**Scenario:**

You are building a secure website that will be used by a recruitment firm. Users will securely upload and download user information and CVs documents.

The website will have the following features and functionality:

* User registration, authentication, and authorisation/access controls (see Sections 3 and 4)
* File upload, file sharing and downloads in a secure manner (see Sections 4, 5 and 6)
* Logging (see Section 7).

**Section 1**

**Identify Threats and Possible Exploits (KU1.3, 5 marks)**

Identify at least 2 assets and at least 2 threats to the website to be developed. Use the templates in Appendix A and write appropriate documentation to:

* identify at least 3 trust levels (1 mark)
* identify at least 2 assets (2 marks)
* identify at least 2 threats related to the identified assets (2 marks)

**Note:** The marks for the mitigation strategy are awarded in Section 2.

**Note:** You need to identify your own assets; and threats. Using the assets; and threats in the sample documentation do not count towards the limit.

**Note:** You may use relevant trust levels in the appendix if the trust levels are correct for the identified assets and threats.

**Trust Levels:**

|  |  |  |
| --- | --- | --- |
| Id | Name | Description |
| T1 | Remote Anonymous User | A user who has not yet authenticated to the website |
| T2 | Generic User | A registered user with valid credentials |
| T3 | Manager | User who can review and manage job applications & CV’s |
| T4 | Admin | A user who can configure the website. |
| T5 | IT Employee | A user responsible for technical maintenance & support |
| T6 | Employer | A user who can maintain job postings and review CV’s |

**Assets:**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Name | Description | Trust Level |
| A1.1 | User’s login Credentials | User’s username & password | T2 – Generic User  T3 – Manager  T4 – Admin  T5 – IT Employee  T6 - Employer |
| A1.2 | User Personal Information | Personal data pertaining to the user such as name, email, phone number, etc. | T2 – Generic User  T3 – Manager  T4 – Admin  T5 – IT Employee  T6 - Employer |
| A1.3 | Uploaded CV’s & Documents | CV’s & Documents uploaded by the users | T2 - Generic User  T3 - HR Manager  T6 - Recruiter |
| A1.4 | Job Postings Data | Information about job postings including job description, requirements, etc. | T4 - Website Admin  T6 - Employer |

**Threats:**

|  |  |
| --- | --- |
| Id | TR1 |
| Name | Unauthorised File Access |
| Description | An attacker accesses or modifies uploaded CV’s and documents without permission |
| STRIDE | Tampering, Information Disclosure |
| Entry Points | File upload/download page |
| Assets | A1.3 - Uploaded CV’s & Documents |
| Mitigation Strategy | Implement file access controls, Use encryption, Secure file storage |

|  |  |
| --- | --- |
| Id | TR2 |
| Name | Malicious File Upload |
| Description | An attacker uploads a file containing malicious code (E.g. a script or executable) |
| STRIDE | Tempering, Information Disclosure, Elevation of Privilege. |
| Entry Points | File upload page |
| Assets | A1.3 - Uploaded CV’s & Documents |
| Mitigation Strategy | Validate file types, restrict file execution permissions, sanitise filenames |

|  |  |
| --- | --- |
| Id | TR3 |
| Name | Modification to Job Posting Data |
| Description | An attacker attempts to modify job postings, altering descriptions, requirements, etc. |
| STRIDE | Tempering, Elevation of Privilege. |
| Entry Points | Admin Control panel |
| Assets | A1.4 - Job Postings Data |
| Mitigation Strategy | Implementing Role Based Access Control (RBAC), Input validation. |

**Section 3**

**Identify Threats and Possible Exploits (AA2.1, 7 marks)**

User input should be validated properly. Use data annotations in the model to validate user input and the ModelState.IsValid check. Show, using screenshots your use of:

• A Data Annotation that makes sure that the file extension is of an appropriate type (you require to create a custom annotation for this). This custom annotation will take comma separated values of valid extension files. (2 marks).

Custom annotation created:

A computer screen shot of a program

Description automatically generated

Custom annotation being used:

A computer screen shot of a program code

Description automatically generated

• Use the data annotation implemented to verify that uploaded documents have one of the following extensions: “.docx” or “.pdf”. The Action that receives the document to be uploaded will use IsValid to verify that the file has the correct extension before processing the file (1 mark).

Using custom annotation, using IsValid to check it.

A computer screen with text

Description automatically generated

**Section 7**

**Apply tools and scanners (AA4.2, 7 marks)**

Build a website with the vulnerabilities specified.

Make use of any tools which you are familiar with, such as the ZAP tool, to test and explore the implemented vulnerabilities. Demonstrate how the vulnerabilities can be detected using the tool of choice and how it is exploited. Add step-by-step screenshots describing how you tested for vulnerabilities, adding descriptions for each screenshot.

1. **Create an account with a username “admin” and password “123” (2 marks).**

* 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 of 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.

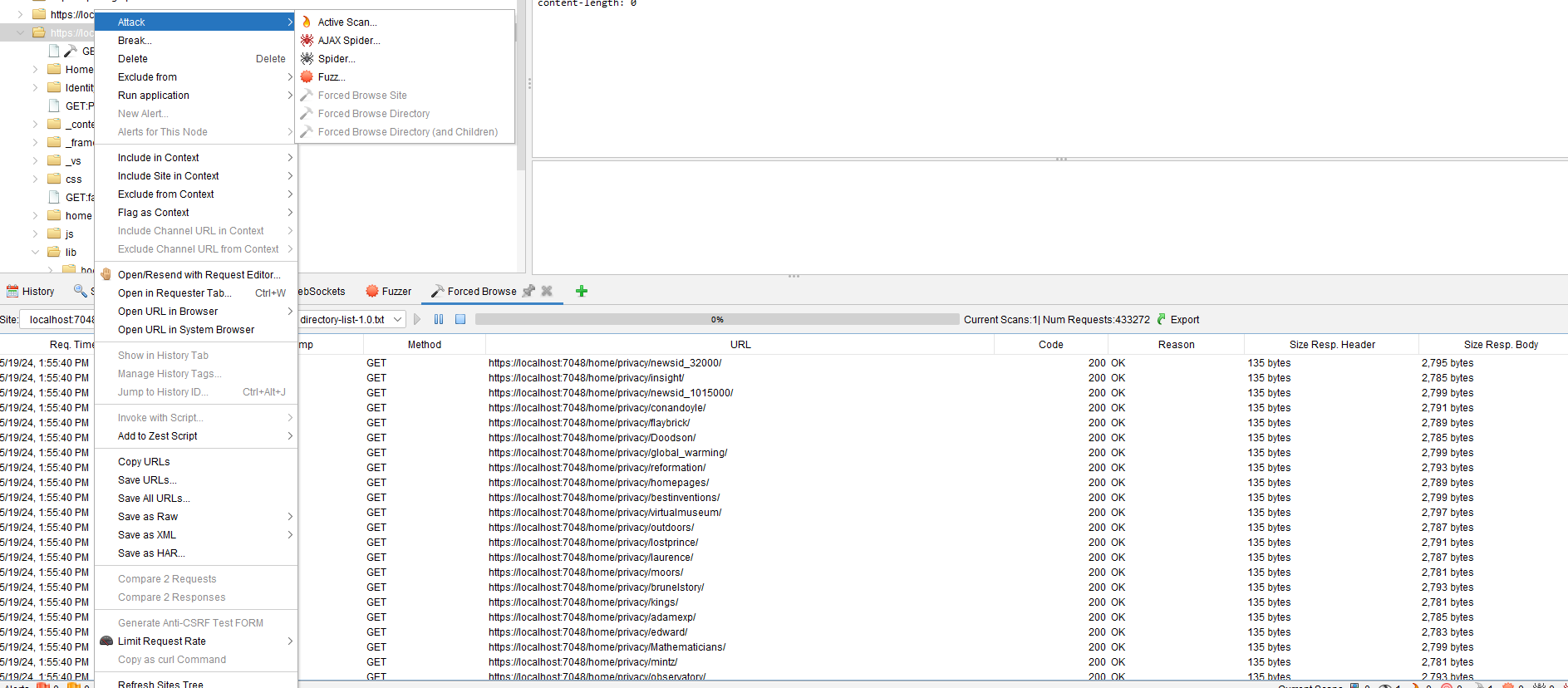
A screenshot of a computer

Description automatically generated

* A screenshot of a computer

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

1. Place a file with confidential information called “secrets.txt” in the wwwroot folder. The content of secrets.txt must include your name and surname and you must show how the tool used can find this information (2marks).



* For this, we right click on the website currently being used, and choose Forced browse directory, which 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, I would be able to find the secrets.txt file.

Place another file with confidential information called “confidential.txt” in a folder called “Confidential”. The folder “Confidential” must not be placed in the wwwroot folder.

• Use the tool to show that the file cannot be downloaded through an HTTP Get request (1 mark).

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.

• Describe, giving an example, why adding confidential information to the website source is a bad idea (2 marks).

Any file in the wwwroot is accessible to the public. Any sensitive information would be easily accessible. For instance, In the scenario we had of a recruitment agency, if the files had to be saved in the wwwroot, unauthorised users would be able to download CVs belonging to other users, and also collect the naming convention used for it to be stored in the database, if they get access to the database, they can find the user who submitted it by collecting their userID and get all of their information.

**Section 8**

**Create a report based on findings using a security tool (SE4.3, 10 marks)**

Use screenshots, images, and examples as necessary to explain and justify the results.

**Reflected XSS and HTML.Raw**

Build an action named “vulnerable” that receives a name as a parameter called name and displays the name using @Html.Raw in the View.

Explain how this implementation can be exploited by an attacker that tricks a victim into clicking a link to your action (2 marks).

By using @Html.Raw, it allows any JS script or html script passed to be executed on the browser, making it vulnerable to cross site scripting. So, if an attacker had to entice a user by showcasing the users name and telling them it’s a link leading them to their account manager, it would allow them to execute the script and collect data from there.

On your application create an action called “secret” that returns your name and surname (the secret information). Only the administrator is authorized to read the secret information.

On your application create a post action called “malicious” that accepts a payload and stores the information to file.

Craft a malicious link that when clicked by the administrator would call the vulnerable with a malicious payload that:

• displays “admin” in the view.

• requests the “secret” action (2 marks).

• posts the secret information obtained from the secret action to the “malicious” action (2 marks).

**Flawed Business Logic**

A clothing store website gives you a 30% discount on the first day when you register.

The website allows the user to purchase €10 gift cards that can be redeemed from the website itself to generate €10 store credit.

Describe, step-by-step, how you can exploit this website to purchase a €400 jacket while paying just €7 (4 marks).

By creating an account, you can purchase a €10 gift card for €7. Claim that and the user will have €13. The user can keep on buying more gift cards of €10 for just €7 and accumulating €3 more every time until they reach the €400.