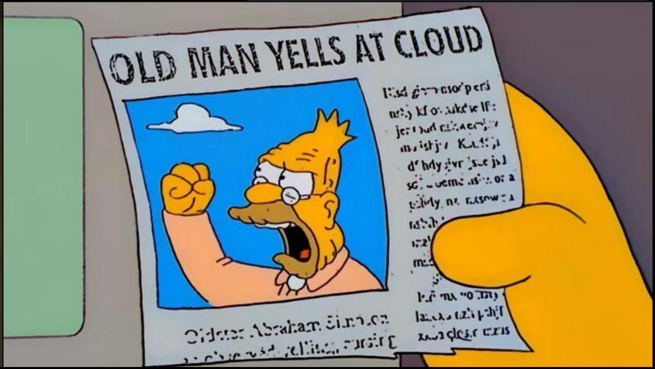
Governing the Ungovernable

Leveraging the Cloud to Secure the Cloud

Chris Farris
PrimeHarbor Technologies

Who Am I?

- Built the cloud security programs for some media companies
- Founder: fwd:cloudsec conference
- Rants a lot on ~~Twitter~~ Mastodon and Bluesky
- Somehow was named a Security Hero by AWS
- Cloud Security Consultant

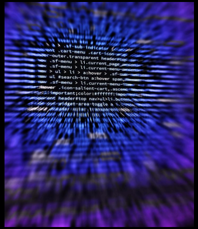


Security spectrum

Invariants live here

Educated and empowered developers

Architectural and design reviews



IaC scanning

Prevention

Auto remediation

Spreadsheet hell

Time before appearing in production

Time after appearing in production

A security invariant is a system property that relates to the system's ability to prevent security issues from happening. Security invariants are statements that will always hold true for your business and applications. – AWS

Why invariants matter

- Most security incidents are due to common mistakes, not complex attacks
- Invariants reduce developer burden
  - No backlog
  - No battles
  - Nothing to integrate or add to code
- Invariants reduce security burden
  - Fewer incidents
  - Fewer issues to chase

Choose your guardrail

IaC scanning

Auto remediation

Service control policies
Resource control policies

We're here today

What makes good invariants

Specific

Includes all actions, principals, and conditions

Enforceable

Can be enforced via policy, code, or automation tooling

Realistic

Reflects real needs and won't break needed business/ops

Avoid exceptions

Exceptions are part of the invariant, not dealt with manually

. . . will always hold true . . .

“No one can create a VPC”

vs.

“Only the network engineering team can create a VPC”

Examples

- “Only the network engineering team may create a VPC, alter route tables, or attach an IGW”
- “Only the security and privacy team may make an S3 bucket public”
- “Only procurement may subscribe to or accept an offer in AWS Marketplace”
- “Only cloud engineering can enable new opt-in regions”

Enforcing invariants

Organization-based policies

- Service control policies
- Resource control policies
- Declarative policies

Identity-based policies

- Permission policies
- Permission boundaries

Automation/guardrails

- Declarative controls (Block Public Access)
- Automated remediation

Service control policies

Managed via the AWS Organizations management account (aka “payer”)

Defines the “maximum permissions of the account”

(This includes the root user)

Applies to **your identities**

Resource control policies

NEW!!!

Managed via the Organizations Management Account (aka “payer”)

Applies to all principals – every AWS Customer

Only some services for now:

S3, STS/IAM, SQS, Secrets Manager

SCPs, RCPs, and permissions boundaries don't grant permissions, they define the maximum permissions available

Declarative policies

NEWER!!!

Managed via Organizations

But not IAM policies

Enforced at the service's control plane

This exists outside of IAM

Supports:

- EBS Snapshots
- AMI
- VPC
- IMDSv2

Permissions Boundaries

Managed in the account as a normal IAM Policy

Must be attached to each User & Role

Defines the “maximum permissions" of the User or Role

Managed by Identity Center for roles managed by Identity Center

Prerequisites

AWS Organizations

Never in a workload account

AWS IAM Identity Center

Tie this with your corporate identity system

Infrastructure as code

Critical for :

- Auditability
- Transparency
- Reproducibility

How to build an IAM Invariant

1. Define invariant – plain language
2. Determine actions
3. Determine resources
4. Determine “principals” (if SCP)
5. Determine conditions/define the exceptions

Define invariant in plain language

“Only the security and privacy team may make an Amazon S3 bucket public”

- Specific – “. . . make an Amazon S3 bucket public”
- Enforceable – Use S3 Block Public Access with SCP
- Realistic – Teams can create buckets,
  they cannot remove the default BPA
- Avoids exceptions – “Only the security and privacy team . . .”

Examples

Manage Regions: SCP Invariant

Only Cloud Engineering can enable new opt-in regions after ensuring GRC sign-off and the implementation of appropriate security telemetry and governance controls.

Why?

GuardDuty and STS are regional. If you don't setup the new region in your security account, you've got a detection gap.

Manage Regions: SCP Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventAccountRegionChanges",

      "Effect": "Deny",
      "Action": [
           "account:EnableRegion",
           "account:DisableRegion"
       ],
      "Resource": ["*"]
      }
    }
  ]
}
```

Explicit Deny

Actions to enable or disable regions

Manage Regions: SCP Policy Exception

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Sid": "PreventAccountRegionChanges",
    "Effect": "Deny",
    "Action": ["account:EnableRegion","account:DisableRegion"],
    "Resource": ["*"],
    "Condition": {
      "StringNotLike": {
        "aws:PrincipalArn": [
         "arn:aws:iam::*:role/OrganizationAccountAccessRole",
         "arn:aws:iam::*:role/aws-reserved/sso.amazonaws.com/AWSReservedSSO_Admi
         nistratorAccess*"
        ]
      }
    }
  }
  ]
}
```

Define the two roles permitted to manage regions

Protect Sensitive Roles: SCP Invariant

Only the automated account management processes can alter security and cloud management roles in member accounts.

Why?

Account owners (or threat actors) with full admin access could impact core security and management functionality.

Protect Sensitive Roles: SCP Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PreventRoleModification",
            "Effect": "Deny",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DeleteRole",
                "iam:DeleteRolePermissionsBoundary",
                ...
            ],
            "Resource": [
                "arn:aws:iam::*:role/security-audit",
                "arn:aws:iam::*:role/OrganizationAccountAccessRole",
                "arn:aws:iam::*:role/stacksets-exec-*",
                "arn:aws:iam::*:role/aws-reserved/sso.amazonaws.com/AWSReservedSSO*"
            ],
            "Condition": {
                "StringNotLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::*:role/stacksets-exec-*",
                        "arn:aws:iam::*:role/OrganizationAccountAccessRole"
                    ]
                }
            }
        }
    ]
}
```

Actions (not inclusive list)

Roles to protect

Allow Stacksets and OrganizationAccountAccessRole to alter roles

S3 Data Perimeter: RCP Invariant

"Only approved S3 Buckets may be public or shared outside of the Organization"

Why?

Avoid This:

ACHIEVEMENT UNLOCKED!
S3 Bucket Negligence Award
You have failed to adequately safeguard the data with which you were entrusted. You have failed those who relied upon you.

S3 Data Perimeter: RCP Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3DataPerimeterWithApprovedExceptions",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringNotEqualsIfExists": {
                    "aws:PrincipalOrgID": "MY_ORG_ID"
                },
                "BoolIfExists": {
                    "aws:PrincipalIsAWSService": "false"
                }
            }
        }
    ]
}
```

All RCPs must have these

Apply this to all actions

Don't apply the Deny if the principal is in my organization, or if the principal is an AWS service (like CloudTrail or Athena)

S3 Data Perimeter: RCP Policy with exceptions

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3DataPerimeterWithApprovedExceptions",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "NotResource": [
                "arn:aws:s3:::my-public-bucket/*"
            ],
            "Condition": {
                "StringNotEqualsIfExists": {
                    "aws:PrincipalOrgID": "MY_ORG_ID"
                },
                "BoolIfExists": {
                    "aws:PrincipalIsAWSService": "false"
                }
            }
        }
    ]
}
```

Define your list of approved buckets

IAM Access Analyzer can be used to generate the list of externally accessible buckets.

No Shared EBS or AMI: Declarative Invariant

AMIs and SnapShots may never be shared with all AWS accounts.

Why?

Most companies don't need to do this and snapshots and AMIs usually have sensitive information. You wouldn't give randos your server hard drives would you?

No Shared EBS or AMI: Declarative Policy

```
{
  "ec2_attributes": {
    "exception_message": {
      "@@assign": "Sharing of AMIs is denied by Organizational Policy"
    },
    "image_block_public_access": {
      "state": {
        "@@assign": "block_new_sharing"
      }
    },
    "snapshot_block_public_access": {
      "state": {
        "@@assign": "block_all_sharing"
      }
    }
  }
}
```

No Shared EBS or AMI: Declarative Policy Exception

- Unlike Authorization Policies (SCP, RCPs) these Management Policies don't inherit the same way.
- You can override the block on a specific account

```
{
  "ec2_attributes": {
    "image_block_public_access": {
      "state": {
        "@@assign": "unblocked"
      }
    },
    "snapshot_block_public_access": {
      "state": {
        "@@assign": "unblocked"
      }
    }
  }
}
```

Prevent Account Closure: Permission Boundary

"Only the Cloud Engineering team may close an AWS account"

Why?

Bad things happen if you accidentally close prod.

This typically applies when you have lots of folks in your payer account.

Prevent Account Closure: Permission Boundary

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListForAllButCloudAdmin",
            "Effect": "Allow",
            "NotAction": [
                "organizations:AttachPolicy",
                "organizations:CloseAccount",
                "organizations:Create*",
                "organizations:Delete*",
                "organizations:DetachPolicy",
                "organizations:Disable*",
                "organizations:Enable*",
                "organizations:InviteAccountToOrganization",
                "organizations:LeaveOrganization",
                "organizations:MoveAccount",
                "organizations:PutResourcePolicy",
                "organizations:RemoveAccountFromOrganization",
                "organizations:Update*",
                "organizations:InviteAccountToOrganization",
           ],
            "Resource": ["*"]
        }
```

Allow on NotAction says "Everything but these few actions"

Remember this doesn't grant permissions, rather defines **maximum permissions**

Prevent Account Closure: Boundary exception

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudAdminEverything",
      "Effect": "Allow",
      "Action": ["*"],
      "Resource": ["*"],
      "Condition": {
        "ArnLike": {
          "aws:PrincipalArn": [
            "arn:aws:iam::*:role/CLOUDADMIN"
          ]
        }
      }
    },
```

Because no explicit deny was present, this layers on additional permissions based on the PrincipalArn

Permissions Boundary Caveat

Permissions Boundaries must be applied to every principal to make this work.

Leverage EventBridge Rules to apply to any new Role/User when they are created or if the boundary is removed

https://github.com/primeharbor/pht-payer-invariants

Know your limits - SCPs & RCPs

SCPs (and RCPs) have a number of limits:

- Policy size cannot exceed 5,120 bytes
  - Including whitespace!
- You can have up to five SCPs and 5 RCPs per OU level
  - And up to 5 levels of OUs
- You must include the "FullAWSAccess" at each level
  - You can have up to ~~five~~ four SCPs/RCPs per OU level
- No more than 2,000 policies per organization

Know your limits - Declarative Policies

- Management Policies have looser restrictions.
- Declarative Policies only support some EC2 capabilities (today)
- Complex inheritance rules - RTFM or FAFO

Know your limits - Permissions Boundaries

- Avoid "Deny" statements if you want to layer permissions
- One Boundary policy per principal
- Doesn't apply to IAM Groups

Putting it all together

Guardrails are like nuclear power. One accident, and suddenly everyone is against the idea.

Chris Farris

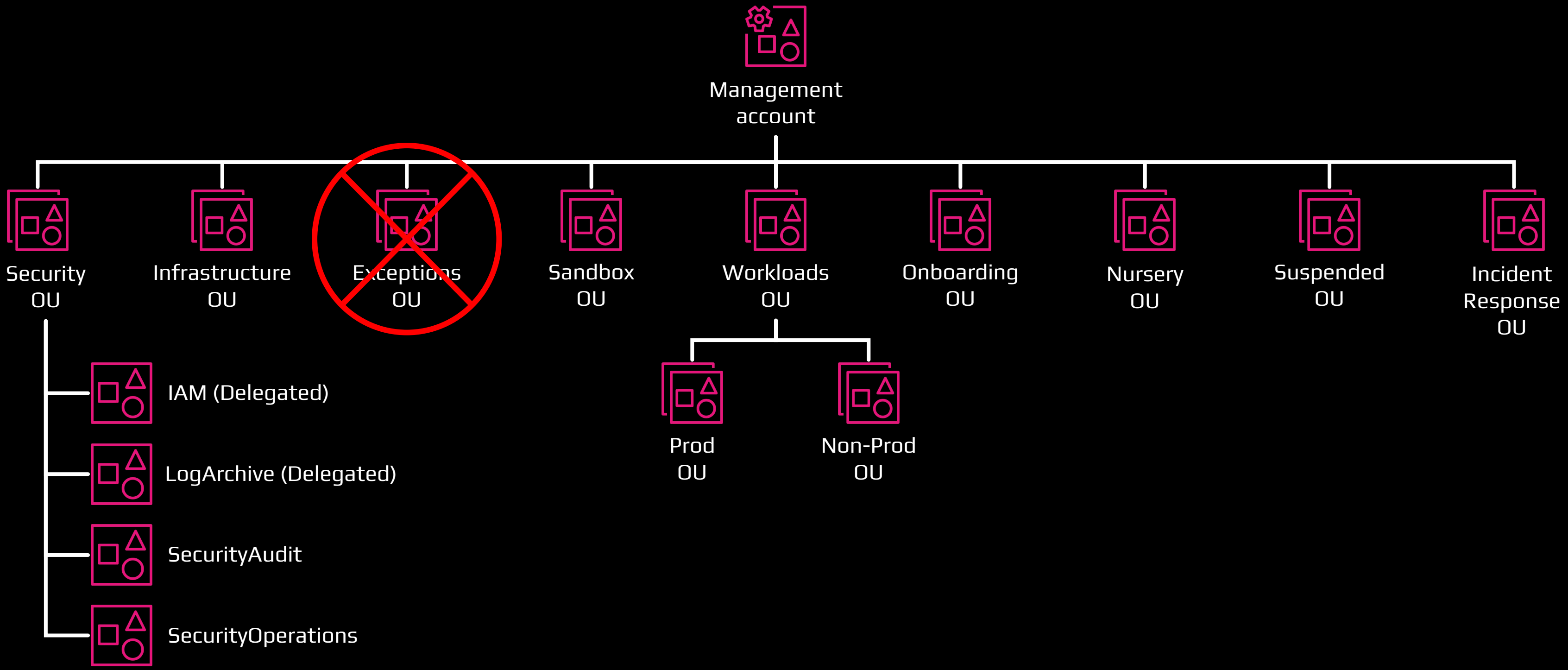
Here be dragons

- AWS provides no ability to test/audit service control policies
- You need to leverage your SIEM
  - Query for the actions you intend to block
  - Look at the conditions
  - Determine if the action should have been allowed
- Have a conversation with the builder

Maintenance

- Manage this via infrastructure as code
- Invariants should be well communicated
  - GitHub “internal” repos are good for this
- Understand the trust boundaries for your pipeline
  - Can GitHub administrators, who don’t have permission to the org management account, have the capability to alter invariants?

Organization hierarchy



Resources

Blog: Defining Security Invariants

Blog: Implementing Security Invariants in an AWS Management Account

GitHub Repo: aws-organizational-policies

GitHub Repo: pht-payer-invariants

AWS Docs: Service Authorization Reference

Actions, resources, and condition keys for AWS services

QUESTIONS?

I'm available for consulting!
https://pht.us

@jcfarris [@infosec.exchange]
https://github.com/jchrisfarris
https://www.linkedin.com/in/jcfarris
http://www.chrisfarris.com

https://www.primeharbor.com
https://pht.us/invariants
https://pht.us/LondonSecEng