MCAST

Securing Application

Assignment Guidelines

Read the following instructions carefully before you start the assignment. If you do not understand any of them, ask your lecturer.

- The assignment coversheet should be the first sheet in your assignment. Moreover, the coversheet should be fully completed with all the necessary details.
- All text\code must be properly referenced. In the absence of proper referencing, the assignment will be regarded as plagiarised.
- Copying is strictly prohibited and will be penalized in line with the College's disciplinary procedures.
- When the deadline specified by your lecturer is due, you shall hand all the required deliverables as explained in class.
- You are also required to submit your assignment to the relevant plagiarism detection service by the same deadline. If necessary, your lecturer will forward you details in order to submit your assignment to this service.
- The lecturer may hold a post-submission interview. Attendance to such interview is mandatory. Moreover, marks assigned to the criteria may be affected by the interview performance.
- Presentation of code is not sufficient to obtain the marks. Make sure to answer all the question and meet all the question requirements. Working features and/or code are also a requirement to obtain the marks.
- **All work that has been carried out, must be written down and included within the assignment as evidence. No marks will be awarded for work that is not presented.**
- **Note that some of the techniques described in this assignment are being thought for educational purposes, as well as to understand how cyber-attacks are carried out and can be defended against. They should not be used on third-party systems.**
- The deadline for this assignment is Monday 20th May 2024.

Scenario

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); and
- Logging (see Section 7).

You need to implement code and functionality that meets all the requirements set out in the Sections below.

Further, you need to write documentation, carry out tests and write reports as described in the Sections below.

MCAST

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)**; and

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.

Section 2

Strong authentication and authorisation (SE2.4, 10 marks)

Configure identity options to force the password to contain: digits; uppercase characters; lowercase characters; special characters; and have a length of at least 8 **(1 mark)**.

Create website roles and a default user during setup **(2 marks)**.

The website roles should be:

- Admin
- Employer
- Employee

A default Admin should be included during setup.

Do not create duplicate roles and/or users during setup and that the website must not crash during setup.

Use Authorize filters, where relevant to implement authentication and authorisation and prevent directory browsing **(1 mark)**.

Implement OAuth for your website to allow new users to register using a 3rd party OAuth service, such as Google Authentication, or otherwise **(3 marks)**.

Implement 2FA (two-factor authentication) **(3 marks)**.

The 2FA must make use of QR codes. Use an authenticator app to verify that the 2FA works correctly.

Section 3
Mitigate XXE and XSS (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)**.
- 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)**.

Employers should be able to create a Job Opportunity that includes a job title and description **(1 mark)**.

Employees should be able to upload a file to Job Opportunities and only the Employer who created the Job Opportunities that the Employee uploaded the file to, should be authorized to view the file. File access should be restricted only to the owner of the file and authorized users. Restrict all direct file access – file access should only be provided through relevant Actions. Access control should be checked using ActionFilters **(3 marks)**.

MCAST

Section 4

Prevention of file injection (AA2.2, 7 marks)

Users can submit two types of files as a document:

- DOCX files; and
- PDF files.

When a file is submitted to the system:

- Check that the file is of the type specified by the extension by inspecting the file prefix and file suffix "magic bytes", or file signatures, and verifying that the file is of the correct type **(4 marks)**;
- Every user must have his own cryptographic key pair. This cryptographic key pair must be used to digitally sign the file, so that each time a file is downloaded, it is possible to verify that **(3 marks)**:
    - the file was submitted by the user claiming to have submitted the file; and
    - the file did not change since the submission.

Section 5

Hybrid encryption (SE3.3, 10 marks)

Files should be uploaded and downloaded through https **(1 mark)**.

Whenever a file is uploaded it needs to be stored encrypted at rest using hybrid encryption **(5 marks)**.

Whenever a file is requested for download, it needs to be decrypted **(4 marks)**.

MCAST

Section 6

Keep sufficient logging and prevent information leaks (AA2.3, 7 marks)

File Access Logs and Error Logs should be kept in a both a database and a file system. Implement custom exceptions that do not disclose system information to the end user.

To do so, you must:

- Implement custom error pages in a proper way **(2 marks)**;
- Use Serilog to create ***both*** rolling log files and database storage to store error logs and information logs **(4 marks)**; and
- Keep File Access logs. Stored information should be stored in the database and include information such as the: ip address; timestamp; user; and other information **(1 marks)**.

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.

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

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

MCAST

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)**.

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)**.

Appendix

Trust Levels:

ID	Name	Description
T1	Remote anonymous user	A user who has not yet authenticated to the website
T2	Authenticated user	A registered user who has valid credentials
T3	Database server administrator	User who can do any operation on the underlying database
T4	Website administrator	A user who can configure the website by for example uploading new website versions, deleting files from the web server etc.
T5	Clerk	A user having this role can maintain products, their prices, categories, and sub-categories.

Assets:

ID	Name	Description	Trust level
A1	User	Assets that relate to a website user	
A1.1	User's login data	User's credentials username and password. This asset needs protection because if it is stolen another user would be able to do anything which the user can do	T2 Authenticated user T3 Database server admin
A1.2	User's personal data	User's personal data including contact information. This needs protection because some personal data might be important such as telephone number	T2 Authenticated user T3 Database server admin

Threats:

Id	TR1
Name	Adversary tries to supply malicious data when logging in
Description	Adversary tries to input special characters to be able pose as another user, or logs in without having an appropriate username and password. Handling of data is critical in this regards.
Stride	Tampering, Elevation of privilege
Entry Points	(E1.1) Login page
Assets	(A1.2) User's personal data, (A2) Backend database
Mitigation Strategy	Using stored procedures or parameterized queries