



Record AWS API calls to improve IAM Policies

Michael Wittig – 11 Sep 2020



Have you ever looked at an IAM policy and wondered: Is it really necessary to grant access to this specific action? Or do you need to know which API calls a legacy or 3rd party application is actually sending to come up with a secure IAM policy? CloudTrail can help here, but there is something better: Record API calls with the AWS SDKs and CLI (including the stuff that is not visible in CloudTrail).



Do you prefer listening to a podcast episode over reading a blog post? Here you go!



In this blog post, you learn to capture the data without touching source code. You also analyze the data and use the results to improve your IAM policies.

Record AWS API calls to improve IAM Policies in Action

Watch the following video to learn how to record AWS API calls to improve your IAM Policies.

Turning Client Side Monitoring on

All AWS SDKs and the AWS CLI support “Client Side Monitoring (CSM)”. Luckily, most applications use AWS SDKs to integrate with AWS. If you enable CSM, each API request is reported via UDP on port 31000. You can turn on CSM by setting the environment variable `AWS_CSM_ENABLED` to `true` or via the shared config file `~/.aws/config`:

```
[default]
csm_enabled = true

[profile profile1]
csm_enabled = true
```

Warning Keep in mind that you need to add the `csm_enabled` setting for each Linux user, e.g.:

```
/root/.aws/config
/home/ec2-user/.aws/config
```

Warning Keep in mind that you have to restart the process that uses an AWS SDK after changing the config!

You can check if API calls are reported with this command:

```
tcpdump -i lo -n udp port 31000 -A
```

To debug a process where no data shows up, get the PID of the process with `ps -ef`, and inspect the environment variables:

```
xargs -0 -L1 -a /proc/PID/environ
```

If `HOME` or `AWS_CSM_ENABLED` are not defined, CSM will not work. If `AWS_CONFIG_FILE` is defined, you have to edit that file and append

```
csm_enabled = true
```

```
csm_enabled = true.
```

If the process is started by `systemd`, edit the unit file (e.g., `/usr/lib/systemd/system/amazon-ssm-agent.service`) and turn on `AWS_CSM_ENABLED` in the `[Service]` section:

```
[Service]
Environment=AWS_CSM_ENABLED=true
```

Next, you will learn how to capture and archive the CSM data to S3. Doing so allows you to analyze the data with the help of Athena later.

Capturing the data

First, create an S3 bucket for storing the data.

`fluentd` can listen on a UDP port, transform and buffer the data, and upload it to S3.

On Amazon Linux 2 EC2 instance, installing `fluentd` is a one-liner (other distros are supported as well):

```
curl -L https://toolbelt.treasuredata.com/sh/install-amazon2-td-agent4.sh | sh
```

Allow your EC2 instance to interact with the newly created bucket (replace `BUCKET_NAME` with the name of your bucket) by adding the following IAM policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject", "s3:PutObject"],
      "Resource": "arn:aws:s3:::BUCKET_NAME/awscsm/*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::BUCKET_NAME"
    }
  ]
}
```

Last but not least, configure `fluentd` by replacing the file `/etc/td-agent/td-agent.conf` with the following content (replace `BUCKET_NAME` with the name of your bucket, and `REGION` with your AWS region (e.g., `us-east-1`)):

```
<source>
  @type udp
  tag awscsm
```

```

<parse>
  @type json
</parse>
port 31000
bind 127.0.0.1
</source>
<filter awscsm>
  @type record_transformer
  <record>
    hostname "#{Socket.gethostname}"
  </record>
</filter>
<match awscsm>
  @type s3
  s3_region REGION
  s3_bucket BUCKET_NAME
  check_apikey_on_start false
  check_bucket false
  path ${tag}/year=%Y/month=%m/day=%d/
  <buffer tag,time>
    @type memory
    timekey 10m
    timekey_use_utc true
    timekey_wait 1m
    chunk_limit_size 256m
  </buffer>
  <format>
    @type json
  </format>
</match>

```

Activate the new **fluentd** configuration with this command:



Please support our work!

We have published 327 articles, 42 podcast episodes, and 15 videos. It's all **free** and means **a lot of work in our spare time**.

If you value the work we do, you should support us. **With your help, we can spend enough time to keep publishing great content in the future.** We look forward to sharing our AWS knowledge with you.

Support us

```
systemctl start td-agent.service
```

If data comes in, **fluentd** uploads a file to S3 every 10 minutes (or every 256 MB) with a 1-minute delay.

I recommend waiting for a couple of days to capture enough data.

Analyzing the data

Create an AWS Glue Crawler that looks into **s3://BUCKET_NAME/awscsm/** every hour. Run the crawler manually to speed up table creation for the first time.

After the crawler finished, switch to Athena. There you will find a table **awscsm** that you can query.

To get an idea of how the data looks like, run:

```
SELECT * FROM awscsm LIMIT 25
```

Only get API calls, but not the attempts (one call can have multiple attempts):

```
SELECT * FROM awscsm WHERE type='ApiCall' LIMIT 25
```

Get the most popular API calls:

```
SELECT service, api, COUNT(*) as count FROM awscsm WHERE type='ApiCall' GROUP by service, api ORDER BY count DESC LIMIT 25
```

Comparing with CloudTrail

I use the following CloudWatch Insights query to get the most popular API calls (replace **IAM_ROLE_NAME** with the name of the IAM role attached to your EC2 instance):

```
fields @timestamp, @message
| filter userIdentity.arn like "IAM_ROLE_NAME"
| stats count() as count by eventSource, eventName
| sort count desc
| limit 25
```

The top calls look entirely different. From CSM, I get:

Results			
	service ▾	api ▾	count ▾
1	CloudWatch Logs	PutLogEvents	21247
2	CloudWatch	PutMetricData	4320
3	SQS	ReceiveMessage	4318
4	SSM MDS	GetMessages	4278
5	SSM	UpdateInstanceInformation	288
6	S3	HeadObject	146
7	S3	PutObject	145
8	SSM	ListInstanceAssociations	144
9	IAM	ListUsers	144
10	SSM	ListAssociations	144
11	ec2metadata	GetMetadata	7
12	ec2metadata	GetToken	4
13	ec2metadata	GetDynamicData	3
14	SQS	DeleteMessageBatch	2
15	Auto Scaling	CompleteLifecycleAction	2

And from CloudTrail, I get:

#	eventSource	eventName	count
▶ 1	ssm.amazonaws.com	UpdateInstanceInformation	289
▶ 2	iam.amazonaws.com	ListUsers	144
▶ 3	autoscaling.amazonaws.com	CompleteLifecycleAction	2

As you can see, CloudTrail does not capture most of the “data” events. Unfortunately, most calls of a typical application are categorized as “data” events.

Summary

Securing IAM policies of running systems is hard. You need all available data to reduce the risk of accidentally removing permissions required by the system. CloudTrail provides a good foundation. Unfortunately, not all API calls are visible in CloudTrail. E.g., SQS “data events” are not captured by CloudTrail. Client Side Monitoring (CSM) can be used to capture the calls that are made with AWS SDKs and the AWS CLI. Both sources combined can help you to detect IAM permissions that are not needed anymore.

Thanks, [Scott Piper](#), for bringing CSM to my attention.

Written by Michael Wittig on 11 Sep 2020

Tags: [aws](#) [iam](#) [highlight](#) [csm](#)





Michael Wittig

I launched cloudonaut.io in 2015 with my brother Andreas. Since then, we have published hundreds of articles, podcast episodes, and videos. It's all free and means a lot of work in our spare time. We enjoy sharing our AWS knowledge with you. Have you learned something new by reading, listening, or watching our content? If so, we kindly ask you to support us in producing high-quality & independent AWS content. We look forward to sharing our AWS knowledge with you.

[Support us](#)

Feedback? Questions? You can reach me via [Email](#), [Twitter](#), or [LinkedIn](#).

Further reading

[Review: AWS App Mesh – A service mesh for EC2, ECS, and EKS](#)

[Run the AWS CLI v2 inside Docker](#)

[Messaging on AWS](#)

Subscribe

Deepen your knowledge about AWS, stay up to date!

☐

I want to subscribe to the newsletter with new content as well as announcements regarding products and services. The newsletter performance is measured based on opens and clicks. I agree with the [privacy policy](#).

[Subscribe](#)

Do you prefer RSS? [Feedly](#) or [RSS feed](#)

Network

[Blog](#)

[Podcast](#)

[Twitter](#)

[YouTube](#)

Legal

[Imprint](#)

[Privacy Policy](#)

Projects

[Malware protection for Amazon S3](#)

[Chatbot for AWS Monitoring](#)

[Complete AWS IAM Reference](#)

[Free Templates for AWS CloudFormation](#)

[Rapid CloudFormation: modular, production ready, open source.](#)