Detecting AWS cross-account activity

Applicability

- Product: Splunk Cloud Platform or Splunk Enterprise
- Feature: Splunk App for AWS and Splunk Add-on for AWS
- Data: Amazon Web Services: CloudTrail and CloudWatch
- Function: Monitoring

Scenario

You are an Amazon Web Services (AWS) admin who manages access to AWS resources and services across your organization. You do this by using AWS's Identity and Access Management (IAM) functionality. IAM provides the ability to create and manage AWS users, groups, and roles—each with their own unique set of privileges and defined access to specific resources (such as EC2 instances, the AWS Management Console, API, or the command-line interface). Unlike conventional (human) users, IAM roles are assumable by anyone in the organization. They provide users with dynamically created, temporary security credentials which expire within a set time period.

However, problems can occur in between the time when the temporary credentials are issued and when they expire. This gap represents a window of opportunity for a malicious actor to leverage the temporary credentials to spin up or remove instances, create new users, elevate privileges, and perform other malicious activities throughout the environment.

Since AWS CloudTrail tracks cross-account activity to its origin, you can run searches that will help you monitor your AWS CloudTrail logs for evidence of this type of suspicious activity. For example, while accessing multiple AWS accounts and roles may be perfectly valid behavior, it may be suspicious when an account requests privileges of an account it has not accessed in the past. After identifying suspicious activities, you can use the provided investigative searches to help you probe more deeply.

Some commands, parameters, and field names in the searches below may need to be adjusted to match your environment. In addition, to optimize the searches shown below, you should specify an index and a time range when appropriate. To run these searches, install the AWS App for Splunk (version 5.1.0 or later) and Splunk Add-on for AWS (version 4.4.0 or later), then configure your CloudTrail inputs.

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
Support searches

**Previously seen AWS cross-account activity**

This search looks for AssumeRole events where the requesting account differs from the requested account, then writes these relationships to a lookup file.

Validate the user name entries in `previously_seen_aws_cross_account_activity.csv`, a lookup file created by this support search.

```
sourcetype=aws:cloudtrail eventName=AssumeRole | spath output=requestingAccountId path=userIdentity.accountId | spath output=requestedAccountId path=resources{}.accountId | search requestingAccountId=* | where requestingAccountId!=requestedAccountId | stats earliest(_time) AS firstTime latest(_time) AS lastTime BY requestingAccountId, requestedAccountId | outputlookup previously_seen_aws_cross_account_activity | stats count
```

Detection searches

**AWS detect attach to role policy**

To run this search, install the AWS App for Splunk (version 5.1.0 or later) and Splunk Add-on for AWS (version 4.4.0 or later), then configure your CloudWatch inputs.

This search provides detection of an user attaching itself to a different role trust policy. This can be used for lateral movement and escalation of privileges.

Attach to policy can create a lot of noise. This search can be adjusted to provide specific values to identify cases of abuse (i.e status=failure). The search can provide context for common users attaching themselves to higher privilege policies or even newly created policies.

```
| search (sourcetype="aws:cloudwatchlogs:eks" attach policy) | spath requestParameters.policyArn | table sourceIPAddress, user_access_key, "userIdentity.arn", "userIdentity.sessionContext.sessionIssuer.arn", eventName, errorCode, errorMessage, status, action, "requestParameters.policyArn", "userIdentity.sessionContext.attributes.mfaAuthenticated", "userIdentity.sessionContext.attributes.creationDate"
```

**AWS detect permanent key creation**

To run this search, install the AWS App for Splunk (version 5.1.0 or later) and Splunk Add-on for AWS (version 4.4.0 or later), then configure your CloudWatch inputs.

This search provides detection of accounts creating permanent keys. Permanent keys are not created by default and
they are only needed for programmatic calls. Creation of Permanent key is an important event to monitor.

False positives from this search may occur since not all permanent key creations are malicious. If there is a policy of rotating keys this search can be adjusted to provide better context.

```
| search (sourcetype="aws:cloudwatchlogs:eks" CreateAccessKey)
| spath eventName
| search (eventName=CreateAccessKey "userIdentity.type"=IAMUser) | table sourceIPAddress, userName, "userIdentity.type", userAgent, action, status, "responseElements.accessKey.createDate", "responseElements.accessKey.status", "responseElements.accessKey.accessKeyId"
```

► AWS detect role creation

To run this search, install the AWS App for Splunk (version 5.1.0 or later) and Splunk Add-on for AWS (version 4.4.0 or later), then configure your CloudWatch inputs.

This search provides detection of role creation by IAM users. Role creation is an event by itself if user is creating a new role with trust policies different than the available in AWS and it can be used for lateral movement and escalation of privileges.

CreateRole is not very common in common users. This search can be adjusted to provide specific values to identify cases of abuse. In general AWS provides plenty of trust policies that fit most use cases.

```
| search (action=created event_name=CreateRole "requestParameters.description"=Allows*) sourcetype="aws:cloudwatchlogs:eks" "userIdentity.type"=AssumedRole)
| table sourceIPAddress, "userIdentity.principalId", "userIdentity.arn", action, event_name, awsRegion, http_user_agent, mfa_auth, msg, "requestParameters.roleName", "requestParameters.description", "responseElements.role.arn", "responseElements.role.createDate"
```

► AWS detect sts:AssumeRole abuse

To run this search, install the AWS App for Splunk (version 5.1.0 or later) and Splunk Add-on for AWS (version 4.4.0 or later), then configure your CloudTrail inputs.

This search provides detection of suspicious use of sts:AssumeRole. These tokens can be created on the go and used by attackers to move laterally and escalate privileges.

Sts:AssumeRole can be very noisy as it is a standard mechanism to provide cross account and cross resources access. This search can be adjusted to provide specific values to identify cases of abuse.

```
| search (sourcetype=aws:cloudtrail "userIdentity.sessionContext.sessionIssuer.type"=Role userType=AssumedRole)
| table sourceIPAddress, "userIdentity.arn", user_agent, user_access_key, status, action, "requestParameters.roleName", "responseElements.role.roleName", "responseElements.role.createDate"
```

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
AWS detect sts:GetSessionToken abuse

To run this search, install the AWS App for Splunk (version 5.1.0 or later) and Splunk Add-on for AWS (version 4.4.0 or later), then configure your CloudWatch inputs.

This search provides detection of suspicious use of sts:GetSessionToken. These tokens can be created on the go and used by attackers to move laterally and escalate privileges.

sts:GetSessionToken can be very noisy as in certain environments numerous calls of this type can be executed. This search can be adjusted to provide specific values to identify cases of abuse. In specific environments the use of field requestParameters.serialNumber will need to be used.

| search (sourcetype="aws:cloudwatchlogs:eks" "userIdentity.type"=IAMUser ASIA) | spath eventName | search eventName=GetSessionToken | table sourceIPAddress, eventTime, "userIdentity.arn", userName, userAgent, user_type, status, region |

Cross-account activity from previously unseen account

Run the Previously seen AWS cross account activity support search only once to create the baseline of previously seen cross account activity.

This search looks for AssumeRole events where an IAM role in a different account is requested for the first time.

Click here to read a detailed breakdown of how this search works.

| sourcetype=aws:cloudtrail eventName=AssumeRole | spath output=requestingAccountld path=userIdentity.accountld | spath output=requestedAccountld path=resources{}.accountld | search requestingAccountld=*
| where requestingAccountld != requestedAccountld | inputlookup append=t previously_seen_aws_cross_account_activity | multireport [ | stats min(firstTime) AS firstTime max(lastTime) AS lastTime BY requestingAccountld, requestedAccountld | outputlookup previously_seen_aws_cross_account_activity | where fact=fiction [ | eventstats earliest(eval(coalesce(_time, firstTime))) AS firstTime, latest(eval(coalesce(_time, lastTime))) AS lastTime BY requestingAccountld, requestedAccountld | where firstTime >= relative_time(now(), "-1h@h") AND isnotnull(_time) | spath output=accessKeyId path=responseElements.credentials.accessKeyId | spath output=requestingARN path=resources{}.ARN | stats values(awsRegion) AS awsRegion values(sharedEventID) AS sharedEventID, values(requestingARN) AS src_user, values(responseElements.assumedRoleUser.arn) AS dest_user by _time, requestingAccountld, requestedAccountld, accessKeyId | table _time, src_user, requestingAccountld, dest_user, requestedAccountld, awsRegion, accessKeyId, sharedEventID |
Investigative searches

► Investigate user activities by AccessKeyId

This search retrieves the times, ARN, source IPs, AWS regions, event names, and the result of the event for specific credentials.

```
| search sourcetype=aws:cloudtrail userIdentity.accessKeyId={accessKeyId} | spath output=user path=userIdentity.arn | rename sourceIPAddress AS src_ip | table _time, user, src_ip, awsRegion, eventName, errorCode, errorMessage
```

► Investigate user activities by source user

This search retrieves the times, ARN, source IPs, AWS regions, event names, and the result of the event for specific ARNs.

```
| search sourcetype=aws:cloudtrail userIdentity.arn={src_user} | spath output=user path=userIdentity.arn | rename sourceIPAddress AS src_ip | table _time, user, src_ip, awsRegion, eventName, errorCode, errorMessage
```

Additional resources

This use case is included within Splunk Enterprise Security, a Splunk app that provides prebuilt content and searches to help answer root-cause questions in real-time about malicious and anomalous events in your IT infrastructure. In addition, Splunk Enterprise Security provides a number of other searches to help reinforce your Cloud Security posture, including:

- Detect suspicious new instances in your Elastic Compute Cloud (EC2) environment
- Detecting AWS network ACL activity
- Detecting AWS suspicious provisioning activities
- Monitoring users in AWS
- Monitoring AWS Elastic Compute Cloud (EC2) for suspicious login activities
- Monitoring AWS S3 for suspicious activities
- Monitoring AWS for suspicious traffic
- Monitoring AWS Elastic Compute Cloud (EC2) for unusual modification