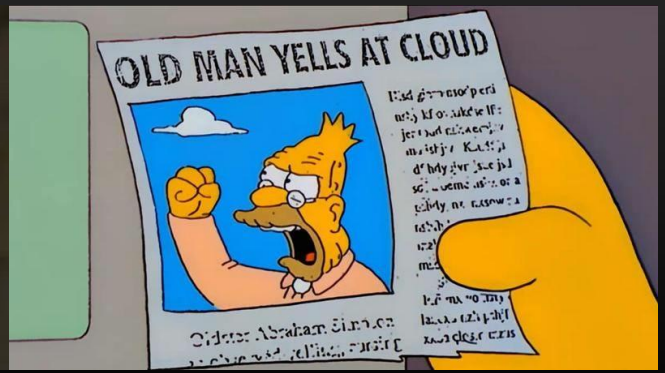
Incident Response in AWS

Don't panic, take a deep breath, you've got this.

Chris Farris
PrimeHarbor Technologies

Who Am I?

- Built the cloud security programs for some media companies
- Founder: fwd:cloudsec conference
- Rants a lot on Twitter
- And I'm about to start a new job at a Cloud Security company next month



What this talk will cover…

Cloud Plane Detection & Response

- Cloud Security in 4 Slides
- Getting your cloud telemetry in order
- Finding your incident
- How to safely disable access to the cloud plane
- Figuring out what was done and what was taken

(NIST 800-61 Framework)

Cloud Security in 4 Slides

Identity is the new perimeter

Cloud Plane vs Network Plane

or

You need to defend three dimensionally

or

"*Cute network controls you have there if would be a shame if someone just routed around them*"

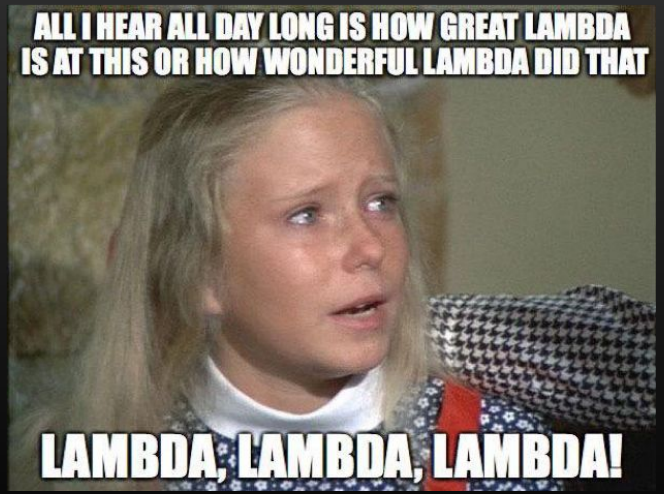
Creds, creds, everywhere creds….

… Leaking out of everything, Breakin' my mind

- Everything in AWS requires Auth
- 90% of the time that's an Access Key, Secret, and sometimes session token
- Services that have been given permissions can request these creds
- Which means if your code has flaws, they can be used to exfil creds

Serverless

- New application architectures that move more responsibility to the CSP
- Reduced Surface area
- No access to the low level telemetry sources
- Lots of settings to misconfigure
- Examples:
  - Lambda
  - Fargate
  - RDS
  - S3

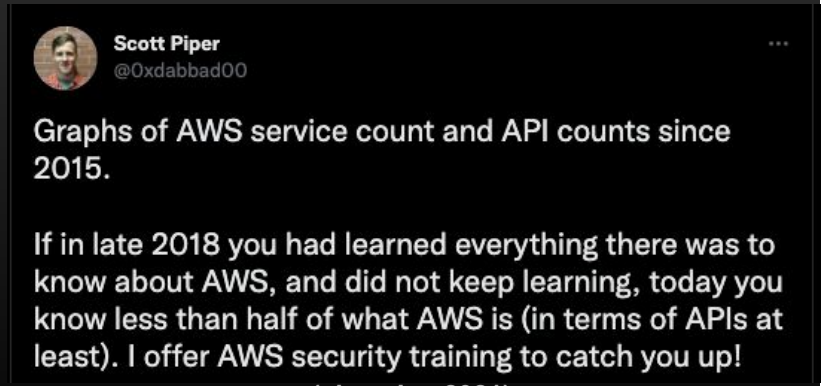


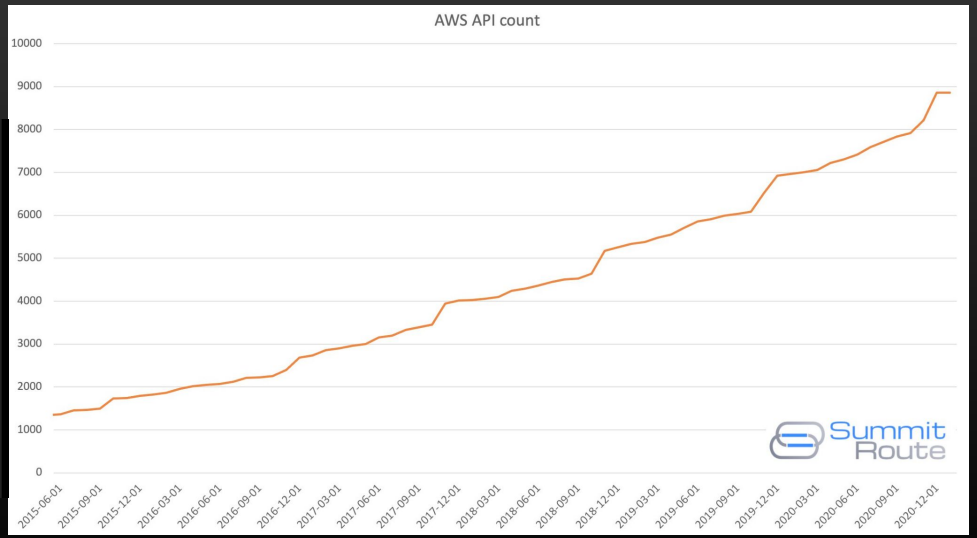
Cloud Hygiene

Everyone's maturity will vary here…

Public Buckets are a thing because AWS lets them be a thing

There are just too many services





AWS does a bad job of making it hard to do stupid things

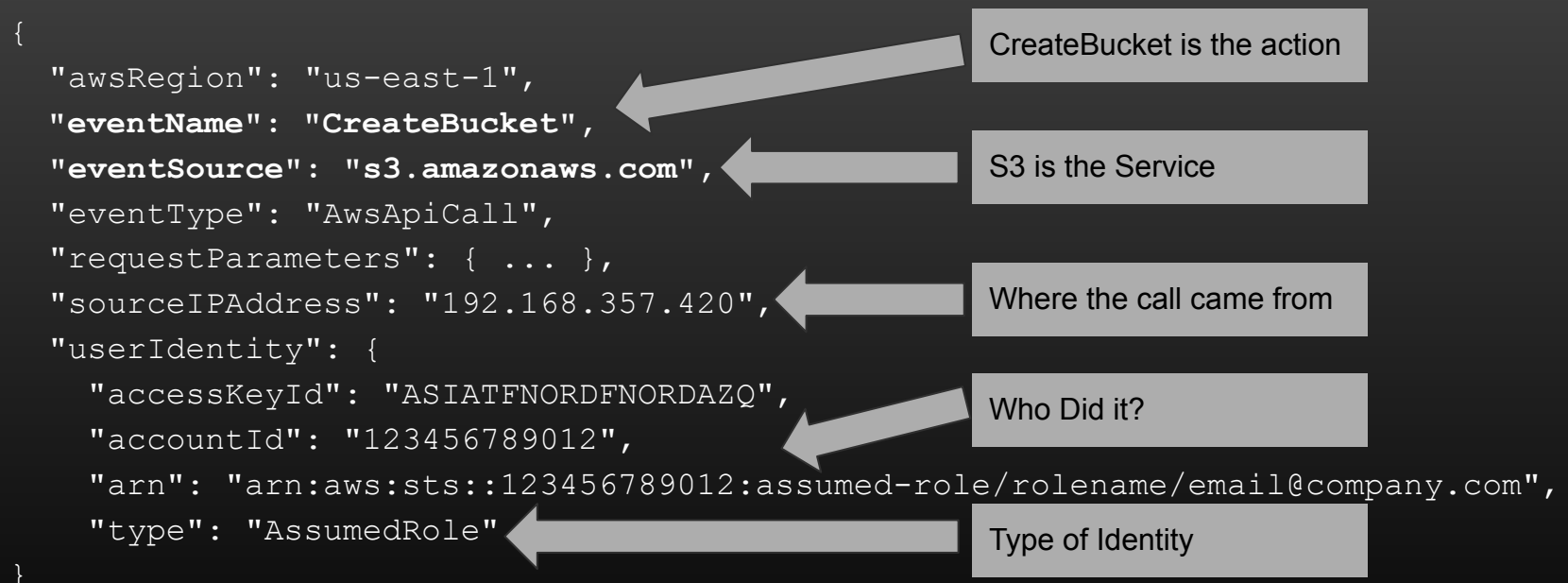
Preparation

CloudTrail

The most important AWS service to have for incident response

- Records every authenticated API call to your AWS account
- Includes the principal who performed the API call
- Source IP address where the API call came from
- When the API call was made
- What Action was performed against which Service
- What Resources were impacted by the call

Anatomy of a CloudTrail Event



A digression into IAM

AWS Identity and Access Management is the most complex AWS service

- Every API Call must be authorized
- AWS operates on implicit deny
- Actions are broken down by service and action
  - ec2:RunInstances
  - s3:Get Object
- CloudTrail eventSource and eventName map to these actions

CloudTrail DataEvents

- Default CloudTrail only does "Management" Events
- Data Events, like S3 read & write are not logged

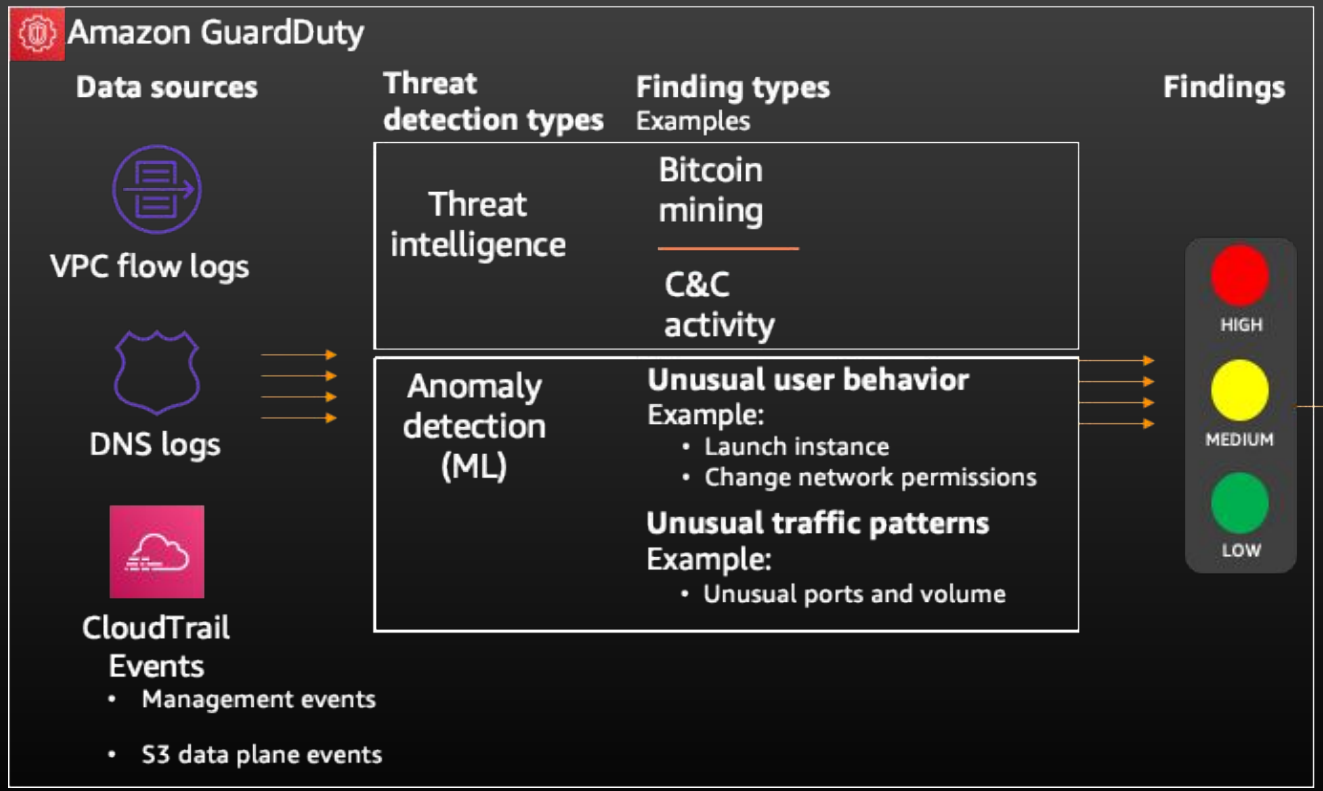
Data Events can definitively prove or disprove data has been accessed

Other sorts of Data Events:

- Lambda Function Invocations
- DynamoDB queries
- Glue, BlockChain, Outposts

Cost: $0.10 per 100,000 events

Amazon GuardDuty



Security Account

- Hosts the managed AWS Security Services
  - a.k.a Delegated Admin
- Stores logs
- Trusted for organization wide Audit & IR

Security workloads should be hosted in a different account

SEIM/Search/SOAR

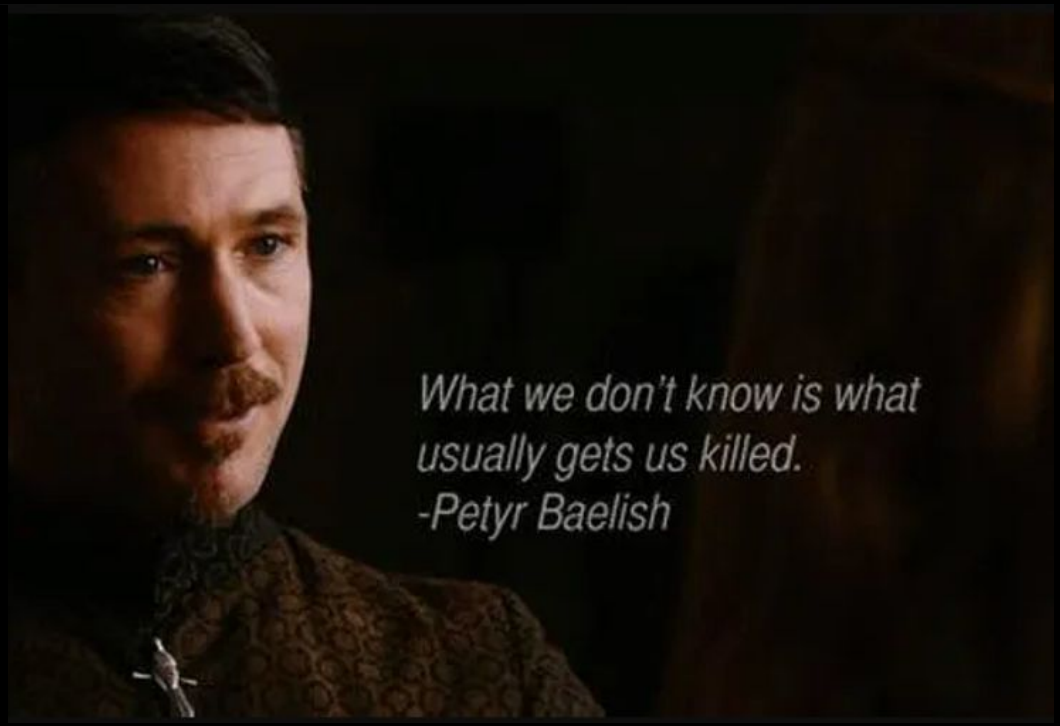
- This depends on your organization's maturity (and budget!)
- AWS has some rudimentary capabilities for this.
- Cross Referencing with other enterprise telemetry is critical!

At the very minimum send GuardDuty to Slack and action on those alerts!

Inventory

What is SANS/CIS Critical Control One and Two?

- Know your Accounts
- Know your Account Owners
- Know what is in those accounts



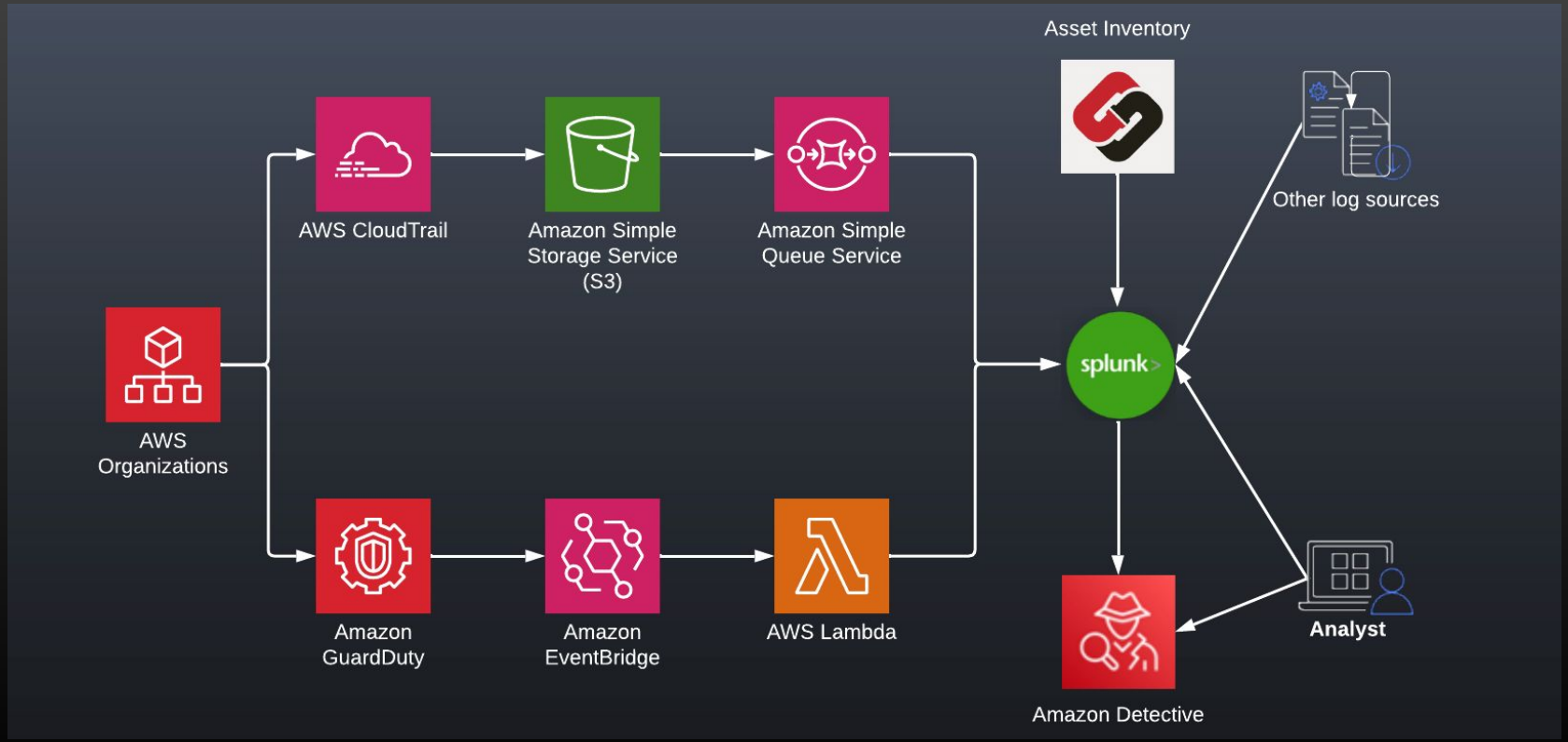
Decoration

- Context is critical!
- Is the resource a production or development resource?
- Does the AWS Account contain PII or public marketing fluff?
- Who is the application or resource owner?
- Tagging information can help prioritize the investigation of events

There are a number of tools that can pull data from your cloud accounts to do decoration.

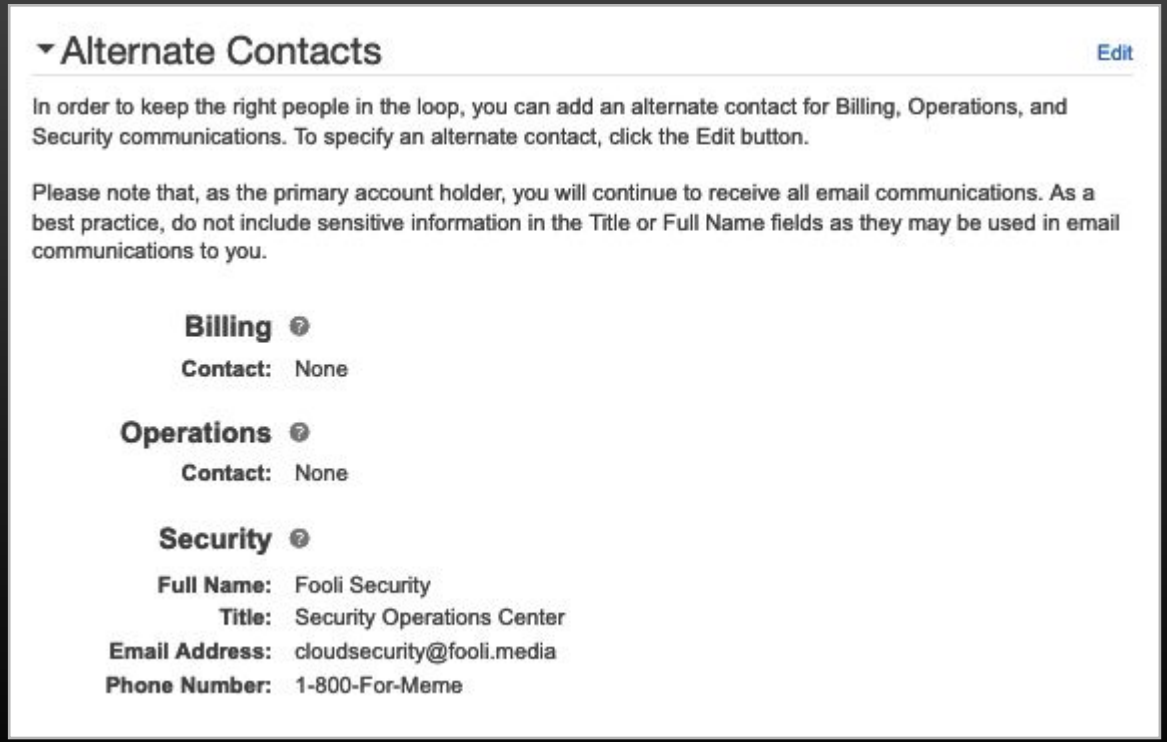
https://github.com/jchrisfarris/steampipe_splunk_tables

Glue it all together



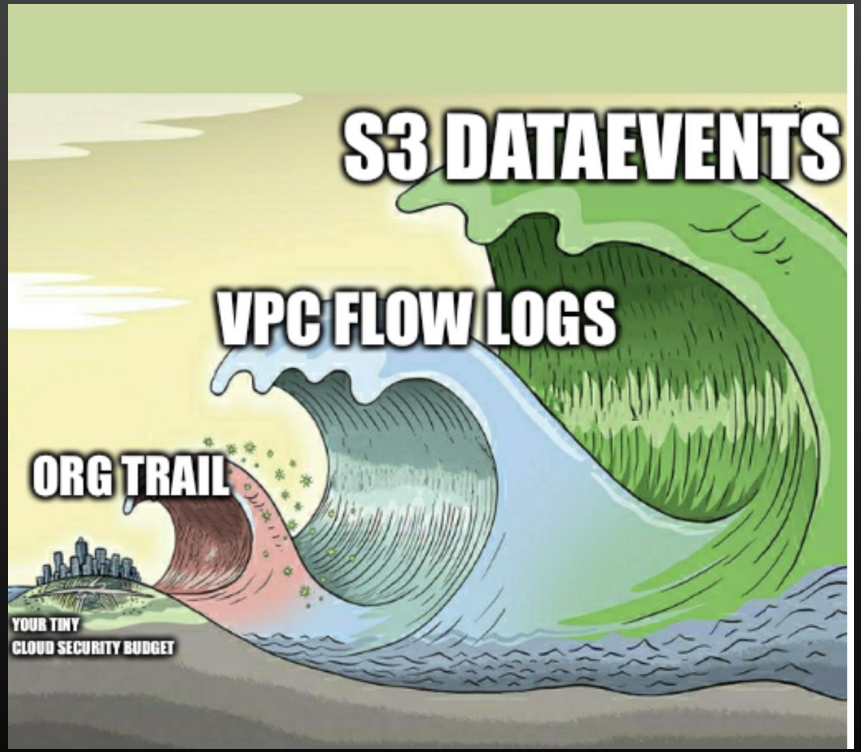
Root Email & Security Contacts

- Monitor Root User
- Set Security Contacts
- You can now do this org wide!



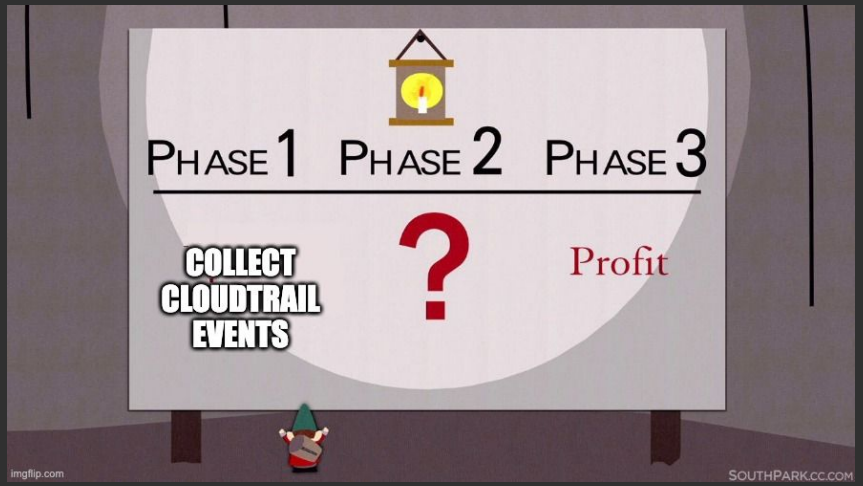
Nice to Have

- VPC Flow Logs
- EC2 Forensics Capability
- Amazon Detective
- Amazon Macie



Preparation

Building a Detection Catalog



Ok, I have 3 billion blobs of json in S3, now what?

eventNames to look for

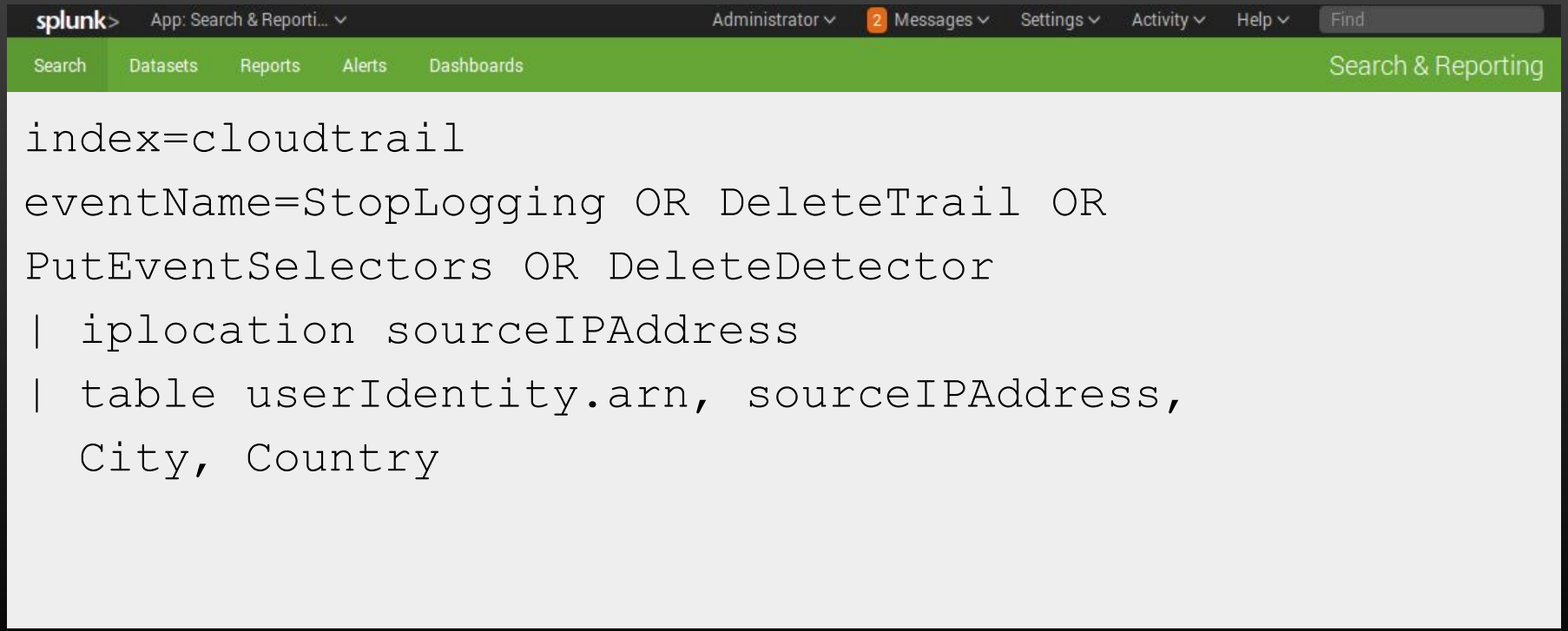
Higher Fidelity Events:

- CreateTrustAnchor
- CreateUser
- CreateLoginProfile
- UpdateLoginProfile
- CreateAccessKey
- AttachUserPolicy
- DeleteTrail
- PutEventSelectors
- StopLogging
- LeaveOrganization
- DeleteFlowLogs
- DeleteVpc
- GetPasswordData
- GetSecretValue
- ModifyImageAttribute

Common but Significant:

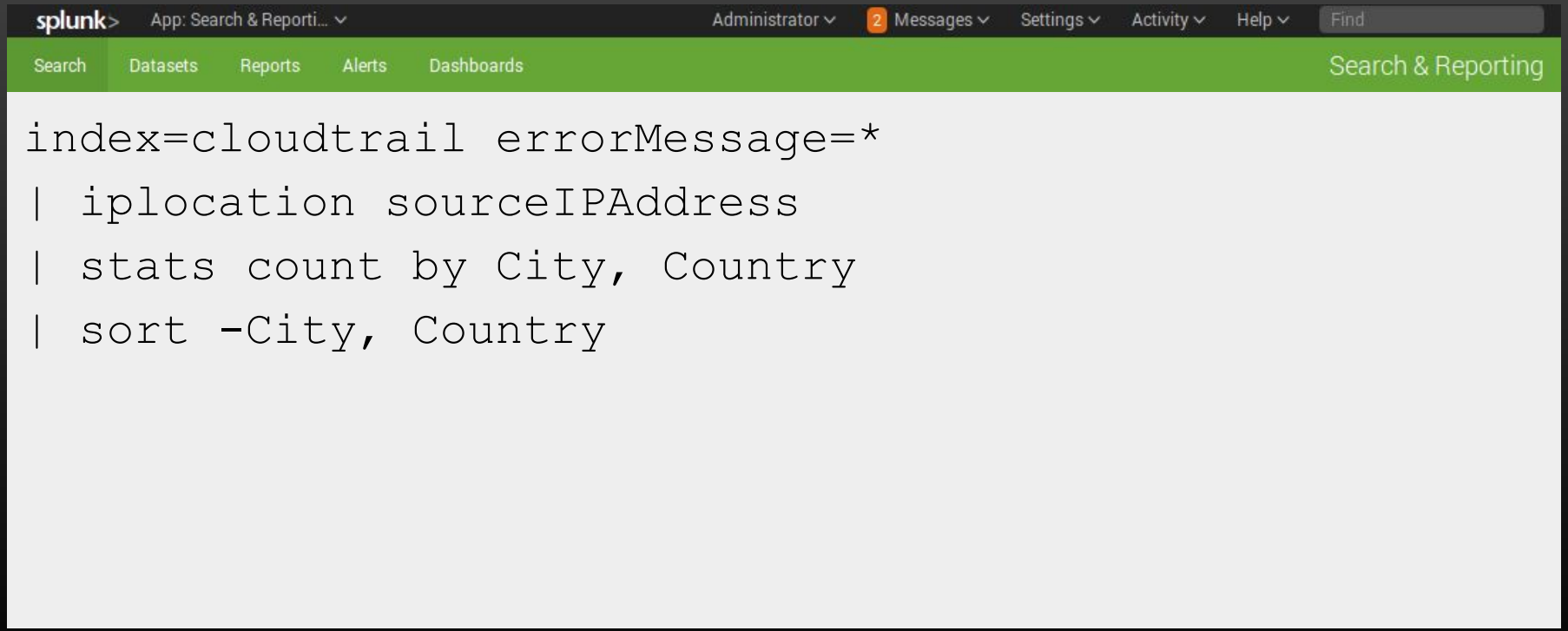
- ConsoleLogin
- GetFederationToken
- StartSession
- GetAuthorizationToken
- CreateKeyPair
- CreateRole
- PutUserPolicy
- PutGroupPolicy
- CreateGroup
- AttachRolePolicy
- PutRolePolicy
- CreatePolicyVersion
- UpdateAssumeRolePolicy
- UpdateFunctionConfiguration
- ListSecrets
- ModifySnapshotAttribute
- PutBucketPolicy
- PutBucketAcl

Evasion Detection

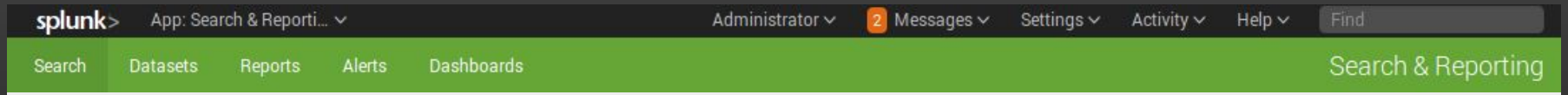



```
index=cloudtrail
eventName=StopLogging OR DeleteTrail OR
PutEventSelectors OR DeleteDetector
| iplocation sourceIPAddress
| table userIdentity.arn, sourceIPAddress,
  City, Country
```

Where are Errors Coming From?

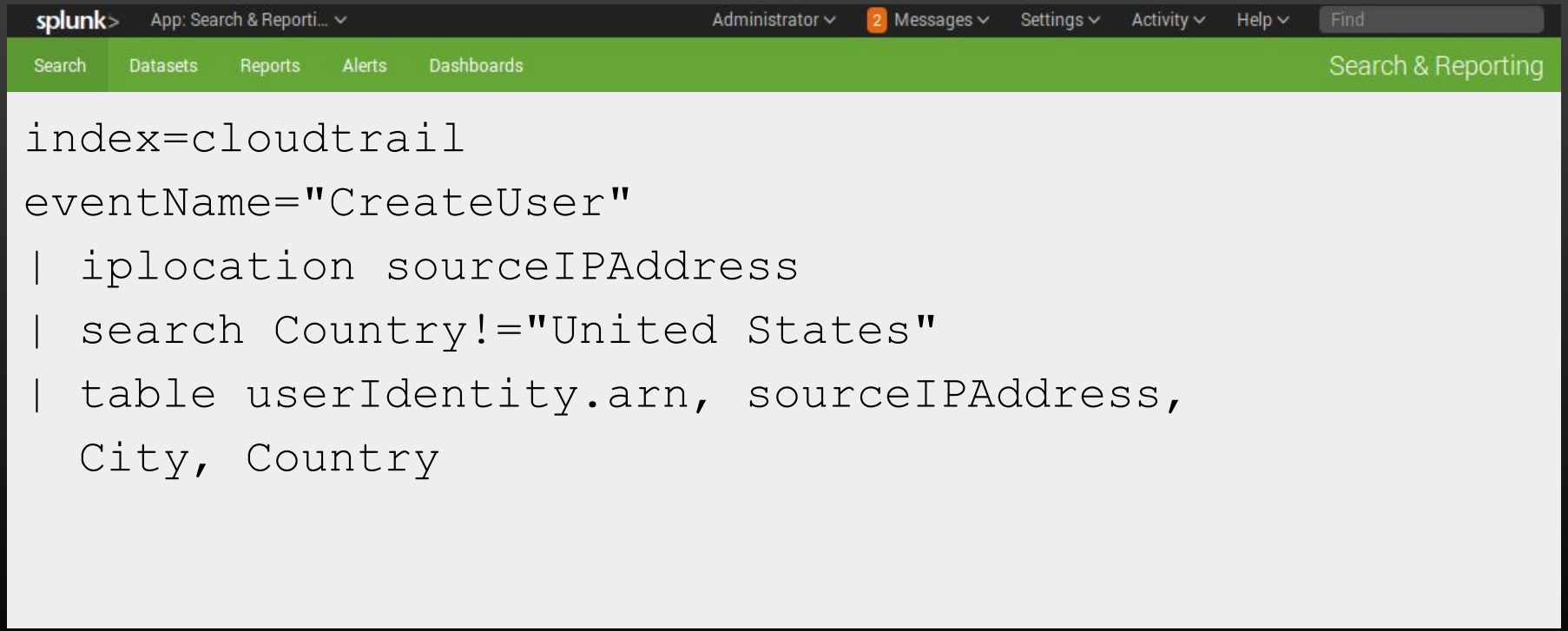


Weird Credential Usage




```
index=cloudtrail
eventName=GetCallerIdentity OR ListBuckets OR
DescribeInstances
| iplocation sourceIPAddress
| table userIdentity.arn, sourceIPAddress,
  City, Country
| sort -City, Country
```

Persistence Detection



Identification

… or…

How your day will be ruined

How you'll get notified of an incident

- AWS Trust & Safety
- Your AWS Bill / Billing Alerts
- Service Impacts
- Your Detection Catalog
- GuardDuty
- Twitter

Dear AWS customer,

Your AWS Account is compromised! Please review the following notice and take immediate action to secure your account. We have also opened an outbound Support Case if you have any additional questions or concerns regarding this notice.

Your security is important to us. We have become aware that the AWS Access Key AKIAXJ HI (belonging to IAM user "[redacted].com") along with the corresponding Secret Key is publicly available online at

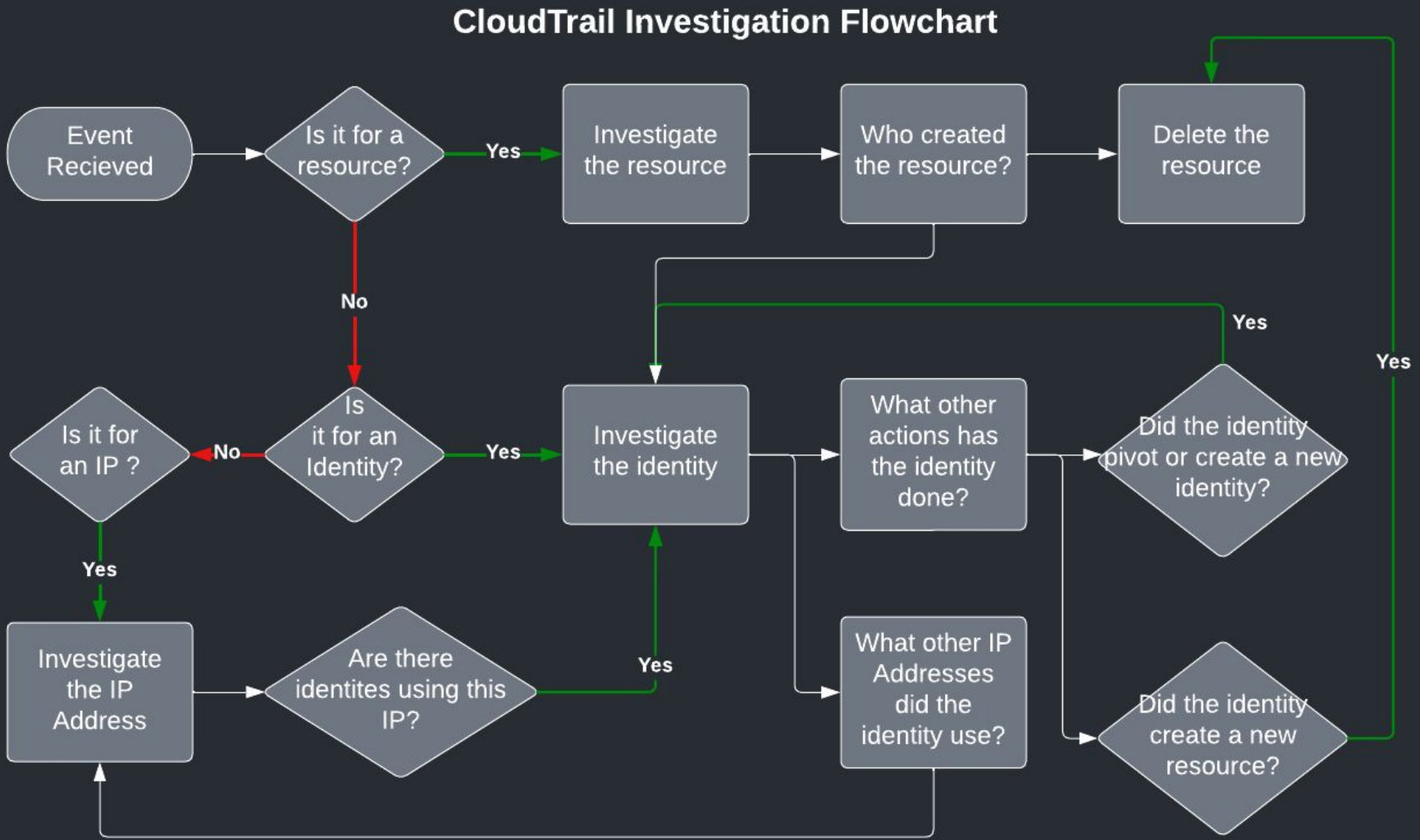

CloudTrail Investigation Flowchart
Event Recieved
Is it for a resource?
Yes
Investigate the resource
Who created the resource?
Delete the resource
No
Yes
Yes
Is it for an IP ?
No
Is it for an Identity?
Yes
Investigate the identity
What other actions has the identity done?
Did the identity pivot or create a new identity?
Yes
Investigate the IP Address
Are there identites using this IP?
Yes
What other IP Addresses did the identity use?
Did the identity create a new resource?
Yes

Figure out what the attacker did

The various event names tie into the various TTPs

s3:ListBuckets is enumeration

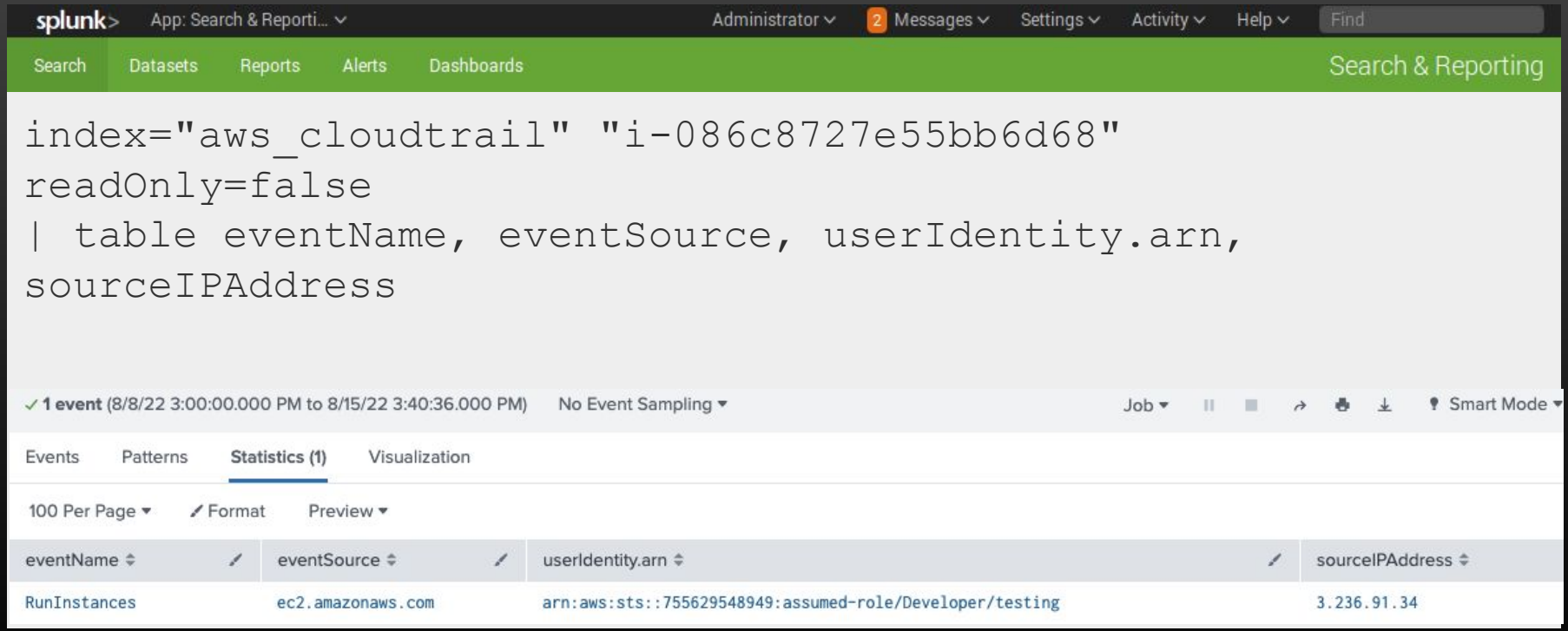
s3:GetObject is exfiltration

cloudtrail:StopLogging is evasion

iam:CreateUser is persistence

https://github.com/zmallen/cloudtrail2sightings

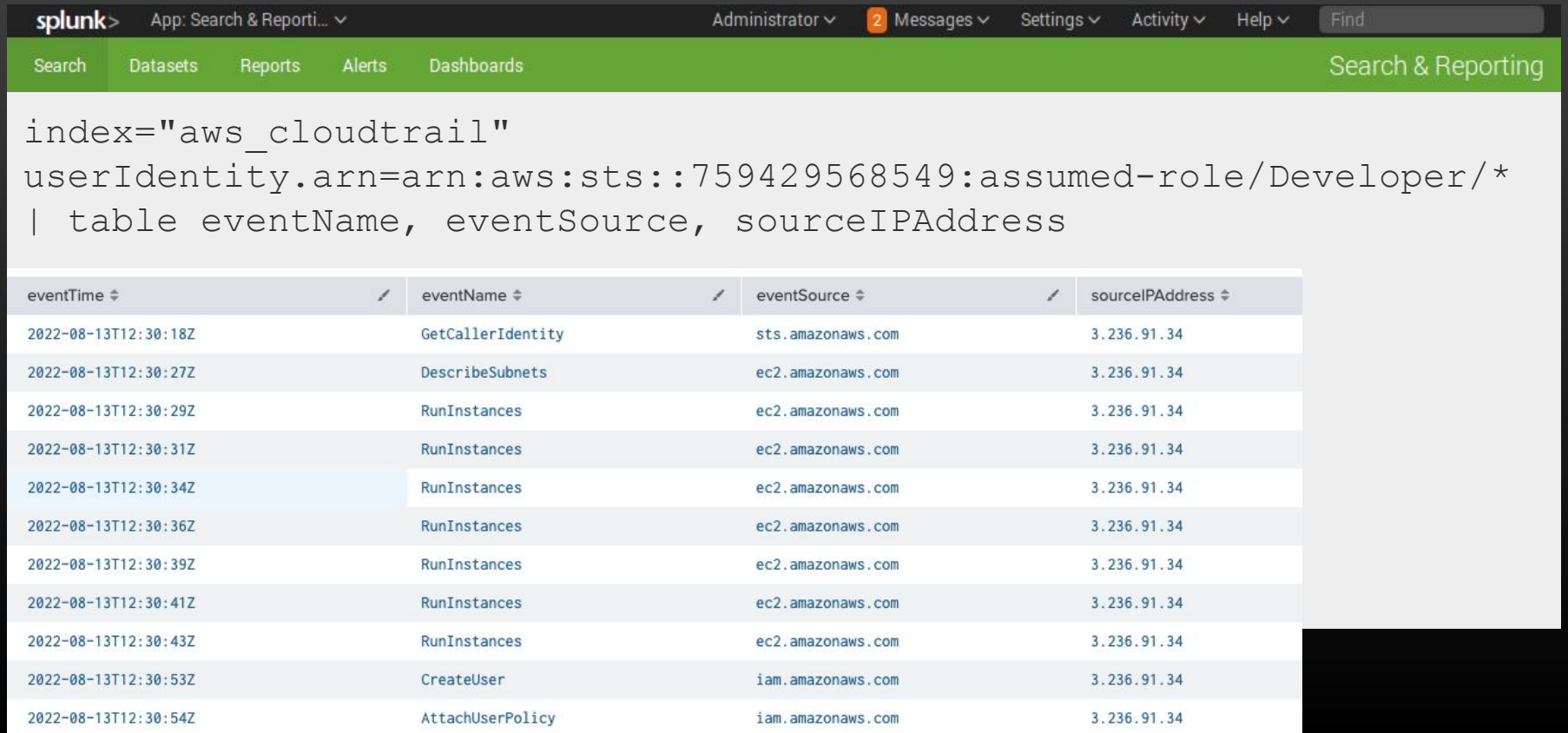
Sample CloudTrail Query



Investigate the Instance


```
{ [-]
   awsRegion: us-east-1
   eventCategory: Management
   eventID: 7b4856eb-8c0c-4e66-969c
   eventName: RunInstances
   eventSource: ec2.amazonaws.com
   eventTime: 2022-08-13T12:30:36Z
   eventType: AwsApiCall
   eventVersion: 1.08
   managementEvent: true
   readOnly: false
   recipientAccountId: 755629548949
   requestParameters: { [-]
     blockDeviceMapping: { [+] }
     disableApiStop: false
     disableApiTermination: false
     instanceType: t2.micro
     instancesSet: { [-]
       items: [ [-]
         { imageId: ami-0a25ed80a0ff1d536
         } ] }
     monitoring: { [+] }
     subnetId: subnet-0dc76374a87f3d69c
   }
   responseElements: { [+] }
   sourceIPAddress: 3.236.91.34
   tlsDetails: { [+] }
   userAgent:
aws-cli/2.7.20 Python/3.9.11
Linux/5.15.0-1015-aws exe/x86 64.ubuntu.22
prompt/off command/ec2.run-instances
   userIdentity: { [-]
     accessKeyId: ASIA273YH4WKQTS7ZTUB
     accountId: 755629548949
     arn:
arn:aws:sts::755629548949:assumed-role/ Developer
/testing
     principalId: AROA273YH4WKXW4OHPYYD:testing
     sessionContext: { [+]
     }
     type: AssumedRole
   }
}
```

Investigate the Role



eventTime	eventName	eventSource	sourceIPAddress
2022-08-13T12:30:18Z	GetCallerIdentity	sts.amazonaws.com	3.236.91.34
2022-08-13T12:30:27Z	DescribeSubnets	ec2.amazonaws.com	3.236.91.34
2022-08-13T12:30:29Z	RunInstances	ec2.amazonaws.com	3.236.91.34
2022-08-13T12:30:31Z	RunInstances	ec2.amazonaws.com	3.236.91.34
2022-08-13T12:30:34Z	RunInstances	ec2.amazonaws.com	3.236.91.34
2022-08-13T12:30:36Z	RunInstances	ec2.amazonaws.com	3.236.91.34
2022-08-13T12:30:39Z	RunInstances	ec2.amazonaws.com	3.236.91.34
2022-08-13T12:30:41Z	RunInstances	ec2.amazonaws.com	3.236.91.34
2022-08-13T12:30:43Z	RunInstances	ec2.amazonaws.com	3.236.91.34
2022-08-13T12:30:53Z	CreateUser	iam.amazonaws.com	3.236.91.34
2022-08-13T12:30:54Z	AttachUserPolicy	iam.amazonaws.com	3.236.91.34

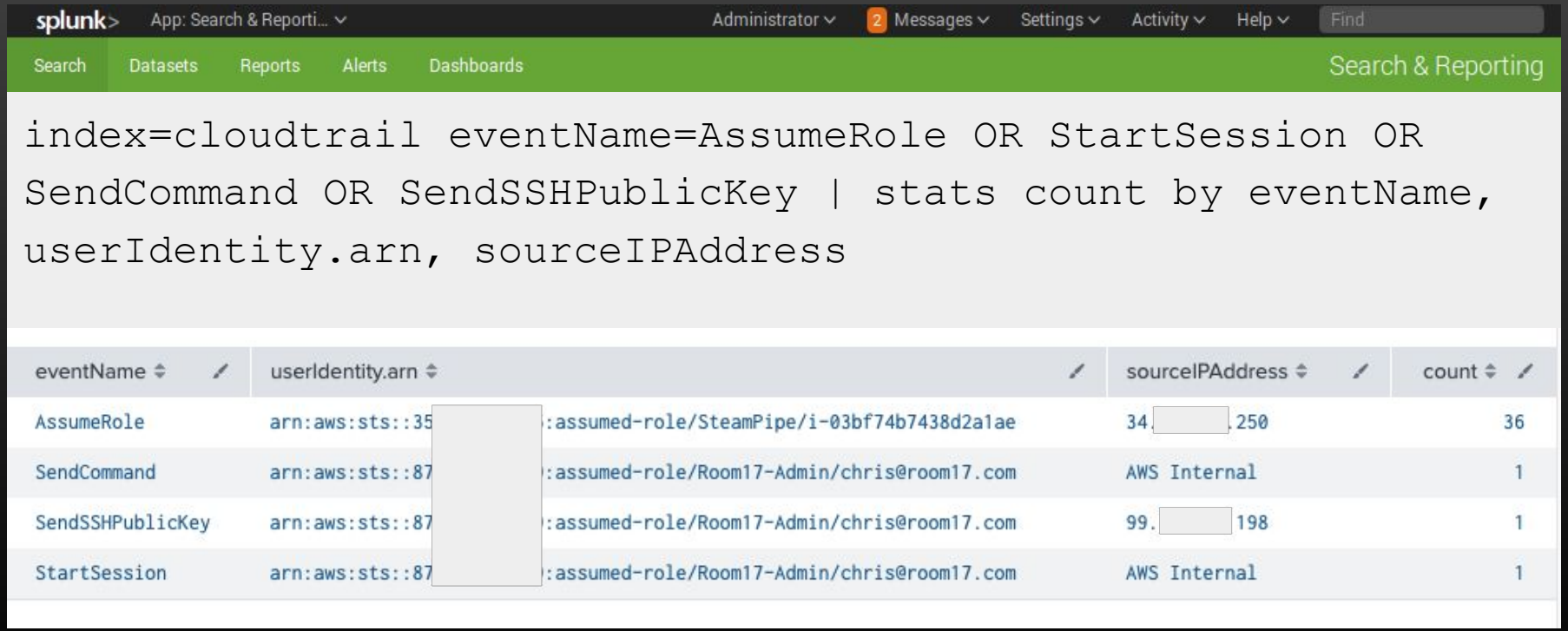
Look for Lateral/Vertical Movement

Signs of Lateral or Vertical Movement:

- sts:AssumeRole (Cloud to Cloud)
- ssm:StartSession (Cloud to Ground)
- ssm:SendCommand (Cloud to Ground)
- ec2-instance-connect:SendSSHPublicKey (Cloud to Ground)
- ec2:AuthorizeSecurityGroupIngress (Cloud to Ground, or Ground to Ground)
- VPC Flow Logs (Ground to Ground)

Note: these actions are abbreviated using IAM Action syntax

Lateral Movement Query



Containment & Eradication

Containment

1. What IR processes have already happened?
2. Invalidate the access keys or session token
3. Identify the initial access vector
4. Figure out what the attacker did
5. Look for Lateral/Vertical Movement

Access Key Quarantine

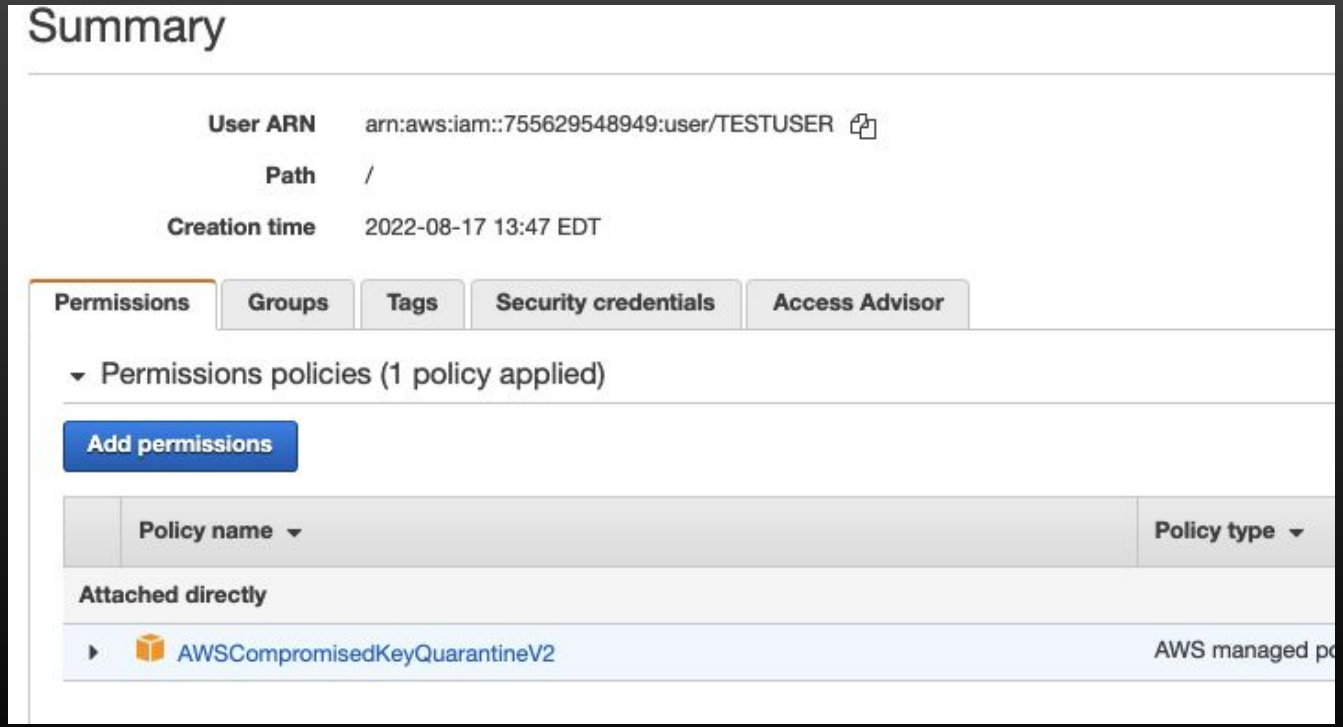
AWS will apply a policy for publicly exposed access keys

Policy denies some actions that can be used for privesc and resource theft

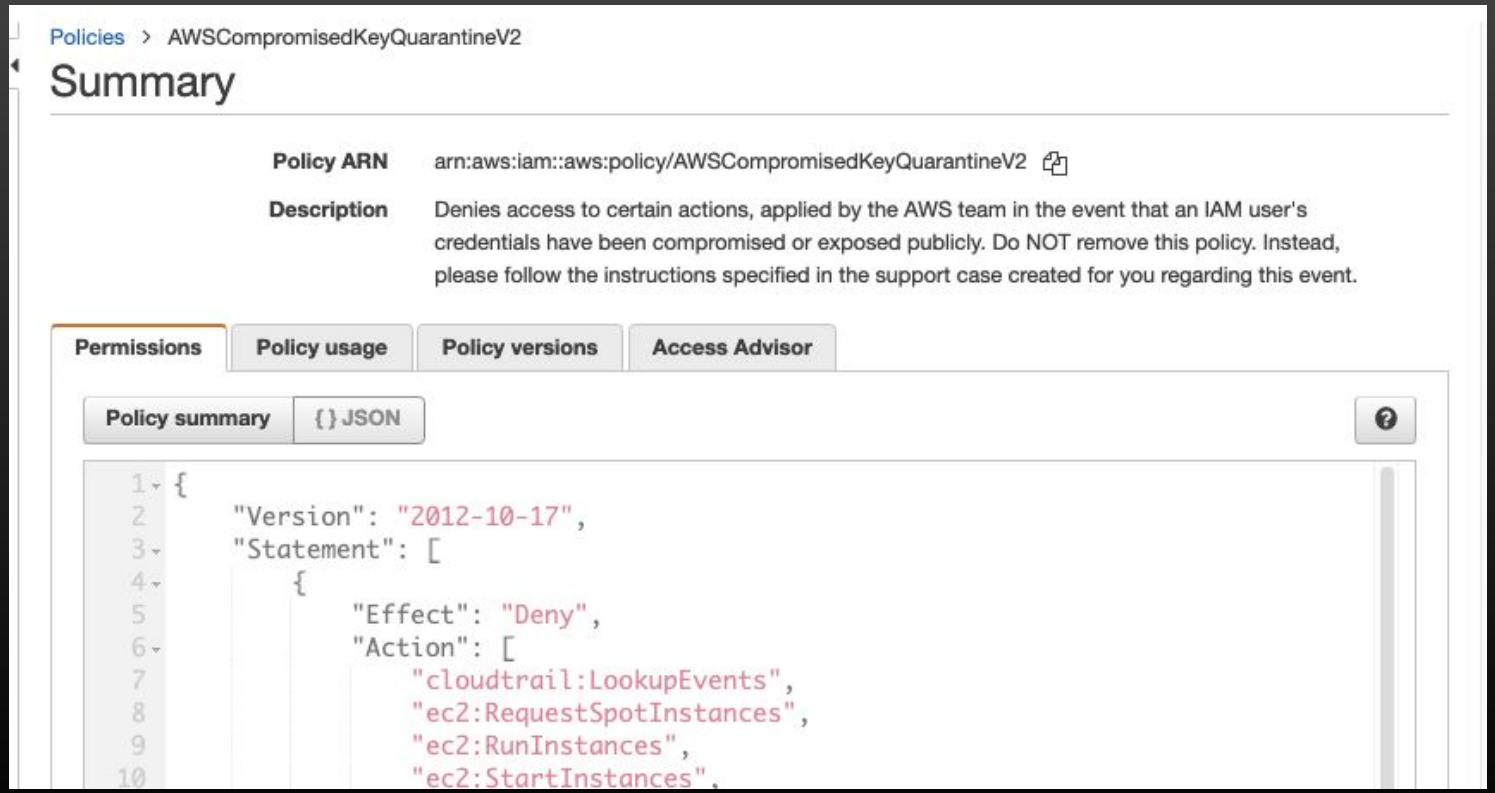
To protect your account from excessive charges and unauthorized activity, we have applied the "AWSCompromisedKeyQuarantineV2" AWS Managed Policy ("Quarantine Policy") to the IAM User listed above. The Quarantine Policy applied to the User protects your account by denying access to high risk actions like iam:CreateAccessKey and ec2:RunInstances.
You can view the policy here: https://console.aws.amazon.com/iam/home#policies/arn:aws:iam::aws:policy/AWSCompromisedKeyQuarantineV2$jsonEditor?section=permissions .

For your security, DO NOT remove the Quarantine Policy before following the instructions below. In cases where the Quarantine Policy is causing production issues you may detach the policy from the user. NOTE: Only users with admin privileges or with access to iam:DetachUserPolicy may remove the policy. For instructions on how to remove managed policies go here: https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage-attach-detach.html#remove-policies-console . In the event of the unauthorized use of your AWS account, we may, at our sole discretion, provide you with concessions. However, a failure to follow the instructions below may jeopardize your ability to receive a concession.

Access Key Quarantine

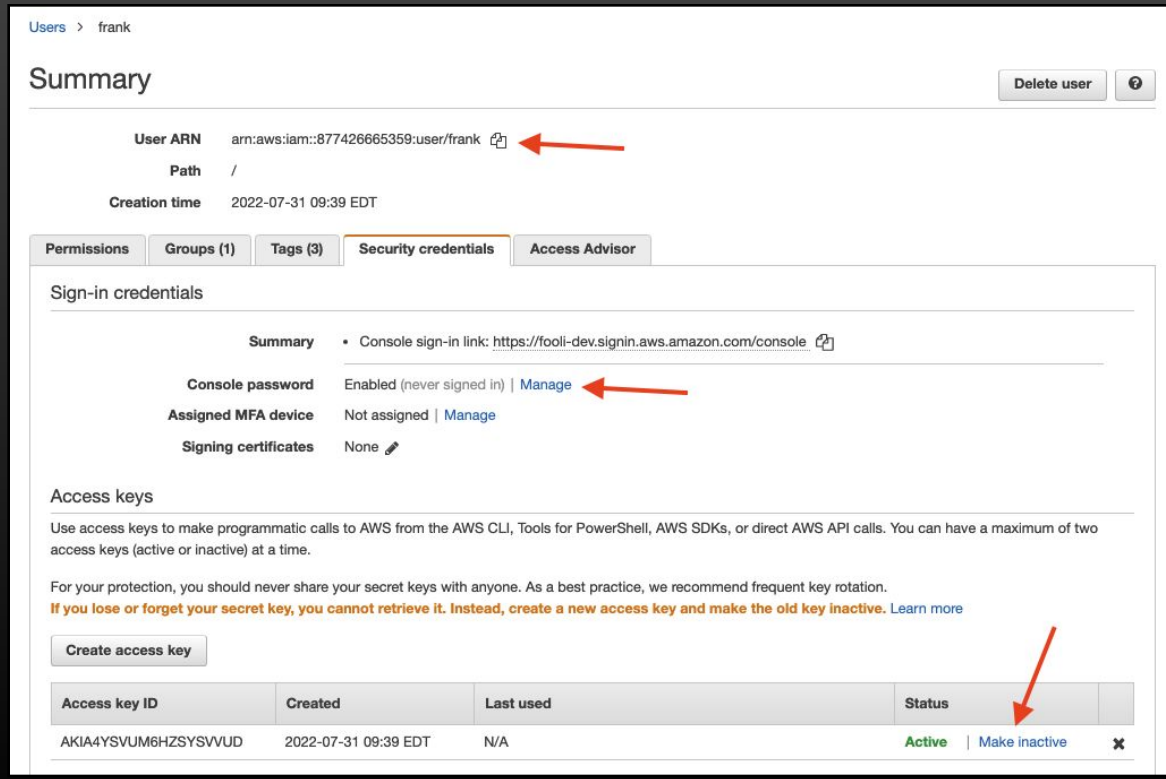


Access Key Quarantine



Disable Access Key

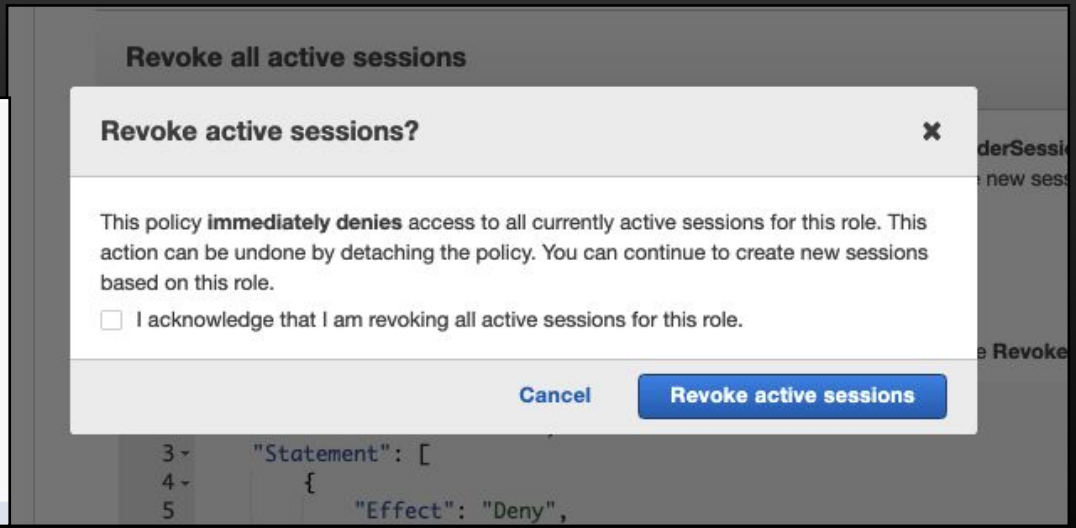
- Quarantine Policy isn't enough!
- Disable the Key ASAP!
- Disable don't Delete



Revoke Active Sessions

- Applicable to Roles which generate temporary credentials
- Applies an explicit DENY for all sessions created before specific timestamp
- Forces apps to get new creds

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "*"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "DateLessThan": {
                    "aws:TokenIssueTime": "[policy creation time]"
                }
            }
        }
    ]
}
```



Apply your own Deny Policy

- Another method of quarantine is to just apply a Deny Policy
- In AWS, an explicit DENY statement overrides any Allow statement
- Dropping a Deny on a compromised key prevents additional damage…
- … while still allowing your to monitor the attacker's *attempts* in CloudTrail

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": ["*"],
      "Resource": ["*"]
    }
  ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "NotAction": ["THINGS YOU NEED"],
      "Resource": ["*"]
    }
  ]
}
```

Apply IP Address Condition

- If you know a key should always be used from a specific set of CIDRs….
- Helps avoid service impacts while executing response

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": ["*"],
      "Resource": ["*"],
      "Condition": {
        "Bool": {"aws:ViaAWSService": "false"},
        "NotIpAddress": {
          "aws:SourceIp": ["192.0.2.0/24","203.0.113.19/32"]
        }
      }
    }
  ]
}
```

Identify the initial access vector

Disabling sessions or keys doesn't help if your attacker can just get new ones

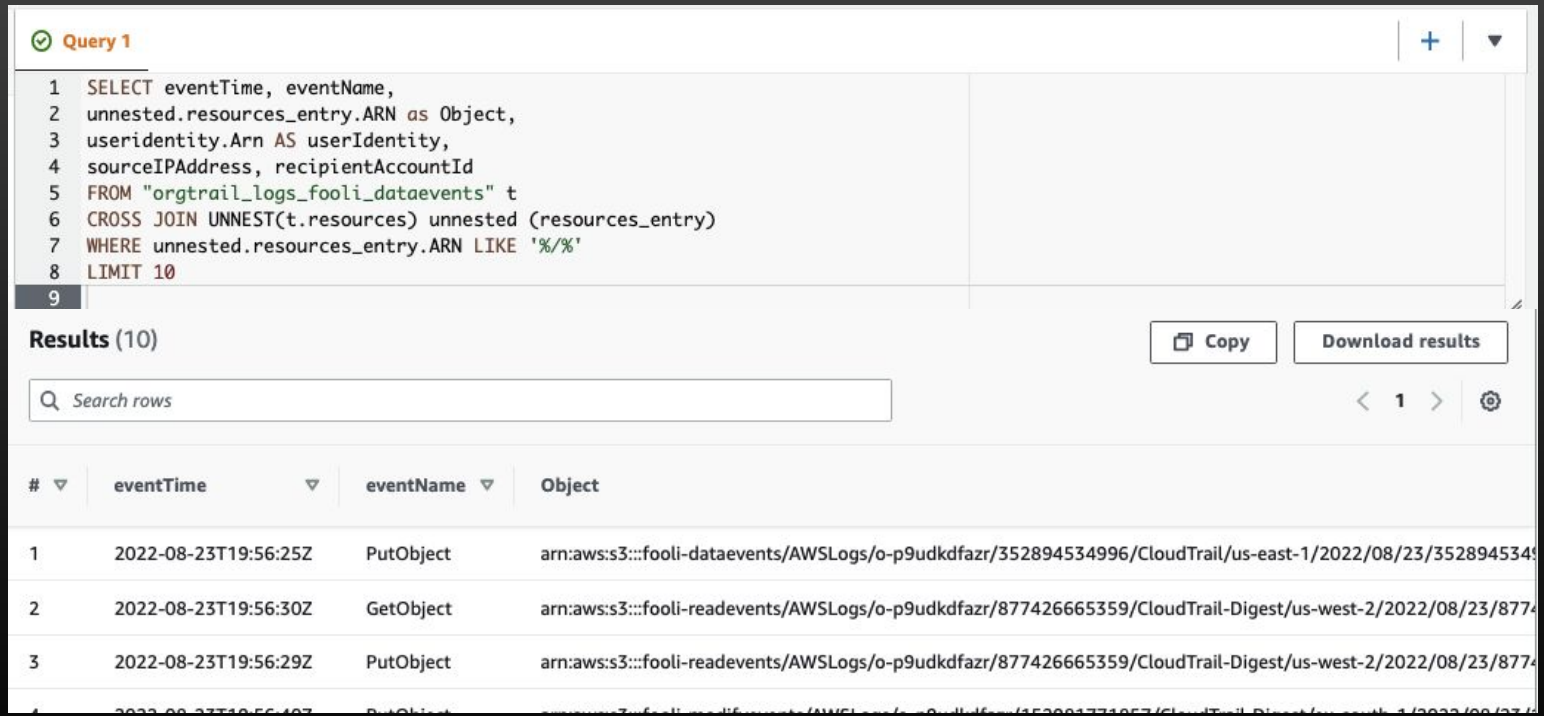
Priority is tracking down where the creds came from

With EC2 Instance Roles, the Instance ID is part of the Role Session Name

Was there a data breach?

- Requires CloudTrail Data Events (aka DataTrails)
- ListBuckets is often the only management event you'll see
- Most cost-effective method here is AWS Athena
- Best to set this up before you need it
- and to practice some of these queries

Athena Query for Data Events



Shameless plug

https://pht.us/class

Two-Day class at BSides Augusta

Students will participate in investigating a simulated attack and subsequent breach

Multiple attack paths, we'll cover CloudTrail, GuardDuty, search queries in Splunk

Dates: September 28-29

Cost: $575

Includes ticket to Security Onion Con

and BSides Augusta

QUESTIONS?

 @jcfarris

 https://github.com/jchrisfarris

 https://www.linkedin.com/in/jcfarris

 http://www.chrisfarris.com