Article
José Pereira · Dec 22, 2021 5m read

Open Exchange

ZAP API scan GitHub action

What about having your IRIS REST APIs scanned every push you did and being reported on possible vulnerabilities? This is what I am going to show you in this article.

Recently, we had the Security Contest with amazing applications and examples showing how to improve security on your IRIS solutions. One of such examples was the zap-api-scan-sample, made by me and my colleague Henrique Dias. Our application shows how to use the OWASP ZAP API scanner to perform security tests on your REST APIs OpenAPI definitions generated by IRIS.

Now, we did an improvement on such example, using ZAP GitHub Action to automate security APIs tests on every push to a GitHub repository.

What are GitHub Actions?

GitHub actions are plugins for GitHub repositories, which perform tasks triggered by some events, like pushes for instance. They are nice tools to set up CD/CI pipelines.

You can set up actions by creating YAML files, located in the .github/workflows directory. Below an example is presented showing how to start a ZAP API scan. You can find more information in the quick start from GitHub Docs .

GitHub actions are available for download in the GitHub marketplace.

What is OWASP ZAP?

The Open Web Application Security Project® (OWASP) is a nonprofit foundation created and maintained by security enthusiasts around the world, developing amazing projects.

One of such projects is the OWASP® Zed Attack Proxy (ZAP), a web app scanner free and open source. This tool can perform scanning on web applications and/or APIs (REST, SOAP and GraphQL) searching for common vulnerabilities. You can also run penetration tests using this great tool.

You can find more information about ZAP here.

Example

So, as you can imagine, the OWASP ZAP API scan GitHub action is a hook to perform API scan automatically on repository events, such as pushes.

In order to set up this action, you need to create a file with any name in the .github/workflows directory. Inside this directory, you must create a YAML file with a name for your action, the event which triggers the action and a ZAP job config.

Below the action set up for perform API scan in our application example is presented:

```
name: owasp-zap-api-scan
on: push

jobs:
  zap_scan:
    name: Scan REST APIs
    runs-on: ubuntu-latest
    steps:
      - name: ZAP Scan for crudall API
        uses: zaproxy/action-api-scan@v0.1.0
        with:
          format: openapi
          target: 'https://zapsample.demo.community.intersystems.com/crudall/_spec'
```

In our example, this file is located here.

As you can see by the value " push " in property " on " , the ZAP job configured in " jobs " property is executed on every push to the repository. So, when a push is done, an action starts, like this one:

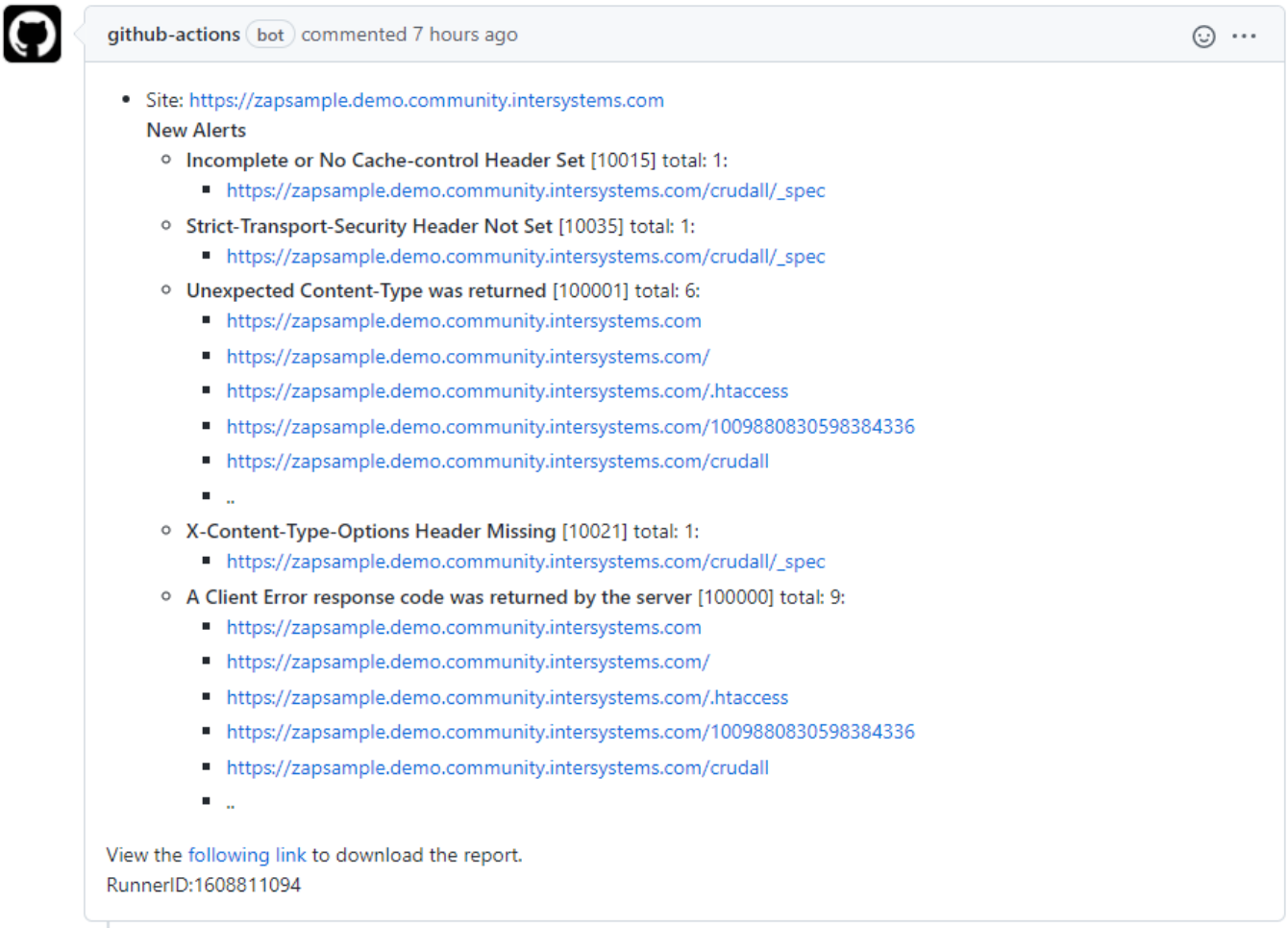
```
Run zaproxy/action-api-scan@v0.1.0
[@octokit/rest] `const Octokit = require("@octokit/rest")` is deprecated. Use `const
{ Octokit } = require("@octokit/rest")` instead
starting the program
github run id :1608811094
/usr/bin/docker pull owasp/zap2docker-stable -q
docker.io/owasp/zap2docker-stable:latest
/usr/bin/docker run --user root -v /home/runner/work/zap-api-scan-sample/zap-api-scan
-sample:/zap/wrk/:rw --network=host -t owasp/zap2docker-stable zap-api-scan.py -t htt
ps://zapsample.demo.community.intersystems.com/crudall/_spec -f openapi -J report_jso
n.json -w report_md.md -r report_html.html
2021-12-21 21:50:20,854 Could not find custom hooks file at /home/zap/.zap_hooks.py
2021-12-21 21:50:32,043 Number of Imported URLs: 3
Total of 10 URLs
PASS: Directory Browsing [0]
PASS: Vulnerable JS Library [10003]
...
WARN-NEW: Unexpected Content-Type was returned [100001] x 5
    https://zapsample.demo.community.intersystems.com/1009880830598384336 (404 Not Fo
und)
    https://zapsample.demo.community.intersystems.com/crudall (404 Not Found)
    https://zapsample.demo.community.intersystems.com/ (404 Not Found)
    https://zapsample.demo.community.intersystems.com/elmah.axd (404 Not Found)
    https://zapsample.demo.community.intersystems.com/.htaccess (404 Not Found)
WARN-NEW: Incomplete or No Cache-control Header Set [10015] x 1
    https://zapsample.demo.community.intersystems.com/crudall/_spec (200 OK)
WARN-NEW: X-Content-Type-Options Header Missing [10021] x 1
    https://zapsample.demo.community.intersystems.com/crudall/_spec (200 OK)
WARN-NEW: Strict-Transport-Security Header Not Set [10035] x 1
    https://zapsample.demo.community.intersystems.com/crudall/_spec (200 OK)
FAIL-NEW: 0 FAIL-INPROG: 0  WARN-NEW: 4 WARN-INPROG: 0  INFO: 0 IGNORE: 0   PASS: 73
[@octokit/rest] `const Octokit = require("@octokit/rest")` is deprecated. Use `const
{ Octokit } = require("@octokit/rest")` instead
Scanning process completed, starting to analyze the results!
Alerts present in the current report: true
Process completed successfully and a new issue #3 has been created for the ZAP Scan.
Total size of all the files uploaded is 9209 bytes
Finished uploading artifact zap_scan. Reported size is 9209 bytes. There were 0 items
```

```
that failed to upload
```

You can find all actions output in the Actions tab in GitHub.

Note that you must have your API online in order for ZAP to scan it. In our example, we used another action to deploy the project to the cloud first and then, set the API endpoint which uses IRIS to generate the OpenAPI specification - defined in the “ target ” property.

This action also opens an issue with warnings and errors found, and send notifications to GitHub and to your e-mail:



Conclusion

In this article you learned how to use GitHub Actions to automate APIs scan tests on every push done to your repositories.
Such a tool could help you to quickly identify vulnerabilities into your REST APIs on your IRIS solutions.

#API #GitHub #REST API #Security #InterSystems IRIS #Open Exchange
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/zap-api-scan-github-action