**Penetration & Vulnerability Testing**

In the following I make use of tools such as the ZAP to test and explore the implemented vulnerabilities in my project. Demonstrating how the vulnerabilities can be detected using ZAP and how it is exploited. Below are step-by-step screenshots describing how I tested for vulnerabilities.

***The following is a showcase brute force techniques to login into an unsecure account.***

***For this test an account with a username “admin” and password “123” was created.***

* A screenshot of a computer

  Description automatically generatedA screenshot of a computer program

  Description automatically generatedGoing to options to generate the CA certificate.
* Importing the certificate for the desired web browser and selecting the local host pertaining to the project

A screenshot of a computer

Description automatically generatedA screenshot of a certificate

Description automatically generated

* A screenshot of a computer

  Description automatically generatedI choose to attack the login attempts using fuzz.
* Which will give me a window with what a user has entered for the username and password. With this, if I know an email, I can attempt to brute force into the account.
* By highlighting the password and pressing on “add” in the first window, and “add” again on the second window, I’ll be given another window, where I can upload a text file of passwords (By selecting “File” from the dropdown where there is “Strings”), which zap will go through each one attempting to use that password to login.

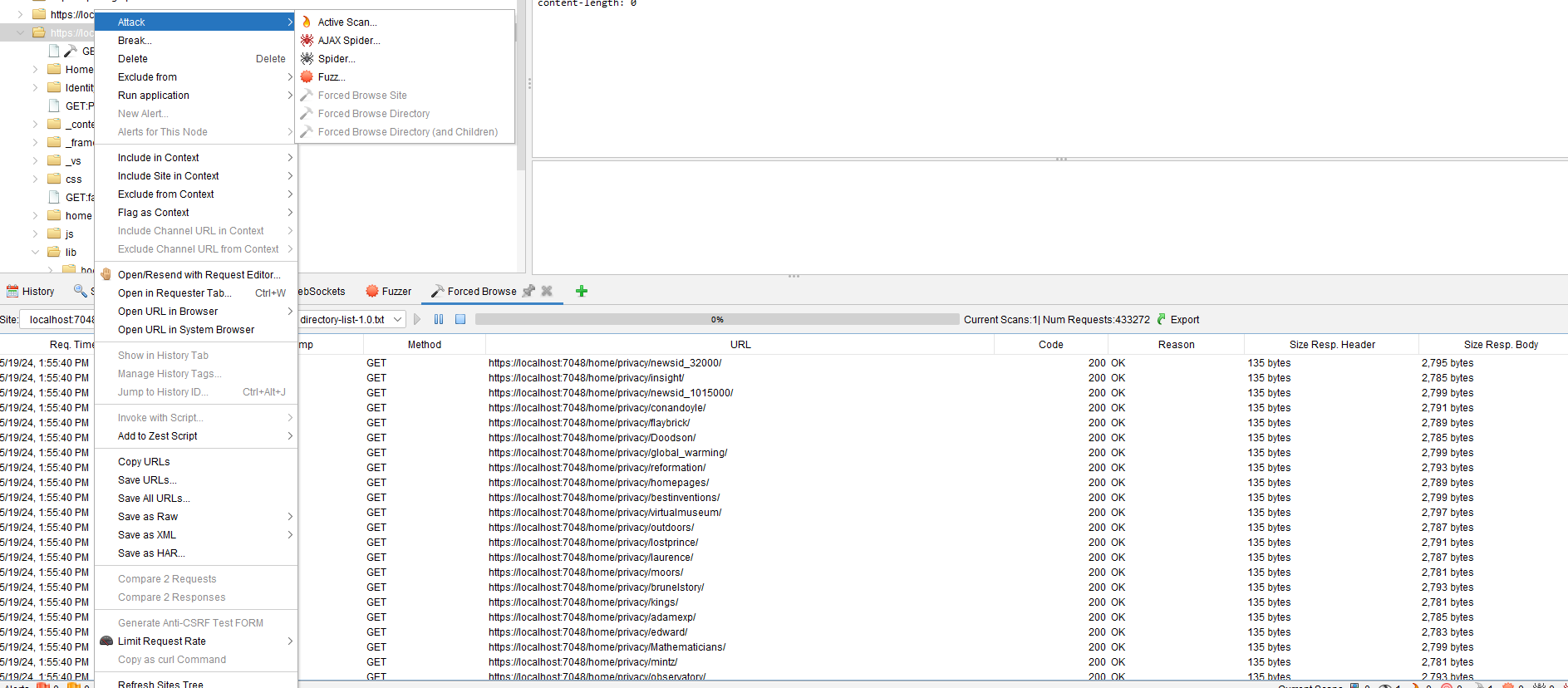
A screenshot of a computer

Description automatically generated

* A screenshot of a computer

  Description automatically generatedBy uploading a file of some of the most common passwords (Found online), now I can attempt to log in into the account.

***The following showcases the vulnerability of saving confidential information, in this case a file called “secrets.txt”, in the wwwroot folder. For this, the content of secrets.txt include my name and surname and this will show how the tool used can find this information.***



* By right clicking on the website currently being used, and choosing “Forced browse directory”, this will give me the files in the wwwroot and eventually the secrets.txt file. It is greyed out due to it already being run and collecting the directories as seen in the output window.
* Unfortunately, throughout the the execution of this, my ZAP proxy ends up crashing due to the number of files being found, but on better hardware, it would be able to find the secrets.txt file.

***To showcase this vulnerability further, I placed another file with confidential information called “confidential.txt” in a folder called “Confidential”. This time though the folder “Confidential” was not placed in the wwwroot folder.***

A screenshot of a computer

Description automatically generated

* By using the attack feature “Spider”, I can supply the URL of where the file is, which in this case is /Confidential/. If the file could be downloaded, the file would be listed down in the output window, but it is not.

From this one can gather that any file in the wwwroot is accessible to the public and any sensitive information would be easily accessible. For instance, using the scenario from the recruitment agency, if the files had to be saved in the wwwroot, any unauthorised users would be able to download the CVs files uploaded. This would allow the attacker to collect the naming convention used to store them in the database. If they manage to get access to the database, they can find the user who submitted this file by collecting their userID and get all their information.