Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЁТ
к лабораторной работе №6
на тему

ЗАЩИТА ОТ АТАКИ МЕТОДОМ ВНЕДРЕНИЯ SQL-КОДА

Выполнил: студент гр.253505 Сенько. Н. С.

Проверил: ассистент кафедры информатики
Герчик А.В.

Минск 2025

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ

Цель данной работы заключается в исследовании природы и механизмов возникновения уязвимостей, связанных с SQL-инъекциями, а также в разработке программы, наглядно демонстрирующей уязвимый и защищённый подходы к обработке пользовательского ввода в базу данных.

SQL-инъекции представляют собой один из наиболее распространённых типов атак на веб-приложения. Они позволяют злоумышленникам выполнять произвольные SQL-запросы, извлекать, изменять или удалять данные, а в некоторых случаях — получать полный контроль над базой данных.

В рамках данной работы были созданы две версии программы на языке Python с использованием библиотеки psycopg2 для взаимодействия с базой данных PostgreSQL.

Уязвимая версия программы формирует SQL-запросы путём объединения строк, что позволяет злоумышленнику внедрить вредоносный код, например, используя конструкцию «OR '1' = '1». Защищённая версия, напротив, использует параметризованные запросы, которые обрабатывают пользовательский ввод как обычные данные, исключая возможность выполнения произвольных команд.

Такая реализация позволяет наглядно продемонстрировать, как простые, но критически важные меры защиты могут предотвратить потенциальные атаки. Использование подготовленных запросов, проверка и экранирование пользовательских данных являются ключевыми элементами безопасной разработки, которые минимизируют риск эксплуатации уязвимостей и обеспечивают стабильную работу программы.

В результате работы подчёркивается важность внимательной обработки пользовательского ввода и демонстрируется, что внедрение базовых защитных механизмов значительно повышает уровень безопасности программного обеспечения.

2 ХОД РАБОТЫ

В процессе выполнения работы сначала необходимо подготовить программу, чтобы она могла корректно взаимодействовать с базой данных. Для этого запускается *Python*-скрипт в терминале, и программа выводит текущий статус защиты от *SQL*-инъекций, что позволяет сразу понять, в каком режиме она функционирует – защищённом или уязвимом.

Затем, для демонстрации уязвимости, отключается защита, что можно сделать, изменив флаг *PROTECTION_ENABLED* на *False*. После этого в поля ввода логина и пароля можно ввести специально сформулированную строку, например, ' *OR* '1'='1, которая является типичным примером *SQL*-инъекции. В уязвимом режиме программа, не проверяя ввод, выполняет запрос, авторизуя пользователя даже при заведомо неверных данных.

После демонстрации уязвимости включается защита путём установки флага *PROTECTION_ENABLED* в *True*. Повторный ввод той же *SQL*-инъекции теперь не приведёт к авторизации – программа отклонит запрос, потому что параметризованные запросы корректно обрабатывают пользовательский ввод, исключая возможность выполнения вредоносного кода.

В финале анализа сравнивается поведение программы в двух режимах, чтобы на практике оценить эффективность защитных механизмов. На основании наблюдений делается вывод о важности использования подготовленных запросов и правильной обработки входных данных. Такая демонстрация позволяет чётко увидеть, насколько критичны меры предосторожности при работе с пользовательским вводом, и как даже простые методы могут защитить приложение от серьёзных угроз безопасности.

ЗАКЛЮЧЕНИЕ

В ходе исследования были рассмотрены методы SQL-инъекций, их воздействие на безопасность веб-приложений, а также способы защиты от подобных атак.

Разработка двух версий программы — уязвимой и защищённой — позволила наглядно продемонстрировать, насколько опасны некорректная обработка пользовательских данных и как применение простых, но эффективных методов, таких как параметризованные запросы, может предотвратить потенциальные угрозы.

Результаты исследования указывают на то, что защита от SQL-инъекций является неотъемлемой частью разработки безопасных приложений. Даже базовые меры, такие как проверка вводимых данных и ограничение доступа к базе данных, значительно снижают вероятность успешного взлома.

В целом, исследование подчёркивает важность внимательного подхода к безопасности на всех этапах разработки программного обеспечения. Применение современных методов защиты и регулярное тестирование на уязвимости способствуют созданию надёжных систем, способных противостоять реальным угрозам в условиях постоянно меняющегося ландшафта киберугроз.