Advanced IAM Concepts: Complex IAM Policies and Conditions

Andru Estes
Principal Author
andru-estes

Troubleshooting Overlapping IAM Policies

Know the Permission Evaluation Order

IAM will first look to see if there are any explicit deny statements in this chain.

As it does this, it will follow a simple "Allowed" or "Denied" logic tree.

Remember that there is an implicit deny for any requests!

Policy Conflicts Example

We are explicitly allowing all s3:Get and s3:List actions on all S3 resources

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:Get*", "s3:List*"],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/*"
    },
    {
      "Effect": "Deny",
      "Action": ["s3:PutObject", "s3:DeleteObject"],
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/production/*"
    }
  ]
}
```

Policy Conflicts Example

We have another explicit allow statement that is allowing the IAM identity to use an s3:PutObject API call within the super-fancy-bucket-name s3 bucket

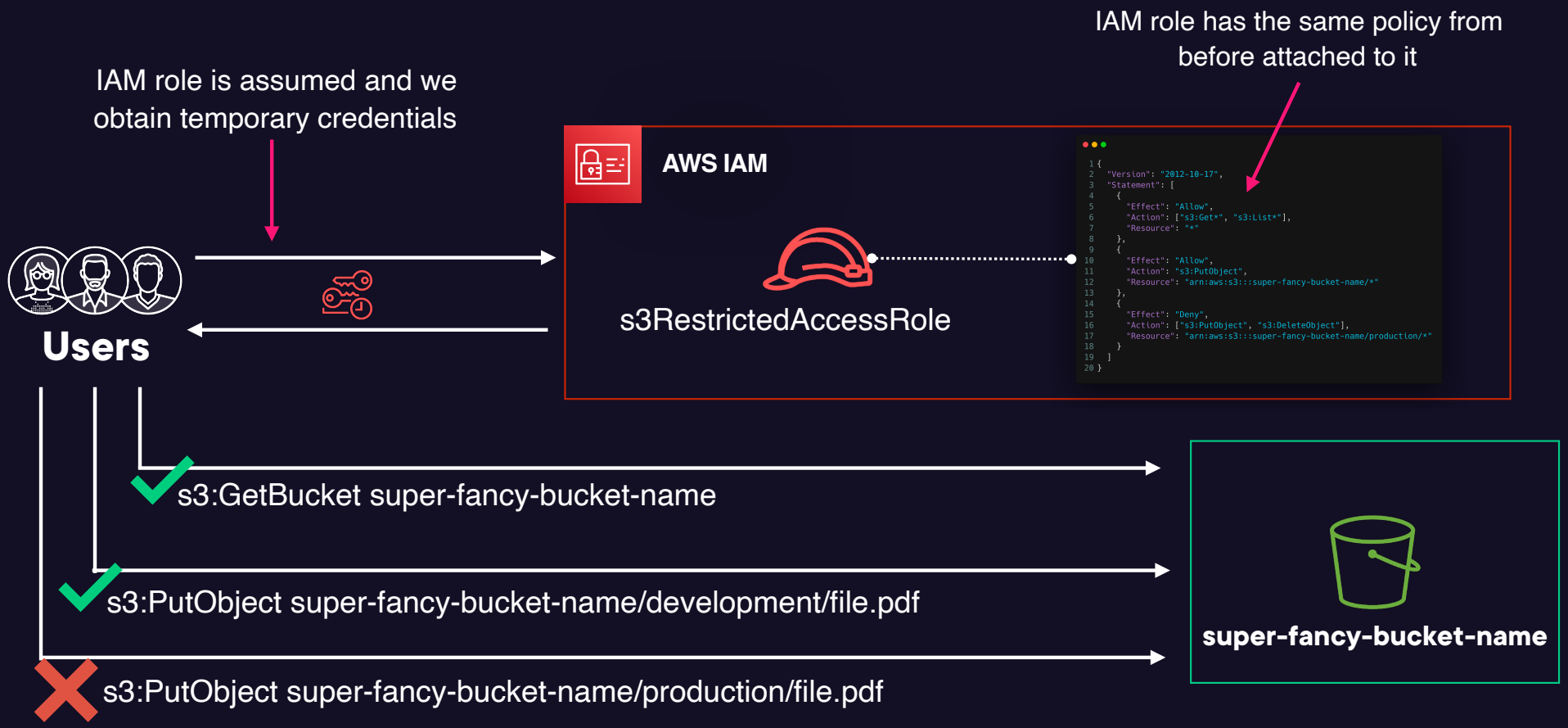
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:Get*", "s3:List*"],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/*"
    },
    {
      "Effect": "Deny",
      "Action": ["s3:PutObject", "s3:DeleteObject"],
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/production/*"
    }
  ]
}
```

Policy Conflicts Example

We have an explicit deny that is denying any S3:PutObject calls, as well as any s3:DeleteObject calls within the production prefix the same super-fancy-bucket-name bucket

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:Get*", "s3:List*"],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/*"
    },
    {
      "Effect": "Deny",
      "Action": ["s3:PutObject", "s3:DeleteObject"],
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/production/*"
    }
  ]
}
```

Permissions Evaluation Order Example



Permissions Evaluation Order Example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:Get*", "s3:List*"],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/*"
    },
    {
      "Effect": "Deny",
      "Action": ["s3:PutObject", "s3:DeleteObject"],
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/production/*"
    }
  ]
}
```

✔ s3:GetBucket super-fancy-bucket-name

✔ s3:PutObject super-fancy-bucket-name/development/file.pdf

✘ s3:PutObject super-fancy-bucket-name/production/file.pdf

Permissions Evaluation Order Example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:Get*", "s3:List*"],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/*"
    },
    {
      "Effect": "Deny",
      "Action": ["s3:PutObject", "s3:DeleteObject"],
      "Resource": "arn:aws:s3:::super-fancy-bucket-name/production/*"
    }
  ]
}
```

✓ s3:GetBucket super-fancy-bucket-name

✓ s3:PutObject super-fancy-bucket-name/**development**/file.pdf

✗ s3:PutObject super-fancy-bucket-name/**production**/file.pdf

Permissions Boundaries Overview

This is a feature where you use a managed policy to set the **maximum permissions** that an identity-based policy can grant to an IAM entity

An IAM entity can only perform the actions that are allowed by both its identity-based policies and its permissions boundaries (*where the permissions overlap*)

Example use case: Delegating IAM responsibility to an IAM administrator, and you need to limit the maximum permissions they can actually grant, as well as preventing the removal of the boundaries in place

Custom Conditions and Statements in IAM Policies

Condition Statements You Need to Know

ExternalID

aws:MultiFactorAuthPresent

aws:SourceIp

aws:PrincipalOrgID

ExternalID

Summary

A unique identifier that can be required when you assume a role in another account

Third-party service needs to access resources in your AWS account, this adds an extra layer of security to prevent the "confused deputy" problem

ExternalID Example

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "UniqueExternalId123"
        }
      }
    }
  ]
}
```

aws:MultiFactorAuthPresent

Summary

Checks if the user has authenticated with multi-factor authentication

Enforcing additional security for sensitive operations, like requiring MFA before termination of instances

aws:MultiFactorAuthPresent Example

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3ActionsWithMFA",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "aws:MultiFactorAuthPresent": "true"
                }
            }
        }
    ]
}
```

aws:SourceIp

Summary

Restricts access based on the IP address of the requester

Limiting access to AWS resources to specific network locations, like only your office location

aws:SourceIp Example

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "NotIpAddress": {
                "aws:SourceIp": [
                    "10.1.0.0/24",
                    "103.0.99.0/24"
                ]
            }
        }
    }
}
```

aws:PrincipalOrgID

Summary

Validates that the request comes from an account that's a member of a specific AWS Organization

You can use this to share resources across accounts within your organization while preventing access from external accounts

aws:PrincipalOrgID Example

```
{
 "Version": "2012-10-17",
 "Statement": {
   "Sid": "AllowPutObject",
   "Effect": "Allow",
   "Principal": "*",
   "Action": "s3:PutObject",
   "Resource": "arn:aws:s3:::pluralsight-bucket/*",
   "Condition": {"StringEquals":
     {"aws:PrincipalOrgID":"o-abcd1234512df"}
   }
 }
}
```

Module Summary and Exam Tips

Permission Evaluation Order Exam Tips

Permission Evaluation Order Exam Tips

1. Explicit Deny

An explicit deny anywhere in the evaluation order always wins

New IAM entities always start with zero permissions.

By default, there is an implicit deny present during all action evaluation processes!

Permissions Boundaries Exam Tips

These are meant to restrict the maximum permissions allowed

They do not grant permissions to IAM entities

IAM delegation is a popular scenario on the exam

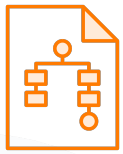
Understand when and how to use the correct condition keys within IAM policies!

Popular Condition Keys

ExternalID

aws:MultiFactorAuthPresent

aws:SourceIp



aws:PrincipalOrgID