# Black & White JEC Logo.png

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CCS374 – Web Application Security Question Bank**

**III YEAR – 06th SEM BATCH: 2021 -2025**

**Vision of Institution**

To build Jeppiaar Engineering College as an Institution of Academic Excellence in Technical education and Management education and to become a World Class University.

**Mission of Institution**

|  |  |
| --- | --- |
| **M1** | To excel in teaching and **learning, research and innovation** by promoting the principles of scientific analysis and creative thinking |
| **M2** | To participate in the production, **development and dissemination of knowledge** and interact with **national and international communities** |
| **M3** | To equip students with **values, ethics and life skills** needed to enrich their lives and enable them to meaningfully contribute to the **progress of society**  |
| **M4** | To prepare students **for higher studies and lifelong learning**, enrich them with the **practical and entrepreneurial skills** necessary to excel as future professionals and contribute to **Nation’s economy** |

***Program Outcomes (POs)***

|  |  |
| --- | --- |
| **PO1** | **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| **PO2** | **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| **PO3** | **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations |
| **PO4** | **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| **PO5** | **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |
| **PO6** | **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. |
| **PO7** | **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **PO8** | **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| **PO9** | **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| **PO10** | **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| **PO11** | **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| **PO12** | **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

**Vision of Department**

To emerge as a globally prominent department, developing ethical computer professionals, innovators and entrepreneurs with academic excellence through quality education and research.

**Mission of Department**

|  |  |
| --- | --- |
| **M1** | To create **computer professionals** with an ability to identify and **formulate the engineering problems** and also to provide **innovative solutions** through **effective teaching learning process.** |
| **M2** | To **strengthen the core-competence** in computer science and engineering and to create an ability to **interact** effectively with industries. |
| **M3** | To produce engineers with good professional skills, **ethical values** and life skills for the **betterment of the society.** |
| **M4** | To encourage students towards **continuous and higher level learning** on technological advancements and provide a platform for **employment and self-employment.** |

#### Program Educational Objectives (PEOs)

|  |  |
| --- | --- |
| **PEO1** | **To address the real time complex engineering problems using innovative approach with strong core computing skills.** |
| **PEO2** | **To apply core-analytical knowledge and appropriate techniques and provide solutions to real time challenges of national and global society** |
| **PEO3** | **Apply ethical knowledge for professional excellence and leadership for the betterment of the society.** |
| **PEO4** | **Develop life-long learning skills needed for better employment and entrepreneurship** |

#### Program Specific Outcomes (PSOs)

Students will be able to

|  |  |
| --- | --- |
| **PSO1** | An ability to understand the core concepts of computer science and engineering and to enrich problem solving skills to analyze, design and implement software and hardware based systems of varying complexity. |
| **PSO2** | To interpret real-time problems with analytical skills and to arrive at cost effective and optimal solution using advanced tools and techniques. |
| **PSO3** | An understanding of social awareness and professional ethics with practical proficiency in the broad area of programming concepts by lifelong learning to inculcate employment and entrepreneurship skills. |

# SYLLABUS

# UNIT I FUNDAMENTALS OF WEB APPLICATION SECURITY

## Two-Mark Questions

1. What is the significance of recognizing web application security threats?

Recognizing threats helps in identifying vulnerabilities and implementing appropriate measures to mitigate risks.

1. Define Web Application Security.

Web application security refers to measures taken to protect web applications from threats and vulnerabilities, ensuring their integrity, confidentiality, and availability.

1. Differentiate between Authentication and Authorization.

Authentication verifies the identity of users, while authorization determines what actions users are allowed to perform.

1. What is Secure Socket Layer (SSL)?

SSL is a cryptographic protocol that provides secure communication over a computer network, commonly used to secure web traffic.

1. Explain Transport Layer Security (TLS).

TLS is a successor to SSL, providing secure communication over a computer network, including the internet, ensuring privacy and data integrity.

1. Why is session management crucial in web applications?

Session management is vital for maintaining user authentication across multiple requests and ensuring secure communication between the client and server.

1. What is the primary purpose of input validation in web applications?

Input validation helps to ensure that data entered by users is accurate, safe, and compliant with expected formats, thereby preventing security vulnerabilities such as injection attacks.

1. Name one common web application security threat. Cross-Site Scripting (XSS).
2. How does Cross-Site Request Forgery (CSRF) work?

CSRF involves an attacker tricking a user into unknowingly executing actions on a web application in which they are authenticated.

1. What role does encryption play in web application security?

Encryption helps in protecting sensitive data by encoding it in such a way that only authorized parties can access and understand it.

1. Explain the concept of SQL Injection.

SQL Injection is a web security vulnerability that allows attackers to interfere with the queries that an application makes to its database, potentially gaining unauthorized access to data.

1. Define Cross-Site Scripting (XSS).

XSS is a type of security vulnerability typically found in web applications where malicious scripts are injected into web pages viewed by other users.

1. What is the purpose of a Content Security Policy (CSP)?

CSP is a security standard that helps prevent various types of attacks, such as XSS and data injection, by allowing developers to specify which sources of content are allowed to be loaded on a web page.

1. What is the importance of security headers in web applications?

Security headers provide additional protection against various attacks by instructing web browsers on how to behave when interacting with a web application, enhancing its security posture.

1. Why is it essential to update software regularly in the context of web application security?

Regular updates help to patch known vulnerabilities and address security issues, reducing the risk of exploitation by attackers.

1. Explain the role of a Web Application Firewall (WAF).

A WAF monitors and filters HTTP traffic between a web application and the internet, providing an additional layer of security by blocking malicious requests and preventing attacks.

1. What is Cross-Origin Resource Sharing (CORS), and why is it relevant to web application security?

CORS is a security feature that allows web servers to specify which web applications are permitted to access their resources, helping to prevent cross-origin attacks.

1. Define Distributed Denial of Service (DDoS) attack.

A DDoS attack is a malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming it with a flood of internet traffic.

1. What are the key elements of a strong password policy?

Strong password policies typically include requirements such as minimum length, complexity, and periodic password changes to enhance security.

1. Explain the concept of Clickjacking.

Clickjacking is a technique used by attackers to trick users into clicking on malicious links or buttons by hiding them beneath legitimate clickable elements on a web page.

## Sixteen-Mark Questions

1. Discuss the evolution of web application security over the years, highlighting key milestones and challenges faced by developers.
2. Explain the various authentication mechanisms commonly employed in web applications, along with their strengths and weaknesses. Compare and contrast session-based and token-based authentication methods.
3. Describe the components and processes involved in Secure Socket Layer (SSL) and Transport Layer Security (TLS), elucidating how they ensure secure communication over the internet.
4. Critically analyze the importance of input validation in web application security, discussing common techniques and best practices for implementing robust input validation mechanisms. Provide examples of vulnerabilities that can be mitigated through effective input validation.
5. Examine the role of encryption in web application security, exploring different encryption algorithms and their applications in safeguarding sensitive data transmitted over networks. Discuss the trade-offs between symmetric and asymmetric encryption schemes.
6. Evaluate the significance of session management in web applications, outlining potential security risks associated with poor session handling practices. Discuss strategies for implementing secure session management mechanisms, including session tokens and session expiration policies.
7. Analyze the impact of common web application security threats such as Cross-Site Scripting (XSS), SQL Injection, and Cross-Site Request Forgery (CSRF) on the security posture of web applications. Propose mitigation strategies to address these threats effectively.
8. Discuss the importance of implementing security headers and Content Security Policy (CSP) directives in web applications to mitigate various types of attacks, including XSS and data injection. Provide examples of security headers and their respective functionalities.
9. Examine the role of Web Application Firewalls (WAFs) in protecting web applications from various threats, including SQL Injection, DDoS attacks, and malicious bots. Compare the effectiveness of network-based and host-based WAF deployments in different scenarios.
10. Critically evaluate the challenges associated with securing web applications in modern distributed environments, such as microservices architectures and cloud computing platforms. Discuss strategies for integrating security into the DevOps pipeline to ensure continuous security testing and deployment.

# UNIT II SECURE DEVELOPMENT AND DEPLOYMENT

## Two Mark Questions with Answers:

1. What is the purpose of security testing in web applications?

The purpose of security testing in web applications is to identify and mitigate vulnerabilities or weaknesses in the application's code or configuration before deployment.

1. Define Security Incident Response Planning.

Security Incident Response Planning involves developing procedures and protocols to detect, respond to, and recover from security incidents effectively, minimizing the impact on the organization's operations and data.

1. What is Microsoft Security Development Lifecycle (SDL)?

Microsoft SDL is a security assurance process that helps developers build more secure software by integrating security requirements and best practices into every phase of the software development lifecycle.

1. Explain OWASP CLASP.

OWASP CLASP (Comprehensive Lightweight Application Security Process) is a methodology that provides guidelines and best practices for integrating security into the software development process, focusing on simplicity and effectiveness.

1. Briefly define Software Assurance Maturity Model (SAMM).

SAMM is a framework for assessing and improving an organization's software security practices by providing a structured approach to building, measuring, and enhancing its software security capabilities over time.

1. Why is security testing crucial in the development of web applications?

Security testing is crucial in the development of web applications to identify and address vulnerabilities that could be exploited by attackers to compromise the confidentiality, integrity, or availability of the application and its data.

1. Describe the key components of Security Incident Response Planning.

Key components of Security Incident Response Planning include incident detection mechanisms, incident categorization, response procedures, communication protocols, escalation procedures, containment strategies, eradication measures, recovery plans, and post-incident review processes.

1. How does Microsoft SDL contribute to secure software development?

Microsoft SDL contributes to secure software development by providing a framework that integrates security practices, tools, and guidance into every phase of the software development lifecycle, helping developers identify and mitigate security risks effectively.

1. What are the primary objectives of OWASP CLASP?

The primary objectives of OWASP CLASP are to promote security awareness among developers, integrate security into the software development process, prioritize security activities based on risk, and facilitate the adoption of secure coding practices.

1. How does SAMM help organizations in improving their software security?

SAMM helps organizations improve their software security by providing a structured approach to assessing their current security posture, identifying areas for improvement, and implementing security initiatives based on industry best practices and organizational goals.

1. Mention a common technique used in security testing of web applications.

One common technique used in security testing of web applications is penetration testing, where security experts simulate real-world attacks to identify vulnerabilities and assess the effectiveness of security controls.

1. Outline the steps involved in Security Incident Response Planning. The steps involved in Security Incident Response Planning include:
	1. Establishing an incident response team
	2. Developing incident detection mechanisms
	3. Creating incident response procedures
	4. Implementing communication and escalation protocols
	5. Conducting regular training and drills
	6. Reviewing and updating the plan regularly.
2. What are some advantages of implementing Microsoft SDL?

Some advantages of implementing Microsoft SDL include improved software security, reduced risk of security breaches, enhanced trust and confidence from customers, and compliance with regulatory requirements.

1. What role does OWASP CLASP play in ensuring application security?

OWASP CLASP provides guidelines and best practices for integrating security into the software development process, helping developers identify and mitigate security risks effectively, thus ensuring the security of the application.

1. How does SAMM assess and enhance an organization's security posture?

SAMM assesses an organization's security posture by evaluating its software security practices against a set of predefined maturity levels and providing guidance on how to improve those practices over time, thus enhancing the organization's overall security posture.

1. Name a widely used tool for security testing in web applications.

One widely used tool for security testing in web applications is Burp Suite, which is a comprehensive platform for performing security testing, including scanning for vulnerabilities, analyzing application behavior, and testing the effectiveness of security controls.

1. What are the key considerations for effective Security Incident Response Planning?

Key considerations for effective Security Incident Response Planning include clear roles and responsibilities, rapid detection and response capabilities, effective communication channels, predefined escalation procedures, regular training and drills, and continuous improvement through post-incident review and analysis.

1. Discuss the phases of the Microsoft Security Development Lifecycle. The phases of the Microsoft Security Development Lifecycle include:
	1. Training and awareness
	2. Requirements
	3. Design
	4. Implementation
	5. Verification
	6. Release
	7. Response
	8. Security servicing.
2. What are some challenges associated with implementing OWASP CLASP?

Some challenges associated with implementing OWASP CLASP include resistance from developers to adopt new security practices, the complexity of integrating security into existing development processes, and the need for ongoing training and support to ensure successful implementation.

1. How does SAMM differ from other software security models?

SAMM differs from other software security models in its focus on maturity levels, its emphasis on building and measuring security capabilities over time, and its flexibility to accommodate different organizational contexts and development methodologies.

## Sixteen Mark Questions:

1. Explain the process of security testing in web applications, highlighting its importance in ensuring robust security measures.
2. Discuss the significance of Security Incident Response Planning in mitigating and managing security breaches in web applications. Provide a detailed framework for implementing an effective response plan.
3. Compare and contrast the Microsoft Security Development Lifecycle (SDL) and OWASP Comprehensive Lightweight Application Security Process (CLASP) in terms of their approaches, methodologies, and effectiveness in enhancing web application security.
4. Evaluate the strengths and weaknesses of the OWASP Comprehensive Lightweight Application Security Process (CLASP) and its applicability in diverse software development environments. Provide recommendations for overcoming potential limitations.
5. Assess the effectiveness of the Software Assurance Maturity Model (SAMM) in improving software security across different stages of the development lifecycle. Discuss its impact on organizational security practices and its alignment with industry standards and best practices.
6. Discuss the role of security testing tools in identifying vulnerabilities and ensuring the integrity of web applications. Provide examples of commonly used tools and their functionalities.
7. Outline the key components of a comprehensive Security Incident Response Plan (SIRP), detailing the steps involved in incident detection, analysis, containment, eradication, recovery, and post- incident review.
8. Explore the challenges and benefits associated with integrating security into the software development lifecycle through the Microsoft Security Development Lifecycle (SDL). Provide strategies for overcoming barriers to implementation and maximizing its effectiveness.
9. Explain how OWASP CLASP addresses the security concerns of web applications and enhances the overall security posture of an organization. Illustrate its implementation process and key considerations for successful adoption.
10. Critically analyze the Software Assurance Maturity Model (SAMM) as a framework for improving software security practices. Assess its scalability, flexibility, and adaptability to different organizational contexts, along with its potential impact on risk management and compliance efforts.

# UNIT III SECURE API DEVELOPMENT

## Two Mark Questions with Answers:

1. What is the purpose of session cookies in API security?

Session cookies are used to maintain user sessions and authenticate requests to API endpoints.

1. What authentication method relies on the exchange of tokens for access to APIs? Token-based authentication relies on the exchange of tokens for access to APIs.
2. How does rate limiting contribute to the availability of APIs?

Rate limiting helps prevent abuse or overload of APIs by limiting the number of requests a client can make within a specific time frame, thus ensuring availability.

1. What is the role of encryption in API security?

Encryption is used to protect data transmitted between clients and APIs from unauthorized access or interception.

1. Why is audit logging important in API security?

Audit logging helps track and monitor API activity, providing visibility into who accessed the API, when, and what actions were performed, aiding in security analysis and compliance.

1. What is the purpose of API keys in securing service-to-service APIs?

API keys are used to authenticate service-to-service API requests, ensuring that only authorized services can access the API.

1. Which authorization framework is commonly used for securing APIs and granting access to resources?

OAuth2 is commonly used for securing APIs and granting access to resources through authorization mechanisms like access tokens.

1. How does a service mesh contribute to securing microservice APIs?

A service mesh provides a dedicated infrastructure layer for handling communication between microservices, enabling features like encryption, authentication, and traffic control to enhance security.

1. What measure is used for locking down network connections in API security?

Firewalls and network policies are used for locking down network connections to control traffic flow and restrict access to API endpoints.

1. How are incoming requests secured in API development?

Incoming requests are secured through mechanisms like authentication, authorization, encryption, and validation to ensure that only authorized and valid requests are processed by the API.

1. What security mechanism is commonly used to maintain user sessions in API interactions?

Session cookies are commonly used to maintain user sessions in API interactions, ensuring stateful communication between clients and servers.

1. Explain the concept of token-based authentication in API security.

Token-based authentication involves the exchange of tokens (e.g., JSON Web Tokens) between clients and servers to authenticate and authorize API requests, eliminating the need for server-side sessions.

1. How does rate limiting contribute to the protection of APIs from abuse?

Rate limiting restricts the number of requests that can be made to an API within a specified time frame, preventing abuse or overload of the API infrastructure by limiting the rate at which requests can be processed.

1. What role does encryption play in securing data transmitted between clients and APIs?

Encryption ensures that data transmitted between clients and APIs is encrypted, preventing unauthorized access or interception of sensitive information during transmission.

1. Why is audit logging considered essential in API security?

Audit logging provides a record of API activity, including details such as who accessed the API, when, and what actions were performed, enabling security monitoring, analysis, and compliance enforcement.

1. How are API keys used to authenticate service-to-service API requests?

API keys are used as credentials to authenticate service-to-service API requests, allowing only authorized services with valid API keys to access the API endpoints.

1. What is OAuth2, and how does it contribute to securing APIs?

OAuth2 is an authorization framework that allows third-party applications to obtain limited access to a user's resources without exposing their credentials. It enhances API security by providing mechanisms for secure access delegation and resource authorization.

1. What are the primary security benefits of using a service mesh in microservice architectures?

The primary security benefits of using a service mesh in microservice architectures include centralized management of security policies, encryption of communication between services, authentication, and authorization mechanisms, and fine-grained traffic control.

1. How can network connections be locked down to enhance API security?

Network connections can be locked down by implementing firewall rules, network segmentation, and access control policies to restrict unauthorized access to API endpoints and resources.

1. What measures can be implemented to secure incoming requests in API development?

Incoming requests can be secured by implementing authentication mechanisms, authorization checks, input validation, encryption of sensitive data, and rate limiting to mitigate potential security threats and vulnerabilities.

## Sixteen Mark Questions:

1. Discuss the role of session cookies and token-based authentication in securing APIs. Compare and contrast their implementation, security implications, and suitability for different use cases in web application development.
2. Explain the concept of API security controls and their importance in addressing threats to API endpoints. Provide examples of security controls such as rate limiting, encryption, and audit logging, and discuss how they contribute to mitigating common API vulnerabilities.
3. Evaluate the effectiveness of different authentication mechanisms, including API keys and OAuth2, in securing service-to-service APIs. Discuss their strengths, weaknesses, and suitability for various deployment scenarios, considering factors such as scalability, manageability, and security requirements.
4. Explore the challenges and benefits of securing microservice APIs using a service mesh architecture. Discuss how service mesh technologies facilitate secure communication, traffic management, and observability in distributed microservice environments, and assess their impact on overall system reliability and security posture.
5. Analyze the importance of locking down network connections in API security. Discuss strategies for implementing network-level security measures such as firewalls, network segmentation, and access control policies to protect API endpoints from unauthorized access and malicious attacks.
6. Discuss the significance of securing incoming requests in API development and explore the various techniques and best practices for achieving this goal. Provide examples of security mechanisms such as input validation, parameterized queries, and content filtering, and explain how they help prevent common security threats such as injection attacks and cross-site scripting (XSS).
7. Critically evaluate the role of session management in API security. Discuss the challenges associated with session management in distributed environments, such as stateless APIs and microservices, and propose strategies for implementing secure session handling mechanisms to protect against session hijacking and other session-related vulnerabilities.
8. Compare and contrast token-based authentication with other authentication mechanisms, such as HTTP basic authentication and API keys. Discuss the advantages and disadvantages of each approach in terms of security, scalability, and ease of implementation, and provide recommendations for selecting the most appropriate authentication method based on specific use case requirements.
9. Explore the security considerations and best practices for implementing encryption in API development. Discuss the different types of encryption algorithms and protocols commonly used to protect data in transit and at rest, and provide guidelines for selecting and configuring encryption mechanisms to ensure confidentiality, integrity, and authenticity of API communications and data.
10. Assess the role of audit logging in API security and compliance. Discuss the importance of maintaining detailed audit logs to track API activity, detect security incidents, and demonstrate regulatory compliance, and provide recommendations for designing and implementing effective audit logging mechanisms that meet the requirements of various security standards and regulations.

# UNIT IV VULNERABILITY ASSESSMENT AND PENETRATION TESTING

## Two Mark Questions with Answers:

1. What is the Vulnerability Assessment Lifecycle?

The Vulnerability Assessment Lifecycle consists of four stages: Preparation, Identification, Remediation, and Verification.

1. Name a type of vulnerability assessment tool used for scanning vulnerabilities in cloud environments.

Cloud-based vulnerability scanners are used for scanning vulnerabilities in cloud environments.

1. Which type of vulnerability scanner focuses on identifying security flaws in individual computer systems?

Host-based vulnerability scanners focus on identifying security flaws in individual computer systems.

1. What is the primary purpose of network-based vulnerability scanners?

The primary purpose of network-based vulnerability scanners is to identify vulnerabilities in network devices, such as routers, switches, and firewalls.

1. Which type of vulnerability scanner specializes in identifying security weaknesses in database systems?

Database-based vulnerability scanners specialize in identifying security weaknesses in database systems.

1. Name a common type of penetration test used to assess the security of external network infrastructure.

External Testing is a common type of penetration test used to assess the security of external network infrastructure.

1. What aspect of systems does Web Application Testing penetration tests focus on?

Web Application Testing penetration tests focus on assessing the security of web applications and their underlying infrastructure.

1. In penetration testing, what does SSID or Wireless Testing focus on?

SSID or Wireless Testing in penetration testing focuses on assessing the security of wireless networks and their associated configurations.

1. Which type of penetration test evaluates the security of internal network infrastructure? Internal Penetration Testing evaluates the security of internal network infrastructure.
2. What is the primary objective of Mobile Application Testing in penetration testing?

The primary objective of Mobile Application Testing in penetration testing is to assess the security of mobile applications and their interaction with backend services.

1. What are the stages involved in the Vulnerability Assessment Lifecycle?

The stages involved in the Vulnerability Assessment Lifecycle are Preparation, Identification, Remediation, and Verification.

1. Name a type of vulnerability assessment tool commonly used for scanning vulnerabilities in cloud environments.

Cloud-based vulnerability scanners are commonly used for scanning vulnerabilities in cloud environments.

1. Which type of vulnerability scanner focuses on identifying security weaknesses in individual computer systems?

Host-based vulnerability scanners focus on identifying security weaknesses in individual computer systems.

1. What is the primary purpose of network-based vulnerability scanners?

The primary purpose of network-based vulnerability scanners is to identify vulnerabilities in network infrastructure and devices.

1. Which type of vulnerability scanner specializes in identifying security vulnerabilities in database systems?

Database-based vulnerability scanners specialize in identifying security vulnerabilities in database systems.

1. Name a common type of penetration test used to assess the security of external network infrastructure.

External Testing is a common type of penetration test used to assess the security of external network infrastructure.

1. What aspect of systems does Web Application Testing penetration tests focus on?

Web Application Testing penetration tests focus on assessing the security of web applications and their underlying infrastructure.

1. In penetration testing, what does SSID or Wireless Testing focus on?

SSID or Wireless Testing in penetration testing focuses on assessing the security of wireless networks and their configurations.

1. Which type of penetration test evaluates the security of internal network infrastructure? Internal Penetration Testing evaluates the security of internal network infrastructure.
2. What is the primary objective of Mobile Application Testing in penetration testing?

The primary objective of Mobile Application Testing in penetration testing is to assess the security of mobile applications and their interaction with backend services.

## Sixteen Mark Questions:

1. Discuss the Vulnerability Assessment Lifecycle in detail, outlining each stage's significance and activities involved. Provide examples of tools and techniques commonly used in each stage to effectively identify, remediate, and verify vulnerabilities within an organization's infrastructure.
2. Compare and contrast various types of vulnerability assessment tools, including cloud-based, host-based, network-based, and database-based scanners. Evaluate their strengths, weaknesses, and suitability for different environments and scenarios, considering factors such as scalability, accuracy, and ease of use.
3. Explore the importance of penetration testing in identifying and mitigating security risks within an organization's infrastructure. Discuss the different types of penetration tests, including External Testing, Web Application Testing, Internal Penetration Testing, SSID or Wireless Testing, and Mobile Application Testing, and provide examples of when each type should be employed.
4. Evaluate the effectiveness of External Testing as a penetration testing technique for assessing the security of an organization's external network infrastructure. Discuss the methodology, tools, and best practices involved in conducting External Testing, and provide recommendations for addressing common challenges and limitations.
5. Discuss the significance of Web Application Testing in penetration testing and its role in identifying security vulnerabilities in web applications and their underlying infrastructure. Explore common attack vectors and techniques used in Web Application Testing, and provide recommendations for securing web applications against potential threats.
6. Analyze the challenges and considerations involved in conducting Internal Penetration Testing to assess the security of an organization's internal network infrastructure. Discuss the methodology, scope, and limitations of Internal Penetration Testing, and provide strategies for overcoming common obstacles and ensuring comprehensive coverage.
7. Explore the importance of SSID or Wireless Testing in penetration testing and its role in assessing the security of wireless networks and their configurations. Discuss the methodology, tools, and techniques used in SSID or Wireless Testing, and provide recommendations for securing wireless networks against potential threats.
8. Evaluate the significance of Mobile Application Testing in penetration testing and its role in identifying security vulnerabilities in mobile applications and their interaction with backend services. Discuss common security challenges faced by mobile applications, and provide strategies for securing mobile applications against potential threats.
9. Discuss the role of audit logging and reporting in vulnerability assessment and penetration testing. Explore the importance of documenting findings, vulnerabilities, and remediation efforts, and provide recommendations for effectively communicating assessment results to stakeholders and decision-makers.
10. Critically assess the overall effectiveness of vulnerability assessment and penetration testing as security measures in mitigating cybersecurity risks within an organization. Discuss their strengths, weaknesses, and limitations, and provide recommendations for integrating them into a comprehensive security strategy to enhance overall cyber resilience

# UNIT V HACKING TECHNIQUES AND TOOLS

## Two Mark Questions with Answers:

1. What is Social Engineering in the context of hacking?

Social Engineering is the manipulation of individuals to divulge confidential information or perform actions that compromise security.

1. What type of attack is commonly associated with injecting malicious code into databases?

Injection attacks, such as SQL injection, involve injecting malicious code into databases to manipulate or retrieve data.

1. What vulnerability does Cross-Site Scripting (XSS) exploit?

Cross-Site Scripting (XSS) exploits vulnerabilities in web applications to inject malicious scripts into web pages viewed by other users.

1. What security risk arises from broken authentication and session management?

Broken authentication and session management can lead to unauthorized access to accounts and sensitive data due to vulnerabilities in authentication and session handling mechanisms.

1. What type of attack involves tricking a user into unknowingly performing actions on a web application?

Cross-Site Request Forgery (CSRF) involves tricking a user into unknowingly performing actions on a web application in which they are authenticated.

1. What is the consequence of security misconfiguration in web applications?

Security misconfiguration can lead to unintended exposure of sensitive data or unauthorized access due to improperly configured security settings.

1. What vulnerability is exploited when sensitive data is stored in an insecure manner?

Insecure cryptographic storage occurs when sensitive data, such as passwords or credit card numbers, is stored without proper encryption, making it vulnerable to unauthorized access.

1. What risk is associated with failure to restrict URL access?

Failure to restrict URL access can lead to unauthorized access to sensitive information or functionality by users who are not authorized to access it.

1. Name a vulnerability assessment tool commonly used for network scanning.

OpenVAS (Open Vulnerability Assessment System) is a commonly used tool for network scanning and vulnerability assessment.

1. What role does Burp Suite play in the field of cybersecurity?

Burp Suite is a comprehensive platform for web application security testing, including scanning for vulnerabilities, analyzing application behavior, and testing the effectiveness of security controls.

1. Define Social Engineering and its significance in cybersecurity.

Social Engineering is a psychological manipulation technique used by attackers to deceive individuals into divulging confidential information or performing actions that compromise security. It is significant in cybersecurity as it exploits human vulnerabilities, often being the weakest link in the security chain.

1. What are the common types of injection attacks, and how do they exploit vulnerabilities?

Common types of injection attacks include SQL injection, LDAP injection, and XML injection. These attacks exploit vulnerabilities in input validation mechanisms to inject malicious code or commands into systems, potentially leading to data breaches or system compromise.

1. Discuss the impact of Cross-Site Scripting (XSS) attacks on web applications and users. Cross-Site Scripting (XSS) attacks can have various impacts on web applications and users,

including theft of sensitive information, session hijacking, defacement of web pages, and the execution of malicious scripts within the context of a trusted website.

1. How do broken authentication and session management vulnerabilities compromise the security of web applications?

Broken authentication and session management vulnerabilities can compromise the security of web applications by allowing attackers to bypass authentication controls, hijack user sessions, or gain unauthorized access to sensitive data or functionality.

1. Explain the concept of Cross-Site Request Forgery (CSRF) attacks and their potential consequences.

Cross-Site Request Forgery (CSRF) attacks involve tricking authenticated users into unknowingly executing unauthorized actions on a web application. The consequences of CSRF attacks may include unauthorized transactions, data manipulation, or account compromise.

1. What are the risks associated with security misconfiguration in web applications?

Security misconfiguration in web applications can lead to various risks, such as unauthorized access to sensitive data, exposure of system information, increased attack surface, and compliance violations.

1. Discuss the importance of secure cryptographic storage in protecting sensitive data.

Secure cryptographic storage is crucial for protecting sensitive data from unauthorized access or disclosure. By encrypting data at rest, organizations can mitigate the risk of data breaches and maintain compliance with security standards and regulations.

1. How does the failure to restrict URL access pose a security risk to web applications?

Failure to restrict URL access can allow unauthorized users to access sensitive information or perform actions that are not intended for their role or privilege level, potentially leading to data breaches or unauthorized system manipulation.

1. Name a tool used for vulnerability scanning that focuses on web application security.

Nikto is a widely used tool for web application vulnerability scanning, focusing on identifying common security issues such as outdated software versions, misconfigurations, and vulnerabilities in web servers and applications.

1. How does Comodo contribute to cybersecurity efforts?

Comodo provides cybersecurity solutions, including antivirus software, endpoint protection, secure web browsing, and SSL certificates, to help organizations protect against malware, phishing attacks, and other cyber threats.

## Sixteen Mark Questions:

1. Explore the techniques and psychological principles behind Social Engineering attacks, and discuss their effectiveness in bypassing traditional cybersecurity defenses. Provide real-world examples of Social Engineering attacks and analyze their impact on organizations and individuals.
2. Discuss the various types of injection attacks, including SQL injection, LDAP injection, and XML injection, and explain how they exploit vulnerabilities in web applications. Evaluate the severity of injection attacks in terms of potential damage and provide recommendations for mitigating these risks.
3. Analyze the prevalence of Cross-Site Scripting (XSS) vulnerabilities in web applications and their impact on security. Discuss the different types of XSS attacks, such as reflected XSS, stored XSS, and DOM-based XSS, and provide strategies for detecting, preventing, and mitigating XSS vulnerabilities.
4. Evaluate the risks associated with broken authentication and session management vulnerabilities in web applications. Discuss common causes of these vulnerabilities, such as weak passwords, session fixation, and insufficient session expiration policies, and provide best practices for improving authentication and session management security.
5. Discuss the concept of Cross-Site Request Forgery (CSRF) attacks and their potential impact on web application security. Explore common CSRF attack scenarios, such as CSRF with GET requests, CSRF with POST requests, and CSRF with AJAX requests, and provide recommendations for preventing and mitigating CSRF vulnerabilities.
6. Explore the risks associated with security misconfigurations in web applications and their impact on cybersecurity. Discuss common security misconfigurations, such as default credentials, directory listings, and improper file permissions, and provide guidelines for identifying and remediating these misconfigurations effectively.
7. Analyze the importance of secure cryptographic storage in protecting sensitive data from unauthorized access and disclosure. Discuss common cryptographic storage techniques, such as hashing, encryption, and salting, and provide best practices for implementing secure cryptographic storage mechanisms in web applications.
8. Discuss the risks associated with failure to restrict URL access in web applications and their potential impact on security. Explore common URL access control vulnerabilities, such as predictable resource locations, insecure direct object references (IDOR), and lack of access controls, and provide strategies for mitigating these risks.
9. Evaluate the effectiveness of vulnerability assessment tools such as OpenVAS, Nexpose, and Nikto in identifying and mitigating security vulnerabilities in web applications and network infrastructure. Compare and contrast the features, capabilities, and limitations of these tools, and provide recommendations for selecting the most appropriate tool for a given scenario.
10. Critically assess the role of penetration testing tools such as Burp Suite in identifying security vulnerabilities and weaknesses in web applications. Discuss the features and functionalities of Burp Suite, including its proxy, scanner, and intruder modules, and provide recommendations for leveraging its capabilities to conduct comprehensive penetration tests effectively.