MCAST
MALTA COLLEGE of ARTS, SCIENCE & TECHNOLOGY

Assignment Cover Sheet

Please read carefully

- This assignment sheet is to be returned back to the lecturer by the student with the completed work. Work handed in after the deadline date will be penalized.
- Students caught copying from other students or plagiarizing (copying from lecturers' notes, handouts, slides, internet, books or any other printed or digital media) will be disqualified and will get a REFERRAL for their assignment or a FAIL if it is the last resit.
- An assessor has the right to ask the student to attend an interview without prior notice if the assessor wishes to confirm that the work submitted has been clearly understood by the student.
- It is the students' responsibility to keep a copy of the assignment for revision.
- This refers ONLY to Level 4 Year 1 students - All resubmissions must be authorised by the Lead Internal Verifier. Only one resubmission is possible per assignment providing that the the learner has met initial deadlines set in the assignment or has met an agreed deadline extension. Moreover, the tutor considers that the learner will be able to provide improved evidence without further guidance. Finally evidence submitted for assessment has been authenticated and accompanied by a signed and dated declaration of authenticity by the learner. **Any resubmission evidence must be submitted within 10 working days of receipt of results of assessment.

Student's Name	**Wayne Caruana**		
Programme	**B.SC (Hons.) in Software**	Academic Year	**2014/2015**
Assessor's Name	**Ryan Attard**	Group/s	**1BSC2S**

Unit No	**11**	Unit Name	**Secure Software Development**		
Assignment No	**2**	Sit	**First Sit**	Type	**Home**
Assignment Title	**Threat Modeling, Design and Implementation of a secure website**				
Issue Date		Deadline Date		Date returned to students	
Assignment IV	**Frankie Inguanez**		Date	**25 Mar 2015**	

Pass Assessment Criteria			Merit Assessment Criteria			Distinction Assessment Criteria		
Criteria	Met	Not Met	Criteria	Met	Not Met	Criteria	Met	Not Met
Unit 11-SSD : P2.1			Unit 11-SSD : M1.3			Unit 11-SSD : D1.2		
Unit 11-SSD : P2.2			Unit 11-SSD : M1.4			Unit 11-SSD : D1.3		
Unit 11-SSD : P3.1			Unit 11-SSD : M1.5					
Unit 11-SSD : P3.2								
Unit 11-SSD : P4.1								
Unit 11-SSD : P4.2								
Unit 11-SSD : P4.3								
Unit 11-SSD : P4.4								

Note : Computation of final grade for the unit will take into consideration each individual outcome as per assessment criteria.

(C*) denotes that the criteria was carried from a previous sit.

Assignment Status	

Assessment Criteria Description	
Unit 11-SSD : P2.1	Design a secure application according to specific requirements set
Unit 11-SSD : P2.2	Apply threat modeling to an application
Unit 11-SSD : P3.1	Implement security patterns in your application
Unit 11-SSD : P3.2	Document the relevant secure development activities applied to your application
Unit 11-SSD : P4.1	Perform black box security testing on a third party application
Unit 11-SSD : P4.2	Document the security features of a third party application
Unit 11-SSD : P4.3	Prepare a presentation of the security features of your application
Unit 11-SSD : P4.4	Perform code reviews of a third party application
Unit 11-SSD : M1.3	Discuss and provide alternatives for security patterns used
Unit 11-SSD : M1.4	Include detailed DFDs in your threat modeling document
Unit 11-SSD : M1.5	Implementation of further security techniques in website
Unit 11-SSD : D1.2	Argue what your chosen security patterns cannot protect the application against
Unit 11-SSD : D1.3	Perform attacks on a third party application and document them

MCAST
MALTA COLLEGE of ARTS, SCIENCE & TECHNOLOGY
INSTITUTE OF INFORMATION & COMMUNICATION TECHNOLOGY

Secure Software Development

BSc (Hons) in Software Development 2014-2015

Assignment 2
Threat Modelling, Design and Implementation of a Secure Website

First Sit

Assignment Guidelines

Read the following instructions carefully before you start working on the tasks:

- This is a home assignment
- Fill in the assignment Cover Sheet and include it with your submission.
- For your work all sources and media are allowed (e.g. internet, books, notes, implementation frameworks, etc…) but it is necessary to **include a FULL BIBLIOGRAPHY** of sources used for all your activities, even if these are not citations from textbooks
- Examine well the task descriptions which explain the requirements to achieve each criterion.
- Answers should be **properly organised and presented** in a professional layout including proper section, task and question titling or numbering.
- Copying and Plagiarism are strictly prohibited and will be penalized through the College's disciplinary procedures.
- A proper bibliography following the **APA Standard** should be presented for a referenced material used in presenting this assignment.
- Your documentation provided should be neatly bound, while also uploaded on Turnitin

Group	Class Id
BSC2	9720467
BSC3	9720470
BSC4	9720473

- A cd with the application described below should be submitted with the documentation. A database script [with data] is also necessary.

IT IS YOUR RESPONSIBILITY TO IMPLEMENT THESE REQUIREMENTS IN THE MOST SECURE WAY

Take the shopping cart website you implemented last year or in the first semester and make sure you apply the following modifications particularly with regards to security. The following are the minimum requirements you are asked to abide with:

1. Users has a profile
2. A user can be a buyer and/or seller
3. An administrator of the website can allocate/deallocate buyer/seller roles to users
4. [if seller] a user has a page where he can upload and sell software packages (.zip/.rar files)
5. [if buyer] a user can browse and decide to buy software packages. He/she can pay for any orders (with order details of packages he has chosen).
6. Shopping cart allows the user to download any software packages given that he has paid for that package (even packages he has paid for in the past)

The following are the **minimum** security requirements you are asked to implement in your website:

1. All users password should be encrypted by a strong hashing algorithm
2. An implementation of appropriate authentication and authorization mechanisms for your website.
3. Proper checks should be made to disallow simple passwords.
4. Use of captcha in registration
5. Any material downloaded from the website should be verified for its authenticity. You should use the owner's public key (who uploaded the package) to determine whether the material is valid or not. This means that before the material is downloaded it is verified by the system whether its authentic or not and thus whether the file has not been modified while stored on the server. This should be done by digital signatures.
6. Users should **not** be able to download any resources in ANY way unless they have paid for it.
7. All resources uploaded should be stored in an encrypted format on the webserver's hard disk. The encryption algorithm which should be used for this functionality is a mix of symmetric and asymmetric algorithms.
8. [redacted]en a user uploads packages you should check and allow only .rar/.zip file types.
9. [redacted]bsite menus should be loaded dynamically according to role the user has and users cannot access pages/actions which they don't have access to

You will not be assessed on your website design. However you still need to make your website user friendly.

Task 1- Definition of requirements, Threat Modeling and design (P2.1, P2.2)

For the scenario presented you should present a document outlining all the security requirements, together with any functional and non-functional requirements.

Moreover you should also present a proper Threat Model document. You should ensure that the threat model document is complete and that you cover all the sub tasks for proper Threat Modeling. The DFDs presented in this document can be Level 0 DFDs.

You should ensure that all documentation presented in this section is sufficiently detailed.

To achieve P2.1 and P2.2 you need to provide the appropriate threat modelling documentation.

Task 2– DFDs in Threat Modeling document (M1.4)

In this task you will be required to also include DFDs in your threat modeling document. These DFDs will then help you to find all entry points in your system. Your DFDs should include till Level 1.

You should also ensure that your Threat modeling document is organized and formatted properly.

To achieve M1.4 you need to present a proper DFD (till Level 1) of your system

Task 3 – Implementation of secure website (P3.1)

In this task you are required to implement a website that surely implements the requirements above.

You should also ensure that the website developed implements security patterns to make it secure. Thus, you should not only limit yourself to the security features which were listed in section 1 of this assignment. For example, you should ensure that you validate all your inputs **properly**, even though this was not listed in the requirements set in the scenario presented. Safeguarding the files stored on the server's website is the most critical requirement of the scenario presented. Make sure you mitigate any other critical vulnerabilities (also mentioned during lectures) which you come across while developing your application.

These are the security patterns.

1. Cryptography
2. Secure communication
3. User management and authorization
4. Authentication
5. Secure session management

To achieve P3.1 you need to implement a system which meets the requirements set in the scenario presented in this assignment by using appropriate security practices and patterns. Failure to secure system appropriately will fail in this criterion.

Task 4 – Documentation of security patterns and practices (P3.2/P4.3)

Prepare a presentation where each slide documents the security patterns and practices you adopted in your application and say what possible threats is each of the practices and patterns is protecting against.

Your presentation should be printed as part of the documentation.

To achieve P3.2 you need to provide documentation on at least three security practices and patterns which your application addresses. You documentation should go into detail on these aspects.

Task 5 – Implementation of further security techniques in website (M1.5)

1. To ensure optimal security for user payments you should use a secure Payment Gateway. Thus to achieve this criterion you should allow the users to pay for courses or material using their PayPal account. You should use the PayPal Adaptive Payments SDK to allow users to buy these documents. In order to achieve this criterion the transaction should pass through PayPal successfully and the user should then be allowed access to the respective resource.

2. Each of the pages/actions in your website should be secure i.e. you should verify that the user has the required role to access that action/page. Find an efficient way to implement this in such a way that when the roles (of the user or the required ones to the method/page) are changed at runtime, this works accordingly.

Task 6 – Documentation of alternative security patterns (M1.3)

To each of (1) and (2) in Task 5, discuss alternative ways how you could have implemented these. Remember this is a Merit criterion so a detailed and researched documentation (such as screenshots, code, references) is expected.

Task 7 – Security patterns – Argue on what you cannot protect against (D1.2)

Implement the following 2 additional security features with regards to 2 of the security patterns mentioned above.

1. Authentication (*replaces the login requirement above)
    - Users should connect to your system using oAuth 2.0
2. Secure session management
    - Creating an api with a sample of your methods and connect to them from your client application in a secure way

After, argue what you cannot still protect against even after applying the measures above.

Remember this is a Distinction so everything should be researched and special attention will be given to the arguments discussed.

Task 8 – Testing and Reviewing a 3rd party application (P4.1/P4.2/P4.4)

Choose a fellow class mate and perform black box testing on each other's application. Document the name of your colleague.

After you have tested and documented the test cases, analyse your colleague code by going through it and document at least 1 strong and 1 weak part of his code. Note: These should make sense i.e. use technical writing (also accompanies by screenshots/snippets of code) while describing these reviews!

Task 9 - Perform attacks on a third party application and document them (D1.3)

Take the top then most critical web application security risks as suggested by OWASP (2013) and make sure you perform attacks on your 3rd party application for any 5 of these (after you follow any guidelines you may find in documents provided by OWASP), while documenting the attacks you will perform in the documentation. Include screenshots where possible.