Discovering Vulnerabilities With Nuclei

Author: https://github.com/small-goblin

This is a quick introduction to the tool nuclei. Nuclei is a DAST scanner, or dynamic application scanning tool. DAST is a complex topic; the TLDR is that it can enable researchers and testers to

- discover new vulnerabilities
- test for existing, known vulnerabilities (like CVEs)
- and provide a mechanism to make vulnerabilities re-testable to other technical teams

Nuclei really shines with it’s supportive community, scanning speed, and robust template engine.

You should

- already have some familiarity with Linux, and the terminal
- understand the gist of common vulnerabilities for web applications, like SQL injection

If not, check out:

- https://ubuntu.com/tutorials/command-line-for-beginners#1-overview
- https://www.w3schools.com/sql/sql_injection.asp

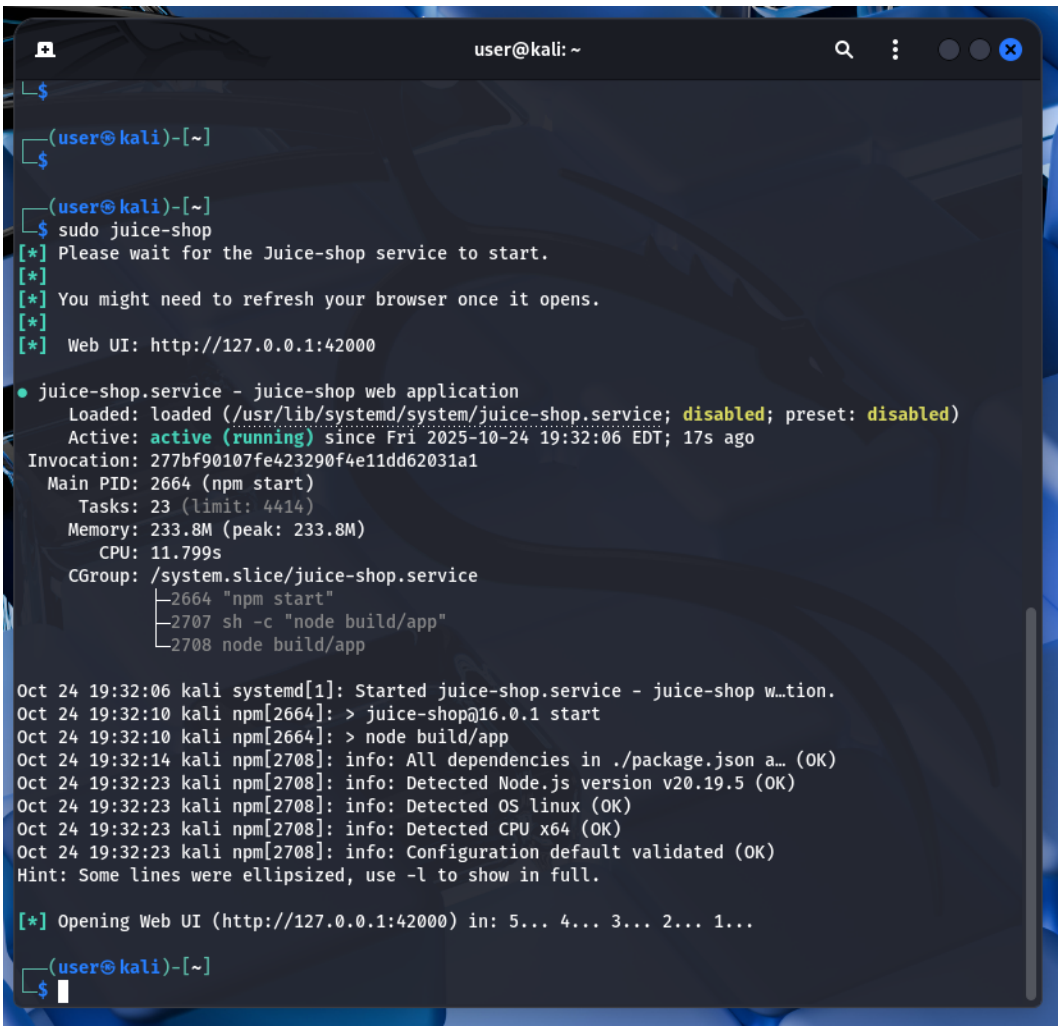
Lab

For much of this lab, you will be using the terminal.

The terminal is this thing, on the far left:

1. Start Juice Shop

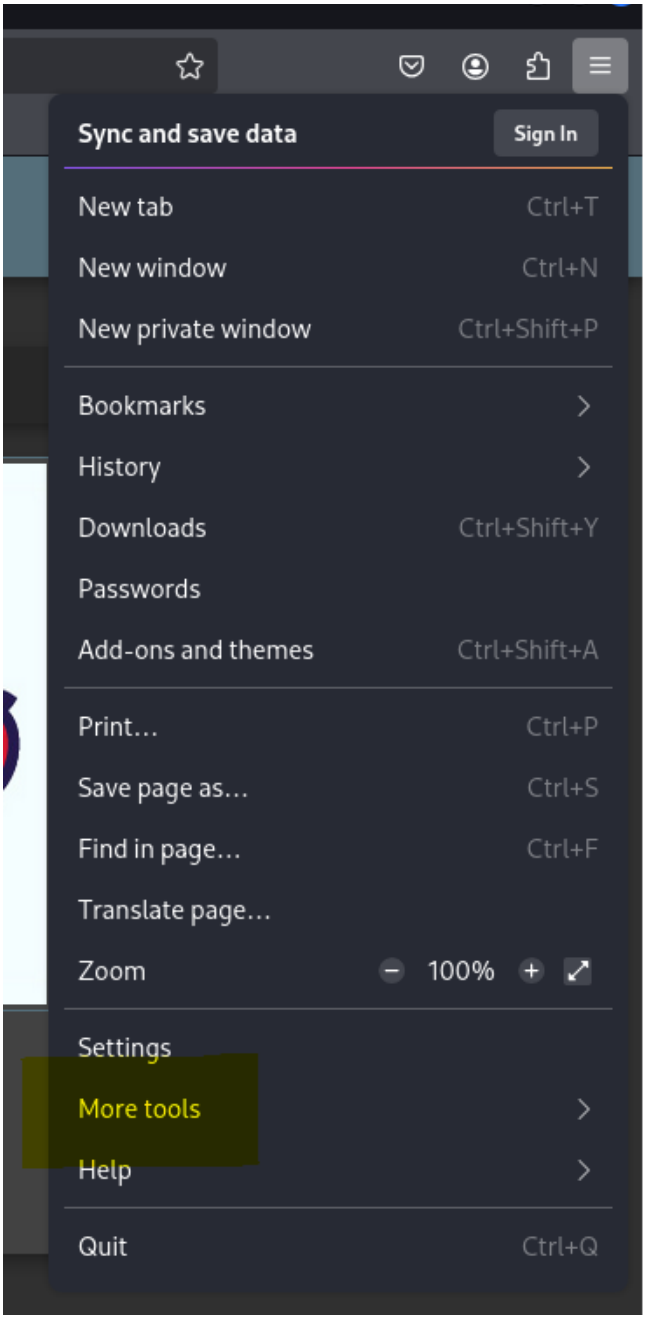
Terminal: sudo apt install juice-shop
Terminal: sudo juice-shop

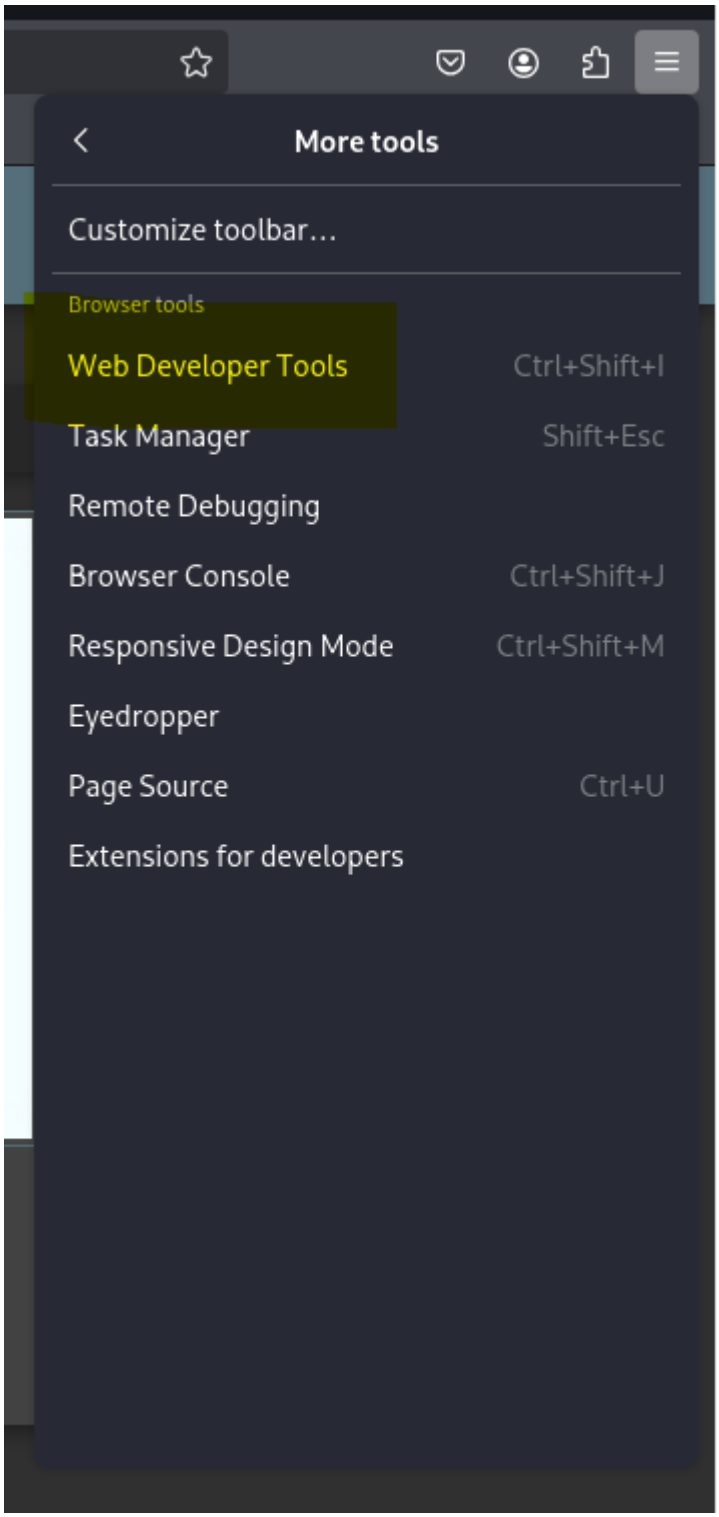


A web browser should automatically open.

If not, the default address is: http://127.0.0.1:4200

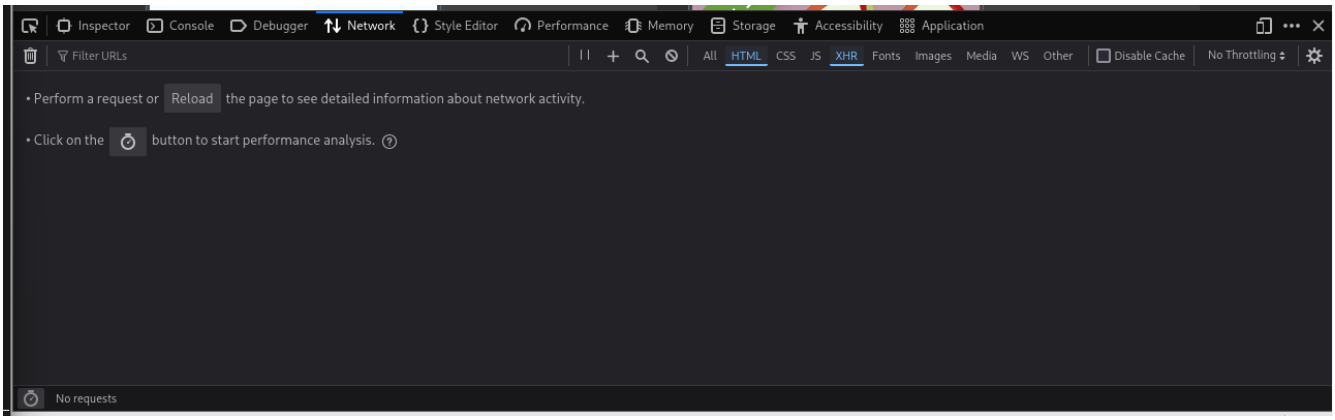
2. Open up Firefox Web Developer Tools.





Or Ctrl Shift I

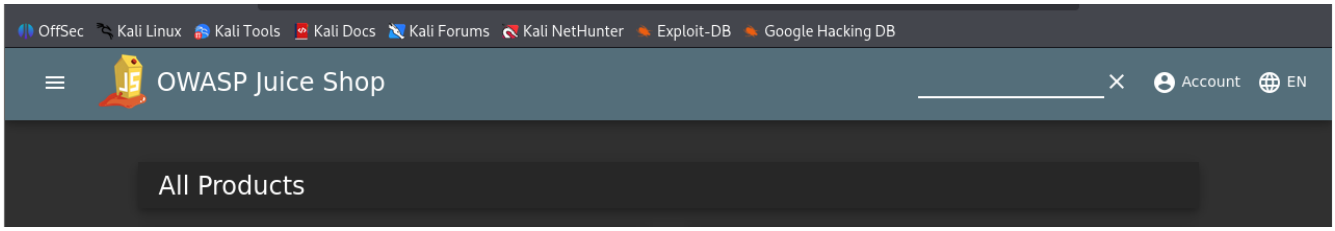
Select the “Network” tab



Before testing, normally, you would first explore an application and map out the entire API, cookies, parameters, etc.

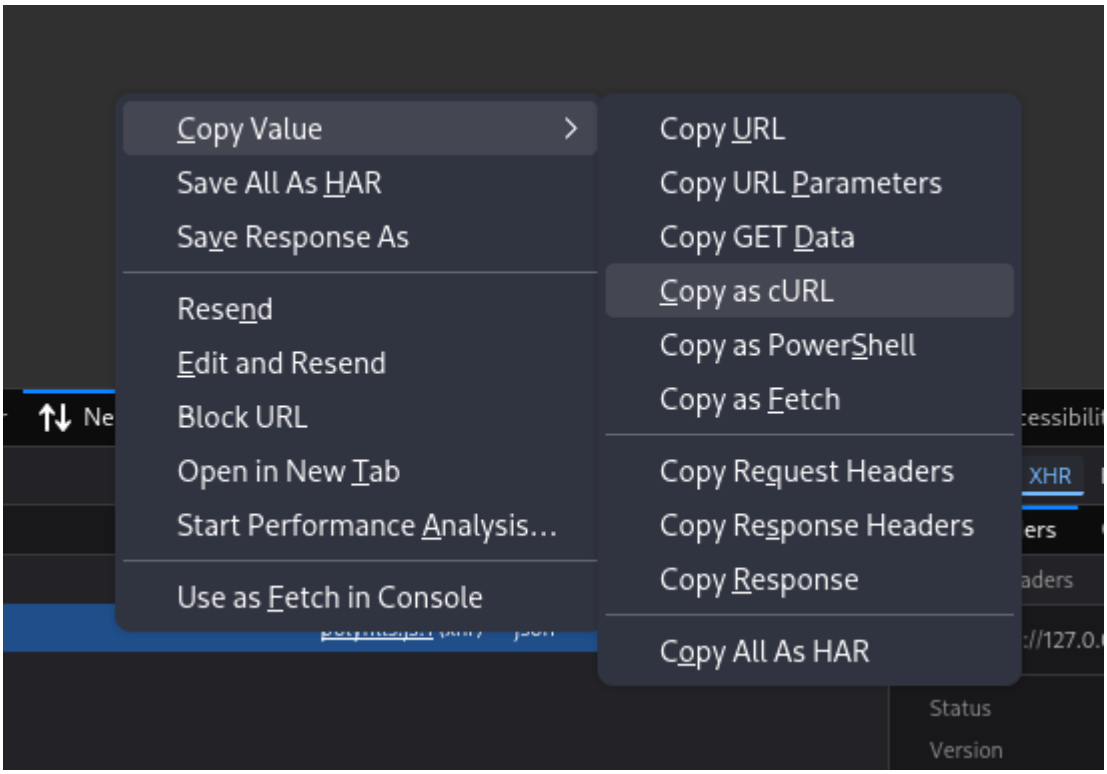
We’re going to keep an eye out for interesting traffic.

Lets search for products

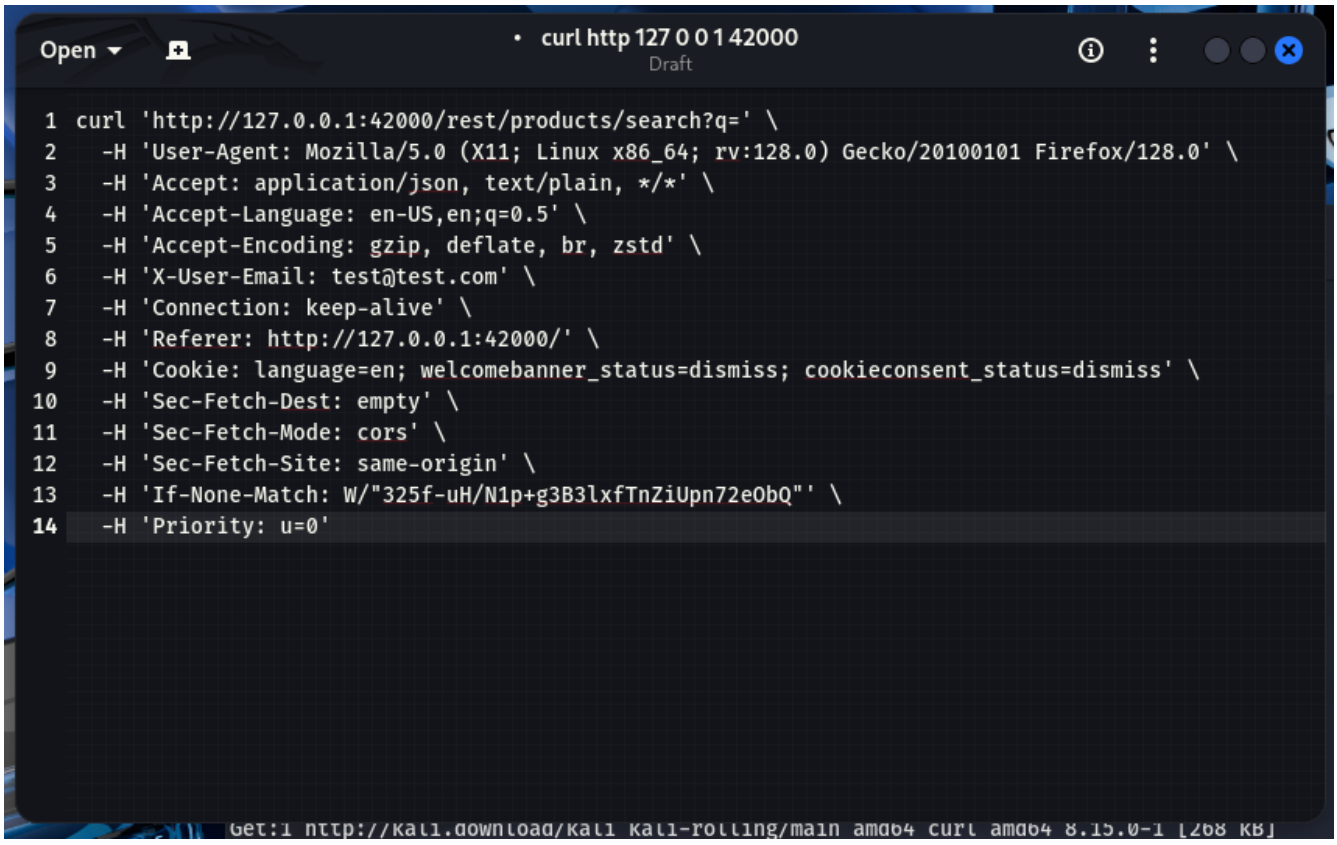


And explore the resulting request

Right click on the request, “copy as cURL”



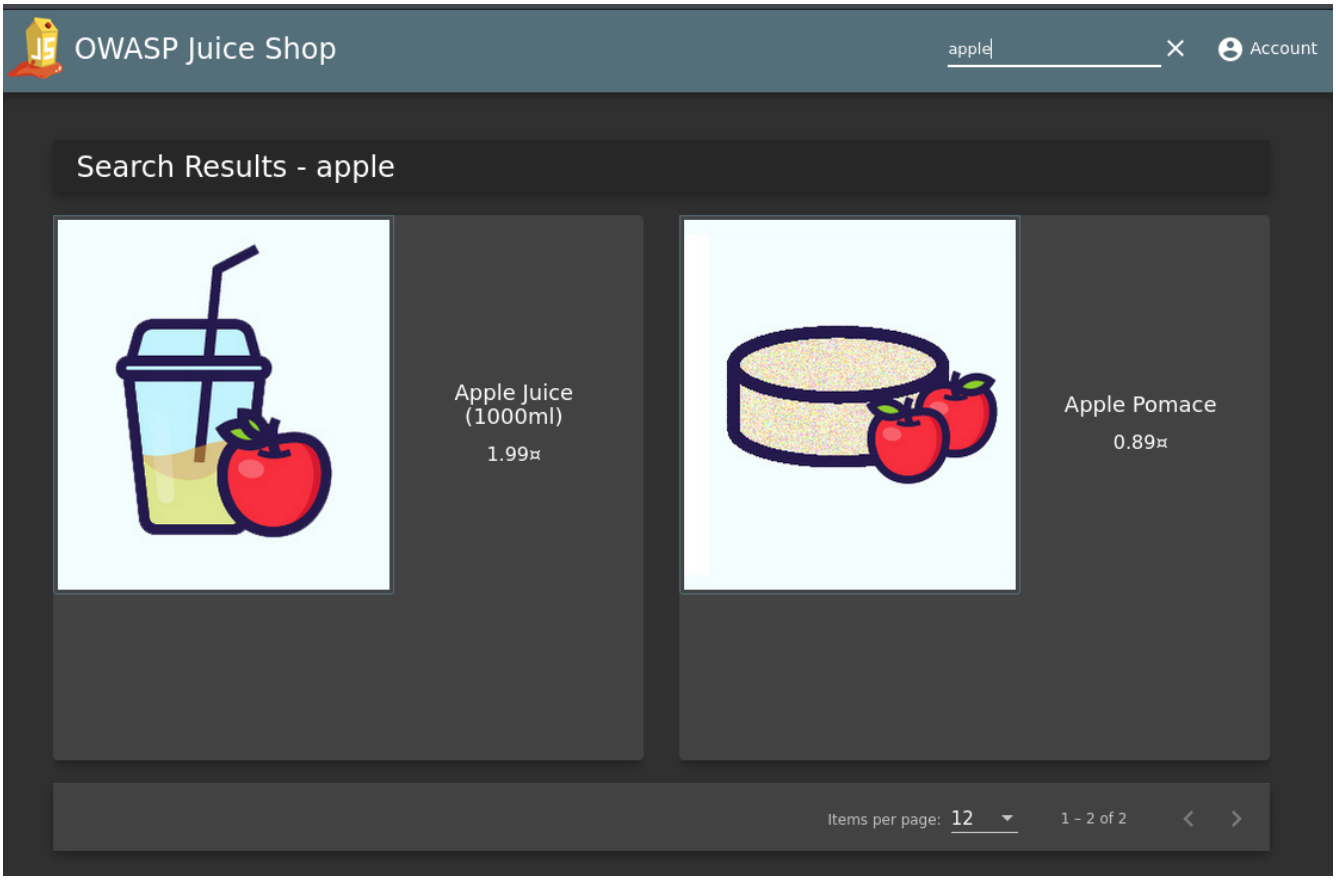
Paste in a Text Editor



If we were to run this in the terminal, it would return nothing. Likely, “search?q=” is a parameter for the term we are searching on, and searching on nothing returns nothing.

If we change this to “search?q=apple” we get data about products that match that search term.

Compare that to the data in the UI



We can consider this an “interesting” api call, and scan it using nuclei.

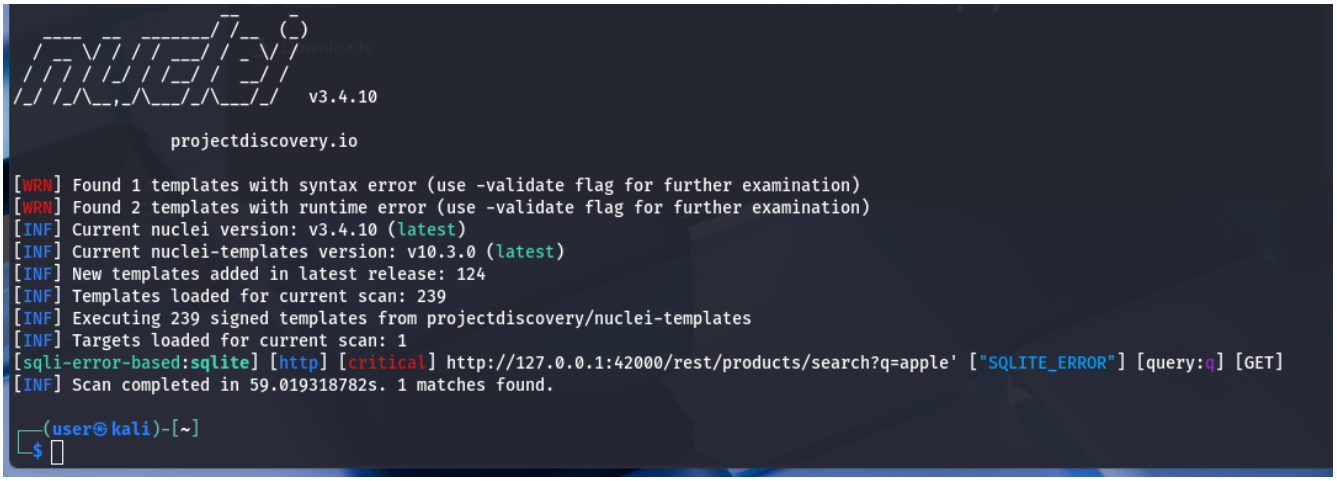
3. First update nuclei template(s)

nuclei -ut

The flags we need to scan a single api endpoint

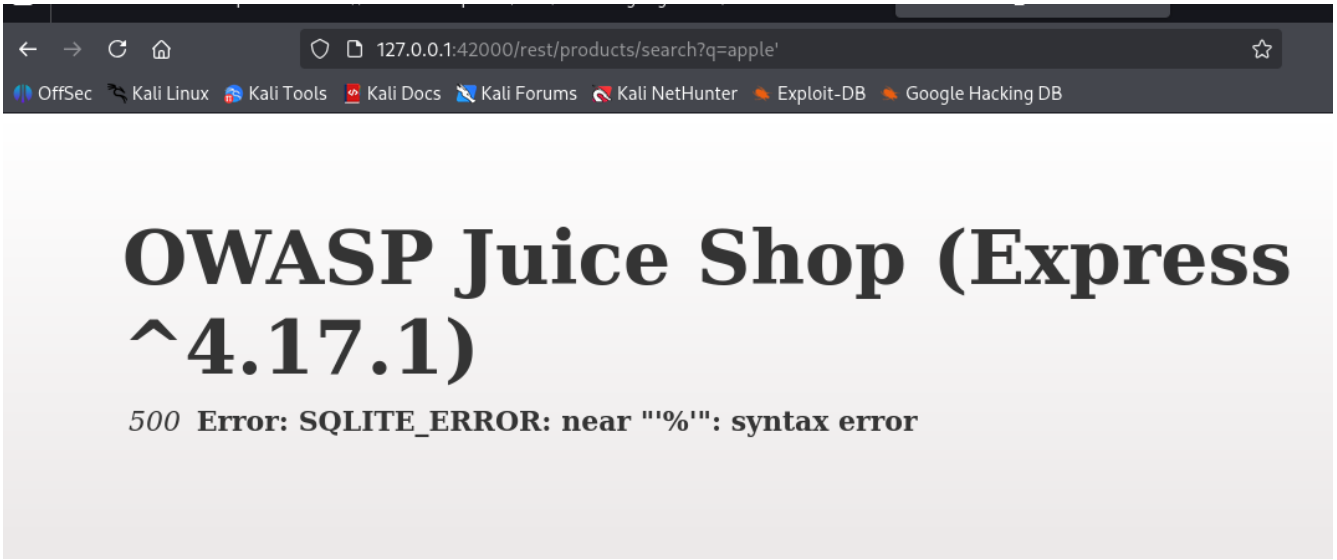
```
1 nuclei -dast \
2 -u http://127.0.0.1:42000/rest/products/search?q=apple \
3 -fa high
```

-dast to enable the dynamic scanner
-u to point to our request
-fa high to enable fuzzing aggression to “high”



Nuclei finds a sql injection vulnerability using the sqli-error-based template, and correctly identifies the database as being sqlite.

In the browser this request looks like this



Lets investigate the template nuclei is running. It’s actually very simple, compared to other templates.

The template is: error-based-sql-injection

https://raw.githubusercontent.com/geeknik/the-nuclei-templates/main/error-based-sql-injection.yaml

The template sends a single ‘ to our endpoint then triggers if a recognizable database error is returned. This is a common error based sqli test.

```
requests:
  - method: GET
    path:
      - "{{BaseURL}}/'"
```

In our case, it was likely “SQLITE_ERROR”

```
  - "Pdo[./_\\\\]Firebird"
  # SQLite
  - "SQLite/JDBCDriver"
  - "SQLite\\.Exception"
  - "(Microsoft|System)\\.Data\\.SQLite\\.SQLiteException"
  - "Warning.*?\\W(sqlite_|SQLite3::)"
  - "\\[SQLITE_ERROR\\]"
  - "SQLite error \\d+:"
  - "sqlite3.OperationalError:"
  - "SQLite3::SQLException"
  - "org\\.sqlite\\.JDBC"
  - "Pdo[./_\\\\]Sqlite"
  - "SQLiteException"
  # SAP MaxDB
```

The real strength is nuclei’s ability to runs 100s of templates like this one against 100s of known endpoints, incredibly fast. We’ve only tested one!

Read More:

- https://github.com/projectdiscovery/nuclei
- https://corgea.com/Learn/all-you-need-to-know-about-dast-in-2025-comprehensive-guide
- https://projectdiscovery.io/blog/nuclei-fuzzing-for-unknown-vulnerabilities