Stage – 1: Vulnerability Assessment Report

Target Website: http://www.itsecgames.com/

Vulnerability Overview Table

Sr. No	Vulnerability Type	CWE ID
1	Insecure Direct Object References (IDOR)	CWE-639
2	Cross-Site Request Forgery (CSRF)	CWE-352
3	Improper Security Configuration	CWE-16
4	Unchecked URL Redirects and Forwards	CWE-601
5	XML External Entity (XXE) Vulnerability	CWE-611

Detailed Vulnerability Reports

1. Insecure Direct Object References (IDOR)

- **CWE:** CWE-639
- **Category:** Broken Access Control (OWASP/SANS A01:2021)
- **Overview:**
  The application is prone to IDOR flaws, allowing attackers to modify URL parameters (for instance, altering account_id values) to access unauthorized data.
- **Business Impact:**
  - Exposure of confidential user data
  - Unauthorized changes to records
  - Violations of privacy
- **Method of Discovery:**
  - Intercepted HTTP traffic using Burp Suite
  - Manually altered identifier parameters to test for data access
  - Confirmed unauthorized access through response validation

2. Cross-Site Request Forgery (CSRF)

- **CWE:** CWE-352
- **Category:** Software and Data Integrity Failures (OWASP/SANS A08:2021)

- **Overview:**
  The application lacks proper CSRF protections, making it vulnerable to deceptive attacks where users are tricked into submitting unauthorized actions (such as password changes) without their awareness.
- **Business Impact:**
  - Unauthorized modifications to user accounts
  - Loss of control over account settings
  - Financial fraud risks
- **Method of Discovery:**
  - Crafted a fake HTML form to simulate a password change
  - Deployed the form and induced a logged-in user to submit it
  - Verified that the action occurred without proper CSRF tokens

3. Security Misconfiguration

- **CWE:** CWE-16
- **Category:** Security Misconfiguration (OWASP/SANS A05:2021)
- **Overview:**
  The system operates with default credentials, has enabled debug modes, and inadvertently exposes sensitive configuration files.
- **Business Impact:**
  - Broad attack surface exposure
  - Leakage of internal system details
  - Risk of unauthorized administrative access
- **Method of Discovery:**
  - Tested default login credentials (e.g., bee/bug)
  - Located exposed files such as phpinfo.php
  - Identified backup files through directory brute-forcing techniques

4. Unchecked URL Redirects and Forwards

- **CWE:** CWE-601
- **Category:** Server-Side Request Forgery (SSRF) (OWASP/SANS A10:2021)

- **Overview:**
  Weak redirect mechanisms in the application allow attackers to construct malicious URLs that reroute users to fraudulent or harmful sites.
- **Business Impact:**
  - Increased risk of phishing attacks
  - Possibility of credential theft
  - Erosion of customer trust
- **Method of Discovery:**
  - Tested by modifying URL parameters on a known redirect endpoint
  - Confirmed that the application did not validate redirection targets properly

5. XML External Entity (XXE) Vulnerability

- **CWE:** CWE-611
- **Category:** Insecure Design (OWASP/SANS A04:2021)
- **Overview:**
  The application inadequately processes XML input, enabling attackers to execute SSRF attacks, access local files, or trigger denial-of-service conditions.
- **Business Impact:**
  - Exposure of sensitive files (e.g., system passwd files)
  - Potential for SSRF attacks
  - Increased risk of application crashes from resource overuse
- **Method of Discovery:**
  - Located the XML processing endpoint (e.g., redirect.php?url=)
  - Injected a crafted XML payload with an external entity
  - Observed data extraction confirming the flaw

Target Website: https://owasp.org/www-project-juice-shop/

Vulnerability Overview Table

S.No	Vulnerability Type	CWE ID
1	Cross-Site Scripting (XSS)	CWE-79
2	Cross-Site Request Forgery (CSRF)	CWE-352
3	Insecure Direct Object References (IDOR)	CWE-639
4	SQL Injection	CWE-89
5	Broken Authentication	CWE-287

Detailed Vulnerability Reports

1. Cross-Site Scripting (XSS)

- **CWE:** CWE-79
- **Category:** Injection
- **Overview:**
  The application is vulnerable to XSS attacks. It fails to sanitize user input in its search functionality, allowing malicious scripts to be executed in users' browsers.
- **Method of Discovery:**
  - Injected sample script payloads into search fields
  - Monitored the reflected output for execution of unauthorized scripts
- **Business Impact:**
  - Unauthorized session access
  - Theft of sensitive user data
  - Defacement of the website leading to reputational damage

2. Cross-Site Request Forgery (CSRF)

- **CWE:** CWE-352
- **Category:** CSRF
- **Overview:**
  Insufficient CSRF protection enables attackers to trick authenticated users into performing unintended actions, such as altering account settings.

- **Method of Discovery:**
    - Generated crafted attack vectors
    - Validated that actions were executed without proper verification
- **Business Impact:**
    - Unauthorized transactions
    - Modification of user information
    - Potential for financial loss

3. Insecure Direct Object References (IDOR)

- **CWE:** CWE-639
- **Category:** Authorization
- **Overview:**
  Modifying object identifiers in requests allows attackers to access data that should remain restricted.
- **Method of Discovery:**
    - Altered URL parameters to test data access
    - Confirmed unauthorized resource access
- **Business Impact:**
    - Exposure of confidential data
    - Data integrity issues
    - Risk of regulatory penalties

4. SQL Injection

- **CWE:** CWE-89
- **Category:** Injection
- **Overview:**
  The application's input fields, particularly in the login process, are susceptible to SQL injection attacks, enabling attackers to manipulate database queries.
- **Method of Discovery:**
    - Entered malicious SQL commands into input fields
    - Analyzed database error messages for evidence of exploitation

- **Business Impact:**
    - Unauthorized database access
    - Potential data loss or corruption
    - Severe financial and reputational consequences

5. Broken Authentication

- **CWE:** CWE-287
- **Category:** Authentication
- **Overview:**
  Weak authentication mechanisms, such as poor session management and weak password policies, allow attackers to bypass security controls and impersonate users.
- **Method of Discovery:**
    - Tested login functionalities with weak credentials
    - Exploited session management flaws to gain unauthorized access
- **Business Impact:**
    - Unauthorized account takeovers
    - Compromise of sensitive user data
    - Loss of customer trust and potential legal liabilities

Stage – 2: Nessus Vulnerability Scanner Overview

Overview

Nessus is a leading vulnerability scanner designed to systematically detect weaknesses in networks, applications, and configurations. It plays an essential role in proactive cybersecurity by simulating real-world attacks, identifying potential exploits, and providing detailed reports on system vulnerabilities.

Key Features

- **Automated Deep Scanning:**
  Nessus performs comprehensive scans to detect misconfigurations, outdated software, and known vulnerabilities.
- **Compliance Auditing:**
  Supports standards like PCI DSS, HIPAA, and ISO 27001, ensuring that organizations remain compliant with regulatory requirements.

- **Plugin-Driven Detection:**
  Leverages an extensive library of plugins to keep pace with emerging threats and vulnerabilities.
- **Configuration Analysis:**
  Evaluates system settings and configurations to pinpoint areas that could be exploited.
- **SIEM Integration:**
  Seamlessly integrates with SIEM solutions to enhance incident response and threat intelligence workflows.

Role in Cybersecurity

Before deployment, it is crucial to grasp Nessus's significance in vulnerability management. Organizations utilize Nessus to conduct regular security assessments, prioritize identified risks, and make informed decisions on patch management and system hardening. Its detailed reporting mechanism is vital for both compliance auditing and effective penetration testing.

XSS Vulnerability Report (Sample)

Vulnerability: Cross-Site Scripting (XSS)

- **Severity:** High
- **Scanning Tool:** OWASP ZAP (Zed Attack Proxy)
- **Tested Port:** 80 (HTTP)

Description

The target web application demonstrates a significant XSS vulnerability within its search functionality. Due to improper sanitization of user input, attackers can inject malicious scripts that execute in the context of other users' browsers, leading to unauthorized session access and potential data theft.

Mitigation Strategies

- **Input Validation:**
  Enforce strict input validation and output encoding to neutralize malicious scripts.
- **Content Security Policy (CSP):**
  Implement CSP headers to restrict the execution of untrusted code.
- **Regular Updates:**
  Continuously update and patch the application to close known security loopholes.

Business Impact

- **Data Theft:** Unauthorized access to user sessions may lead to sensitive data exposure.
- **Reputation Damage:** Website defacement and compromised user trust can significantly harm brand image.
- **Legal Risks:** Non-compliance with data protection regulations could result in severe legal penalties.