14/03/2024

Thursday

Shaik Zahida
B. Tech iv year, AIML
208X1A4240
KALLAM HARANADHAREDDY
INSTITUTE OF TECHNOLOGY.

Assignment-4

STEP-1(OWASP TOP-10 VUNERABILITIES)

The Open Web Application Security Project, or OWASP, is an international nonprofit organization dedicated to web application security.

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging and Monitoring

POTENTIAL IMPACTS OF THIS VULNERABILITIES:

1.**Injection:** Inserting harmful code into apps to run unauthorized commands.

2.**Broken Authentication:** Flaws in login systems allow account compromise.

3.**Sensitive Data Exposure:** Failure to protect vital info leads to unauthorized access.

4.**XML External Entities (XXE):** Exploiting XML parser weaknesses to access files.

5.**Broken Access Control:** Inadequate user restrictions enable unauthorized access.

6.**Security Misconfiguration:** Poor security settings expose apps to attacks.

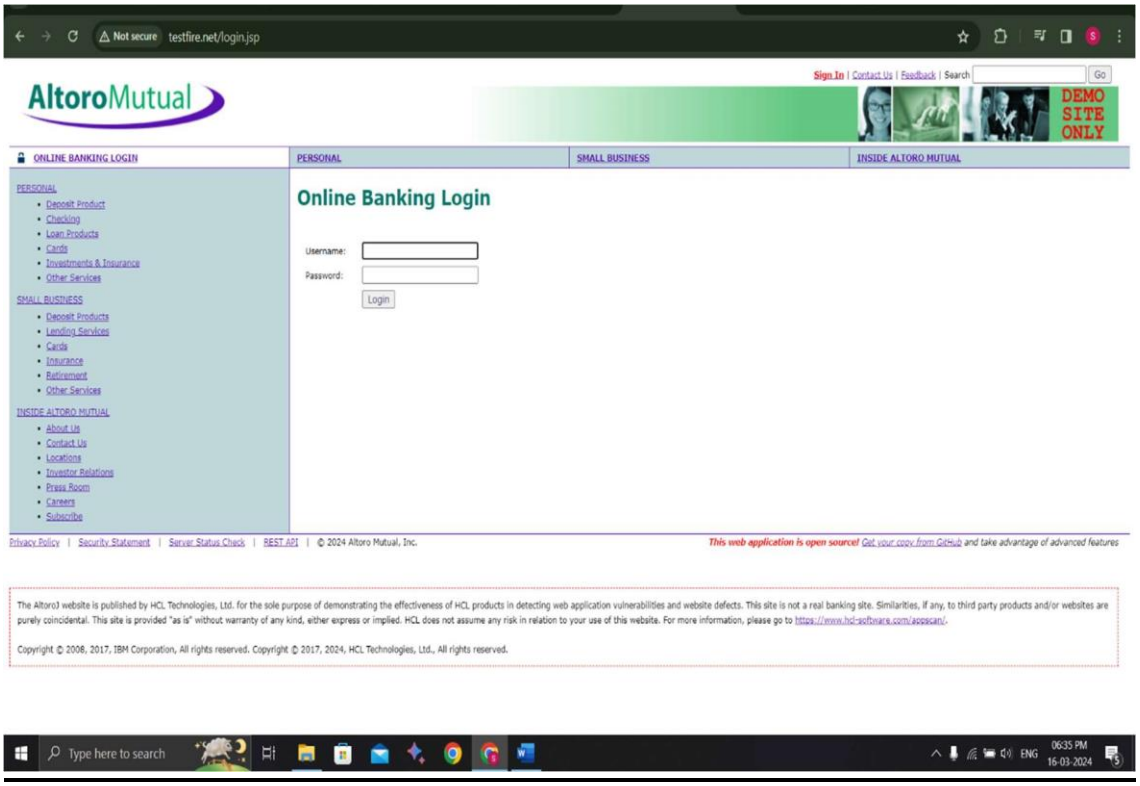
7.**Cross-Site Scripting (XSS):** Injecting malicious scripts to compromise sessions.

8.**Insecure Deserialization:** Unsafe data conversion leads to code execution.

9.**Using Components with Known Vulnerabilities:** Employing outdated software risks exploitation.

10.**Insufficient Logging and Monitoring:** Poor oversight delays threat detection and response.

STEP 2(ALTORO MUTUALS WEBSITE ANALYSIS)



1. Login Page: Allows registered users to securely access their accounts by entering their credentials, typically username/email and password.
2. User Registration: New users can create accounts by providing personal information like name and email, and setting up a password, often with optional additional details.
3. Payment Portal: Enables users to make secure payments for services or products, requiring input of payment details like credit/debit card information in a protected environment.
4. Contact Forms: Users can send inquiries or feedback by submitting their name, email, and message details through a form, sometimes with optional fields for additional information.

Identification of vulnerabilities

1. SQL Injection : Injecting malicious SQL queries into input fields to manipulate or access the database illicitly.

2. Cross-Site Scripting (XSS): Injecting malicious scripts into web pages viewed

by other users, enabling unauthorized actions or data theft.

3. Insecure Authentication Mechanisms: Weak authentication methods allowing attackers to bypass login mechanisms and gain unauthorized access.
4. Insecure Direct Object References: Exposing internal object references, enabling attackers to access unauthorized data or functionality.

Step 3(Vulnerability Identification Report)

IP address of this website-65.61.137.117

```
Nmap scan report for 65.61.137.117 Host is
up (0.046s latency).
PORT     STATE    SERVICE
21/tcp   filtered ftp
22/tcp   filtered ssh
23/tcp   filtered telnet

80/tcp   open     http
110/tcp  filtered pop3
143/tcp  filtered imap

443/tcp  open     https
3389/tcp filtered ms-wbt-server
```

Website Structure and Functionality:

1. Homepage : Provides an overview of Altro Mutual's services, promotions, and news updates.
2. Account Management: Allows users to register, log in, view account details, and manage their profiles.
3. Financial Products: Showcases Altro Mutual's range of financial products such as savings accounts, loans, and investment options.
4. Payment and Transactions: Enables users to make payments, transfers, and view transaction history securely.
5. Customer Support: Offers customer service options such as FAQs, contact forms, and live chat for assistance.
6. Educational Resources: Provides resources like articles, guides, and tools to help users make informed financial decisions.

Potential Areas of Vulnerability:

- Input Validation: Insufficient validation in forms may lead to SQL injection or XSS vulnerabilities.

- Authentication: Weak authentication measures or lax password policies can compromise accounts.
- Session Management: Poor session handling may result in session hijacking.
- Data Protection: Inadequate encryption or storage practices can lead to data breaches.
- Third-Party Integrations: Vulnerabilities in plugins or APIs pose security risks.
- Access Control: Weak access controls may allow unauthorized access to sensitive information or functionality.

mitigating strategy of each top-10 vulnerability

1. Injection : Use parameterized queries, input validation, and stored procedures to prevent malicious code execution in databases.
2. Broken Authentication: Implement strong password policies, multi-factor authentication, and secure session management to prevent unauthorized access.
3. Sensitive Data Exposure: Encrypt sensitive data, enforce access controls, and use secure communication protocols to protect data confidentiality.
4. XML External Entities (XXE): Disable external entity parsing, validate XML input, and utilize secure XML parsers to prevent XXE attacks.
5. Broken Access Control: Implement proper access controls, enforce least privilege, and perform regular access control audits to prevent unauthorized access.
6. Security Misconfiguration: Follow secure configuration guidelines, apply patches promptly, and use automated tools for configuration management.
7. Cross-Site Scripting (XSS): Implement input validation, output encoding, and Content Security Policy (CSP) to prevent XSS attacks.
8. Insecure Deserialization: Avoid deserialization of untrusted data, implement integrity checks, and use secure deserialization libraries.
9. Using Components with Known Vulnerabilities: Regularly update software components, use software composition analysis tools, and monitor vulnerability databases.
10. Insufficient Logging and Monitoring: Implement comprehensive logging, establish real-time monitoring, and utilize intrusion detection/prevention systems for timely incident response.

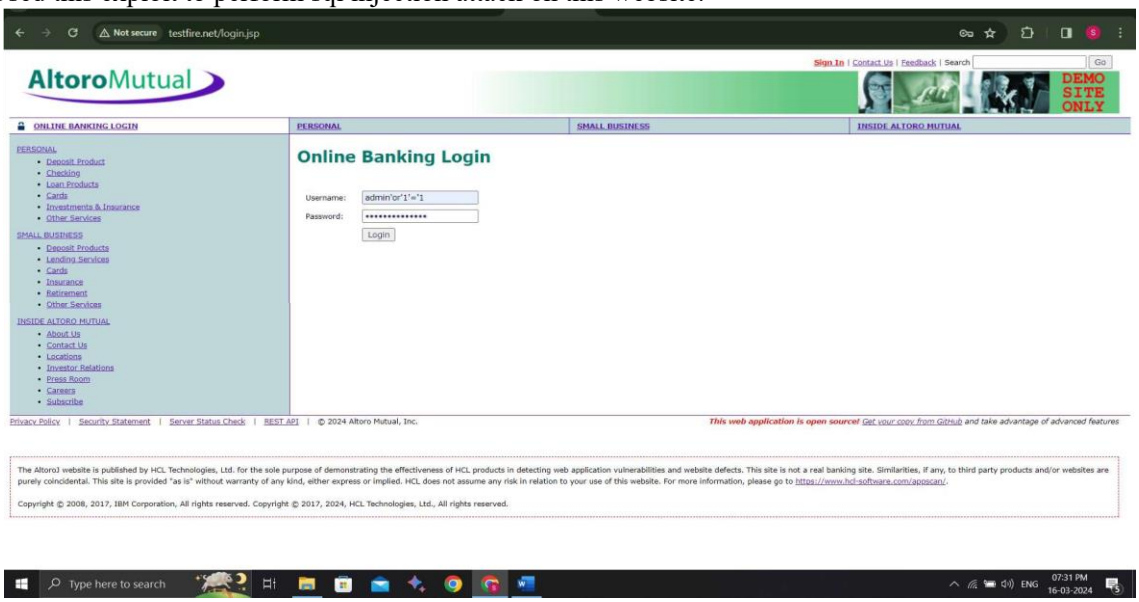
Step 4(Vulnerability Exploitation on Demonstration)

1.SQL INJECTION EXPLOIT-

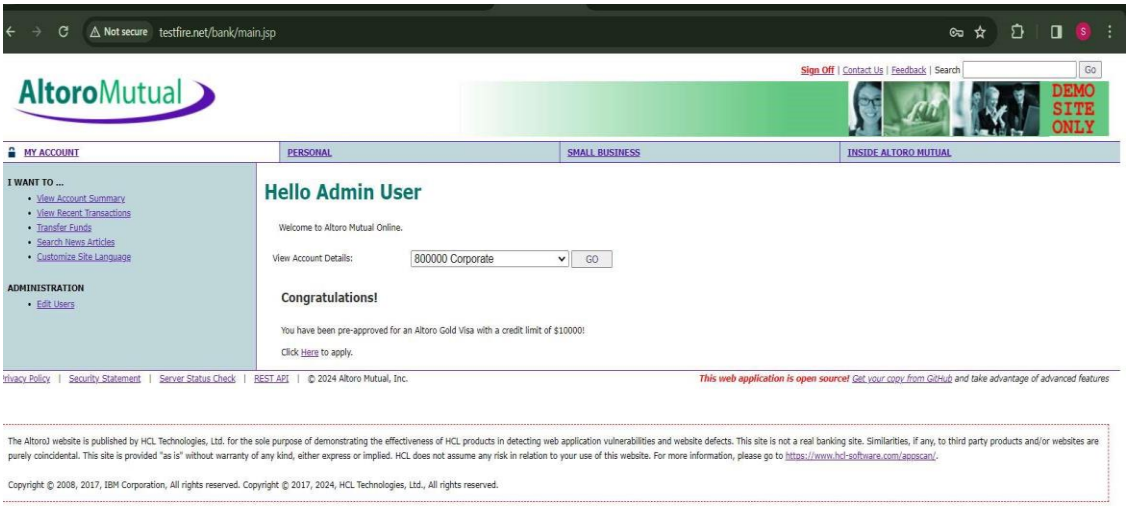
Username: admin'or'1'='1

Password: admin'or'1'='1

Used this exploit to perform sql injection attack on this website.

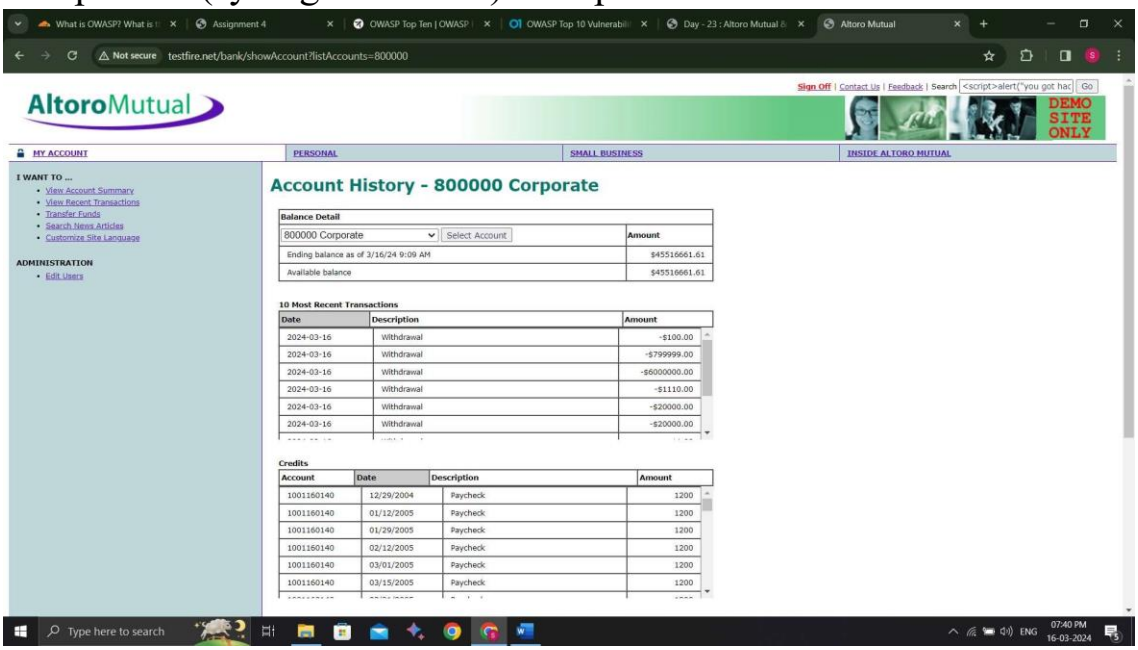


After press login button the attack will started working to access the all content in that website

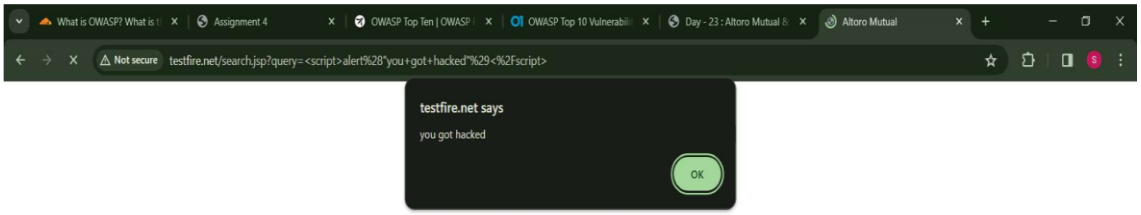


2.Cross-site scripting-

Using this script in search bar to modify the code in backend
<script>alert("you got hacked")</script>



After pressing go option the exploit work



Step 5(Mitigation Strategy Proposal)

1. **Identify High-Risk Vulnerabilities**: Use vulnerability assessments to find critical vulnerabilities.
2. **Assess Risks:** Evaluate the impact and likelihood of exploitation for each vulnerability.
3. **Prioritize Based on Severity**: Use industry-standard metrics like CVSS to rank vulnerabilities.
4. **Establish Criteria**: Set clear criteria for prioritizing vulnerabilities, considering factors like system criticality and patch availability.
5. **Patch Management**: Quickly deploy patches for high-risk vulnerabilities and regularly monitor effectiveness.
6. **Remediation Plan**: Develop a plan to systematically address high-risk vulnerabilities, considering operational constraints.
7. **Continuous Monitoring**: Keep an eye on the effectiveness of mitigation efforts and adapt strategies as needed.
8. **Communication**: Maintain open communication with stakeholders to keep them informed about prioritized vulnerabilities and mitigation progress.

Step 6(Documenting the Exploit Process)

In this step, the exploit process is thoroughly documented. This documentation in understanding the vulnerabilities more comprehensively and in developing effective countermeasures. Accurately capturing the steps involved in exploiting vulnerabilities informs the remediation process and enhances overall cybersecurity posture.

THANK YOU

Shaik Zahida