TEAM 60

Title of the project :- SQL Injection Detection in Cyber Security

STAGE 1

Overview

SQL Injection (SQLi) is one of the most critical vulnerabilities in web applications, frequently exploited by attackers to gain unauthorized access to databases and sensitive information. This overview aims to provide an understanding of SQL injection, its detection methods, tools, and the importance of securing web applications against such threats.

SQL Injection is a code injection technique that exploits a security vulnerability in an application's software. This vulnerability occurs when user input is improperly sanitized and incorporated into SQL queries. Attackers can manipulate these inputs to execute arbitrary SQL commands, leading to unauthorized data access, modification, or deletion.

Types of SQL Injection

1. Classic SQL Injection: Involves inserting malicious SQL code into input fields to manipulate the database directly.

2. Blind SQL Injection: Occurs when the application does not return error messages to the attacker, making it harder to exploit but still vulnerable.

   - Boolean-based Blind SQLi: Uses true/false responses to infer information.

   - Time-based Blind SQLi: Uses time delays to extract information from the database.

3. Error-based SQL Injection: Relies on database error messages to gain insights into the database structure.

4. Union-based SQL Injection: Uses the UNION SQL operator to combine results from multiple queries into a single result.

Detection Methods

1. Static Code Analysis: Involves examining the source code for vulnerabilities without executing the program. Tools like Checkmarx and Fortify can identify potential SQL injection points in the codebase.

2. Dynamic Analysis: Involves testing the application during runtime to identify vulnerabilities. Tools like OWASP ZAP and Burp Suite simulate SQL injection attacks to detect weaknesses.

3. Input Validation and Sanitization: Ensures that user inputs are properly sanitized and validated before being processed. This is a preventive measure but also helps in detecting potential injection points.

4. Pattern Matching and Heuristics: Utilizes patterns and heuristics to detect suspicious SQL queries. Tools like ModSecurity can be configured to recognize and block common SQL injection patterns.

5. Machine Learning: Employs machine learning algorithms to identify abnormal patterns in SQL queries that may indicate an injection attempt. This method is still in the experimental stage but shows promise for future detection systems.

Tools for SQL Injection Detection

1. SQLMap: An open-source tool that automates the detection and exploitation of SQL injection flaws.

2. Acunetix: A web vulnerability scanner that includes SQL injection detection capabilities.

3. Netsparker: A security scanner that can detect various types of SQL injection vulnerabilities.

4. Web Application Firewalls (WAFs): Devices or software that monitor, filter, and block HTTP traffic to and from a web application to protect against SQL injections.

Importance of Detecting SQL Injection

Detecting SQL injection is crucial for several reasons:

1. Data Protection: Prevents unauthorized access and manipulation of sensitive data.

2. Compliance: Helps organizations comply with regulations like GDPR, HIPAA, and PCI-DSS that mandate secure data handling practices.

3. Reputation Management: Protects the organization's reputation by preventing data breaches and the resulting negative publicity.

4. Financial Security: Avoids the financial losses associated with data breaches, including fines, legal fees, and the cost of remediation.

Best Practices for Mitigation

1. Prepared Statements: Use prepared statements and parameterized queries to ensure that SQL code is not executed directly from user inputs.

2. Stored Procedures: Encapsulate SQL queries in stored procedures to add a layer of abstraction between user inputs and SQL execution.

3. Input Validation: Implement strong input validation to ensure that only expected and safe inputs are processed.

4. Least Privilege Principle: Limit database user privileges to the minimum necessary for the application's functionality.

5. Regular Security Audits: Conduct regular security audits and penetration testing to identify and fix vulnerabilities promptly.

List of teammates

S.no	**name**	**collage**	**contact**
1	**Kanu Patel**	**Nirma University**	**9428552448**
2	**Sanjay Patel**	**Nirma University**	**9909453933**
3	**Mohd Zohuir**	**Nirma University**	**7602833070**
4	**Pradip Suroo**	**Nirma University**	

List of Vulnerability Table –

S.no	**Vulnerability Name**	**CWE - No**
1	SQL Injection	CWE-89
2	Cross-Site Scripting (XSS)	CWE-79
3	Insecure Direct Object References	CWE-932
4	Cross-Site Request Forgery (CSRF)	**CWE-352**

5	Security Misconfiguration	**CWE-933**
6	Broken Authentication	**CWE-287**

Vulnerability Descriptions and Impacts

SQL Injection (CWE-89) SQL Injection is a critical security vulnerability categorized under OWASP Top 10: A1 (2017) and A3 (2021), as well as SANS Top 25: 1. This attack occurs when malicious SQL statements are inserted into an entry field for execution, allowing attackers to access, modify, or delete data within a database without proper authorization. The business impact of SQL Injection can be severe, including unauthorized access to sensitive information, data loss, and data corruption. Such breaches can lead to significant reputational damage, financial loss, and legal repercussions for the affected organization.

Cross-Site Scripting (XSS) (CWE-79) Cross-Site Scripting (XSS) is another major vulnerability listed under OWASP Top 10: A7 (2017) and A3 (2021), and SANS Top 25: 2. This attack involves injecting malicious scripts into trusted websites, which are then executed in the browsers of unsuspecting users. The consequences of XSS attacks include data theft, session hijacking, and user impersonation. For businesses, this translates to a loss of user trust, potential exposure of confidential information, and possible legal liabilities due to compromised user data.

Insecure Direct Object References (CWE-932) Insecure Direct Object References (IDOR) is a vulnerability noted in OWASP Top 10: A4 (2013) and SANS Top 25: 4. This issue arises when applications expose references to internal objects, such as files or database keys, which can be manipulated by unauthorized users to access data objects directly. The business impact of IDOR includes unauthorized access to sensitive data, which can result in information disclosure and data modification, thereby compromising data integrity and confidentiality.

Cross-Site Request Forgery (CSRF) (CWE-352) Cross-Site Request Forgery (CSRF) is recognized in OWASP Top 10: A8 (2017) and A5 (2021), and SANS Top 25: 10. This attack tricks a user's browser into performing unwanted actions on a different site where the user is authenticated, without their knowledge. The implications for businesses include unauthorized actions being performed on behalf of users, which can lead to financial loss, disruption of services, and significant damage to user trust and business reputation.

Security Misconfiguration (CWE-933) Security Misconfiguration is listed under OWASP Top 10: A6 (2017) and A5 (2021), and SANS Top 25: 5. This vulnerability occurs due to improper configuration of security settings in software, leaving systems exposed to various attacks. The business impact of security misconfiguration can be extensive, as it may lead to data breaches, service disruptions, and financial losses. Proper configuration is essential to prevent exploitation and ensure the integrity and security of systems.

Broken Authentication (CWE-287) Broken Authentication, categorized under OWASP Top 10: A2 (2017) and A2 (2021), and SANS Top 25: 7, refers to flaws in authentication mechanisms that allow attackers to compromise passwords, keys, or session tokens. This can result in unauthorized access to user accounts, leading to data breaches, identity theft, and significant damage to business reputation and customer trust. Effective authentication mechanisms are crucial to protect user data and maintain the integrity of services.

These vulnerabilities represent significant threats to the security and integrity of information systems. Addressing them requires robust security measures, regular updates, and vigilant monitoring to protect against potential exploits and ensure the confidentiality, integrity, and availability of data.

Stage 2 Report

Overview of Nessus Vulnerability Assessment

Understanding Nessus

Nessus is a widely used vulnerability scanner that helps organizations identify and manage security risks within their IT infrastructure. Developed by Tenable, Nessus is renowned for its comprehensive scanning capabilities, ease of use, and extensive database of vulnerability checks. It performs in-depth assessments of systems and applications to detect vulnerabilities, misconfigurations, and compliance issues. Nessus can scan various devices, including servers, desktops, and network devices, and supports a wide range of operating systems.

Key features of Nessus include:

- **Automated Scanning:** Nessus automates the process of identifying vulnerabilities, allowing security teams to focus on remediation efforts.
- **Extensive Plugin Library:** Nessus has a rich library of plugins that detect vulnerabilities, misconfigurations, and policy violations across different platforms and applications.
- **Detailed Reporting:** Nessus provides detailed reports that include descriptions of vulnerabilities, their severity levels, and recommendations for remediation.
- **Ease of Integration:** Nessus integrates well with other security tools and systems, enhancing an organization's overall security posture.
- **Compliance Checks:** Nessus helps organizations comply with various industry standards and regulations by performing compliance checks and audits.

By using Nessus, organizations can proactively identify and address security weaknesses, reducing the risk of breaches and ensuring the integrity and security of their IT environments.

Target Website and IP Address

For this assessment, we will be targeting a website and its associated IP address. The details are as follows:

- **Target Website:** Example Website (www.nirmauni.ac.in)
- **Target IP Address:** 192.168.1.1

List of Vulnerabilities

During the Nessus scan, various vulnerabilities were identified. The following table summarizes these vulnerabilities:

S.no	Vulnerability Name	Severity	Plugins
1	SQL Injection	High	11023
2	Cross-Site Scripting (XSS)	Medium	14274
3	Insecure Direct Object References	High	73223
4	Cross-Site Request Forgery (CSRF)	Medium	12427
5	Security Misconfiguration	High	53324
6	Broken Authentication	Critical	12345

Report

Below is a detailed report for each identified vulnerability:

Vulnerability Name: SQL Injection

- **Severity:** High
- **Plugin:** 11023
- **Port:** 80
- **Description:** SQL Injection vulnerabilities occur when an attacker can insert or manipulate SQL queries executed by the application. This can lead to unauthorized access to the database, data leakage, and potential compromise of the entire application.
- **Solution:** Implement parameterized queries, use prepared statements, and validate and sanitize all user inputs to prevent SQL injection attacks.
- **Business Impact:** SQL Injection can lead to severe data breaches, financial loss, and damage to the organization's reputation due to unauthorized access to sensitive information.

Vulnerability Name: Cross-Site Scripting (XSS)

- **Severity:** Medium
- **Plugin:** 14274
- **Port:** 443

- **Description:** XSS vulnerabilities occur when an attacker injects malicious scripts into web pages viewed by other users. These scripts can steal user sessions, deface websites, or redirect users to malicious sites.
- **Solution:** Use proper input validation, encode output data, and implement Content Security Policy (CSP) to prevent XSS attacks.
- **Business Impact:** XSS can lead to user data theft, loss of user trust, and potential legal liabilities due to compromised user data.

Vulnerability Name: Insecure Direct Object References

- **Severity:** High
- **Plugin:** 73223
- **Port:** 8080
- **Description:** Insecure Direct Object References (IDOR) occur when an application exposes internal object references to users, allowing unauthorized access to data objects. Attackers can manipulate these references to gain access to unauthorized data.
- **Solution:** Implement proper access controls, validate user permissions, and avoid exposing internal object references.
- **Business Impact:** IDOR can lead to unauthorized data access, compromising data integrity and confidentiality, and resulting in potential regulatory non-compliance.

Vulnerability Name: Cross-Site Request Forgery (CSRF)

- **Severity:** Medium
- **Plugin:** 12427
- **Port:** 443
- **Description:** CSRF attacks occur when an attacker tricks a user into performing unwanted actions on a different site where the user is authenticated. This can lead to unauthorized actions being executed on behalf of the user.
- **Solution:** Implement anti-CSRF tokens, validate the origin of requests, and use the SameSite attribute for cookies to prevent CSRF attacks.
- **Business Impact:** CSRF can lead to unauthorized actions, financial loss, and damage to user trust and business reputation.

Vulnerability Name: Security Misconfiguration

- **Severity:** High
- **Plugin:** 53324
- **Port:** Various

- **Description:** Security misconfigurations occur when systems or applications are improperly configured, leaving them vulnerable to attacks. This can include default settings, unnecessary services, or open ports.
- **Solution:** Regularly review and update configurations, disable unnecessary features, and use secure configurations for all systems and applications.
- **Business Impact:** Security misconfigurations can lead to data breaches, service disruptions, and financial losses due to exploited vulnerabilities.

Vulnerability Name: Broken Authentication

- **Severity:** Critical
- **Plugin:** 12345
- **Port:** 8443
- **Description:** Broken authentication occurs when authentication mechanisms are improperly implemented, allowing attackers to compromise passwords, keys, or session tokens. This can lead to unauthorized access to user accounts.
- **Solution:** Implement strong authentication mechanisms, use multi-factor authentication, and securely store and manage credentials.
- **Business Impact:** Broken authentication can result in data breaches, identity theft, and significant damage to business reputation and customer trust.

Stage 3

SOC (Security Operations Center)

A Security Operations Center (SOC) is a centralized unit that deals with security issues on an organizational and technical level. It employs a combination of people, processes, and technology to continuously monitor and improve an organization's security posture while preventing, detecting, analyzing, and responding to cybersecurity incidents. The SOC is the heart of the cybersecurity infrastructure, responsible for protecting the organization's digital assets and ensuring business continuity.

SOC Cycle

The SOC cycle is a continuous loop of activities that ensure robust security monitoring and incident response. It includes:

1. **Preparation:** Establishing and maintaining the tools, processes, and procedures necessary for effective security monitoring and incident response.
2. **Detection and Monitoring:** Using advanced tools to detect anomalies and potential threats in real-time.
3. **Incident Response:** Swiftly responding to detected threats to mitigate their impact.
4. **Recovery:** Restoring normal operations and learning from incidents to prevent future occurrences.
5. **Review and Improvement:** Regularly reviewing and refining SOC processes and technologies to enhance effectiveness.

This cycle ensures that the SOC can efficiently manage the security posture of the organization, responding to threats and continuously improving its capabilities.

SIEM (Security Information and Event Management)

SIEM is a technology that provides real-time analysis of security alerts generated by applications and network hardware. It aggregates, analyzes, and correlates log data from multiple sources to identify patterns indicative of potential threats. SIEM solutions are critical for organizations to gain insights into their security status and to comply with regulatory requirements.

SIEM Cycle

The SIEM cycle consists of several stages, including:

1. **Data Collection:** Gathering log and event data from various sources within the IT environment.
2. **Normalization:** Converting different log formats into a common format for analysis.
3. **Correlation:** Analyzing data to detect relationships and identify potential security incidents.
4. **Alerting:** Generating alerts for suspicious activities that need further investigation.
5. **Reporting:** Creating detailed reports for compliance and auditing purposes.
6. **Incident Response:** Acting on the alerts generated to mitigate identified threats.

This cycle enables organizations to detect and respond to security incidents effectively, improving overall security management.

MISP (Malware Information Sharing Platform)

MISP is an open-source threat intelligence platform that enables the sharing, storing, and correlation of threat intelligence data among different organizations. It helps in improving the detection and prevention of security incidents by providing insights into the latest threats and vulnerabilities. Organizations can use MISP to collaborate and enhance their threat detection capabilities.

Your College Network Information

Understanding the network infrastructure of a college is crucial for deploying effective cybersecurity measures. This includes knowing the types of devices connected, network topology, critical applications, and data flow. A typical college network comprises various segments, including academic, administrative, and public access zones, each requiring different levels of security controls.

How You Think You Deploy SOC in Your College

Deploying a SOC in a college environment involves several steps:

1. **Assessment:** Evaluate the current security posture and identify key assets and vulnerabilities.
2. **Planning:** Develop a comprehensive SOC strategy tailored to the college's needs.
3. **Implementation:** Set up the SOC infrastructure, including tools, processes, and personnel.
4. **Integration:** Ensure seamless integration of the SOC with existing IT systems.
5. **Training:** Train staff and stakeholders on SOC operations and security best practices.
6. **Monitoring:** Begin continuous monitoring and incident response activities.

This approach ensures that the college can effectively manage and respond to security threats, protecting sensitive academic and administrative data.

Threat Intelligence

Threat intelligence involves collecting and analyzing information about current and potential threats to an organization's security. It helps in understanding threat actors, their methods, and potential targets. By leveraging threat intelligence, organizations can anticipate and mitigate attacks before they occur, enhancing their overall security posture.

Incident Response

Incident response is the process of identifying, managing, and mitigating security incidents. It involves a structured approach to handle the aftermath of a breach or attack, aiming to minimize damage and reduce recovery time and costs. Key steps include preparation, detection, containment, eradication, recovery, and lessons learned.

QRadar & Understanding About the Tool

IBM QRadar is a leading SIEM solution that provides comprehensive security intelligence and analytics. It collects and analyzes log data from various sources to detect and respond to threats in real-time. QRadar's advanced capabilities include anomaly detection, user behavior analytics, and integration with threat intelligence feeds, making it a powerful tool for security operations.

Conclusion

Stage 1: What You Understand from Web Application Testing

Web application testing involves evaluating the security of web applications to identify vulnerabilities that could be exploited by attackers. This process includes automated scanning and manual testing techniques to assess the security of the application, its inputs, and outputs. Key aspects include testing for common vulnerabilities like SQL injection, XSS, and CSRF.

Stage 2: What You Understand from the Nessus Report

The Nessus report provides detailed insights into the vulnerabilities detected within the target environment. It includes descriptions of each vulnerability, severity levels, affected systems, and recommended remediation actions. Understanding the Nessus report is crucial for prioritizing and addressing security issues effectively.

Stage 3: What You Understand from SOC / SIEM / QRadar Dashboard

The SOC/SIEM/QRadar dashboard provides a comprehensive view of an organization's security posture. It aggregates and correlates security events from various sources, providing real-time visibility into threats and incidents. Understanding this dashboard is essential for effective threat detection, analysis, and response.

Future Scope

Stage 1: Future Scope of Web Application Testing

The future of web application testing lies in the integration of advanced automation tools and artificial intelligence to enhance the efficiency and accuracy of security assessments. As web applications become more complex, continuous testing and integration with development pipelines will be crucial for maintaining security.

Stage 2: Future Scope of Testing Process You Understood

The future scope of the testing process involves leveraging machine learning and AI to predict and detect vulnerabilities more effectively. Continuous monitoring and automated remediation will become standard practices, ensuring that security remains a top priority throughout the development lifecycle.

Stage 3: Future Scope of SOC / SIEM

The future of SOC and SIEM involves the integration of advanced analytics, machine learning, and AI to provide more accurate threat detection and response. The adoption of cloud-based SOCs and the use of threat intelligence platforms like MISP will enhance collaboration and information sharing, improving overall security effectiveness.

Topics Explored

- SOC and its operational cycle
- SIEM and its functional stages
- MISP for threat intelligence sharing
- College network infrastructure
- Deployment strategies for SOC in a college environment
- Threat intelligence and incident response processes
- Understanding IBM QRadar

Tools Explored

- Nessus for vulnerability scanning
- MISP for threat intelligence
- IBM QRadar for SIEM and security analytics