МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Анализ инъекций SQL

АВТОРЕФЕРАТ
дипломной работы

студентки 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Рубцовой Марты Александровны

Научный руководитель М. Б. Абросимов
д. ф.-м. н., доцент _____
22.01.2022 г.

Заведующий кафедрой М. Б. Абросимов
д. ф.-м. н., доцент _____
22.01.2022 г.

Саратов 2022

ВВЕДЕНИЕ

Современное интернет-пространство построено на основе модели клиент-серверного взаимодействия. Серверы получают запросы пользователя и взаимодействуют с внутренней базой данных, возвращая соответствующие ответы на запросы. Базы данных могут содержать в том числе различные приватные данные: номера кредитных карт, ценные документы, пароли, финансовые, медицинские реквизиты и другую частную информацию. Следовательно, обеспечению безопасности написания веб-приложений, управляемых базами данных, необходимо уделить особое внимание.

Одним из распространенных способов взлома сайтов и программ, основанных на базах данных, является внедрение вредоносного SQL-кода. С его помощью можно обойти аутентификацию, получить доступ, изменить и удалить данные.

Цель дипломной работы сфокусирована на изучении SQL-инъекций и реализации системы, способной распознать и тем самым предотвратить возможность внедрения в веб-приложение несанкционированного SQL-кода.

Дипломная работа состоит из введения, 5 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 42 страницы, из них 39 страниц – основное содержание, включая 18 рисунков и список использованных источников из 24 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

1 Основные понятия: функционирование веб-приложений

В данном разделе вводятся основные понятия, связанные с технологией работы веб-сервера, его взаимодействием с базой данных и различием систем управления базами данных.

1.1 Технология работы веб-сервера

Физически веб-серверы – это компьютеры, хранящие файлы сайта и доставляющие их на устройство пользователя. Со стороны ПО веб-серверы производят контроль доступа веб-пользователя к файлам, размещенным на сервере.

Запрос к серверу определяется такими параметрами как метод сервера, адрес страницы, версия протокола.

1.2 Взаимодействие веб-сервера и базы данных

База данных – это упорядоченный набор структурированных самостоятельных материалов, хранящихся в электронном виде в компьютерной системе.

Такие составляющие архитектуры веб-приложения как база данных, модуль, работающий под управлением браузера, модуль, работающий под управлением веб-сервера порождают два вида связи серверной части: с браузером и с базой данных.

1.3 Различия СУБД

База данных управляется **системой управления базами данных** – совокупностью программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Известные СУБД MySQL, MSSQL, MSACCESS, PostgreSQL и наиболее используемая по статистике DB-Engines СУБД Oracle схожи и отличны между

собой в различных операциях. Например, объединение запросов «union» является общим для данных СУБД, комментирование «--» является общим для них за исключением MSAccess, объединение строк схоже у Oracle и PostgreSQL.

2 Базы данных и реестры уязвимостей

Уязвимость представляет собой недостаток в системе, маневрируя которым, можно намеренно нарушить ее целостность и вызвать неправильную работу.

Данный раздел содержит описание, структуру и возможности таких баз данных и реестров уязвимостей как ФСТЭК и CVE.

2.1 ФСТЭК

Федеральная служба по техническому и экспортному контролю – это федеральный орган исполнительной власти России, осуществляющий реализацию государственной политики, организацию межведомственной координации и взаимодействия, специальные и контрольные функции в области государственной безопасности.

2.2 CVE

Common Vulnerabilities and Exposures – это международная база данных общеизвестных уязвимостей информационной безопасности.

3 SQL-инъекции

Раздел 3 включает в себя общие сведения об SQL-инъекциях, их классификацию, эксплуатацию, причины возникновения, примеры данной уязвимости в CVE и способы защиты.

3.1 Общие сведения

SQL-инъекция – это одна из уязвимостей системы безопасности, позволяющаая злоумышнику завладеть доступом к SQL-запросам в базу данных.

3.2 Классификация SQL-инъекций

Известны разные классификации SQL-инъекций, различающиеся по разным показателям. Например, по типу входящего параметра.

Наиболее подробно в работе рассмотрена классификация, изображенная на рисунке 1.

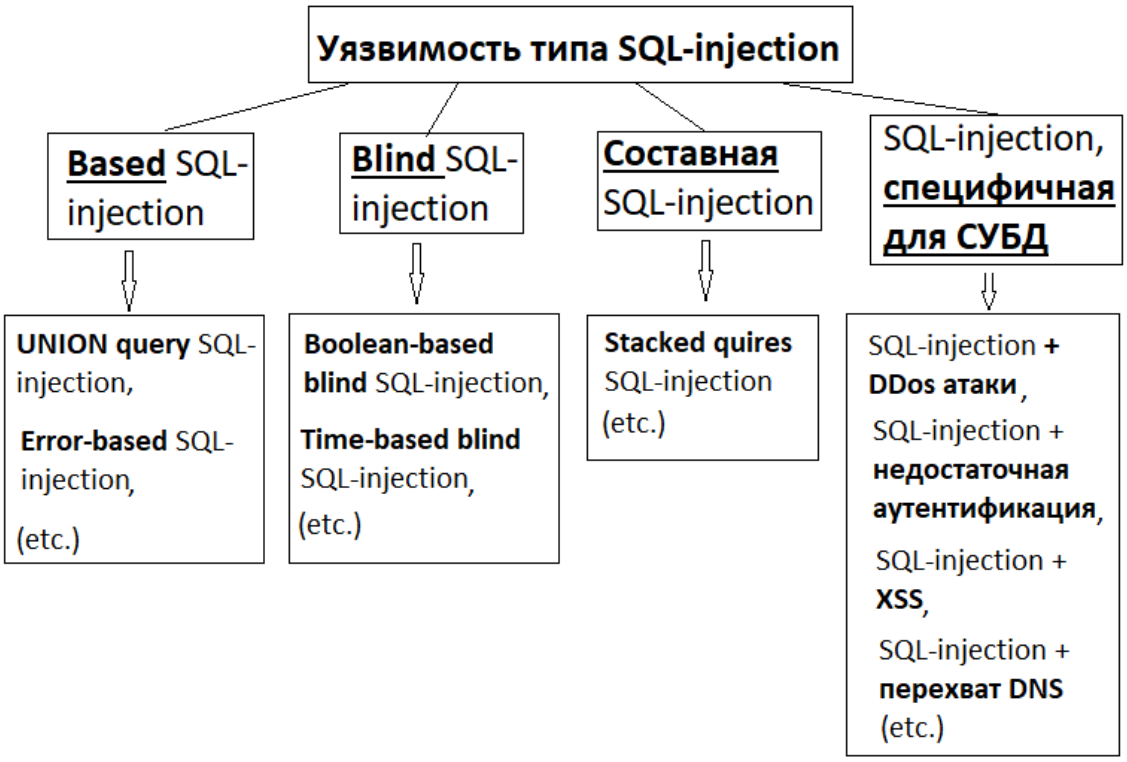

Рисунок 1 – Классификация SQL-инъекций

3.2.1 Based SQL-инъекция

Пункт 3.2.1 содержит описание таких техник эксплуатации классического варианта SQL-инъекции как **UNION query SQL-инъекции** и **Error-based SQL-инъекции.**

3.2.2 Blind SQL-инъекция

В Blind SQL-инъекции данные, содержащиеся в базе данных, напрямую в исходном виде уязвимое веб-приложение не возвращает. Информация взимается «вслепую».

Данный пункт содержит описание **Boolean-based SQL-инъекции** и **Time-based blind SQL-инъекции.**

3.2.3 SQL-инъекция, специфичная для СУБД

Специфичная для СУБД SQL-иньекция работает на конкретный тип СУБД. Пример такой инъекции – **Stacked queries SQL-инъекция**.

3.2.4 Составная SQL-инъекция

Составные SQL-инъекции представляют собой смешение различных уязвимостей и атак:

- SQL-инъекция + недостаточная аутентификация;
- SQL-инъекция + DDoS-атаки;
- SQL-инъекция + перехват DNS;
- SQL-инъекция + XSS.

3.3 Эксплуатация SQL-инъекций

Процесс эксплуатации заключается в пяти основных этапах, изображенных на рисунке 2 и подробно рассмотренных в следующих пунктах:

3.3.1 Выявление SQL-инъекций

3.3.2 Определение типа и версии СУБД

3.3.3 Определение имени пользователя и привилегий

3.3.4 Повышение привилегий

3.3.5 Внедрение SQL-инъекций

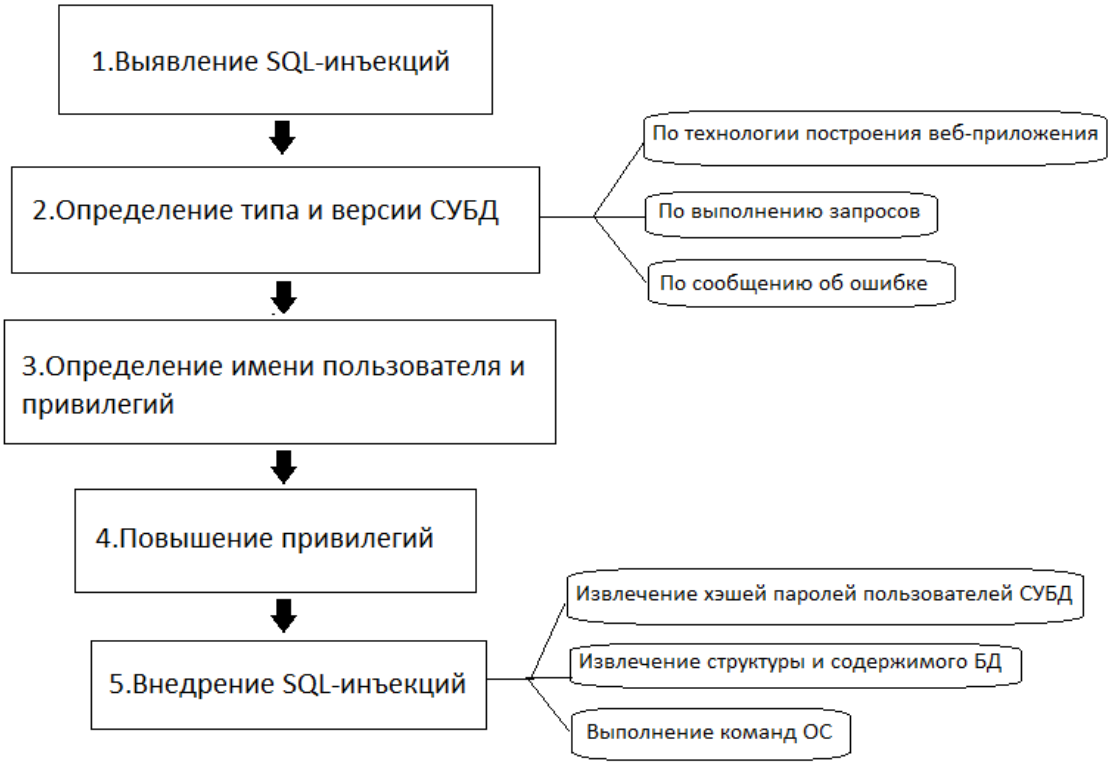

Рисунок 2 – Эксплуатация SQL-инъекций

3.4 Причины возникновения SQL-инъекций

Существуют основные причины возникновения SQL-инъекций в веб-приложениях, взаимодействующих с СУБД:

- Динамическое построение SQL-запросов;
- Некорректная обработка исключений;
- Некорректная обработка специальных символов;
- Некорретктная обработка типов;
- Небезопасная конфигурация СУБД.

3.5 SQL-инъекции в CVE

Международная база данных общеизвестных уязвимостей информационной безопасности CVE содержит в том числе уязвимости вида SQL-инъекций. На дату 10.01.22 в базе имеется 9242 записи данной уязвимости, 5 из которых – за 2022 год.

В подразделе рассмотрены примеры последних записей данной

уязвимости.

3.6 Способы защиты от SQL-инъекций

Существует множество способов препятствия возможной атаке. Следует использовать подготовленные запросы или функции экранирования соответствующих расширений работы с базами данных.

К эффективным способам защиты от SQL-инъекций относятся:

- функция mysql_real_escape_string;
- приведение данных к числу;
- подготовленные запросы;
- готовые библиотеки.

К неэффективным способам защиты от SQL-инъекций относятся:

- функция htmlspecialchars();
- фильтрация по черному списку символов;
- функция stripslashes();
- функция addslashes().

4 Разбор существующих сканеров уязвимостей

В качестве примеров средств, осуществляющих контроль над уязвимостями тех или иных технических продуктов, рассматриваются сканеры Redcheck, SqlMap.

4.1 Redcheck

Redcheck – это многофункциональное средство контроля защищенности и аудита хостов и сетевого оборудования, разработанное специалистами российской компании «АЛТЭКС-СОФТ», прошедшее сертификацию в системе ФСТЭК России.

4.2 SqlMap

SqlMap – это программное обеспечение, предназначенное для

автоматизированного сканирования уязвимостей SQL и их дальнейшего использования.

5 Реализация SQL-сканера инъекций веб-приложений

Программа sqlinj, реализованная на языке программирования Python, имеет следующие возможности:

- запускается с командной строки;
- получает на вход URL-адрес веб-ресурса;
- получает на вход необязательный параметр – Cookie;
- формирует список внутренних ссылок заданного URL;
- проверяет ресурс на возможность внедрения классических и слепых SQL- инъекций как в строковом, так и в числовом параметрах;
- выдает причину появления инъекций;
- выдает рекомендацию по устранению выявленных инъекций.

5.1 Принцип работы

В программе были использованы дополнительные библиотеки. В том числе:

- Re – предоставляет операции регулярных выражений;
- Argparse – позволяет производить парсинг аргументов;
- Requests – необходима для работы с запросами HTTPS;
- Random – позволяет выполнять функцию рандомизации.

Формируя массив нагрузок из файла payloads.txt, программа поочередно подставляет их, имитируя таким образом эксплуатацию инъекции и проверяя возможность ее исполнения. Если нагрузки сработали, на выходе выдается сообщение, полученное от сервера. Например, это может быть ошибка SQL-синтаксиса. Далее программа выводит разработчикам рекомендацию отфильтровать данные. В частности, использовать mysql_real_escape_string() – функцию, добавляющую обратную косую черту к некоторым символам, чтобы обезопасить данные, вставляемые в запрос перед отправкой его сервером в базу данных.

5.2 Тестирование

Для тестирования программы sqlinj анализа SQL-инъекций рассматриваются веб-приложение DVWA, установленное на дистрибутив Linux, и веб-сайт «Мир графов»: graphworld.ru.

ЗАКЛЮЧЕНИЕ

Определенно, сейчас все реже можно столкнуться с простейшими видами такой уязвимости, как SQL-инъекции. Тем не менее, внедрения вредоносного кода могут приобретать разный характер, и проблема остается актуальной. Разработчикам важно знать, какие угрозы могут найти свое исполнение в неграмотно написанной работе. Необходимо учитывать все нюансы безопасности взаимодействия сервера с СУБД, чтобы избежать несанкционированного распространения засекреченной, частной информации.

В работе были рассмотрены ключевые теоретические особенности SQL-инъекций: общее представление, классификация, актуальность, причины возникновения, средства эксплуатации и борьбы.

Также была выполнена практическая часть, задачей которой стояло сканирование веб-приложений на SQL-инъекции в классическом и слепом вариантах. На языке программирования Python была разработана программа, строящая по заданному обязательному параметру URL и необязательному параметру cookie список внутренних ссылок и определяющая, имеют ли в данном веб-приложении место быть рассмотренные уязвимости.