black hat®
USA 2022

IAM The One Who Knocks

Igal Gofman, Noam Dahan

Igal Gofman

@IgalGofman

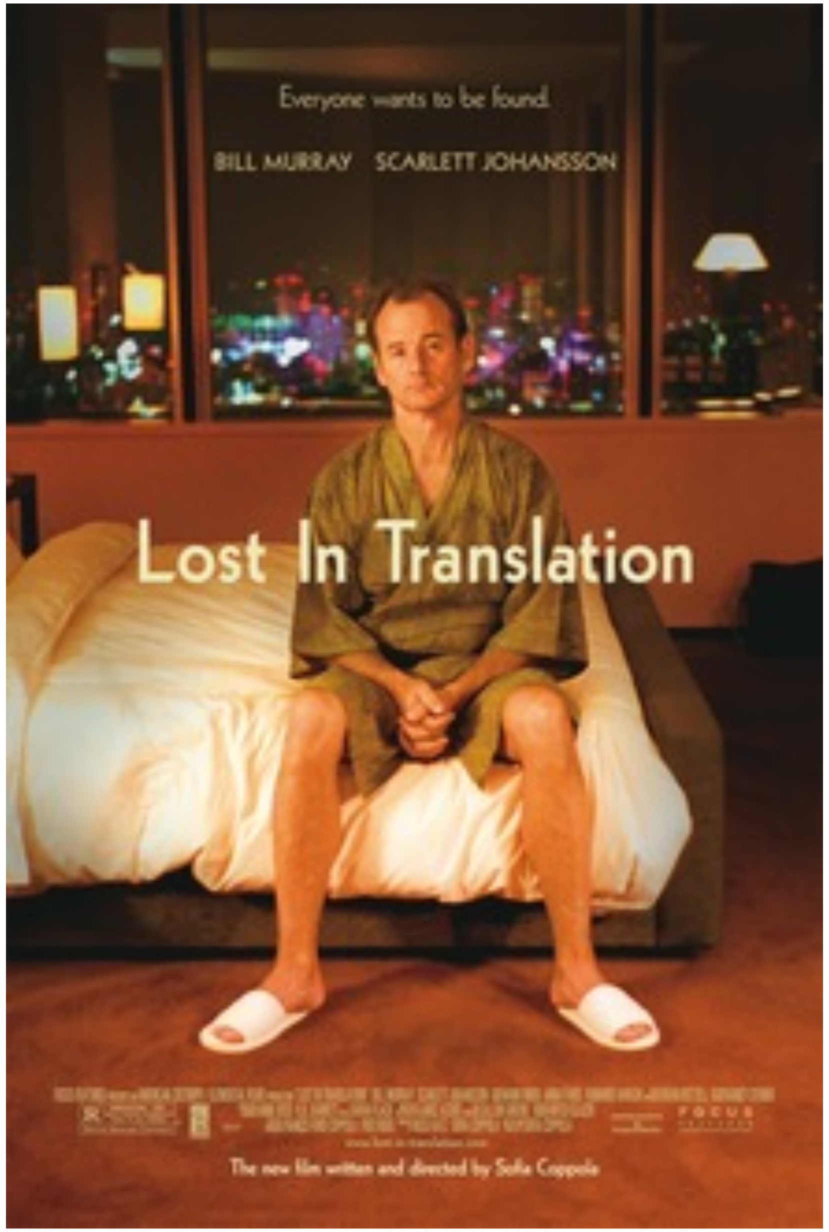
- Head of Research, Ermetic
- Microsoft MSTIC
- Microsoft security research
- Active Directory expert

Noam Dahan

@NoamDahan

- Cloud security researcher
- Love/hate relationship with embedded devices
- Offensive background

Why are we here?

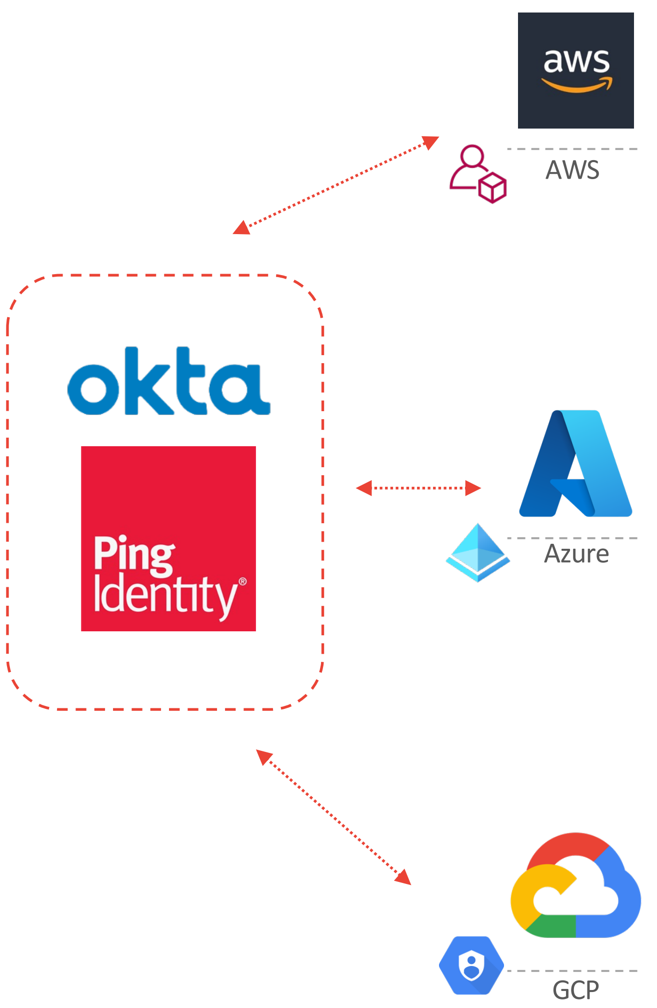

Everyone wants to be found.
BILL MURRAY SCARLETT JOHANSSON
Lost In Translation
The new film written and directed by Sofia Coppola

Agenda

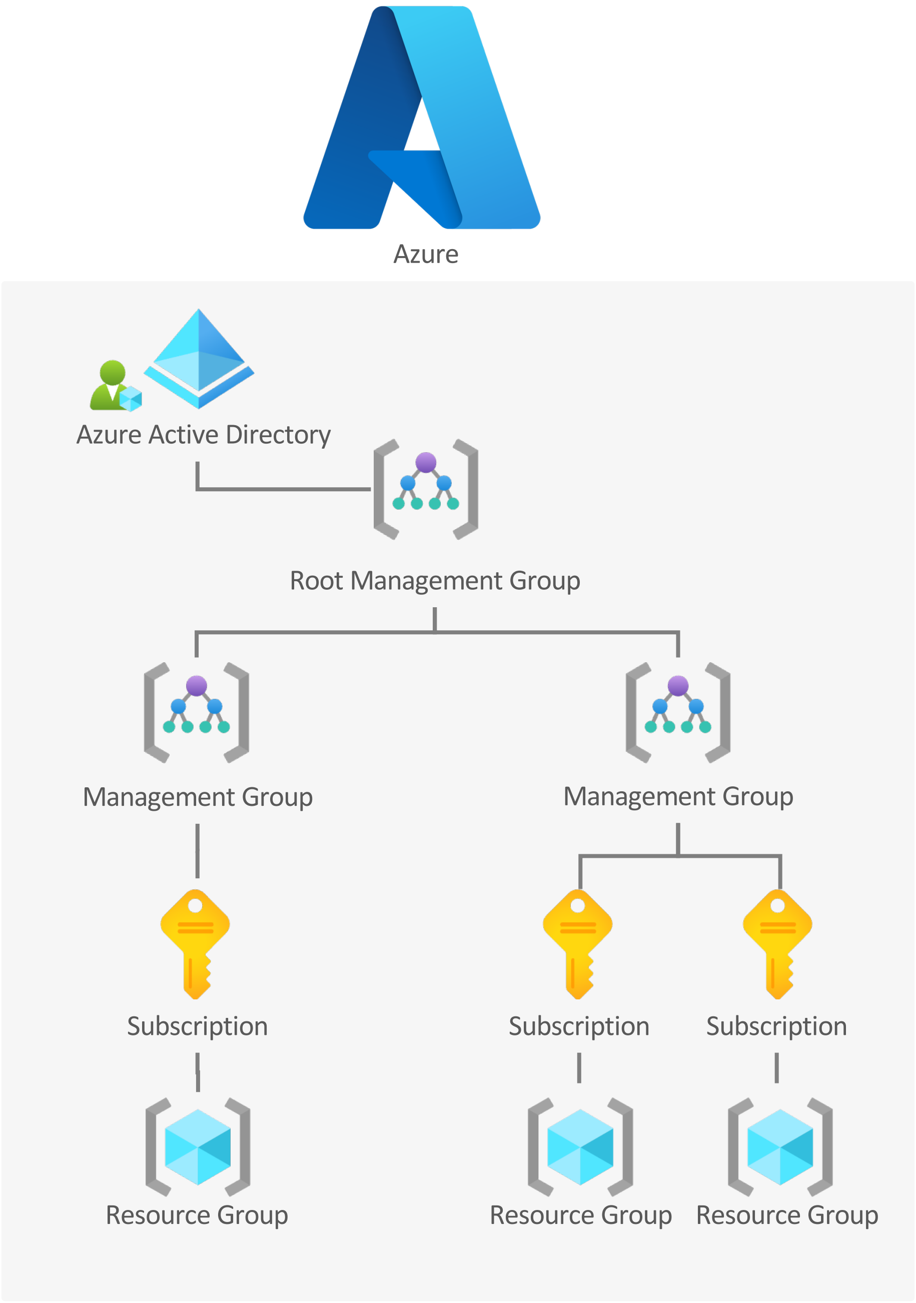
- IAM Crash Course
- Cloud IAM weak spots (permissions landscape)
- Things are not always what they seem
- Defense, Monitoring techniques, demo

black hat
USA 2022

IAM Crash Course



Identity		Permissions	Scope
Users Groups IAM Role AWS SSO directory Federated Identities	Attach	AWS Policy Customer Managed Inline policy	
Users Groups Managed identity Service Principal Federated Identities	Assign	Role (Built-In | Custom)	Scope
Users Groups Service Account Pre-Defined Groups | Special groups Federated Identities	Bind	Role (Basic | Predefined Custom)	Scope



Azure

Azure Active Directory

Root Management Group

Management Group

Management Group

Subscription

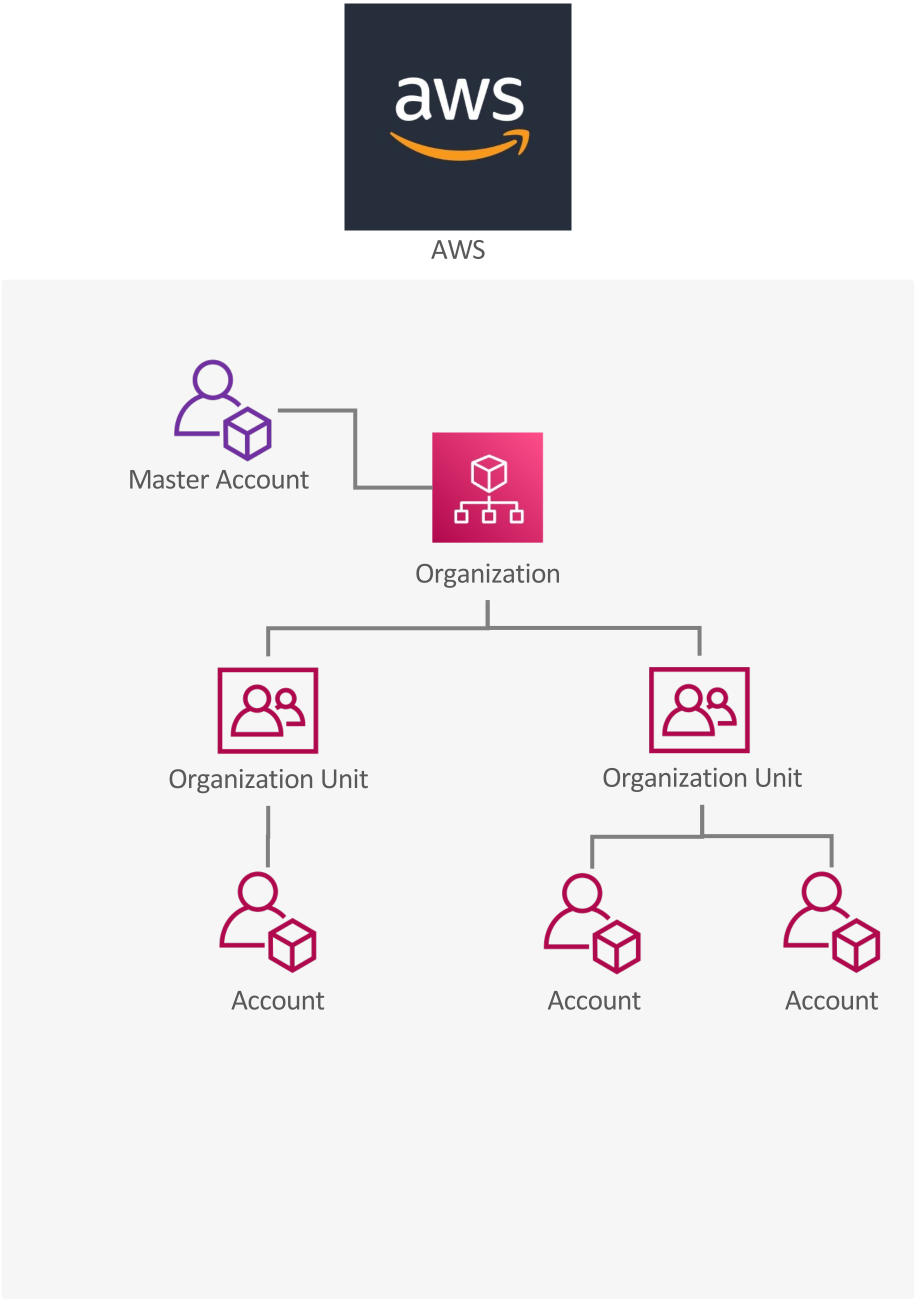
Subscription

Subscription

Resource Group

Resource Group

Resource Group



AWS

Master Account

Organization

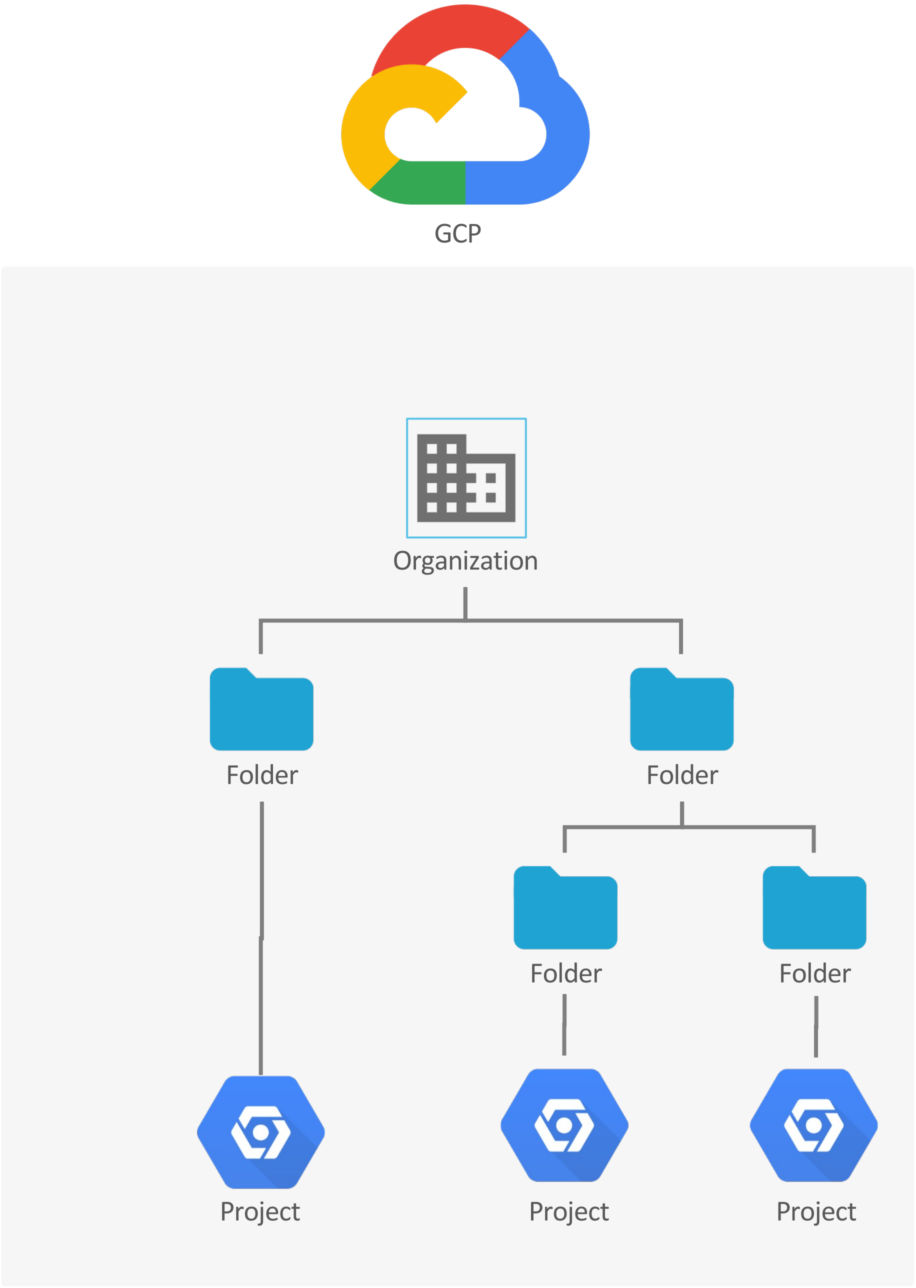
Organization Unit

Organization Unit

Account

Account

Account



GCP

Organization

Folder

Folder

Folder

Folder

Project

Project

Project

black hat USA 2022

Cloud IAM weak spots

- Assignment operations
- Code Execution 😇
- Grants and Delegation
- New credentials | secrets
- Cryptographic key management

Dangerous permissions

Assignment | Code Execution | Grants and Delegation | New credentials

Assignment

- Azure - Microsoft.Authorization/roleAssignments/write
- Azure - Microsoft.Authorization/roleDefinitions/write
- GCP - iam.roles.update
- GCP - orgpolicy.policy.set
- GCP - resourcemanager.projects.setIamPolicy
- AWS - lambda:AddPermission
- AWS - iam:AttachUserPolicy
- AWS - iam:AttachGroupPolicy
- AWS - iam:AttachRolePolicy

Grants and Delegation

- GCP - iam.serviceAccounts.implicitDelegation
- GCP - deploymentmanager.deployments.create
- GCP - iam.serviceAccounts.actAs
- AWS - iam:PassRole
- Azure - Microsoft.ManagedIdentity/userAssignedIdentities/*/assign/action
- AWS - kms:CreateGrant

Code Execution

- AWS - lambda:CreateFunction
- AWS - lambda:InvokeFunction
- AWS - lambda:UpdateFunctionConfiguration
- AWS - cloudformation:CreateStack
- GCP - cloudscheduler.jobs.create
- GCP - cloudbuild.builds.create
- GCP - cloudfunctions.functions.create
- GCP - cloudfunctions.functions.update
- GCP - run.services.create

New Credentials

- AWS - iam:CreateLoginProfile
- AWS - iam:UpdateLoginProfile
- AWS - iam:CreateAccessKey
- GCP - iam.serviceAccountKeys.create
- GCP - iam.serviceAccounts.signJwt
- GCP - serviceusage.apiKeys.create
- GCP - iam.serviceAccounts.getAccessToken

black hat USA 2022

Things are not always what they seem

Non-human Identities

AWS

Service role

Azure

Managed Identities

GCP

Service account

- How cloud providers handle non-human credentials (Certificates)
- How cloud consumers handle non-human credentials (Short-lived tokens)
- The Instance metadata, local addresses, and environment variables
- Beware of the hybrid Instance metadata

Non-human Identities

Non-human Identities

- How cloud providers handle non-human credentials (Certificates)
- How cloud consumers handle non-human credentials (Short-lived tokens)
- The Instance metadata, local addresses, and environment variables
- Beware of hybrid Instance metadata

AWS

Service Role

Azure

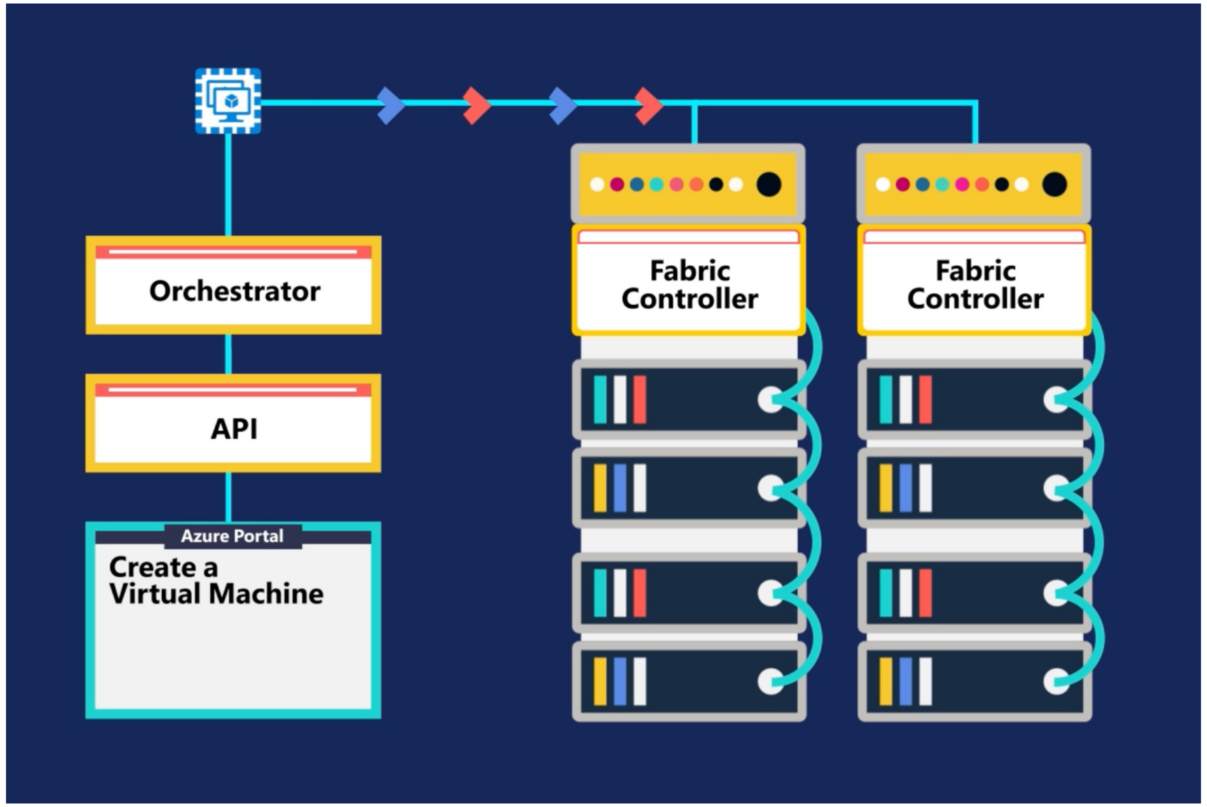
Managed Identities

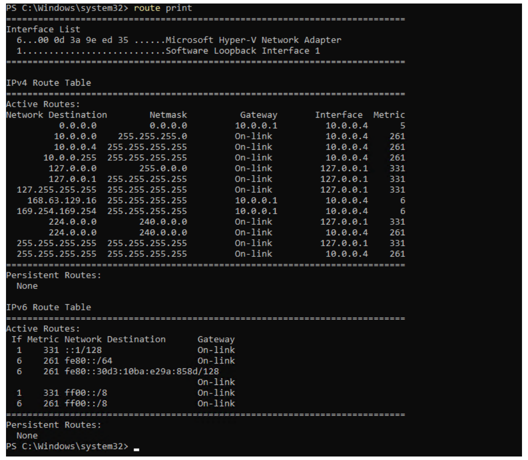
GCP

Service account

Lesson #1: Beware of non-human identities

- The **Fabric Controller** (**FC**) is responsible for maintaining and monitoring all the resources in the data center cluster.

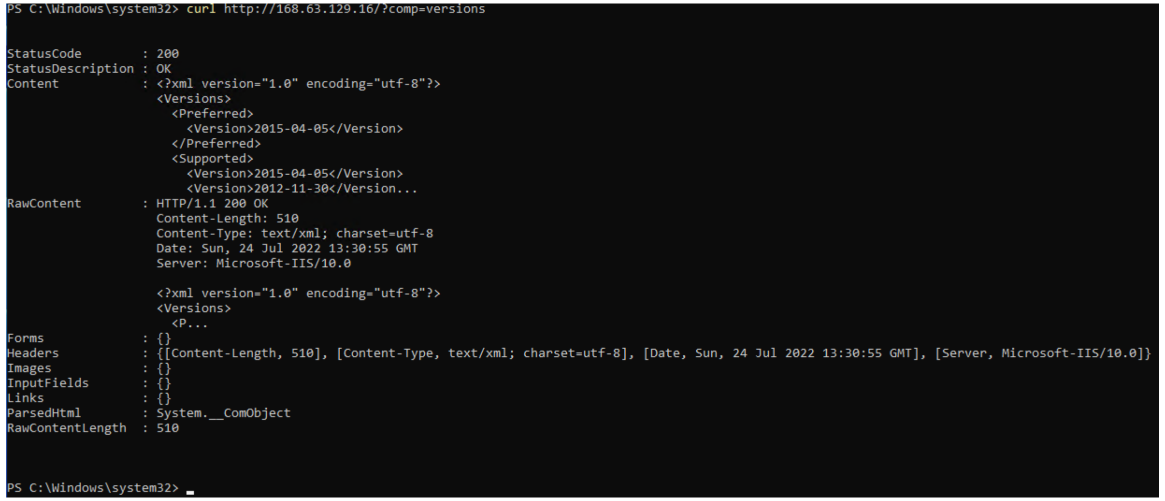




```
PS C:\Windows\system32> route print
===========================================================================
Interface List
  6...00 0d 3a 9e ed 35 ......Microsoft Hyper-V Network Adapter
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0         10.0.0.1        10.0.0.4      5
         10.0.0.0    255.255.255.0         On-link         10.0.0.4    261
         10.0.0.4  255.255.255.255         On-link         10.0.0.4    261
       10.0.0.255  255.255.255.255         On-link         10.0.0.4    261
        127.0.0.0        255.0.0.0         On-link        127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link        127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link        127.0.0.1    331
     168.63.129.16  255.255.255.255        10.0.0.1        10.0.0.4      6
  169.254.169.254  255.255.255.255        10.0.0.1        10.0.0.4      6
        224.0.0.0        240.0.0.0         On-link        127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link         10.0.0.4    261
  255.255.255.255  255.255.255.255         On-link        127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link         10.0.0.4    261
===========================================================================
Persistent Routes:
  None

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
  6    261 fe80::/64                On-link
  6    261 fe80::30d3:10ba:e29a:858d/128
                                    On-link
  1    331 ff00::/8                 On-link
  6    261 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
PS C:\Windows\system32> _
```



```
PS C:\Windows\system32> curl http://168.63.129.16/?comp=versions


StatusCode        : 200
StatusDescription : OK
Content           : <?xml version="1.0" encoding="utf-8"?>
                    <Versions>
                      <Preferred>
                        <Version>2015-04-05</Version>
                      </Preferred>
                      <Supported>
                        <Version>2015-04-05</Version>
                        <Version>2012-11-30</Version...
RawContent        : HTTP/1.1 200 OK
                    Content-Length: 510
                    Content-Type: text/xml; charset=utf-8
                    Date: Sun, 24 Jul 2022 13:30:55 GMT
                    Server: Microsoft-IIS/10.0

                    <?xml version="1.0" encoding="utf-8"?>
                    <Versions>
                      <P...
Forms             : {}
Headers           : {[Content-Length, 510], [Content-Type, text/xml; charset=utf-8], [Date, Sun, 24 Jul 2022 13:30:55 GMT], [Server, Microsoft-IIS/10.0]}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 510



PS C:\Windows\system32> _
```

Lesson #2: Defaults are an attacker's best friend (intro)

- Before that: why defaults?
- Different CSP approaches to defaults
- Common + vulnerable → dangerous
- Selection of IAM-focused default risks

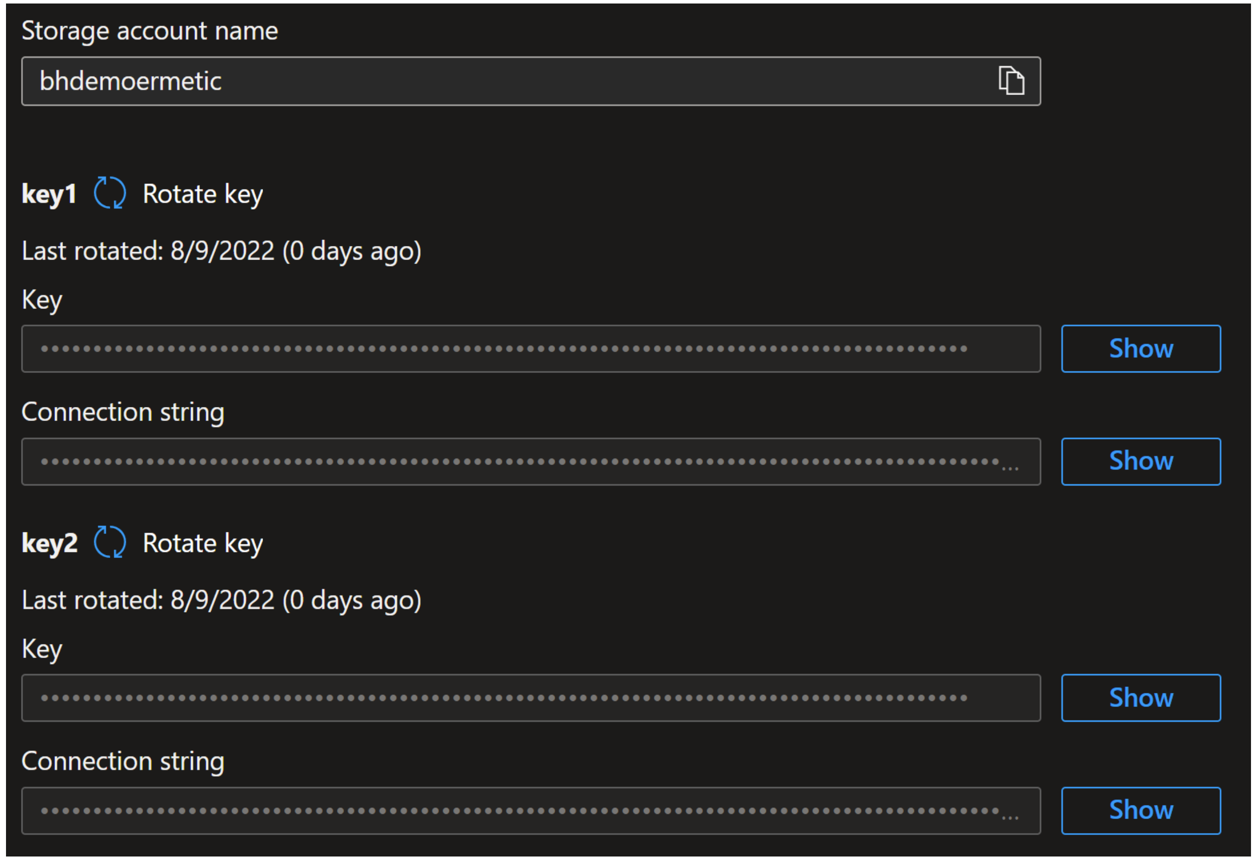
Lesson #2: Defaults are an attacker's best friend (AWS)

- AWS managed policies: Inherently broad permissions
- A "temporary" fix that becomes permanent
- **Attackers can leverage:**
  - ReadOnlyAccess
  - CloudTrailReadOnlyAccess
  - PassRole
  - Permission modifiers
  - AssumeRole

Lesson #2: Defaults are an attacker’s best friend (Azure)

- Custom role limits (5000)
- **Attackers can leverage:** Read permissions, Assignment permissions (self-assignment)
- Access keys → IAM bypass, created by default

Lesson #2: Defaults (Azure access keys)



Storage account name

bhdemoermetic

key1 Rotate key

Last rotated: 8/9/2022 (0 days ago)

Key

•• Show

Connection string

•••... Show

key2 Rotate key

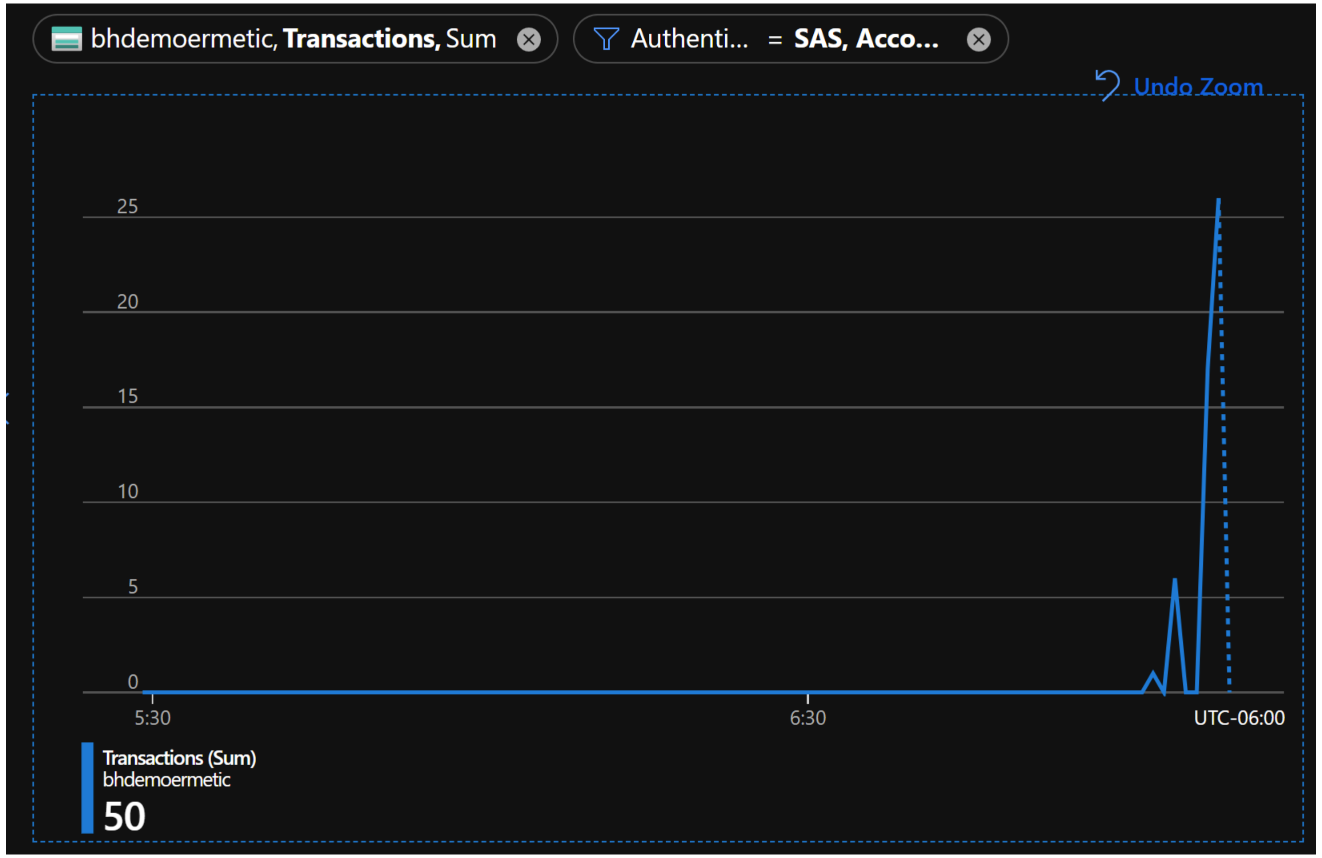
Last rotated: 8/9/2022 (0 days ago)

Key

•• Show

Connection string

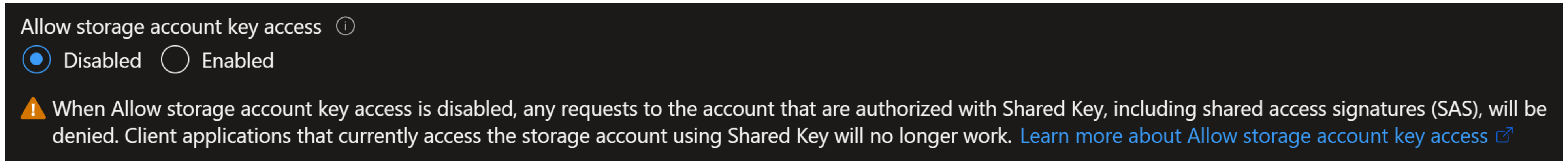
•••... Show



bhdemoermetic, **Transactions,** Sum

Authenti... = **SAS, Acco...**

Transactions (Sum)
bhdemoermetic
50



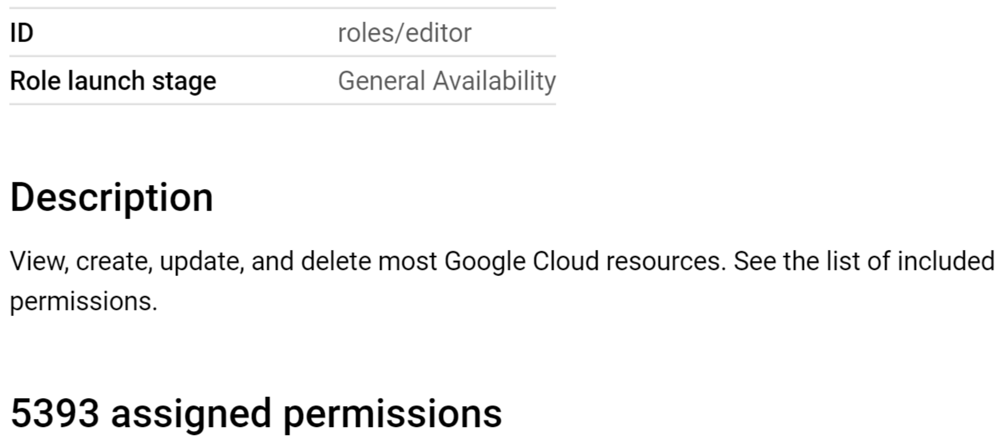
Allow storage account key access ⓘ

● Disabled ○ Enabled

⚠ When Allow storage account key access is disabled, any requests to the account that are authorized with Shared Key, including shared access signatures (SAS), will be denied. Client applications that currently access the storage account using Shared Key will no longer work. Learn more about Allow storage account key access

Lesson #2: Defaults are an attacker's best friend (GCP)

- **Basic roles** (Viewer, Editor) have strong and broad permissions
- GCE legacy mechanism: **Access scopes**
- **Default service accounts**
- Compute engine default service account



ID	roles/editor
Role launch stage	General Availability

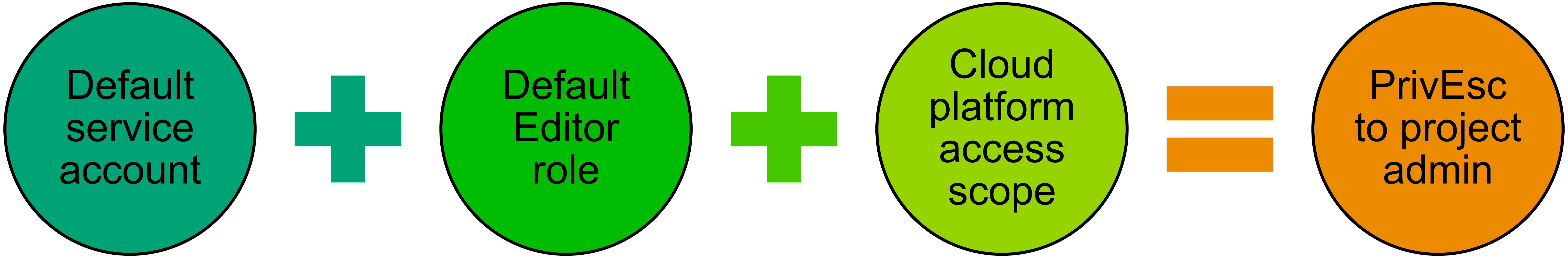
Description

View, create, update, and delete most Google Cloud resources. See the list of included permissions.

5393 assigned permissions

Lesson #2: Defaults are an attacker’s best friend (GCP)

- Attacker’s perspective 😈



Default service account + Default Editor role + Cloud platform access scope = PrivEsc to project admin

Lesson #3: Logs have limits

- Logging is important!
- To know what’s going on, detection, IR
- To build better permissions
- **Attackers can hide behind:** unlogged APIs, opaque APIs, log manipulation, distributed logging
- Log whatever you can (afford to)

Lesson #3: Logs have limits (AWS)

- Passive reconnaissance
- Data actions
- CloudTrail manipulation
- Cross-account data exfiltration[1]

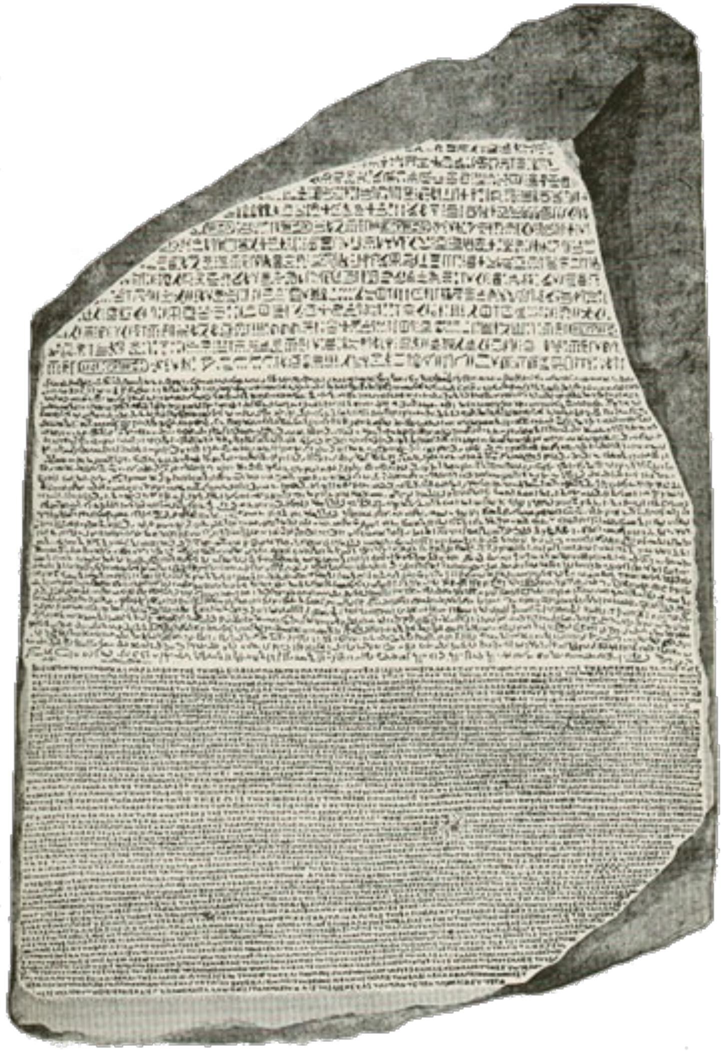
[1] Kat Traxler, Vectra AI, https://www.vectra.ai/blogpost/abusing-the-replicator-silently-exfiltrating-data-with-the-aws-s3-replication-service

Lesson #3: Logs have limits (Azure)

- Read actions are not logged to the activity log
- Distributed logging

Lesson #3: Logs have limits (Multicloud)

- Multiple clouds multiply log dispersal
- Consolidated logging has very different schemas
- No one-to-one translation
- No magic solution…



black hat®
USA 2022

Practical Practices for Defenders

1) Limiting the effect of mistakes

- **One** AWS account/GCP project/Azure resource group **per workload**
- **Deploy** organizational policies to limit disasters
- **Avoid** permanent credentials
- **Secure** human identities

2) Sculpting permissions from marble or clay

Clay (constructive)

- **Challenge:** knowing exactly what you need
- **Risk:** dysfunctionality

Marble (reductive)

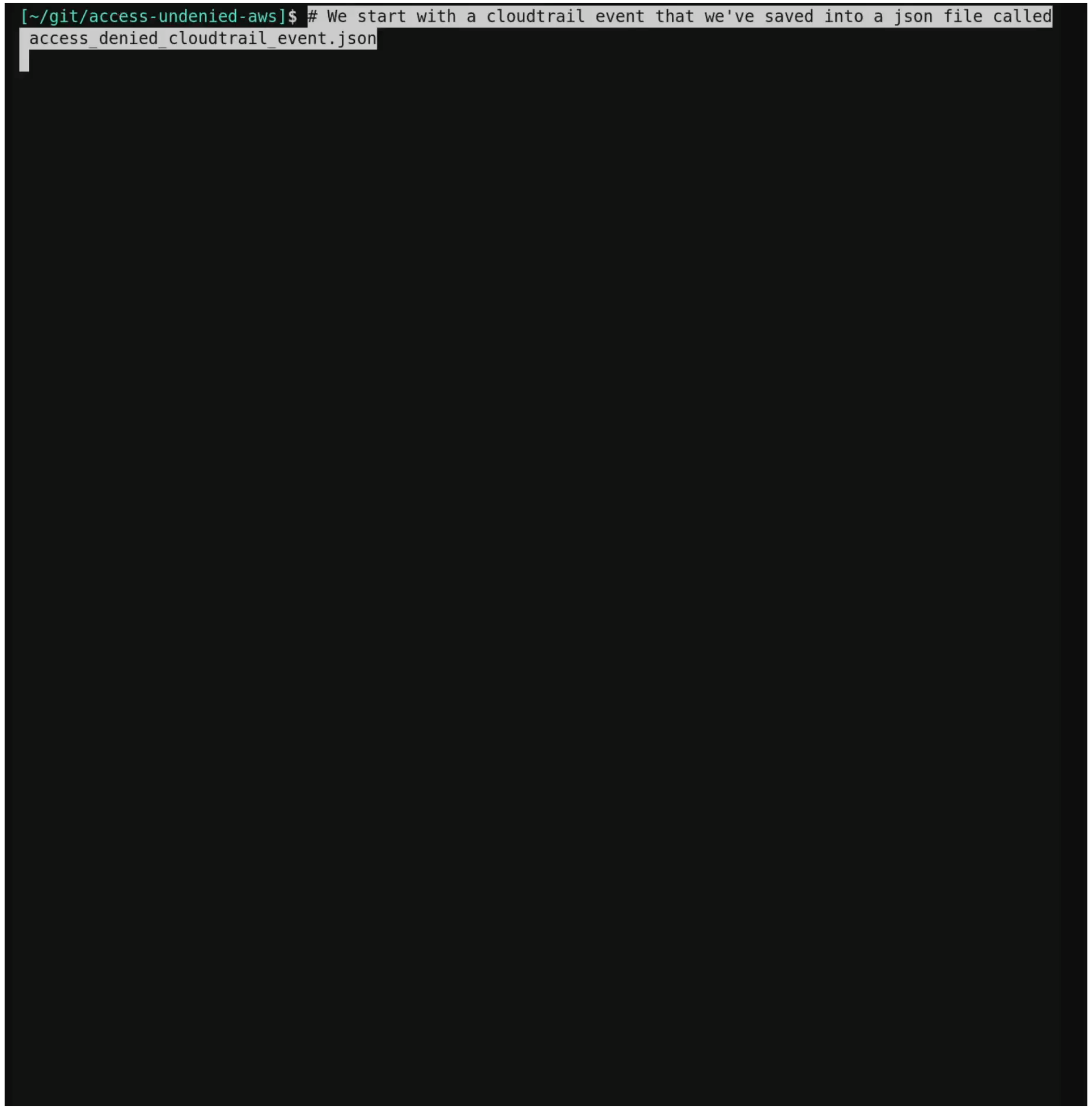
- **Challenge:** proving a negative
- **Risk:** Overpermissive

In practice: many choose marble, and then never cut down permissions

Recommendation: hybrid approach

Access Undenied on AWS

- Built to make clay sculpting easier
- Some deny messages are not detailed
- Built to prevent permission sprawl
- Scans SCPs, permission boundaries, identity policies and resource policies
- Tells you exactly what permission to add (or which deny policy to modify)



```
[~/git/access-undenied-aws]$ # We start with a cloudtrail event that we've saved into a json file called
 access_denied_cloudtrail_event.json
```

Tooling

- **Clay open-source tools (AWS):** **policy-sentry** (Salesforce, Kinnaird McQuade), **iamlive** (Ian McKay), **access-undenied-aws**
- **Marble open-source tools (AWS):** **Cloudtracker** (Duo Labs, Scott Piper), **Repokid** (Netflix), **IamSpy** (WithSecure, Nick Jones, Mohit Gupta), **PMapper** (NCC Group, Erik Steringer), **Cloudsplaining** (Salesforce, Kinnaird McQuade)

Questions?