Name: Shivani Narayan L

Regno: 21BLC1119

Week 1 Assignment: Top 5 OWASP

A01:2021-Broken Access Control:

CWE: 284: Improper Access Control

Description: The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor

Business Impact: This weakness allows an attacker to bypass intended security restrictions and perform a variety of actions depending on the source of error and functionality of the application. An attacker might be able to perform certain actions by gaining elevated privileges, reading otherwise restricted information, executing commands, bypassing implemented security mechanisms, etc.

A02:2021-Cryptographic Failures

CWE-326: Inadequate Encryption Strength

Description: The product stores or transmits sensitive data using an encryption scheme that is theoretically sound, but is not strong enough for the level of protection required. A weak encryption scheme can be subjected to brute force attacks that have a reasonable chance of succeeding using current attack methods and resources.

Business Impact: An attacker may be able to decrypt the data using brute force attacks leading to data breaches which could further lead to legal consequences, reputational damage, financial losses, operational disruptions, competitive disadvantage, loss of intellectual property, and potential lawsuits. Strong encryption and compliance with regulations are crucial to mitigate these risks.

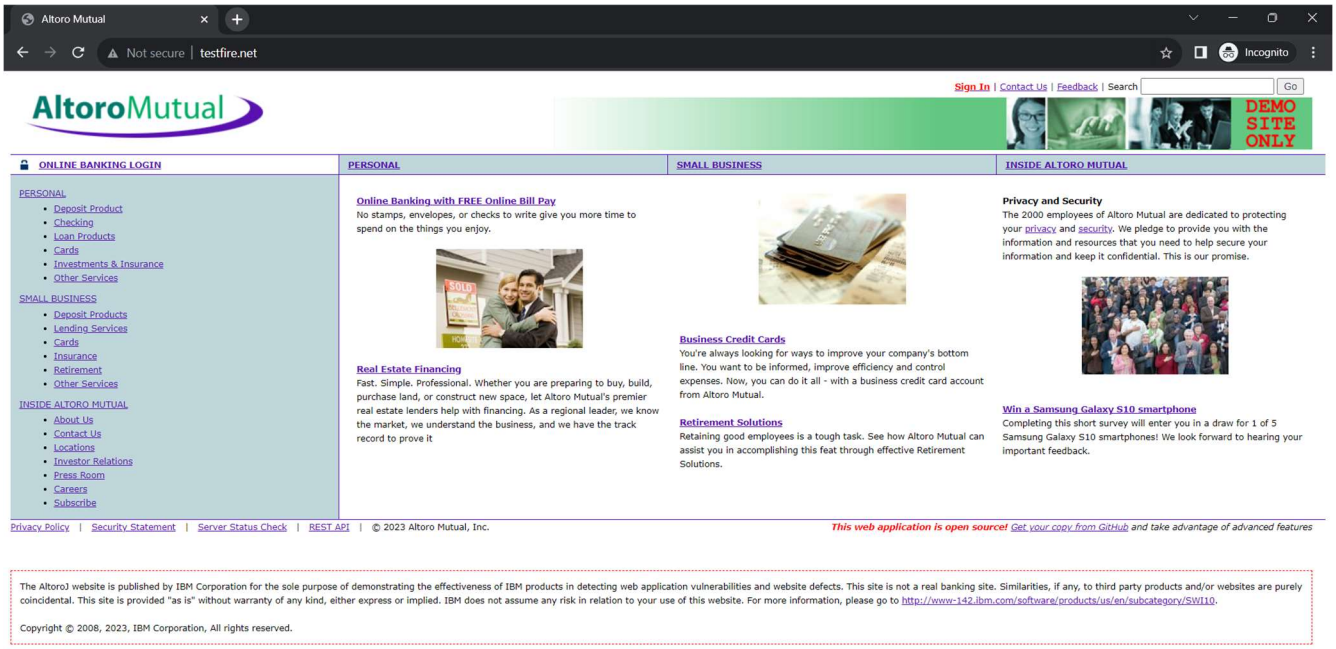
A03:2021-Injection

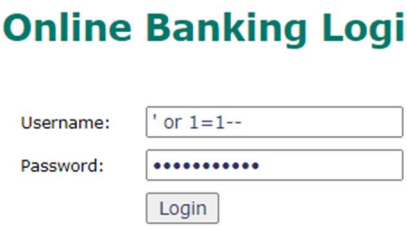
CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

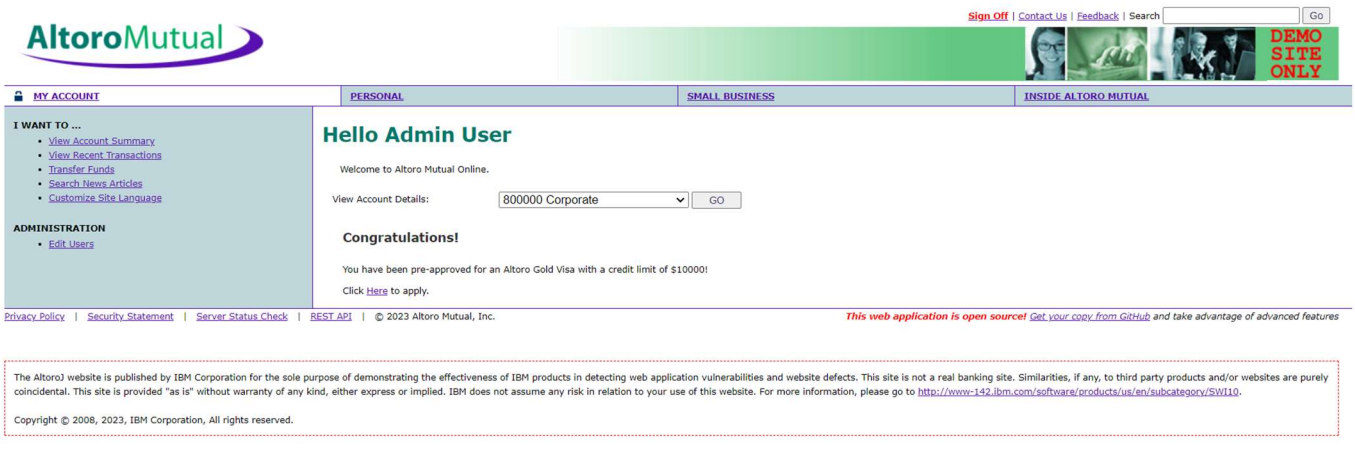
Description: The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component.

Business Impact: SQL injection vulnerabilities pose significant risks to a system's security. They can result in unauthorized reading of application data, undermining confidentiality, especially when databases store sensitive information. Additionally, these vulnerabilities can enable attackers to bypass protection mechanisms, potentially accessing the system without proper credentials and even modify or delete application data, compromising data integrity.

Example: http://testfire.net/







A04:2021 – Insecure Design

CWE-256: Plaintext Storage of a Password

Description: Storing a password in plaintext may result in a system compromise. Password management issues occur when a password is stored in plaintext in an application's properties, configuration file, or memory. Storing a plaintext password in a configuration file allows anyone who can read the file access to the password-protected resource. In some contexts, even storage of a plaintext password in memory is considered a security risk if the password is not cleared immediately after it is used.

Business Impact: Storing a plaintext password in a configuration file allows anyone who can read the file access to the password-protected resource. This can lead to unauthorized access, identity theft, financial loss, data breaches, and reputational damage. Developers sometimes feel they are helpless in safeguarding the application against someone with configuration file access, but this mindset aids potential attackers.

A05:2021 – Security Misconfiguration

CWE-611: Improper Restriction of XML External Entity Reference

Description: The product processes an XML document that can contain XML entities with URIs that resolve to documents outside of the intended sphere of control, causing the product to embed incorrect documents into its output.

Business Impact: Improper Restriction of XML External Entity Reference can have a significant impact on security. It can lead to unauthorized access to internal files, potential data exposure, and even denial of service. By exploiting this vulnerability, attackers may manipulate XML parsers to access or modify sensitive information, disrupting system functionality and compromising data integrity and confidentiality. Preventing and mitigating this vulnerability is crucial to maintaining a secure software environment.