SQL Injection attacks on MOP website

By

Usman Tariq

S217034263

S217034263@deakin.edu.au

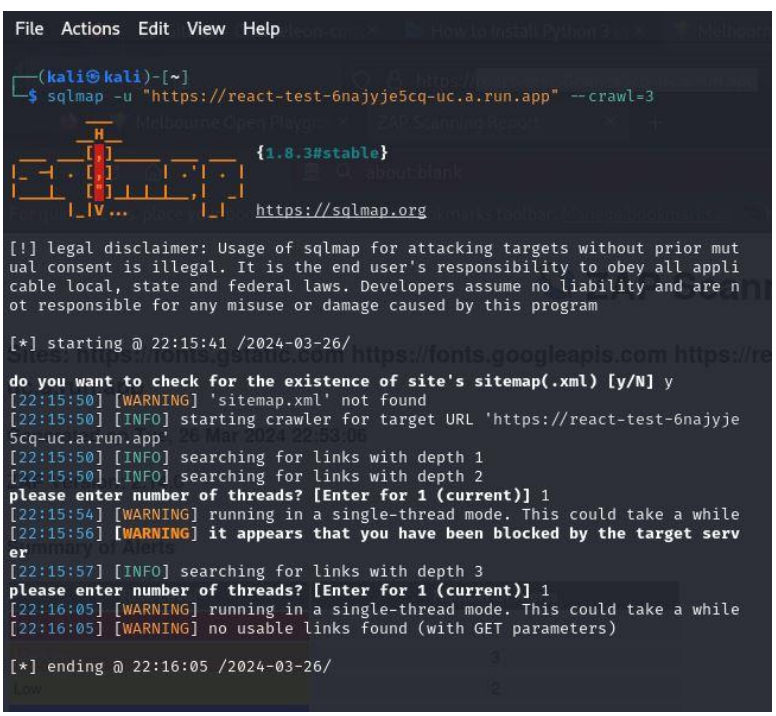
&

Rewniz Patell

Using SQL map

sqlmap -u "https://react-test-6najyje5cq-uc.a.run.app" --crawl=3

This above command will crawl the website for potential injectable points in the webapp



As we can see from the above screenshot. There were no injectable points found.

sqlmap -u "https://react-test-6najyje5cq-uc.a.run.app" -p x-frame-options

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "https://react-test-6najyje5cq-uc.a.run.app" -p x-frame-options
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.3#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program

[*] starting @ 22:57:37 /2024-03-26/

[22:57:37] [CRITICAL] all testable parameters you provided are not present wi
thin the given request data

[*] ending @ 22:57:37 /2024-03-26/
```

sqlmap -u "https://react-test-6najyje5cq-uc.a.run.app" -p x-content-type-options

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "https://react-test-6najyje5cq-uc.a.run.app" -p x-content-type-
options
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.3#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program

[*] starting @ 22:58:11 /2024-03-26/

[22:58:11] [CRITICAL] all testable parameters you provided are not present wi
thin the given request data

[*] ending @ 22:58:11 /2024-03-26/
```

sqlmap -u "https://react-test-6najyje5cq-uc.a.run.app" -p cache-control

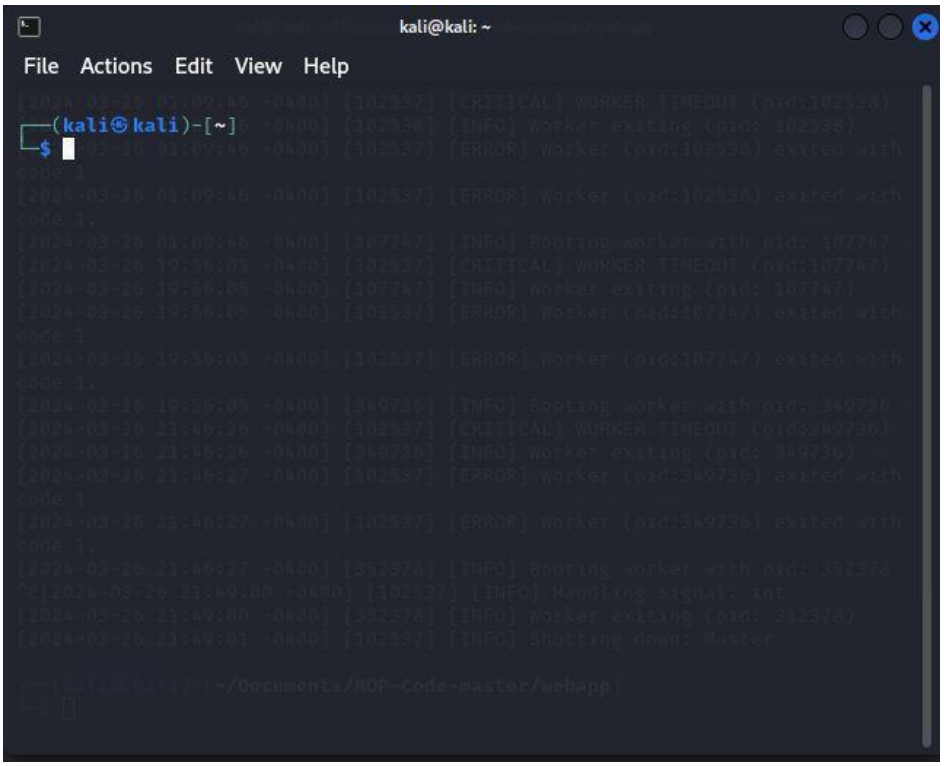
```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "https://react-test-6najyje5cq-uc.a.run.app" -p cache-control
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.3#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program

[*] starting @ 22:58:55 /2024-03-26/

[22:58:55] [INFO] testing connection to the target URL
[22:58:55] [INFO] testing if the target URL content is stable
[22:58:55] [INFO] target URL content is stable
[22:58:55] [INFO] testing if (custom) HEADER parameter 'Cache-Control' is dyn
amic
[22:58:56] [WARNING] (custom) HEADER parameter 'Cache-Control' does not appea
r to be dynamic
[22:58:56] [WARNING] heuristic (basic) test shows that (custom) HEADER parame
ter 'Cache-Control' might not be injectable
[22:58:56] [INFO] testing for SQL injection on (custom) HEADER parameter 'Cac
he-Control'
[22:58:56] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[22:58:58] [INFO] testing 'Boolean-based blind - Parameter replace (original
value)'
[22:58:58] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDE
R BY or GROUP BY clause (EXTRACTVALUE)'
[22:59:00] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING claus
e'
[22:59:01] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHER
E or HAVING clause (IN)'
[22:59:03] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (X
MLType)'
[22:59:04] [INFO] testing 'Generic inline queries'
[22:59:05] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[22:59:06] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comme
nt)'
[22:59:07] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE
- comment)'
[22:59:08] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
'
[22:59:10] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[22:59:11] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[22:59:13] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least
one other (potential) technique found. Do you want to reduce the number of re
quests? [Y/n] Y
[22:59:19] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[22:59:22] [WARNING] (custom) HEADER parameter 'Cache-Control' does not seem
```

We also tried testing a few parameters as we can see in 3 screenshots above. They were also not injectable so there was no sql injection vulnerability found.

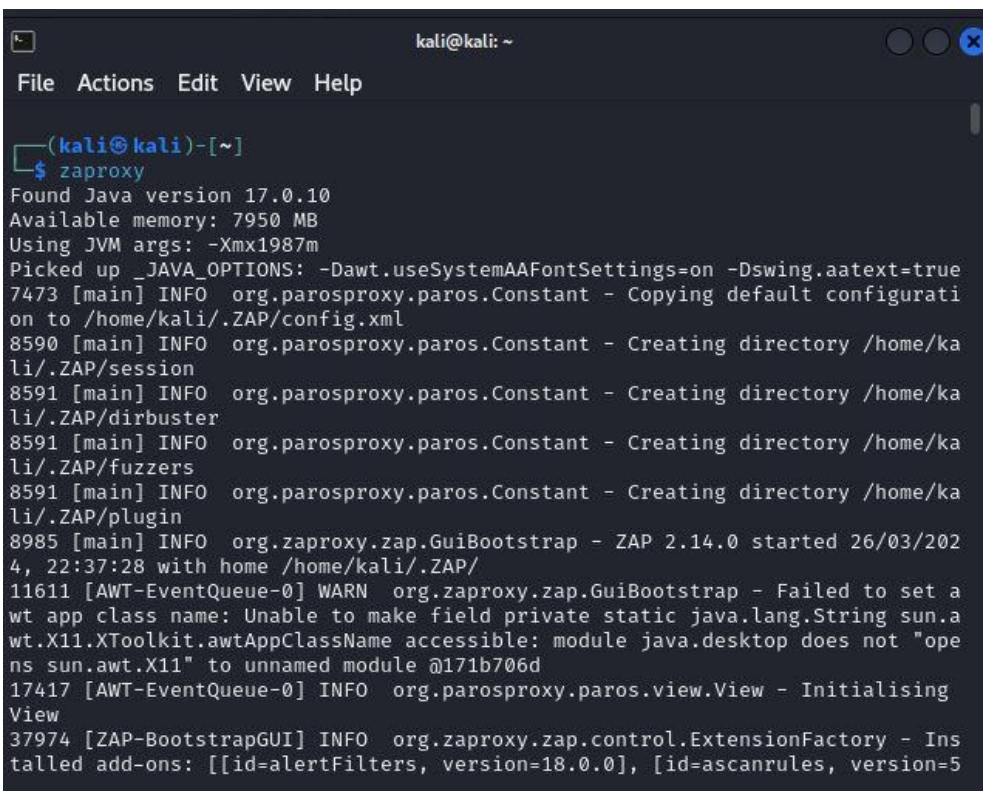
Testing using OWASP ZAP

As it is clear using 'sql map' we weren't not able to find any sql injection vulnerability. To confirm it we'll also try looking for sql injection vulnerability using 'OWASP ZAP'. Which is a great open-source tool.

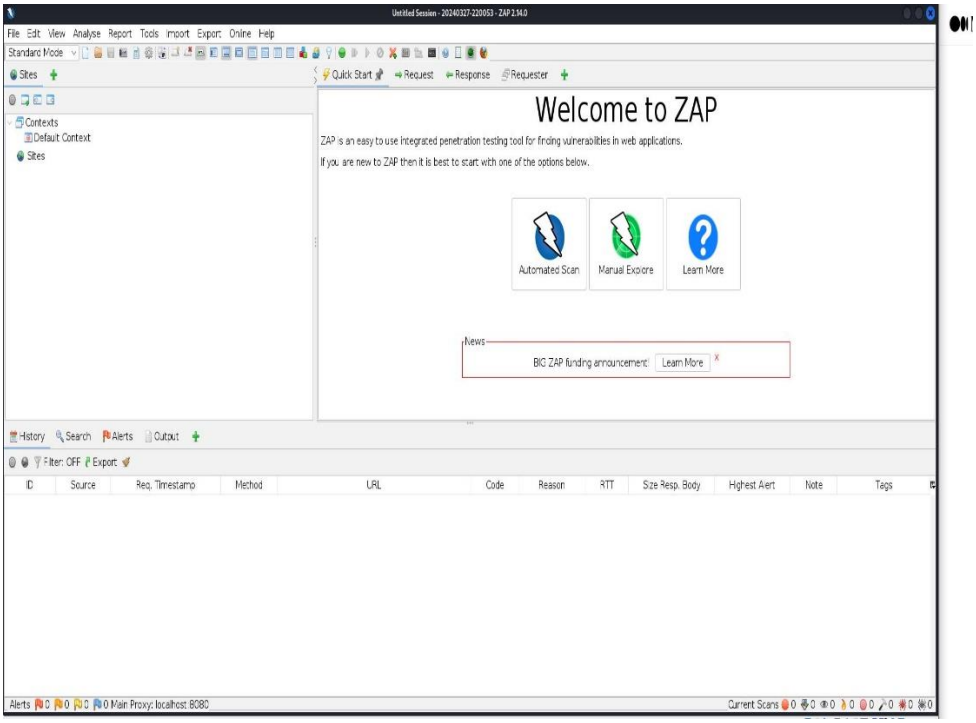
To use zap tool we need to launch it using the terminal. As shown in screenshot below



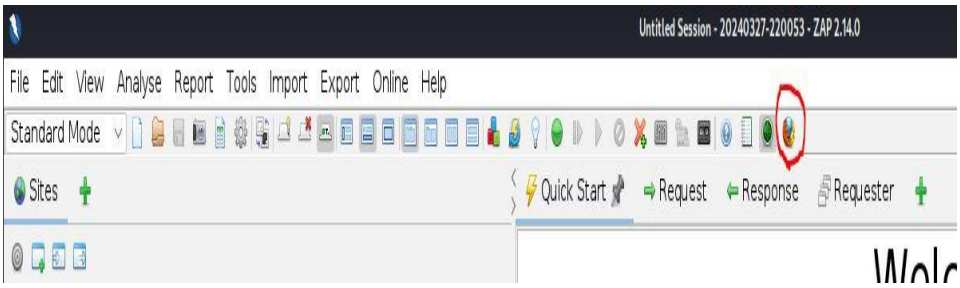
Now we need to type **'zaproxy'** in terminal to launch zap tool. As show screenshot below.



You'll be presented with the following screen.

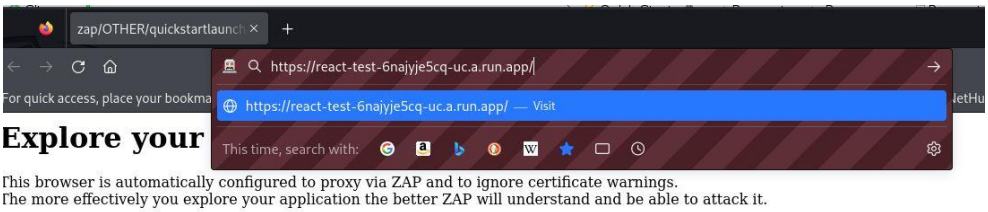


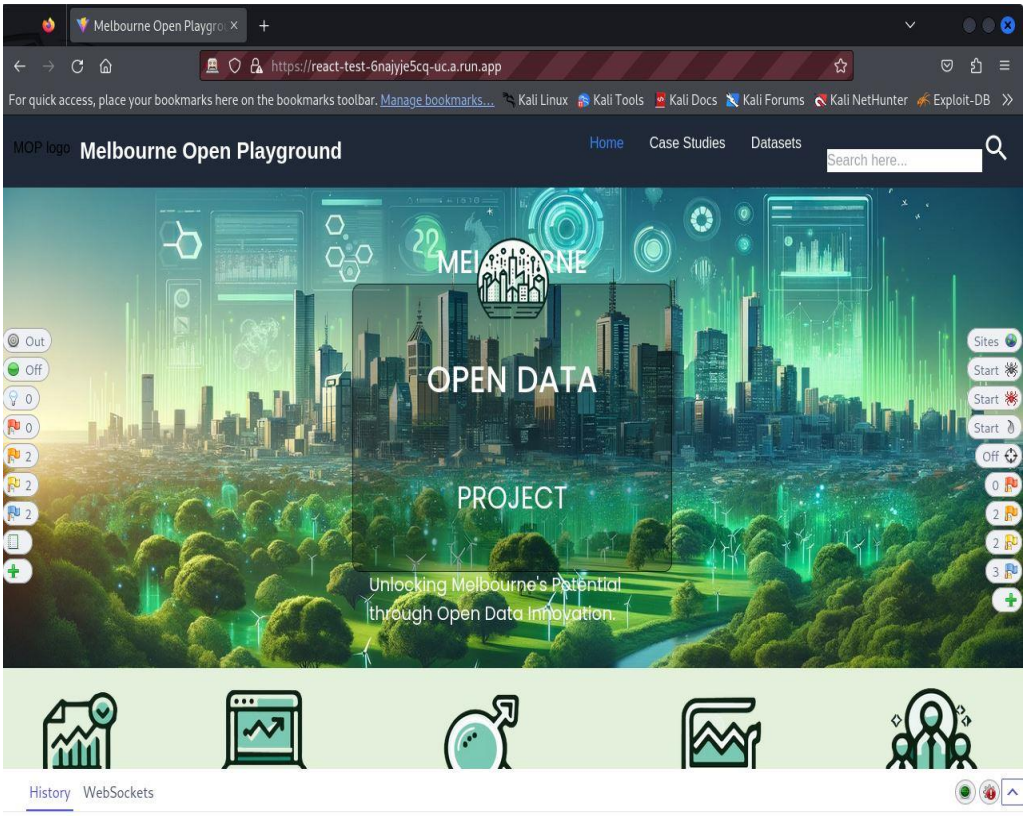
Now we need to click firefox icon at the top of the screen and launch firefox. Prior to this we need to make sure that firefox is installed in our kali vm.



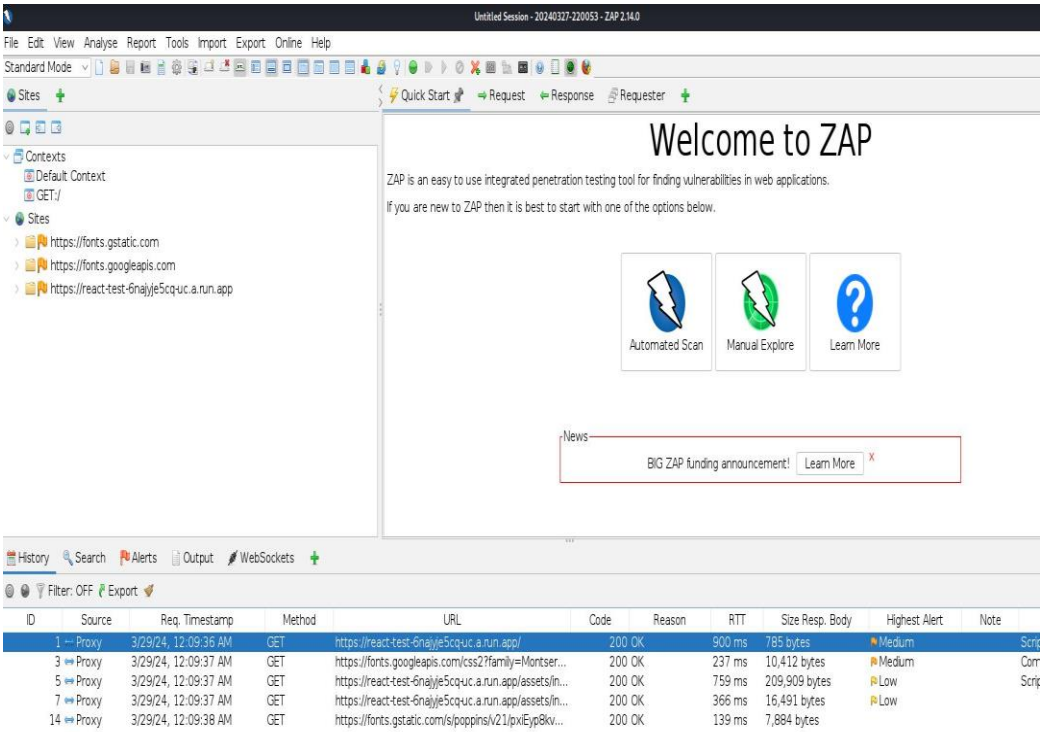
Firefox will launch as soon you click it. Now enter the following url in address bar.

https://react-test-6najyje5cq-uc.a.run.app/

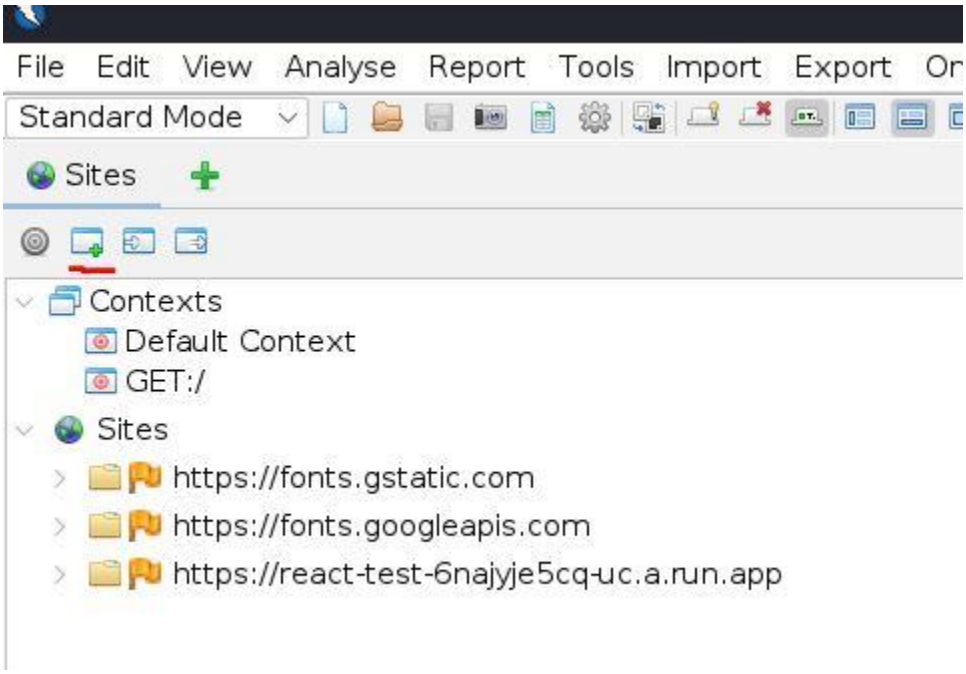




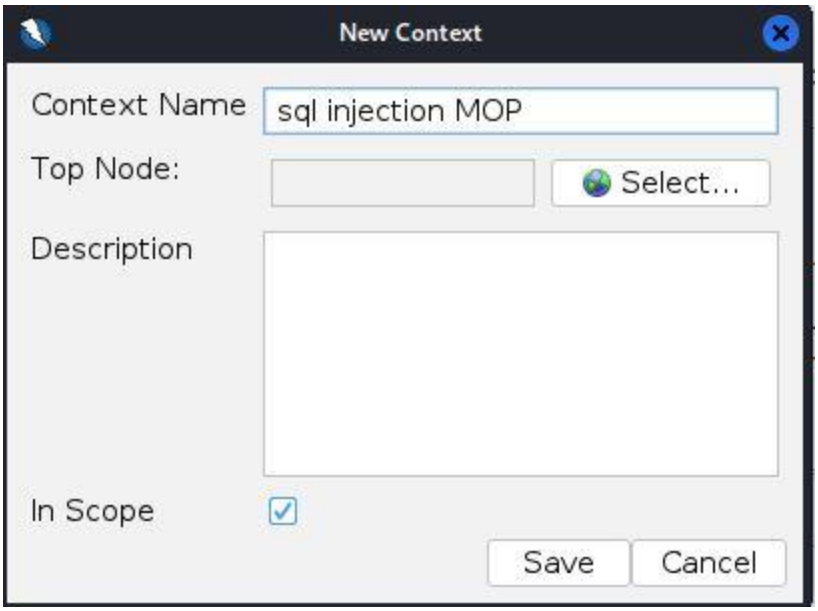
Now go back to zap GUI and you'll see the following screen



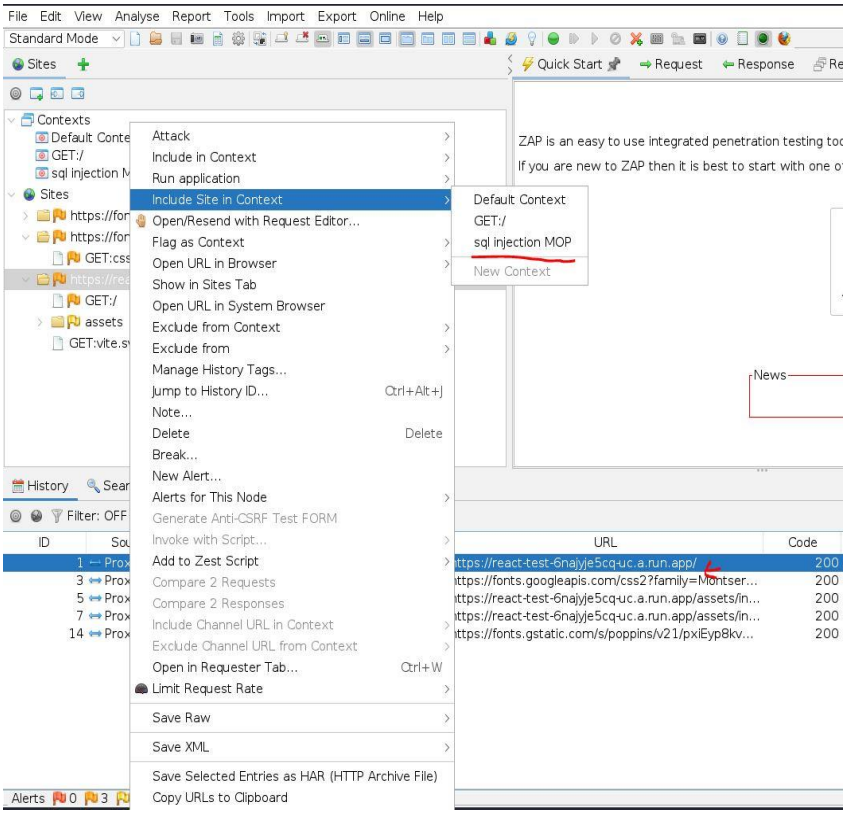
Now click on the new context icon at the top of the zap screen as shown in screenshot below. (Look for the icon which is underlined with red marker).



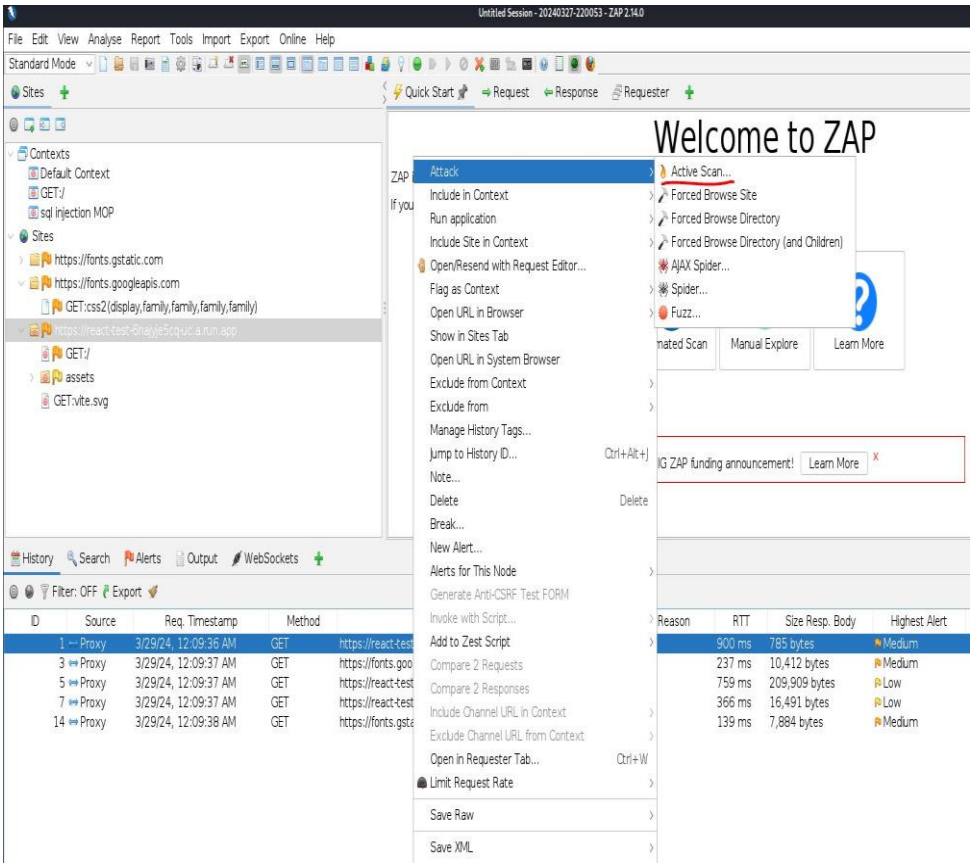
Now name it whatever you want and click save as shown in screenshot below



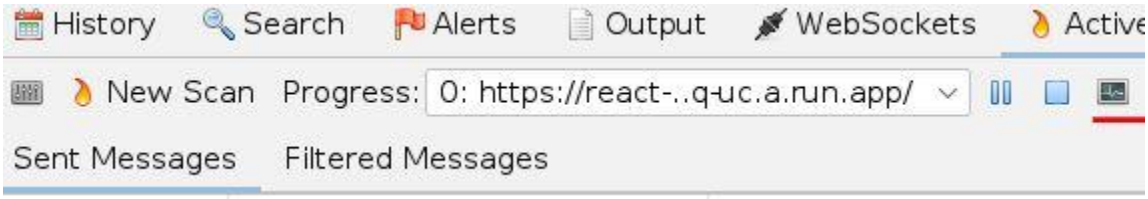
Now include the https://react-test-6najyje5cq-uc.a.run.app/ from the history tab at the lower part of the zap gui in the context we created.



Now right click again on the same from history portion of the zap gui and go to 'Attack' and run an 'Active scan'.



After scan is complete click on the icon next to stop icon and you'll be presented with the following report.



https://react-..q-uc.a.run.app/ Scan Progress

Progress | Response Chart

Host: https://react-test-6najyje5cq-uc.a.run.app

	Strength	Progress	Elapsed	Reqs	Alerts	Status
Analyser			00:00.935	2		
Plugin						
Path Traversal	Medium		00:27.005	135	0	✔
Remote File Inclusion	Medium		00:19.336	90	0	✔
Heartbleed OpenSSL Vulnerability	Medium		00:00.389	4	0	✔
Source Code Disclosure - /WEB-INF folder	Medium		00:00.618	3	0	✔
Source Code Disclosure - CVE-2012-1...	Medium		00:00.617	1	0	✔
Remote Code Execution - CVE-2012-1...	Medium		00:00.410	2	0	✔
External Redirect	Medium		00:15.528	81	0	✔
Server Side Include	Medium		00:07.288	36	0	✔
Cross Site Scripting (Reflected)	Medium		00:09.320	45	0	✔
Cross Site Scripting (Persistent) - Prime	Medium		00:01.695	9	0	✔
Cross Site Scripting (Persistent) - Spider	Medium		00:00.198	1	0	✔
Cross Site Scripting (Persistent)	Medium		00:00.019	0	0	✔
SQL Injection	Medium		00:47.268	234	0	✔
SQL Injection - MySQL	Medium		00:12.157	63	0	✔
SQL Injection - Hypersonic SQL	Medium		00:10.511	54	0	✔
SQL Injection - Oracle	Medium		00:10.923	54	0	✔
SQL Injection - PostgreSQL	Medium		00:11.471	54	0	✔
SQL Injection - SQLite	Medium		00:16.241	81	0	✔
Cross Site Scripting (DOM Based)	Medium		00:18.785	54	0	✔
SQL Injection - MsSQL	Medium		00:01.970	14	0	✔
Log4Shell	Medium		00:00.001	0	0	⊘
Spring4Shell	Medium		00:00.421	2	0	✔
Server Side Code Injection	Medium		00:13.891	72	0	✔
Remote OS Command Injection	Medium		01:03.653	315	0	✔
XPath Injection	Medium		00:05.896	27	0	✔
XML External Entity Attack	Medium		00:00.001	0	0	✔
Generic Padding Oracle	Medium		00:00.025	0	0	✔
Cloud Metadata Potentially Exposed	Medium		00:00.489	4	0	✔
Server Side Template Injection	Medium		00:24.520	126	0	✔
Server Side Template Injection (Blind)	Medium		00:22.053	108	0	✔
Directory Browsing	Medium		00:00.212	1	0	✔
Buffer Overflow	Medium		00:01.781	9	0	✔

Copy to Clipboard | Close

As it is clear from above report this site is not injectable. So after using 2 tools it’s concluded there no sql injection vulnerability present.