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

Поиск SQL-уязвимостей в web-приложениях

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Климова Кирилла Владимировича

Научный руководитель
к. п. н., доцент _____ А. С. Гераськин
23.01.2021 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
23.01.2021 г.

Саратов 2021

ВВЕДЕНИЕ

В современном мире практически ни одно технологическое предприятие не может эффективно функционировать без использования автоматизированных информационных систем управления. Использование данных систем значительно повышает эффективность производственных процессов предприятия, но также создает характерные для информационных систем риски, связанные с угрозами безопасности информации. Эксплуатация уязвимостей подобных систем может привести не только к хищению конфиденциальной информации предприятия, но и привести к серьезным сбоям производства и физическому повреждению производственного оборудования.

Большинство подобных систем представляют собой web-ресурс, доступ к которому может быть осуществлен посредством глобальной сети Интернет. Поэтому одна из наиболее важных проблем функционирования таких систем – обеспечение информационной безопасности, главный принцип которой лежит в анализе потенциальных угроз, проведении мер по их устранению и последующем анализе состояния безопасности системы.

Согласно ГОСТ Р 50922-2006 «Защита информации. Основные термины и определения»[1] условием реализации угрозы безопасности обрабатываемой в системе информации может быть недостаток или слабое место (уязвимость) системы. Атака злоумышленников как раз и активирует ту или иную уязвимость, присутствующую в системе.

Рассмотрим одну из наиболее распространенных уязвимостей систем – атака с использованием SQL-инъекций. Это атака, при которой злоумышленником производится внедрение вредоносного SQL-кода в строки, передающиеся на сервер системы управления базой данных (СУБД) для последующего синтаксического анализа и выполнения[2]. Данный способ взлома web-приложений очень распространен и занимает лидирующие места в отчетах организаций в сфере информационной безопасности. Так, например, SQL-

---

[1] ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. М.: Стандартинформ, 2008. - 12 с.

[2] Соколин, Д. Д. Методы комплексного обеспечения безопасности SQL-сервера от атак типа SQL-инъекции. / Д. Д. Соколин, А. С. Тимохович // URL: https://www.elibrary.ru/item.asp?id=28433904 (дата обращения: 20.12.2020). - Загл. с экрана. - Яз. рус.

инъекции в отчете Positive Technologies, международной компании, специализирующаяся на разработке программного обеспечения в области информационной безопасности, опубликовала статистику за 2018 год, в которой SQL-инъекции, располагаясь на первом месте, занимают порядка 30% от общего количества атак на web-приложения за данный период[3]. Большинство атак пришлось на государственные организации и финансовые учреждения. Поэтому можно с полной уверенностью говорить об актуальности проблемы защиты информации от атак с применением SQL-инъекций.

Целью данной работы является разработка программного обеспечения для поиска SQL-инъекций в web-приложениях.

Для решения поставленной цели необходимо выполнить следующие задачи:

- провести исследование видов уязвимостей и определить основные причины их появления;
- рассмотреть пример атаки с использованием SQL-инъекции;
- изучить процесс обнаружения SQL-уязвимостей;
- проанализировать представленное на рынке программное обеспечение для поиска SQL-уязвимостей;
- осуществить разработку программного обеспечения для поиска SQL-уязвимостей в web-приложениях.

Дипломная работа состоит из введения, 4 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 74 страницы, из них 39 страниц – основное содержание, включая 19 рисунков, список использованных источников из 20 наименований.

[3] Positive Technologies [Электронный ресурс]: Уязвимости в АСУ ТП: итоги 2018 года. URL: https://www.ptsecurity.com/ru-ru/research/analytics/ics-vulnerabilities-2019/?sphrase_id=66914 (дата обращения: 11.10.2020). - Загл. с экрана. - Яз. рус.

КРАТКОЕ СОДЕРЖАНИЕ

Первый раздел посвящен знакомству с SQL-инъекциями. Он состоит из 5 подразделов. В первом дается понятие SQL-инъекции, а также вводная информация. Во втором подразделе рассматриваются основные виды SQL-инъекций, при помощи которых злоумышленник может получить доступ к содержимому БД. В результате были выделены следующие виды:

- **Union-based SQL-инъекция**. Это способ внедрения SQL – кода, когда в уязвимый параметр происходит передача выражения, начинающегося с «UNION ALL SELECT». Эта техника работает, когда web-приложения напрямую возвращают результат вывода команды SELECT на страницу: с использованием цикла for или похожим способом, так что каждая запись полученной из БД выборки последовательно выводится на страницу[4];
- **Error-based SQL-инъекция**. Данная техника используется, когда приложение некорректно обрабатывает исключения, возникающие при работе с СУБД, и сообщение о возникшем исключении отображается пользователю;
- **Stacked Queries SQL-инъекция**. Stacked Queries обеспечивают большой контроль злоумышленнику. Завершив исходный запрос и добавив новый, можно будет изменить данные и вызвать хранимые процедуры. Этот метод широко используется в атаках с использованием SQL-инъекций, и понимание его принципа необходимо для правильного понимания этой проблемы безопасности;
- **Blind SQL-инъекция.** Используется данная техника в случае, если нельзя использовать Union и Error-based SQL-инъекцию, то есть если результат выполнения запроса не отображается пользователю, либо приложение корректно обрабатывает исключения. Однако, если при этом, модифицируя запрос, можно влиять на логику работы приложения: при определенных входных данных некоторые страницы

[4] Пестриков, Р. А. Исследование и устранение уязвимости SQL-инъекции в плагине Apptha WordPress Video Gallery для CMS WordPress. / Р. А. Пестриков // URL: http://vestnik.psu.ru/docs/2018/3/1/201831100.pdf (дата обращения: 07.11.2020). - Загл. с экрана. - Яз. рус.

отображаются неправильно или запрос возвращает только часть информации. В этом случае можно использовать технику Blind SQL – инъекции. Составляется SQL – выражение, которое при истинном значении не нарушает логику работы приложения. При ложном же значении возникает аномальное поведение в работе web-приложения: страницы неправильно отображаются либо возвращается только часть данных[5];

- **Time-based SQL-инъекция.** Данная техника похожа на Blind SQL – инъекцию. Также составляется SQL-выражение, но анализируется не возвращаемый результат, а время отклика сервера СУБД[6]. При ложном значении SQL-выражения время отклика незначительно, а при истинном значении составляет несколько секунд. Используя данную технику, можно проверять логические условия, например, наличие роли администратора базы данных у текущего пользователя, а также посимвольно извлекать данные из таблиц БД. Time-based SQL-инъекция выполняется медленнее чем Blind SQL-инъекции[7];
- **Out-Of-Band SQL-инъекция.** В случае если ни одна из вышеперечисленных техник эксплуатации SQL-инъекций не применима можно воспользоваться Out-Of-Band техникой. Для применения данной техники необходим удаленный сервер, который находится под контролем злоумышленника, либо доступ к директории на сервере СУБД. Техника заключается в следующем: злоумышленник направляет вывод результата SQL – запроса на удаленный сервер, используя протоколы DNS, HTTP, SMTP, или осуществляет запись в

[5] Justin Clarke. SQL Injection Attacks and Defense. Syngress ISBN: 978-1- 59749-963-7 pp. 325-350. - Яз. англ.
[6] Алекперов, З. А. Анализ угроз безопасности веб-приложений. / З. А. Алекперов // URL: https://www.elibrary.ru/item.asp?id=38937186 (дата обращения: 02.01.2021). - Загл. с экрана. - Яз. рус.
[7] Национальной библиотеки им. Н. Э. Баумана [Электронный ресурс]: SQL – инъекция. URL: https://ru.bmstu.wiki/SQL-%D0%B8%D0%BD %D1%8A%D0%B5%D0%BA%D1%86%D0%B8%D1%8F (дата обращения: 06.01.2021). Загл. с экрана. Яз. рус.

файл, который расположен в доступной для злоумышленника директории на сервере СУБД[8].

Третий подраздел посвящен основным причинам возникновения SQL-уязвимостей. Были рассмотрены следующие виды:

- динамическое построение SQL-запросов;
- некорректная обработка исключений;
- некорректная обработка специальных символов;
- некорректная обработка типов;
- небезопасная конфигурация СУБД.

Четвертый и пятый подразделы посвящены примерам атак с использованием SQL-инъекции.

Второй раздел «Обнаружение и методы борьбы с SQL-уязвимостями» состоит из двух подразделов. В первом рассматривается процесс выявления и эксплуатации SQL-уязвимости начиная с обнаружения уязвимого параметра и заканчивая извлечением требуемой информации из БД. Данный процесс состоит из пяти основных шагов:

- выявления SQL-инъекции;
- определения типа и версии СУБД;
- определения имени пользователя и его привилегий;
- повышения привилегий;
- эксплуатации уязвимости.

Второй подраздел содержит описание методов борьбы с SQL-уязвимостями. Выделяются следующие методы:

- фильтрация строковых параметров;
- фильтрация целочисленных параметров;
- усечение входных параметров;
- автоматическое определение зарезервированных SQL-слов в тексте и блокировка пользователей;
- использование параметризованных запросов;

---

[8] Acunetix [Электронный ресурс]: Blind Out-of-band SQL Injection vulnerability testing added to AcuMonitor URL: https://www.acunetix.com/blog/articles/blind-out-of-band-sql-injection-vulnerability-testing-added-acumonitor/

- использование принципа наименьших привилегий при предоставлении доступа к базам данных;
- использование тестирования и мониторинга для защиты от SQL-инъекций.

Третий раздел содержит описание и анализ существующего на данный момент на рынке программного обеспечения, осуществляющего поиск SQL-уязвимостей. Данные программы обычно имеют две составляющих – сканирование сайта на возможные уязвимости и их использование для получения доступа к данным. Их функционал в значительной мере облегчает проверку сайта на возможность взлома SQL-инъекцией[9]. В данном разделе рассмотрено 4 программы: jSQL Injection, Sqlmap, SQLi Dumper v.7 и Acunetix Web Vulnerability Scanner.

Анализ существующего программного обеспечения показал, что программы для поиска SQL-уязвимостей, представленные на рынке, имеют не только плюсы, но и минусы. Общим для многих программ недостатком является медленная работа, сложный интерфейс и неинформативные отчеты.

Четвертый раздел «Программная реализация SQL-сканнера» содержит описание и тестирование реализованного в ходе выполнения дипломной работы продукта. Программа написана на языке Python с использованием библиотеки requests, которая позволяет выполнять HTTP-запросы. Для реализации программного интерфейса использовалась библиотека PySimpleGUI. Также при помощи библиотеки xlswriter была осуществлена генерация отчетов о результате сканирования web-сайта.

На вход программа получает абсолютный URL тестируемого сайта, Cookie и параметры, в которые будет внедряться SQL-инъекция. Также после сканирования в поле Отчет отображается краткая информация о результате работы программы, где указывается тип используемой базы данных и уязвимый URL с успешно внедренной инъекцией. Более подробный отчет с результатом сканирования генерируется автоматически и выводится в файл report.xlsx. В

(дата обращения: 15.12.2020). - Загл. с экрана. - Яз. англ.

[9] IT-Black [Электронный ресурс]: SQL – инъекция. URL: https://it-black.ru/sql-inyektsiya/ (дата обращения: 19.10.2020). - Загл. с экрана. - Яз. рус.

отчете указывается количество найденных уязвимостей, тип базы данных, используемой на данном web-сайте, время работы программы, дата и время выполнения сканирования, страница, на которой найдена уязвимость, и сама инъекция, при помощи которой уязвимость была обнаружена.

Функция getDatabase() отправляет GET-запрос на сервер и при помощи регулярных выражений в зависимости от ответа сервера определяет тип используемой базы данных. Функция define_database_type() на основе полученного типа БД добавляет инъекции для данной БД в общий список инъекций для внедрения, а затем функция scanVuln() определяет уязвим ли URL посредством анализа контента страницы.

Тестирование программы будем производить на специальных сайтах, предназначенных для учебных целей. В качестве такого ресурса было выбрано bWAPP – бесплатное и заведомо небезопасное web-приложение, написанное на PHP и использующее базу данных MySQL[10].

В результате тестирования программы были выделены определенные преимущества. Главными преимуществами являются возможность определения существования уязвимости на сайте, скорость работы программы, подробный вывод информации в виде отчета, а также простой интерфейс.

[10] bWAPP [Электронный ресурс]: bWAPP an extremely buggy web app! URL: http://www.itsecgames.com/ (дата обращения: 02.01.2021). - Загл. с экрана. - Яз. англ.

ЗАКЛЮЧЕНИЕ

В процессе работы, целью которой являлась разработка программного обеспечения для выявления и эксплуатации SQL-уязвимостей в web-приложениях, были решены поставленные в начале исследования задачи. Были рассмотрены виды SQL-уязвимостей. Этими видами являются Union-based, Error-based, Stacked Quaries, Blind, Time- based и Out-Of-Band SQL-уязвимости. Также были определены основные причины появления SQL-уязвимостей, которые включают в себя динамическое построение SQL-запросов, некорректную обработку исключений, специальных символов и типов, и небезопасную конфигурацию СУБД.

Были рассмотрены примеры атак с использованием данных видов уязвимостей и был изучен процесс обнаружения и эксплуатации SQL-уязвимостей, который заключается в выявлении SQL-инъекции, определении типа и версии СУБД, определении имени пользователя и его привилегий, повышении привилегий и эксплуатации уязвимости. Помимо этого, были описаны методы борьбы с SQL-уязвимостями: фильтрация строковых параметров, фильтрация целочисленных параметров, усечение входных параметров, автоматическое определение зарезервированных SQL-слов в тексте и блокировка пользователей, использование параметризованных запросов, использование принципа наименьших привилегий при предоставлении доступа к базам данных, использование тестирования и мониторинга для защиты от SQL-инъекций.

Также было проанализировано существующее на рынке программное обеспечение для поиска SQL-уязвимостей. Более детально рассматривались программы jSQL Injection, Sqlmap, SQLi Dumper v.7 и Acunetix Web Vulnerability Scanner.

На основе данного анализа было разработано собственное приложение для поиска SQL-уязвимостей в web-приложениях. Данное ПО позволяет осуществлять поиск SQL-уязвимостей на сайтах посредством внедрения инъекций в параметры запросов. Разработанный программный продукт может использоваться различными компаниями и частными лицами с целью проверки

собственного web-ресурса на наличие SQL-уязвимостей для предотвращения утечек конфиденциальной информации и ущерба, связанного с посторонним вмешательством в базы данных.

Разработанный сканнер отлично показал себя при тестировании, которое проводилось на заведомо уязвимых сайтах, предназначенных для учебных целей. Тестирование показало, что разработанное ПО имеет свои преимущества. Главными преимуществами являются возможность определения существования уязвимости на сайте, скорость работы программы, а также подробный вывод информации в виде отчета.

Все поставленные цели были достигнуты, задачи выполнены.