NAME: MUNGILI CHETAN SAI RAJU

ASSIGNMENT

TOP 5 OWASP VULNERABLILITIES

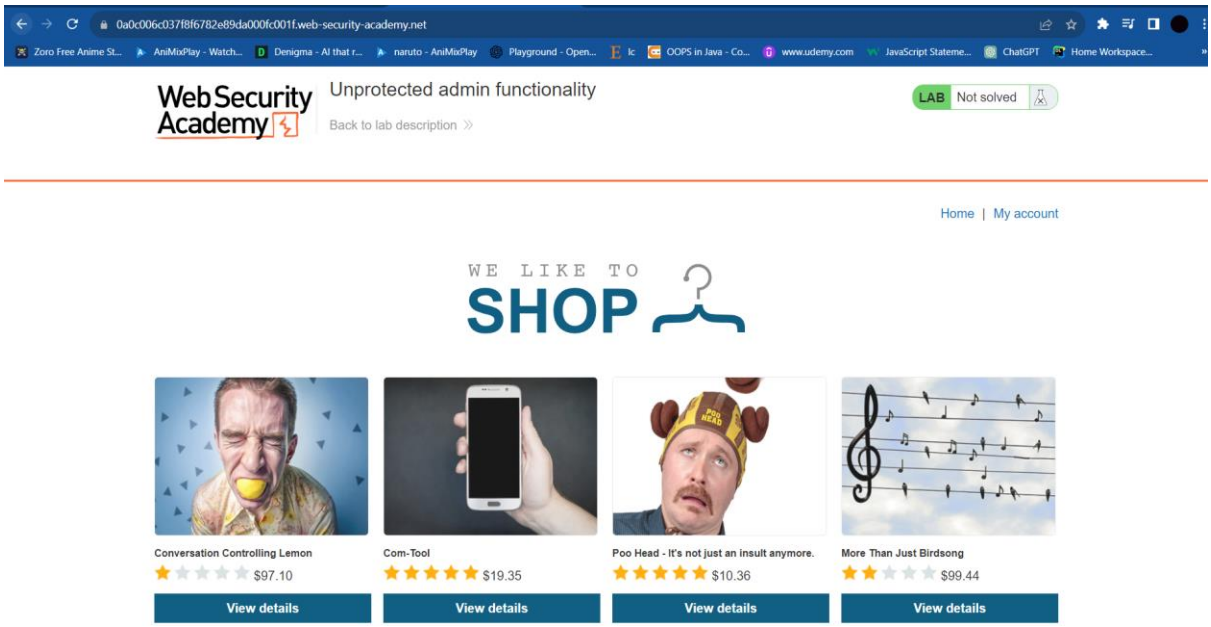
1. A01:2021-Broken Access Control:

Broken Access Control vulnerabilities occur when an application fails to properly enforce restrictions on what authenticated users are allowed to do. In other words, it's a situation where users gain unauthorized access to certain resources or actions that they shouldn't have access to.

CWE-284: Improper Access Control:

- **Description**: The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.
- **Business Impact**: CWE-284: Improper Access Control refers to a vulnerability where an application doesn't properly enforce restrictions on who can access certain resources or perform specific actions. This can occur due to missing or weak authorization checks, incorrect permissions, or flawed logic. Attackers can exploit this weakness to gain unauthorized access, escalate their privileges, or perform actions they shouldn't be allowed to. For instance, an attacker might access sensitive data, manipulate functionalities, or even become an administrator. To prevent this, applications should implement strong access controls, validate user permissions, and thoroughly test their authorization mechanisms to ensure only authorized users can access appropriate resources and actions.

Performing the Broken Access Control:



In the URL we use,(/administrator-panel) to access the accounts in the page

2. A02:2021-Cryptographic Failures:

Cryptographic failures in this context relate to vulnerabilities and weaknesses in the implementation and use of cryptographic mechanisms. Cryptography is used to secure sensitive data and communications by encoding information in a way that only authorized parties can understand. However, incorrect or weak cryptographic practices can lead to vulnerabilities that attackers can exploit.

CWE-327: Use of a Broken or Risky Cryptographic Algorithm:

- **Description**: The product uses a broken or risky cryptographic algorithm or protocol.
- **Business Impact**: This CWE highlights a vulnerability stemming from the utilization of outdated, flawed, or insecure encryption methods. Employing such cryptographic algorithms increases the likelihood of successful attacks, as attackers can exploit weaknesses in these algorithms to compromise the security of sensitive data. For instance, weak encryption might be easily cracked, allowing unauthorized access to confidential information. To mitigate this vulnerability, it's vital to use modern, well-established cryptographic algorithms that have undergone thorough security evaluation. Regularly updating and replacing deprecated algorithms is essential to maintain the confidentiality and integrity of data within an application.

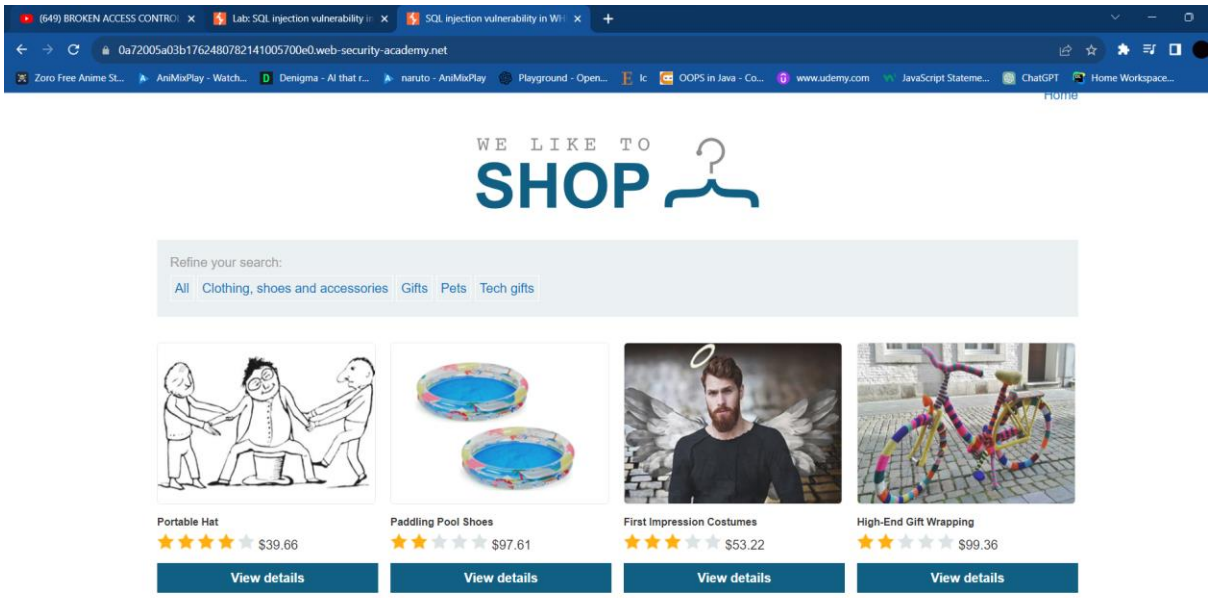
3. A03:2021-Injection:

Injection vulnerabilities occur when malicious data is injected into an application's input fields, causing the application to execute unintended commands or actions. This can lead to unauthorized access, data breaches, and other forms of exploitation.

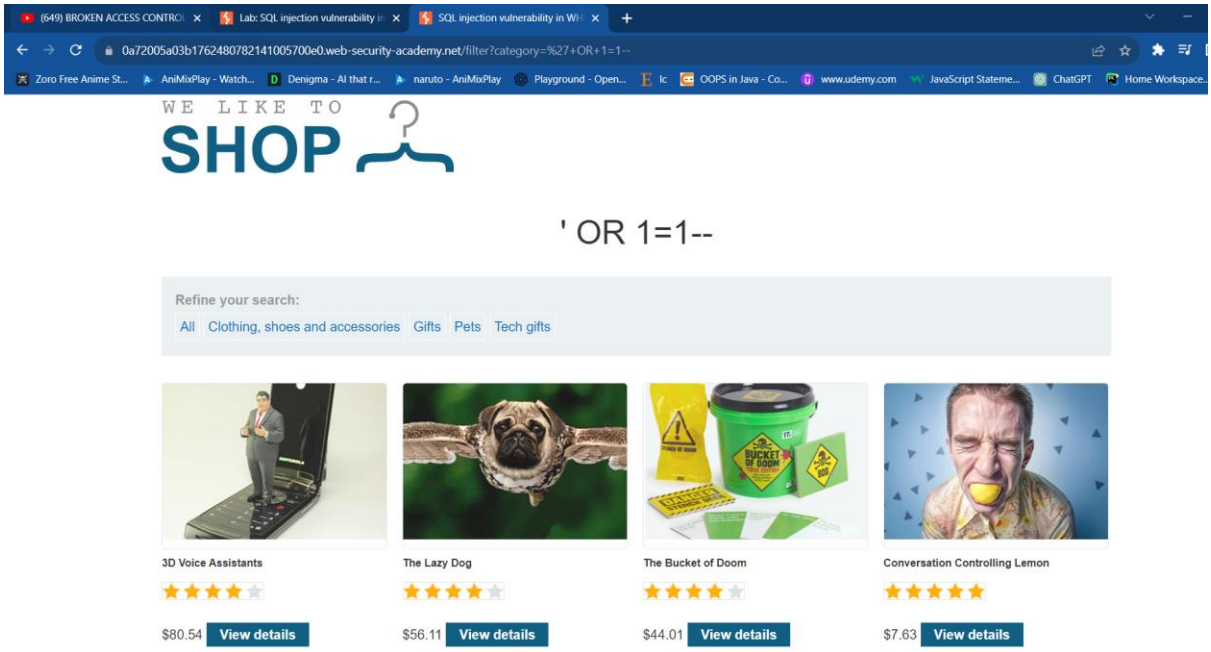
CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection'):

- **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 can have severe consequences on application security. Attackers can exploit this vulnerability to extract, modify, or delete sensitive data stored in databases. This can lead to unauthorized access to user accounts, exposure of confidential information, and potentially even the takeover of the entire database. Additionally, attackers can abuse SQL Injection to execute arbitrary commands, affecting application functionality and potentially compromising the underlying server. The impact includes data breaches, loss of trust from users, regulatory penalties, and potential legal consequences.

Performing the SQL injection:



The SQL injection is used in the URL('+OR+1=1--) to access the unpublished products before hand.



4. A04:2021-Insecure Design

A04:2021-Insecure Design refers to a category of security vulnerabilities outlined in the OWASP Top Ten Project. It involves fundamental flaws in the design of a software application, resulting in security weaknesses that can be exploited by attackers. Insecure design can lead to persistent vulnerabilities throughout the application's lifecycle.

CWE-601: URL Redirection to Untrusted Site ('Open Redirect'):

- **Description:** A web application accepts a user-controlled input that specifies a link to an external site, and uses that link in a Redirect. This simplifies phishing attacks.
- **Business Impact:** Insecure design can have severe consequences, allowing attackers to exploit inherent weaknesses from the application's foundation. It might result in widespread vulnerabilities, data breaches, and unauthorized access. Attackers could manipulate input, bypass authentication, or execute arbitrary code. Additionally, insecure design often demands extensive efforts for remediation, impacting development time and costs. As applications built upon flawed designs are challenging to secure retrospectively, addressing these issues during the design phase is essential.

5. A05:2021-Security Misconfiguration

It involves the improper setup of security settings, configurations, and permissions in software applications, servers, and other components. Security misconfigurations can arise from default settings, unnecessary services, exposed sensitive information, and more, leaving systems vulnerable to attacks. Attackers can exploit these weaknesses to gain unauthorized access, steal data, or compromise the integrity of systems.

CWE-548: Exposure of Information Through Directory Listing

- **Description:** A directory listing is inappropriately exposed, yielding potentially sensitive information to attackers.
- **Business Impact:**CWE-548 can have significant consequences for an application's security. When directories are exposed, attackers gain insights into the application's internal structure, potentially identifying vulnerabilities and points of attack. The impact extends beyond information disclosure, as exposed sensitive files, configuration data, or proprietary information could be accessed by unauthorized parties. This may lead to data breaches, privacy violations, and unauthorized use of critical resources. Additionally, attackers can leverage this information to craft more targeted attacks, increasing the overall risk to the system. To mitigate this vulnerability, web servers and applications should be configured to disable directory listing by default, ensuring that sensitive information remains hidden from prying eyes and reducing the risk of information exposure. Regular security assessments can help identify and address this weakness.