Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский государственный национальный исследовательский университет»

Кафедра общей физики

Отчет по научно-исследовательской работе

**Сайт-тренажер уязвимостей «VulTest»**

Выполнили:

Студенты 3-го курса

физического факультета

Заболотько Семён

Яценко Дмитрий

Веялко Алексей

Сюткин Егор

Аристова Дарья

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Научный руководитель:

И.Н.Черепанов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

г.Пермь 2023

Содержание

1. **Введение**
2. **Основные уязвимости**
3. **Функционал сайта**
   1. Регистрация пользователя
   2. Авторизация пользователя
   3. Выбор уязвимости
   4. XSS
   5. SQL injection
   6. Исполнение произвольного кода
4. **Программная часть**
5. **Заключение**
6. **Список литературы**

Введение

Уязвимость в информационной безопасности — это слабость в информационной системе, программном обеспечении или процедуре, которую злоумышленник может использовать для проникновения в систему, нарушения ее работы или доступа к ней.

Поэтому **целью работы является** демонстрация основных уязвимостей веб-сайтов, а также обучение работы с ними(атака и защита). Для выполнения цели необходимо выполнить следующие задачи:

1. Разработать веб-сайт на локальном сервере
2. Организовать регистрацию и авторизацию пользователей сайта
3. Организовать пользователю интерфейс для кибератаки путём использования основных уязвимостей
4. Организовать пользователю интерфейс для защиты от основных уязвимостей

Данная работа является актуальной, так как веб-сайты часто становятся недоступными из-за атак типа отказано в обслуживании, или отображение изменённой (и часто повреждённой) информации на их страницах. В других случаях миллионы паролей, адресов электронной почты и данные кредитных карт становились общедоступными, подвергая пользователей веб-сайта личному смущению или к финансовым рискам.

Цель веб-безопасности заключается в предотвращении этих (или других) видов атак. Более формальным определением веб-безопасности является: *способы защиты веб-сайтов от несанкционированного доступа, использования, изменения, уничтожения или нарушения работы.*

Основные уязвимости

На нашем сайте реализованы 3 основные уязвимости: Cross-Site Scripting(XSS) – «Межсайтовый скриптинг», SQL injection(SQLi) – «Внедрение SQL-кода» и исполнение произвольного кода».

**Cross-Site Scripting(XSS)**

XSS – это тип [атаки](https://ru.wikipedia.org/wiki/%D0%A5%D0%B0%D0%BA%D0%B5%D1%80%D1%81%D0%BA%D0%B0%D1%8F_%D0%B0%D1%82%D0%B0%D0%BA%D0%B0) на [веб-системы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), заключающийся во внедрении в выдаваемую веб-системой [страницу](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) [вредоносного кода](https://ru.wikipedia.org/wiki/%D0%92%D1%80%D0%B5%D0%B4%D0%BE%D0%BD%D0%BE%D1%81%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) (который будет выполнен на [компьютере](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) пользователя при открытии им этой страницы) и взаимодействии этого кода с веб-сервером злоумышленника. Является разновидностью атаки «[Внедрение кода](https://ru.wikipedia.org/wiki/%D0%A5%D0%B0%D0%BA%D0%B5%D1%80%D1%81%D0%BA%D0%B0%D1%8F_%D0%B0%D1%82%D0%B0%D0%BA%D0%B0#%D0%98%D0%BD%D1%8A%D0%B5%D0%BA%D1%86%D0%B8%D1%8F_%D0%BA%D0%BE%D0%B4%D0%B0)». На сегодняшний день XSS является третьим по значимости видом рисков для веб-приложений. Его основная опасность заключается в том, что на веб-страницах содержится много пользовательских или иных уязвимых данных. Злоумышленник может использовать их для доступа к платежным картам, компьютерам пользователей и т.д.

Основной способ внедрения вредоносного кода на сайт или в веб-приложение — через интерактивные элементы сайта. Например, его можно разместить в строке поиска, форме обратной связи или авторизации, поле для публикации комментария. Это доступные и самые простые «точки входа» для злоумышленника, который по своей сути изначально является одним из посетителей ресурса.

Для защиты от данной угрозы рекомендуется:

* настроить фильтрацию и экранизацию входных параметров, то есть информации, вводимой пользователем через интерактивные поля и формы;
* настроить автоматическую замену спецсимволов, чтобы четко отделить пользовательский текст от исполняемого кода;
* на каждой странице сайта разместить кодировку перед какими-либо пользовательскими полями;
* установить ограничения домена и путей приема cookie-файлов с помощью SSL или параметра HttpOnly;
* регулярно проверять сайт или веб-приложение на уязвимости такими инструментами, как Nessus, Nikto Web Scanner и т.д.;
* задать список желательных источников для загрузки контента с помощью заголовка Content Security Policy;
* регулярно обновлять браузер до актуальной версии и использовать расширения для проверки URL, скриптов, интерактивных форм и других потенциальных источников угроз.

Практика показывает, что на 100% от XSS-атак не защищен ни один ресурс или браузер. В ответ на появление новых средств защиты злоумышленники разрабатывают новые пути их обхода. Однако использование актуальных способов цифровой гигиены и обычная бдительность позволяют снизить риск межсайтового скриптинга до приемлемого минимума.

**SQL injection**

Внедрение SQL-кода — один из распространённых способов взлома [сайтов](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82) и [программ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), работающих с [базами данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), основанный на внедрении в запрос произвольного [SQL](https://ru.wikipedia.org/wiki/SQL)-кода.

Внедрение SQL, в зависимости от типа используемой [СУБД](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) и условий внедрения, может дать возможность атакующему выполнить произвольный запрос к базе данных (например, прочитать содержимое любых [таблиц](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C)), удалить, изменить или добавить [данные](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5)), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере.

Атака типа внедрения SQL может быть возможна из-за некорректной обработки входных данных, используемых в SQL-запросах. Разработчик прикладных программ, работающих с базами данных, должен знать о таких уязвимостях и принимать меры противодействия внедрению SQL.

Ситуация опасна тем, что большинство веб-форм не имеют механизмов, которые бы исключали ввод дополнительной информации в поле. Это дает злоумышленникам возможность передать в базу данных собственные запросы через поля ввода формы. Они могут использовать эту уязвимость в разных преступных целях, начиная с кражи конфиденциальных данных и заканчивая манипулированием сведениями в базе.

Прежде всего запомните, что безопасных систем не бывает, поэтому нужно постоянно искать уязвимости, которые могут быть у вашего сайта:

### Не пользуйтесь методом GET в формах

Заключение