SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF CYBER SECURITY

QUESTION BANK

IV SEMESTER-SECOND YEAR

CY3462 – SECURE SOFTWARE ENGINEERING

Regulation – 2023

Academic Year: 2024 – 2025 (EVEN)

Prepared by

Ms. T.Sathya, Assistant Professor/CYS

SRM VALLIAMMAI ENGINEERING COLLEGE

SRM Nagar, Kattankulathur-603203

DEPARTMENT OF CYBER SECURITY

QUESTION BANK

SUBJECT : CY3462 – Secure Software Engineering

SEM / YEAR : IV SEMESTER/ SECOND YEAR

UNIT -I SECURITY A SOFTWARE ISSUE AND WHAT MAKES SOFTWARE SECURE

Introduction, the problem, Software Assurance and Software Security, Threats to software security, Sources of software insecurity, Benefits of Detecting Software Security, Properties of Secure Software, Influencing the security properties of software, Asserting and specifying the desired security properties.

UNIT –I [PART-A]

Q.No	Question	Competence	Level
1	Define software security and its significance.	Remembering	BTL1
2	What is software assurance, and how does it differ from software security?	Understanding	BTL2
3	List two common threats to software security.	Remembering	BTL1
4	Identify two key properties of secure software.	Remembering	BTL1
5	Name two sources of software insecurity.	Remembering	BTL3
6	State two benefits of detecting software security issues.	Remembering	BTL1
7	Explain the importance of software assurance in secure software development.	Understanding	BTL2
8	What are the consequences of ignoring software security threats?	Remembering	BTL1
9	Describe how security properties impact the overall software lifecycle.	Understanding	BTL2
10	How do software assurance and security properties interrelate?	Understanding	BTL2
11	Provide examples of internal and external sources of software insecurity.	Remembering	BTL1
12	Summarize the benefits of incorporating secure practices in software development.	Understanding	BTL2
13	Illustrate how threats to software security can arise during development.	Remembering	BTL1
14	Compare the characteristics of secure and insecure software.	Understanding	BTL2
15	State two ways in which software assurance can mitigate software security threats.	Remembering	BTL1
16	What role do desired security properties play in the design of software?	Understanding	BTL2
17	Explain why early detection of software security issues is critical.	Understanding	BTL2
18	List two techniques for asserting security properties in software.	Remembering	BTL1
19	How do security threats impact software performance and reliability?	Understanding	BTL2
20	Describe the influence of security properties on software quality.	Remembering	BTL1
21	What are the benefits of specifying desired security properties during software development?	Understanding	BTL2
22	Explain the relationship between secure software and user trust.	Remembering	BTL1
23	Name two potential consequences of software insecurity for an organization.	Remembering	BTL1
24	How can understanding software security threats improve development practices?	Understanding	BTL2
25	Summarize the advantages of integrating security measures into the software lifecycle.	Understanding	BTL2

UNIT –I [PART-B]					
Q.No	**Question**		**Marks**	**Competence**	**Level**
1		Identify internal and external sources of software insecurity.	16	Analysing	BTL1
2		Evaluate the role of software assurance in balancing functionality and security in software systems.	16	Evaluating	BTL5
3		Illustrate the role of software assurance in preventing specific security threats such as SQL Injection and Cross-Site Scripting (XSS).	16	Applying	BTL3
4		Compare and contrast different approaches to influencing the security properties of software during the development process.	16	Analysing	BTL4
5		Investigate the consequences of failing to integrate security properties into software development.	16	Analysing	BTL4
6		Assess the role of security property specification in ensuring high-quality software.	16	Evaluating	BTL3
7		Develop a security strategy to mitigate the threats caused by software insecurity.	16	Applying	BTL4
8	**A**	Apply the principles of secure software development to redesign an insecure application module.	08	Applying	BTL1
	B	Analyze the benefits of incorporating secure practices into the software lifecycle with examples.	08	Analyzing	BTL4
9		Examine the effectiveness of current software assurance practices in mitigating security threats.	16	Analysing	BTL4
10		Justify the inclusion of security properties in the requirements phase of software development.	16	Evaluating	BTL1
11	**A**	Categorize various threats to software security based on their impact and likelihood of occurrence.	08	Analyzing	BTL4
	B	Explain how the benefits of detecting software security issues can be maximized in a real-world project.	08	Analyzing	BTL4
12		Evaluate the effectiveness of a chosen secure software development methodology in addressing software insecurity.	16	Evaluating	BTL2
13		Demonstrate how software developers can assert and specify desired security properties during the software development lifecycle.	16	Applying	BTL3
14		Compare the long-term organizational impact of addressing versus ignoring software security threats.	16	Analyzing	BTL4
15		Assess the impact of early detection of security vulnerabilities on the overall success of a software project.	16	Evaluating	BTL6
16		Analyze the relationship between the sources of software insecurity and the types of threats they generate.	16	Analyzing	BTL4
17		Critique a case study where a lack of software assurance led to significant security failures.	16	Evaluating	BTL5
18		Discuss how these sources contribute to specific types of threats.	16	Applying	BTL3
19		Evaluate the effectiveness of integrating security properties into the software development lifecycle and its impact on software reliability.	16	Evaluating	BTL5
20		Design a robust framework to address software security threats and ensure compliance with industry security standards.	16	Creating	BTL6
21		Evaluate the role of software assurance in preventing large-scale security breaches and maintaining system integrity.	16	Evaluating	BTL5

22		Propose a methodology for detecting, specifying, and mitigating software security vulnerabilities in real-time applications.	16	Creating	BTL6
23		Assess the long-term organizational benefits of adopting secure software development practices over traditional methods.	16	Evaluating	BTL6

UNIT -II REQUIREMENTS ENGINEERING FOR SECURE SOFTWARE

Introduction, the SQUARE process Model, Requirements elicitation and prioritization.

UNIT-II [PART-A]

Q.No	Question	Competence	Level
1	What is requirements engineering in the context of secure software?	Remembering	BTL1
2	Define the term 'security requirements' in software development.	Understanding	BTL2
3	What is the SQUARE process model?	Remembering	BTL1
4	Why is requirements engineering crucial for secure software development?	Remembering	BTL1
5	What are the primary goals of requirements elicitation in secure software?	Understanding	BTL2
6	Explain the role of security in the requirements engineering process.	Remembering	BTL1
7	What is the importance of prioritizing security requirements in software development?	Understanding	BTL2
8	List the steps involved in the SQUARE process model.	Remembering	BTL1
9	What is the first step in the SQUARE process model?	Understanding	BTL2
10	Why is risk assessment important in the SQUARE process model?	Remembering	BTL1
11	Explain the concept of 'use case development' in the SQUARE process model.	Understanding	BTL2
12	What are quality requirements in the context of secure software?	Remembering	BTL1
13	Describe the role of asset identification in secure software requirements engineering.	Understanding	BTL2
14	What is the relationship between requirements elicitation and security in software development?	Understanding	BTL1
15	How does prioritization of requirements impact the security of a software product?	Remembering	BTL1
16	Why is it important to involve stakeholders in the requirements elicitation process for secure software?	Remembering	BTl1
17	Explain the concept of 'risk assessment' in secure software development.	Understanding	BTL2
18	What are the challenges in eliciting security requirements for software?	Understanding	BTL2
19	Explain how the SQUARE model helps in improving the security of software.	Remembering	BTL1
20	What is the role of evaluation in the SQUARE process model?	Understanding	BTL2
21	Define 'asset identification' in the context of secure software requirements.	Remembering	BTL1
22	What is the significance of documenting security requirements early in the software development process?	Understanding	BTL2

23	What are the main priorities when eliciting security requirements for a software system?	Remembering	BTL1
24	How does prioritizing security requirements contribute to minimizing software vulnerabilities?	Understanding	BTL2
25	What types of stakeholders are typically involved in requirements elicitation for secure software?	Remembering	BTL1

UNIT -II [PART-B]

Q.No		**Question**	**Marks**	**Competence**	**Level**
1		Analyze the importance of integrating security requirements in the early stages of software development. Discuss how this influences the overall software security.	16	Analyzing	BTL1
2		Apply the SQUARE process model to a case study of a secure software application. Analyze how each step of the model contributes to the overall security of the **system.**	16	Analysing	BTL4
3		Evaluate the role of risk assessment in the SQUARE process model and its impact on prioritizing security requirements.	16	Evaluating	BTL5
4		Apply the SQUARE model to a software development project for a healthcare system and evaluate the steps taken to ensure security requirements are met.	16	Applying	BTL2
5		Analyze the relationship between functional and security requirements in the context of secure software and discuss how to balance both during the requirements elicitation phase.	16	Analyzing	BTL3
6		Evaluate the effectiveness of requirements prioritization in the SQUARE process for managing limited resources during secure software development.	16	Evaluating	BTL3
7		Analyze how use case development in the SQUARE process model helps in identifying potential security vulnerabilities early in the software development lifecycle.	16	Analyzing	BTL2
8		Discuss the role of stakeholders in the requirements elicitation process for secure software, and evaluate the challenges of managing diverse stakeholder needs.	16	Evaluating	BTL5
9		Evaluate how asset identification in the SQUARE model aids in defining security requirements for critical systems.	16	Evaluating	BTL3
10		Analyze the role of documentation in the requirements elicitation phase for secure software. How does proper documentation contribute to reducing vulnerabilities?	16	Analyzing	BTL4
11		Evaluate the importance of continuous evaluation during the requirements engineering phase and its impact on secure software development.	16	Evaluating	BTL5
12		Apply the principles of the SQUARE model to an e-commerce website. Evaluate how it ensures secure transaction and customer data protection.	16	Applying	BTL1
13		Analyze the impact of incomplete or poorly defined security requirements on the final security posture of a software application.	16	Analyzing	BTL1

14		Evaluate how the use of the SQUARE process model can help in identifying and mitigating potential security risks during the software development lifecycle.	16	Evaluating	BTL5
15		Evaluate the effectiveness of the SQUARE process model in identifying and managing security requirements in software engineering.	16	Evaluating	BTL2
16		Analyze the challenges and solutions in eliciting security requirements for a large-scale distributed system.	16	Analysing	BTL1
17		Apply the SQUARE process model to identify the security requirements of a mobile banking application. Discuss the risks and the prioritized security features.	16	Applying	BTL4
18		Evaluate the effectiveness of the SQUARE process model in integrating security requirements into the software development lifecycle. Discuss its advantages and limitations.	16	Evaluating	BTL5
19		Design a security requirements engineering framework for a financial application, using the SQUARE process model. Discuss how each step ensures the security of the application.	16	Creating	BTL6
20		Assess the role of stakeholders in the requirements elicitation phase for secure software. Propose strategies for effectively managing diverse stakeholder expectations and priorities related to security.	16	Evaluating	BTL5
21		Evaluate how risk assessment in the SQUARE process model helps prioritize security requirements in a cloud-based system. Discuss how this prioritization can impact the development and maintenance of secure software.	16	Creating	BTL6
22		Create a comprehensive requirements elicitation and prioritization strategy for a secure e-commerce platform, incorporating both functional and non-functional security requirements. Discuss how the strategy mitigates potential security risks.	16	Evaluating	BTL5
23		Discuss the challenges and solutions in prioritizing security requirements in agile software development.	16	Evaluating	BTL5

UNIT –III SECURE SOFTWARE ARCHITECTURE AND DESIGN

Introduction, software security practices for architecture and design: architectural risk analysis, software security knowledge for architecture and design: security principles, security guidelines and attack patterns Secure coding and Testing: Code analysis, Software Security testing, Security testing considerations throughput the SDLC.

UNIT-III [PART-A]

Q.No	Question	Competence	**Level**
1	What is architectural risk analysis in the context of software security?	Remembering	BTL1
2	Why is software security knowledge important for architecture and design?	Understanding	BTL2
3	What are the primary objectives of secure coding practices?	Remembering	BTL4
4	Define "attack patterns" in the context of software architecture and design. *(Remembering)*	Remembering	BTL1
5	What is the role of security principles in software architecture	Understanding	BTL2

	and design? *(Understanding)*		
6	What is the purpose of software security testing throughout the SDLC?	Understanding	BTL2
7	Explain the term "code analysis" in secure software development.	Understanding	BTL2
8	What are security guidelines for software architecture and design?	Remembering	BTL1
9	What is the importance of security testing in the software development lifecycle?	Understanding	BTL2
10	Describe the role of secure coding practices in preventing common software vulnerabilities.	Understanding	BTL1
11	What is the significance of attack patterns in identifying vulnerabilities in software design?	Understanding	BTL2
12	What are the key phases in software security testing?	Remembering	BTL1
13	How does architectural risk analysis help in identifying security risks during the design phase?	Understanding	BTL2
14	What are some common security principles used in architecture and design?	Remembering	BTL1
15	What are the challenges of integrating security testing throughout the SDLC?	Understanding	BTL2
16	What is the relationship between secure coding and reducing software vulnerabilities?	Understanding	BTL2
17	Why is continuous security testing important during the SDLC?	Remembering	BTL1
18	How do security guidelines contribute to developing secure software architecture?	Understanding	BTL2
19	What role do security principles play in minimizing vulnerabilities in software architecture?	Remembering	BTL1
20	What are attack patterns and how do they help in identifying potential risks in the software design?	Remembering	BTL1
21	What is the goal of software security testing at the system and integration levels?	Understanding	BTL2
22	How do secure coding practices help in addressing input validation vulnerabilities?	Understanding	BTL2
23	What are some common tools used for code analysis in software security?	Remembering	BTL1
24	What is the importance of threat modeling during the software design phase?	Understanding	BTL2
25	Explain how secure software testing helps in detecting vulnerabilities that may not be visible through code review.	Understanding	BTL2

UNIT -III [PART-B]

Q.No		**Question**	**Marks**	**Competence**	**Level**
1		Analyze the importance of architectural risk analysis in the context of building secure software. How does it affect the overall security of the application?	16	Analysing	BTL1
2	**A**	Evaluate the role of software security principles in preventing security vulnerabilities during the design and architecture phase of software development.	08	Evaluating	BTL5
	B	Apply secure coding practices to design a secure software application and analyze how these practices help prevent common vulnerabilities like buffer overflows and SQL injection.	08	Analysing	BTL4

3		Evaluate the effectiveness of various software security testing techniques (e.g., static analysis, dynamic testing) in identifying security flaws at different stages of the SDLC.	16	Evaluating	BTL2
4		Analyze how architectural risk analysis can help prioritize security risks and mitigation efforts in the software design phase.	16	Evaluating	BTL5
5		Discuss how software security knowledge (such as secure coding and threat modeling) impacts software architecture and design decisions. Provide an example from real-world scenarios.	16	Analysing	BTL2
6		Evaluate the importance of integrating security testing early in the SDLC. Discuss how this practice can prevent costly vulnerabilities later in the software development process.	16	Evaluating	BTL1
7	**A**	Apply security testing considerations to a case study of a web application. Analyze how security testing can be integrated throughout the SDLC to minimize vulnerabilities.	08	Applying	BTL3
	B	Analyze the security implications of architectural choices in a cloud-based system. How can architectural risk analysis help in identifying vulnerabilities unique to cloud environments?	08	Analysing	
8		Evaluate the effectiveness of secure coding guidelines in preventing common vulnerabilities like cross-site scripting (XSS) and cross-site request forgery (CSRF).	16	Evaluating	BTL5
9		Discuss how software security testing considerations vary during the different phases of the SDLC (design, development, testing, and deployment).	16	Evaluating	BTL5
10	**A**	Analyze how attack patterns influence the design of secure software. Discuss how identifying common attack patterns can help in designing more robust security features.	08	Analysing	BTL4
	B	Evaluate the role of code analysis in identifying and mitigating software security vulnerabilities. Discuss the challenges and limitations of relying on code analysis alone.	08	Evaluating	
11	**-**	Discuss the role of security guidelines in the architectural design of software systems. How do these guidelines help prevent vulnerabilities such as privilege escalation or information leakage?	16	Evaluating	BTL5
12	**A**	Analyze how security testing is implemented across different stages of the SDLC. Discuss the key considerations for performing security testing at the system integration and deployment stages.	08	Analysing	BTL1
	B	Evaluate the challenges and solutions related to implementing architectural risk analysis in agile software development methodologies.	08		
13		Apply the principles of secure coding to design a secure login mechanism for a mobile app. Evaluate the security measures that should be implemented to protect user data.	16	Applying	BTL2
14		Analyze the impact of integrating security testing throughout the SDLC. Discuss how early testing influences the security of the final product and the overall cost of software development.	16	Analysing	BTL6

15		Evaluate the effectiveness of architectural risk analysis in identifying potential security vulnerabilities during the software design phase. Discuss its impact on the security of the final software product and suggest ways to enhance its effectiveness.	16	Evaluating	BTL5
16		Design a secure software architecture for an online banking system, considering common security threats such as SQL injection, Cross-Site Scripting (XSS), and data leakage. Discuss how the principles of secure coding and security guidelines can be incorporated into this architecture to mitigate these risks.	16	Creating	BTL6
17		Evaluate how secure coding practices, such as input validation, error handling, and secure authentication, can prevent common vulnerabilities in software systems. Discuss how these practices can be implemented during the design and development phases to enhance software security.	16	Evaluating	BTL6
18		Discuss the integration of security testing throughout the software development lifecycle (SDLC). Evaluate the challenges and solutions in applying different security testing techniques (e.g., static analysis, dynamic testing) at various stages of the SDLC and their role in identifying and mitigating vulnerabilities.	16	Creating	BTL5
19		Create a comprehensive security testing plan for a web-based application. Include strategies for code analysis, penetration testing, and security testing during the SDLC. Evaluate how your plan addresses security issues throughout the development and deployment phases.	16	Evaluating	BTL5

UNIT –IV SECURITY AND COMPLEXITY

System Assembly Challenges: introduction, security failures, functional and attacker perspectives for security analysis, system complexity drivers and security.

UNIT -IV [PART-A]

Q.No	Question	Competence	Level
1	What is system assembly in the context of software development?	Remembering	BTL1
2	Define security failures in the context of system assembly.	Remembering	BTL1
3	Why is security a challenge during the system assembly process?	Understanding	BTL2
4	What are functional perspectives in security analysis?	Remembering	BTL1
5	Explain the concept of attacker perspectives in security analysis.	Understanding	BTL2
6	How does system complexity affect security?	Understanding	BTL2
7	What are some common types of security failures in system assembly?	Remembering	BTL1
8	What is the role of authentication mechanisms in preventing security failures?	Remembering	BTL1
9	How does inadequate input validation contribute to security failures?	Understanding	BTL2
10	What is the functional perspective of security in a system?	Remembering	BTL1
11	Describe the attacker perspective in security analysis.	Remembering	BTL1
12	What are some of the security challenges associated with system integration?	Understanding	BTL2

13	What is the importance of security testing during system assembly?	Remembering	BTL1
14	What role does system configuration play in preventing security failures?	Understanding	BTL2
15	Explain how complex systems are more vulnerable to security failures.	Remembering	BTL1
16	Why is it important to consider attacker perspectives during system assembly?	Understanding	BTL2
17	What are the main drivers of system complexity that influence security?	Remembering	BTL1
18	How does system misconfiguration lead to security vulnerabilities?	Remembering	BTL1
19	What is the importance of integrating security measures in the system assembly process?	Understanding	BTL2
20	How can system complexity increase the risk of security breaches?	Remembering	BTL1
21	What is the significance of threat modeling during system assembly?	Understanding	BTL2
22	What is the relationship between functional requirements and security in system assembly?	Remembering	BTL1
23	How can security failures in one component affect the entire system?	Understanding	BTL2
24	What are some common strategies to mitigate security failures during system assembly?	Remembering	BTL1
25	What is the role of continuous monitoring in ensuring system security during assembly?	Understanding	BTL2

UNIT -IV [PART-B]

Q.No		**Question**	**Marks**	**Competence**	**Level**
1	**A**	Analyze the impact of system complexity on security failures during the assembly phase. Discuss how increased complexity can introduce new vulnerabilities and complicate security measures.	8	Analysing	BTL2
	B	Evaluate the importance of considering both functional and attacker perspectives in security analysis during system assembly. How do these perspectives help in identifying potential vulnerabilities?	8		
2		Apply the concept of security failures to a case study of a software system undergoing assembly. Identify potential security risks and suggest strategies to mitigate them.	16	Applying	BTL1
3		Evaluate how system misconfiguration can lead to security breaches during system assembly. Discuss strategies to avoid misconfigurations and enhance system security.	16	Evaluating	BTL1
4		Analyze how system integration can introduce security risks, and propose solutions to minimize these risks during the assembly phase.	16	Applying	BTL3
5		Discuss how the functional and attacker perspectives can be used to guide the development of secure software systems. Provide examples of each perspective in a system assembly scenario.	16	Analysing	BTL4
6	**A**	Evaluate the role of continuous security monitoring during system assembly. How can it help detect and mitigate security failures?	8	Evaluating	BTL5
	B	Analyze how the interaction between various system components during assembly can lead to new vulnerabilities. Discuss how these vulnerabilities can be	8	Analysing	BTL4

		identified and mitigated.			
7	**A**	Evaluate the challenges in balancing security and functionality during system assembly. How can these challenges be addressed to ensure a secure system without compromising functionality?	8	Evaluating	BTL5
	B	Discuss how threat modeling can be used to identify potential security failures in the system assembly process. How does it help in preventing vulnerabilities?	8		
8	**A**	Analyze how the attacker perspective can inform the identification of potential security risks in the assembly phase of a system. Discuss how this perspective helps identify threats that functional analysis may miss.	8	Analysing	BTL1
	B	Evaluate how complex systems can be made secure by focusing on both technical and organizational factors during the assembly process. Discuss specific examples of security best practices for complex systems.	8		BTL2
9	**A**	Analyze the relationship between system complexity and the effectiveness of security controls. How can system designers ensure that security measures are scalable as system complexity increases?	8	Analysing	BTL3
	B	Evaluate the impact of system failures on overall security and performance during the assembly phase. How can system designers address these failures to ensure a secure and stable system?	8		
10		Apply an example of a security failure from a real-world system assembly scenario. Analyze the failure and suggest preventive measures to avoid similar vulnerabilities in future systems.	16	Applying	BTL5
11		Discuss how software architects can address the security challenges posed by system complexity. Analyze strategies for ensuring that security is integrated throughout the system assembly process.	16	Remembering	BTL1
12		Evaluate how proper risk analysis and mitigation strategies can address security failures in system assembly. Discuss how these strategies improve the overall security posture of a system.	16	Evaluating	BTL3
13		Analyze how attacker perspectives, combined with functional requirements, can guide secure system design. Provide examples of how such combined analysis improves system security.	16	Analysing	BTL6
14		Evaluate the impact of system complexity on security during system assembly. Discuss how increasing complexity introduces new security risks and how these risks can be mitigated through proper design and testing.	16	Evaluating	BTL5
15		Design a security strategy for a large-scale distributed system, considering both functional and attacker perspectives in security analysis. Explain how you would address potential security failures during system assembly.	16	Creating	BTL6
16		Evaluate how system misconfiguration contributes to security failures during system assembly. Discuss preventive measures and best practices that can be	16	Evaluating	BTL5

		applied to ensure secure configurations throughout the assembly process.			
17		Discuss how an attacker's perspective can be integrated into the system assembly process to identify vulnerabilities. Evaluate the effectiveness of this approach in preventing security breaches before deployment.	16	Evaluating	BTL6
18		Create a comprehensive security testing framework for a complex system under assembly. Analyze how different testing techniques (e.g., static code analysis, penetration testing) can be employed to identify and resolve security vulnerabilities during system assembly.	16	Creating	BTL5

UNIT -V GOVERNANCE AND MANAGING MORE SECURE SOFTWARE

Governance and security, Adopting an enterprise software security framework, How much security is enough?, Security and project management, Maturity of Practice.

[PART-A]

Q.No	Question	Competence	Level
1	What is governance in the context of security?	Remembering	BTL1
2	Why is adopting an enterprise software security framework important for organizations?	Understanding	BTL2
3	What are the key components of an enterprise software security framework?	Remembering	BTL1
4	How does an enterprise security framework help in managing security risks?	Understanding	BTL2
5	What does the term "security maturity" mean in software development?	Remembering	BTL1
6	What is the relationship between governance and security in an organization?	Understanding	BTL2
7	Why it is important to assess how much security is enough for an organization?	Remembering	BTL1
8	What factors should be considered when determining the appropriate level of security for a system?	Understanding	BTL3
9	How does the adoption of an enterprise software security framework help in managing compliance requirements?	Understanding	BTL2
10	What are the different stages in the security maturity model?	Remembering	BTL1
11	What role does project management play in ensuring software security?	Understanding	BTL2
12	What are the challenges in determining how much security is enough for an organization?	Understanding	BTL5
13	What is the significance of continuous monitoring in an enterprise security framework?	Understanding	BTL2
14	What is the purpose of risk assessment in adopting a software security framework?	Remembering	BTL1
15	How does a mature security practice improve an organization's overall security posture?	Remembering	BTL1
16	What is the role of security governance in preventing security breaches in an organization?	Understanding	BTL2
17	What are the benefits of a well-defined security policy in an enterprise software security framework?	Remembering	BTL1
18	How can security frameworks be integrated with an organization's overall business strategy?	Understanding	BTL2
19	What is the role of incident response in an enterprise software	Understanding	BTL2

	security framework?			
20	How do maturity models help in assessing the security practices of an organization?		Remembering	BTL1
21	What are the primary objectives of a security framework in enterprise software development?		Understanding	BTL2
22	Why is project management essential for the successful implementation of a security framework?		Remembering	BTL1
23	How does the maturity of security practices impact the effectiveness of software security?		Remembering	BTL1
24	What are some common risks associated with inadequate security frameworks in enterprise software?		Understanding	BTL2
25	Why is it important to continuously improve security practices in an organization?		Remembering	BTL1

UNIT -V [PART-B]

Q.No		**Question**	**Marks**	**Competence**	**Level**
1	**A**	Analyze the challenges an organization might face when determining how much security is enough. Discuss how these challenges can be overcome through risk management strategies.	8	Analysing	BTL4
	B	Evaluate the importance of adopting an enterprise software security framework in an organization. Discuss how such a framework can reduce security vulnerabilities and improve risk management.	8	Evaluating	BTL5
2	**-**	Discuss how project management practices can ensure that security is integrated into the software development process. Analyze how different project management models support or hinder security integration.	16	Analysing	BTL4
3	**-**	Evaluate the role of security governance in managing the security risks of an enterprise. How does governance influence the development and enforcement of security policies?	16	Evaluating	BTL5
4	**A**	Analyze the relationship between security maturity and the effectiveness of software security practices. Discuss how an organization can measure and improve its security maturity.	8	Analysing	BTL4
	B	Evaluate how security maturity models (e.g., CMMI, SAMM) can help organizations improve their software security practices. Discuss the stages of security maturity and the steps to reach a high maturity level.	8	Evaluating	BTL5
5	**-**	Apply the concept of security maturity to a case study of an enterprise. Evaluate how the security maturity level impacts the organization's ability to prevent and respond to security breaches.	16	Applying	BTL4
6	**A**	Evaluate the effectiveness of an enterprise security framework in ensuring compliance with regulations like GDPR, HIPAA, or PCI DSS. Discuss the role of such frameworks in preventing legal and financial penalties.	8	Evaluating	BTL5
	B	Analyze the role of continuous security monitoring in maintaining a robust security framework. Discuss the technologies and strategies that can be used to ensure	8	Analysing	BTL4

		effective security monitoring.			
7	-	Evaluate the costs and benefits of implementing a security framework in an enterprise. Discuss how organizations can balance the need for security with budget constraints.	16	Evaluating	BTL5
8	**A**	Analyze the process of adopting an enterprise software security framework and its impact on an organization's overall business strategy. Discuss how security is integrated into the broader business goals.	8	Analysing	BTL4
	B	Evaluate the effectiveness of risk management strategies in an enterprise security framework. Discuss how risk assessment and mitigation practices help in achieving the right level of security.	8	Evaluating	BTL5
9	-	Discuss how the maturity of security practices within an organization influences its ability to adapt to new security threats. Evaluate strategies for improving the maturity level of security practices over time.	16	Analysing	BTL4
10	-	Evaluate the importance of security awareness and training in an enterprise software security framework. Discuss how training employees helps prevent security failures and improves overall system security.	16	Evaluating	BTL5
11	-	Discuss the challenges of scaling security practices in large organizations. Evaluate the strategies that can be employed to implement an enterprise security framework effectively across various departments and teams.	16	Evaluating	BTL5
12	-	Analyze the role of security testing and auditing within an enterprise software security framework. How do regular audits and testing contribute to maintaining a secure environment?	16	Analysing	BTL4
13	-	Evaluate the integration of security practices into agile and DevOps methodologies. Discuss how these methodologies can support or challenge the implementation of an enterprise software security framework.	16	Evaluating	BTL5
14		Apply the concept of 'how much security is enough?' to a case study of a startup or small business. Discuss how small businesses can adopt security practices proportionate to their resources while ensuring adequate protection.	16	Applying	BTL3
15	-	Evaluate the effectiveness of adopting an enterprise software security framework in an organization. Discuss the benefits and challenges of such frameworks, and how they help in improving an organization's security posture.	16	Evaluating	BTL5
16	-	Create a detailed security strategy for an enterprise, considering the factors that determine "how much security is enough?" Discuss how risk management and security maturity models can be applied to design a tailored security approach for an organization.	16	Creating	BTL6
17	-	Evaluate how security governance can be integrated with project management practices to ensure secure software development. Discuss the key practices and	16	Evaluating	BTL5

		policies that should be adopted to align security with organizational goals and project timelines.			
18		Discuss how security maturity models can guide an organization in improving its security practices over time. Evaluate the steps an organization should take to progress through various levels of security maturity.	16	Evaluating	BTL5
19		Create a comprehensive plan for integrating security practices throughout the SDLC in an enterprise, considering factors such as security governance, project management, and maturity models. Evaluate how this approach ensures the development of secure software systems.	16	Creating	BTL6