



AWS
re:Invent

SEC339

Actionable threat hunting in AWS

Chris Farris

Cloud Security Lead
WarnerMedia

Suman Koduri

Sr. Technical Account Manager
Amazon Web Services

Agenda

Incident handling 101

Preparation

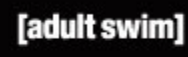
Identification

Containment, eradication, recovery

AWS Support enterprise support plan for incident response

AWS technical account manager

Who	Designated point of contact for AWS Support
What	Provides guidance and advocacy
Where	Operational excellence
When	Application launch, incident management, operational maturity



WarnerMedia™



Incident handling 101

SANS incident handling 101

- Preparation
- Identification
- Containment
- Eradication
- Recovery
- Lessons learned

Ten places your security group should spend time

1. **Accurate account info**
2. Use MFA
3. No hard-coding secrets
4. Limit security groups
5. Intentional data policies
6. **Centralize AWS CloudTrail logs**
7. Validate IAM roles
8. **Take action on GuardDuty findings**
9. Rotate your keys
10. Being involved in dev cycle

Preparation

Preparation

- AWS CloudTrail
- Amazon GuardDuty
- Inventory
- Vulnerability detection
- AWS Support

Centralized AWS CloudTrail

- CloudTrail deployed via CFT in all accounts
- Events written to one bucket per payer
- Dedicated logging account
- Splunk ingests the CloudTrail events



Scale

- 800 AWS accounts
 - 12 organizational payers
- 8.1m CloudTrail events per hour
- 37% are management events
- 18% AssumeRole
- 10% Decrypt

CloudTrail primer

```
{
```

```
  "awsRegion": "us-east-1",
```

```
  "eventName": "CreateBucket",
```

← CreateBucket is the action

```
  "eventSource": "s3.amazonaws.com",
```

← s3 is the AWS service

```
  "eventType": "AwsApiCall",
```

```
  "requestParameters": {},
```

```
  "sourceIPAddress": "192.168.357.420",
```

← Where the call came from

```
  "userIdentity": {
```

```
    "accessKeyId": "ASIATFNORDFNORDAZQ",
```

```
    "accountId": "123456789012",
```

Who did it

```
    "arn": "arn:aws:sts::123456789012:assumed-role/rolename/email@company.com",
```

```
    "type": "AssumedRole"  }
```

← The type of identity

Centralized Amazon GuardDuty

- All GuardDuty findings fed to centralized account
- Amazon CloudWatch Events triggers a push to Splunk via HTTP event collector (HEC)
- Caveat: Must be done in all regions

Sample Code:

<https://github.com/turnerlabs/aws-guardduty-enterprise>

How does GuardDuty work ?

- Baselines accounts
- 30-day learning period
- Leverages AWS internal "threat lists," Proofpoint & CrowdStrike
- You can add your own set of trusted and bad actor IPs



GuardDuty: Event summary

```
index=guardduty
```

```
| dedup id
```

```
| stats count by detail.type
```

- 79% are PortProbeUnprotectedPort
- 4% are unusual IAM recon activity
- 2.5% are logins from unusual IP addresses

Inventory (Antiope)

- Lots of accounts and lots of regions makes for a big haystack
- Enterprise tools are expensive
- Requirement to track cross-account trust relationships
- Search engine to help find gaping security holes
- Opensource
- Pronounced An-Tie-Oh-Pee

<https://github.com/turnerlabs/antiope>

What Antiope collects

- Amazon EC2 instances
- Security groups
- Elastic network interfaces
- Amazon Route 53 domains
- Route 53 zones
- Amazon Elasticsearch Service (Amazon ES)
- Amazon Elastic Container Service (Amazon ECS) tasks & clusters
- Amazon Elastic Container Registry (Amazon ECR) repos
- Amazon CloudFront
- AWS CloudFormation
- AMIs
- VPCs, VPN & direct connect
- AWS Identity and Access Management (IAM) roles & users
- AWS Lambda & Lambda layers
- Trusted advisor
- Support cases

CloudSploit

- Open Source Cloud Vulnerability Scanner
- WarnerMedia executes across all accounts hourly
- Integrated to Antiope
- Security issues presented to account owners via Scorecards (Excel)
- Paid versions available



PSA: Set your security contact

- My new goal is to find account compromise before AWS does
- But if I don't, AWS Abuse team or technical account manager (TAM) will be reaching out
- Set the account security contact to your SOC or IR



Identification

Alternate session title: So now I have three billion compressed json blobs in S3. What's next?

Identification strategy

- CloudTrail to detect events we know are bad
- GuardDuty to correlate events in CloudTrail
- GuardDuty to find events in VPCFlow logs & DNS logs we can't see
- CloudSploit for misconfigured resources
- Antiope to manage, AWS accounts find where a resource is

CloudTrail - IAM Login with no MFA

The screenshot shows the Splunk Search & Reporting interface. The top navigation bar includes the Splunk logo, a dropdown for 'App: Search & Reporti...', user information 'Administrator', a notification badge '2 Messages', and links for 'Settings', 'Activity', and 'Help'. A search bar with the text 'Find' is on the right. Below the navigation bar is a green header with tabs for 'Search', 'Datasets', 'Reports', 'Alerts', and 'Dashboards'. The main content area displays a search query with annotations. The query is: `index=cloudtrail ConsoleLogin`, `"additionalEventData.MFAUsed"!=Yes`, `"userIdentity.type"=IAMUser`, `| dedup userIdentity.arn`, `sourceIPAddress`, `| table "userIdentity.accountId"`, `"userIdentity.arn"`, `sourceIPAddress`, and `"responseElements.ConsoleLogin"`. Three yellow arrows point from annotations to the query: 'Find ConsoleLogin' points to `ConsoleLogin`, 'MFA is not there' points to `"additionalEventData.MFAUsed"!=Yes`, and 'And is an IAM user' points to `"userIdentity.type"=IAMUser`.

```
splunk> App: Search & Reporti... Administrator 2 Messages Settings Activity Help Find
Search Datasets Reports Alerts Dashboards Search & Reporting

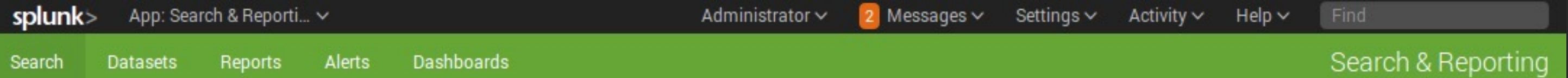
index=cloudtrail ConsoleLogin
"additionalEventData.MFAUsed"!=Yes
"userIdentity.type"=IAMUser
| dedup userIdentity.arn
sourceIPAddress
| table "userIdentity.accountId"
"userIdentity.arn"
sourceIPAddress
"responseElements.ConsoleLogin"
```

Find ConsoleLogin

MFA is not there

And is an IAM user

CloudTrail: Add IAM login locations



```
index=cloudtrail ConsoleLogin "userIdentity.type"=IAMUser
```

```
"additionalEventData.MFAUsed"!=Yes
```

```
| dedup userIdentity.arn sourceIPAddress
```

```
| iplocation sourceIPAddress
```



Process sourceIPAddress

```
| search Country!="United States"
```



Exclude United States

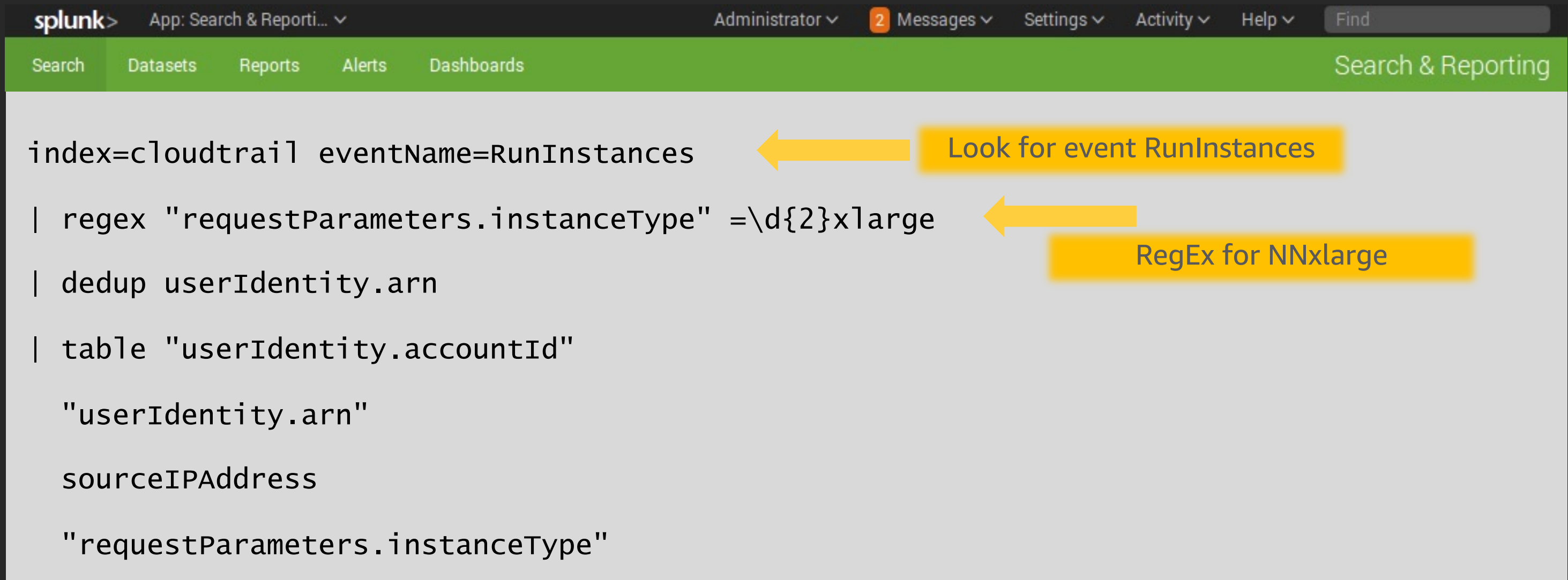
```
| table "userIdentity.accountId"
```

```
  "userIdentity.arn"
```

```
  sourceIPAddress, City, Country
```

```
  "responseElements.ConsoleLogin"
```

CloudTrail: Expensive Amazon EC2 detection



The image shows a screenshot of the Splunk Search & Reporting interface. The top navigation bar includes the Splunk logo, a dropdown menu for 'App: Search & Reporti...', user information 'Administrator', a notification badge '2 Messages', and links for 'Settings', 'Activity', and 'Help'. A search bar with the text 'Find' is on the right. Below this is a green navigation bar with tabs for 'Search', 'Datasets', 'Reports', 'Alerts', and 'Dashboards'. The main content area displays a search query with several annotations:

```
index=cloudtrail eventName=RunInstances  
| regex "requestParameters.instanceType" =\d{2}xlarge  
| dedup userIdentity.arn  
| table "userIdentity.accountId"  
  "userIdentity.arn"  
  sourceIPAddress  
  "requestParameters.instanceType"
```

Annotations and arrows:

- A yellow arrow points from the text "Look for event RunInstances" to the `eventName=RunInstances` part of the query.
- A yellow arrow points from the text "Regex for NNxlarge" to the `regex "requestParameters.instanceType" =\d{2}xlarge` part of the query.

CloudTrail: Open security groups

splunk> App: Search & Reporti... ▾ Administrator ▾ 2 Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

Search Datasets Reports Alerts Dashboards Search & Reporting

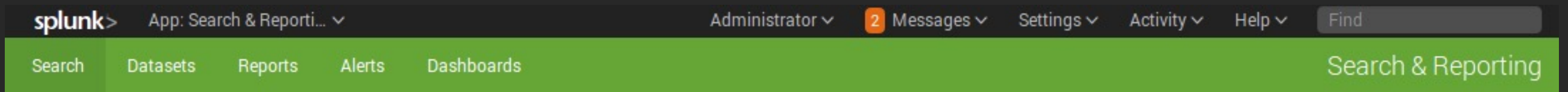
```
index=cloudtrail  
eventName = AuthorizeSecurityGroupIngress  
"requestParameters.ipPermissions.items{}.ipRanges.items{}.cidrIp"="0.0.0.0/0"  
"requestParameters.ipPermissions.items{}.fromPort"=22  
OR "requestParameters.ipPermissions.items{}.fromPort"=3389
```

Request params

Event

Look for SSH or RDP

CloudTrail: User creation detection



```
index=cloudtrail
```

```
eventName="CreateUser"
```



CreateUser

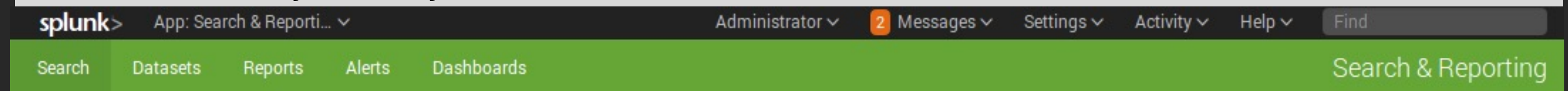
```
sourceIPAddress!="*.amazonaws.com"
```



Not via AWS services

```
| iplocation sourceIPAddress
```

```
| stats count by Country
```



```
index=cloudtrail eventName="CreateUser"
```

```
sourceIPAddress!="*.amazonaws.com"
```



Not via AWS services

```
| iplocation sourceIPAddress
```

```
| search Country!="United States"
```




Exclude from the US

Detection Catalog: CloudTrail Events

- CreateClientVpnEndpoint
- DeleteDetector
- DeleteMembers
- DisassociateFromMasterAccount
- DisassociateMembers
- StopMonitoringMembers
- DeleteTrail
- StopLogging
- UpdateTrail
- AuthorizeSecurityGroupEgress
- AttachInternetGateway
- AttachNetworkInterface*

GuardDuty: Logins from new IP addresses

```
index=guardduty "detail.type"="UnauthorizedAccess:IAMUser/ConsoleLogin"
| dedup detail.service.action.awsApiCallAction.remoteIpDetails.ipAddressV4
| rename "detail.service.action.awsApiCallAction.remoteIpDetails.country.countryName" as Country
| rename "detail.service.action.awsApiCallAction.remoteIpDetails.city.cityName" as City
| rename "detail.service.action.awsApiCallAction.remoteIpDetails.organization.org" as Org
| rename "detail.resource.accessKeyDetails.userName" as UserName
| rename "detail.resource.accessKeyDetails.userType" as LoginType
| rename "detail.service.action.awsApiCallAction.remoteIpDetails.ipAddressV4" as IPAddr
| table UserName City Country IPAddr Org LoginType
```



GuardDuty: Login from new IP addresses results

Atlanta	United States
Atlanta	United States
Los Angeles	United States
Canton	United States
Seattle	United States
Atlanta	United States
Bengaluru	India
Atlanta	United States
Bengaluru	India


Marietta	United States
Accra	Ghana
Chicago	United States
Newark	United States
Lod	Israel

AT&T U-verse	AssumedRole
AT&T U-verse	AssumedRole
Spectrum	IAMUser
Windstream Communications	AssumedRole
T-Mobile USA	AssumedRole
Cyber Wurx LLC	AssumedRole
Jio	AssumedRole
AT&T U-verse	AssumedRole
Bharti Airtel	AssumedRole

AT&T U-verse	AssumedRole
MTN Ghana	IAMUser
Gogo Inflight Internet	AssumedRole
Cogent Communications	IAMUser
INTERWISE Ltd	IAMUser

GuardDuty: RDP brute force report

```
index=guardduty "detail.type"="UnauthorizedAccess:EC2/RDPBruteForce"
| dedup id
| rename
"detail.service.action.networkConnectionAction.remoteIpDetails.country.countryName" as
Country
| rename "detail.service.action.networkConnectionAction.remoteIpDetails.city.cityName" as
City
| rename "detail.service.action.networkConnectionAction.remoteIpDetails.organization.org"
as Org
| rename "detail.resource.instanceDetails.instanceId" as Target
| rename "detail.service.action.networkConnectionAction.remoteIpDetails.ipAddressV4" as
IPAddr
| table City Country IPAddr Org Target
```



GuardDuty: RDP brute force results

City ↕	Country ↕	Org ↕	IPAddr ↕	Port ↕	instan
	Panama	NForce Entertainment B.V.	45.227.255.20	3389	i-0f8
	Panama	NForce Entertainment B.V.	45.227.255.20	3389	i-036
	Russia	Arturas Zavaliauskas	185.254.120.21	3389	i-079
	Moldova	RM Engineering LLC	185.153.196.40	3389	i-095

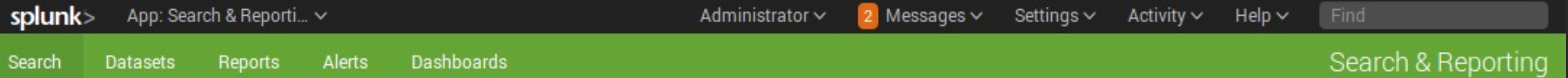
This is the difference between:
"Hey you have misconfigured your security group"
and
"Hey, you're under attack"

Antiope: Public ElasticSearch cluster

```
splunk> App: Search & Reporti... Administrator 2 Messages Settings Activity Help Find
Search Datasets Reports Alerts Dashboards Search & Reporting

index=antiope resourceType="AWS::ElasticSearch::Domain"
NOT configuration.VPCOptions.VPCId=* ← Not in a VPC
NOT ".AccessPolicies.Statement{}.Condition.IpAddress.aws:SourceIp{}"=*
NOT ".AccessPolicies.Statement{}.Condition.IpAddress.aws:SourceIp"=*
NOT ".AccessPolicies.Statement{}.Condition.StringEquals.aws:SourceVpc"=*
| regex ".AccessPolicies.Statement{}.Principal.AWS"="\*" ← Anyone can access
| dedup resourceId
| table configuration.Endpoint resourceName awsAccountName
```

Antiope: Support cases



All support cases

```
index=antiope resourceType="AWS::Support::Case"
```

Focus on the resource type

```
| dedup resourceId
```

Get only the latest

```
| table awsAccountName configuration.serviceCode
```

```
configuration.categoryCode
```

```
configuration.status configuration.subject
```

All support cases opened regarding the AWS account

```
index=antiope resourceType="AWS::Support::Case"
```

```
"configuration.serviceCode"="customer-account"
```

Customer-account is where security problems appear

```
| dedup resourceId
```

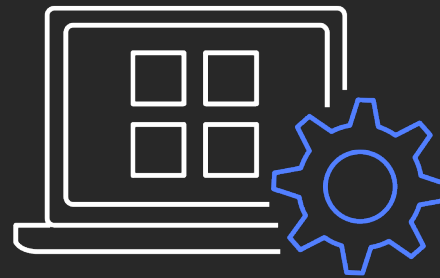
NEW!

Amazon Detective

Quickly analyze, investigate, and identify the root cause of security issues



Built-in data
collection



Automated analysis



Visual insights

Containment, eradication, & recovery

Containment, eradication & recovery

- Review CloudTrail
- What user did it?
- Rotate password & access key
- What else did they do?

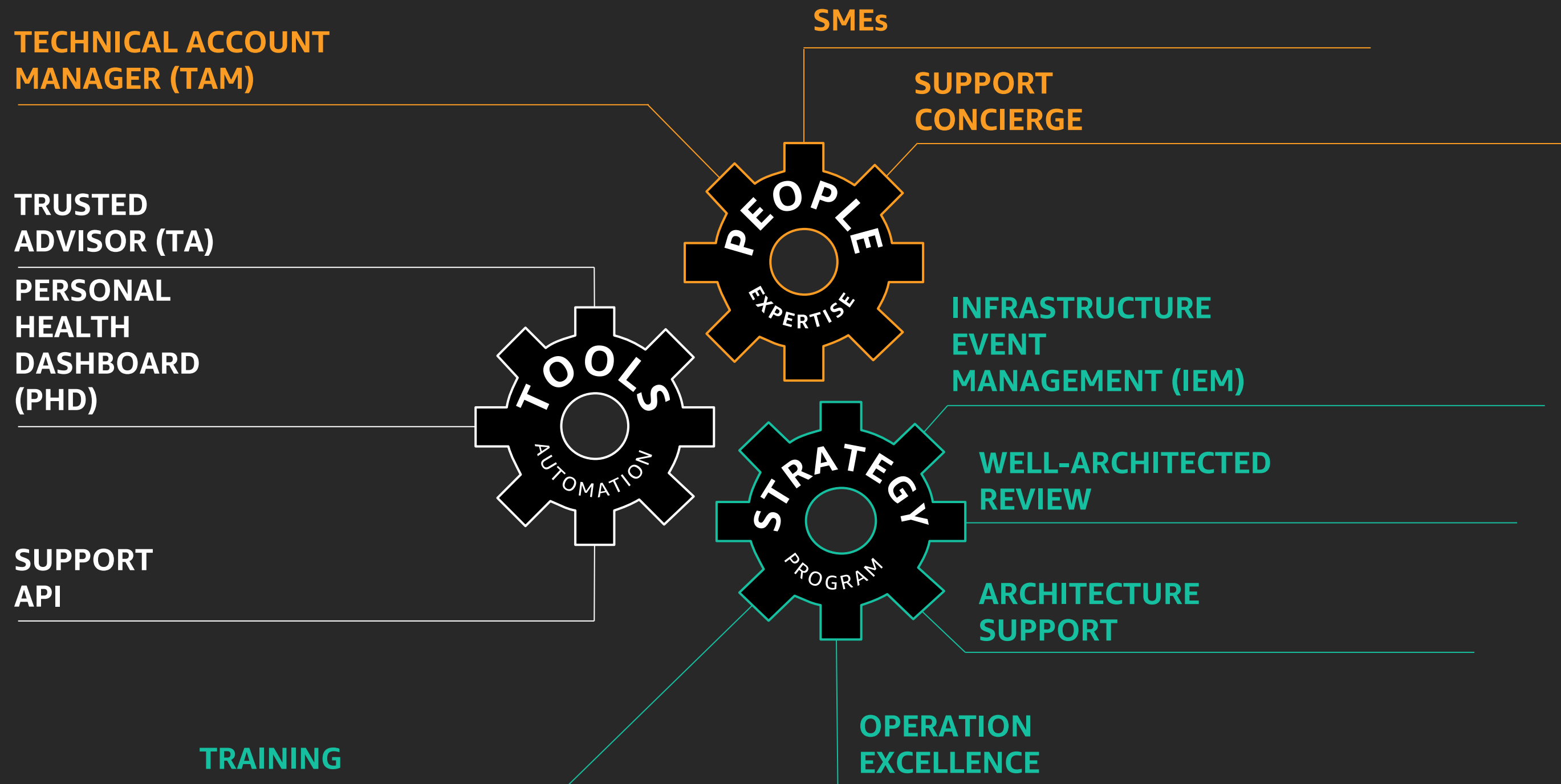
CloudTrail is an effective tool for account compromise analysis

Containment, eradication & recovery

- Isolate instances with pre-built IR security groups
- Leverage tools for instance forensics
 - ssm-acquire can be fully automated
 - Threat Response and Margarita Shotgun are good too
- <https://forensicate.cloud/> for more resources

Enterprise support value

AWS enterprise support



Enterprise support value to security teams

Proactive

Alert on security issues & remediate them

Design

Deliver customized training & help architectural decision

Incident
management

Provide timely support by working with AWS service teams

Operational
excellence

Help optimize & recommend ways to use services more efficiently

Redesign

Enhance the architecture using upcoming features

Links

GuardDuty deployment

<https://github.com/turnerlabs/aws-guardduty-enterprise>

Antiope

<https://github.com/turnerlabs/antiope>

ssm-acquire

<https://github.com/mozilla/ssm-acquire>

CloudSploit

<https://github.com/cloudsploit/scans>

Splunk queries

<https://www.chrisfarris.com/post/reinvent2019-sec339/>

EC2 DFIR

<https://forensicate.cloud/>

Learn security with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate cloud security skills



30+ free digital courses cover topics related to cloud security, including Introduction to Amazon GuardDuty and Deep Dive on Container Security



Classroom offerings, like AWS Security Engineering on AWS, feature AWS expert instructors and hands-on activities



Validate expertise with the **AWS Certified Security - Specialty** exam

Visit aws.amazon.com/training/paths-specialty/

Thank you!

Chris Farris

@jcfarris

www.linkedin.com/in/jcfarris

Suman Koduri

@sumankoduri

www.linkedin.com/in/sumankoduri/



Please complete the session
survey in the mobile app.