Using OWASP ZAP to find Vulnerabilities.

Using OWASP can bring ideas on how to find exploits and running. Below I will be showing how to Use the spider within OWASP to detect vulnerabilities and what they mean:

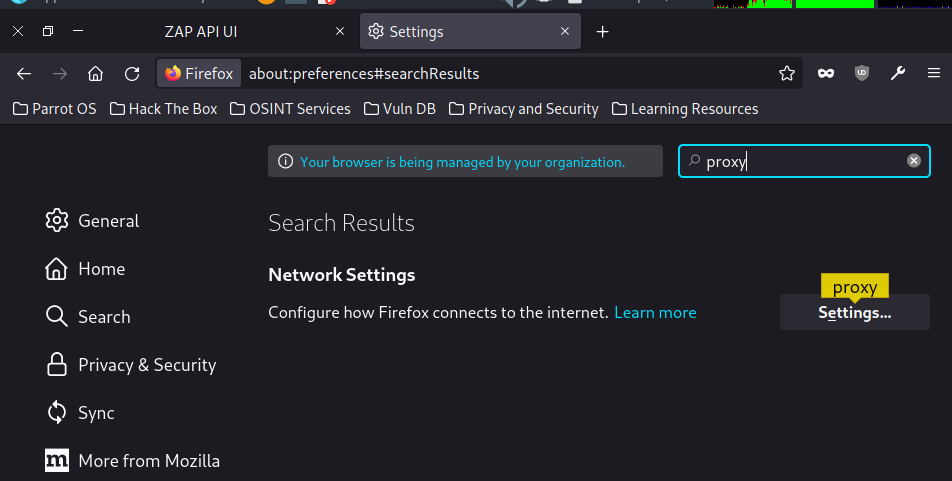
# Setting up:

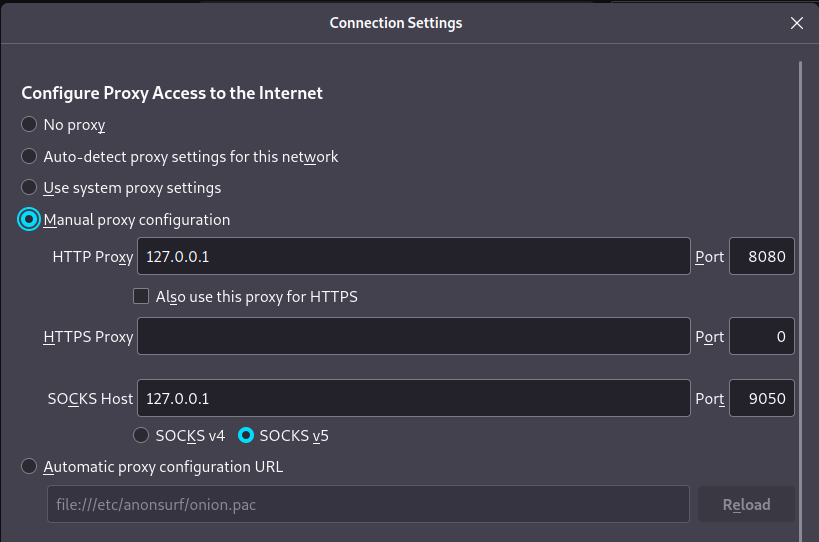
Since I was running ParrotOS no installation was needed as it comes included, however other versions of Linux would use the commands:

sudo apt-get update

sudo apt-get install zaproxy

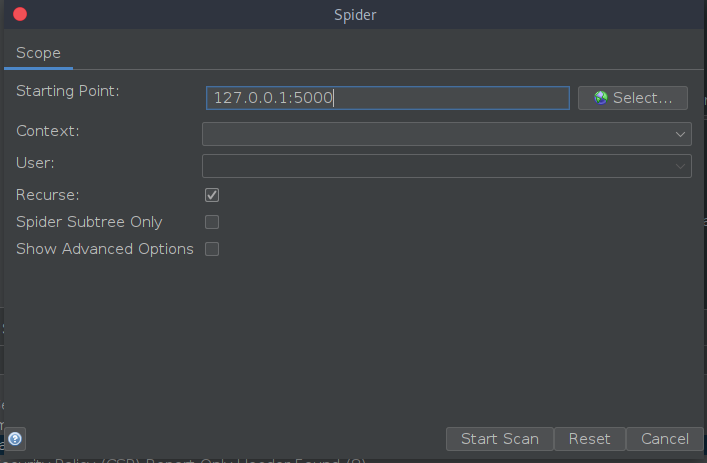
After the installation is complete you need to ensure that your browser settings have to have ZAP running as it should (in this case on port 8080)





OWASP does have any many tools and I will be using the others to further find vulnerabilities however for this one I am using the spider tool (found in the tools tab)

This is how it should look in the spider app, there is no need to add the other pages, as it will find every other page within the webapp and test them for vulnerabilities



Below are the results of the scan and the possible vulnerabilities/flags that have been set off by the spider crawling across the webapp.

Text

Description automatically generated

# Summary of Results:

CSP Header Not Set: - Self explanatory The header for CSP isn’t set which could lead to Cross-site Scripting attacks

Cross-Domain JavaScript Source File Inclusion: Javascript files are coming from a different domain which can allow an attacker to execute malicious code on the web app

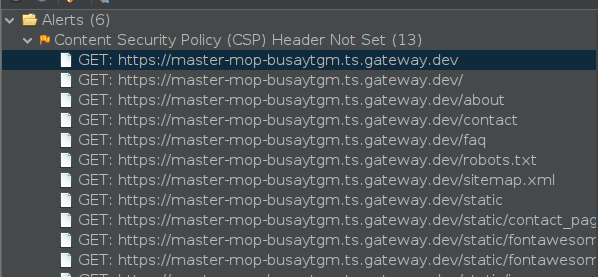
Server leaks Version Information: This shows that within the HTTP response header field the web server is leaking information about the version, this is giving attackers more information to prepare for an attack

Content Security Policy -Report Only Header Found : This shows that within the header it will not enforce any violation but will report it when there is a violation.

Information Disclosure: This warning is showing that there are some comments that may be giving away sensitive information.

Modern Web Application: Using new technologies isn’t a bad thing but the newer technologies can be prone to newer attacks especially if there aren’t proper procedures in place.

After running these tests on the actual live website I was able to get these alerts



Text

Description automatically generated

In normal cases PII would be an immediate fix however since this website is using location co-ordinates it was picked up as a vulnerability.

However with the vulnerable JS library that can be fixed by simply updating the JS library to the latest version

# Suggestions:

Enforcing CSP headers

Validating Input and output – this can be done by changing web server settings

Modify the Server HTTP Response header field in order to prevent data leakage- this can be done by changing server configuration settings or by using server sided scripting.

Changing the CSP to ensure that it isn’t just set to report only.

Since the application is using modern technologies, it should also be using modern security measures to ensure that everything remains secure.

Reviewing source code of the page and make customer error pages and some mechanism to give a unique error reference/identifier to the client while logging the details on the server side.

Updating JS library to the latest version.