2023

Test Report

TEST PLAN VALIDATION

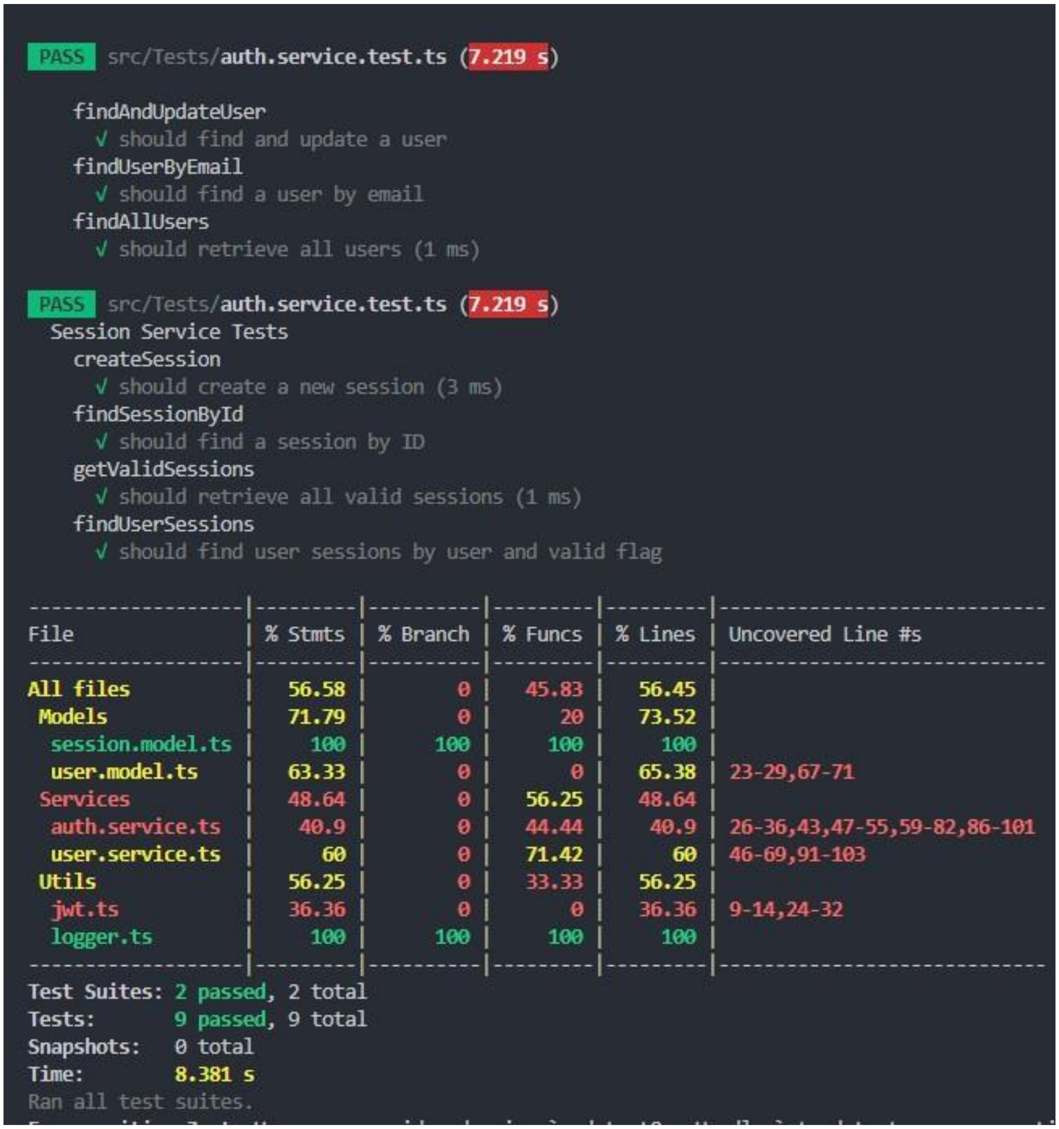
NEDELCHEV,TSANKO T.N.

Contents

Introduction

The purpose of this document is to provide an answer to the test plan that has been defined for this project. This report goes through all the tests defined in the test plan and shows the results of the tests that have been performed based on that plan and the acceptance criteria This report defines whether the test was a success or a failure.

Unit Testing



```
 PASS  src/Tests/auth.service.test.ts (7.219 s)

    findAndUpdateUser
      √ should find and update a user
    findUserByEmail
      √ should find a user by email
    findAllUsers
      √ should retrieve all users (1 ms)

 PASS  src/Tests/auth.service.test.ts (7.219 s)
  Session Service Tests
    createSession
      √ should create a new session (3 ms)
    findSessionById
      √ should find a session by ID
    getValidSessions
      √ should retrieve all valid sessions (1 ms)
    findUserSessions
      √ should find user sessions by user and valid flag

-------------------|---------|----------|---------|---------|------------------------------
File               | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-------------------|---------|----------|---------|---------|------------------------------
All files          |   56.58 |        0 |   45.83 |   56.45 |
 Models            |   71.79 |        0 |      20 |   73.52 |
  session.model.ts |     100 |      100 |     100 |     100 |
  user.model.ts    |   63.33 |        0 |       0 |   65.38 | 23-29,67-71
 Services          |   48.64 |        0 |   56.25 |   48.64 |
  auth.service.ts  |    40.9 |        0 |   44.44 |    40.9 | 26-36,43,47-55,59-82,86-101
  user.service.ts  |      60 |        0 |   71.42 |      60 | 46-69,91-103
 Utils             |   56.25 |        0 |   33.33 |   56.25 |
  jwt.ts           |   36.36 |        0 |       0 |   36.36 | 9-14,24-32
  logger.ts        |     100 |      100 |     100 |     100 |
-------------------|---------|----------|---------|---------|------------------------------
Test Suites: 2 passed, 2 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        8.381 s
Ran all test suites.
```

Unit tests have been defined for the service layer of the auth microservice and the postfeed microservice. The tests run and all pass.

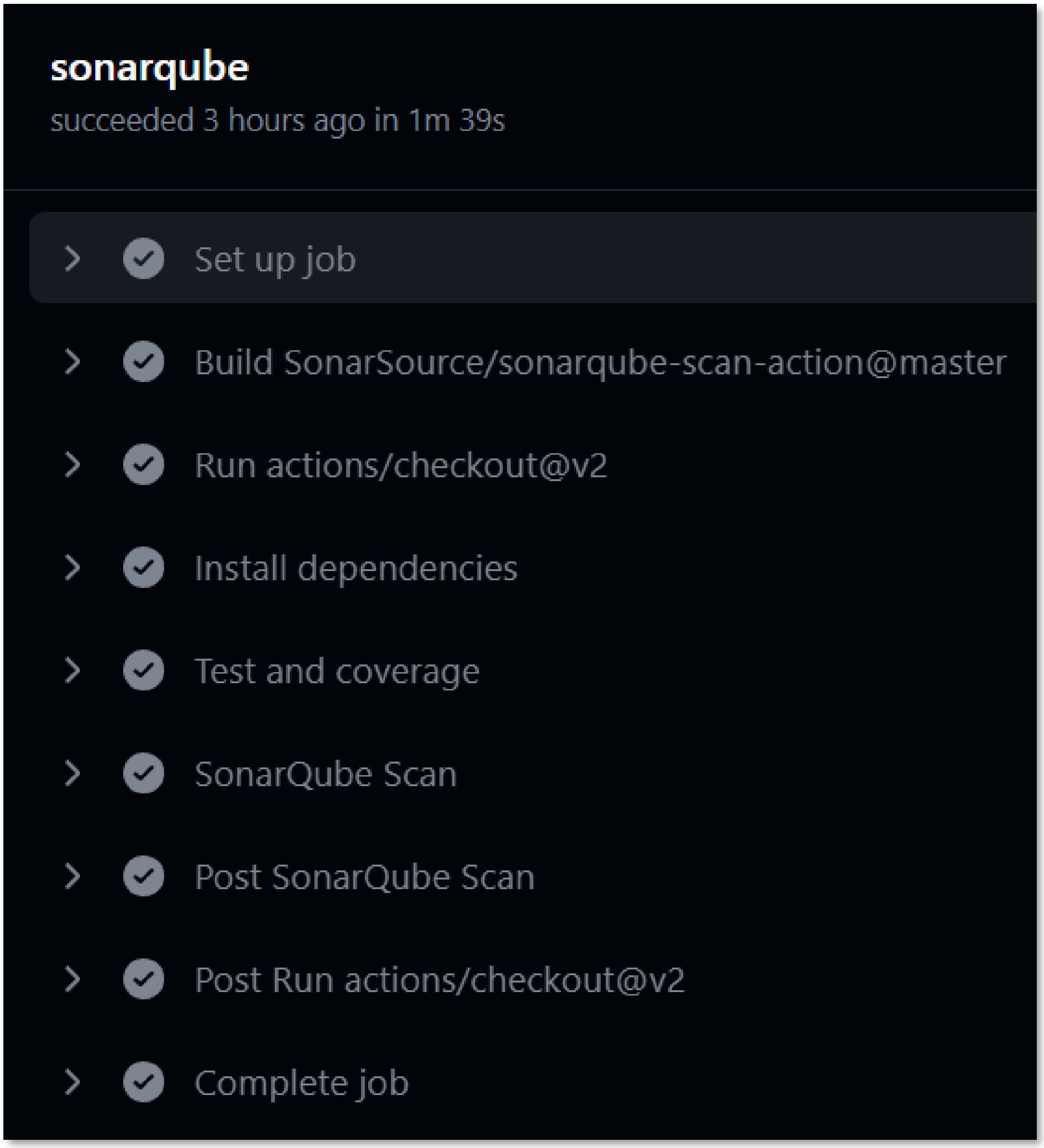
Therefore, this test is a success, based on the criteria defined in the Test Plan.

Integration testing

Integration testing via Cypress as described in the test plan was not performed due to lack of time and therefore this test is considered a failure.

Quality Testing

Quality Testing as defined in the test plan should be covered by SonarQube. SonarQube is setup in the pipeline and it triggers a scan on every push to the dev branch.

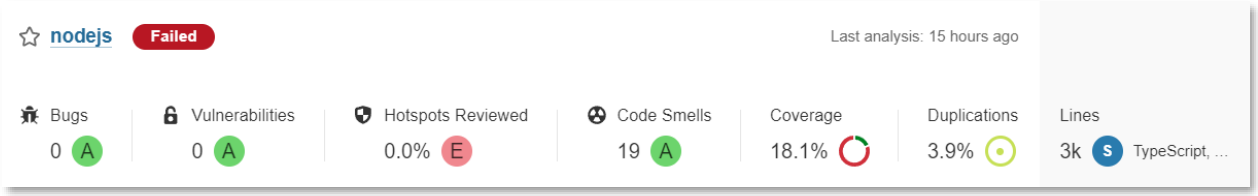


sonarqube

succeeded 3 hours ago in 1m 39s

- ✓ Set up job
- ✓ Build SonarSource/sonarqube-scan-action@master
- ✓ Run actions/checkout@v2
- ✓ Install dependencies
- ✓ Test and coverage
- ✓ SonarQube Scan
- ✓ Post SonarQube Scan
- ✓ Post Run actions/checkout@v2
- ✓ Complete job

The pipeline runs through successfully and triggers a SonarQube Scan.



☆ nodejs Failed

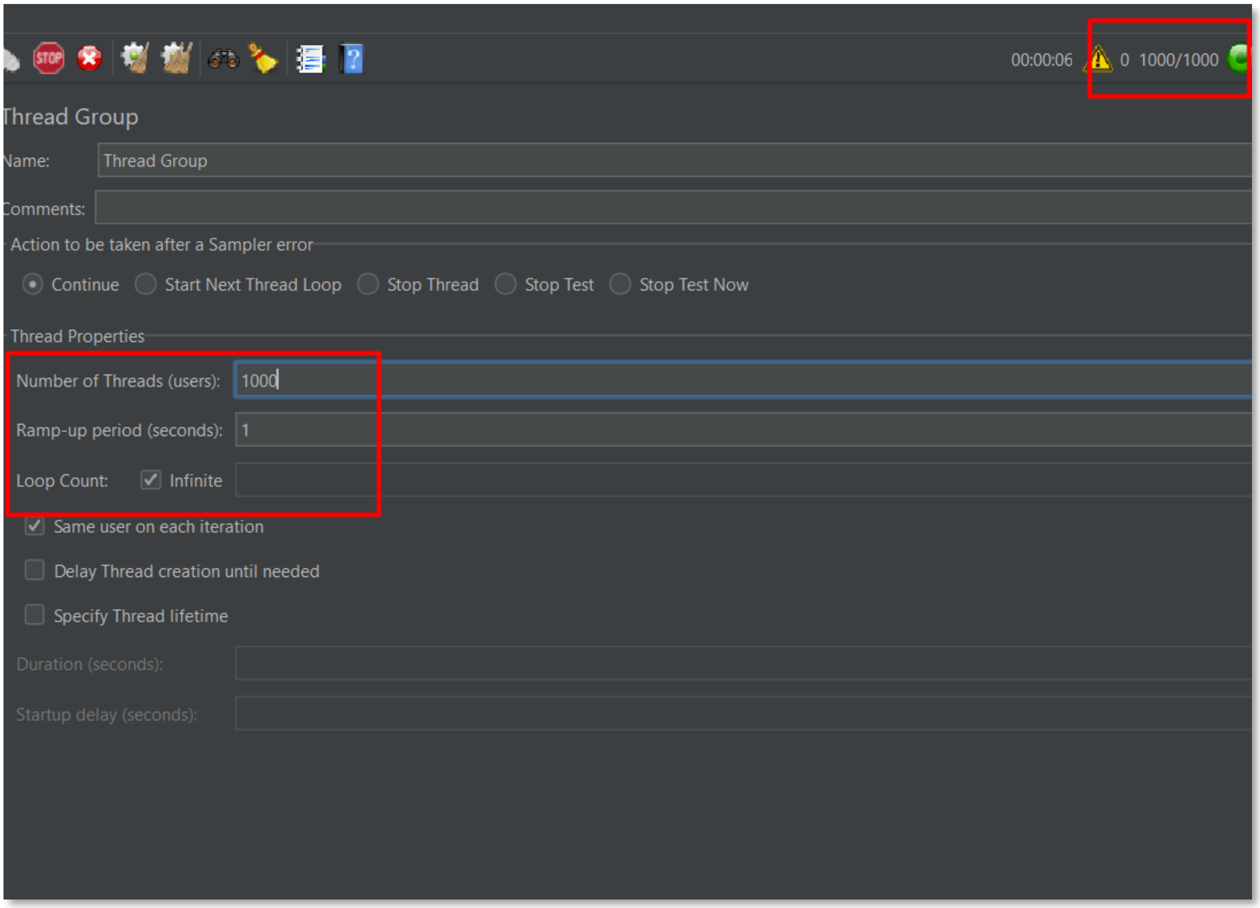
Last analysis: 15 hours ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
0 A	0 A	0.0% E	19 A	18.1%	3.9%	3k S TypeScript, ...

The scan fails due to the Unit Test coverage as the default value is 80% however based on the acceptance criteria from the test plan the test is still a success as there are no bugs present.

Performance Testing

A test was performed using Apache JMeter on the deployed cluster. The test consisted of 1000 users making requests every second to the two back end APIs.



00:00:06 ⚠ 0 1000/1000

Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error

● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now

Thread Properties

Number of Threads (users): 1000

Ramp-up period (seconds): 1

Loop Count: ☑ Infinite

☑ Same user on each iteration

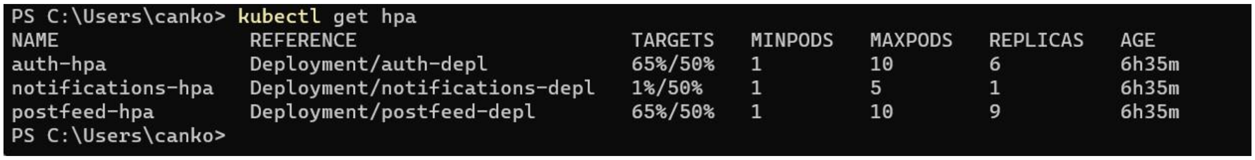
□ Delay Thread creation until needed

□ Specify Thread lifetime

Duration (seconds):

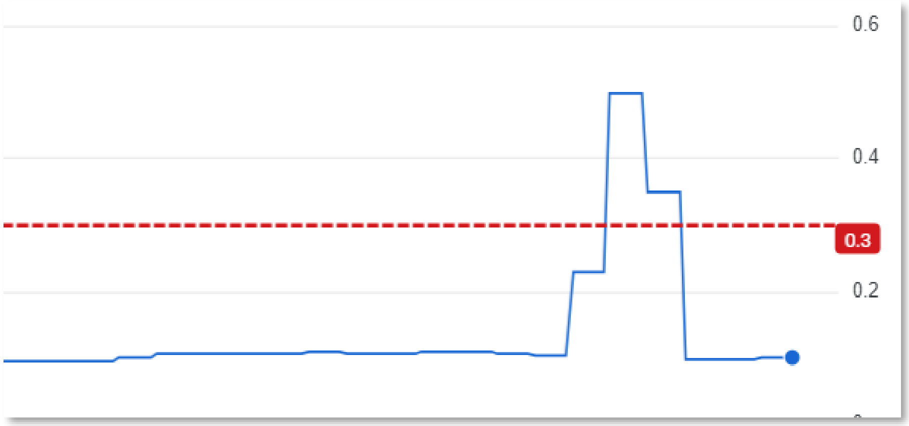
Startup delay (seconds):

The scan is triggered and the test begins. This is equivalent to 2000 requests every second.

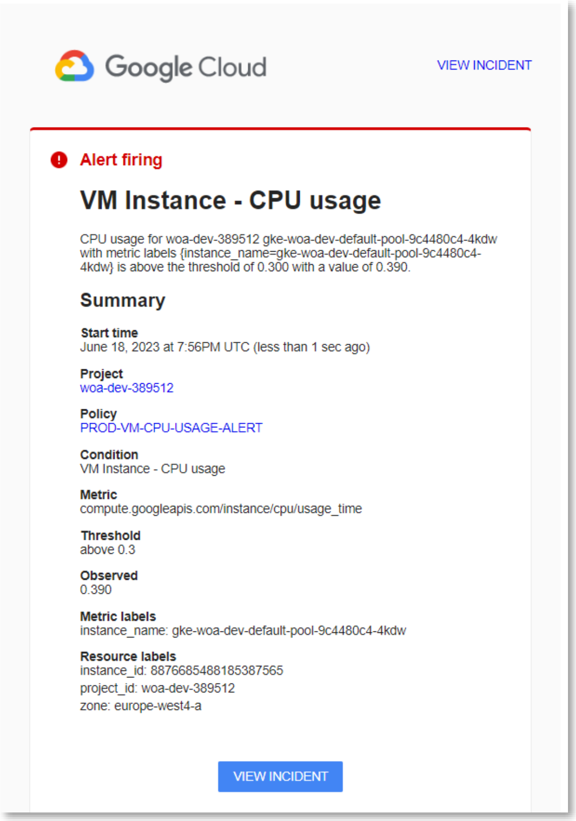


```
PS C:\Users\canko> kubectl get hpa
NAME                REFERENCE                       TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
auth-hpa            Deployment/auth-depl            65%/50%   1         10        6          6h35m
notifications-hpa   Deployment/notifications-depl   1%/50%    1         5         1          6h35m
postfeed-hpa        Deployment/postfeed-depl        65%/50%   1         10        9          6h35m
PS C:\Users\canko>
```

The results of the scan indicate that the pods for the auth api and the postfeed api got under heavy load and started scaling horizontally.



The CPU usage on the VM went way up and triggered an alert. The alert was sent to my email as a notification.



Alert firing

VM Instance - CPU usage

CPU usage for woa-dev-389512 gke-woa-dev-default-pool-9c4480c4-4kdw with metric labels {instance_name=gke-woa-dev-default-pool-9c4480c4-4kdw} is above the threshold of 0.300 with a value of 0.390.

Summary

Start time
June 18, 2023 at 7:56PM UTC (less than 1 sec ago)

Project
woa-dev-389512

Policy
PROD-VM-CPU-USAGE-ALERT

Condition
VM Instance - CPU usage

Metric
compute.googleapis.com/instance/cpu/usage_time

Threshold
above 0.3

Observed
0.390

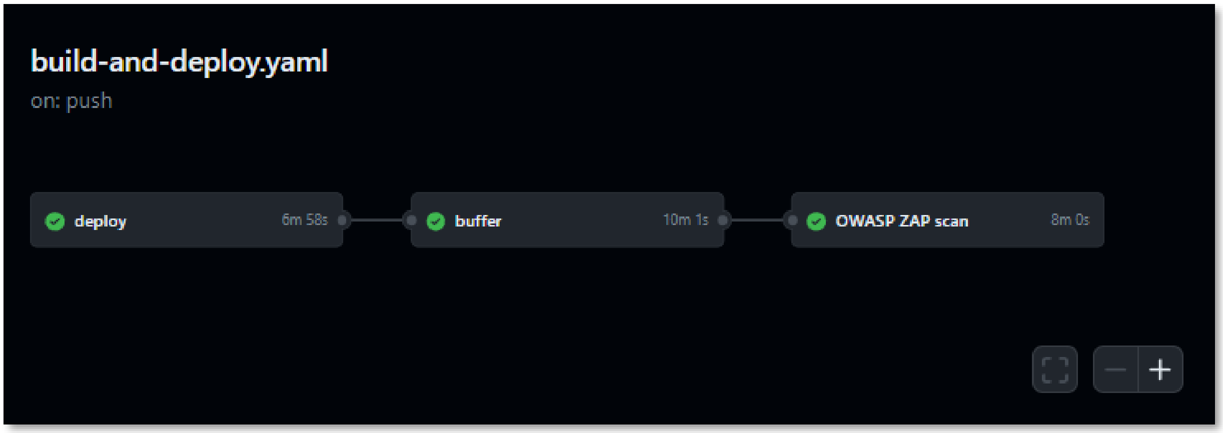
Metric labels
instance_name: gke-woa-dev-default-pool-9c4480c4-4kdw

Resource labels
instance_id: 8876685488185387565
project_id: woa-dev-389512
zone: europe-west4-a

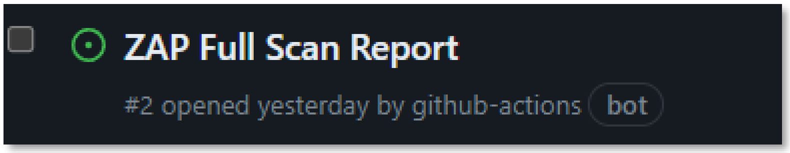
Judging by the criteria set in the test plan this test was a success.

Security testing

The security test defined in the test plan indicated that the deployed application should be scanned with OWASP ZAP which triggers through the pipeline.

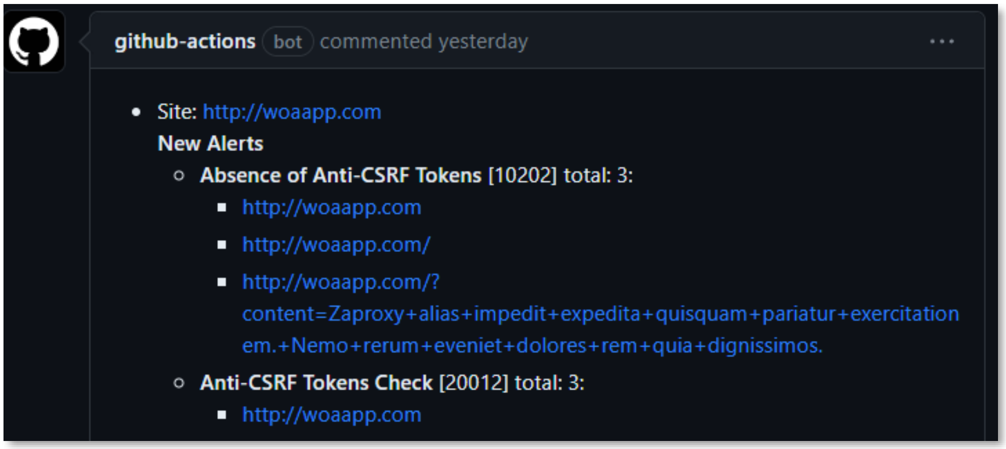


The pipeline setup for deployment and OWASP ZAP scan works in the following way. After the deployment is finished the buffer task begins. The buffer task is a buffer that waits for exactly 10 minutes after the deployment before it triggers the OWASP ZAP scanner. The reason for that distinction is that the cluster needs some time to setup after a deployment. That's why a healthy window of 10 minutes is required. After the cluster has scaled down and stabilized the OWASP ZAP scan is triggered and scans the application for 8 minutes. After it has finished any new issues are automatically added to the GitHub issue board where they await a checkup by the developers.



☐ ZAP Full Scan Report

#2 opened yesterday by github-actions bot

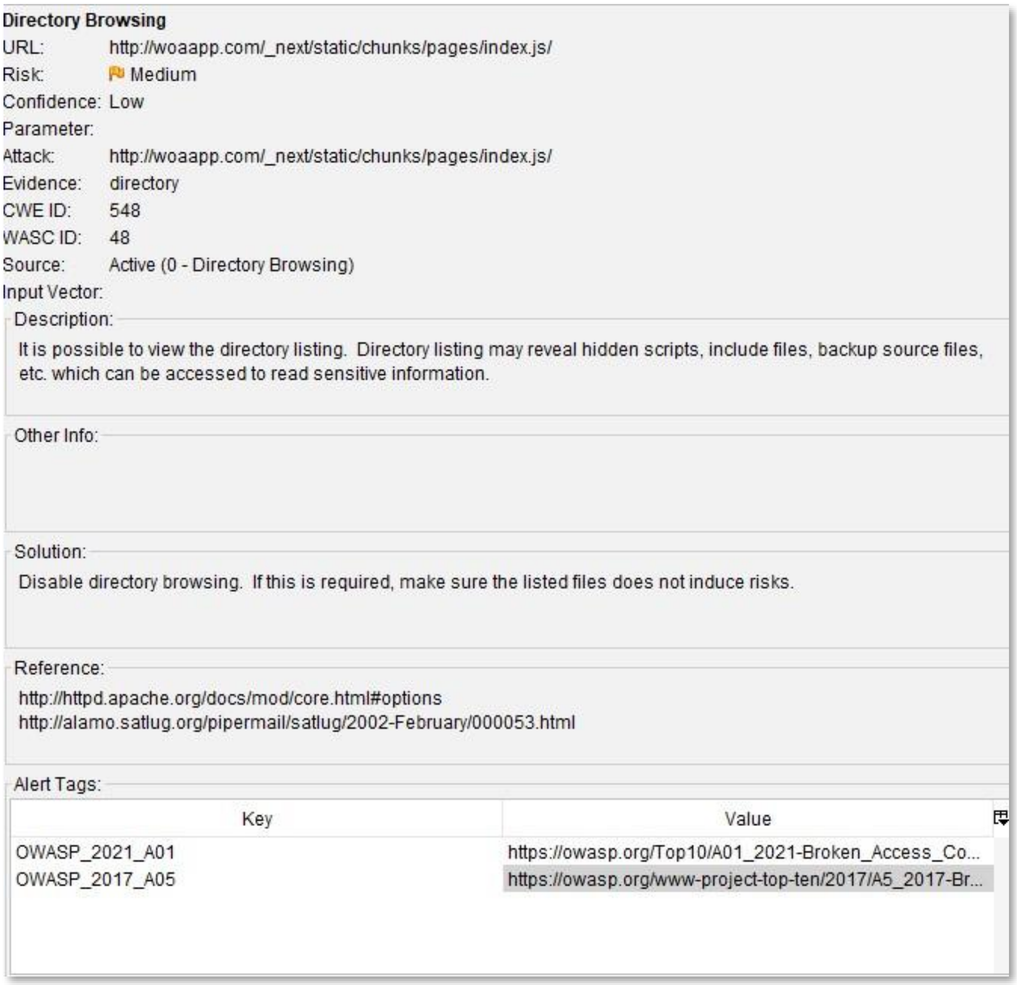


github-actions bot commented yesterday

- Site: http://woaapp.com
  New Alerts
  - **Absence of Anti-CSRF Tokens** [10202] total: 3:
    - http://woaapp.com
    - http://woaapp.com/
    - http://woaapp.com/?content=Zaproxy+alias+impedit+expedita+quisquam+pariatur+exercitationem.+Nemo+rerum+eveniet+dolores+rem+quia+dignissimos.
  - **Anti-CSRF Tokens Check** [20012] total: 3:
    - http://woaapp.com

The OWASP ZAP report after a scan. The test is considered a success despite the high amount of issues most of them are false. One issue that needs to be fixed in the future is the Anti-CSRF token. It would be nice to fix that issue in the future in order to make the cookies of the application more secure.

During another manual scan using OWASP ZAP I discovered a vulnerability that allowed for directory browsing on the client application:



Directory Browsing

URL: http://woaapp.com/_next/static/chunks/pages/index.js/
Risk: 🚩 Medium
Confidence: Low
Parameter:
Attack: http://woaapp.com/_next/static/chunks/pages/index.js/
Evidence: directory
CWE ID: 548
WASC ID: 48
Source: Active (0 - Directory Browsing)
Input Vector:

Description:
It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files, backup source files, etc. which can be accessed to read sensitive information.

Other Info:

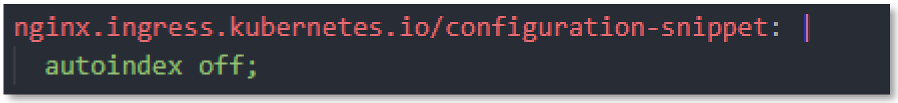
Solution:
Disable directory browsing. If this is required, make sure the listed files does not induce risks.

Reference:
http://httpd.apache.org/docs/mod/core.html#options
http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html

Alert Tags:

Key	Value
OWASP_2021_A01	https://owasp.org/Top10/A01_2021-Broken_Access_Co...
OWASP_2017_A05	https://owasp.org/www-project-top-ten/2017/A5_2017-Br...

After some research I fixed the vulnerability using the Ingress-Nginx setup in the Kubernetes manifests:



```
nginx.ingress.kubernetes.io/configuration-snippet: |
  autoindex off;
```

This snippet is an NGINX configuration directive that disables directory listing

Conclusion

After conducting the tests, it can be concluded that all experiments yielded positive results except for the Cypress test. Regrettably, the Cypress test was not performed due to time constraints.