Министерство Образования, Культуры и Исследований

Молдавский Государственный Университет

Факультет Математики и Информатики

Департамент Информатики

**Лабораторная работа №3**

По курсу: Безопасность web приложений.

На тему: Контроль доступа к веб приложению.

Выполнил: Гуцу Даниил,

группа IA2102

Проверила: Плешка Наталья

Кишинев, 2024

# **Требования к лабораторной работе**

1. Титульный лист (в соответствии с прилагаемой моделью);
2. Информации о неправильных / непроверенных входов в приложениях, путем обхода аутентификации;
3. Print-screen-ы входов в закрытые пространства путем обхода аутентификации;
4. Скрипты обеспечения безопасности приложения при попытки «неправильного» входа и измененная структура БД;
5. Print-screen-ы с попытками входов на созданные закрытые пространства и возможность приложения «заставить» пользователя «работать» по правильному сценарию;
6. Выводы.

**Что такое обход аутентификации?**

**Обход аутентификации** - это процесс, при котором злоумышленник пытается получить доступ к закрытым ресурсам или функциональности веб-приложения без успешной аутентификации. Это может быть сделано различными способами, такими как перехват сеансов, использование известных URL-адресов, атака на сессии и другие методы. Обход аутентификации опасен, потому что злоумышленник может получить доступ к конфиденциальным данным, изменить параметры приложения и повредить его безопасность.

**Предотвращение обхода аутентификации**

Для предотвращения обхода аутентификации необходимо внедрить следующие меры безопасности:

* Использование сессий:

Сессии позволяют приложению идентифицировать пользователя после успешной аутентификации и хранить информацию о его состоянии. Это помогает предотвратить обход аутентификации, так как без сессии злоумышленник не сможет представиться другим пользователем.

* Контроль доступа:

Реализовать строгий контроль доступа к ресурсам и функциональности приложения. Убедитесь, что только аутентифицированные пользователи имеют доступ к закрытым ресурсам.

**Использование сессий**

**Сессии** - это механизм хранения информации о пользователе между разными запросами на сервер. Сессии помогают установить идентичность пользователя и обеспечивают безопасное хранение данных, связанных с пользователем, таких как идентификатор сессии, роли пользователя, настройки и другие параметры. Использование сессий позволяет приложению сохранить состояние пользователя в течение его сеанса работы с приложением, что полезно для создания персонализированного опыта и безопасности.

**Роли пользователей**

**Роли пользователей** - это способ классификации пользователей на различные группы в зависимости от их прав и обязанностей в приложении. Роли могут быть использованы для контроля доступа к различным ресурсам и функциональности. Например, веб-приложение может иметь роли "администратор" и "менеджер", и пользователи могут быть назначены в одну из этих ролей. Это полезно, так как оно позволяет приложению ограничивать доступ к определенным функциям или данным на основе роли пользователя, что повышает безопасность и гибкость приложения.

Внедрение сессий и использование ролей в веб-приложении улучшают его безопасность и обеспечивают более точный контроль доступа к ресурсам и функциональности в зависимости от роли пользователя. Это помогает предотвратить обход аутентификации и повысить общую безопасность приложения.

# **Практическая часть**

Была создана новая таблица в БД. Которая хранит роли пользоваетлей:

CREATE TABLE IF NOT EXISTS user (

    id INT AUTO\_INCREMENT PRIMARY KEY,

    login VARCHAR(50) NOT NULL,

    password VARCHAR(50) NOT NULL,

    role VARCHAR(50) NOT NULL

);

Далее, в файле с аутенфикацией была добавлена проверка на редирект в зависимости от роли:  
  
if ($result->num\_rows > 0) {

    $row = $result->fetch\_assoc();

    $\_SESSION['loggedin'] = true;

    $\_SESSION['role'] = $row['role']; // Добавляем роль в сессию

    if ($row['role'] == 'admin') {

        header("Location: admin\_panel.php");

    } elseif ($row['role'] == 'manager') {

        header("Location: manager\_panel.php");

    } else {

        echo "<script>alert('Unknown user role');</script>";

        echo "<script>window.location.href='login.php';</script>";

        exit;

    }

    exit;

} else {

    echo "<script>alert('Invalid login credentials');</script>";

    echo "<script>window.location.href='login.php';</script>";

    exit;

}

А также в файлах, принадлежащих конкретной роли есть проверка, для того чтобы страница админа была доступна только админу. В случае со страницей менеджера она доступна менеджерам и админам.

<?php

session\_start();

if (!isset($\_SESSION['loggedin']) || $\_SESSION['role'] != 'manager' || $\_SESSION['role'] != 'admin') {

    header("Location: login.php");

    exit;

}

?>

<!DOCTYPE html>

<html>

<head>

    <title>Manager Panel</title>

</head>

<body>

    <h2>Welcome, Manager!</h2>

    <!-- Ваше содержимое панели менеджера -->

    <a href="logout.php">Logout</a>

</body>

</html>

<?php

session\_start();

if (!isset($\_SESSION['loggedin']) || $\_SESSION['role'] != 'admin') {

    header("Location: login.php");

    exit;

}

?>

<!DOCTYPE html>

<html>

<head>

    <title>Admin Panel</title>

</head>

<body>

    <h2>Welcome, Admin!</h2>

    <!-- Ваше содержимое панели администратора -->

    <a href="logout.php">Logout</a>

</body>

</html>

# **Вывод**

В рамках выполнения лабораторной работы по улучшению безопасности веб-приложения с использованием сессий и ролей пользователей, были проведены следующие действия:

* Реализован механизм сессий в веб-приложении. Это позволило приложению идентифицировать пользователей после успешной аутентификации и хранить информацию о их состоянии между запросами. Это также позволило безопасно хранить данные, связанные с пользователями.
* Введены роли пользователей - "администратор" и "менеджер". Роли определяют права и доступ каждого пользователя в приложении. Это обеспечило более точный контроль доступа к различным ресурсам и функциональности в зависимости от роли.
* Была предотвращена возможность обхода аутентификации. Злоумышленник больше не может получить доступ к закрытым ресурсам или функциональности путем простого перехода по известным URL-адресам. Сессии и контроль доступа на основе ролей устраняют эту уязвимость.
* Внедрена возможность выхода из закрытых пространств с возвратом на страницу аутентификации. Это обеспечивает дополнительный уровень безопасности и удобства для пользователей.

В результате выполнения лабораторной работы была значительно улучшена безопасность веб-приложения. Использование сессий и ролей позволило более надежно и гибко управлять доступом пользователей и предотвратить попытки обхода аутентификации. Эти меры безопасности укрепляют защиту приложения и обеспечивают более безопасное и устойчивое функционирование для пользователей.