COMPSCI5104/5093: Assessed Exercise 1 Secured Software Engineering

Due Date: 20-02-2023 @16:30 GMT

Introduction

One objective of this course is to provide you with an understanding of why secure software engineering is important and in turn, allow you to investigate and provide a risk assessment. The aim of this assessed exercise is to test your understanding of risk analysis by asking you to investigate a scenario and identify the potential security threats and provide solutions to tackle the vulnerabilities. Also, the assessment provides the opportunity to identify the risks that affect the security of a software in different stages of Software Development Life Cycle (SDLC). This work will count for 20% of your overall course mark for Secured Software Engineering.

Marking Scheme

There are two tasks to this assessment which is worth a total of 40 marks. Tasks 1 and 2 are both worth 20 marks respectively. You will be awarded marks based on the evidence you provide in your answers. I strongly encourage you to be explicit when answering these questions.

Deliverable Instructions

You are to submit a pdf via the Moodle coursework submission link. You may annotate this document if you wish, or create a new document from scratch if you prefer. You are free to use whatever software/tools you require to create the document, just ensure that your solutions are clear. Your file name must follow the format "**sse_1_*<your_GUID>***".

Deadline

Your submission is due on **20-02-2023 at 16:30**. In accordance with the university Code of Assessment Policy, late submissions will be accepted. However, any late submissions will be penalised by 2 bands for each additional day up to 5 working days; Any late submission after 5 working days will receive the grade H (with a band value of 0). Also note that all submissions will be checked against the university plagiarism monitoring system.

Task 1 (20 Marks)

You are designing and building an e-commerce web application with the following characteristics:

- Client communicates with the web application.
- Javascript is running on the front-end (client side) connected to 3rd party APIs for advertisement and analytics etc.
- Backend Server handles the web requests.
- A database server is connected to the backend server.
- A third-party payment gateway (API) is used in case of payments made by the customers.

⇒ a) In-terms of security analysis, list down the types of attacks expected to occur in your project and give reasons to their inclusion in your list. (10 marks)

1. Sql injection attack: It is a sql malicious command which gives access to the attackers to modify the data, access the confidentiality information, and gives many security risks to the software.
2. Cross Site Scripting attack: cross site scripting allows the attackers to use malicious code to the web page and such vulnerability makes user data leak.
3. Denial-of-service (DoS) attack: DoS attacks happens when attackers trying make huge number of web requests from different IP address
4. Man-in-the-middle (MITM) attacks: MITM is an attack in which an attacker gets into communication between a user and e-commerce website, which allows attackers to steal information.
5. Phishing attack: This is an attack where attackers send fake reports to the user on behalf of the store owner and such results in the user revealing their information.
6. Local file inclusion attack: This attack allows vulnerability by allowing attackers to include files into the webpage by the improper input validation.
7. XML external entity (XXE) attack: This attacks allows attackers to vulnerable in the xml parser to file confidential data and also this attacks causes the denial of service attack
8. Brute Force attack: This attack allows the attacker to crack password or encryption key multiple times until finding the correct one.

⇒ b) Against each identified attack, what can be the vulnerabilities in the design/code and what will be counter measure(s) you will take to address the vulnerabilities and minimise the attack surface. (10 marks)

1. Vulnerability on sql injection would be allowing malicious sql command into the software by the attackers and lack of validation process.
   Countermeasures for sql injection is using parameterized sql queries and proper validation about the user input.

2. Vulnerability on cross site scripting is allowing malicious code into the web page and lack of output encoding and input validation
   Countermeasure for cross site scripting is implementing proper output encoding and input validation to prevent malicious code.

3. Vulnerability on DoS attacks happens because of improper traffic filtering and rate limit
   Countermeasure for DoS attacks to use proper traffic filtering to make fake web requests to get blocked

4. Vulnerability MITM attacks happens because of improper encryption technique which allow attackers to intercept between user and store
   Countermeasure for MITM attacks is to use SSL encryption technique to secure the transmitting data from the attacker.

5. Vulnerability on phishing would be user not knowing the true report from the store owner
   Countermeasures for phishing attacks is to having proper email authentication and using SSL encryption technique

6. Vulnerability on local file inclusion happens because of input validation in file path or directory.

   Countermeasures are having proper input validation and using absolute file paths.

7. Vulnerability on XXE because having improperly configured xml parser.

   Countermeasures for XXE is disabling the external entities and using secure xml parser

8. Vulnerability on brute force attack is giving attacker multiple chance to crack the password or encryption key

   Countermeasures for brute force is to limit the chance of trying password to login, using strong password, using multi factor authentication.

Task 2 (20 Marks)

Security is one of the most critical aspects of software quality. Software security refers to the process of creating and developing software that assures the integrity, confidentiality, and availability of its code, data, and services. Research highlights several risks that affect the security of the software in several phases of SDLC. Read the paper titled "Systematic Literature Review on Security Risks and Its Practices in Secure Software Development".

⇒ a) list the top 5 risks from the paper that involved in the design stage, coding phase, testing phase and deployment phase. (10 marks)

Design Phase

1. lack of security design awareness, guidance and training

2. Improper restriction to share resource access

3. Failure to handle error

4. Lack of developing threat modeling during the design phase

5. Lack of design data encryption and validation features

Coding phase

1. Tampering: is the unauthorized modification of data

2. SQL injection

3. cross site scripting, cross site request forgery

4. spoofing: An attempt to gain access to system by using fake identity

5. HTTP application instead of HTTPS

Testing Phase

1. lack of penetration analysis security testing

2. lack of static and dynamic analysis security testing

3. lack of final security review

4. lack of functional and non functional testing

5. lack of unit testing

Deployment Phase

1. lack of default software configuration

2. improperly enabled services and ports

3. ignoring security breaks

4. lack of certification in final release and archives

5. lack of thread model updating

⇒ b) against each identified risk factor, give reason why this it is a top risk factor and if you don't agree which some factors in the list, would you like to add a new factor to the list? If yes, include the new factor(s) and give reason for the inclusion of new factor(s) in the list. (10 marks)

Design Phase

1. Lack of security design awareness, guidance and training
   To secure the software, the initial step is to have good knowledge of security design with proper training and guidance.

2. Improper restriction to share resource access
   Giving access to the resource is one of the main security tasks in the software. If such a task is not designed properly, then such improper design restriction to share the resource will lead to unauthorized access.

3. Failure to handle error
   Unauthorized access and security breaches will occur, if not handling the errors occurred in the software.

4. Lack of developing threat modeling during the design phase
   Threat modeling will identify the potential threats and vulnerabilities in the software, so lack of developing threat modeling will impact software security.

5. Lack of design data encryption and validation features
   Lot of data will be transmitted and received through the encryption secure method. Because Improper encryption process, unauthorized users will intercept and a lot of information gets leaked. And validation process error will lead to database systems.

Coding Phase

1. Tampering: is the unauthorized modification of data
   Data getting tampered will cause security risks where information gets leaked, multiple data might be changed.

2. SQL injection
   It is a malicious sql command which gives access to the attackers to modify the data, access the confidentiality information, and gives many security risks to the software.

3. Cross Site Scripting, Cross Site Request Forgery
   Cross Site Scripting allows the attackers to use malicious code to the web page and such vulnerability makes user data leak.
   Cross Site Request Forgery allows the attacker to manipulate the user to click some url links which makes attackers get access to user data.

4. Spoofing: An attempt to gain access to system by using fake identity
   Spoofing is done by an unverified user which gets access to information without getting detected.

5. HTTP application instead of HTTPS
   If the application is using HTTP request protocol to transfer data instead of HTTPS, then more possibilities of user data will be stolen.

Testing Phase

1. Lack of penetration analysis security testing
   Improper penetration testing will make vulnerabilities in the system unnoticed and would cause a lot of issues by the attacker.

2. Lack of static and dynamic analysis security testing
   Improper static and dynamic testing will improve attacker chance to find vulnerabilities in the system and gain access

3. Lack of final security review
   Before releasing the software, making a final security review will ensure security of the software to get released to the user.

4. Lack of functional and non functional testing
   Functional and non functional testing make sure about the quality and reliability of the software which prevent from security risks

5. Lack of unit testing
   Unit testing verifies each units or components of the software which prevents from unnoticed vulnerability and software attack

Deployment Phase

1. Lack of default software configuration
   Default configuration errors like default login credentials will lead to vulnerabilities which expose a lot of confidential data in configuration files.

2. Improperly enabled services and ports
   If services and ports are enabled improperly, then such action will risk losing a lot of sensitive information. so having proper services and ports will restrict the network access from the attackers.

3. Ignoring security breaks
   Ignoring security breaks will lead to a lot of data, financial loss and damage with legal consequences and so having a proper security monitoring system is needed.

4. Lack of certification in final release and archives
   Legal certification gives the verification that the system has been designed according to security standards. without certification, such deployed software will be fined.

5. Lack of thread model updating
   If the thread model is not updating properly during the deployment phase, this will lead to a new threat rise in overtime in the software after deployment.