

RSA® Conference 2022

San Francisco & Digital | February 7 – 10

TRANSFORM

SESSION ID: CSCS-R05

CloudTrail Logging Internals: Investigating AWS Security Incidents

Omer Gull

Senior Security Researcher
Hunters
@GullOmer



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, RSA Security LLC 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.

©2022 RSA Security LLC or its affiliates. All rights reserved. RSA Conference logo, RSA and other trademarks are trademarks of RSA Security LLC or its affiliates.

Disclaimer #2

- All work presented here was done by the amazing Eliav Livneh
- Couldn't be here due to very happy circumstances 😊

- eliav@hunters.ai
- /EliavLivneh 



Agenda

- The AWS Control Plane
- Investigating an incident
 - What questions to ask
 - How to ask those questions
 - CloudTrail internals
- Examples of CloudTrail-complexity

The AWS Control Plane

- Actions taken using AWS API
- Over 300 services, Over 10K actions
- Attacks in the cloud will often pivot between the Control and the Data planes
- Typically, defenders are more comfortable in the Data plane

Data Sources for Control Plane Investigation

- Config - Audit and evaluate AWS resources
- CloudTrail - Detailed log of all API actions

GuardDuty Alert as an Example

Discovery: IAMUser/AnomalousBehavior [⊕](#) [⊖](#)

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

[Investigate with Detective](#)

Overview	
Severity	HIGH ⊕ ⊖
Region	us-east-1
Count	1
Account ID	222222222222 ⊕ ⊖
Resource ID	ASIAIOSFODNN7EXAMPLE
Created at	02-07-2022 12:16:28 (3 days ago)
Updated at	02-07-2022 12:45:05 (3 days ago)

Resource affected	
Resource role	TARGET ⊕ ⊖
Resource type	AccessKey ⊕ ⊖
Access key ID	ASIAIOSFODNN7EXAMPLE ⊕ ⊖
Principal ID	AROAB2GAA12IOELQVWWIJ:itadmin ⊕ ⊖
User type	AssumedRole ⊕ ⊖
User name	admin ⊕ ⊖

Action	
Action type	AWS_API_CALL ⊕ ⊖
API	ListUsers ⊕ ⊖
Service name	iam.amazonaws.com ⊕ ⊖
First seen	02-07-2022 12:09:39 (3 days ago)
Last seen	02-07-2022 12:32:50 (3 days ago)

Actor	
Caller type	Remote IP ⊕ ⊖
IP address	21.86.48.7 ⊕ ⊖

GuardDuty Alert as an Example

Discovery: IAMUser/AnomalousBehavior [⊕](#) [⊖](#)

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in [Discovery](#) tactics were invoked by user [AssumedRole : admin](#), under [anomalous circumstances](#). Such activity is not typically seen from this user. [Info](#)

Overview

Severity	HIGH	⊕ ⊖
Region	us-east-1	⊕ ⊖
Count	1	⊕ ⊖
Account ID	222222222222	⊕ ⊖
Resource ID	ASIAIOSFODNN7EXAMPLE	⊕ ⊖
Created at	02-07-2022 12:16:28 (3 days ago)	⊕ ⊖
Updated at	02-07-2022 12:45:05 (3 days ago)	⊕ ⊖

Resource affected

Resource role	TARGET	⊕ ⊖
Resource type	AccessKey	⊕ ⊖
Access key ID	ASIAIOSFODNN7EXAMPLE	⊕ ⊖
Principal ID	AROAB2GAA12IOELQVVWIJ:itadmin	⊕ ⊖
User type	AssumedRole	⊕ ⊖
User name	admin	⊕ ⊖

Action

Action type	AWS_API_CALL	⊕ ⊖
API	ListUsers	⊕ ⊖
Service name	iam.amazonaws.com	⊕ ⊖
First seen	02-07-2022 12:09:39 (3 days ago)	⊕ ⊖
Last seen	02-07-2022 12:32:50 (3 days ago)	⊕ ⊖

Actor

Caller type	Remote IP	⊕ ⊖
IP address	21.86.48.7	⊕ ⊖

Basic triage questions

Discovery: IAMUser/AnomalousBehavior [⊕](#) [⊖](#)

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

[Investigate with Detective](#)

Overview	
Severity	HIGH
Region	us-east-1
Count	1
Account ID	222222222222
Resource ID	ASIAIOSFODNN7EXAMPLE
Created at	02-07-2022 12:16:28 (3 days ago)
Updated at	02-07-2022 12:45:05 (3 days ago)

Resource affected	
Resource role	TARGET
Resource type	AccessKey
Access key ID	ASIAIOSFODNN7EXAMPLE
Principal ID	AROAB2GAA12IOELQVVWIJ:itadmin
User type	AssumedRole
User name	admin

Action	
Action type	AWS_API_CALL
API	ListUsers
Service name	iam.amazonaws.com
First seen	02-07-2022 12:09:39 (3 days ago)
Last seen	02-07-2022 12:32:50 (3 days ago)

Actor	
Caller type	Remote IP
IP address	21.86.48.7

Basic triage questions

1. Who did it?

Discovery: IAMUser/AnomalousBehavior [⊕](#) [⊖](#)

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

[Investigate with Detective](#)

Overview	
Severity	HIGH
Region	us-east-1
Count	1
Account ID	222222222222
Resource ID	ASIAIOSFODNN7EXAMPLE
Created at	02-07-2022 12:16:28 (3 days ago)
Updated at	02-07-2022 12:45:05 (3 days ago)

Resource affected	
Resource role	TARGET
Resource type	AccessKey
Access key ID	ASIAIOSFODNN7EXAMPLE
Principal ID	AROAB2GAA12IOELQVWVJ:itadmin
User type	AssumedRole
User name	admin

Action		
Action type	AWS_API_CALL	⊕ ⊖
API	ListUsers	⊕ ⊖
Service name	iam.amazonaws.com	⊕ ⊖
First seen	02-07-2022 12:09:39 (3 days ago)	
Last seen	02-07-2022 12:32:50 (3 days ago)	
Actor		
Caller type	Remote IP	⊕ ⊖
IP address	21.86.48.7	⊕ ⊖

Basic triage questions

Discovery: IAMUser/AnomalousBehavior  

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

 [Investigate with Detective](#)

Overview	
Severity	HIGH
Region	us-east-1
Count	1
Account ID	222222222222
Resource ID	ASIAIOSFODNN7EXAMPLE
Created at	02-07-2022 12:16:28 (3 days ago)
Updated at	02-07-2022 12:45:05 (3 days ago)

Resource affected	
Resource role	TARGET
Resource type	AccessKey
Access key ID	ASIAIOSFODNN7EXAMPLE
Principal ID	AROAB2GAA12IOELQVWVIJ:itadmin
User type	AssumedRole
User name	admin

1. Who did it?
2. What did they do?

Action		
Action type	AWS_API_CALL	 
API	ListUsers	 
Service name	iam.amazonaws.com	 
First seen	02-07-2022 12:09:39 (3 days ago)	
Last seen	02-07-2022 12:32:50 (3 days ago)	

Actor		
Caller type	Remote IP	 
IP address	21.86.48.7	 

Basic triage questions

Discovery: IAMUser/AnomalousBehavior [⊕](#) [⊖](#)

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

[Investigate with Detective](#)

Overview	
Severity	HIGH
Region	us-east-1
Count	1
Account ID	222222222222
Resource ID	ASIAIOSFODNN7EXAMPLE
Created at	02-07-2022 12:16:28 (3 days ago)
Updated at	02-07-2022 12:45:05 (3 days ago)

Resource affected	
Resource role	TARGET
Resource type	AccessKey
Access key ID	ASIAIOSFODNN7EXAMPLE
Principal ID	AROAB2GAA12IOELQVWVIJ:itadmin
User type	AssumedRole
User name	admin

1. Who did it?
2. What did they do?
3. From where?

Action		
Action type	AWS_API_CALL	⊕ ⊖
API	ListUsers	⊕ ⊖
Service name	iam.amazonaws.com	⊕ ⊖
First seen	02-07-2022 12:09:39 (3 days ago)	
Last seen	02-07-2022 12:32:50 (3 days ago)	

Actor		
Caller type	Remote IP	⊕ ⊖
IP address	21.86.48.7	⊕ ⊖

Basic triage questions

Discovery: IAMUser/AnomalousBehavior [⊕](#) [⊖](#)

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

[Investigate with Detective](#)

Overview	
Severity	HIGH
Region	us-east-1
Count	1
Account ID	222222222222
Resource ID	ASIAIOSFODNN7EXAMPLE
Created at	02-07-2022 12:16:28 (3 days ago)
Updated at	02-07-2022 12:45:05 (3 days ago)

Resource affected	
Access key ID	ASIAIOSFODNN7EXAMPLE
Principal ID	AROAB2GAA12IOELQVVWIJ:itadmin
User type	AssumedRole
User name	admin

1. Who did it?
2. What did they do?
3. From where?

Action	
Action type	AWS_API_CALL
API	ListUsers
Service name	iam.amazonaws.com
First seen	02-07-2022 12:09:39 (3 days ago)
Last seen	02-07-2022 12:32:50 (3 days ago)

Actor	
Caller type	Remote IP
IP address	21.86.48.7

Basic triage questions

Discovery: IAMUser/AnomalousBehavior

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

Overview

Severity	HIGH	
Region	us-east-1	
Count	1	
Account ID	222222222222	
Resource ID	ASIAIOSFODNN7EXAMPLE	
Created at	02-07-2022 12:16:28 (3 days ago)	
Updated at	02-07-2022 12:45:05 (3 days ago)	

Resource affected

Resource role	TARGET	
Resource type	AccessKey	
Access key ID	ASIAIOSFODNN7EXAMPLE	
Principal ID	AROAB2GAA12IOELQVVWIJ:itadmin	
User type	AssumedRole	
User name	admin	

1. Who did it?
2. What did they do?
3. From where?

Action		
Action type	AWS_API_CALL	
API	ListUsers	
Service name	iam.amazonaws.com	

Actor		
Caller type	Remote IP	
IP address	21.86.48.7	

Basic triage questions

Discovery: IAMUser/AnomalousBehavior [⊕](#) [⊖](#)

Finding ID: [bb6bdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

[Investigate with Detective](#)

Overview	
Severity	HIGH
Region	us-east-1
Count	1
Account ID	222222222222
Resource ID	ASIAIOSFODNN7EXAMPLE
Created at	02-07-2022 12:16:28 (3 days ago)
Updated at	02-07-2022 12:45:05 (3 days ago)

Resource affected	
Resource role	TARGET
Resource type	AccessKey
Access key ID	ASIAIOSFODNN7EXAMPLE
Principal ID	AROAB2GAA12IOELQVVWIJ:itadmin
User type	AssumedRole
User name	admin

1. Who did it?
2. What did they do?
3. From where?

Action	
Action type	AWS_API_CALL
API	ListUsers
Service name	iam.amazonaws.com
First seen	02-07-2022 12:09:39 (3 days ago)
Last seen	02-07-2022 12:32:50 (3 days ago)

Actor	
IP address	21.86.48.7

Basic triage questions

Discovery: IAMUser/AnomalousBehavior  

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

[Investigate with Detective](#)

Overview	
Severity	HIGH
Region	us-east-1
Count	1
Account ID	222222222222
Resource ID	ASIAIOSFODNN7EXAMPLE
Created at	02-07-2022 12:16:28 (3 days ago)
Updated at	02-07-2022 12:45:05 (3 days ago)

Resource affected	
Access key ID	ASIAIOSFODNN7EXAMPLE
Principal ID	AROAB2GAA12IOELQVVWIJ:itadmin
User type	AssumedRole
User name	admin

1. Who did it?
2. What did they do?
3. From where?

Action	
Action type	AWS_API_CALL
API	ListUsers
Service name	iam.amazonaws.com

Actor	
IP address	21.86.48.7

Basic triage questions

Discovery: IAMUser/AnomalousBehavior

Finding ID: [bbedbdd845d7fa220a8f10fe73276617](#)

High APIs commonly used in Discovery tactics were invoked by user AssumedRole : admin, under anomalous circumstances. Such activity is not typically seen from this user. [Info](#)

Overview

Severity	HIGH	
Region	us-east-1	
Count	1	
Account ID	222222222222	
Resource ID	ASIAIOSFODNN7EXAMPLE	
Created at	02-07-2022 12:16:28 (3 days ago)	
Updated at	02-07-2022 12:45:05 (3 days ago)	

Resource affected

Resource role	TARGET	
Resource type	AccessKey	
Access key ID	ASIAIOSFODNN7EXAMPLE	
Principal ID	AROAB2GAA12IOELQVVWIJ:itadmin	
User type	AssumedRole	
User name	admin	

- 1. Who did it?**
- 2. What did they do?**
- 3. From where?**

We'll focus on these

Action		
Action type	AWS_API_CALL	
API	ListUsers	
Service name	iam.amazonaws.com	
First seen	02-07-2022 12:09:39 (3 days ago)	
Last seen	02-07-2022 12:32:50 (3 days ago)	

Actor		
Caller type	Remote IP	
IP address	21.86.48.7	

Today's talk

- Methodology for finding out who did it and what they did
- CloudTrail-only
- Maximum accuracy

Disclaimer

- There are simpler ways that can give 80% results
- We're going for 99%



Before we continue...

Prerequisites:

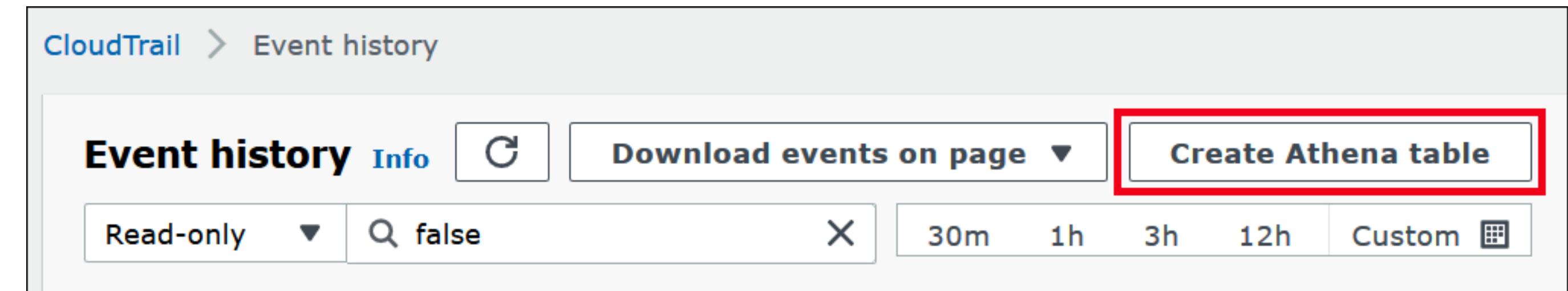
- CloudTrail logging enabled
- Logs accessible and queryable (Athena/Snowflake/etc.)
- Best-practice CloudTrail configuration

CloudTrail Best Practices

- Although CloudTrail provides 90 days of history - Create a Trail
- Apply trails to all regions in all your AWS accounts/organization
- Log both Read/Write API activity
- Log Data events for sensitive resources

Athena & CloudTrail

- CloudTrail stores all logs in S3 buckets
- Logs are saved as gzipped JSON
- Athena allows you to interactively query S3 data without managing any clusters/warehouses
- Can be easily configured from CloudTrail's console



Answering “who did it?”

- We know that following ARN was used:
 - **arn:aws:sts::222222222222:assumed-role/admin/itadmin**
- IAM ARN syntax
 - **arn:aws:sts::aws-account-id:assumed-role/role/session-name**
- But who assumed this role?
 - This is non-trivial to answer accurately

AWS Identity Chains

- There are several API calls that pivot between identities.
- Some pivot between AWS accounts, and/or change the identity ARN.
- They all generate temporary credentials.

AWS Identity Chains

- There are several API calls that pivot between identities.
- Some pivot between AWS accounts, and/or change the identity ARN.
- They all generate temporary credentials.

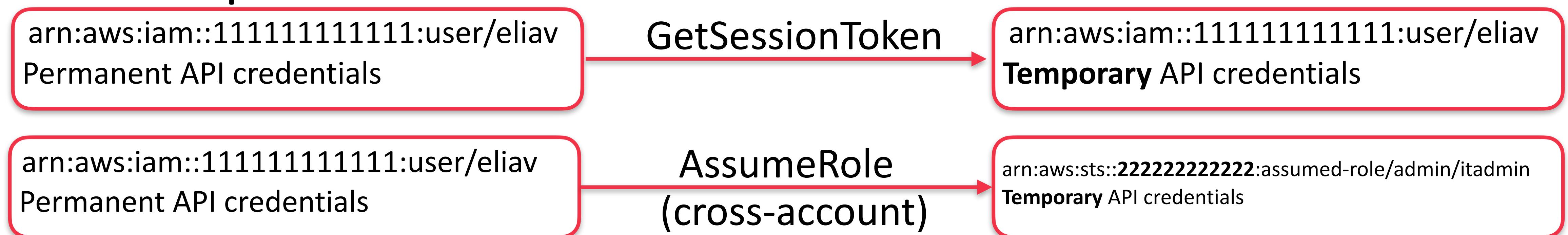
Examples



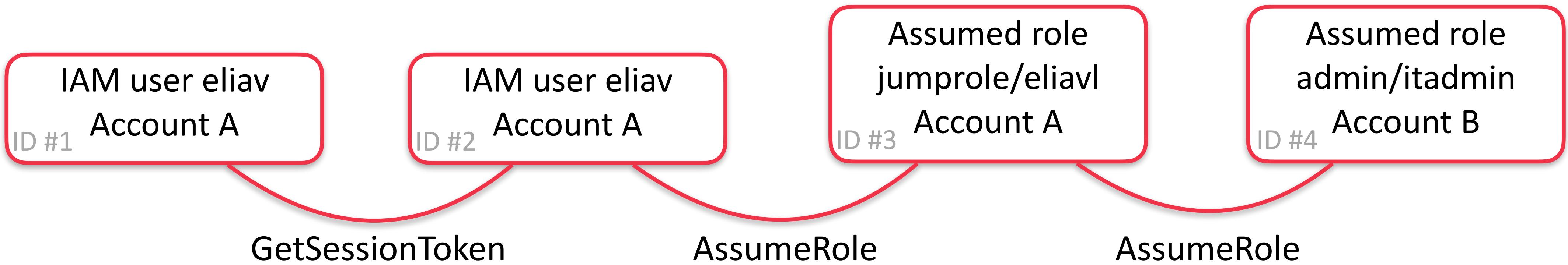
AWS Identity Chains

- There are several API calls that pivot between identities.
- Some pivot between AWS accounts, and/or change the identity ARN.
- They all generate temporary credentials.

Examples



AWS Identity Chains - more complex



- By following the chain backwards, we know “who did it”

Next question - “What did they do?”

- We now have all ARNs related to the activity
- So, shall we just search for all events with the ARNs?



ARNs aren't unique; access keys are

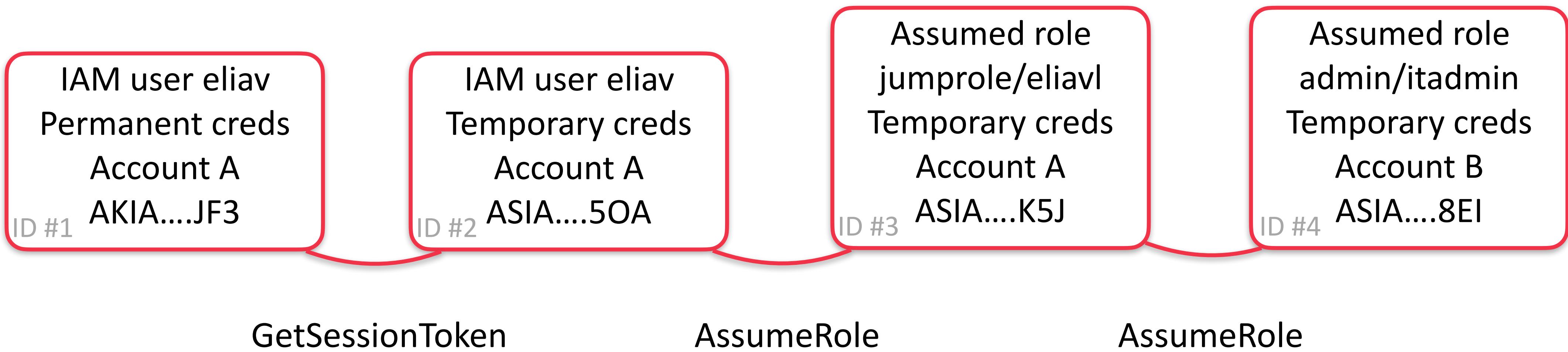
- Multiple people/services can use “arn:aws:sts::222222222222:assumed-role/admin/itadmin” simultaneously.
- Searching for all events with this ARN can lead to results from different “sessions”.
- So, what are 100% unique*? API access keys!

Following the access keys

- With API calls, easy - just follow the identity chain.

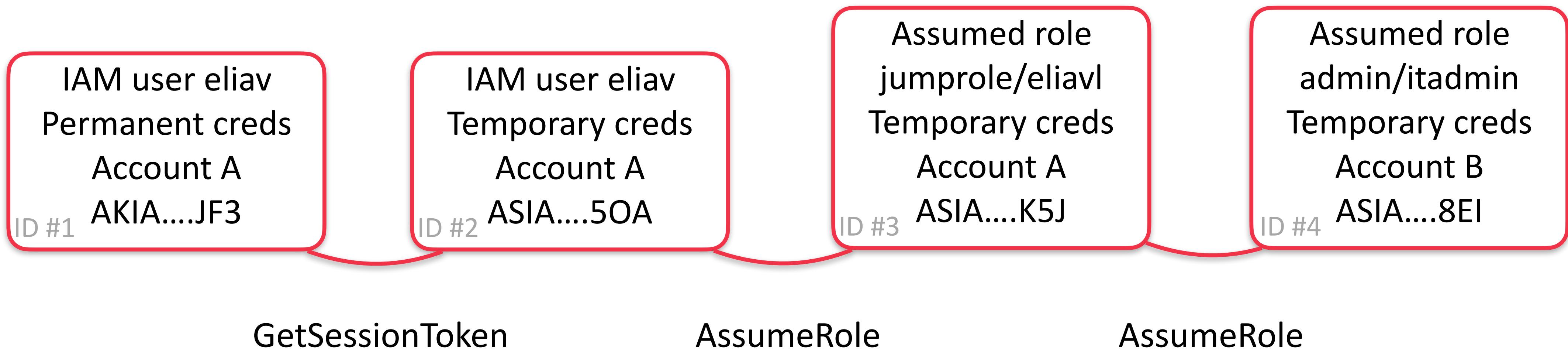
Following the access keys

- With API calls, easy - just follow the identity chain.



Following the access keys

- With API calls, easy - just follow the identity chain.



- However, web sessions are a bit different
 - Web console require username and password
 - All AWS API calls require access key + secret key

Web console internals

- User logs in to web with username and password
- Behind the scenes, temporary API credentials are dynamically generated
- Web console makes API calls on your behalf and displays results
- Different access key generated per region
- How to find these API access keys?

Web console internals

- User logs in to web with username and password
- Behind the scenes, temporary API credentials are dynamically generated
- Web console makes API calls on your behalf and displays results
- Different access key generated per region
- How to find these API access keys?



The screenshot shows the AWS CloudFormation console interface. At the top, there's a navigation bar with the AWS logo, a "Services" dropdown, a search bar containing "Search for services, features, blogs, docs, and more" with a keyboard shortcut "[Option+S]", and a "Resource Groups & Tag Editor" button.

A prominent modal window titled "New EC2 Experience" is open on the left, asking "Tell us what you think" and listing options like "EC2 Dashboard", "EC2 Global View", "Events", "Tags", "Limits", and "Instances".

The main content area is titled "Resources" and displays a summary of EC2 resources in the "US East (N. Virginia) Region". It includes a "EC2 Global view" button and three icons: a circular arrow, a gear, and a refresh symbol.

The resource summary table:

Resource Type	Count
Instances (running)	4
Dedicated Hosts	0
Elastic IPs	2
Instances	6
Key pairs	1
Load balancers	2
Placement groups	0
Security groups	11
Snapshots	2
Volumes	7

At the bottom of the main content area, there's a callout box with an info icon and the text: "Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. [Learn more](#)".

Web console internals

```
"eventName": "ConsoleLogin",
"eventTime": "2021-08-27T19:42:56Z",
"userIdentity": {
  "arn": "arn:aws:iam::111111111111:user/eliasv",
  ...
}
```

Web console internals

```
"eventName": "ConsoleLogin",
"eventTime": "2021-08-27T19:42:56Z",
"userIdentity": {
  "arn": "arn:aws:iam::111111111111:user/eliasv",
  ...
}
```

```
"eventName": "DescribeVolumeStatus",
"eventName": "DescribeAvailabilityZones",
"eventName": "DescribeInstances",
"eventTime": "2021-08-27T19:43:02Z",
"userIdentity": {
  "accessKeyId": "ASIAEXAMPLE01234567",
  "arn": "arn:aws:iam::111111111111:user/eliasv",
  "sessionContext": {
    "attributes": {
      "creationDate": "2019-01-21T19:42:56Z",
    },
    },
  }
}
```

Web console internals

```
"eventName": "ConsoleLogin",
"eventTime": "2021-08-27T19:42:56Z",
"userIdentity": {
  "arn": "arn:aws:iam::111111111111:user/elav",
  ...
}
```

```
"eventName": "DescribeVolumeStatus",
"eventName": "DescribeAvailabilityZones",
"eventName": "DescribeInstances",
"eventTime": "2021-08-27T19:43:02Z",
"userIdentity": {
  "accessKeyId": "ASIAEXAMPLE01234567",
  "arn": "arn:aws:iam::111111111111:user/elav",
  "sessionContext": {
    "attributes": {
      "creationDate": "2019-01-21T19:42:56Z",
    },
    },
  }
}
```

Web console internals

```
"eventName": "ConsoleLogin",
"eventTime": "2021-08-27T19:42:56Z",
"userIdentity": {
  "arn": "arn:aws:iam::111111111111:user/elav",
  ...
}
```

```
"eventName": "DescribeVolumeStatus",
"eventName": "DescribeAvailabilityZones",
"eventName": "DescribeInstances",
"eventTime": "2021-08-27T19:43:02Z",
"userIdentity": {
  "accessKeyId": "ASIAEXAMPLE01234567",
  "arn": "arn:aws:iam::111111111111:user/elav",
  "sessionContext": {
    "attributes": {
      "creationDate": "2019-01-21T19:42:56Z",
    },
    },
  }
}
```

Web console internals

```
"eventName": "ConsoleLogin",
"eventTime": "2021-08-27T19:42:56Z",
"userIdentity": {
  "arn": "arn:aws:iam::111111111111:user/eliasv",
  ...
}
```

```
"eventName": "DescribeVolumeStatus",
"eventName": "DescribeAvailabilityZones",
"eventName": "DescribeInstances",
"eventTime": "2021-08-27T19:43:02Z",
"userIdentity": {
  "accessKeyId": "ASIAEXAMPLE01234567",
  "arn": "arn:aws:iam::111111111111:user/eliasv",
  "sessionContext": {
    "attributes": {
      "creationDate": "2019-01-21T19:42:56Z",
    },
    },
  }
}
```

Web console internals

or **SwitchRole/RenewRole**

```
"eventName": "ConsoleLogin",
"eventTime": "2021-08-27T19:42:56Z",
"userIdentity": {
  "arn": "arn:aws:iam::111111111111:user/eliasv",
  ...
}
```

```
"eventName": "DescribeVolumeStatus",
"eventName": "DescribeAvailabilityZones",
"eventName": "DescribeInstances",
"eventTime": "2021-08-27T19:43:02Z",
"userIdentity": {
  "accessKeyId": "ASIAEXAMPLE01234567",
  "arn": "arn:aws:iam::111111111111:user/eliasv",
  "sessionContext": {
    "attributes": {
      "creationDate": "2019-01-21T19:42:56Z",
    },
    },
  }
}
```

Web console internals - recap

- User logs in to web with username and password
- Behind the scenes, temporary API credentials are dynamically generated
- Web console makes API calls on your behalf and displays results
- Different access key generated per region
- We now know how to find all relevant access keys for the session

Now what?

- Now we need to see which actions those access keys did during the session
- We can simply search for all CloudTrail events with these access keys!
- Well... technically true, but it's hard to consume:
 - Every single API call is logged separately
 - Simple web sessions often generate thousands of events
 - Logging is not entirely comprehensive
 - Varying levels of logging across services

Some tips in constructing meaningful sessions

- We are going to create easily consumable AWS sessions
- To successfully do so it is important to understand some CloudTrail internals

Some CloudTrail corners... #1

resources field

- “on what was the action done”
- Critical for context
 - e.g. Prod DC vs. dev test machine
- Only being populated by ~10%-15% of the services :(
- Non-populating services: IAM, EC2, Lambda, RDS, Secrets Manager, etc...
- We use this field when populated

resources

A list of resources accessed in the event. The field can contain the following information:

- Resource ARNs
- Account ID of the resource owner
- Resource type identifier in the format: AWS::*aws-service-name*::*data-type-name*

Optional: True

Some CloudTrail corners... #2

readOnly field

- Indicates if API call changed resources or not
- Helps zooming in on the sensitive/interesting actions
- ~~Isn't always populated :(~~
 - Historically, only ~20% populated
 - Q4/20, 20% -> 100% population!
- Isn't necessarily a good indicator - these are read-only:
 - STS AssumeRole, GetSessionToken
 - S3 GetObject
 - So, we maintain a list of “sensitive read-only” events

readOnly

Identifies whether this operation is a read-only operation. This can be one of the following values:

- `true` – The operation is read-only (for example, `DescribeTrails`).
- `false` – The operation is write-only (for example, `DeleteTrail`).

Optional: True

Some CloudTrail corners... #3

- When **resources** unpopulated, we can use **requestParameters**
 - Each API targeting a resource has some **request parameter**
- Unfortunately, the parameter names aren't standardized
 - For example, `AttachRolePolicy` receives `policyArn` but `UpdatePolicy` receives `policyId`
- Also, when API call fails with IAM `AccessDenied`, **requestParameters** is empty :(
 - However, **errorMessage** is not :)

```
{  
  "errorCode": "AccessDenied",  
  "eventName": "GetKeyRotationStatus",  
  "eventSource": "kms.amazonaws.com",  
  "requestParameters": null,  
  ...  
  "errorMessage": "User: arn:aws:sts::ACCOUNT_ID:assumed-role/ROLE_NAME/ROLE_SESSION_NAME  
  is not authorized to perform: kms:GetKeyRotationStatus on resource:  
  arn:aws:kms:us-east-1:ACCOUNT_ID:key/KEY_ID",  
}
```

Some CloudTrail corners... #4

- A common parameter in RequestParameters
- filterSet:items
 - Available in “resource-searching” API calls
 - Very helpful in understanding the intention of the session

```
{  
  "eventName": "DescribeInstances",  
  "eventSource": "ec2.amazonaws.com",  
  "requestParameters": {  
    "filterSet": {"items":  
      [{"name": "tag:Name", "valueSet": {"items": [{"value": "web-app-001-prod"}]}}]  
    }  
  }  
}
```

So, best effort

To summarize all events in a consumable way, we group by API call, and:

- Count number of total calls
- Count successful / failed calls
- Extract a sample resources JSON (if populated)
- Extract target resource from IAM access denied error messages
- Extract a sample requestParameters JSON (for non-read-only events)
- Extract a sample requestParameters filter (for read-only events)

From this



	EVENT_TIME	AWS_REGION	EVENT_ID	EVENT_NAME	EVENT_SOURCE	EVENT_TYPE
1	2021-09-04 14:16:27.000000000 +00:00	us-east-1	748159bd-d469-4603-accf-ac516ca3601f	DescribeEventAggregates	health.amazonaws.com	AwsApiCall
2	2021-09-04 14:18:54.000000000 +00:00	us-east-1	606e72ae-e862-4e40-a6c9-93eb0badd214	DescribeEventAggregates	health.amazonaws.com	AwsApiCall
3	2021-09-04 14:18:40.000000000 +00:00	us-east-1	71e4e29f-76dd-488e-8168-527ae2ebb340	DescribeEventAggregates	health.amazonaws.com	AwsApiCall
4	2021-09-04 14:21:31.000000000 +00:00	us-east-1	be13196b-9768-45f1-8c8b-9c26824ea778	DescribeEventAggregates	health.amazonaws.com	AwsApiCall
5	2021-09-04 14:23:58.000000000 +00:00	us-east-1	89fadcc53-bcd7-4b80-9c6e-f841eb69cddd	DescribeEventAggregates	health.amazonaws.com	AwsApiCall
6	2021-09-04 14:26:35.000000000 +00:00	us-east-1	87378e89-2074-40f6-a354-8af6c550c1fd	DescribeEventAggregates	health.amazonaws.com	AwsApiCall
7	2021-09-04 17:00:27.000000000 +00:00	us-west-2	2d9a91af-eea6-470c-afbd-8fe9ba39f65c	DescribeRouteTables	ec2.amazonaws.com	AwsApiCall
8	2021-09-04 17:00:27.000000000 +00:00	us-west-2	3233fa17-3dce-463c-b6e8-7fac39e03d5d	DescribeNetworkAcls	ec2.amazonaws.com	AwsApiCall
9	2021-09-04 17:00:26.000000000 +00:00	us-west-2	90b35cfcd-252b-4f70-be4e-49c8be2db35b	DescribeSubnets	ec2.amazonaws.com	AwsApiCall
10	2021-09-04 17:00:27.000000000 +00:00	us-west-2	aaceb1f7-cbbe-448d-91be-3d0db931610a	DescribeSecurityGroups	ec2.amazonaws.com	AwsApiCall
11	2021-09-04 17:00:47.000000000 +00:00	us-west-2	6ebc5f73-0152-400e-b4ba-a490a958a6d0	DescribeAccountAttributes	ec2.amazonaws.com	AwsApiCall
12	2021-09-04 17:00:48.000000000 +00:00	us-west-2	2357076f-44af-4927-8c8d-c062f75b4b00	DescribeSecurityGroups	ec2.amazonaws.com	AwsApiCall
13	2021-09-04 17:00:47.000000000 +00:00	us-west-2	bd129cc5-b56d-4c1a-8f1d-914e286a1a55	DescribeSpotPriceHistory	ec2.amazonaws.com	AwsApiCall
14	2021-09-04 17:00:47.000000000 +00:00	us-west-2	0c640dd8-7211-4f6b-a0be-e15016858f8b	DescribeSpotPriceHistory	ec2.amazonaws.com	AwsApiCall
15	2021-09-04 17:00:48.000000000 +00:00	us-west-2	e62a5173-1df0-4bbb-95ed-ab618fa79495	DescribeSubnets	ec2.amazonaws.com	AwsApiCall
16	2021-09-04 17:00:48.000000000 +00:00	us-west-2	c747e99e-4a3c-4911-8639-ab6c07c04942	DescribeRouteTables	ec2.amazonaws.com	AwsApiCall
17	2021-09-04 17:00:48.000000000 +00:00	us-west-2	a8671bdb-db78-47b4-a26d-95cbf01b60a8	DescribeNetworkAcls	ec2.amazonaws.com	AwsApiCall

To this

ARN Session Statistics - read_only_event_statistics							
<input type="text"/> Search here.. ← →							
event_source ↑	event_name	request_parameters_sample	occurrences	success_count	error_message_sample	access_denied_target_r...	
iam.amazonaws.com	GetGroup	{"maxItems":1000,"groupName":"s3_READONLY"}	1	1	null	null	
iam.amazonaws.com	ListServiceSpecificCredentials	{"userName":"eliav.levy.test_user","serviceName":"codecommit.amazonaws.co..."	6	6	null	null	
iam.amazonaws.com	GetPolicyVersion	{"policyArn":"arn:aws:iam::111111111111:policy/eliav.levy.change.role.to.strong.rol..."	2	2	null	null	
iam.amazonaws.com	ListUserPolicies	{"maxItems":1000,"userName":"eliav.levy.test_user"}	1	1	null	null	
iam.amazonaws.com	ListUsers	{"maxItems":1000}	33	33	null	null	
iam.amazonaws.com	ListSSHPublicKeys	{"userName":"eliav.levy.test_user"}	3	3	null	null	
iam.amazonaws.com	GetUser	{"userName":"eliav.levy.test_user"}	1	1	null	null	
iam.amazonaws.com	ListAccessKeys	null	11	0	Must specify userName when calling with...	null	
iam.amazonaws.com	ListGroups	{"maxItems":1000}	16	16	null	null	
iam.amazonaws.com	GetUser	{"userName":"eliav.levy.test_user"}	17	17	null	null	
iam.amazonaws.com	ListRoles	{"maxItems":1000}	13	13	null	null	
iam.amazonaws.com	ListGroupsForUser	{"maxItems":1000,"userName":"eliav.levy.test_user"}	13	13	null	null	
iam.amazonaws.com	ListRoles	{"maxItems":1000}	10	10	null	null	
iam.amazonaws.com	GetPolicyVersion	{"policyArn":"arn:aws:iam::111111111111:policy/hunters-s3-upload","versionId":"v..."	5	5	null	null	
iam.amazonaws.com	ListAttachedGroupPolicies	{"groupName":"s3_READONLY"}	1	1	null	null	
iam.amazonaws.com	GetRole	null	253	253	null	null	
iam.amazonaws.com	ListRoleTags	{"roleName":"AWSReservedSSO_Admin_ab5efe6c115c678b"}	7	7	null	null	
iam.amazonaws.com	ListGroups	{"maxItems":1000}	1	1	null	null	
iam.amazonaws.com	GetAccountPasswordPolicy	null	50	0	The Password Policy with domain name 1...	null	
iam.amazonaws.com	GetAccountSummary	null	29	29	null	null	
iam.amazonaws.com	ListGroupsForUser	{"userName":"prod-code-reader-account"}	1	1	null	null	
iam.amazonaws.com	GetPolicy	{"policyArn":"arn:aws:iam::111111111111:policy/eliav.levy.change.role.to.strong.rol..."	2	2	null	null	

To this

ARN Session Statistics - read_only_event_statistics							
ARN Session Statistics - not_read_only_event_statistics							
<input type="text"/> Search here..							
<input type="checkbox"/> Drag here to set row groups							
event_source ↑	event_name	request_parameters_sample	access_denied_target_resource_sample	error_message_sample	occurrences	success_count	failure_count
iam.amazonaws.com	DetachUserPolicy	{"userName":"eliav.levy.test_user","policyArn":"arn:aws:iam::111111111111:policy/eliav.levy.second.p...}	null	null	1	1	0
iam.amazonaws.com	AttachUserPolicy	{"userName":"eliav.levy.test_user","policyArn":"arn:aws:iam::aws:policy/AmazonS3FullAccess"}	null	null	3	3	0
iam.amazonaws.com	GenerateServiceLastAc...	{"arn":"arn:aws:iam::111111111111:role/aws-reserved/sso.amazonaws.com/AWSReservedSSO_Admi..."}	null	null	3	3	0
iam.amazonaws.com	CreateRole	{"tags":[],"roleName":"eliav.levy.test.strong.user.iam.ec2.s3","description":"Strong iam, ec2 and s..."}	null	null	2	1	1
iam.amazonaws.com	CreatePolicyVersion	{"policyArn":"arn:aws:iam::111111111111:policy/eliav.levy.change.role.to.strong.role.policy","setAsDe..."}	null	null	3	3	0
iam.amazonaws.com	CreatePolicy	{"tags":[],"policyName":"eliav.levy.change.role.to.strong.role.policy","policyDocument":":{\n \"Versi..."}	null	null	3	3	0
iam.amazonaws.com	CreateInstanceProfile	{"instanceProfileName":"eliav.levy.test.strong.user.iam.ec2.s3"}	null	null	1	1	0
iam.amazonaws.com	CreateUser	{"tags":[],"userName":"eliav.levy.test_user"}	null	null	1	1	0
iam.amazonaws.com	DeletePolicy	{"policyArn":"arn:aws:iam::111111111111:policy/eliav.levy.second.poilcy.try.strong.user"}	null	null	1	1	0
iam.amazonaws.com	AttachRolePolicy	{"roleName":"eliav.levy.test.strong.user.iam.ec2.s3","policyArn":"arn:aws:iam::aws:policy/AmazonE..."}	null	null	2	2	0
iam.amazonaws.com	CreateAccessKey	{"userName":"eliav.levy.test_user"}	null	null	1	1	0
iam.amazonaws.com	AddRoleToInstanceProfile	{"roleName":"eliav.levy.test.strong.user.iam.ec2.s3","instanceProfileName":"eliav.levy.test.strong.u..."}	null	null	1	1	0
signin.amazonaws.com	SwitchRole	null	null	switchrole.error.invalidparams	10	0	0
signin.amazonaws.com	ConsoleLogin	null	null	null	6	6	0
sts.amazonaws.com	AssumeRole	null	arn:aws:iam::111111111111:role/eliav.levy.test...	null	10	0	0
sts.amazonaws.com	AssumeRole	null	eliav.levy.test.strong.user.iam.ec2.s3	null	1	0	0
iam.amazonaws.com	ListGroupsForUser	{"userName":"prod-code-reader-account"}	1	1	null	0	0
iam.amazonaws.com	GetPolicy	{"policyArn":"arn:aws:iam::111111111111:policy/eliav.levy.change.role.to.strong.rol..."}	2	2	null	0	0

RSA® Conference 2022

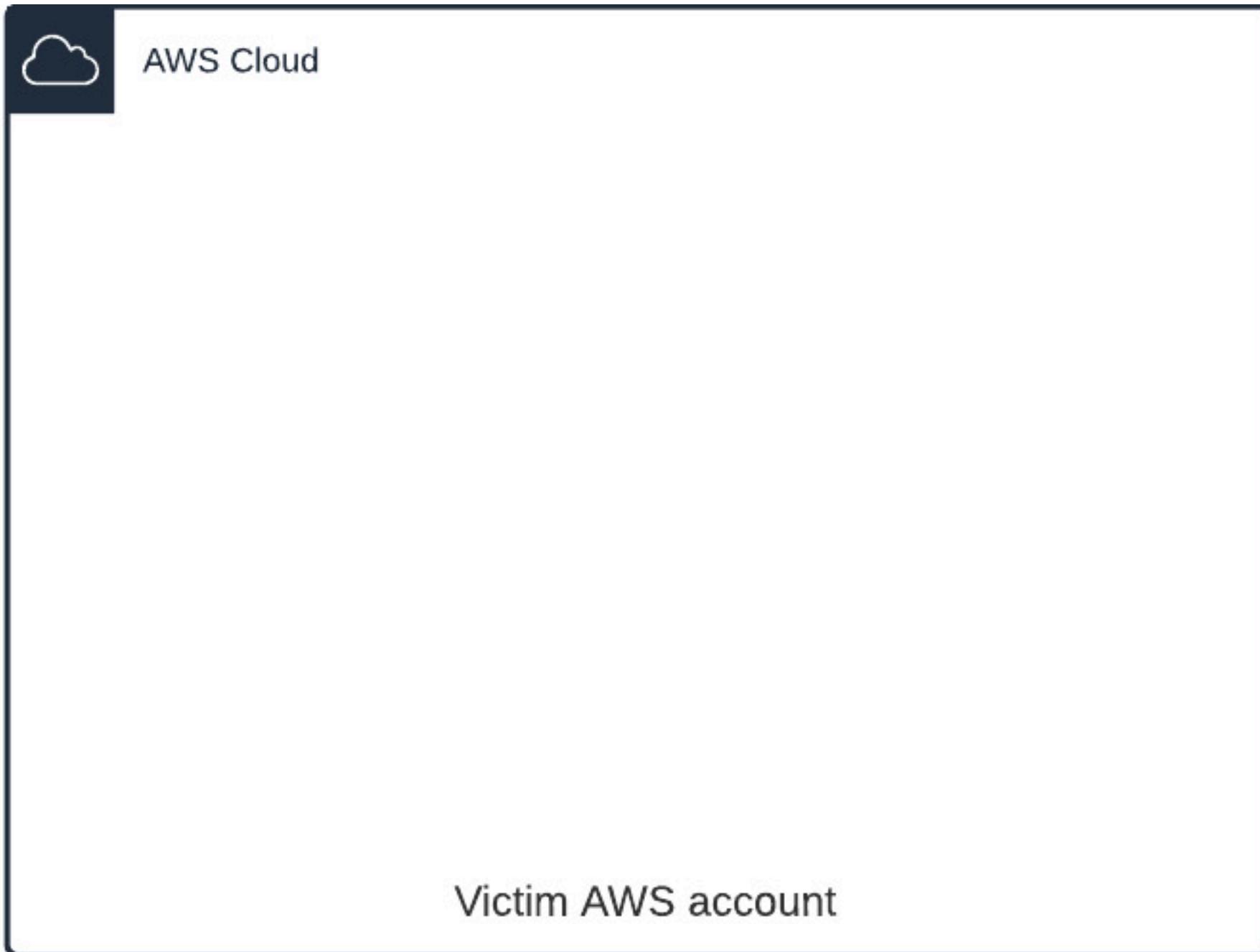
CloudTrail quirks



Fun quirk #1 - CloudTrail source IP spoofing

- Once obtaining compromised AWS API credentials
- By utilizing VPC endpoints
- Attackers can spoof the source IP that appears in the victim's CloudTrail logs.

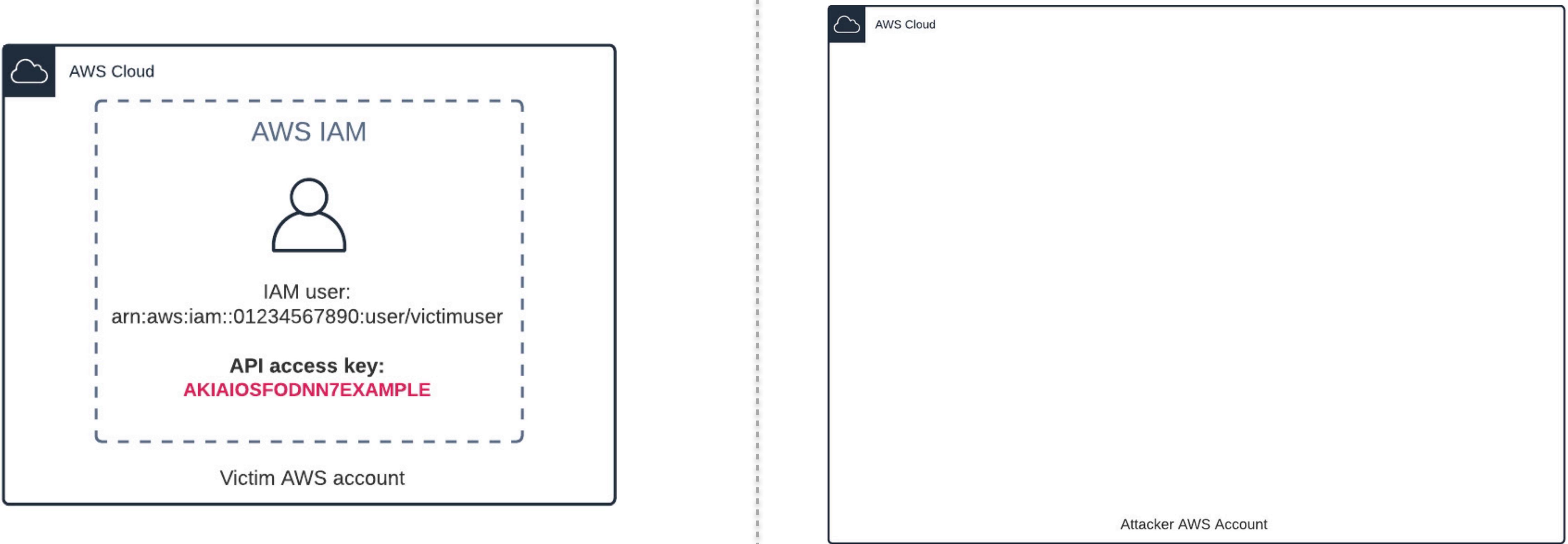
Technique Overview



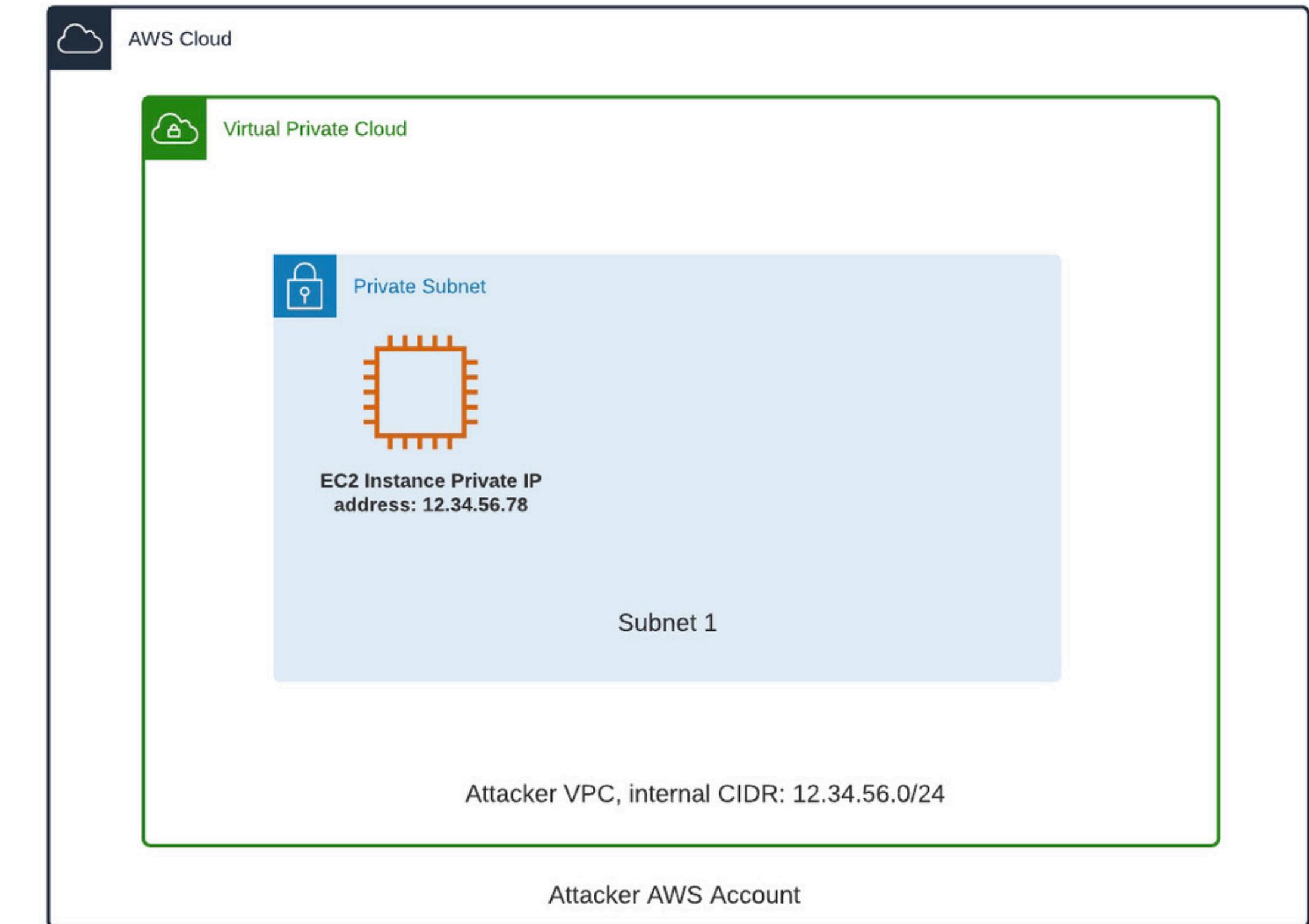
Technique Overview



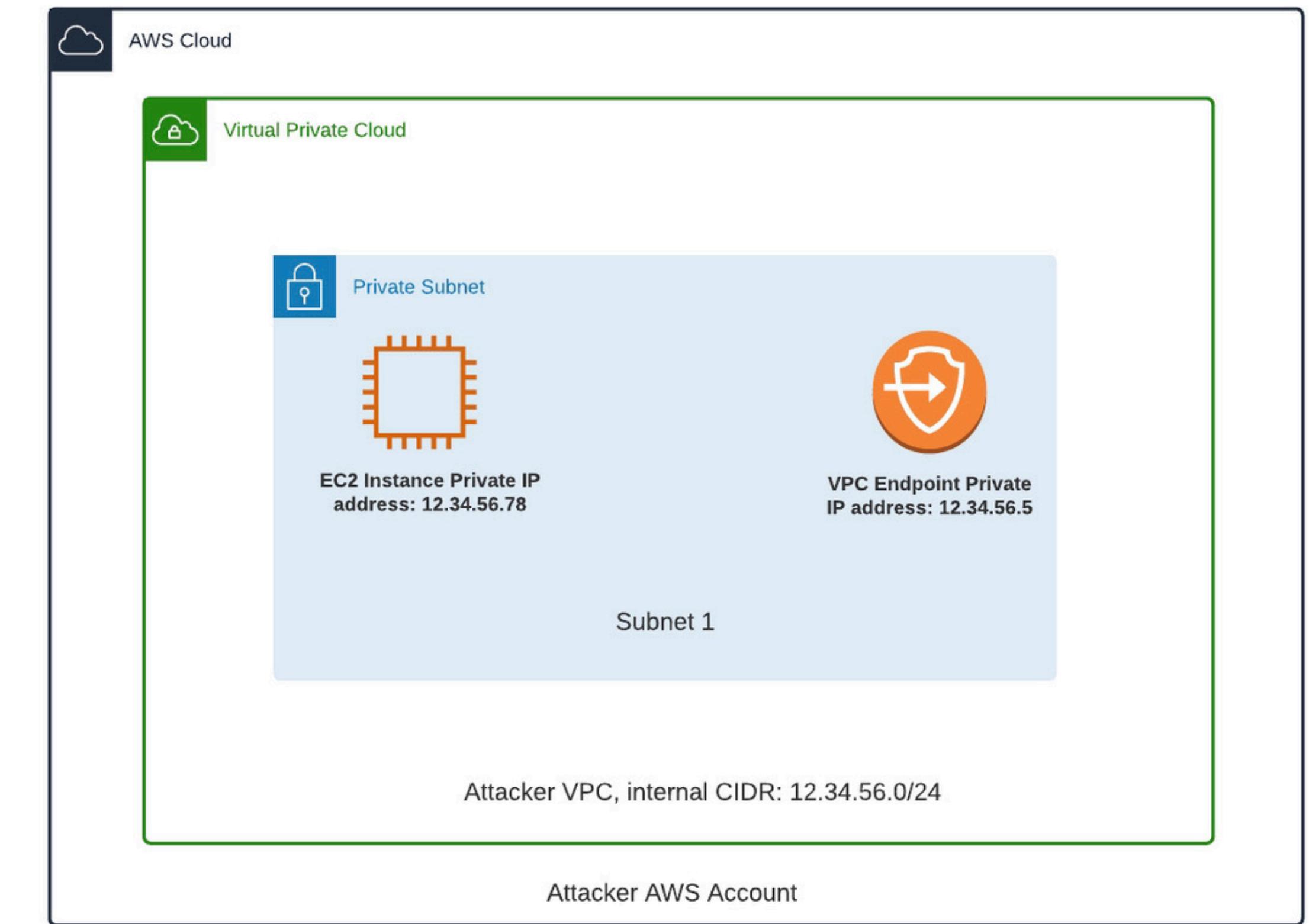
Technique Overview



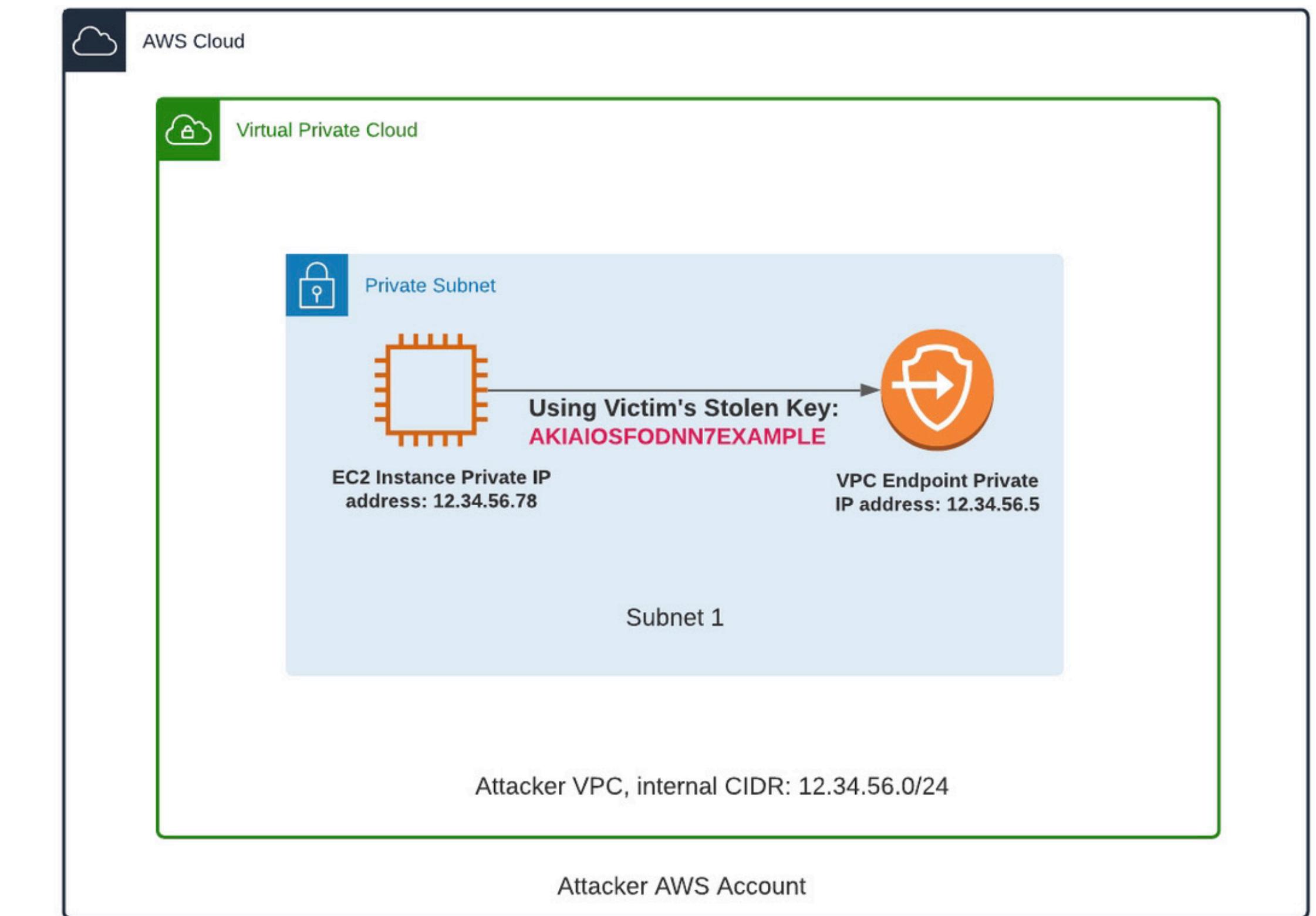
Technique Overview



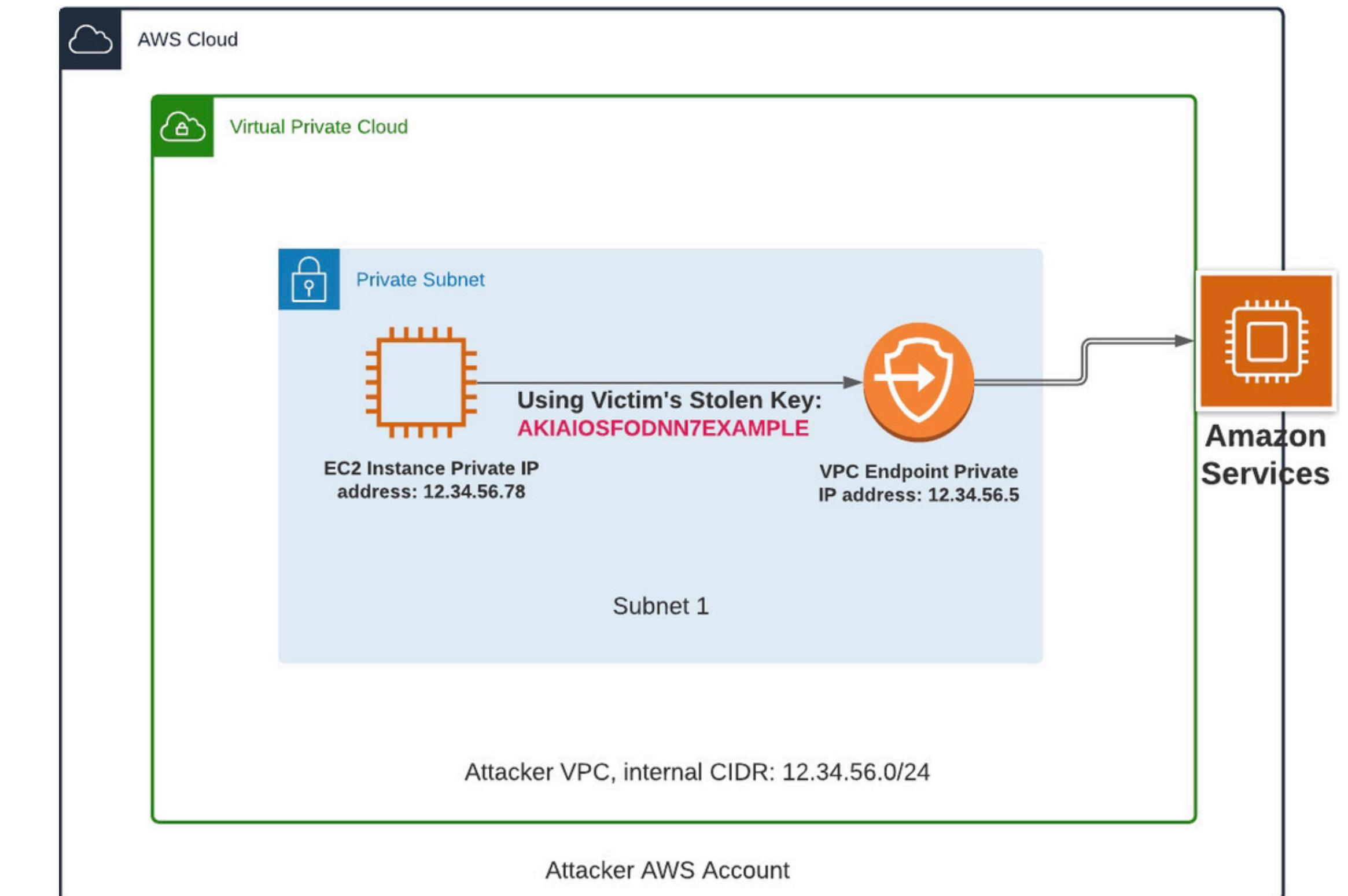
Technique Overview



Technique Overview



Technique Overview



Technique Overview



Fun quirk #2 - API access key recycling

- API access keys are supposed to be unique...

Credentials

[PDF](#)

AWS credentials for API authentication.

Contents

AccessKeyId

The access key ID that identifies the temporary security credentials.

Type: String

Length Constraints: Minimum length of 16. Maximum length of 128.

Pattern: [\\w]*

Required: Yes

Expiration

The date on which the current credentials expire.

Type: Timestamp

Understanding Unique ID Prefixes

IAM uses the following prefixes to indicate what type of resource each unique ID applies to.

Prefix	Resource Type
ABIA	AWS STS service bearer token
ACCA	Context-specific credential
AGPA	Group
AIDA	IAM user
AIPA	Amazon EC2 instance profile
AKIA	Access key
ANPA	Managed policy
ANVA	Version in a managed policy
APKA	Public key
AROA	Role
ASCA	Certificate
ASIA	Temporary (AWS STS) keys

Fun quirk #2 - API access key recycling

- API access keys are supposed to be unique...
- Or so I thought

Fun quirk #2 - API access key recycling

```
{
  "awsRegion": "us-east-1",
  "eventID": "889087b1-852f-4871-b1bd-cc7bde9dff0c",
  "eventName": "AssumeRole",
  "eventSource": "sts.amazonaws.com",
  "eventTime": "2021-08-21T19:36:15Z",
  "responseElements": {
    "assumedRoleUser": {
      "arn": "arn:aws:sts::*****_6fd7f06b",
      "assumedRoleId": "AROA*****_6fd7f06b"
    },
    "credentials": {
      "accessKeyId": "ASIA*****TEASS",
      "expiration": "Aug 21, 2021 8:36:15 PM",
      "sessionToken":
        "*****9UOs5x3qs=="
    }
  }
}
```

```
"awsRegion": "us-east-1",
"eventID": "368084bd-cdab-4be9-95b2-d304c57dea25",
"eventName": "AssumeRole",
"eventSource": "sts.amazonaws.com",
"eventTime": "2021-08-21T20:01:51Z",
"responseElements": {
  "assumedRoleUser": {
    "arn": "arn:aws:sts::*****_e6880947",
    "assumedRoleId": "AROA*****_e6880947"
  },
  "credentials": {
    "accessKeyId": "ASIA*****TEASS",
    "expiration": "Aug 21, 2021 9:01:51 PM",
    "sessionToken":
      "*****msA4Zq1iM="
  }
}
```

Fun quirk #2 - API access key recycling

```
{  
  "awsRegion": "us-east-1",  
  "eventID": "889087b1-852f-4871-b1bd-cc7bde9dff0c",  
  "eventName": "AssumeRole",  
  "eventSource": "sts.amazonaws.com",  
  "eventTime": "2021-08-21T19:36:15Z",  
  "responseElements": {  
    "assumedRoleUser": {  
      "arn": "arn:aws:sts::*****_6fd7f06b",  
      "assumedRoleId": "AROA*****_6fd7f06b"  
    },  
    "credentials": {  
      "accessKeyId": "ASIA*****TEASS",  
      "expiration": "Aug 21, 2021 8:36:15 PM",  
      "sessionToken":  
        "*****9UOs5x3qs=="  
    }  
  }  
}
```

```
"awsRegion": "us-east-1",  
"eventID": "368084bd-cdab-4be9-95b2-d304c57dea25",  
"eventName": "AssumeRole",  
"eventSource": "sts.amazonaws.com",  
"eventTime": "2021-08-21T20:01:51Z",  
"responseElements": {  
  "assumedRoleUser": {  
    "arn": "arn:aws:sts::*****_e6880947",  
    "assumedRoleId": "AROA*****_e6880947"  
  },  
  "credentials": {  
    "accessKeyId": "ASIA*****TEASS",  
    "expiration": "Aug 21, 2021 9:01:51 PM",  
    "sessionToken":  
      "*****msA4Zq1iM=="  
  }  
}
```

Fun quirk #2 - API access key recycling

```
{
  "awsRegion": "us-east-1",
  "eventID": "889087b1-852f-4871-b1bd-cc7bde9dff0c",
  "eventName": "AssumeRole",
  "eventSource": "sts.amazonaws.com",
  "eventTime": "2021-08-21T19:36:15Z",
  "responseElements": {
    "assumedRoleUser": {
      "arn": "arn:aws:sts::*****_6fd7f06b",
      "assumedRoleId": "AROA*****_6fd7f06b"
    },
    "credentials": {
      "accessKeyId": "ASIA*****TEASS",
      "expiration": "Aug 21, 2021 8:36:15 PM",
      "sessionToken":
        "*****9UOs5x3qs=="
    }
  }
}
```

```
"awsRegion": "us-east-1",
"eventID": "368084bd-cdab-4be9-95b2-d304c57dea25",
"eventName": "AssumeRole",
"eventSource": "sts.amazonaws.com",
"eventTime": "2021-08-21T20:01:51Z",
"responseElements": {
  "assumedRoleUser": {
    "arn": "arn:aws:sts::*****_e6880947",
    "assumedRoleId": "AROA*****_e6880947"
  },
  "credentials": {
    "accessKeyId": "ASIA*****TEASS",
    "expiration": "Aug 21, 2021 9:01:51 PM",
    "sessionToken":
      "*****msA4Zq1iM="
  }
}
```

Fun quirk #2 - API access key recycling

```
{
  "awsRegion": "us-east-1",
  "eventID": "889087b1-852f-4871-b1bd-cc7bde9dff0c",
  "eventName": "AssumeRole",
  "eventSource": "sts.amazonaws.com",
  "eventTime": "2021-08-21T19:36:15Z",
  "responseElements": {
    "assumedRoleUser": {
      "arn": "arn:aws:sts::*****_6fd7f06b",
      "assumedRoleId": "AROA*****_6fd7f06b"
    },
    "credentials": {
      "accessKeyId": "ASIA*****TEASS",
      "expiration": "Aug 21, 2021 8:36:15 PM",
      "sessionToken":
        "*****9UOs5x3qs=="
    }
  }
}
```

```
"awsRegion": "us-east-1",
"eventID": "368084bd-cdab-4be9-95b2-d304c57dea25",
"eventName": "AssumeRole",
"eventSource": "sts.amazonaws.com",
"eventTime": "2021-08-21T20:01:51Z",
"responseElements": {
  "assumedRoleUser": {
    "arn": "arn:aws:sts::*****_e6880947",
    "assumedRoleId": "AROA*****_e6880947"
  },
  "credentials": {
    "accessKeyId": "ASIA*****TEASS",
    "expiration": "Aug 21, 2021 9:01:51 PM",
    "sessionToken":
      "*****msA4Zq1iM="
  }
}
```

Fun quirk #2 - API access key recycling

```
{  
  "awsRegion": "us-east-1",  
  "eventID": "889087b1-852f-4871-b1bd-cc7bde9dff0c",  
  "eventName": "AssumeRole",  
  "eventSource": "sts.amazonaws.com",  
  "eventTime": "2021-08-21T19:36:15Z",  
  "responseElements": {  
    "assumedRoleUser": {  
      "arn": "arn:aws:sts::*****_6fd7f06b",  
      "assumedRoleId": "AROA*****_6fd7f06b"  
    },  
    "credentials": {  
      "accessKeyId": "ASIA*****TEASS",  
      "expiration": "Aug 21, 2021 8:36:15 PM",  
      "sessionToken":  
        "*****9UOs5x3qs=="  
    }  
  }  
}
```

```
"awsRegion": "us-east-1",  
"eventID": "368084bd-cdab-4be9-95b2-d304c57dea25",  
"eventName": "AssumeRole",  
"eventSource": "sts.amazonaws.com",  
"eventTime": "2021-08-21T20:01:51Z",  
"responseElements": {  
  "assumedRoleUser": {  
    "arn": "arn:aws:sts::*****_e6880947",  
    "assumedRoleId": "AROA*****_e6880947"  
  },  
  "credentials": {  
    "accessKeyId": "ASIA*****TEASS",  
    "expiration": "Aug 21, 2021 9:01:51 PM",  
    "sessionToken":  
      "*****msA4Zq1iM=="  
  }  
}
```

Fun quirk #2 - API access key recycling

```
{  
  "awsRegion": "us-east-1",  
  "eventID": "889087b1-852f-4871-b1bd-cc7bde9dff0c",  
  "eventName": "AssumeRole",  
  "eventSource": "sts.amazonaws.com",  
  "eventTime": "2021-08-21T19:36:15Z",  
  "responseElements": {  
    "assumedRoleUser": {  
      "arn": "arn:aws:sts::*****_6fd7f06b",  
      "assumedRoleId": "AROA*****_6fd7f06b"  
    },  
    "credentials": {  
      "accessKeyId": "ASIA*****TEASS",  
      "expiration": "Aug 21, 2021 8:36:15 PM",  
      "sessionToken":  
        "*****9UOs5x3qs=="  
    }  
  }  
}
```

```
"awsRegion": "us-east-1",  
"eventID": "368084bd-cdab-4be9-95b2-d304c57dea25",  
"eventName": "AssumeRole",  
"eventSource": "sts.amazonaws.com",  
"eventTime": "2021-08-21T20:01:51Z",  
"responseElements": {  
  "assumedRoleUser": {  
    "arn": "arn:aws:sts::*****_e6880947",  
    "assumedRoleId": "AROA*****_e6880947"  
  },  
  "credentials": {  
    "accessKeyId": "ASIA*****TEASS",  
    "expiration": "Aug 21, 2021 9:01:51 PM",  
    "sessionToken":  
      "*****msA4Zq1iM=="  
  }  
}
```

Fun quirk #2 - API access key recycling

- API access keys are supposed to be unique...
- Or so I thought
- Turns out they are recycled across identities

Hello Eliav,

Greetings from AWS Premium Support !

Thank you for contacting us. I'll be assisting you with this case today.

From the case notes, I understand that while investigating Cloudtrail logs for your customers, you saw that in some cases the access key IDs assigned after AssumeRole call were same for some users and these access key IDs were valid for overlapping time period. Please correct me if I have misunderstood.

This is undesired behavior and access key IDs are unique identifiers for an IAM entity. As you correctly understood, two entities having the same access key ID can cause logging and investigating issues difficult as Cloudtrail will have all identical fields for both the entities (if role assumed and the RoleSessionName used are also the same).
Please note, session token and secret access key are not logged in clouptrail for security purposes.

Fun quirk #2 - API access key recycling

- API access keys are supposed to be unique...
- Or so I thought
- Turns out they are recycled across identities

Hello Eliav,

Thank you for writing back to us.

First of all, I would like to appreciate the amount of research you have done on Cloudtrail logs and access keys - it is very impressive!

I checked all the event IDs you provided and can confirm this behavior (Same access key IDs with both overlapping and non-overlapping time periods).

With this information I have escalated this issue to our IAM service team. They should be able to clarify this behavior and answer if this is desired behavior or not for both scenarios (with and without overlapping time periods)

Please note, session token and secret access key are not logged in clouptrail for security purposes.

Fun quirk #2 - API access key recycling

- API access keys are supposed to be unique...
- Or so I thought
- Turns out they are recycled across identities

Hello Eliav,

Greetings from AWS Premium Support !
Thank you for writing back to us.

Thank you for contacting us. I'll be assisting you with this case today.

From the above email, I got the following update from the service team:

I got the following update from the service team:

Is this desired behavior?

>>> This is expected behavior. Due to the sheer amount of AssumeRole calls made throughout AWS the ASIA**** access key identifier will eventually be generated again(even within same time frame). Remember that in order to make an API call with temporary credentials you will have to supply an access key id, a secret access key, and a session token. Those three combinations are permanently unique. I understand that you cannot search CloudTrail via secret access key or session token, and that if an access key is reused it could cause some confusion in some cases.

Fun quirk #2 - API access key recycling

Understanding Unique ID Prefixes

IAM uses the following prefixes to indicate what type of resource each unique ID applies to.

Prefix	Resource Type
ABIA	AWS STS service bearer token
ACCA	Context-specific credential
AGPA	Group
AIDA	IAM user
AIPA	Amazon EC2 instance profile
AKIA	Access key
ANPA	Managed policy
ANVA	Version in a managed policy
APKA	Public key
AROA	Role
ASCA	Certificate
ASIA	Temporary (AWS STS) keys

Fun quirk #2 - API access key recycling

Understanding Unique ID Prefixes

IAM uses the following prefixes to indicate what type of resource each unique ID applies to.

Prefix	Resource Type
ABIA	AWS STS service bearer token
ACCA	Context-specific credential
AGPA	Group
AIDA	IAM user
AIPA	Amazon EC2 instance profile
AKIA	Access key
ANPA	Managed policy
ANVA	Version in a managed policy
APKA	Public key
AROA	Role
ASCA	Certificate
ASIA	Temporary (AWS STS) keys

Understanding unique ID prefixes

IAM uses the following prefixes to indicate what type of resource each unique ID applies to.

Prefix	Resource type
ABIA	AWS STS service bearer token
ACCA	Context-specific credential
AGPA	User group
AIDA	IAM user
AIPA	Amazon EC2 instance profile
AKIA	Access key
ANPA	Managed policy
ANVA	Version in a managed policy
APKA	Public key
AROA	Role
ASCA	Certificate
ASIA	Temporary (AWS STS) access key IDs use this prefix, but are unique only in combination with the secret access key and the session token.

Fun quirk #2 - API access key recycling

- API access keys are supposed to be unique...
- Or so I thought
- Turns out they are recycled across identities
- **Not a security risk**
 - Happens only rarely
 - Only recycled in-account
 - No “cross-permissions” possible

What AWS can improve

- More comprehensive logging
- Better documentation

AWS Detective

- Amazon's own investigation service utilizing
 - VPC flow logs
 - CloudTrail
 - GuardDuty
- Helps contextualize alerts and spot anomalies
- Limited chain following capabilities

Apply What You Have Learned Today

- Store CloudTrail in queryable manner
- Understand the concept of AWS Sessions
- Able to follow the activity of a single user across cloudtrail logs
- Use the knowledge provided to create consumable AWS Session summary query
- Avoid common pitfalls and quirks when investigating AWS incidents