AWS re:Invent

DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

DEV401

Security invariants: From enterprise chaos to cloud order

Chris Farris
Security Hero
PrimeHarbor

Rich Mogull
Security Sidekick (Community Builder)
Securosis

Agenda

What are security invariants?

What makes a good invariant

How to write a good invariant

How to **not** write a good invariant (let gen AI do it for you)

Applying invariants

Choose your guardrail

IaC scanning

Auto remediation

Service control policies

↑

We’re here today

> 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

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

What makes a good invariant?

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

Identity-based policies

- Permission policies
- Permission boundaries

Automation/guardrails

- Declarative controls (Block Public Access)
- Declarative policies
- 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

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

Prerequisites

Prerequisites

AWS Organizations

Never in a workload account

AWS IAM Identity Center

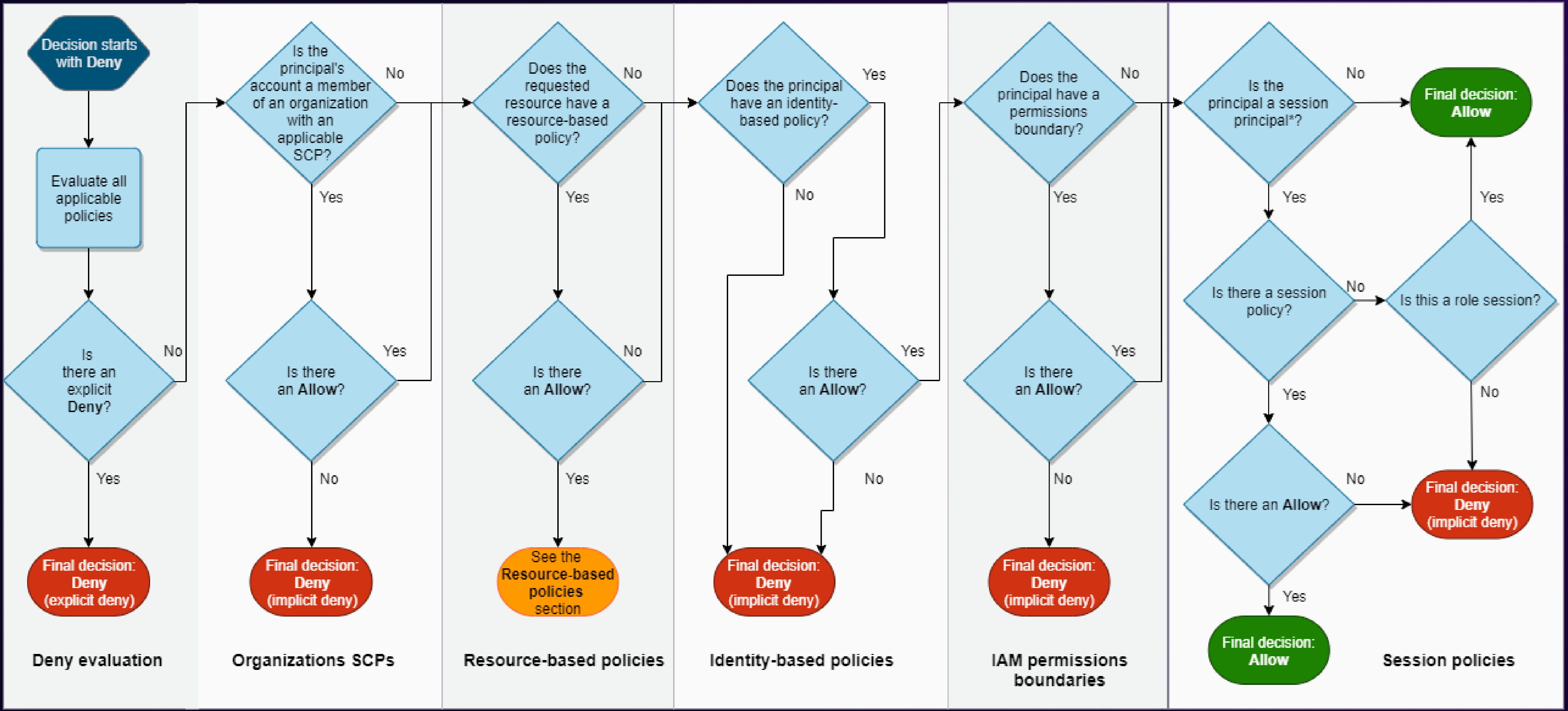
Tie this with your corporate identity system

Infrastructure as code

Critical for :

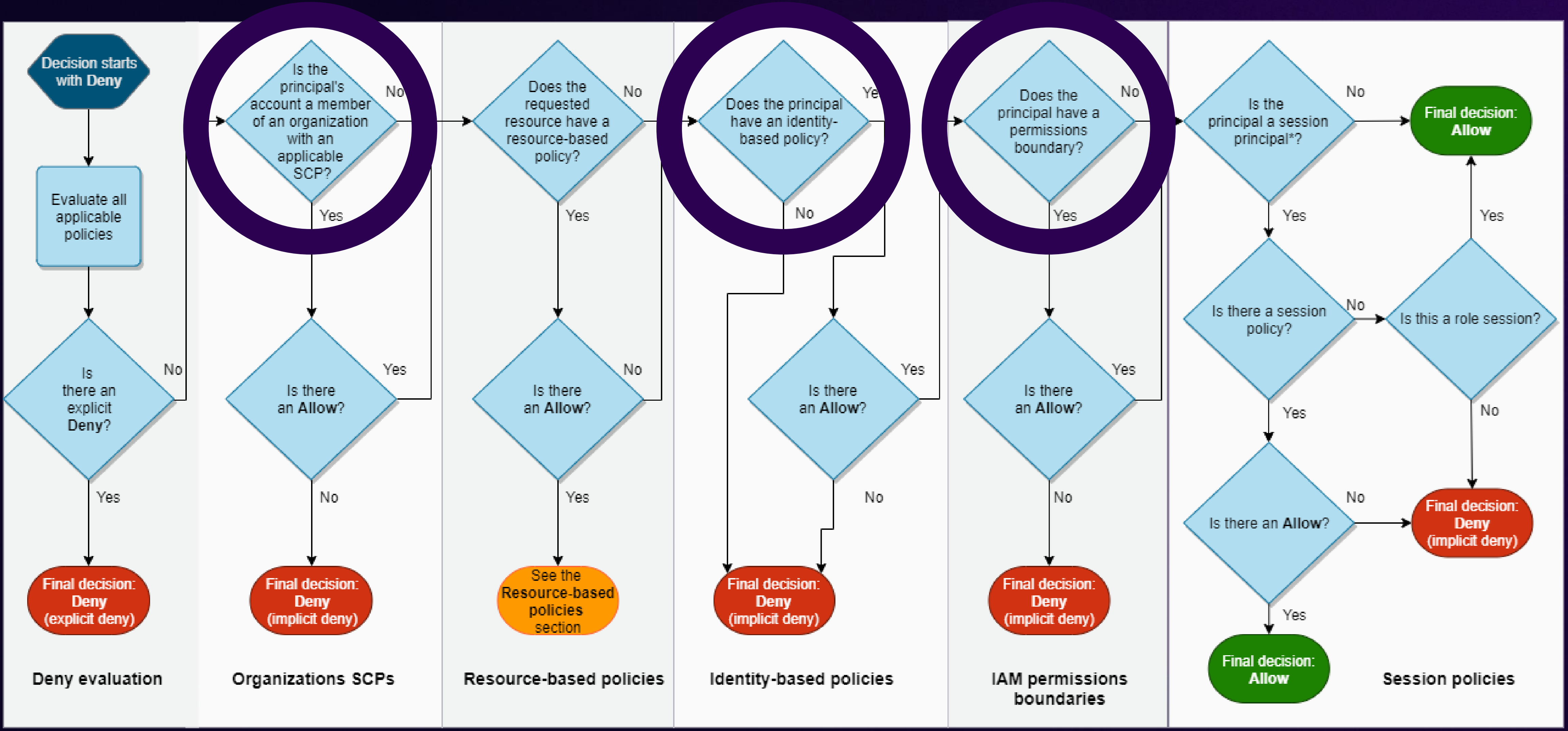
- Auditability
- Transparency
- Reproducibility

Organization and identity policies: Evaluation


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

*A session principal is either a role session or an IAM federated user session.

Organization and identity policies: Evaluation



Decision starts with Deny

Evaluate all applicable policies

Is there an explicit Deny?

Yes

No

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

*A session principal is either a role session or an IAM federated user session.

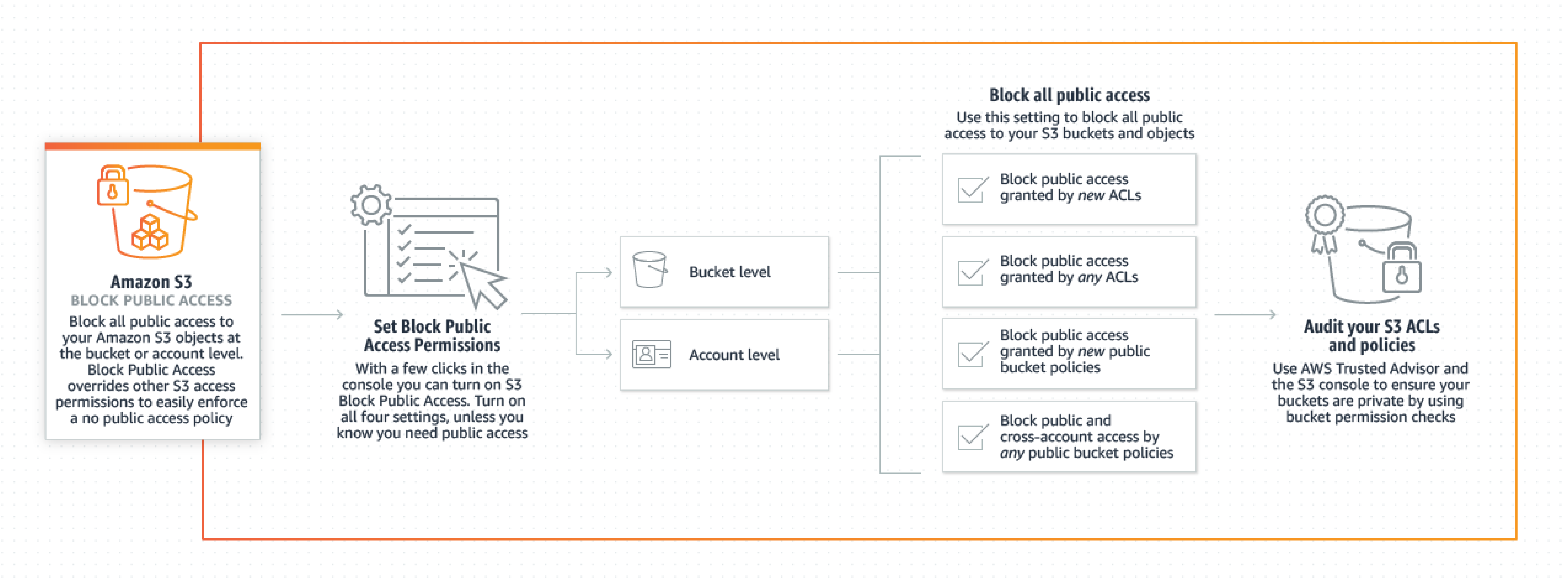
SCPs, RCPs, and permissions boundaries don’t grant permissions, they define the **maximum permissions available**

Declarative and other controls

- Block Public Access
  - Amazon S3, Amazon EBS snapshots, AMIs, VPCs
- Default Amazon EBS encryption
- Delegated administration
- IMDSv2 requirements

These can work in conjunction with service control policies

S3 Block Public Access



Amazon S3

BLOCK PUBLIC ACCESS

Block all public access to your Amazon S3 objects at the bucket or account level. Block Public Access overrides other S3 access permissions to easily enforce a no public access policy

Set Block Public Access Permissions

With a few clicks in the console you can turn on S3 Block Public Access. Turn on all four settings, unless you know you need public access

Bucket level

Account level

Block all public access

Use this setting to block all public access to your S3 buckets and objects

✓ Block public access granted by *new* ACLs

✓ Block public access granted by *any* ACLs

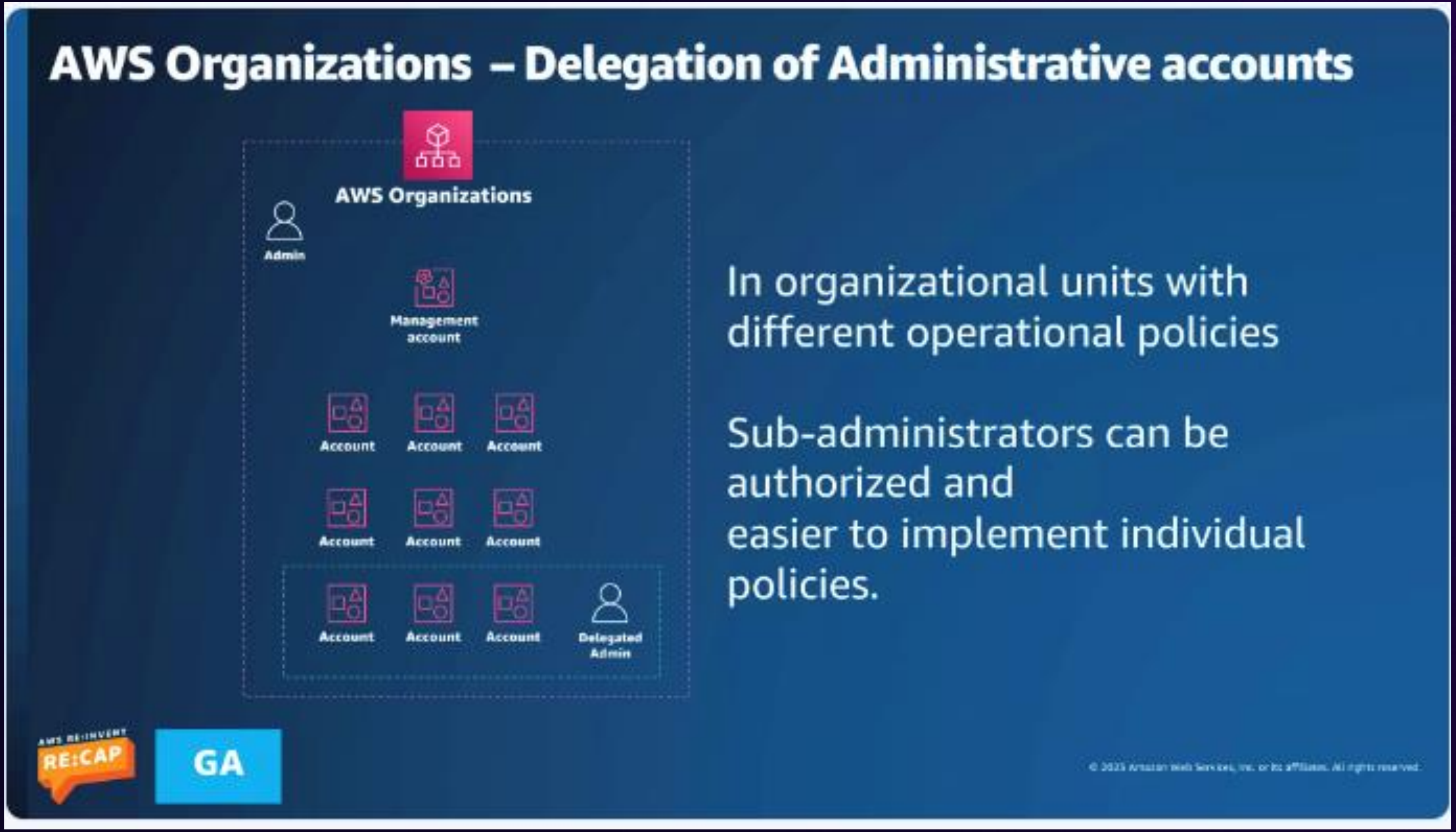
✓ Block public access granted by *new* public bucket policies

✓ Block public and cross-account access by *any* public bucket policies

Audit your S3 ACLs and policies

Use AWS Trusted Advisor and the S3 console to ensure your buckets are private by using bucket permission checks

Delegated admin



AWS Organizations – Delegation of Administrative accounts

In organizational units with different operational policies

Sub-administrators can be authorized and easier to implement individual policies.

How to write a good invariant

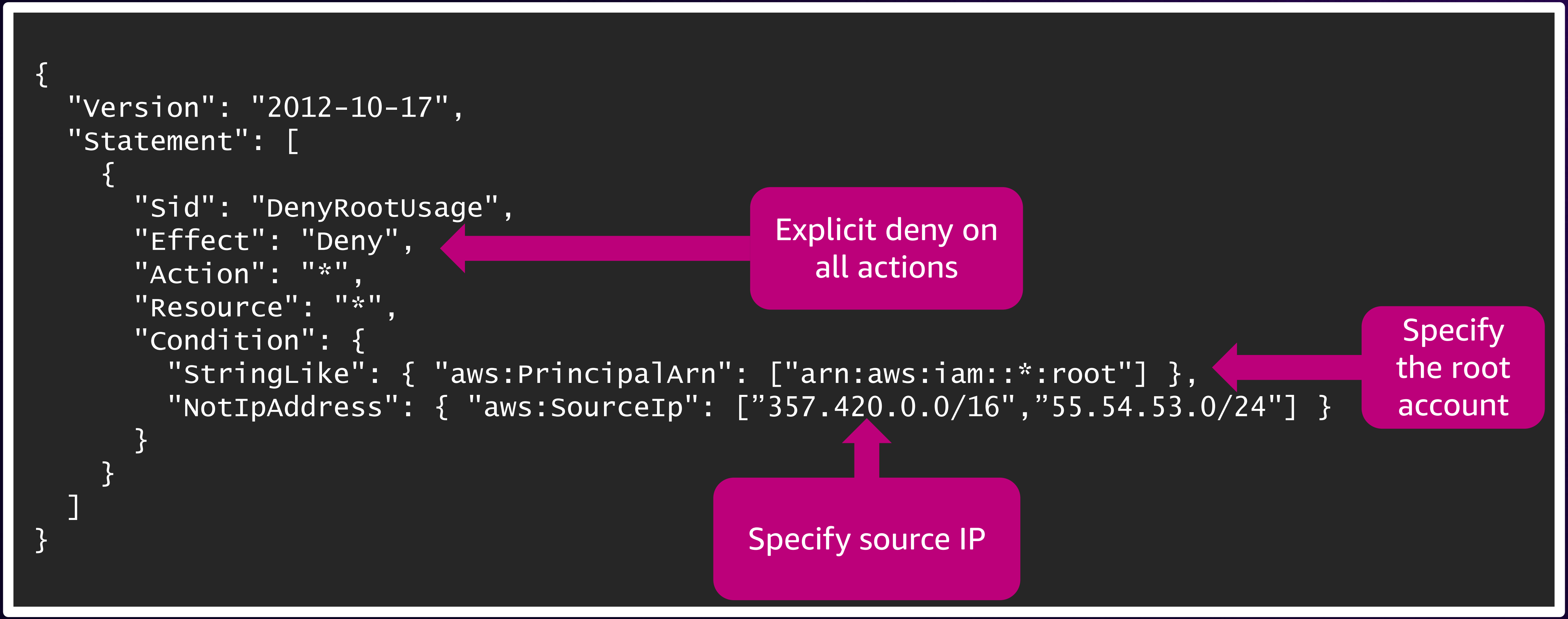
Getting started

- Personas
  - Use IAM Identity Center, please!
- S3 Block Public Access
  - Every account, every region
  - Enforce this at account creation
- Delegated admin
  - Configure it for all the services you use

SCP components

- Effect: Deny
- Resource: "*"
- Action: List of things you want to prevent
- Conditions: This is where the **magic** happens
- **The allows needed to not unintentionally break things**

SCP: You can only log in as root from the corporate VPN/office



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyRootUsage",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringLike": { "aws:PrincipalArn": ["arn:aws:iam::*:root"] },
        "NotIpAddress": { "aws:SourceIp": [”357.420.0.0/16",”55.54.53.0/24"] }
      }
    }
  ]
}
```

Explicit deny on all actions

Specify the root account

Specify source IP

How to build an SCP/permissions boundary

Define invariant – plain English

Determine actions

Determine resources

Determine “principals” (if SCP)

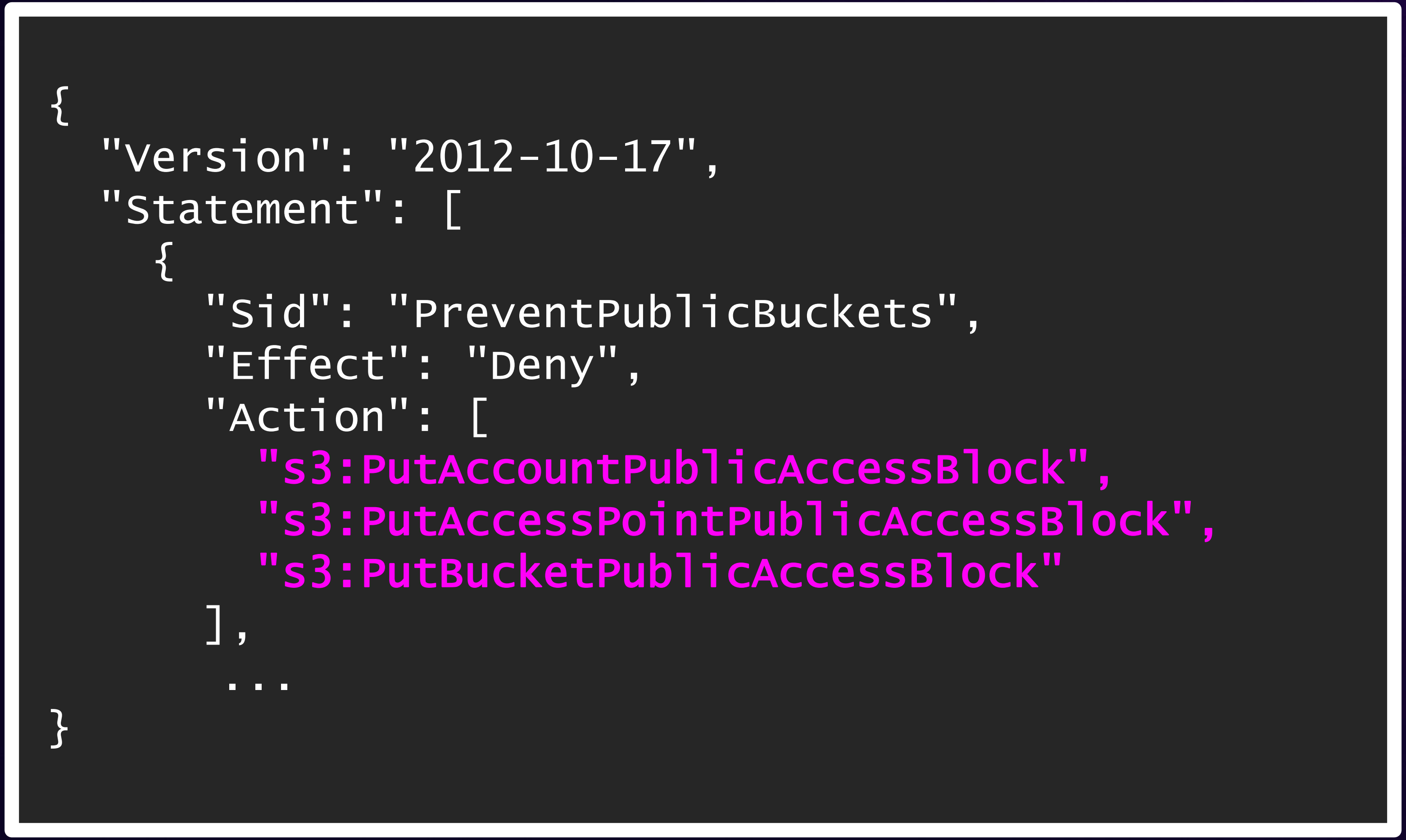
Determine conditions/define the exceptions

Define invariant in plain English

- **“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 . . .”

Determine actions



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventPublicBuckets",
      "Effect": "Deny",
      "Action": [
        "s3:PutAccountPublicAccessBlock",
        "s3:PutAccessPointPublicAccessBlock",
        "s3:PutBucketPublicAccessBlock"
      ],
       ...
}
```

Determine resources

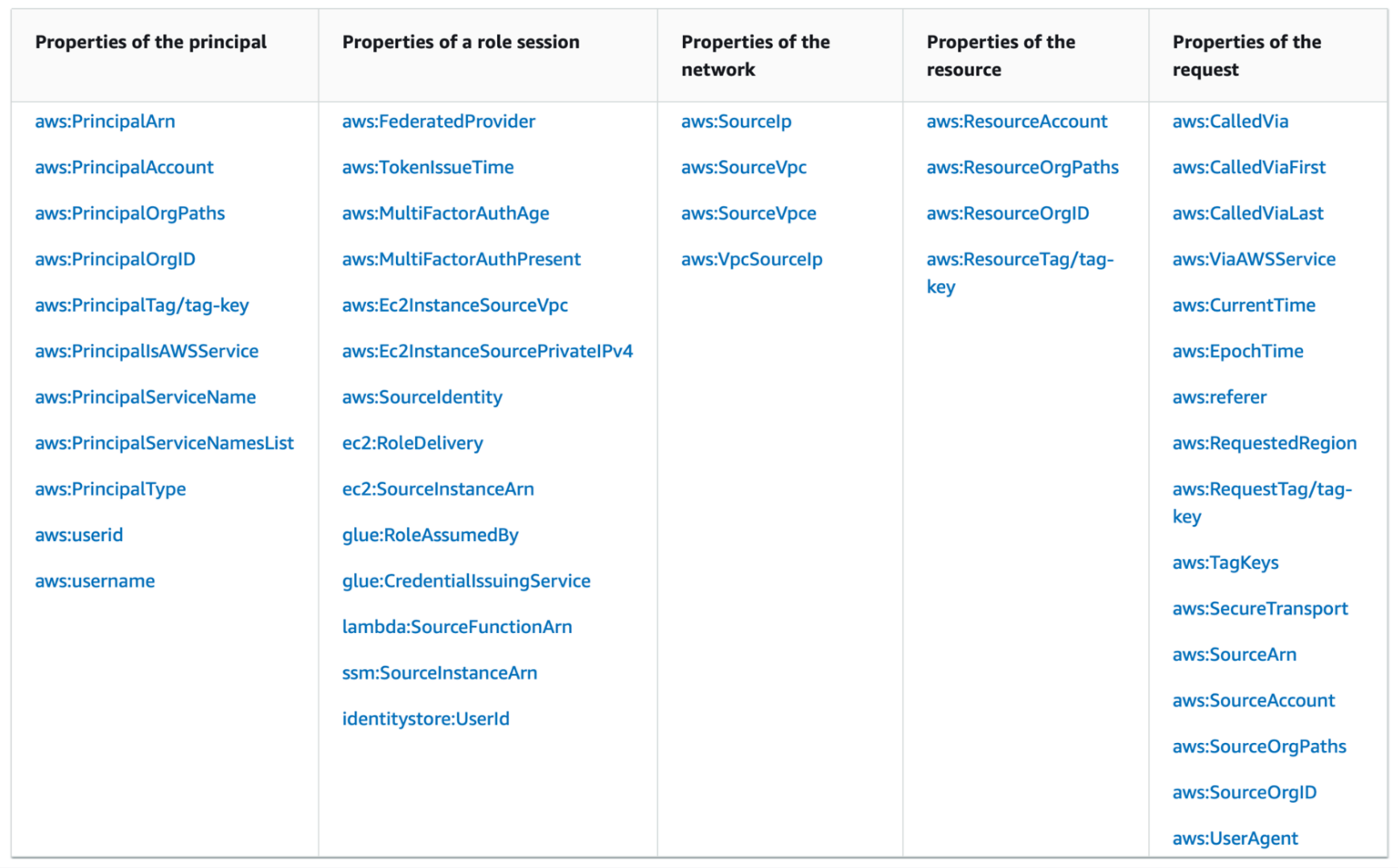
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventPublicBuckets",
      "Effect": "Deny",
      "Action": [
        "s3:PutAccountPublicAccessBlock",
        "s3:PutAccessPointPublicAccessBlock",
        "s3:PutBucketPublicAccessBlock"
      ],
      "Resource": "*",
       ...
}
```

Determine conditions/define exceptions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventPublicBuckets",
      "Effect": "Deny",
      "Action": [
        "s3:PutAccountPublicAccessBlock",
        "s3:PutAccessPointPublicAccessBlock",
        "s3:PutBucketPublicAccessBlock"
      ],
      "Resource": "*",
        "Condition": {
          "StringNotLike": {
            "aws:PrincipalArn": [
              "arn:aws:iam::*:role/aws-reserved/sso.amazonaws.com/AWSReservedSSO_CloudSecurity_*",
              "arn:aws:iam::*:role/aws-reserved/sso.amazonaws.com/AWSReservedSSO_PrivacyAnalyst_*"
            ]
          }
        }
 }]}
```

Global condition context keys

CRITICAL FOR GOOD INVARIANTS



Properties of the principal	Properties of a role session	Properties of the network	Properties of the resource	Properties of the request
aws:PrincipalArn	aws:FederatedProvider	aws:SourceIp	aws:ResourceAccount	aws:CalledVia
aws:PrincipalAccount	aws:TokenIssueTime	aws:SourceVpc	aws:ResourceOrgPaths	aws:CalledViaFirst
aws:PrincipalOrgPaths	aws:MultiFactorAuthAge	aws:SourceVpce	aws:ResourceOrgID	aws:CalledViaLast
aws:PrincipalOrgID	aws:MultiFactorAuthPresent	aws:VpcSourceIp	aws:ResourceTag/tag-key	aws:ViaAWSService
aws:PrincipalTag/tag-key	aws:Ec2InstanceSourceVpc			aws:CurrentTime
aws:PrincipalIsAWSService	aws:Ec2InstanceSourcePrivateIPv4			aws:EpochTime
aws:PrincipalServiceName	aws:SourceIdentity			aws:referer
aws:PrincipalServiceNamesList	ec2:RoleDelivery			aws:RequestedRegion
aws:PrincipalType	ec2:SourceInstanceArn			aws:RequestTag/tag-key
aws:userid	glue:RoleAssumedBy			aws:TagKeys
aws:username	glue:CredentialIssuingService			aws:SecureTransport
	lambda:SourceFunctionArn			aws:SourceArn
	ssm:SourceInstanceArn			aws:SourceAccount
	identitystore:UserId			aws:SourceOrgPaths
				aws:SourceOrgID
				aws:UserAgent

Key condition keys for invariants

- Principal
  - aws:PrincipalArn
  - aws:PrincipalAccount
  - aws:PrincipalOrgID
- Role session
  - aws:FederatedProvider
  - aws:Ec2InstanceSourceVpc
- Network
  - All
- Resource
  - aws:ResourceOrgID
  - aws:ResourceTag/tag-key (less reliable)
- Request
  - CalledVia
  - ViaAWSService
  - SourceArn
  - SourceAccount
  - SoureOrg/OrgPaths
  - UserAgent

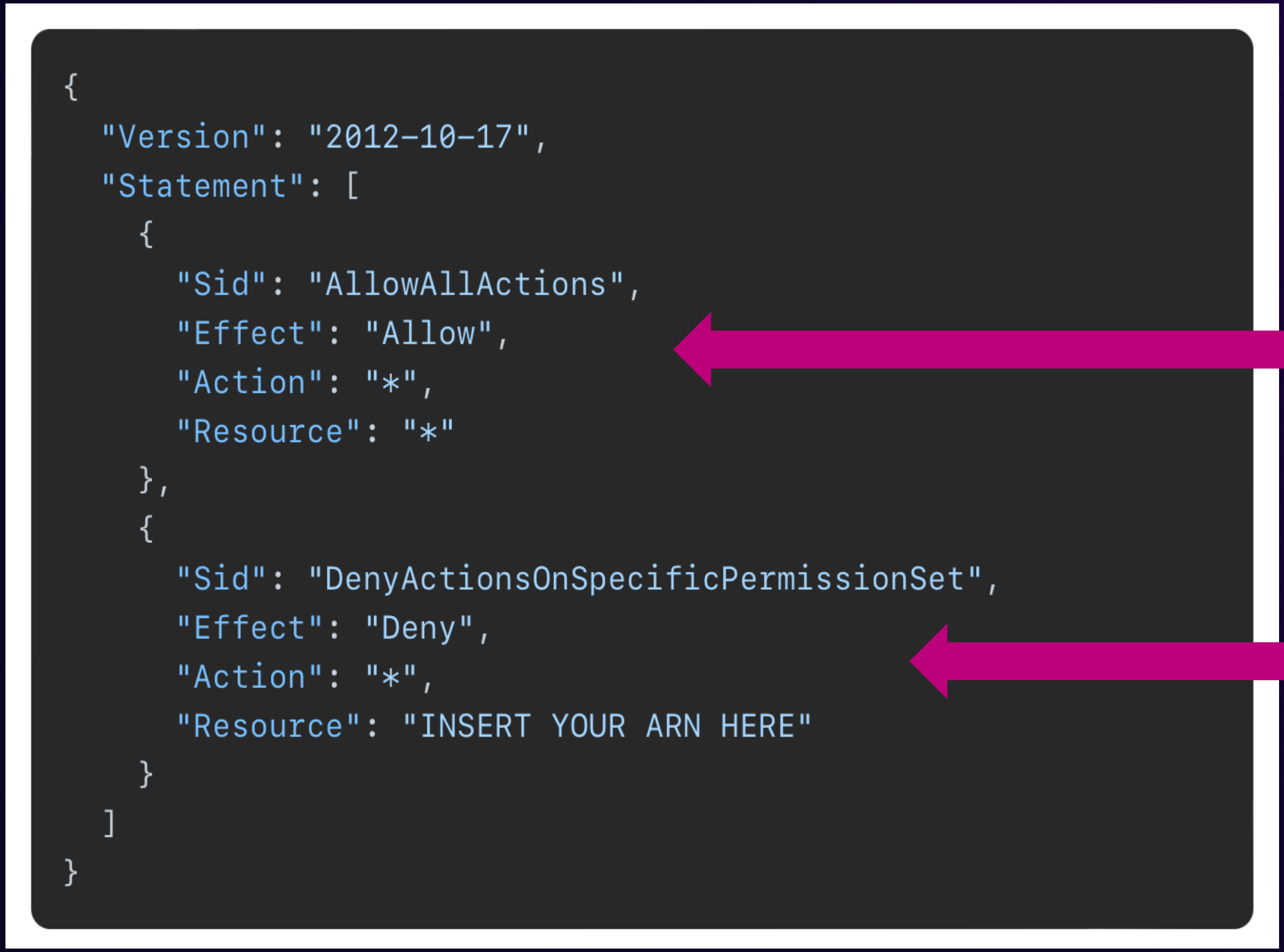
Many services also have their own condition keys!

SCP: No public lambda

```
 {
   "Sid":
"PreventPublicLambdaPolicy",
   "Effect": "Deny",
   "Action":
["lambda:AddPermission"],
   "Resource": ["*"],
   "Condition": {
     "StringEquals": {
       "lambda:Principal": ["*"]
      }
    }
 }
```

```
{
  "Sid": "PreventUnAuthFuncURL",
  "Effect": "Deny",
  "Action": [
    "lambda:CreateFunctionUrlConfig",
    "lambda:UpdateFunctionUrlConfig"
  ],
  "Resource":
"arn:aws:lambda:*:*:function/*",
  "Condition": {
     "StringNotEquals": {
        "lambda:FunctionUrlAuthType":
"AWS_IAM"
     }
   }
}
```

Permission boundary: IAM Identity Center administrators can’t expand their own permissions



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAllActions",
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    },
    {
      "Sid": "DenyActionsOnSpecificPermissionSet",
      "Effect": "Deny",
      "Action": "*",
      "Resource": "INSERT YOUR ARN HERE"
    }
  ]
}
```

Required or the default deny breaks everything

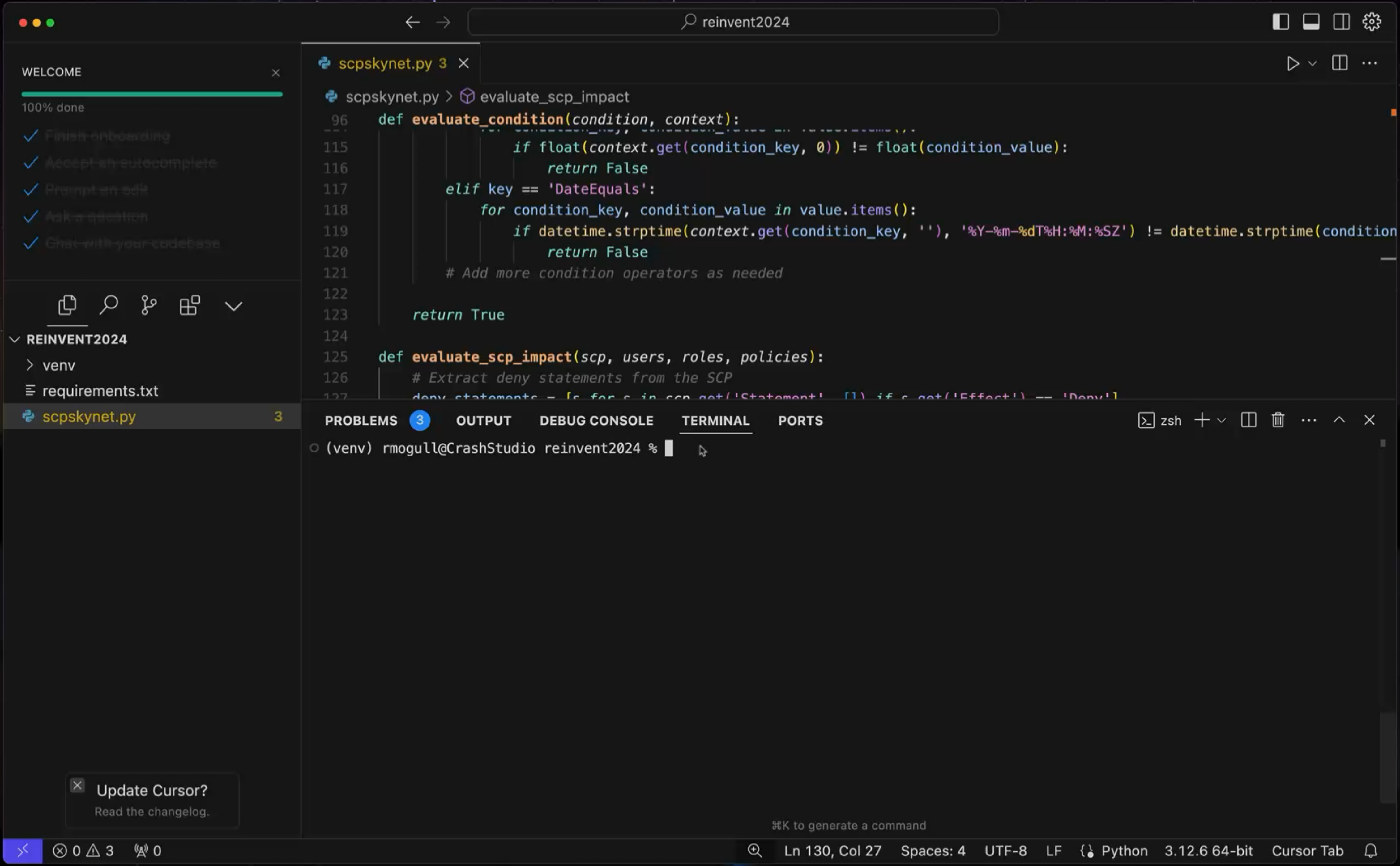
Deny modifying the admin’s own permission set

Know your limits

SCPs (and RCPs) have a number of limits:

1. Size of SCPs cannot exceed 5,120 bytes (Including whitespace!)
2. You can have up to five SCPs per OU level
   - And up to 5 levels of OUs
3. You **must** include the "FullAWSAccess" **at each level**
4. You can have up to ~~five~~ **four** SCPs per OU level
5. No more than 2,000 SCPs per organization

How to not write a good invariant



```python
def evaluate_condition(condition, context):
            for condition_key, condition_value in value.items():
                if float(context.get(condition_key, 0)) != float(condition_value):
                    return False
        elif key == 'DateEquals':
            for condition_key, condition_value in value.items():
                if datetime.strptime(context.get(condition_key, ''), '%Y-%m-%dT%H:%M:%SZ') != datetime.strptime(condition
                    return False
        # Add more condition operators as needed

    return True

def evaluate_scp_impact(scp, users, roles, policies):
    # Extract deny statements from the SCP
    deny_statements = [s for s in scp.get('Statement', []) if s.get('Effect') == 'Deny']
```

```
(venv) rmogull@CrashStudio reinvent2024 %
```

From concept to production

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

Chris Farris

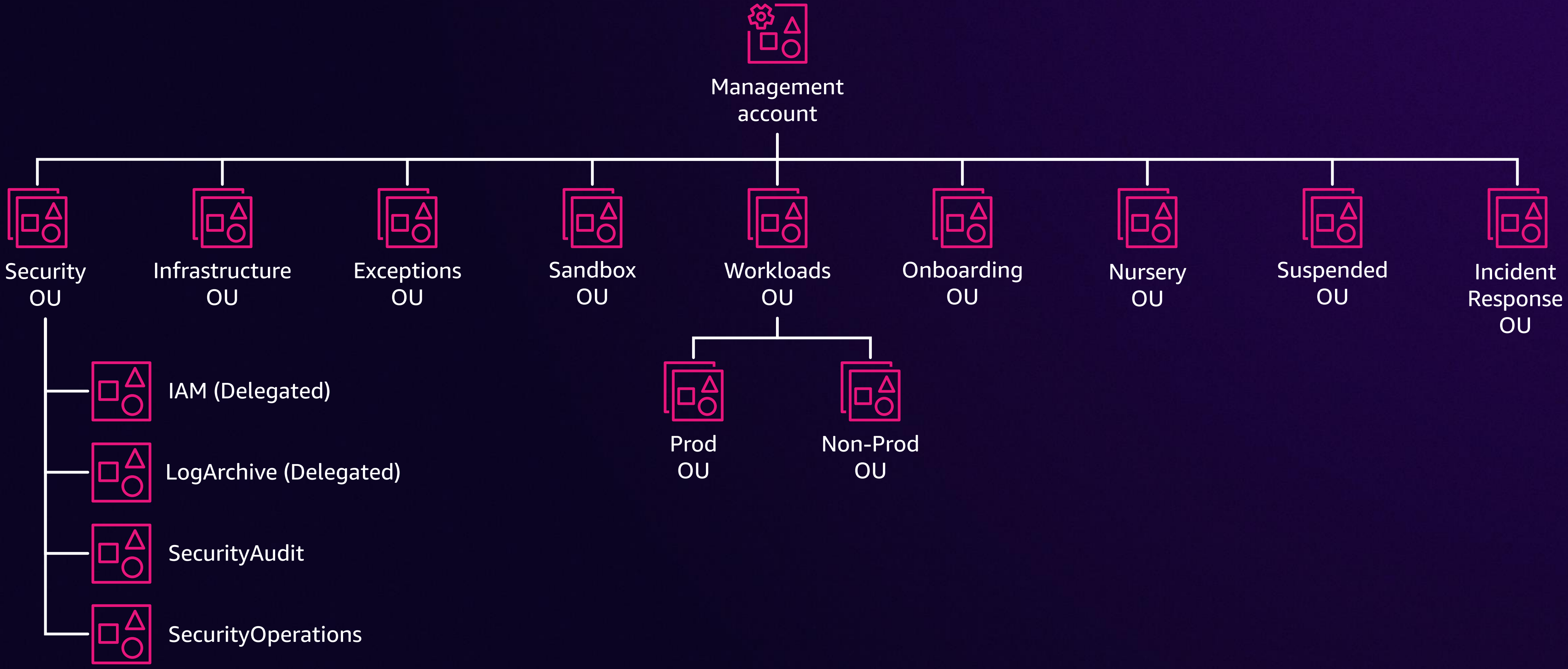
Here be dragons

- 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



Thank you!

Please complete the session survey in the mobile app

Chris Farris

@jcfarris.bsky.social
@jcfarris@infosec.exchange
www.chrisfarris.com

Rich Mogull

@rmogull.com (BlueSky)
@rmogull@defcon.social
securosis.com
slaw.securosis.com

Jen AI

aws.amazon.com/bedrock