CSC 515: Spring 2023 – Workshop 7: OWASP Tools

This workshop will be composed of two parts:

a. Exploit a vulnerable web application by using the ZAP tool.
b. Examine the web application for vulnerable components

PART A: ZAP Tool

#	Task	Description
1	Install ZAP	Install the ZAP Attack Proxy tool
2	Using ZAP Automated Scan	Use an automated scan against the web application
3	Fuzzing with ZAP	Perform fuzzing with ZAP
4	On Your Own	Perform additional attacks and report on your progress

Install ZAP

Download and install OWASP's Attack Proxy tool: https://owasp.org/www-project-zap/

Using ZAP Automated Scan

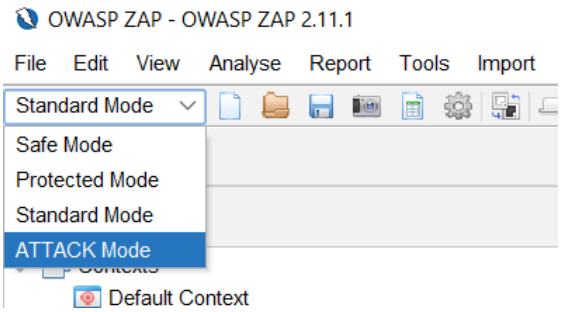
Warning!

You are only allowed to use ZAP to scan the OWASP Juice Shop web application. You should only use ZAP to attack an application you have permission to test with an active attack. Because this is a simulation that acts like a real attack, actual damage can be done to a site's functionality, data, etc.

Using ZAP to attack any other websites may result in academic integrity violations, university computer use violations, or even local/state/federal charges.

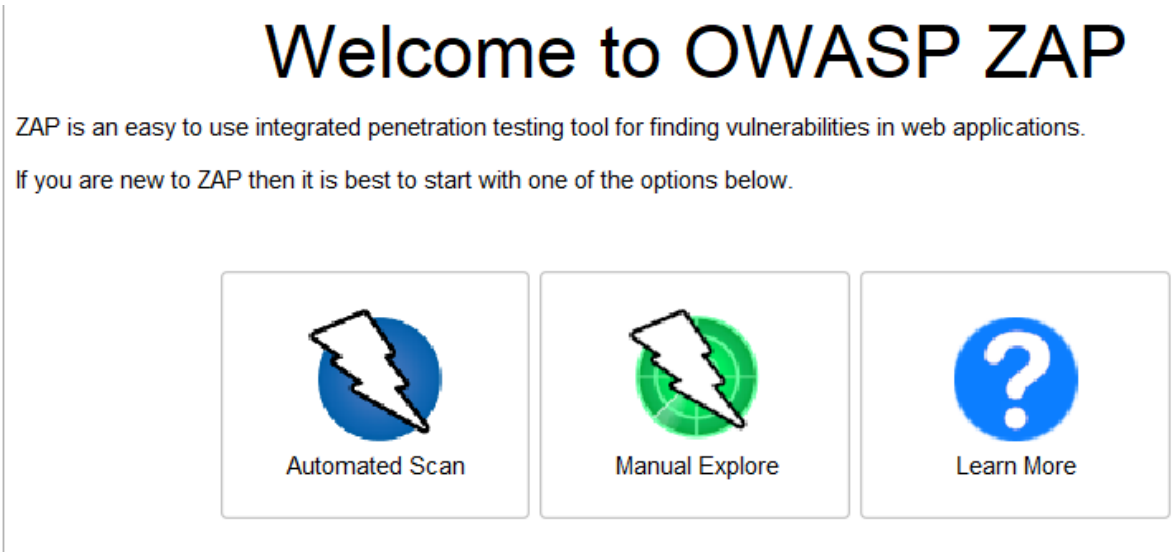
The instructor of this course does not have a get out of jail free card for you to use!

1. Open ZAP. In the top left corner of ZAP, change the "mode" to ATTACK mode. In the main window, select the button for an Automated Scan.



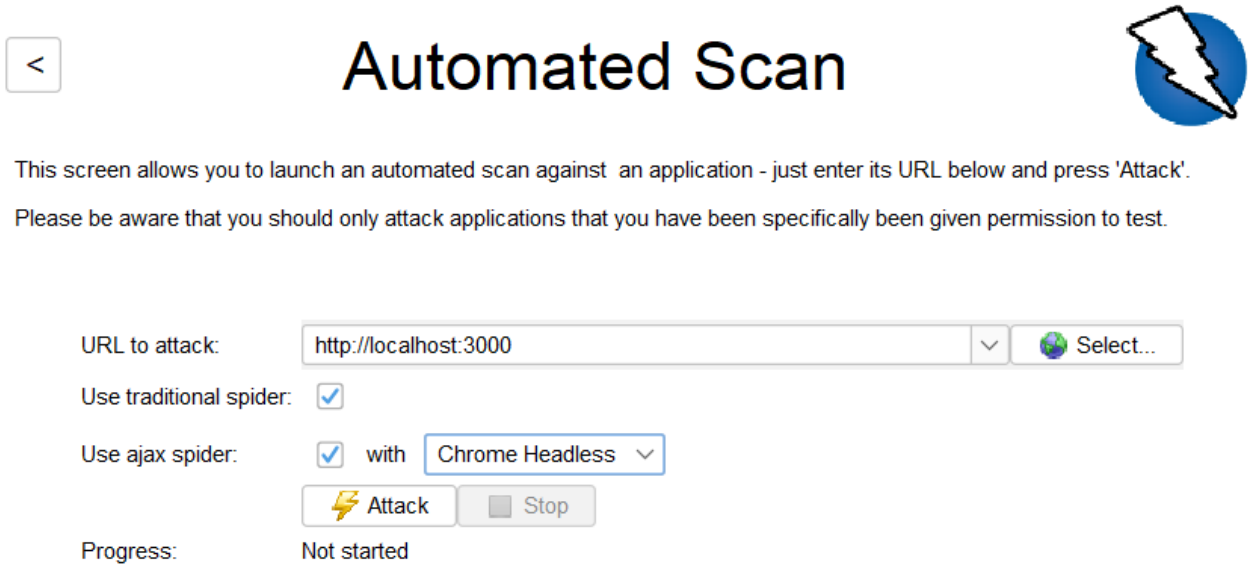
ATTACK Mode Dropdown

2. Click on the Automated Scan Icon in the center of the window.



Quick Start Icons

3. Enter the address to the OWASP Juice Shop web application. Select both traditional spider and ajax spider. Then click the attack button.

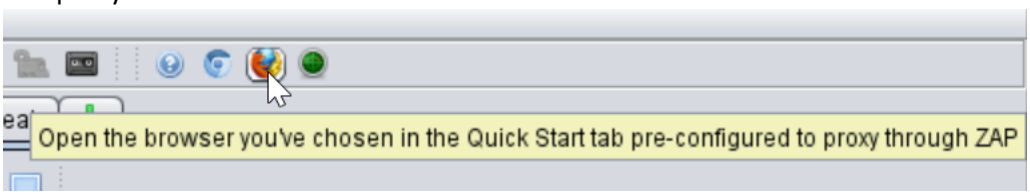


Automated Scan Dialog / Wizard

NOTE: it may take more than 1 hour for the scan to complete!

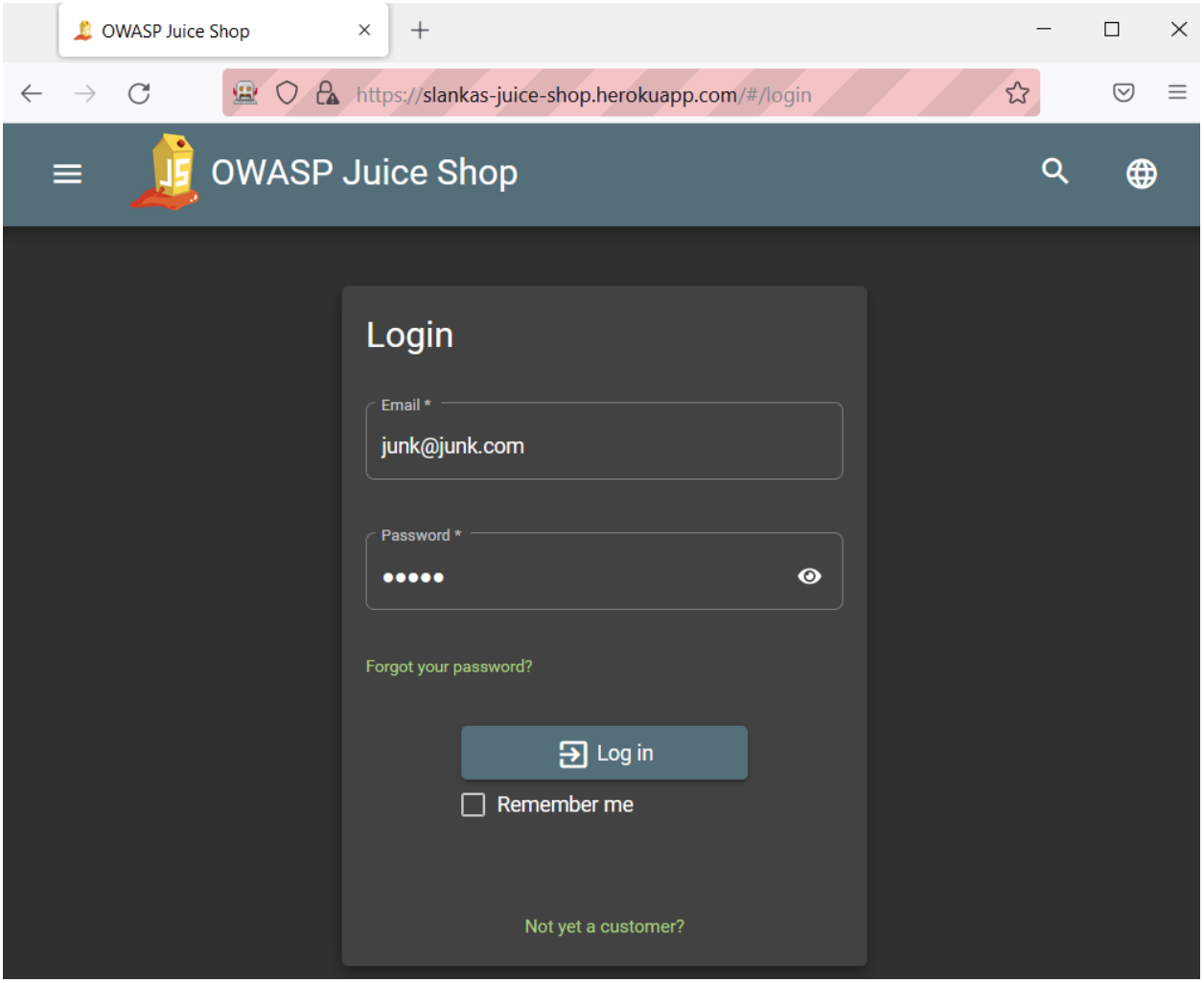
Fuzzing using ZAP

1. Open ZAP. Make sure you have either (A) configured a local proxy for ZAP to use (https://www.zaproxy.org/docs/desktop/start/proxies/) and updated your browser to use the proxy; or (B) in ZAP, use the "quick start" browser that is automatically configured to use the ZAP proxy.

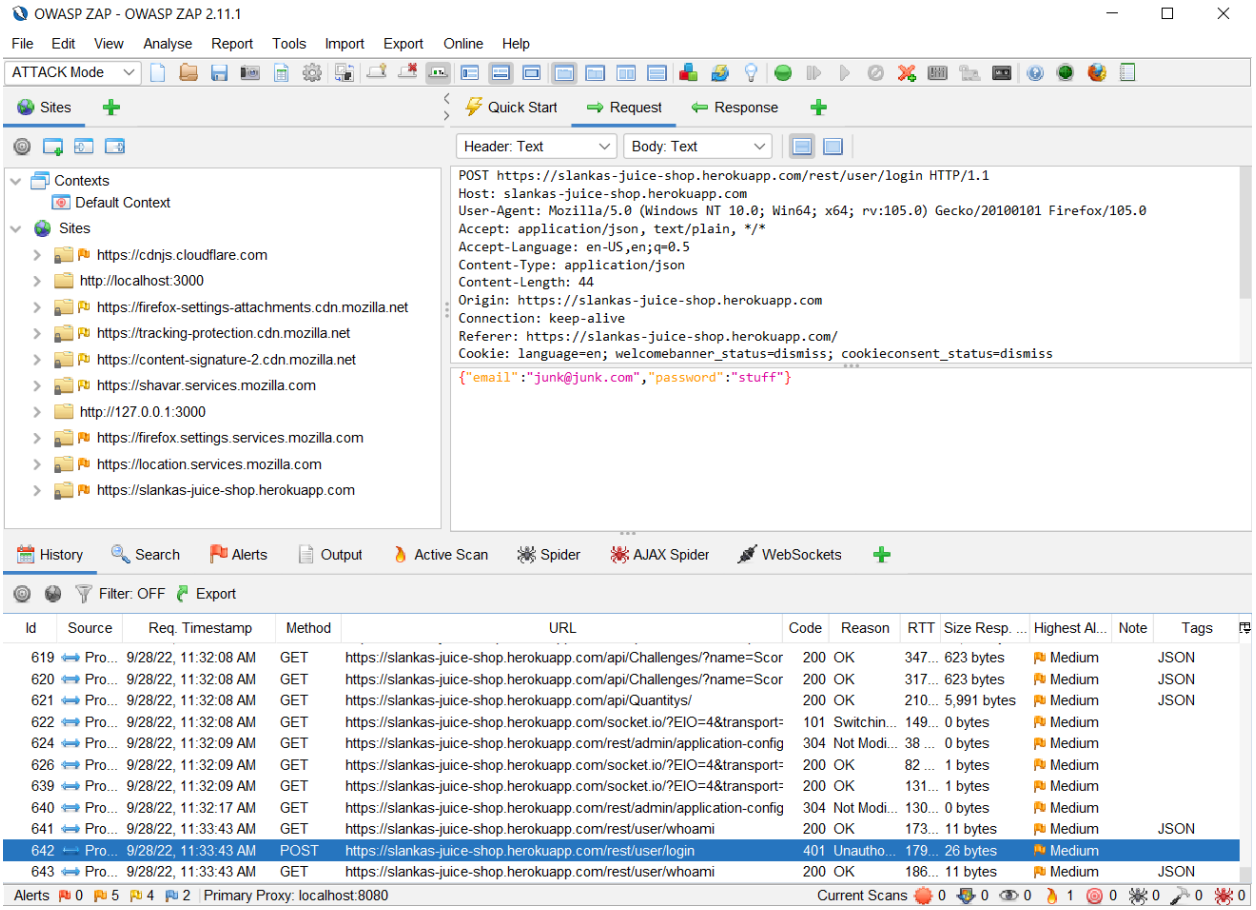


Quick Start Browser

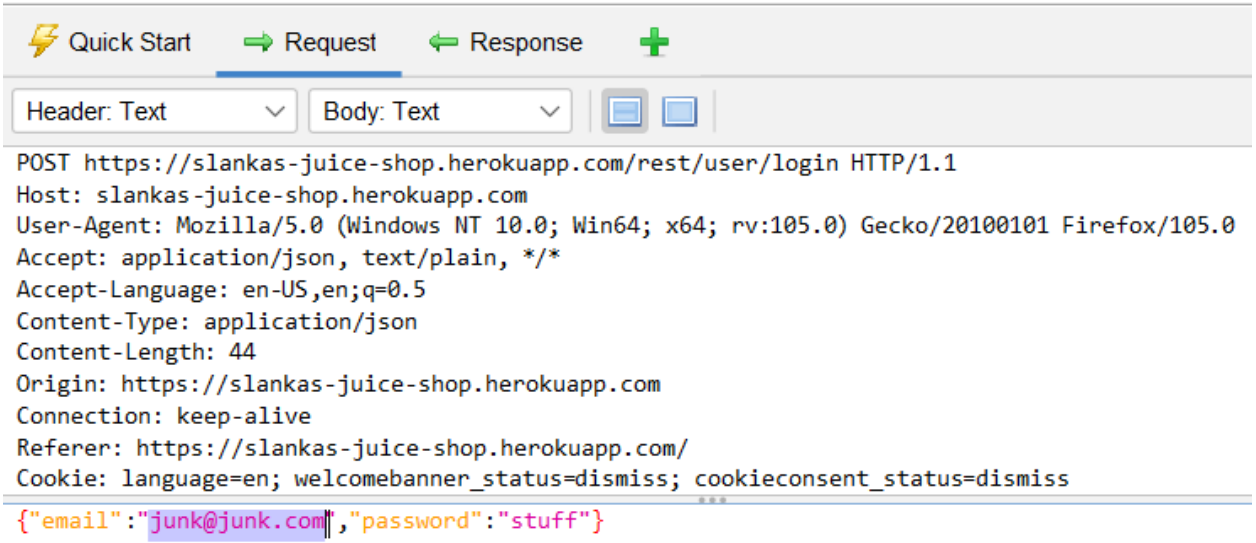
2. Open OWASP Juice Shop in the browser that is using the ZAP proxy. Attempt to login using a fake email and password, then switch back to the ZAP application.



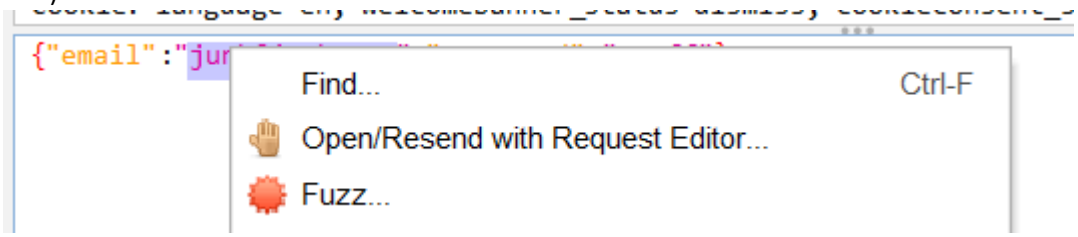
3. In the "History" tab, look for the login request. Click the login request.

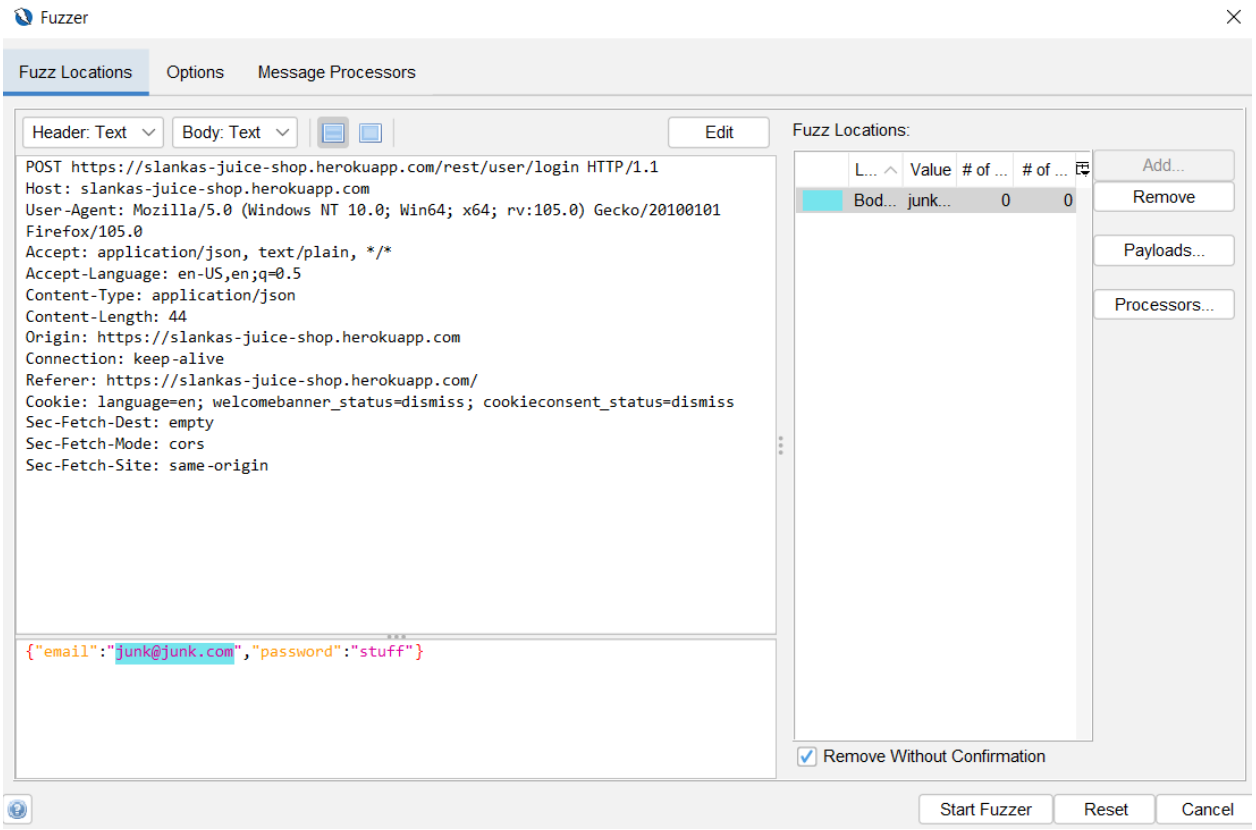


4. In the "Request" window, highlight the provided email address that you used when trying to login.

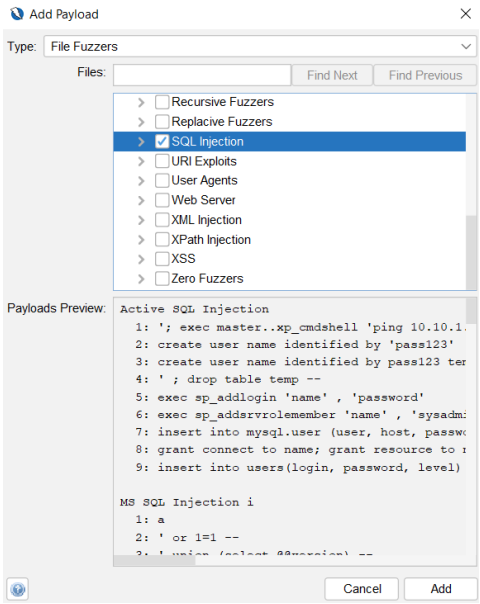


5. Right-click the highlighted text, and choose Fuzz from the right-click menu. The Fuzzer window should open (*note: if the fuzzer window does not automatically open, try closing and restarting ZAP).





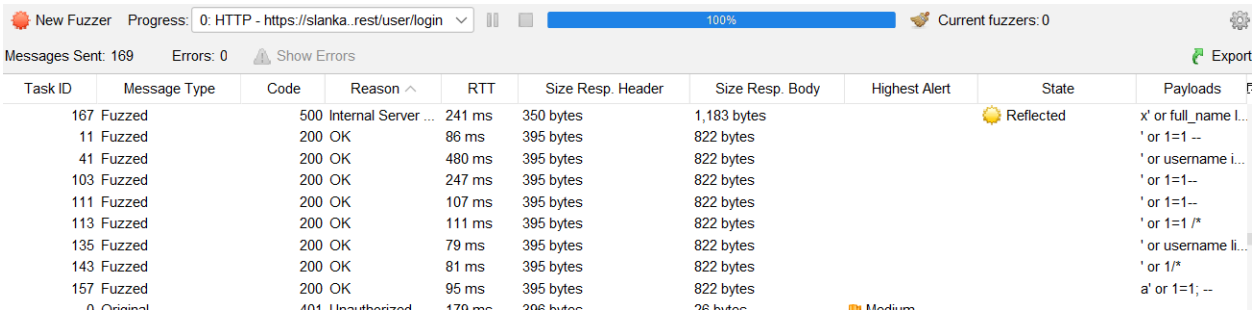
6. Click the Payloads button. Click the button to Add... a payload. In the “Payload type”, choose File Fuzzers -> jbrofuzz -> SQL Injection. This file contains sample SQL injection attack strings.



Click Add, then OK to return back to the Fuzzer window.

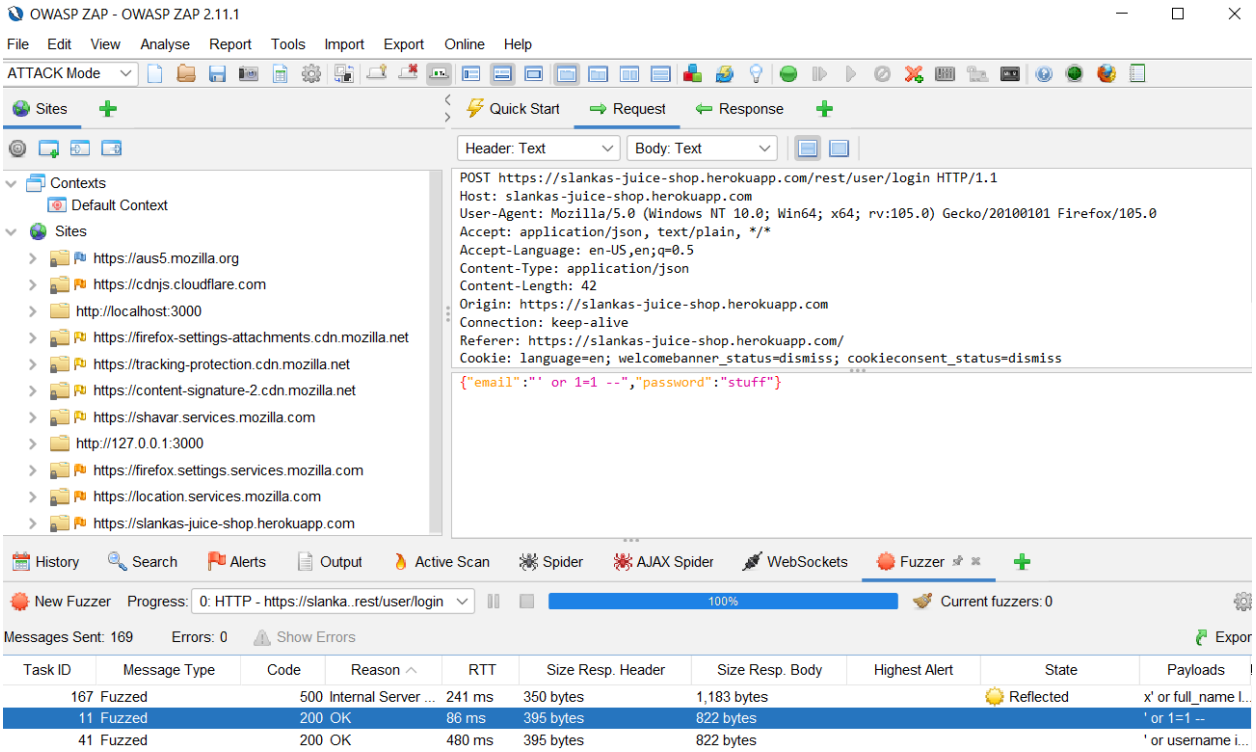
7. Click Start Fuzzer to begin the fuzzing process. The fuzzer will automatically try logging in by using the various SQL injection strings for the email parameter that we highlighted earlier.

8. When the fuzzing process completes, sort the results by the Reason column.



Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
167	Fuzzed	500	Internal Server ...	241 ms	350 bytes	1,183 bytes		Reflected	x' or full_name l...
11	Fuzzed	200	OK	86 ms	395 bytes	822 bytes			' or 1=1 --
41	Fuzzed	200	OK	480 ms	395 bytes	822 bytes			' or username i...
103	Fuzzed	200	OK	247 ms	395 bytes	822 bytes			' or 1=1--
111	Fuzzed	200	OK	107 ms	395 bytes	822 bytes			' or 1=1--
113	Fuzzed	200	OK	111 ms	395 bytes	822 bytes			' or 1=1 /*
135	Fuzzed	200	OK	79 ms	395 bytes	822 bytes			' or username li...
143	Fuzzed	200	OK	81 ms	395 bytes	822 bytes			' or 1/*
157	Fuzzed	200	OK	95 ms	395 bytes	822 bytes			a' or 1=1; --

9. Scroll through the results to find requests that returned a response of OK, which likely indicates the SQL injection attack worked. To double-check, you can click the request to see the payload that was sent with the request.



10. Using the example above, go back to the login page in your browser and enter the following string into the email field: **' or 1=1 –**
    Submit the form.

On Your Own

In Gradescope, answer the questions in the Workshop 7 activity. You will be asked to reflect on the output of the automated attack scan and fuzzing results.

PART B: Dependency Check

#	Task	Description
1	Install OWASP Dependency Checker	Install OWASP Dependency Checker
2	Use the Dependency Checker	Use the automated scanner against Juice Shop
3	On Your Own	Review the results from the report

Install OWASP Dependency Checker

Download and install a copy of the OWASP Dependency Check command-line tool:
https://owasp.org/www-project-dependency-check/

The application requires Java 11+ to be installed.

Once you have zip extracted, you can view the command-line arguments:

Unix/macOS:

```
dependency-check.sh --help
```

Windows:

```
dependency-check.bat --help
```

Use the Dependency Check Tool

Open a terminal window, change the directory to the bin directory of the dependency-checker installation directory, then execute the following command. Be sure to replace `path\to\JuiceShop` with the path to JuiceShop project codebase on your machine.

Unix/macOS:

```
dependency-check.sh --project "Juice Shop" --scan path\to\JuiceShop -o owaspJuiceShop
```

Windows:

```
dependency-check.bat --project " Juice Shop" --scan path\to\JuiceShop -o owaspJuiceShop
```

Command-line option notes:

- --project will set the project name to “WolfpackShop” in the output report
- --scan indicates the path to the project codebase you want to scan
- -o specifies an output directory that will contain the results of the scan; you can change this to whatever value you would like

Wait for the scan to complete. NOTE: the scan could take about 15 minutes to run. Once the scan finishes, go to the directory specified with -o (relative to your dependency-checker’s bin location) and open the report.

On Your Own

In Gradescope, answer the questions in the Workshop 7 activity. You will be asked questions about some of the vulnerabilities contained in the report.