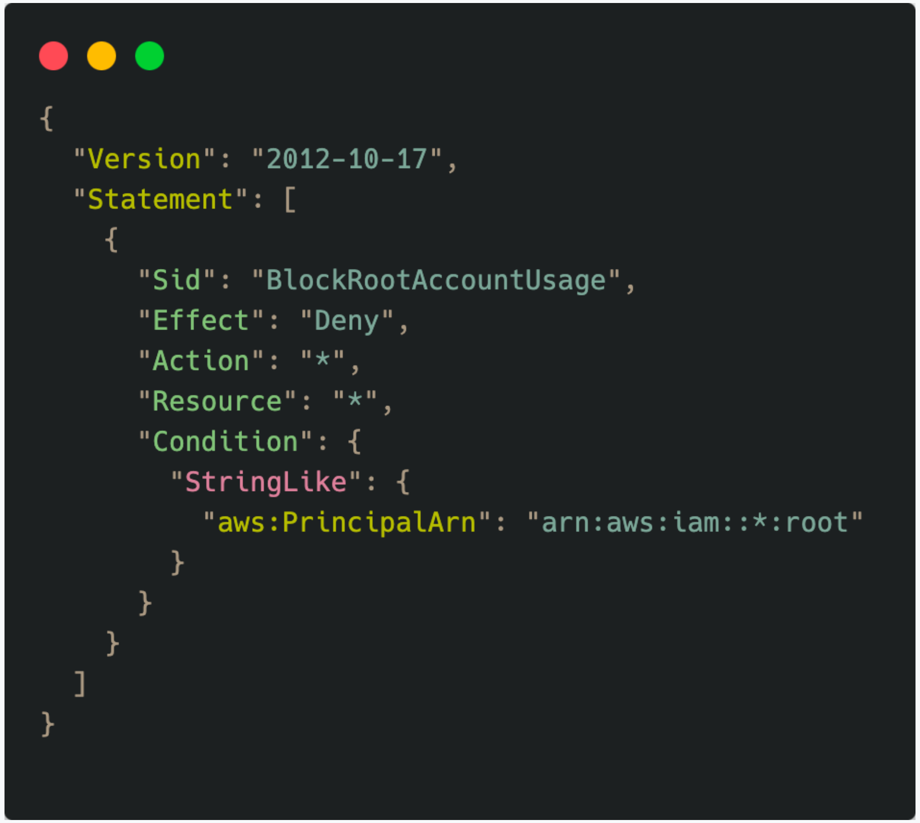
Source: https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html#policy-eval-denyallow



```
alias: "account-alias"
email: "some-email+account-alias@nexthink.com"
env: "null"
account-env: "prod"
name: "Account Friendly Name"
ou: "Organization OU Name"
owner:
  email: "owner-email@nexthink.com"
  name: "Owner Team Name"
```

Organization SCPs

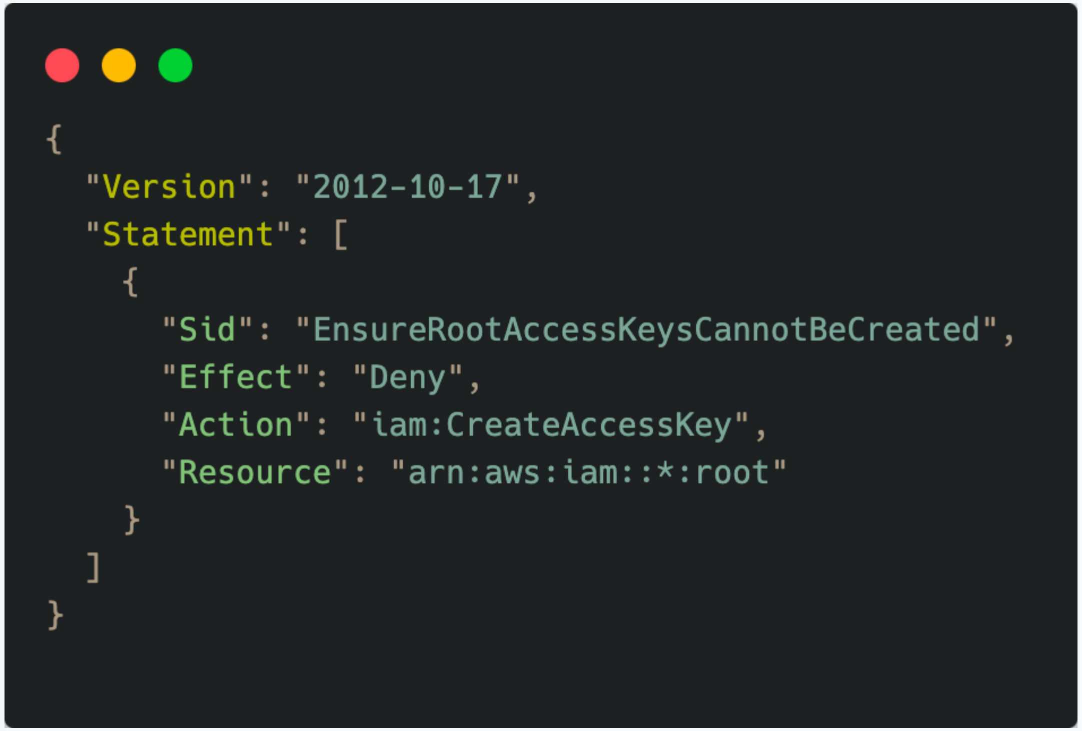
- Block root account usage



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BlockRootAccountUsage",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:root"
        }
      }
    }
  ]
}
```

Organization SCPs

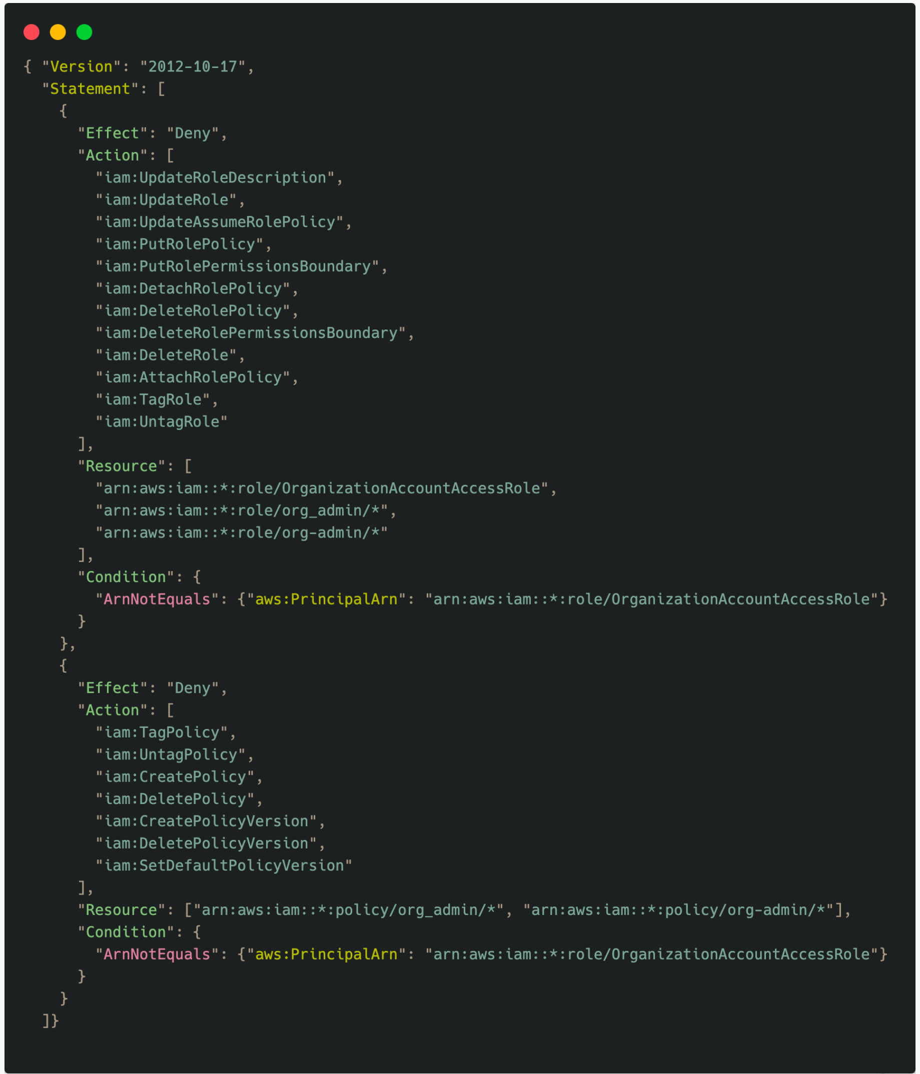
- Block root account usage
- Block root key creation



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnsureRootAccessKeysCannotBeCreated",
      "Effect": "Deny",
      "Action": "iam:CreateAccessKey",
      "Resource": "arn:aws:iam::*:root"
    }
  ]
}
```

Organization SCPs

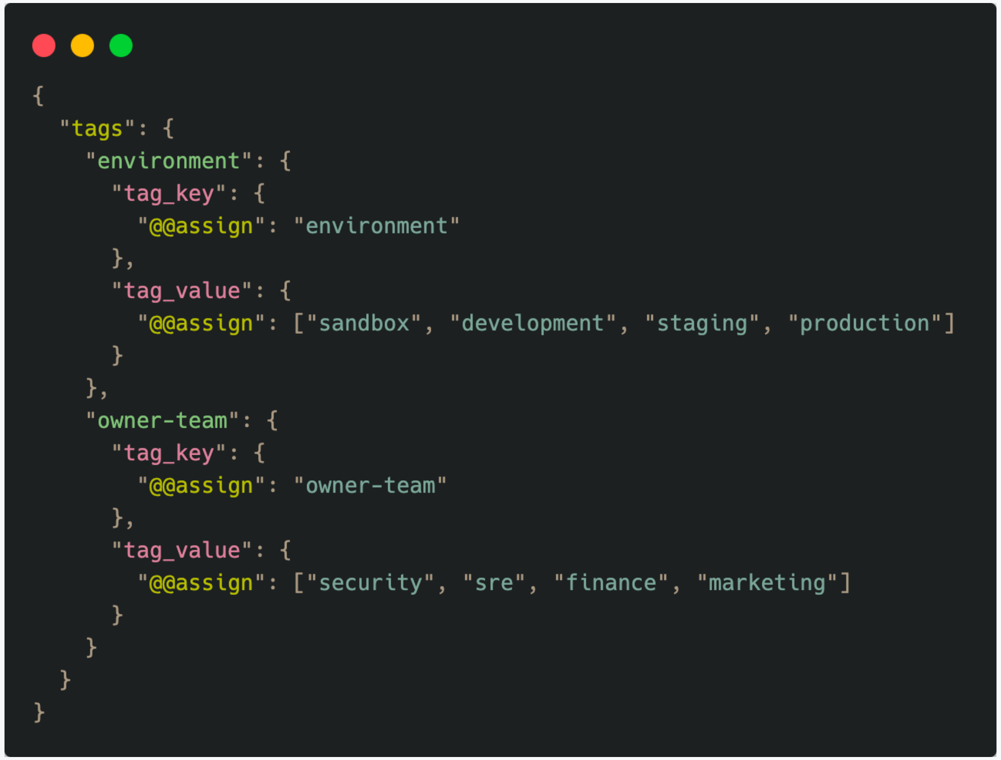
- Block root account usage
- Block root key creation
- Protect sensitive IAM roles
- Apply logic in SCPs using meaningful IAM role paths
  - arn:aws:iam::*:role/org-admin/…
  - arn:aws:iam::*:role/ec2-admin/…
  - arn:aws:iam::*:role/eks-admin/…



```
{ "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "iam:UpdateRoleDescription",
        "iam:UpdateRole",
        "iam:UpdateAssumeRolePolicy",
        "iam:PutRolePolicy",
        "iam:PutRolePermissionsBoundary",
        "iam:DetachRolePolicy",
        "iam:DeleteRolePolicy",
        "iam:DeleteRolePermissionsBoundary",
        "iam:DeleteRole",
        "iam:AttachRolePolicy",
        "iam:TagRole",
        "iam:UntagRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/OrganizationAccountAccessRole",
        "arn:aws:iam::*:role/org_admin/*",
        "arn:aws:iam::*:role/org-admin/*"
      ],
      "Condition": {
        "ArnNotEquals": {"aws:PrincipalArn": "arn:aws:iam::*:role/OrganizationAccountAccessRole"}
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "iam:TagPolicy",
        "iam:UntagPolicy",
        "iam:CreatePolicy",
        "iam:DeletePolicy",
        "iam:CreatePolicyVersion",
        "iam:DeletePolicyVersion",
        "iam:SetDefaultPolicyVersion"
      ],
      "Resource": ["arn:aws:iam::*:policy/org_admin/*", "arn:aws:iam::*:policy/org-admin/*"],
      "Condition": {
        "ArnNotEquals": {"aws:PrincipalArn": "arn:aws:iam::*:role/OrganizationAccountAccessRole"}
      }
    }
  ]}
```

Organization SCPs

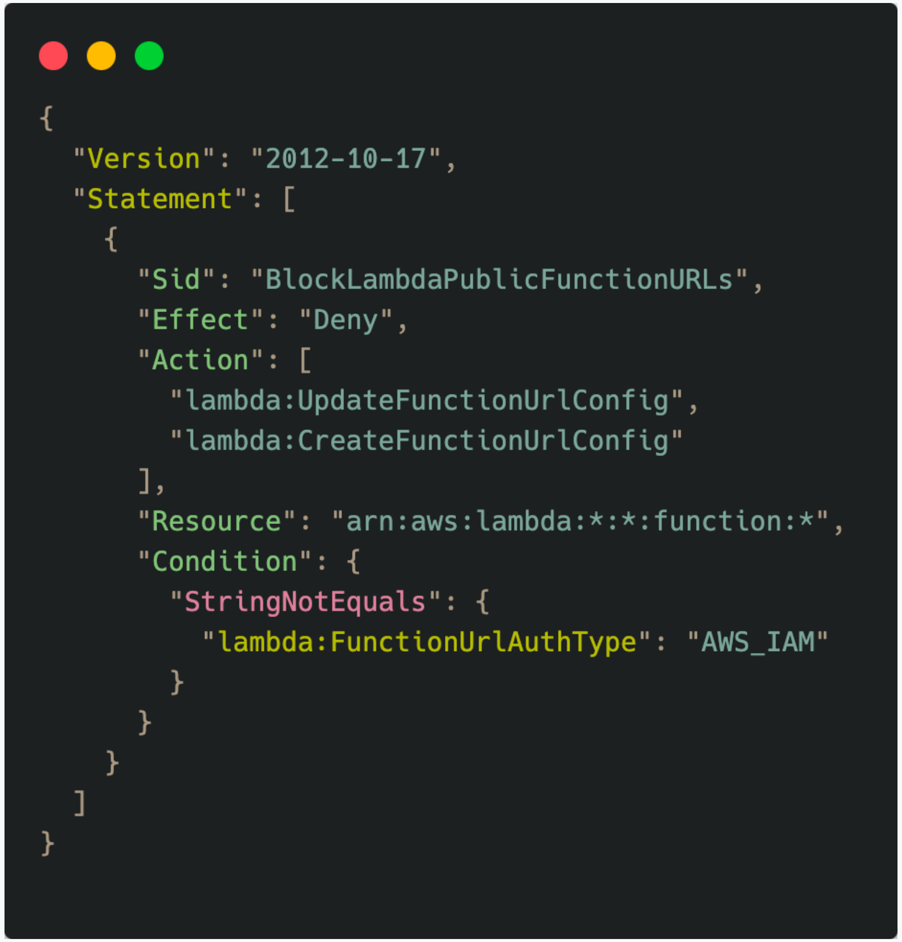
- Block root account usage
- Block root key creation
- Protect sensitive IAM roles
- Apply logic in SCPs using meaningful IAM role paths
  - arn:aws:iam::*:role/org-admin/…
  - arn:aws:iam::*:role/ec2-admin/…
  - arn:aws:iam::*:role/eks-admin/…
- Tag everything
  - Define a tagging standard
  - AWS Organization Tag Policies (enforce tags, tag keys and values, case sensitivity)
  - Use a tag policy with an SCP/IAM Role to fully enforce tagging on specific services
  - Use tags for ABAC



```
{
  "tags": {
    "environment": {
      "tag_key": {
        "@@assign": "environment"
      },
      "tag_value": {
        "@@assign": ["sandbox", "development", "staging", "production"]
      }
    },
    "owner-team": {
      "tag_key": {
        "@@assign": "owner-team"
      },
      "tag_value": {
        "@@assign": ["security", "sre", "finance", "marketing"]
      }
    }
  }
}
```

Organization SCPs

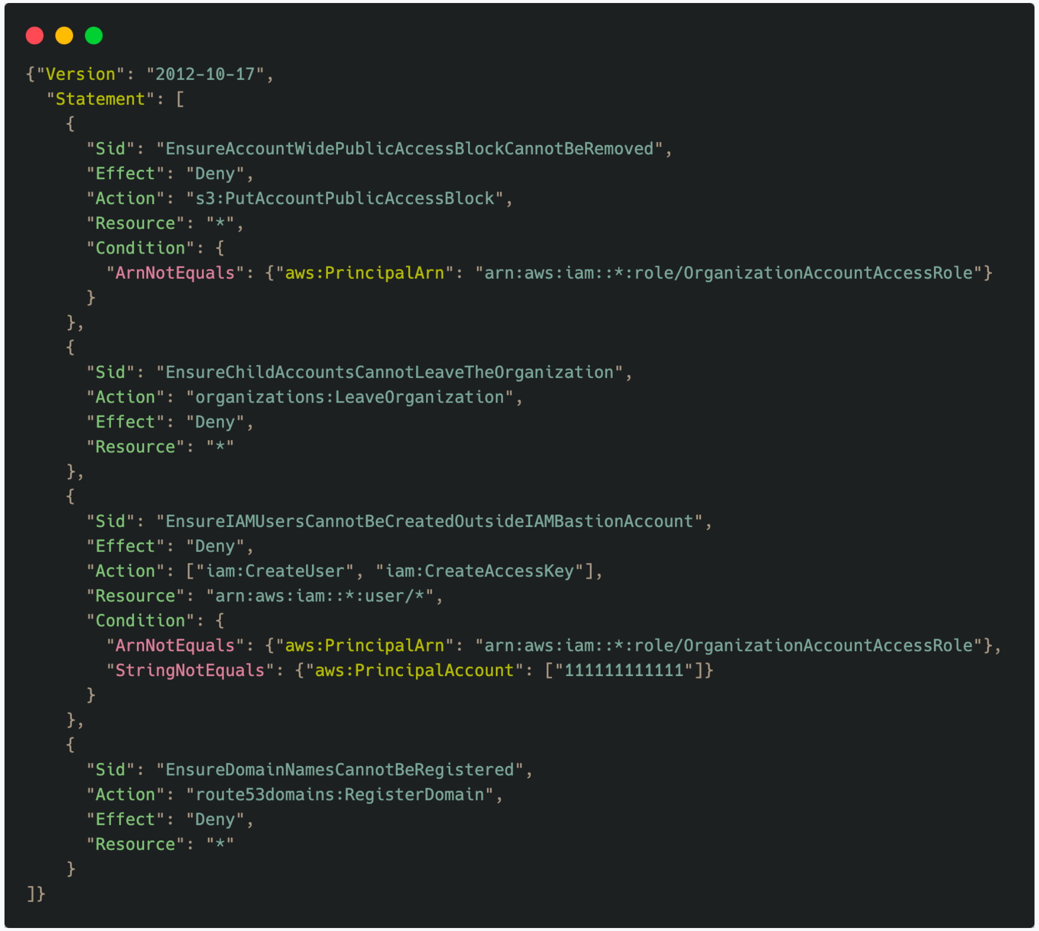
- Block root account usage
- Block root key creation
- Protect sensitive IAM roles
- Apply logic in SCPs using meaningful IAM role paths
    - arn:aws:iam::*:role/org-admin/…
    - arn:aws:iam::*:role/ec2-admin/…
    - arn:aws:iam::*:role/eks-admin/…
- Tag everything
    - Define a tagging standard
    - AWS Organization Tag Policies (enforce tags, tag keys and values, case sensitivity)
    - Use a tag policy with an SCP/IAM Role to fully enforce tagging on specific services
    - Use tags for ABAC
- Deny unauthenticated Lambda URLs (stay up to date with AWS services!)



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BlockLambdaPublicFunctionURLs",
      "Effect": "Deny",
      "Action": [
        "lambda:UpdateFunctionUrlConfig",
        "lambda:CreateFunctionUrlConfig"
      ],
      "Resource": "arn:aws:lambda:*:*:function:*",
      "Condition": {
        "StringNotEquals": {
          "lambda:FunctionUrlAuthType": "AWS_IAM"
        }
      }
    }
  ]
}
```

Organization SCPs

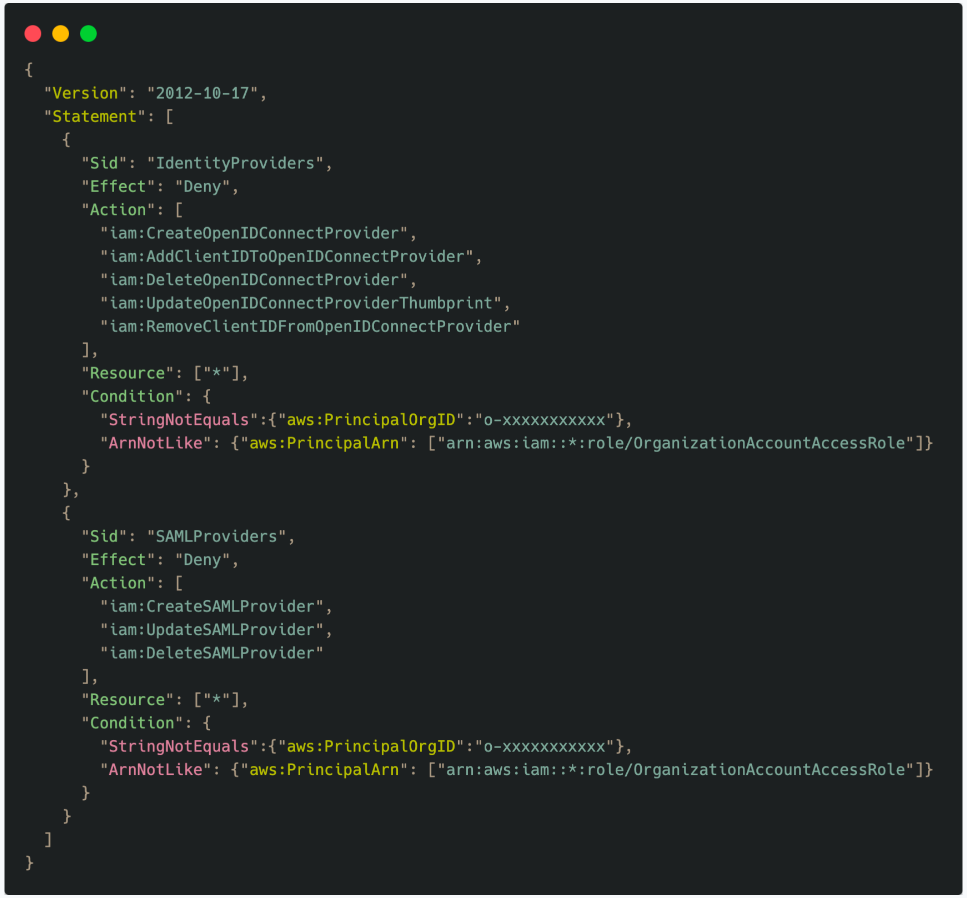
- Block root account usage
- Block root key creation
- Protect sensitive IAM roles
- Apply logic in SCPs using meaningful IAM role paths
  - arn:aws:iam::*:role/org-admin/…
  - arn:aws:iam::*:role/ec2-admin/…
  - arn:aws:iam::*:role/eks-admin/…
- Tag everything
  - Define a tagging standard
  - AWS Organization Tag Policies (enforce tags, tag keys and values, case sensitivity)
  - Use a tag policy with an SCP/IAM Role to fully enforce tagging on specific services
  - Use tags for ABAC
- Deny unauthenticated Lambda URLs (stay up to date!)
- Block Dangerous Actions
- Sid counts for the SCP size (max: 5120 characters)



```
{"Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnsureAccountWidePublicAccessBlockCannotBeRemoved",
      "Effect": "Deny",
      "Action": "s3:PutAccountPublicAccessBlock",
      "Resource": "*",
      "Condition": {
        "ArnNotEquals": {"aws:PrincipalArn": "arn:aws:iam::*:role/OrganizationAccountAccessRole"}
      }
    },
    {
      "Sid": "EnsureChildAccountsCannotLeaveTheOrganization",
      "Action": "organizations:LeaveOrganization",
      "Effect": "Deny",
      "Resource": "*"
    },
    {
      "Sid": "EnsureIAMUsersCannotBeCreatedOutsideIAMBastionAccount",
      "Effect": "Deny",
      "Action": ["iam:CreateUser", "iam:CreateAccessKey"],
      "Resource": "arn:aws:iam::*:user/*",
      "Condition": {
        "ArnNotEquals": {"aws:PrincipalArn": "arn:aws:iam::*:role/OrganizationAccountAccessRole"},
        "StringNotEquals": {"aws:PrincipalAccount": ["111111111111"]}
      }
    },
    {
      "Sid": "EnsureDomainNamesCannotBeRegistered",
      "Action": "route53domains:RegisterDomain",
      "Effect": "Deny",
      "Resource": "*"
    }
]}
```

Organization SCPs - IdP

- Block Identity Provider Registrations
  - IdPs can be used for persistence
  - Admin users can register OIDC providers for testing purposes and not remove them or using non approved services
- Monitor OIDC provider
  - Creation
  - Deletion
  - Modification
- Possible to apply restrictions to action: **sts:AssumeRoleWithWebIdentity**
- Not possible to restrict role creation with identity provider trust policies ☹
  - One solution is IAM Access Analyser

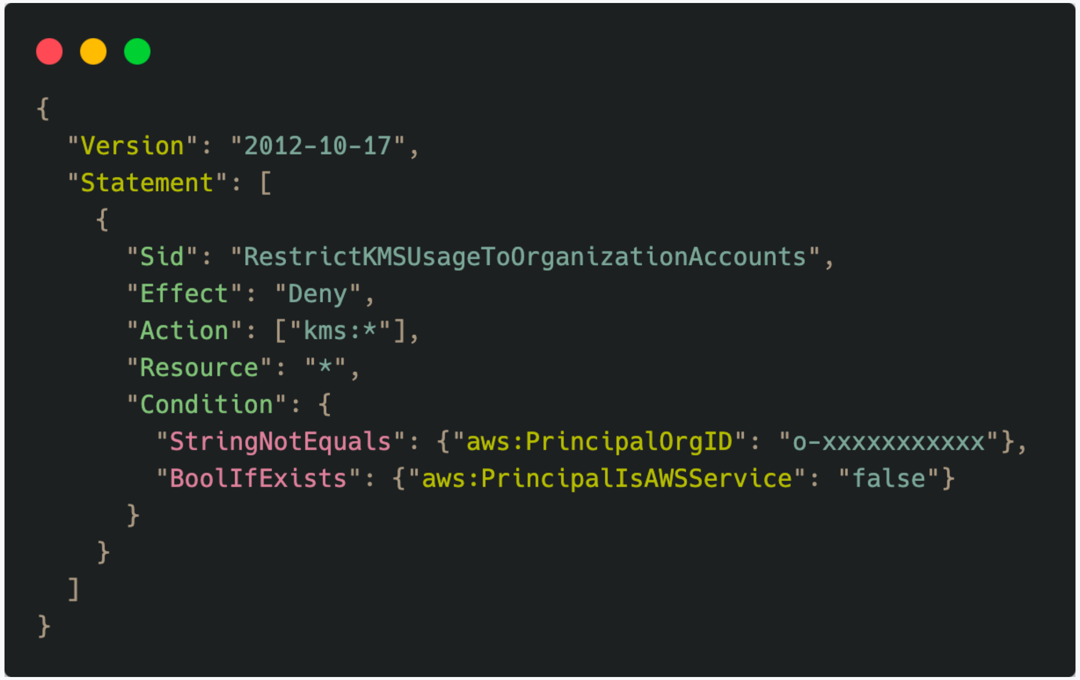


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IdentityProviders",
      "Effect": "Deny",
      "Action": [
        "iam:CreateOpenIDConnectProvider",
        "iam:AddClientIDToOpenIDConnectProvider",
        "iam:DeleteOpenIDConnectProvider",
        "iam:UpdateOpenIDConnectProviderThumbprint",
        "iam:RemoveClientIDFromOpenIDConnectProvider"
      ],
      "Resource": ["*"],
      "Condition": {
        "StringNotEquals":{"aws:PrincipalOrgID":"o-xxxxxxxxxxx"},
        "ArnNotLike": {"aws:PrincipalArn": ["arn:aws:iam::*:role/OrganizationAccountAccessRole"]}
      }
    },
    {
      "Sid": "SAMLProviders",
      "Effect": "Deny",
      "Action": [
        "iam:CreateSAMLProvider",
        "iam:UpdateSAMLProvider",
        "iam:DeleteSAMLProvider"
      ],
      "Resource": ["*"],
      "Condition": {
        "StringNotEquals":{"aws:PrincipalOrgID":"o-xxxxxxxxxxx"},
        "ArnNotLike": {"aws:PrincipalArn": ["arn:aws:iam::*:role/OrganizationAccountAccessRole"]}
      }
    }
  ]
}
```

nexthink

Organization SCPs - KMS

- Restrict KMS keys to organization principals
- Prevent encryption with foreign KMS keys (ransomware)
- Monitor any **kms:Encrypt** actions performed from accounts not in the organization

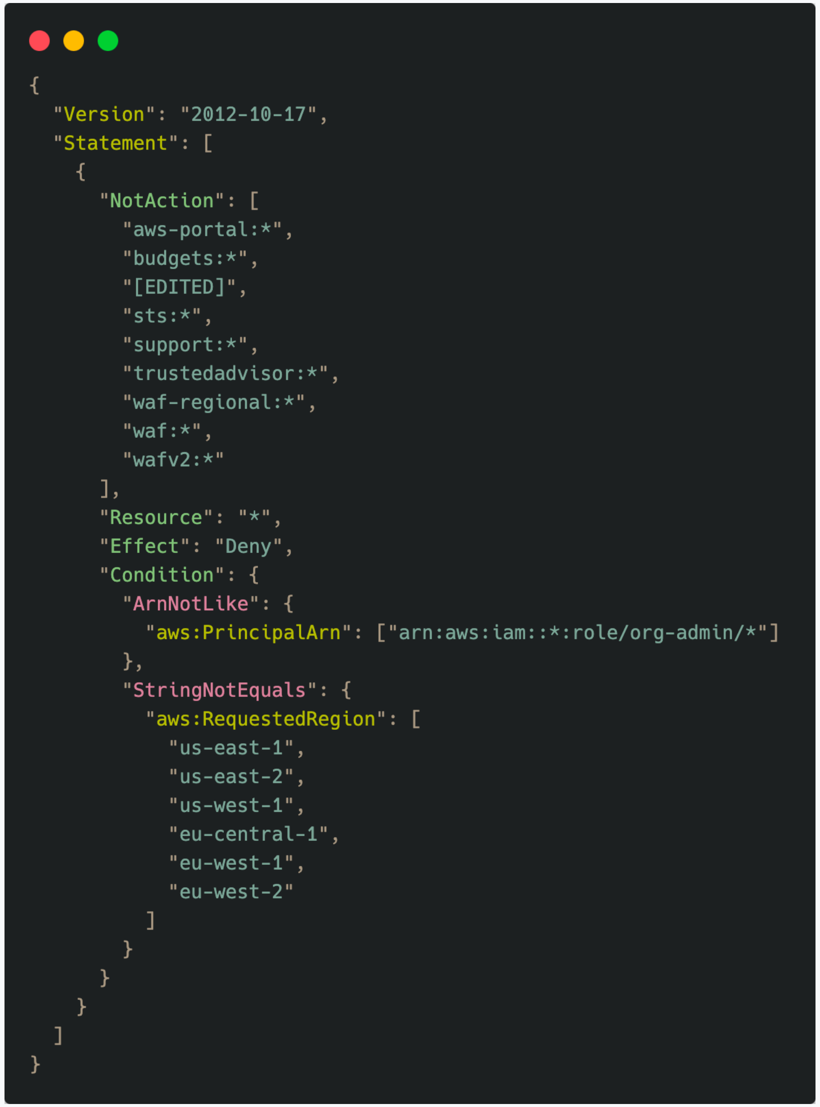


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictKMSUsageToOrganizationAccounts",
      "Effect": "Deny",
      "Action": ["kms:*"],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {"aws:PrincipalOrgID": "o-xxxxxxxxxxx"},
        "BoolIfExists": {"aws:PrincipalIsAWSService": "false"}
      }
    }
  ]
}
```

Organization SCPs- Regions

- Restrict regions per environment/OU
  - Reduce attack surface
  - Reduce cost
  - Compliance (geographical requirements)

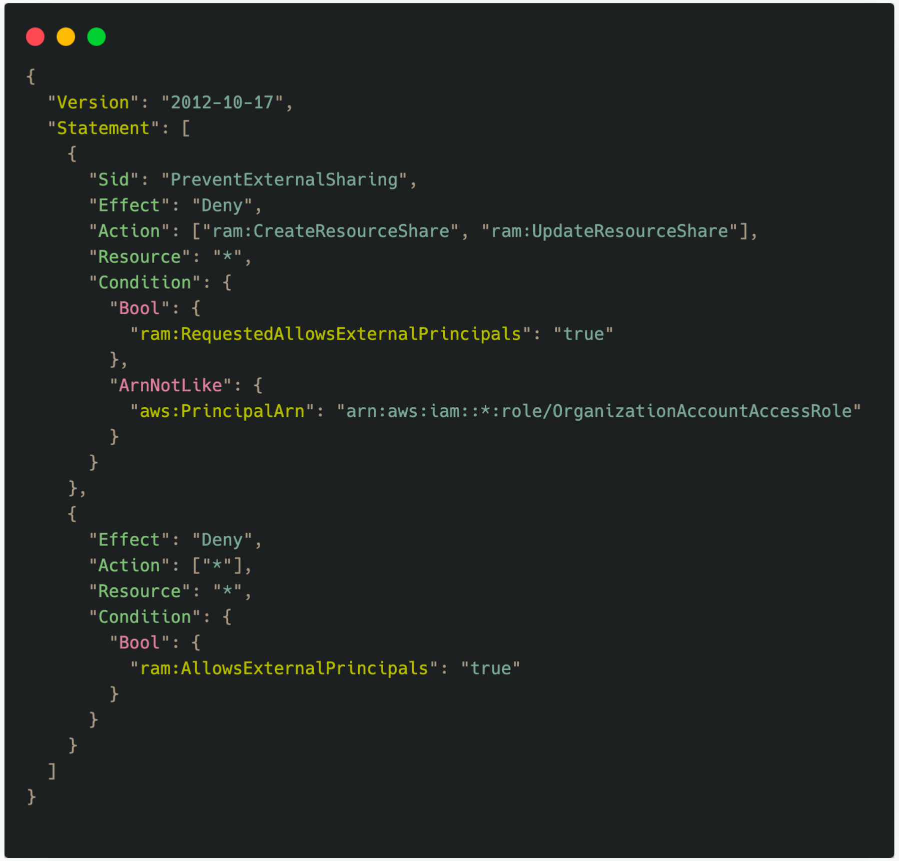
- Ensure global services are allowed
  - IAM
  - CloudFront (us-east-1)
  - S3
  - sts
  - Route53
  - wafv2
  - support



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "NotAction": [
        "aws-portal:*",
        "budgets:*",
        "[EDITED]",
        "sts:*",
        "support:*",
        "trustedadvisor:*",
        "waf-regional:*",
        "waf:*",
        "wafv2:*"
      ],
      "Resource": "*",
      "Effect": "Deny",
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalArn": ["arn:aws:iam::*:role/org-admin/*"]
        },
        "StringNotEquals": {
          "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2",
            "us-west-1",
            "eu-central-1",
            "eu-west-1",
            "eu-west-2"
          ]
        }
      }
    }
  ]
}
```

Organization SCPs – RAM (AWS Resource Access Manager)

- Prevent resource sharing with accounts outside of the organization
  - AWS RAM allows sharing resources across accounts



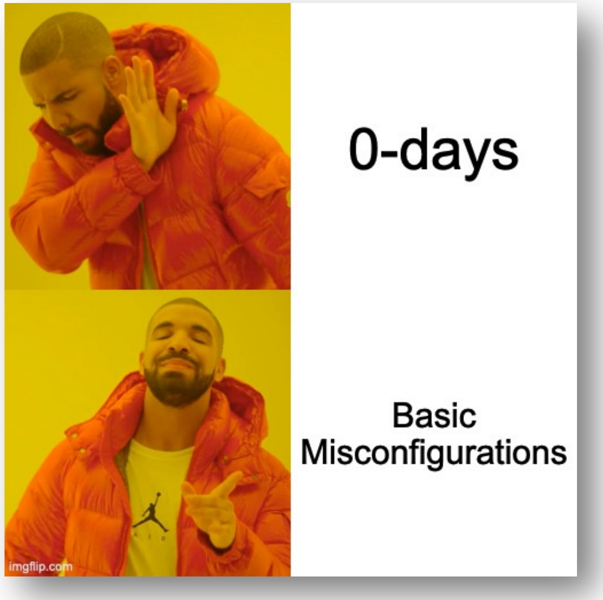
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventExternalSharing",
      "Effect": "Deny",
      "Action": ["ram:CreateResourceShare", "ram:UpdateResourceShare"],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "ram:RequestedAllowsExternalPrincipals": "true"
        },
        "ArnNotLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:role/OrganizationAccountAccessRole"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": ["*"],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "ram:AllowsExternalPrincipals": "true"
        }
      }
    }
  ]
}
```

Organization SCPs - Others

- Restrict regions per environment / OU / account
- Disable S3 bucket ACLs
- Enforce S3 Encryption at rest
- Prevent cross environment actions
- Prevent EBS snapshot public downloads & sharing
- Prevent creation of non-encrypted volumes
- Prevent launching EC2 instances without IMDSv2
- Prevent EC2 instance metadata changes
- Prevent VPCs to get internet access for sensitive workloads
- Prevent resource sharing with accounts outside of the organization
  - AWS RAM allows sharing resources across accounts
- Deny VPC peering
- Deny VPN creation

Access & Configuration

- Cloud Vulnerabilities



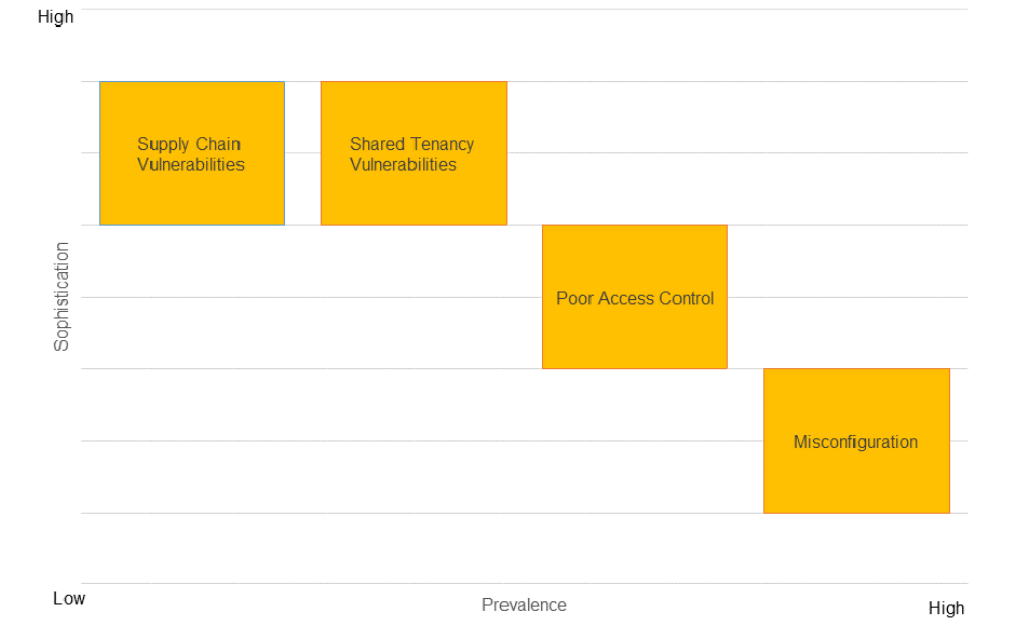
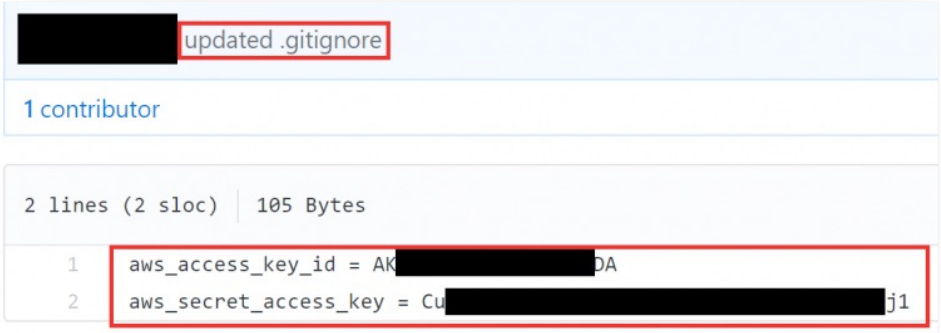


Figure 2: Cloud Vulnerabilities – Prevalence versus Sophistication of Exploitation

https://media.defense.gov/2020/Jan/22/2002237484/-1/-1/0/CSI-MITIGATING-CLOUD-VULNERABILITIES_20200121.PDF

Access and User Management

- SSO (IAM Identity Center) only for human access with MFA enforced and IP restrictions
- AWS Access keys will eventually get leaked in configurations, repositories, file(s) on laptops, etc.
- Check out CLI utility **aws-vault** or **granted** (they work with SSO and the OS keychain)

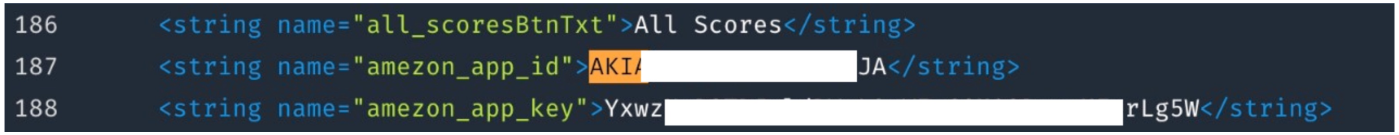


[redacted] updated .gitignore

1 contributor

2 lines (2 sloc) | 105 Bytes

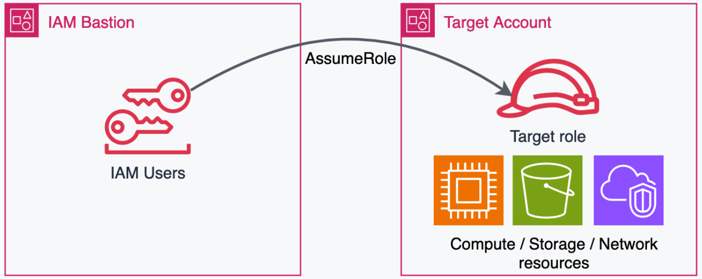
```
aws_access_key_id = AK[redacted]DA
aws_secret_access_key = Cu[redacted]j1
```



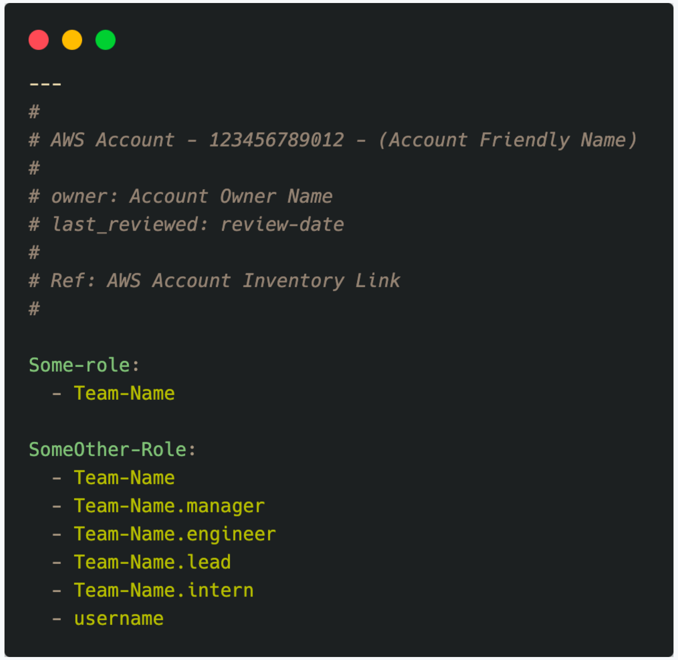
```
        <string name="all_scoresBtnTxt">All Scores</string>
        <string name="amezon_app_id">AKI[redacted]JA</string>
        <string name="amezon_app_key">Yxwz[redacted]rLg5W</string>
```

Access and User Management

- Service users (e.g. jenkins) are centralized in IAM Bastion account & credentials managed by HashiCorp Vault and rotated every 8 hours



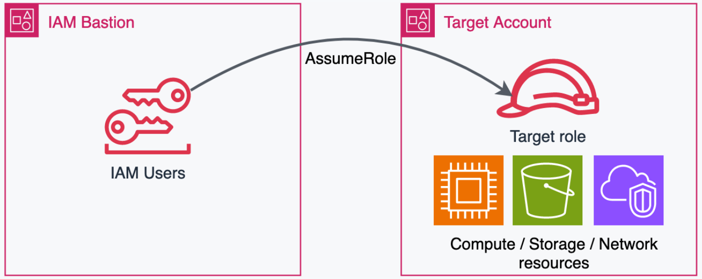
- Permanent access
  - GitHub repo with CODEOWNERS (approvals)

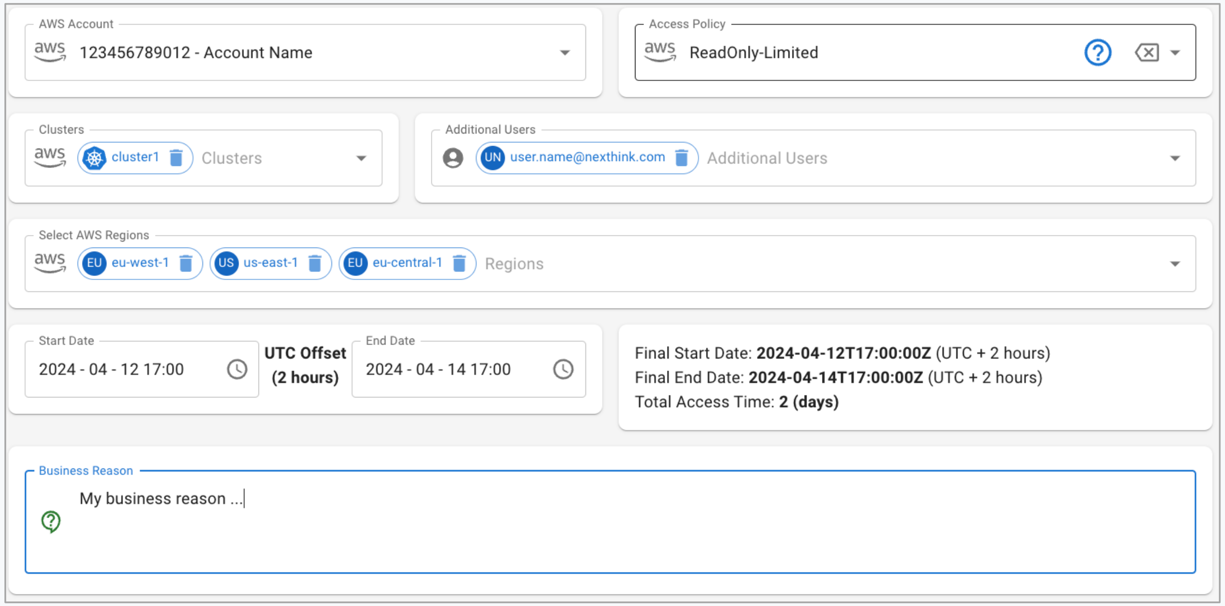


```
---
#
# AWS Account - 123456789012 - (Account Friendly Name)
#
# owner: Account Owner Name
# last_reviewed: review-date
#
# Ref: AWS Account Inventory Link
#

Some-role:
  - Team-Name

SomeOther-Role:
  - Team-Name
  - Team-Name.manager
  - Team-Name.engineer
  - Team-Name.lead
  - Team-Name.intern
  - username
```

Access and User Management

- Service users (e.g. jenkins) are centralized in IAM Bastion account & credentials managed by HashiCorp Vault and rotated every 8 hours





AWS Account: aws 123456789012 - Account Name

Access Policy: aws ReadOnly-Limited

Clusters: aws cluster1 Clusters

Additional Users: UN user.name@nexthink.com Additional Users

Select AWS Regions: aws EU eu-west-1 US us-east-1 EU eu-central-1 Regions

Start Date: 2024 - 04 - 12 17:00

UTC Offset (2 hours)

End Date: 2024 - 04 - 14 17:00

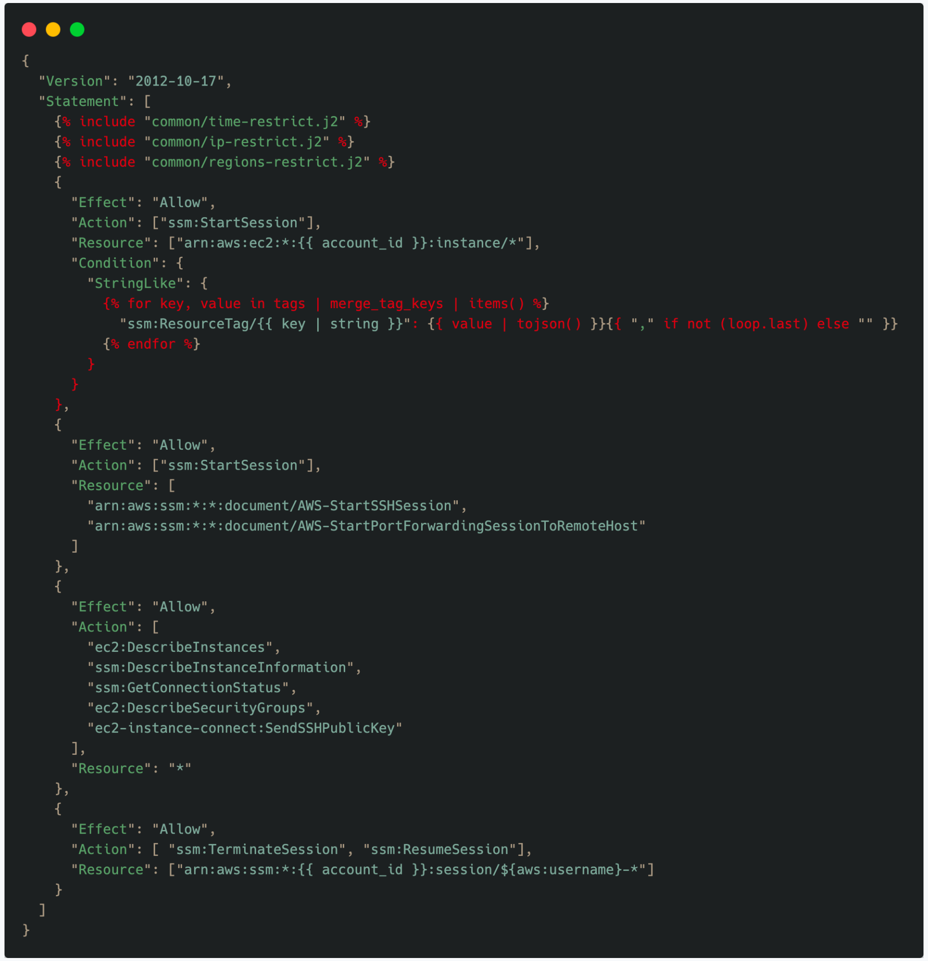
Final Start Date: **2024-04-12T17:00:00Z** (UTC + 2 hours)
Final End Date: **2024-04-14T17:00:00Z** (UTC + 2 hours)
Total Access Time: **2 (days)**

Business Reason: My business reason ...

- Permanent access
  - GitHub repo with CODEOWNERS (approvals)
- Temporary access
  - SecAWS (in-house Just-In-Time Access tool)
    - Access with approval + time limit
    - IaC managed
    - RBAC for roles ⚠
- Break Glass process

Temporary Access- SecAWS

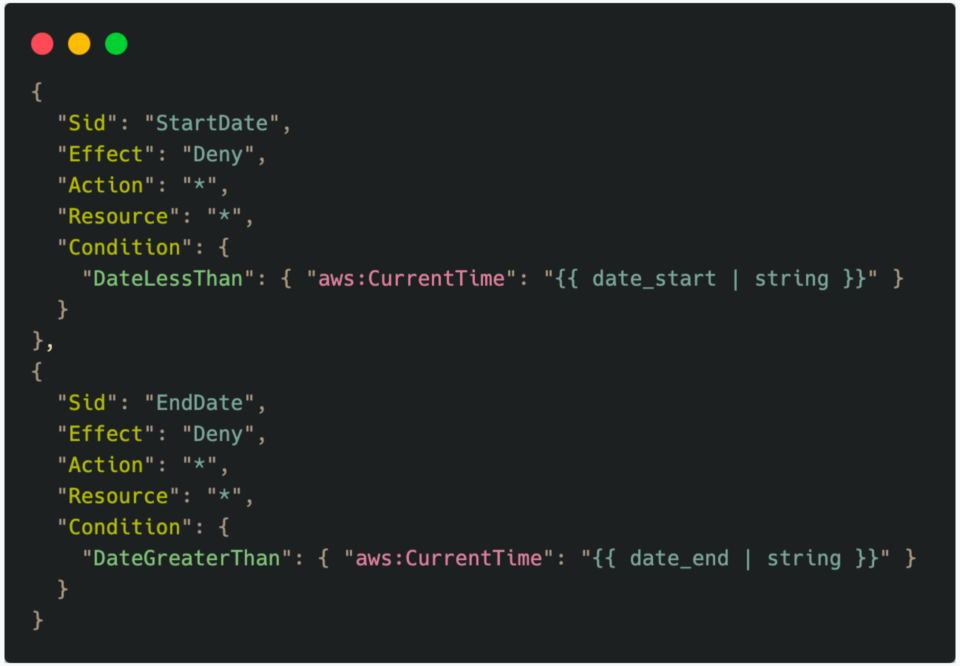
- FastAPI, React & Jinja2
- Jinja templates allows to have least privileges but in different scenarios



```
{
  "Version": "2012-10-17",
  "Statement": [
    {% include "common/time-restrict.j2" %}
    {% include "common/ip-restrict.j2" %}
    {% include "common/regions-restrict.j2" %}
    {
      "Effect": "Allow",
      "Action": ["ssm:StartSession"],
      "Resource": ["arn:aws:ec2:*:{{ account_id }}:instance/*"],
      "Condition": {
        "StringLike": {
          {% for key, value in tags | merge_tag_keys | items() %}
            "ssm:ResourceTag/{{ key | string }}": {{ value | tojson() }}{{ "," if not (loop.last) else "" }}
          {% endfor %}
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": ["ssm:StartSession"],
      "Resource": [
        "arn:aws:ssm:*:*:document/AWS-StartSSHSession",
        "arn:aws:ssm:*:*:document/AWS-StartPortForwardingSessionToRemoteHost"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ssm:DescribeInstanceInformation",
        "ssm:GetConnectionStatus",
        "ec2:DescribeSecurityGroups",
        "ec2-instance-connect:SendSSHPublicKey"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [ "ssm:TerminateSession", "ssm:ResumeSession"],
      "Resource": ["arn:aws:ssm:*:{{ account_id }}:session/${aws:username}-*"]
    }
  ]
}
```

Temporary Access - SecAWS

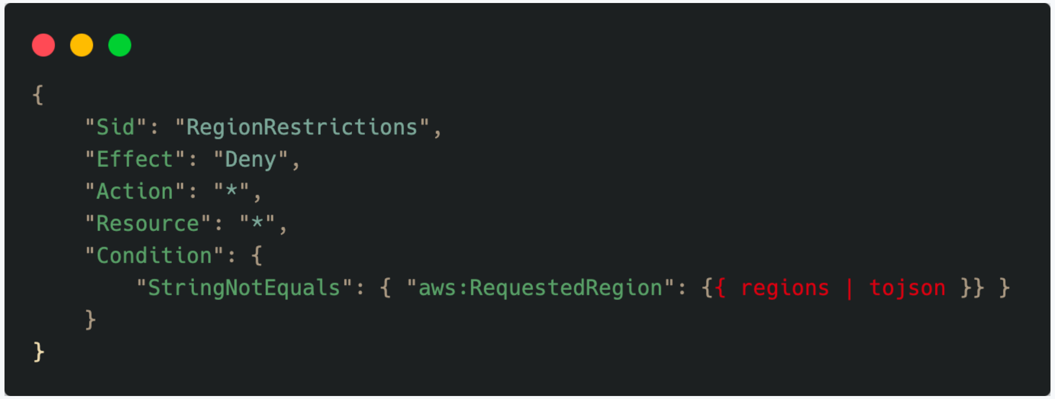
- FastAPI, React & Jinja2
- Jinja templates allows to have least privileges but in different scenarios
- All permission sets have inline IAM date condition for start & end date
  - Date format: %Y-%m-%dT%H:%M:%SZ



```
{
  "Sid": "StartDate",
  "Effect": "Deny",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "DateLessThan": { "aws:CurrentTime": "{{ date_start | string }}" }
  }
},
{
  "Sid": "EndDate",
  "Effect": "Deny",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "DateGreaterThan": { "aws:CurrentTime": "{{ date_end | string }}" }
  }
}
```

Temporary Access - SecAWS

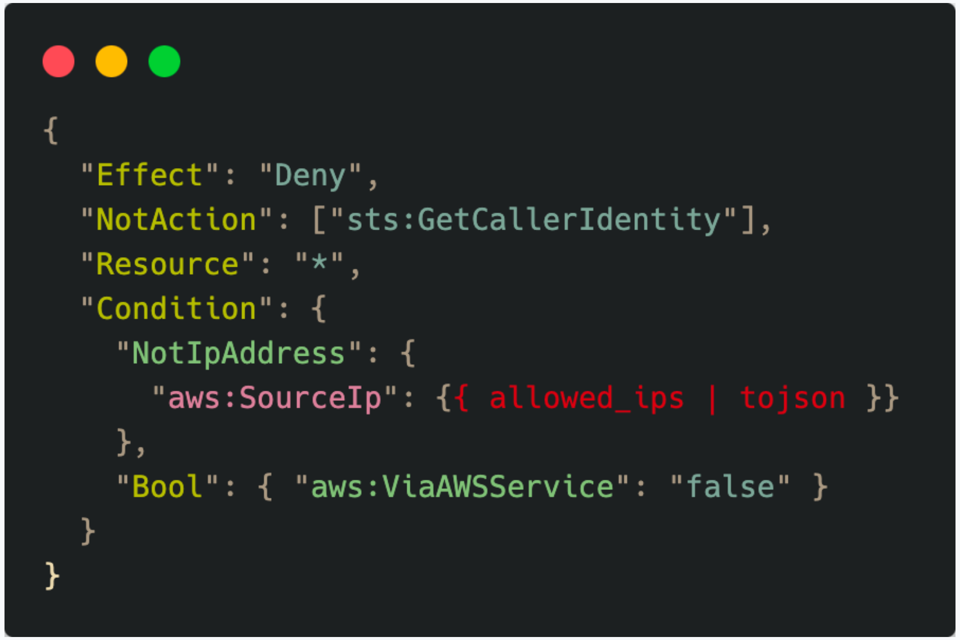
- FastAPI, React & Jinja2
- Jinja templates allows to have least privileges but in different scenarios
- All permission sets have inline IAM date condition for start & end date
    - Date format: %Y-%m-%dT%H:%M:%SZ
- Region restrictions



```
{
    "Sid": "RegionRestrictions",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "StringNotEquals": { "aws:RequestedRegion": {{ regions | tojson }} }
    }
}
```

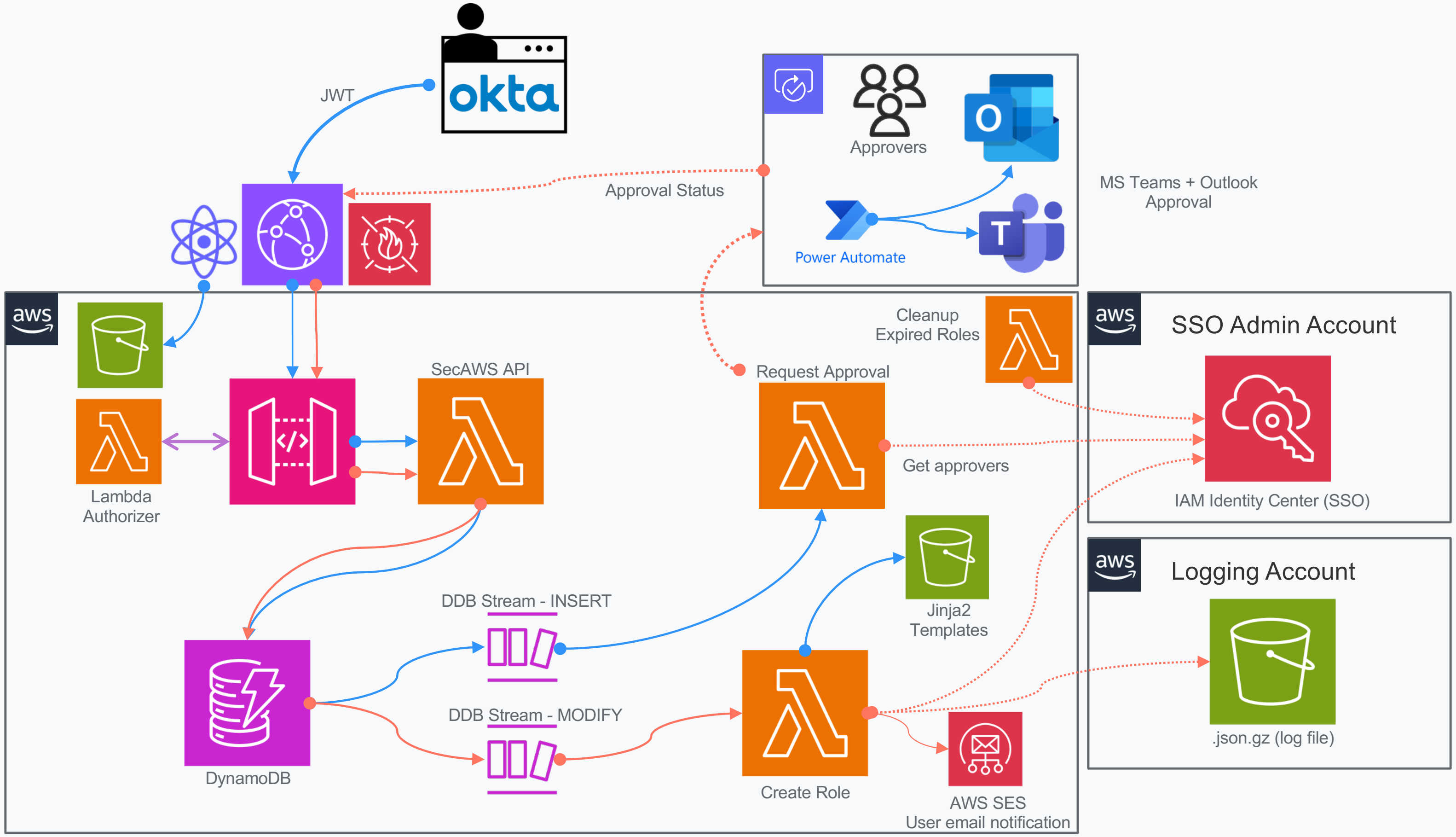
Temporary Access - SecAWS

- FastAPI, React & Jinja2
- Jinja templates allows to have least privileges but in different scenarios
- All permission sets have inline IAM date condition for start & end date
  - Date format: %Y-%m-%dT%H:%M:%SZ
- Region restrictions
- IP restrictions + always on VPN
- Cleanup for expired roles every 4 hours



```
{
  "Effect": "Deny",
  "NotAction": ["sts:GetCallerIdentity"],
  "Resource": "*",
  "Condition": {
    "NotIpAddress": {
      "aws:SourceIp": {{ allowed_ips | tojson }}
    },
    "Bool": { "aws:ViaAWSService": "false" }
  }
}
```

SecAWS - Architecture



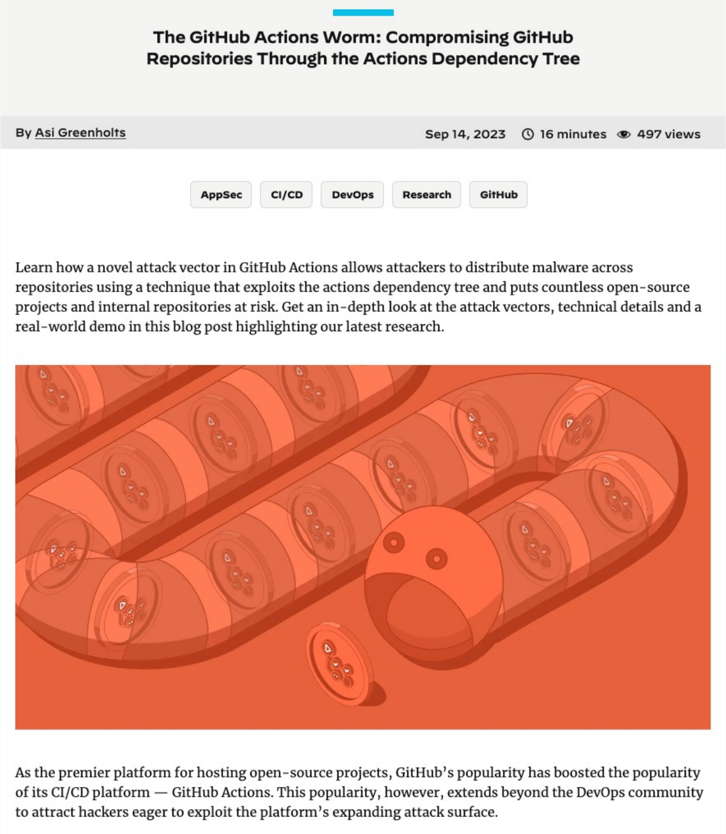
nexthink

SecAWS – Features (+challenges)

- Each request can have up to 6 total users
  - Requester + 5 additional users
    - reduce approvals
    - speed up access during support, incidents
- A user can
  - delete their own provisioned role
  - replay an expired or deleted role
- Add specific approvers for a given policy
- Remove approvals for a given accounts or policy
- All requests, changes, replays go through the approval process
- Each policy is time bound and must have region(s)
- Form checks are configured in the backend (regex for fields, allowed values, etc.)
- All actions are logged in JSON format (immutable logs in S3)
  - SIEM Monitoring, etc.

Access and User Management- IaC

- Migrate IaC from Jenkins to GitHub Actions
  - Special care for approving GH Actions for the GH organization (software supply chain attack, malware, etc..)
- Referencing GH Actions
  - Commit: **actions/checkout@cd7d[...]1a8b017**
  - Branch: **actions/checkout@main**
  - Tag: **actions/checkout@v1**



The GitHub Actions Worm: Compromising GitHub Repositories Through the Actions Dependency Tree

By Asi Greenholts Sep 14, 2023 16 minutes 497 views

AppSec CI/CD DevOps Research GitHub

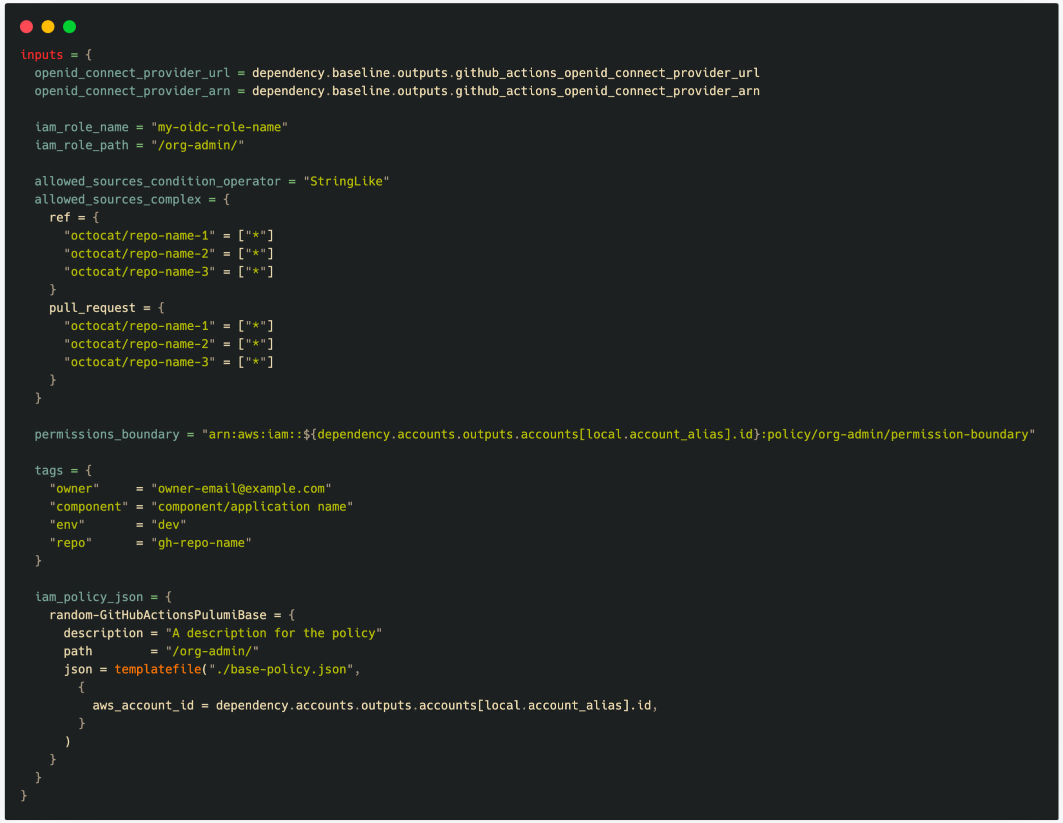
Learn how a novel attack vector in GitHub Actions allows attackers to distribute malware across repositories using a technique that exploits the actions dependency tree and puts countless open-source projects and internal repositories at risk. Get an in-depth look at the attack vectors, technical details and a real-world demo in this blog post highlighting our latest research.

As the premier platform for hosting open-source projects, GitHub's popularity has boosted the popularity of its CI/CD platform — GitHub Actions. This popularity, however, extends beyond the DevOps community to attract hackers eager to exploit the platform's expanding attack surface.

https://www.paloaltonetworks.com/blog/prisma-cloud/github-actions-worm-dependencies/

Access and User Management - IaC

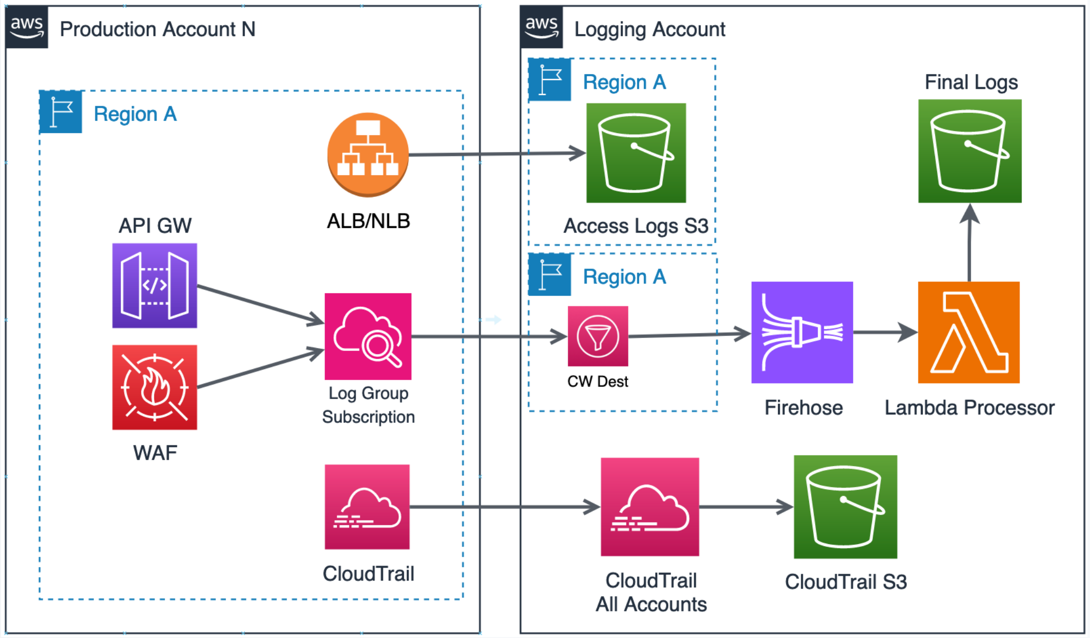
- Migrate IaC from Jenkins to GitHub Actions
  - Special care for approving GH Actions for the GH organization (software supply chain attack, malware, etc..)
- Referencing GH Actions
  - Commit: **actions/checkout@cd7d[...]1a8b017**
  - Branch: **actions/checkout@main**
  - Tag: **actions/checkout@v1**
- OIDC roles
  - No long-lived secrets to manage anymore
  - Can be configured for: pull requests, environment, branch
  - Different roles can be applied based on environment ReadOnly vs. Admin



```
inputs = {
  openid_connect_provider_url = dependency.baseline.outputs.github_actions_openid_connect_provider_url
  openid_connect_provider_arn = dependency.baseline.outputs.github_actions_openid_connect_provider_arn

  iam_role_name = "my-oidc-role-name"
  iam_role_path = "/org-admin/"

  allowed_sources_condition_operator = "StringLike"
  allowed_sources_complex = {
    ref = {
      "octocat/repo-name-1" = ["*"]
      "octocat/repo-name-2" = ["*"]
      "octocat/repo-name-3" = ["*"]
    }
    pull_request = {
      "octocat/repo-name-1" = ["*"]
      "octocat/repo-name-2" = ["*"]
      "octocat/repo-name-3" = ["*"]
    }
  }

  permissions_boundary = "arn:aws:iam::${dependency.accounts.outputs.accounts[local.account_alias].id}:policy/org-admin/permission-boundary"

  tags = {
    "owner"     = "owner-email@example.com"
    "component" = "component/application name"
    "env"       = "dev"
    "repo"      = "gh-repo-name"
  }

  iam_policy_json = {
    random-GitHubActionsPulumiBase = {
      description = "A description for the policy"
      path        = "/org-admin/"
      json = templatefile("./base-policy.json",
        {
          aws_account_id = dependency.accounts.outputs.accounts[local.account_alias].id,
        }
      )
    }
  }
}
```

Logs – Centralization & Parsing

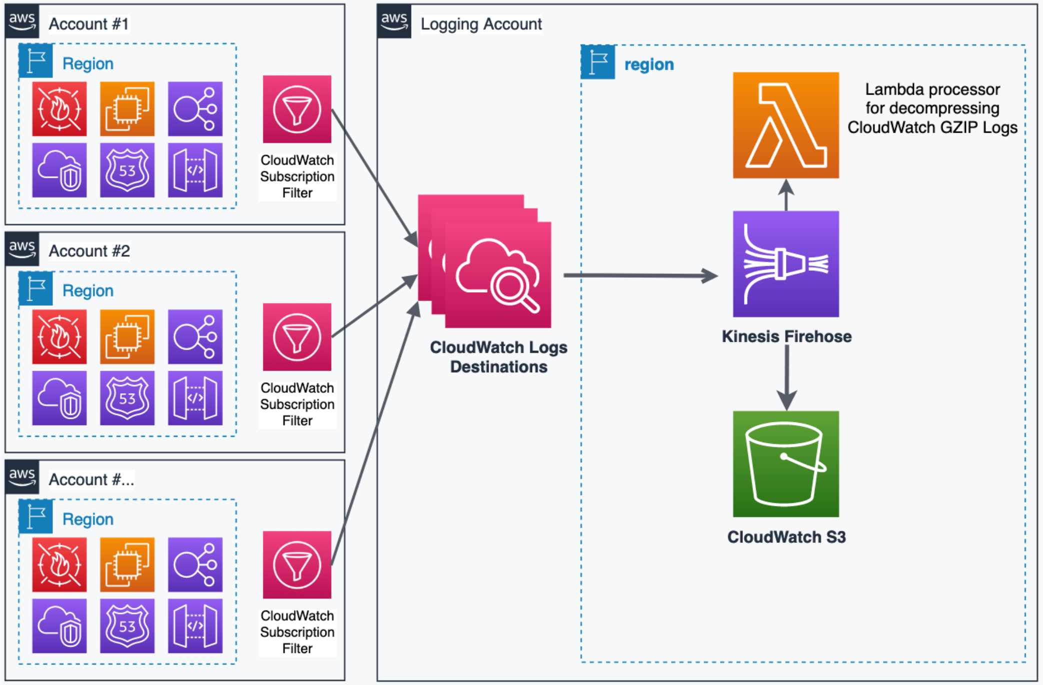
- Send all logs into a central account (Logging)
- Direct S3 upload (easier setup & cheaper)
- Services not logging to S3
  - CloudWatch log destinations (CLI only)
- Security relevant logs are sent to SIEM
- Other logs are made queryable using Athena
  - Configuration and table schemas deployed with IaC
  - Saved queries available for easy search
- VPC Flow logs (can be expensive)
  - What size of logs will be generated?
  - Should VPC flow logs be enabled everywhere (north – south traffic)



nexthink

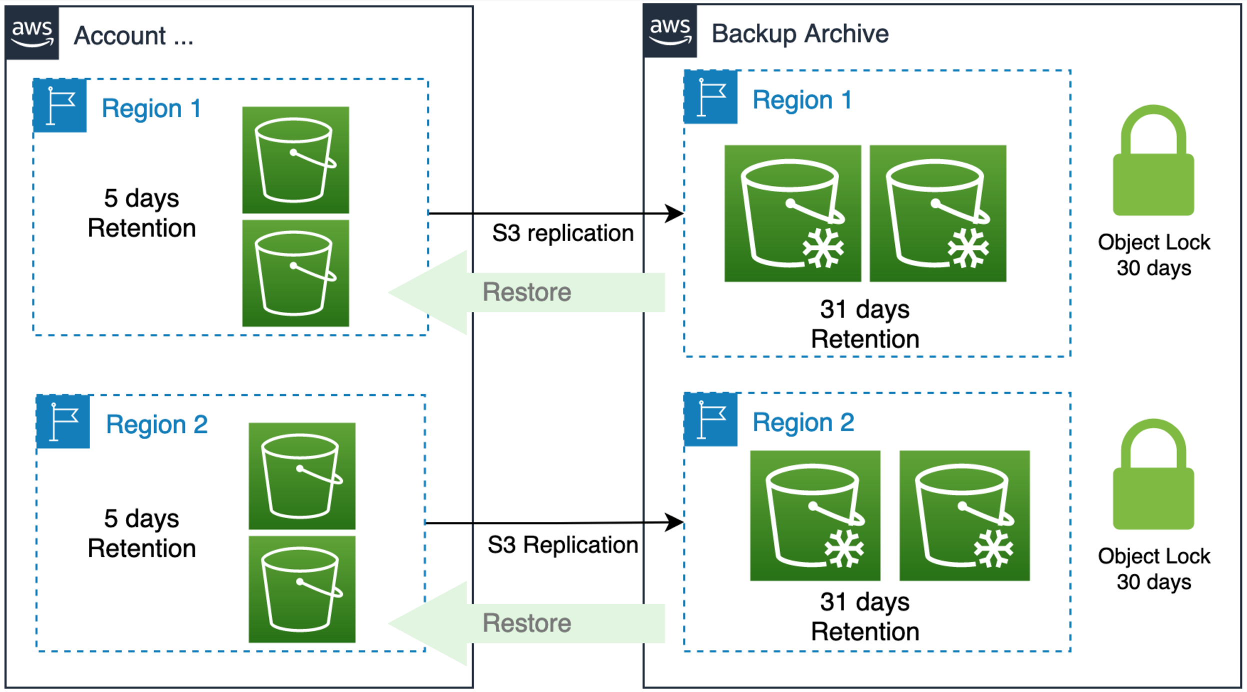
Logs – Centralization & Parsing

- Send all logs into a central account (Logging)
- Direct S3 upload (easier setup & cheaper)
- Services not logging to S3
  - CloudWatch log destinations (CLI only)
- Security relevant logs are sent to SIEM
- Other logs are made queryable using Athena
  - Configuration and table schemas deployed with IaC
  - Saved queries available for easy search
- VPC Flow logs (can be expensive)
  - What size of logs will be generated?
  - Should VPC flow logs be enabled everywhere (north – south traffic)


Account #1
Region
CloudWatch Subscription Filter
Account #2
Region
CloudWatch Subscription Filter
Account #...
Region
CloudWatch Subscription Filter
Logging Account
region
CloudWatch Logs Destinations
Lambda processor for decompressing CloudWatch GZIP Logs
Kinesis Firehose
CloudWatch S3

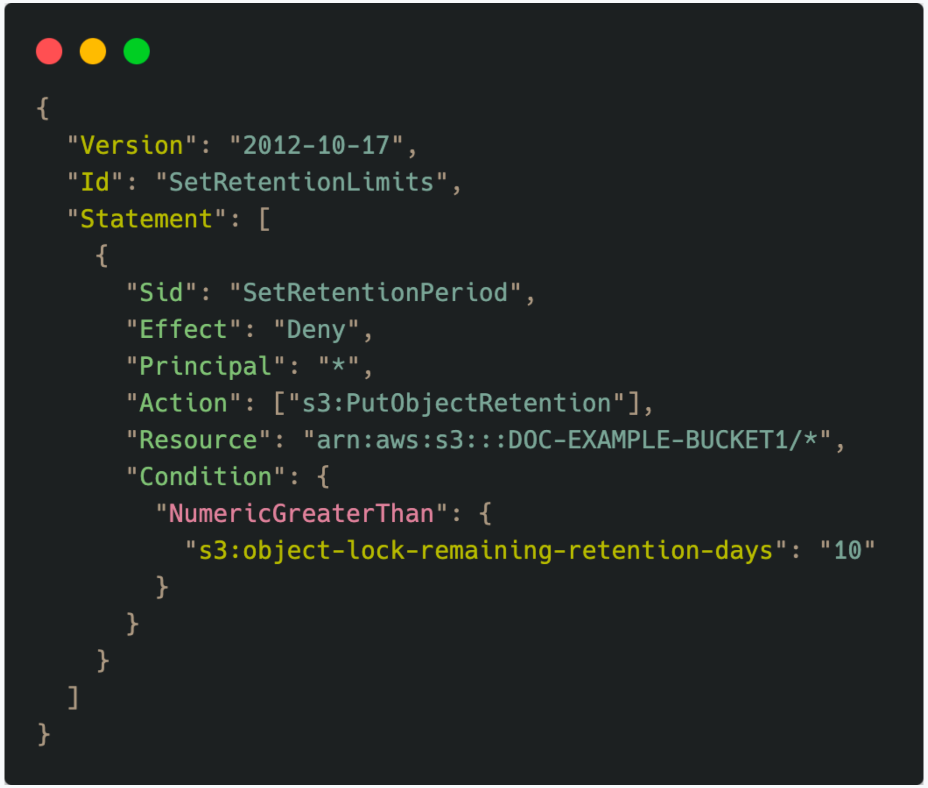
nexthink

Backup Protection (Ransomware)



- S3 Object Lock (COMPLIANCE mode)
- Managed through IaC (terraform + terragrunt)
- Source retention based on use-case
- Options per S3 bucket (lifecycle, retention, etc.)

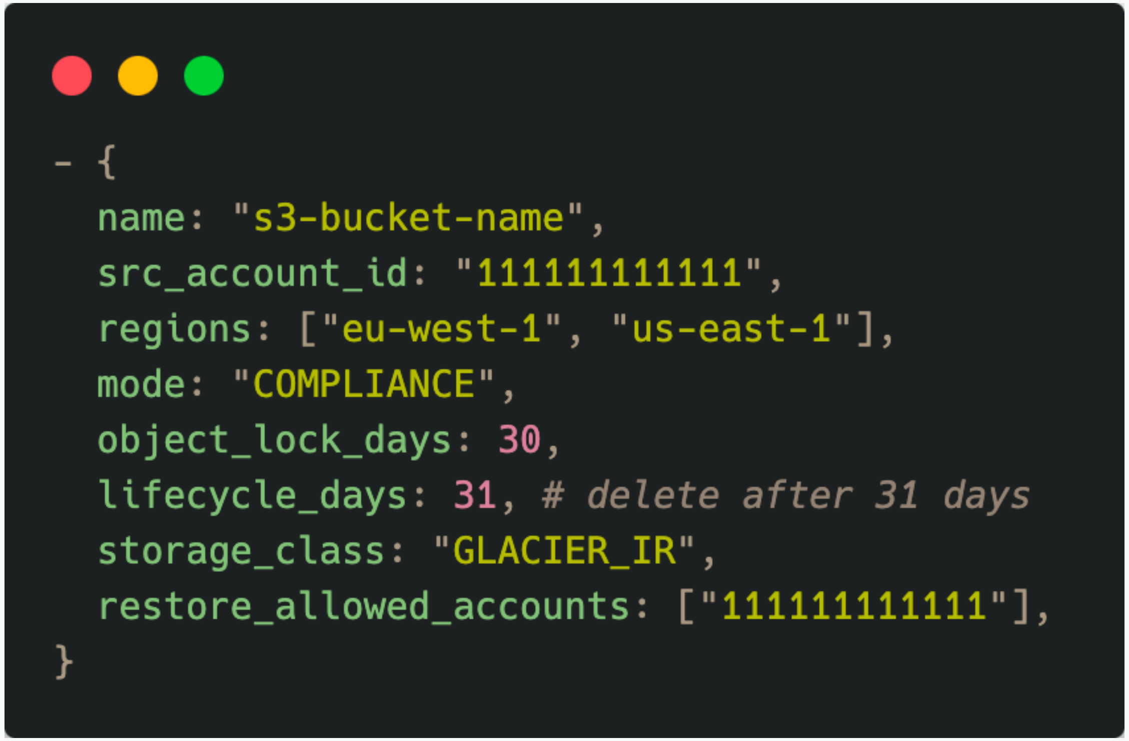
Backup Protection (Ransomware)



```
{
  "Version": "2012-10-17",
  "Id": "SetRetentionLimits",
  "Statement": [
    {
      "Sid": "SetRetentionPeriod",
      "Effect": "Deny",
      "Principal": "*",
      "Action": ["s3:PutObjectRetention"],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
      "Condition": {
        "NumericGreaterThan": {
          "s3:object-lock-remaining-retention-days": "10"
        }
      }
    }
  ]
}
```

- S3 Object Lock (COMPLIANCE mode)
- Managed through IaC (terraform + terragrunt)
- Source retention based on use-case
- Options per S3 bucket (lifecycle, retention, etc.)

Backup Protection (Ransomware)

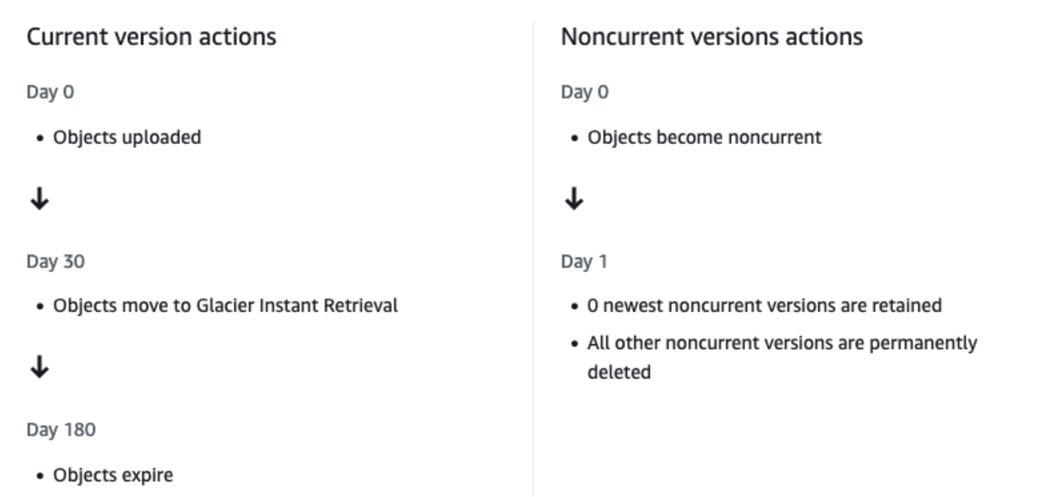


```
- {
  name: "s3-bucket-name",
  src_account_id: "111111111111",
  regions: ["eu-west-1", "us-east-1"],
  mode: "COMPLIANCE",
  object_lock_days: 30,
  lifecycle_days: 31, # delete after 31 days
  storage_class: "GLACIER_IR",
  restore_allowed_accounts: ["111111111111"],
}
```

- S3 Object Lock (COMPLIANCE mode)
- Managed through IaC (terraform + terragrunt)
- Source retention based on use-case
- Options per S3 bucket (lifecycle, retention, etc.)
- YAML abstraction for configuration

Backup Protection – Challenges

- S3 Object Lock – GOVERNANCE vs COMPLIANCE
- Noncurrent versions configuration
- Total retention = current version retention + noncurrent version retention
- S3 Replication Time Control (“S3 RTC” can be expensive)
- Replication monitoring
- Object lock required S3 versioning to be enabled
  - ⚠ If an object is changed at the source, the destination will contain multiple versions of that object and they’ll be “locked” for the retention period
- Possible cost reductions depending on the retention applied on the source bucket
- S3 Storage Class (Standard in source vs. Glacier IR in destination)



Current version actions

Day 0

- Objects uploaded

↓

Day 30

- Objects move to Glacier Instant Retrieval

↓

Day 180

- Objects expire

Noncurrent versions actions

Day 0

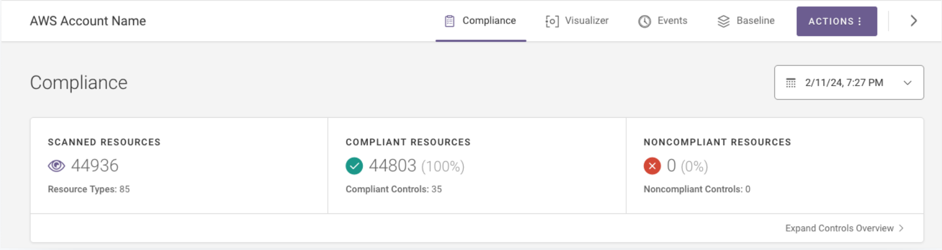
- Objects become noncurrent

↓

Day 1

- 0 newest noncurrent versions are retained
- All other noncurrent versions are permanently deleted

Configuration Hardening



Compliance

2/11/24, 7:27 PM

SCANNED RESOURCES
44936
Resource Types: 85

COMPLIANT RESOURCES
✓ 44803 (100%)
Compliant Controls: 35

NONCOMPLIANT RESOURCES
✕ 0 (0%)
Noncompliant Controls: 0

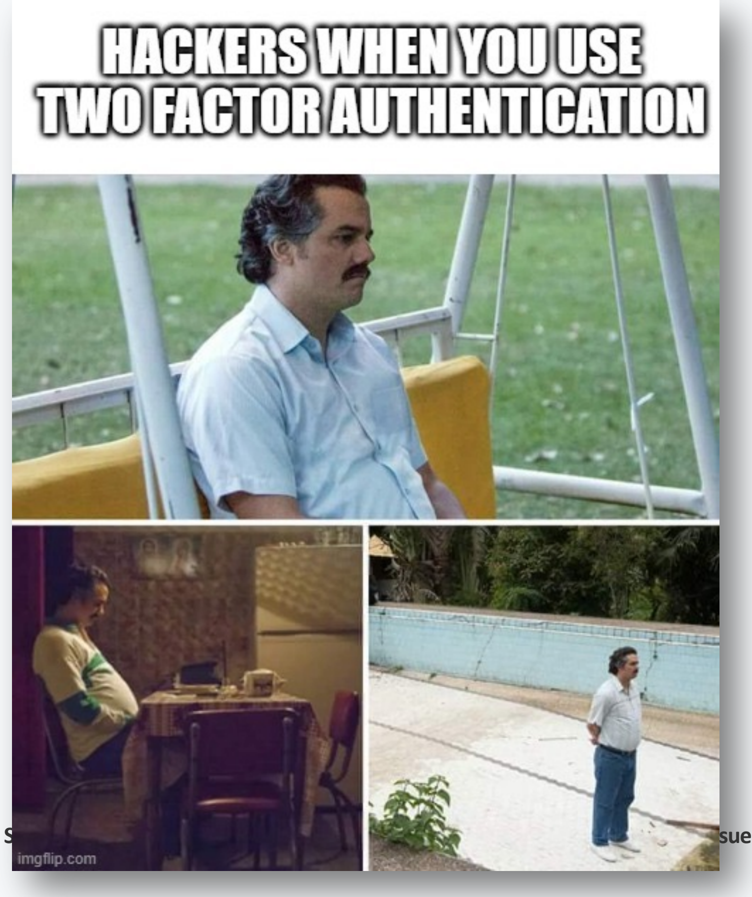
- CSPM (Cloud Security Posture Management)
  - Scans based on a custom AWS Technical Security Standard for each AWS service used
- CNAPP – all-in-one cloud security tool/service

Configuration Hardening

Blog Post - Christophe Tafani-Dereeper
Shifting Cloud Security Left — Scanning Infrastructure as Code for Security Issues

- CSPM (Cloud Security Posture Management)
  - Scans based on a custom AWS Technical Security Standard for each AWS service used
- CNAPP – all-in-one cloud security tool/service
- IaC scans for Terraform & Terragrunt
- Enforce commit signing (YubiKeys)

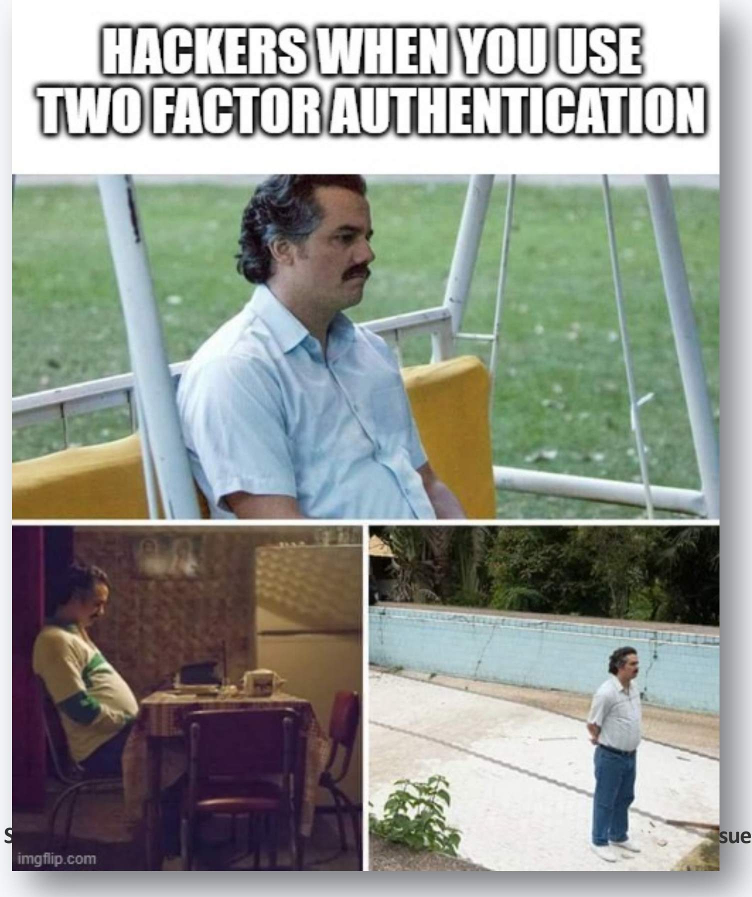
Configuration Hardening



S sues

- CSPM (Cloud Security Posture Management)
  - Scans based on a custom AWS Technical Security Standard for each AWS service used
- CNAPP – all-in-one cloud security tool/service
- IaC scans for Terraform & Terragrunt
- Enforce commit signing (YubiKeys)
- MFA everything

Configuration Hardening



S sues

- CSPM (Cloud Security Posture Management)
  - Scans based on a custom AWS Technical Security Standard for each AWS service used
- CNAPP – all-in-one cloud security tool/service
- IaC scans for Terraform & Terragrunt
- Enforce commit signing (YubiKeys)
- MFA everything
- IAM policy IP restrictions for actions from within a VPC
  - aws:SourceIp does not apply
  - aws:VpcSourceIp, aws:SourceVpc(e)
- EKS IRSA roles
  - Implement a central repository with approval from appropriate stakeholders including security

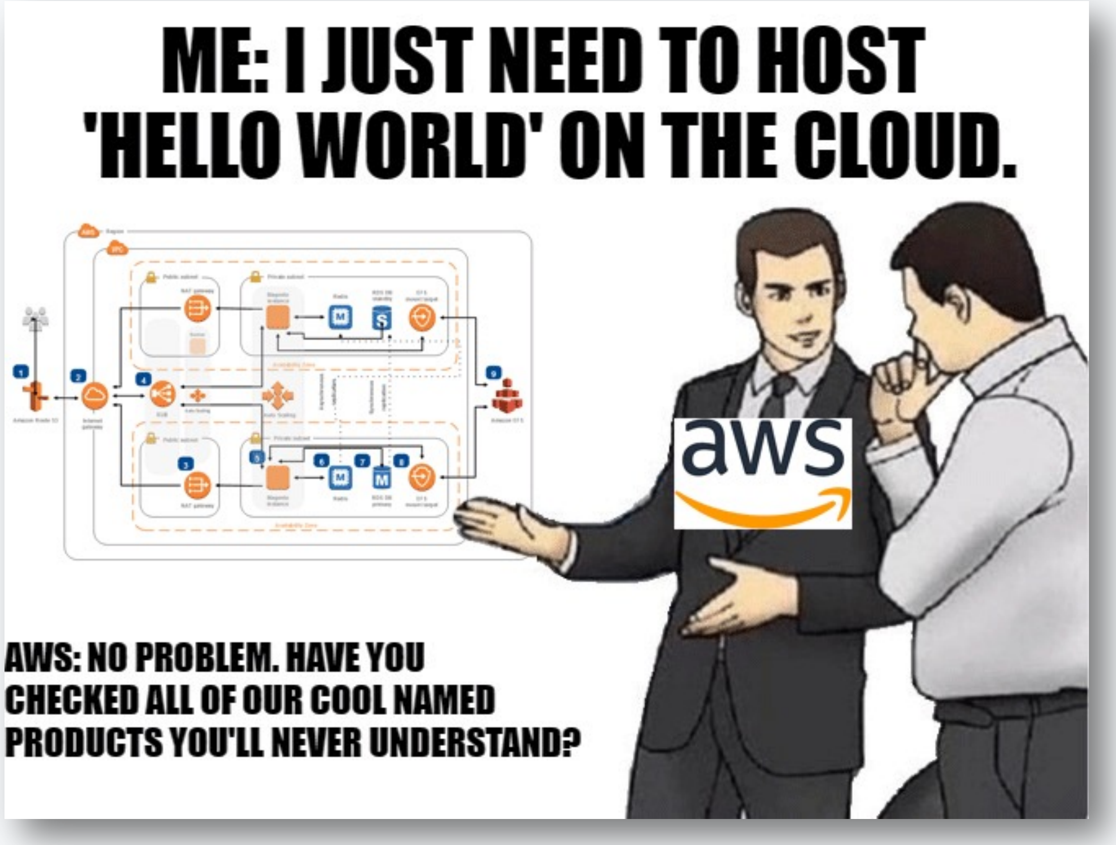
Configuration Hardening

- CSPM (Cloud Security Posture Management)
  - Scans based on a custom AWS Technical Security Standard for each AWS service used
- CNAPP – all-in-one cloud security tool/service
- IaC scans for Terraform & Terragrunt
- Enforce commit signing (YubiKeys)
- MFA everything
- IAM policy IP restrictions for actions from within a VPC
  - aws:SourceIp does not apply
  - aws:VpcSourceIp, aws:SourceVpc(e)
- EKS IRSA roles
  - Implement a central repository with approval from appropriate stakeholders including security
- Bastion access with SSM (AWS SSM Sessions Manager)
- Use IAM Attribute Based Access Control (ABAC)

Takeaways & Recommendations

- Limit or remove permanent human access to production
  - Use only just-in-time access for production (at least for privileged actions)
- Manage everything as code
  - With versioning
  - Enforce commit signing
  - Enforce approval(s) with pull requests
  - Reject PR if changed after approval
- Avoid AWS access keys as much as possible
  - Delete keys and users if not used for, ex:90 days
- Observability and alerting are crucial for Cloud Security
  - CSPM/CNAPP
  - SIEM
  - CloudQuery (CMDB)
- Weekly security meetings with SRE and Architecture teams are a must
- Thoroughly document security incident playbooks
- Automate incident response actions
- Train, train and retrain everyone one AWS, especially on IAM (roles, resource policies, etc.)

Questions


ME: I JUST NEED TO HOST 'HELLO WORLD' ON THE CLOUD.
aws
AWS: NO PROBLEM. HAVE YOU CHECKED ALL OF OUR COOL NAMED PRODUCTS YOU'LL NEVER UNDERSTAND?