**[Слайд 1]**

Добрый день, уважаемые члены комиссии. Меня зовут Кудин Артемий Юрьевич, я студент группы ОБС 21-2. Подготовил курсовую работу, на тему “Анализ современных угроз информационной безопасности на веб-сайтах и способы их предотвращения”.

**[Слайд 2]**

Целью работы является проанализировать современные угрозы безопасности на веб-сайтах и выявить способы их предотвращения.

**[Слайд 3]**

В соответствии с целью, были поставлены следующие задачи:

* Создать среду для тестирования методов защиты от атак;
* Провести атаки на тестовую среду;
* Обеспечить защиту тестовой среды от атак.

**[Слайд 4]**

**Межсайтовый скриптинг**

Межсайтовый скриптинг (XSS или *Cross-Site Scripting* ) - это подтип атаки на веб-страницу, заключающийся во внедрении в веб-страницу вредоносного кода, который будет выполнен на компьютере пользователя при открытии им этой страницы

**[Слайд 5]**

Рассмотрим некоторые способы предотвращения XSS атак:

* **Экранирование и кодирование вывода данных**;
* **Использование HTTP заголовков Content Security Policy (CSP);**
* **Валидация и фильтрация ввода данных;**

И другие

**[Слайд 6]**

**SQLI**

SQL Инъекции (SQLI) - это тип атаки на веб-приложения, при которой злоумышленник внедряет SQL-запросы в поля ввода или другие механизмы общения с базой данных.

**[Слайд 7]**

Способы предотвращения SQL Инъекций, включают в себя способы предотвращения JavaScript XSS, а также:

* **Применение принципа минимальных привилегий;**
* **Использование белого списка;**

И другие

**[Слайд 8]**

**Используемые технологии для разработки тестовой среды**

Начнем с технологий, используемых на стороне клиента.

**HTML;**

**CSS**;

**JavaScript;**

**[Слайд 9]**

Для серверной части проекта, было принято решение использовать следующие технологии:

**MySQL;**

**PHP;**

**[Слайд 10]**

**Подготовка тестовой среды**

Для реализации веб-приложения был выбран хостинг-провайдер под названием “SpaceWeb”, который предоставляет бесплатный сервер для хостинга.

С помощью программы Open Server был развернут локальный сервер, для разработки тестовой среды.

**[Слайд 11]**

**Применение XSS и защита от атаки.**

Для работы с XSS была разработана веб-страница, которая имитирует блок комментариев в интернете. Также, была создана база данных под названием “main” в которой, в свою очередь, была создана таблица с названием “comments”.

На странице присутствует поле ввода, через которое пользователь может ввести текст и сделать запись в базе, а также список комментариев каждый из которых представляет одну запись в базе данных.

**[Слайд 12]**

Рассмотрим примеры внедрения вредоносного кода в базу данных. Пользователь вводит в поле ввода HTML тэг <img/> с ссылкой на картинку, и отправляет эти данные на сервер.

После отправки запроса на сервер, картинку можно увидеть в последнем комментарии на странице.

**[Слайд 13]**

Теперь рассмотрим случай, когда злоумышленник может удалить содержимое страницы. С помощью инструментов разработчика, узнаем название блока с комментариями и отправляем на сервер следующий скрипт. После проделанных действий, при переходе на сайт пользователи увидят пустую страницу.

Чтобы избавиться от данной уязвимости, было принято решение, использовать PHP функцию “htmlspecialchars”, которая экранирует данные, превращая специальные символы в HTML сущности. При переходе на страницу, можно наблюдать результат экранирования.

**[Слайд 14]**

Для работы с SQL Инъекциями была разработана веб-страница, которая имитирует форму авторизации. В приложении Д описан код страницы. Также была создана таблица, под названием “users”.

**[Слайд 15]**

Таблица users имеет запись, которая представляет пользователя под именем “admin” с паролем “admin”.

**[Слайд 16]**

Рассмотрим ситуацию, когда злоумышленник с помощью SQL инъекции может успешно авторизоваться. Часть кода, которая отвечает за авторизацию находится в приложении Е. и на слайде.

Чтобы злоумышленнику добиться работы его введённого вредоносного кода, ему достаточно вписать в поле ввода строчную кавычку, которая позволит коду после нее выполниться как часть запроса. Рассмотрим пример: злоумышленник заходит на страницу и вводит в одно из полей ввода следующую строчку что на слайде.

Данное выражение означает, что после закрытой строки, благодаря кавычке, выполняется оператор OR, код после которого, в итоге позволит злоумышленнику войти в систему. После того, как злоумышленник отправит данные, его перенаправит на страницу, где он увидит сообщение о том, что вход выполнен успешно.

**[Слайд 17]**

Чтобы защититься от взлома, было принято решение использовать функцию “mysqli\_real\_escape\_string”, которая экранирует кавычки в переменных. Благодаря этой функции, злоумышленник не сможет изменить ход выполнения SQL Запроса.

**[Слайд 18]**

**Заключение**

Подводя итоги, были рассмотрены актуальные уязвимости веб-безопасности, и методы их предотвращения. Также, была проведена практическая работа с уязвимостями внедрения вредоносного кода на веб-страницы. Была разработана тестовая среда, на которой были протестированы XSS и SQLI атаки и способы их предотвращения.

По итогам работы, все следующие задачи были выполнены:

**[Слайд 19]**

Спасибо за внимание! Готов выслушать ваши вопросы.