Scenario tests for user story 3

Group 17

Scenario 1 : Installing the ZAP addon

1- Go to the project root Group17

2- Build the project using: *./gradlew clean spotlessApply build* - A ZAP addon (.zap) is created in PROJECT_ROOT/zap-addons/addOns/reportingproxy/build/zapAddOn/bin

3- Start the ZAP proxy program

4- Go to file, load addon and select the zap addon

5- Verify in the output tab that the addon has been successfully installed.

[6-] The addon can be uninstalled via the addon manager.

Scenario 2 : Loading rules

1- After installation of the ZAP addon, the rules can be loaded via going to the 'Tools/ Policy loader' on the menu bar.

2- An example jar file containing our rules is located in solutions/userstory3/rules.jar

3- A confirm or error message will be provided (should be either a success message or an error message if there is an exception.)

For scenario 3, 4, 5, 6 the browser should be set up to use the ZAP proxy at 127.0.0.1:8080.

Scenario 3 : Threshold rule

The threshold rule flags requests to the domain:localhost if they exceed 3 requests/3 secs.

1- For accurate and easier testing, a server will be created by running the server located at other/servers/servers.py:

python3 server.py 8081 0

A server on port 8081 will be created.

2- Going to http://localhost:8081 in the browser after setting the proxy will cause a single request to be formed. By refreshing the page, multiple requests can be made by incrementing steps of 1 request

3- Upon making 4 requests to http://localhost:8081 in under 3 seconds the “Rule_Threshold violated” alert is raised.

4- If the request threshold is not surpassed, then no alert will be raised

Scenario 4 : Common Headers rule

The common header rule flags responses which contain different headers than the previous 5 requests (for any domain).

1- For accurate and easier testing, two servers will be created with different headers each.

Go to other/servers and run the following commands in 2 different terminals:

python3 server_headers.py 8082 0

python3 server.py 8083 0

2 servers on port 8082 and 8083 will be created.

2 - Go to http://localhost:8082/ and refresh the page more than 5 times.

3 - Go to http://localhost:8083/

4 - Violation: “Rule_Common_Headers_Rule violated” alerts will be raised .

Scenario 5 : Hidden Input Field rule

The hidden input field rule flags hidden inputs with name password that are to be submitted to different domain from the one that responded

For accurate and easier testing, an html page containing a violating form has been created.

1 - Run the server using:

python3 server.py 8084 0

A server on port 8084 will be created.

2- Access localhost:8084 and click on hidden_scenario.html

3- An “Rule_Hidden_Field” alert should be shown since the page contains hidden- type input with name=password that submits to a different domain.

Scenario 6 : Request Performance rule

The request performance rule flags hosts which responses are on average slower than other hosts.

1 - Run the (fast) server using:

python3 server.py 8085 1

A server that responds after 1 second will be created.

2- For the rule to have an average threshold, go to http://127.0.0.1:8085 and refresh the page more than 10 times.

3 - Run the (slow) server using:

python3 server.py 8086 6

A server that responds after 6 second will be created.

4- Go to http://localhost:8086 and refresh the page more than 3 times. This will make the rule consider the localhost host slow and raise the “Rule_Request performance” alert.

! To reproduce the scenario the host for step 1 and 3 should indeed be different. In step 1 127.0.0.1 is used whereas localhost is used for step 3.

Scenario 7: Building a report

1- A report with alert’s details can be created by going to *Report/ Policy Violations Report*

2- Choose a destination directory to save the report.

3- Input a name for the report with the extension name of HTML “.html”, e.g “report.html”

4- Click save, an alert showing the success of the operation should be displayed.

5- Browse to the saved html report file and open it with a browser.