RSAConference™2023

San Francisco | April 24 – 27 | Moscone Center

Stronger Together

SESSION ID: CSCO-M05

Destroying Long-Lived Cloud Credentials with Workload Identity Federation

#RSAC

Eric Johnson

Principal Security Engineer, Puma Security
Senior Instructor, SANS Institute
https://www.linkedin.com/in/eric-m-johnson/
@emjohn20

SANS | GIAC CERTIFICATIONS

PUMA SECURITY

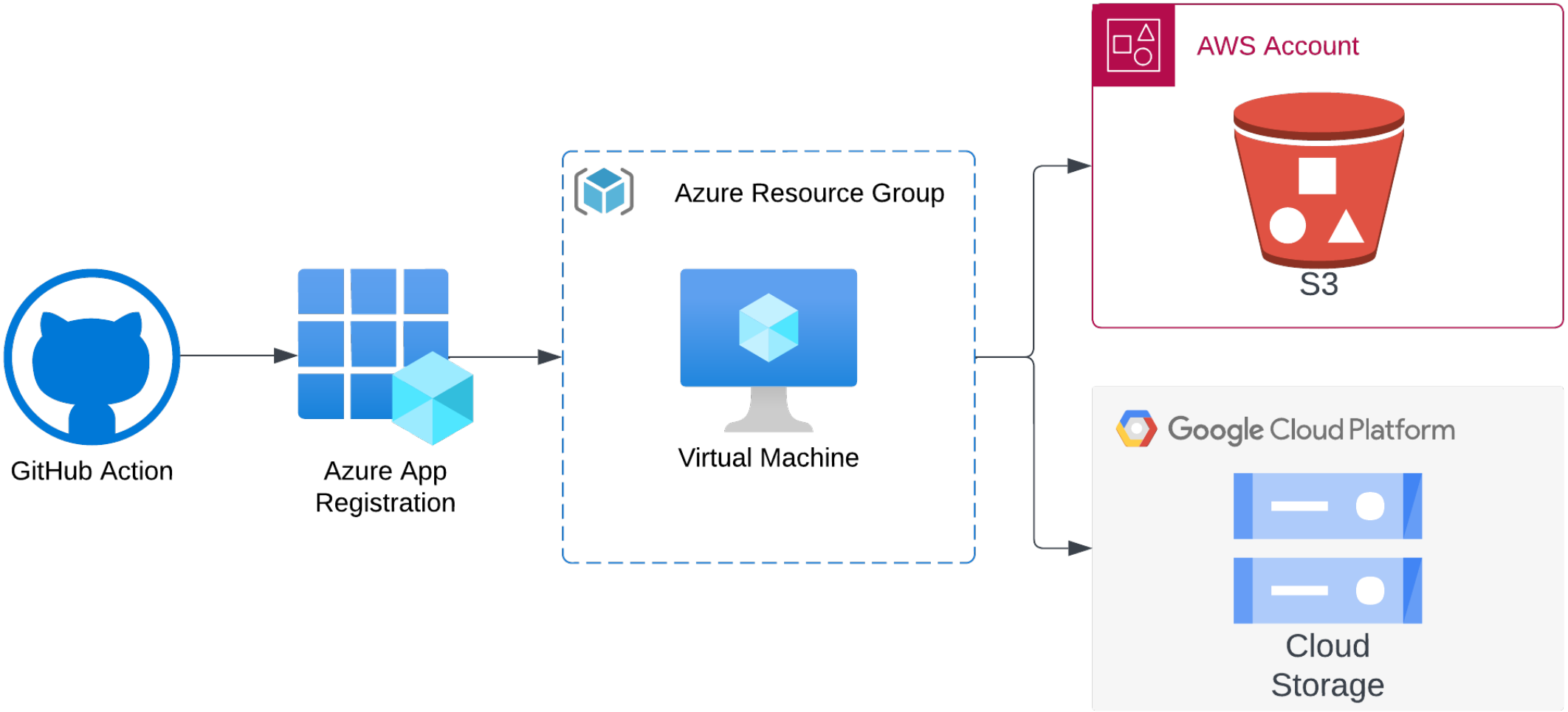
Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference™ or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2023 RSA Conference LLC or its affiliates. The RSA Conference logo and other trademarks are proprietary. All rights reserved.

The Problem: Authenticating Cross Cloud Systems



Session Goals

- Review each cloud provider's long-lived credential type
- Discover insecurely stored cloud credentials
- Learn how to establish trust between an OpenID Connect (OIDC) Identity Provider and each cloud provider
- Configure each cloud provider's workload identity federation capability
- Exchange OpenID Connect (OIDC) access tokens for temporary cloud provider credentials

Open Source: Nymeria's Workload Identity

Nymeria is an open-source repository containing the research for this session:

- Terraform configuration for deploying resources
- Long-lived credential workflow
- Workload Identity Federation workflow
- Step-by-step documentation
- https://github.com/pumasecurity/nymeria

NYMERIA

PUMA SECURITY

RSAConference™2023

Stronger Together

Long-Lived Cloud Credentials

Destroying Long-Lived Cloud Credentials with Workload Identity Federation

Cloud Provider Credential Types

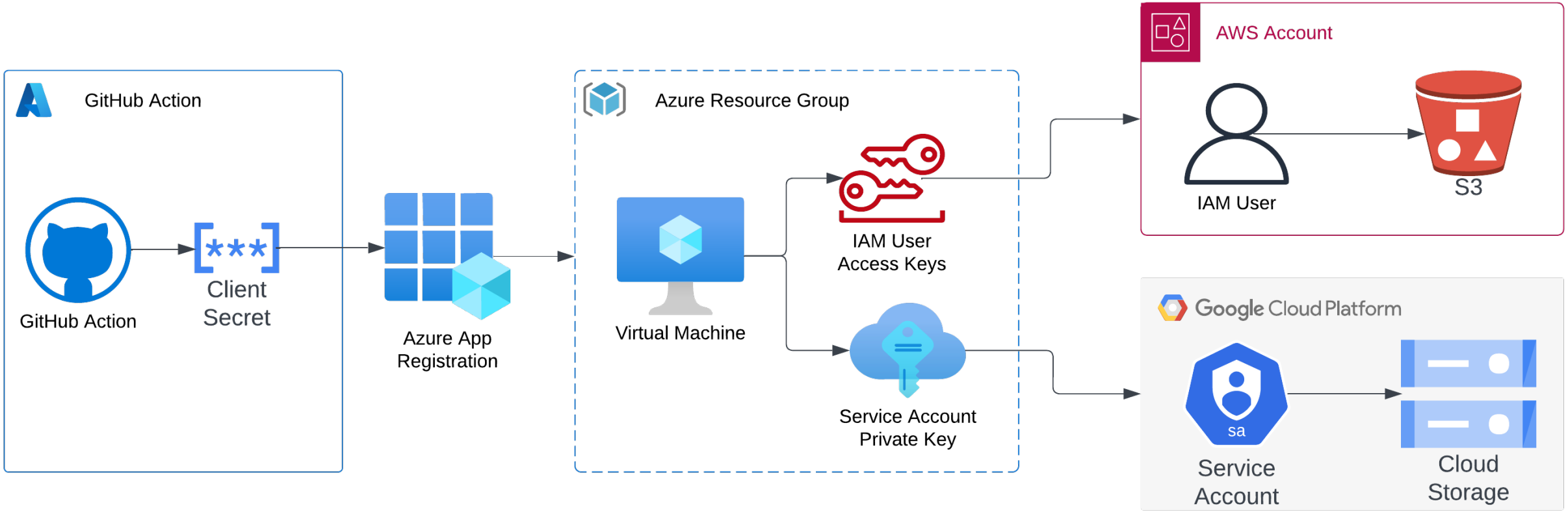
Each cloud provider's Identity & Access Management (IAM) service provides an option for creating long-lived credentials:

IAM User Access Keys

Service Principal Client Secrets

Google Service Account Keys

Nymeria: Long Lived Credential Workflow



MITRE ATT&CK T1522: Unsecured Credentials

MITRE ATT&CK T1522: Discovering insecurely stored IAM and service account credentials:

Persistent Credential Locations		
Bash History	Configuration Files	Source Code
Version Control	Instance Metadata Service	Environment Variables

ATT&CK®

Unsecured Credentials: Configuration Files

Cloud-focused malware (e.g., TeamTNT) will focus on common locations with cleartext credentials:

- ~/.aws/credentials
- ~/.azure/accessTokens.json
- ~/.config/gcloud/*credential*
- ~/.ssh/*

Unsecured Credentials: Environment Variables

- Continuous Integration pipelines, containers, and functions often have secrets stored in environment variables
- Local File Inclusion (LFI) and Command Injection vulnerabilities can allow attackers to exfiltrate environment variables

```
env | grep 'AWS'

AWS_LOG_GROUP_NAME=/org/github/nymeria-ci
AWS_DEFAULT_REGION=us-east-1
AWS_ACCESS_KEY_ID=ASIA54BL6EJRTTJ4SS7A
AWS_SECRET_ACCESS_KEY=aEWSwA8k/U7IY38JetxQDZ9voUG
AWS_SESSION_TOKEN=IQoJb3JpZu2DaXVzLWVhc3Q...4pg9g==
```

Unsecured Credentials: Version Control Systems

Node Package Manager

- Scanned by Aidan Steele in October 2021
- Identifies 117 valid API keys, including 30 AWS root access keys
- https://sec549.com/id259

Python Package Index (PyPI)

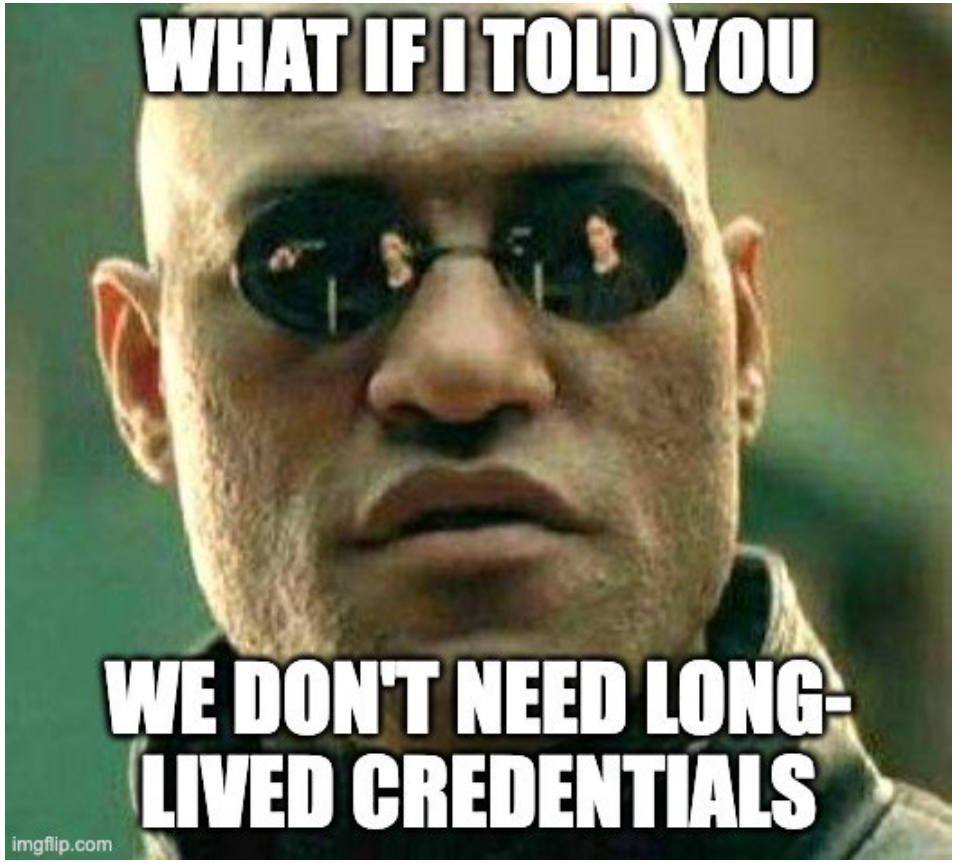
- Scanned by Tom Forbes in January 2023
- Identifies 57 valid AWS API keys, including 11 AWS root access keys
- https://sec549.com/id260

T1522 Unsecured Credentials Mitigations

Common mitigations…

- Version control secrets scanning
- Secrets management services
- Compromised Credentials Detection

But….



RSAConference™2023

Stronger Together

Workload Identity Federation

Destroying Long-Lived Cloud Credentials with Workload Identity Federation

OpenID Connect Identity Provider

Destroying long-lived credentials requires an OpenID Connect Identity Provider (IdP) issuing an Identity Token:

Identity Provider JSON Web Token (JWT)

Identity Provider issues a JSON Web Token (JWT) to the workload or user that requires access to cloud resources:

- Header: Token type (JWT) and signing algorithm (HMAC, RSA)
- Payload: JSON object that includes the issuer, audience, claims, and expiration
- Signature: Signed header and payload for verifying the token and issuer


```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiI1NDIxMDU0ODQiLCJ
jbGFpbSI6IkdldEFjY291bnQifQ.Kho7o2Rz9p42HKi84KfWBxA-
xJAwwKIAdy4msKSy0ZY
```

GitHub Actions JWT Verification Fields

Header: *kid* and *alg* are used to verify the signature:

```
{ "typ": "JWT",
  "alg": "RS256",
  "kid": "78167F727DEC5D801DD1C8784C704A1C880EC0E1" }
```

Payload: *aud, iss*, and *sub* are used to establish trust with the external identity provider:

```
{
  "iss": "https://token.actions.githubusercontent.com",
  "sub": "repo:pumasecurity/nymeria:ref:refs/heads/main",
  "aud": "api://AzureADTokenExchange",
  …
}
```

GitHub Actions OpenID Connect IdP Configuration

- *https://token.actions.githubusercontent.com/.well-known/openid-configuration:*

```
{ "issuer": "https://token.actions.githubusercontent.com",
  "jwks_uri": "https://token.actions.githubusercontent.com/.well-
               known/jwks",
  …
}
```

- *https://token.actions.githubusercontent.com/.well-known/jwks*

```
{ "keys": [ {
      "kid": "78167F727DEC5D801DD1C8784C704A1C880EC0E1",
      "alg": "RS256",
      "x5c": ["MIIDrDCCApSgAwIBAgIQMPdKi0TFTMqmg1HHkqhkiG9w0…"],
      …
} ] }
```

Workload Identity Federation

Each cloud provider's Identity & Access Management (IAM) service supports federated authentication by configuring a trusted OpenID Connect identity provider:

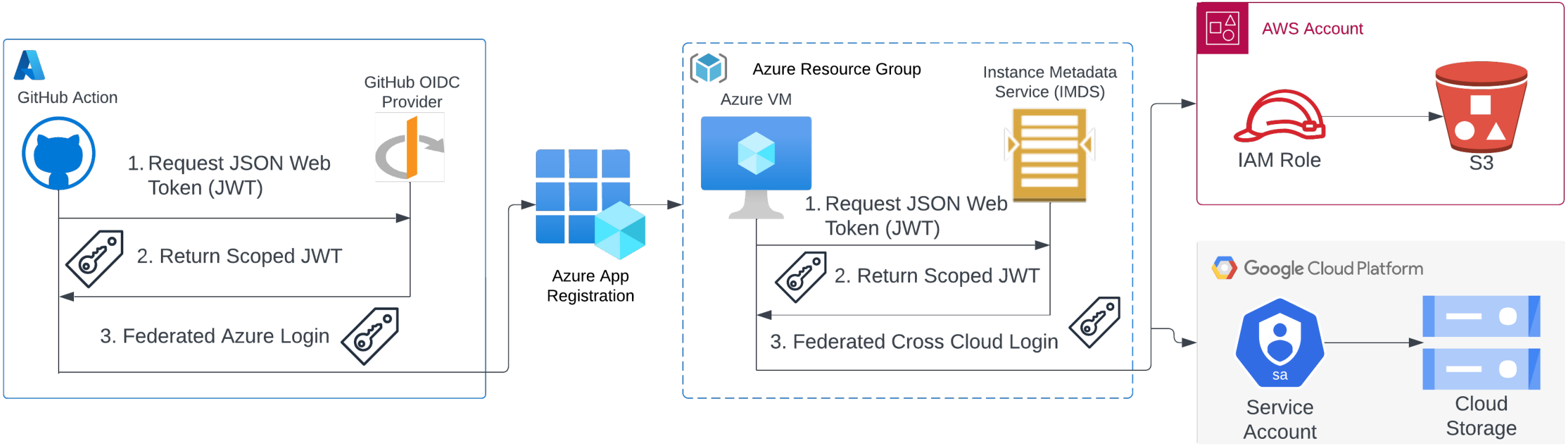
IAM Identity Provider

Service Principal Federated Credentials

Google Workload Identity Federation

OpenID®

Nymeria: Federated Identity Workflow



RSAConference™2023

Stronger Together

Federating Access From GitHub Actions to Azure AD

Destroying Long-Lived Cloud Credentials with Workload Identity Federation

Azure Service Principal Federated Credentials

- **Issuer**: set the value to the JWT's *iss* claim
- **Subject**: set the value to the JWT's *sub* claim
- **Audience**: set the value to the JWT's *aud* claim
- **RBAC**: Role assignments determine permissions for the subject

Issuer ⓘ
https://token.actions.githubusercontent.com
Edit (optional)

Subject identifier * ⓘ
repo:pumasecurity/nymeria:ref:refs/heads/main
Generate this value using your GitHub account de

Credential details

Provide a name and description for this credential and review other details.

Name ⓘ
github-federated-identity

Description ⓘ
Deployments for GH Action

Audience * ⓘ
api://AzureADTokenExchange
Edit (optional)

Azure Federated Login Access

- Request an Azure AD scoped JSON Web Token from the GH Action Identity Provider

```
GH_JWT=$(curl -H "Authorization: bearer $ACTIONS_ID_TOKEN_REQUEST_TOKEN"
"$ACTIONS_ID_TOKEN_REQUEST_URL&audience=api://AzureADTokenExchange" | jq
-r '.value')
```

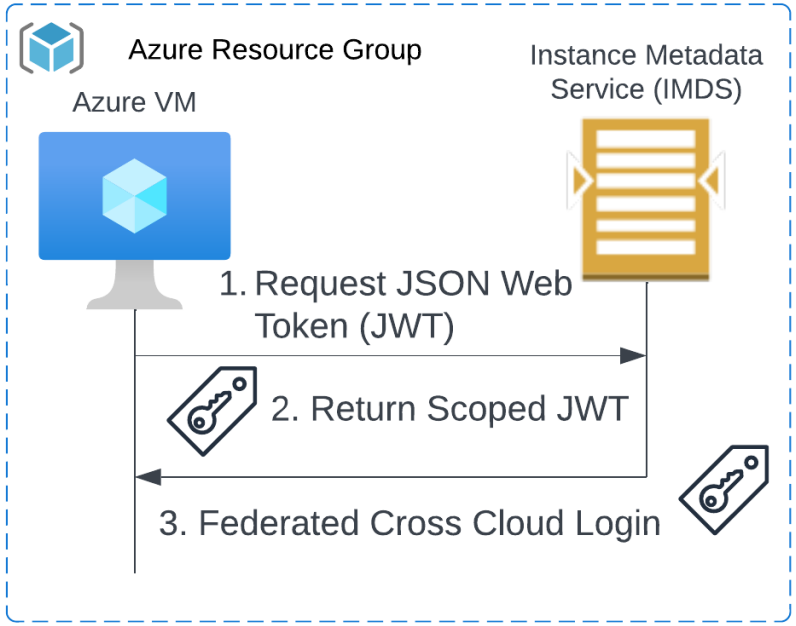
- Sign into the Azure AD tenant's service principal using the federated JSON Web Token

```
az login --service-principal --tenant $AZURE_TENANT_ID --username
$AZURE_CLIENT_ID --federated-token $GH_JWT

terraform init && terraform plan && terraform apply -auto-approve
```

Azure Virtual Machine Managed Identity

- Enable Azure Virtual Machine Managed Identity to assigned a unique identity to the instance
- Request a JSON Web Token from the virtual machine's Instance Metadata Service (IMDS)




```
AZURE_JWT=$(curl -s "http://169.254.169.254/metadata/identity/oauth2/
token?api-version=2018-02-01&resource=api://nymeria-workload-identity"
-H "Metadata: true" | jq -r '.access_token')
```

Managed Identity JSON Web Token

The virtual machine's JWT contains the following claims:

- **Issuer**: OpenID Connect endpoint for the Azure AD tenant
- **Subject**: managed identity principal identifier
- **Audience**: Value specified in the *resource* IMDS request

```
{
  "iss": "https://sts.windows.net/ac7f86b6-99c3-4d3b-ae4e-
          e53c8a0bfa2c/",
  "sub": "9dfedb65-1de5-4f03-a9da-0a08f49a4c9e",
  "aud": "api://nymeria-workload-identity",
  …
}
```

RSAConference™2023

Stronger Together

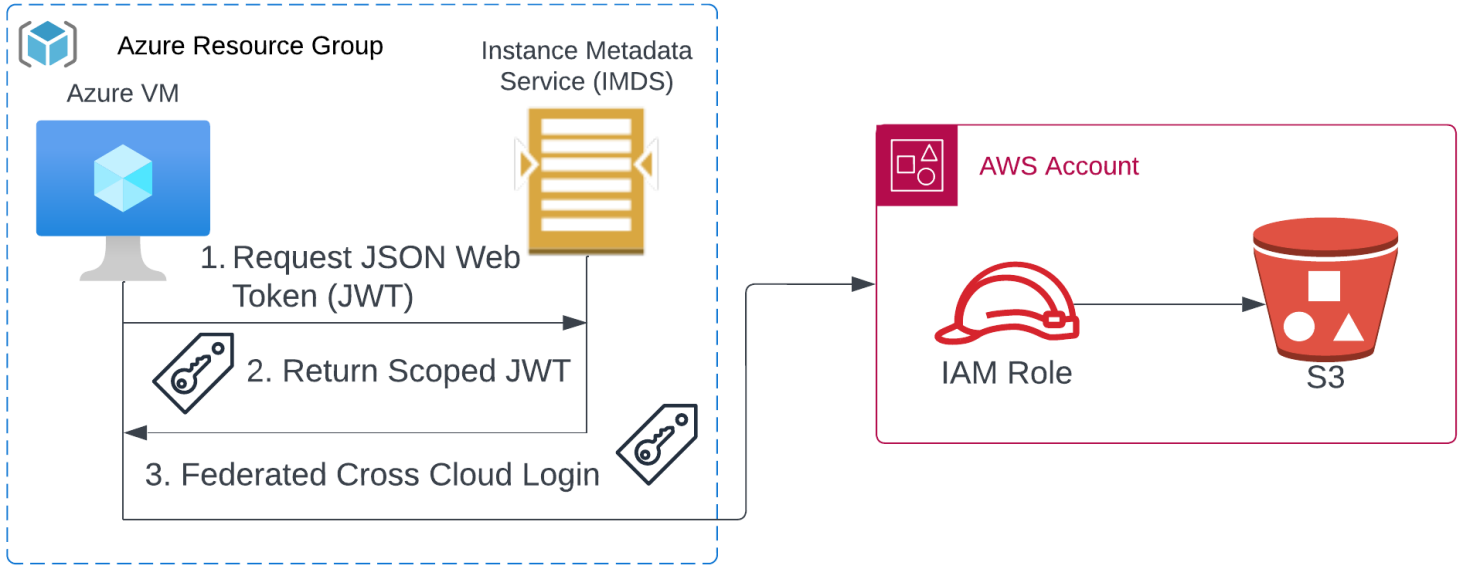
Federating Access From Azure Managed Identity to an AWS Role

Destroying Long-Lived Cloud Credentials with Workload Identity Federation

#RSAC

AWS Federated Identity Configuration

- Configure an AWS Identity Provider that trusts the Azure AD tenant's OpenID Connect provider
- Create an IAM Role trust policy that restricts access to the virtual machine's managed identity principal id



AWS Identity Provider

- **Provider Type**: Choose OpenID Connect
- **Provider URL**: set the value to the JWT's *iss* claim
- **Audience**: set the value to the JWT's *aud* claim
- **Thumbprint**: value is calculated automatically from the JWKS endpoint's certificate

Configure provider

Provider type Info

SAML
Establish trust between your AWS account and a SAML 2.0 compatible Identity Provider such as Shibboleth or Active Directory Federation Services.

OpenID Connect
Establish trust between your AWS account and Identity Provider services, such as Google or Salesforce.

Provider URL
Specify the secure OpenID Connect URL for authentication requests.

https://sts.windows.net/ac7f86b6-99c3-4d

Get thumbprint

Maximum 255 characters. URL must begin with "https"

Audience Info
Specify the client ID issued by the Identity provider for your app.

api://nymeria-workload-identity

Maximum 255 characters. Use alphanumeric or ':_.-/' characters.

AWS IAM Role Trust Policy

IAM Role Trust Policy restricting access from the *iss*, *aud*, and *sub*:

```
1   { "Version": "2012-10-17",
2     "Statement": [
3       { "Effect": "Allow",
4         "Principal": {
5           "Federated": "arn:aws:iam::111111111111:oidc-provider
6              /sts.windows.net/ac7f86b6-99c3-4d3b-ae4e-e53c8a0bfa2c/" },
7         "Action": "sts:AssumeRoleWithWebIdentity",
8         "Condition": {
9           "StringEquals": {
10            "sts.windows.net/ac7f86b6-99c3-4d3b-ae4e-e53c8a0bfa2c/:aud":
11              "api://nymeria-workload-identity",
12            "sts.windows.net/ac7f86b6-99c3-4d3b-ae4e-e53c8a0bfa2c/:sub":
13              "9dfedb65-1de5-4f03-a9da-0a08f49a4c9e"
14        } }
15  } ] }
```

ALLOWS ALL PRINCIPALS FROM THE TRUSTED IDP

RESTRICTS ACCESS TO A SINGLE SUBJECT

AWS Role Assumption Role With Web Identity

- Request temporary AWS access keys with the Managed Identity JSON Web Token:

```
aws sts assume-role-with-web-identity --role-arn "arn:aws:iam::
111111111111:role/azure-role" --role-session-name "federated-identity"
--web-identity-token "$AZURE_JWT"
```

- Response contains temporary access keys for the role:

```
{
  "Credentials": {
    "AccessKeyId": "ASIASZY2ZSU66ET7N23K",
    "SecretAccessKey": "rXG50aGEXOT5Xe/b5LPbsSX9FcBIol123G5gELVv",
    "SessionToken": "IQoJb3JpZ2luX2VjEMP…NFI=",
    "Expiration": "2023-02-28T04:02:16+00:00"
  },
  …
}
```

RSAConference™2023

Stronger Together

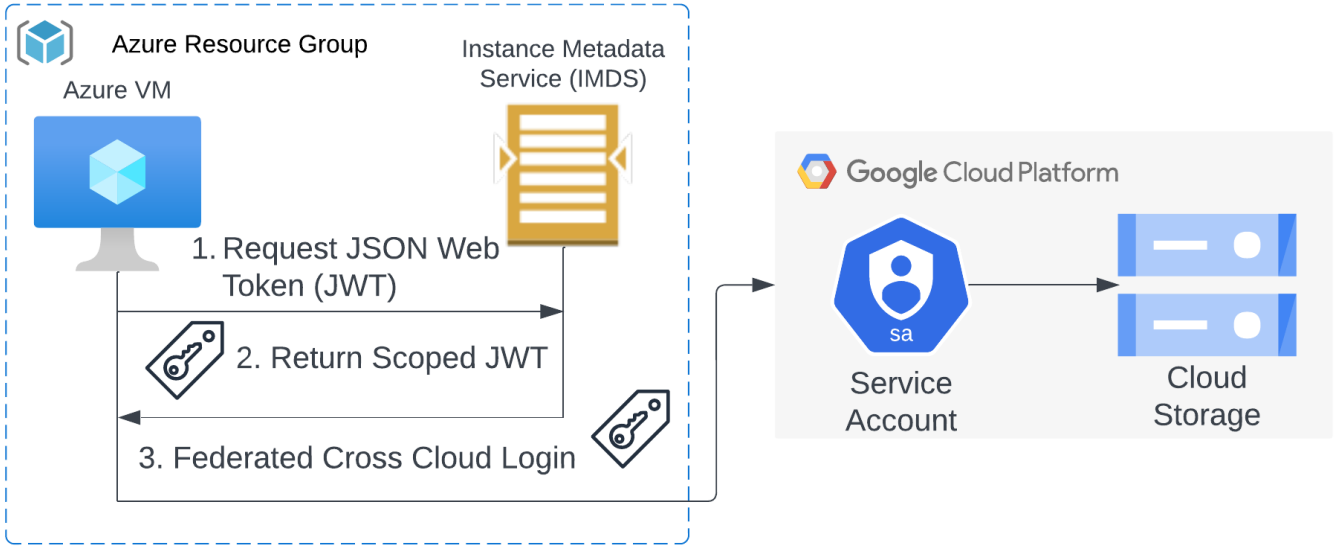
Federating Access From Azure Managed Identity to a Google Cloud Service Account

Destroying Long-Lived Cloud Credentials with Workload Identity Federation

#RSAC

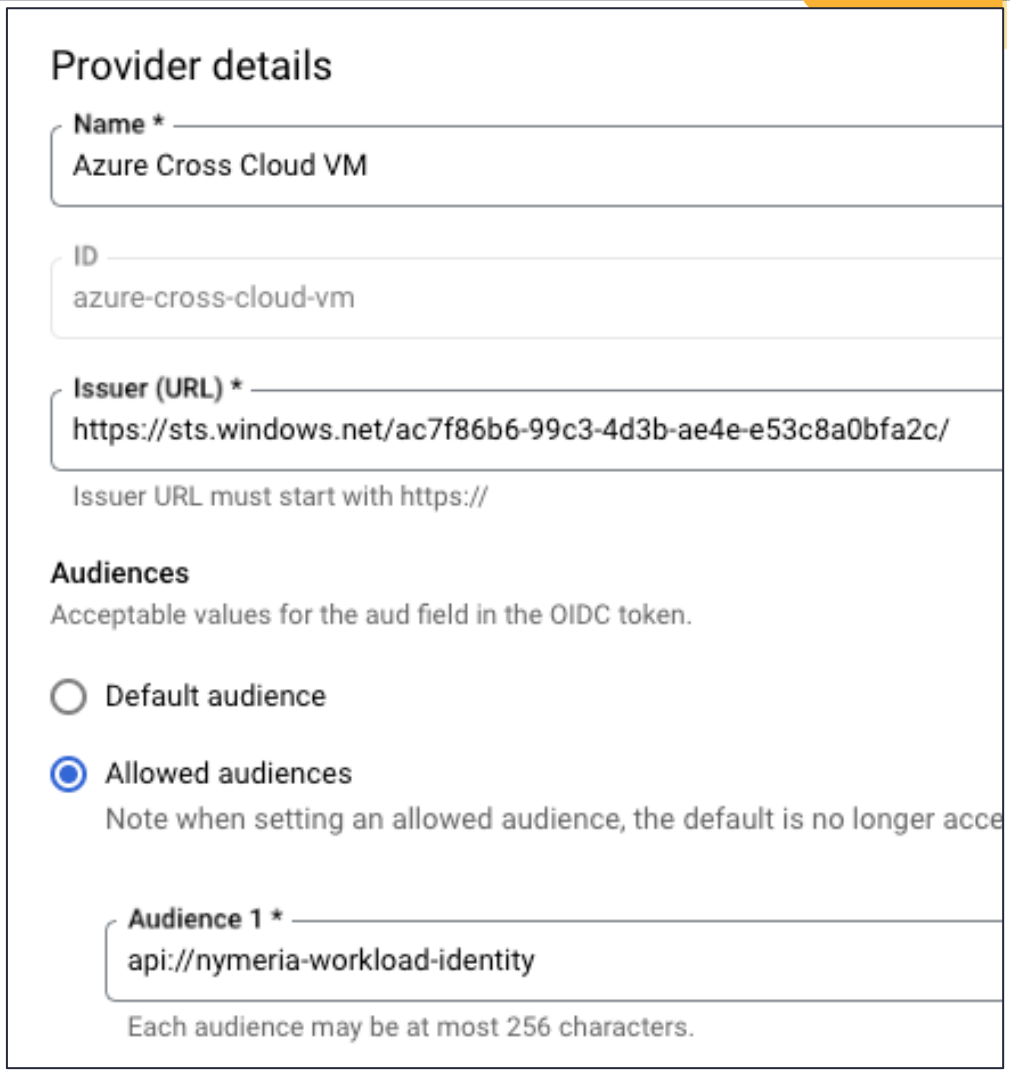
Google Cloud Workload Identity Configuration

- Configure a Workload Identity Pool Provider that trusts the Azure AD tenant's OpenID Connect provider
- Grant the Azure virtual machine's managed identity principal id permissions to impersonate an Identity Pool's Service Account



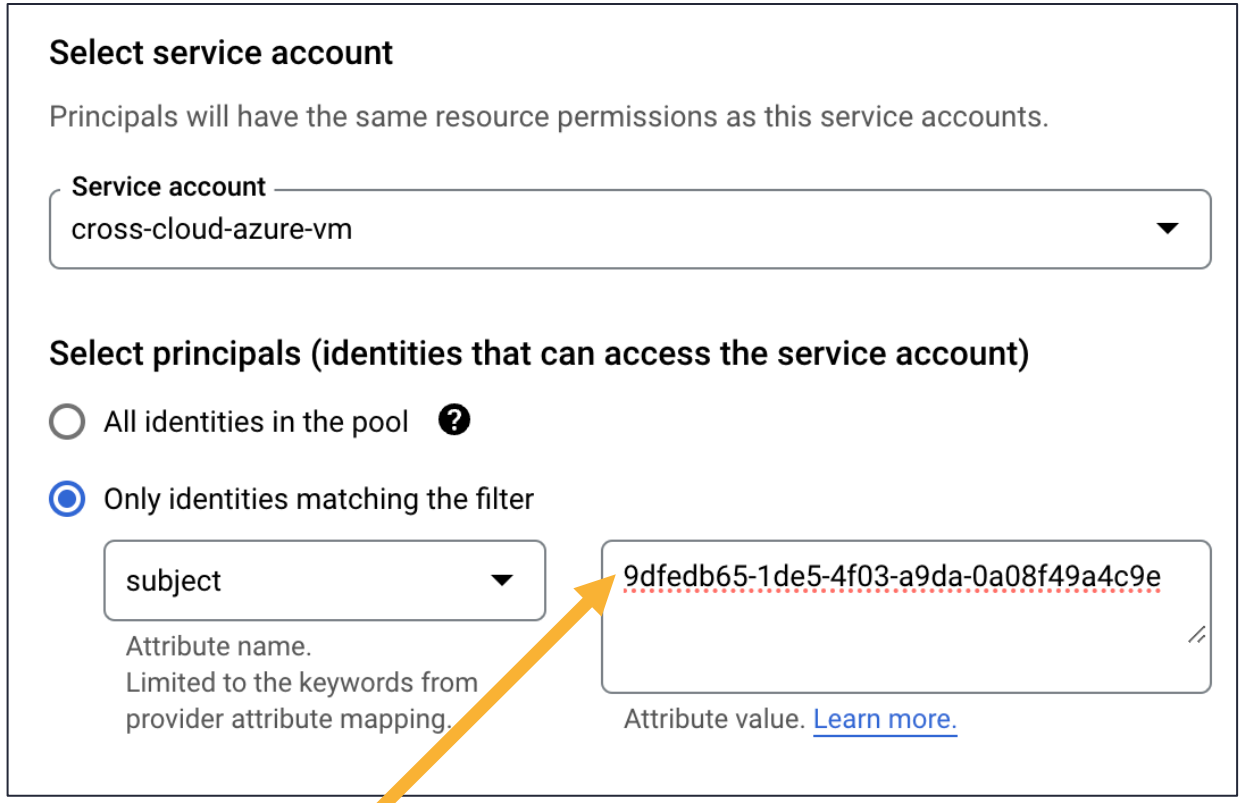
Workload Identity Pool Provider

- **Provider**: Choose Open ID Connect
- **Issuer (URL)**: set the value to the JWT's *iss* claim
- **Audiences**: set the allowed audiences to the JWT's *aud* claim
- **Provider Attributes**: set the *google.subject* key to the *attribute.sub* value



Workload Identity Pool Service Account Access

- **Service Account**: Choose the service account with the desired roles and permissions
- **Select Principals:** Restrict access to a given set of subjects (wildcards are supported)



RESTRICTS ACCESS TO A SINGLE SUBJECT

Google Cloud Workload Identity Login

- Download the service account access client configuration file:

```
{ "type": "external_account",
  …
  "credential_source": {
    "url": "http://169.254.169.254/metadata/identity/oauth2/token?api-
            version=2018-02-01&resource=api://nymeria-workload-identity",
    "headers": {
      "Metadata": "True" }
} }
```

- Sign into the Google Cloud project using the federated JWT:

```
gcloud auth login --cred-file=/home/ubuntu/gcp-azure-cross-cloud.json

Authenticated with external account credentials for: [cross-cloud-azure-
vm@rsa-313853.iam.gserviceaccount.com]
```

Apply What You Have Learned Today

- Next week you should:
  - Inventory IAM Users, Service Principals, and Service Accounts with long-lived credentials
  - Research the external system's Identity Provider (IdP) options
- Within three months, you should:
  - Create workload identity resources that trust the client's IdP tokens
  - Update workflows, scripts, and code to authenticate with IdP tokens
- Within six months, you should:
  - Deactivate the long-lived cloud credentials

Acknowledgements & References

- GitHub:
  - https://docs.github.com/en/actions/deployment/security-hardening-your-deployments/about-security-hardening-with-openid-connect
- GitLab:
  - https://docs.gitlab.com/ee/ci/cloud_services/index.htm
- Aidan Steele:
  - https://awsteele.com/blog/2021/09/15/aws-federation-comes-to-github-actions.html
- Ryan Canty:
  - https://jryancanty.medium.com/stop-downloading-google-cloud-service-account-keys-1811d44a97d9
- Prashant Kulkarni:
  - https://medium.com/google-cloud/google-cloud-workload-identity-federation-with-okta-90c05b985b17
- Brad Geesaman:
  - https://github.com/ghostsecurity/gs-aws-to-gcp-workload-identity

RSAConference™2023

San Francisco | April 24 – 27 | Moscone Center

SESSION ID: CSCO-M05

Destroying Long-Lived Cloud Credentials with Workload Identity Federation

#RSAC

Eric Johnson

Principal Security Engineer, Puma Security
Senior Instructor, SANS Institute
https://www.linkedin.com/in/eric-m-johnson/
@emjohn20