New issue

Jump to bottom

Custom Authorizer and AWS deployed stack don't have the same behavior #1259

Closed leonardoviveiros opened this issue on Aug 5, 2021 · 2 comments

leonardoviveiros commented on Aug 5, 2021

Bug Report

Current Behavior

When using a Custom Authorizer, the behaviour of serverless-offline differs from the deployed stack on AWS.

Sample Code

This is where we define the function event trigger. As it's clear to see, we expect a HTTP POST on `/{stage}/dashboard/`

- file: trailingSlash/index.ts

```
events: [{
      http: {
              method: 'post',
              path: 'dashboard',
              cors: true,
              authorizer: 'auth',
      }
}]
```

Our custom authorizer **generatePolicy** method looks like this.

- file: auth/handler.js

```
const generatePolicy = (principalId: string, methodArn: string, role: string) => {

        const allowedResources = [];
        //arn:aws:execute-api:region:account-id:api-id/stage-name/HTTP-method/resource-path
        const baseArn = methodArn.split(`/`, 2).join('/');

        switch (role.toUpperCase()) {
                case 'ADMIN':
                        allowedResources.push(`${baseArn}/*/*`)
                        break;
                case 'USER':
                        allowedResources.push(`${baseArn}/*/dashboard/*`)
                        break;
                default:
                        break;
        }
        const generatedPolicy = {
                principalId: principalId,
                policyDocument: {
                        Version: "2012-10-17",
                        Statement: [{
                                Action: "execute-api:Invoke",
                                Effect: "Allow",
                                Resource: [
                                        ...allowedResources,
                                ]
                        }],
                },
        };
        return generatedPolicy;
};
```

We're basically generating the following policy to someone with the role "USER":

```
{
        "principalId": "1234567890",
        "policyDocument": {
                "Version": "2012-10-17",
                "Statement": [{
                        "Action": "execute-api:Invoke",
                        "Effect": "Allow",
                        "Resource": [
                        "arn:aws:execute-api:us-east-1:random-account-id:random-api-id/dev/*/dashboard/*"
                        ]
                }]
        }
}
```

Expected behavior/code

When testing locally using serverless-offline, fetching the endpoint http://localhost:3000/dev/dashboard/, the response is 403 Forbidden as the screenshot shows.
But when we deploy the stack to AWS, fetching the endpoint https://RANDOM.execute-api.us-east-1.amazonaws.com/dev/dashboard/, the result is 200 ok as its seen here.

Environment

- `serverless` version: v2.53.0
- `serverless-offline` version: v8.0.0
- `node.js` version: v12.21.0
- os : Linux Mint 19.1 Tessa

Additional context/Screenshots

We found this issue while doing the research The Fault in Our Stars, in which we explore how API Gateway Execute API Policy works under different conditions.
One researcher from our company opened the issue 1191 where he indicates another incorrectly behaviour by serverless-offline regarding the way it evaluates policies. It still lacks a response to this date.

👍 2 🚀 1 👀 3

leonardoviveiros commented on Aug 10, 2021 Author

@dherault We reserved the CVE-2021-38384 with Mitre to this vulnerability, but we also would like to submit it via Github, and also make a pull request to help solve this issue

justinwiley commented on Nov 29, 2021

Wondering if there is an update for this...Is this vulnerability considered acceptable by project authors, or is there a fix in development?

👍 7

dherault closed this as completed on Apr 13

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants