Шанин П.С., Нижлукченко И.Д. Парсинг и защита от парсинга ВЕБ - страниц и баз данных // Международный журнал информационных технологий и энергоэффективности.– 2025. – Т. 10 № 12(62) с. 132–144

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.056

ПАРСИНГ И ЗАЩИТА ОТ ПАРСИНГА ВЕБ - СТРАНИЦ И БАЗ ДАННЫХ

[1]Шанин П.С., **[2]Нижлукченко И.Д.**
ФГБОУ ВО САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФЕССОРА М. А. БОНЧ-БРУЕВИЧА, Санкт-Петербург, Россия (193232, г. Санкт-Петербург, просп. Большевиков, 22, корп. 1), e-mail: [1]pasha_pavel_08@bk.ru, [2]nizhluchenk@gmail.com

В данной работе представлен развернутый анализ проблемы автоматизированного извлечения информации с веб-сайтов и баз данных, рассматриваемой как одна из острых проблем в сфере безопасности информации. Особое внимание уделяется принципам и технологиям, лежащим в основе автоматического сбора данных, а также ключевым опасностям, возникающим при незаконном использовании этих методов. Эти опасности включают в себя раскрытие конфиденциальных сведений, нарушение прав интеллектуальной собственности и получение несправедливых преимуществ в конкурентной борьбе. В статье приводятся иллюстративные примеры разрешенного скраппинга, осуществляемого в рамках собственных цифровых активов. Обсуждаются наиболее действенные способы защиты от несанкционированного сбора данных, а также связанные с ними слабые места и общая необходимость комплексного подхода к обеспечению защиты информации.

Ключевые слова: Парсинг, базы данных, API, HTML-код, веб-страница, ботнет, кибербезопасность, скраппинг, агент пользователя, CAPTCHA

WEB PAGE AND DATABASE PARSING AND ANTI-PARSING PROTECTION

[1]Shanin P.S., **[2]Nizhlukchenko I.D.**
ST. PETERSBURG STATE UNIVERSITY OF TELECOMMUNICATIONS NAMED AFTER PROFESSOR M. A. BONCH-BRUEVICH, St. Petersburg, Russia (193232, St. Petersburg, ave. Bolshevikov, 22, bldg. 1), e-mail: [1]pasha_pavel_08@bk.ru, [2]nizhluchenk@gmail.com

This paper presents a comprehensive analysis of the automated extraction of information from websites and databases, considered a pressing issue in the field of information security. Particular attention is paid to the principles and technologies underlying automatic data collection, as well as the key dangers arising from the illegal use of these methods. These risks include disclosure of confidential information, intellectual property infringement, and unfair competitive advantages. This article provides illustrative examples of legitimate scraping practices within proprietary digital assets. It discusses the most effective methods for protecting against unauthorized data collection, as well as the associated vulnerabilities and the overall need for a comprehensive approach to information security.

Keywords: Scraping, databases, API, HTML code, web page, botnet, cybersecurity, scraping, user agent, CAPTCHA.

В современном цифровом мире, где информация является одним из самых ценных ресурсов, парсинг, как метод извлечения данных, не теряет своей актуальности или эффективности. Процесс предоставляет множество возможностей для общего анализа, агрегирования и иных процессов, что делает его важным инструментом для бизнеса, научных исследований и разработки ПО. С обратной стороны, несанкционированный применение процесса может представлять серьезные угрозы для владельцев сайтов и баз данных,

влекущий за собой риски утечки конфиденциальных данных, нарушения авторских прав и финансовых потерь.

Парсинг (скраппинг) представляет собой процесс автоматического извлечения данных из различных источников, таких как веб-страницы, API и БД. Этот процесс может быть использован как для законных целей, так и для противоправных действий. Основные понятия и принципы включают в себя методы обработки HTML-кода, работы с API и извлечения данных из структурированных и неструктурированных источников. Понимание этих основ поможет лучше осознать, какие угрозы могут возникнуть в результате и как можно защитить свои ресурсы от несанкционированного доступа. [7]

Угрозы и риски парсинга сайтов могут различаться от кражи контента и интеллектуальной собственности, до создания сети ботнетов для проведения DDoS-атак. В условиях, когда конкуренция на рынке возрастает, компании могут сталкиваться с ситуациями, когда их данные становятся объектом интереса со стороны конкурентов или злоумышленников. Это может привести к утечкам ценной информации, нарушению работы сайта и репутационным потерям.

Актуальность темы исследования обусловлена растущими угрозами, связанными с несанкционированным извлечением данных, что может привести к утечкам конфиденциальных данных и нарушению прав пользователей. В условиях стремительного развития цифровых технологий и увеличения объема данных, доступных в интернете, важность эффективных методов защиты становится всё более очевидной. [6]

Принципы парсинга

1. Веб-страницы

Работа автоматизированного сбора данных основана на использовании специализированных технологий и алгоритмов, предназначенных для автоматического разбора HTML-кода, выделения необходимой информации и ее последующей обработки в структурированном виде. Типичный процесс автоматизированного сбора данных с веб-страниц включает несколько этапов: отправку HTTP(S)-запроса на целевой сервер, получение ответа с HTML-кодом, его синтаксический разбор и извлечение требуемых данных. На первом этапе программа, осуществляющая автоматизированный сбор данных, формирует и отправляет HTTP-запрос, указывая адрес нужного ресурса. Сервер обрабатывает запрос и возвращает HTML-код страницы. Далее выполняется анализ полученного кода. Для этого используются программные библиотеки и фреймворки, например, Beautiful Soup, Scrapy или lxml в Python. Эти инструменты предоставляют удобные средства для навигации по дереву документа и позволяют находить и извлекать нужные элементы — текстовые блоки, ссылки, метаданные, изображения и другие структурированные данные.

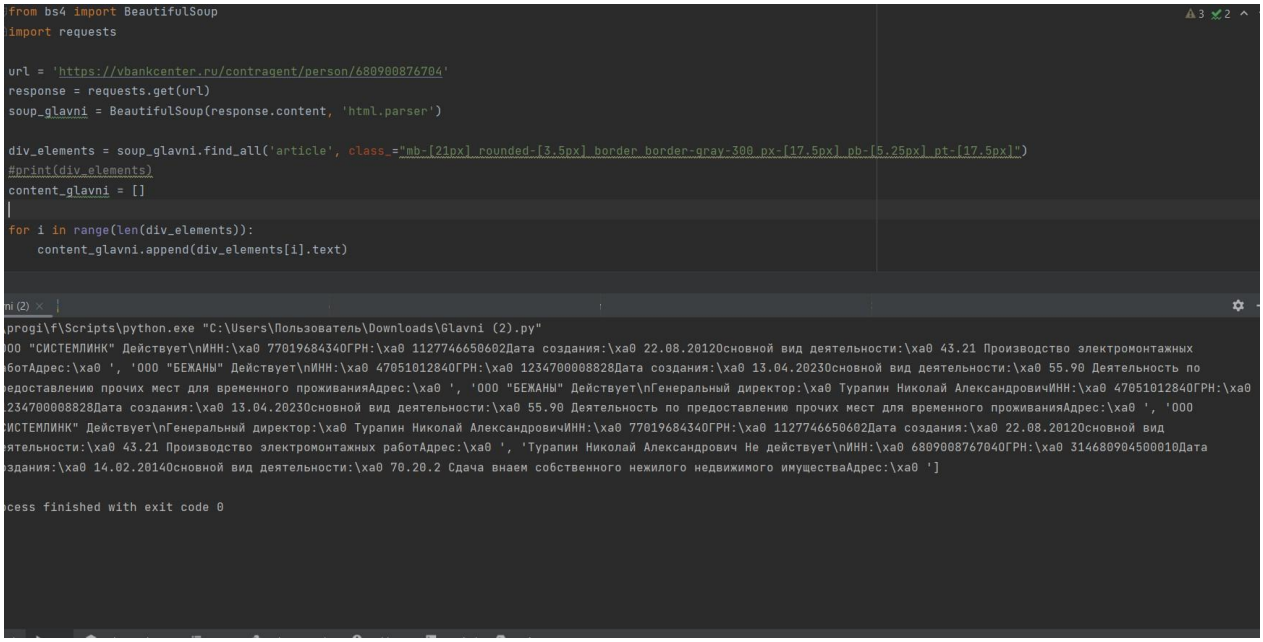

Рисунок 1 - Простой процесс с применением Beautiful Soup

Следующим этапом является обработка и структурирование извлечённых данных. На данной стадии парсер выполняет задачи очистки, нормализации и трансформации сырых данных в форматы, пригодные для последующего хранения и анализа, такие как CSV (Comma-Separated Values), JSON (JavaScript Object Notation) или реляционные и нереляционные БД. Это обеспечивает возможность эффективной работы с массивом собранных данных, проведения статистического анализа, машинного обучения и визуализации результатов. Крайне важно подчеркнуть, что технология сама по себе является нейтральной, а её правовой статус и этическая оценка целиком зависят от целей и контекста применения. Легитимный (разрешённый) парсинг осуществляется в рамках, установленных владельцем ресурса (например, с учётом правил, определённых в файле robots.txt и пользовательском соглашении), и применяется для таких задач, как сбор данных из открытых источников, академические исследования, мониторинг рыночных цен и агрегация публичной информации. Нелегитимный (несанкционированный), напротив, связан с нарушением установленных правил доступа, обходом технических средств защиты, присвоением контента, охраняемого авторским правом, или использованием данных способом, наносящим ущерб владельцу ресурса или третьим лицам. Таким образом, разграничение между допустимым и противоправным использованием определяется не технической реализацией, а соответствием действий существующему законодательству и договорённостям с правообладателем. [8]

2. Базы данных

Ключевая особенность данного вида процесса, заключается в доступе к данным часто требует наличия определенных прав и учетных данных. Обычно имя пользователя, пароль и, возможно, дополнительные механизмы аутентификации, такие как токены или ключи API. Без надлежащих прав доступа попытки извлечь данные могут привести к ошибкам, блокировкам и при наличии сопутствующих обстоятельств, к юридическим последствиям.

Доступ данных через открытый интерфейс НЕ всегда определяет, законность. Защита авторских прав, соглашения о лицензировании и другие юридические аспекты могут ограничивать использование полученной информации.

Поскольку взаимодействие в данном случае может осуществляться через интерфейсы, такие как SQL-запросы, API или специальные библиотеки, необходимо учитывать, особенности и сопутствующие ограничения каждой из этих технологий. SQL-запросы позволяют извлекать данные в структурированном виде, но требуют знания языка SQL и структуры самой БД. В то же время API могут предоставлять более удобный способ доступа к данным, но могут также иметь ограничения по количеству запросов или объему извлекаемых данных. Кроме того, некоторые API могут использовать механизмы защиты, такие как ограничение по времени или количеству запросов, что усложняет процесс. [5]

```python
import pypyodbc
def search_words_in_database(database, words):
    connection = pypyodbc.connect(database)
    cursor = connection.cursor()

    cursor.execute("SELECT name FROM sys.tables")  # Получаем список всех таблиц в базе данных

    tables = cursor.fetchall()

    results = []

    for table in tables:
        table_name = table[0]
        cursor.execute(f"SELECT * FROM {table_name}")

        columns = [column[0] for column in cursor.description]
```

```
baza
Введите ключевые слова через запятую: Шанин
База данных: Driver={SQL Server};Server=DESKTOP-QJQ6H9L;Database=222;
Таблица: Лист1$
(1.0, 'Шанин Павел сергеевич', datetime.datetime(2001, 5, 4, 0, 0), 'Новоколомяжский пр. 13', 'БОНЧ')
База данных: Driver={SQL Server};Server=DESKTOP-QJQ6H9L;Database=222;
Таблица: Лист2$
(1.0, 'Шанин Павел сергеевич', datetime.datetime(2001, 5, 4, 0, 0), 'Новоколомяжский пр. 13', 'нет')
База данных: Driver={SQL Server};Server=DESKTOP-QJQ6H9L;Database=222;
Таблица: Таблица
(1.0, 'Шанин Павел сергеевич', datetime.datetime(2001, 5, 4, 0, 0), 'Новоколомяжский пр. 13', 'БОНЧ')
```

Рисунок 2 Пример простого парсера базы данных

Производительность при процессе играет крайне важную роль. Извлечение больших объемов данных требует значительных ресурсов как со стороны клиента, так и со стороны сервера. Что приводит к замедлению работы системы, особенно если запросы выполняются неэффективно или если сервер не оптимизирован для обработки больших объемов данных. Необходим учет и понимание не только требований к данным, но и возможности инфраструктуры, на которой они хранятся. [6]

Угрозы и риски парсинга

Одной из основных угроз, связанных с скраппингом, является возможность *утечки* конфиденциальных данных. Последствия негативно скажутся как на репутации компании так и финансовом благополучии пользователей, которые могут стать жертвами

мошенничества или кражи личных данных. Соответственно безопасность становится критически важной задачей для компаний, которые обрабатывают и хранят подобные данные, так как они должны обеспечить надежные механизмы защиты, чтобы предотвратить несанкционированный доступ.

Другой серьезной угрозой является возможность использования процесса для проведения *конкурентного анализа*. Это может дать конкурентам преимущество на рынке, позволяя им манипулировать ценами и предлагать более выгодные условия для клиентов. В результате легитимные компании могут потерять свою долю рынка, что может привести к снижению доходов и даже к банкротству.

С технической стороны прасинг имеет свойство вызывать *технические проблемы* для владельцев сайтов процесс умышленно или случайно направляющий большое количество запросов на сервер, приводит к его сбоям. В результате легитимные пользователи могут столкнуться с проблемами доступа к сайту, что негативно скажется на их опыте взаимодействия с ресурсом. Кроме того, такие действия могут привести к увеличению затрат на хостинг и обслуживание сайта, так как владельцам придется инвестировать в более мощные серверные решения для обработки увеличенного трафика. В дополнение, процесс равносильно используется для создания ботнетов и других форм автоматизированного вредоносного ПО. При последующем обнаружении в системе безопасности сайта, несанкционированный доступ к системе упрощается. То приводит к последующей краже данных, повреждению систем и даже кибератакам на другие ресурсы. [1]

С юридической стороны грамотно разработанный парсер может быть использован для обхода систем защиты *авторских прав.* Многие сайты содержат уникальный контент, который защищен авторским правом. Функционал позволяет копирование этого контента и его дальнейшего распространения без разрешения владельцев. [4]

Методы и стратегии противодействия

Следовательно, методы защиты от парсинга становятся важной задачей для веб-разработчиков и администраторов..

1. *Completely Automated Public Turing test to tell Computers and Humans Apart*

Первым и наиболее широко используемым средством защиты служит CAPTCHA. Этот инструмент действует как пропускной пункт, требуя от пользователя успешного решения задачи, легкой для человеческого интеллекта, но сложной для компьютерной программы. Традиционно защита реализуется через распознавание искаженного текста, определение объектов на рисунках, либо выбор связанных тематически изображений из предложенных вариантов. Однако, развитие искусственного интеллекта, особенно машинного обучения и компьютерного зрения, постепенно подрывает надежность классических CAPTCHA. Современные алгоритмы, опирающиеся на сверточные нейронные сети, показывают высокую результативность в решении задач, когда-то считавшихся исключительной способностью человека. Поэтому, эффективность данной защиты как отдельного защитного барьера постоянно ослабевает. [8]

2. *Фильтрация трафика*

Ключевым индикатором потенциальной опасности служит превышение установленных лимитов на количество запросов, поступающих с конкретного IP-адреса или группы адресов за определенный промежуток времени. В ответ на это администраторы прибегают к

блокировке вызывающих подозрения IP-адресов или устанавливают ограничения на скорость передачи данных. Указанные меры могут быть внедрены на разных уровнях инфраструктуры, начиная с настроек веб-сервера и заканчивая использованием межсетевых экранов для веб-приложений и систем предотвращения вторжений (IPS). Эти инструменты позволяют не только пресекать нетипичный трафик, но и анализировать его содержание, выявляя тем самым комплексные атаки. Успешность фильтрации трафика напрямую зависит от корректности установленных пороговых значений и способности системы оперативно реагировать на попытки обхода обароны. [7]

3. Динамическая генерация контента

Вместо отгрузки целого HTML-документа, где уже есть все необходимые данные, сервер отправляет упрощенный HTML-скелет и JavaScript-код. Этот код отвечает за динамическую подгрузку и отображение контента с помощью асинхронных запросов. Это значительно усложняет задачу для простых HTTP-скрейперов, анализирующих статический HTML, так как целевой контент изначально отсутствует и появляется только после выполнения JavaScript на стороне клиента. Однако у этого подхода есть две основные слабости. Во-первых, его защита не эффективна против инструментов автоматизации, способных запускать JavaScript в контролируемой среде. Безголовые браузеры, такие как Puppeteer и Selenium, полностью имитируют работу браузера и позволяют преодолеть эту защиту. Во-вторых, активное использование динамической генерации контента может негативно сказаться на поисковой оптимизации (SEO). Традиционные поисковые краулеры исторически испытывали трудности с индексацией контента, требующего выполнения JavaScript. Несмотря на улучшения (например, предварительный рендеринг), сохраняется вероятность неполной или некорректной индексации. [8]

4. Использование токенов аутентификации и сессий.

Ключевым элементом функционирования является предоставление клиенту, успешно прошедшему проверку подлинности, токена, защищенного с использованием криптографии. ОН должен быть предъявлен при каждом запросе к защищенному ресурсу. Серверная сторона системы осуществляет верификацию, проверяя валидность, срок действия и привязку к текущему сеансу. Отсутствие или обнаружение несоответствий приводит к отклонению запроса с сообщением об ошибке аутентификации. Данный подход серьезно затрудняет неавторизованный доступ. Злонамеренному субъекту необходимо не просто отправлять запросы для извлечения данных, а скрупулезно воспроизводить всю процедуру установления сессии: проходить аутентификацию, корректно обрабатывать куки или заголовки авторизации и поддерживать состояние сессии. Тем не менее, рассматриваемый метод не обеспечивает абсолютной защиты. Уязвимость заключается в потенциальной возможности его обхода путем точного копирования действий авторизованного пользователя. Современные инструменты автоматизации и специализированные библиотеки для работы с HTTP-сессиями дают возможность программам-скрапперам досконально имитировать поведение пользователя, включая получение и использование действующего токена. [6]

5. Использование уникальных заголовков HTTP

Обычные браузеры и приложения отправляют ряд основных и необязательных заголовков (User-Agent, Accept, Accept-Language, Referer), которые указывают на их тип, версию, языковые настройки и источник запроса. Если веб-сервер или межсетевой экран

настроены на проверку наличия, структуры и значений этих заголовков, это позволяет выявлять и блокировать запросы, поступающие от простых автоматизированных скриптов, которые, как правило, применяют упрощенный или нестандартный набор заголовков. Однако этот метод имеет ограниченную эффективность, так как заголовки легко подделать. Современные парсинговые инструменты, например, библиотеки requests в Python или curl, дают возможность имитировать любые заголовки, вплоть до точного воспроизведения сигнатур широко используемых браузеров. Поэтому фильтрация на основе заголовков не может служить надежной и независимой защитой. Её целесообразно использовать в качестве дополнительного уровня в многоуровневой системе безопасности. В этой роли она помогает отсеивать наиболее простые боты и затрудняет процесс анализа защитных механизмов ресурса для потенциальных злоумышленников. [7]

6. Обфускация данных

Техническая реализация данного метода включает в себя трансформацию исходной структуры и представления данных. В контексте веб-ресурсов это может выражаться в динамическом изменении HTML-кода, применении нетривиальных форматов данных для их передачи, а также в использовании приёмов клиентского шифрования или представления текстовых данных в виде растровых изображений. Подобные меры создают значительные препятствия для работы стандартных парсеров, повышая стоимость и сложность автоматизированного сбора.

Однако применение обфускации сопряжено с существенными компромиссами. Основным ограничением является негативное воздействие на пользовательский опыт и доступность. Усложнённый код может привести к снижению производительности загрузки страницы, нарушить работу вспомогательных технологий и затруднить индексацию ресурса поисковыми системами. Обфускацию следует рассматривать как специализированный инструмент, применение которого требует взвешенной оценки соотношения выигрыша в безопасности и потенциального ущерба для функциональности, производительности и возможности использования веб-сервиса. [2]

7. Регулярное обновление и мониторинг системы безопасности.

Важнейший аспект безопасности веб-сайта и любого продукта в целом. Непрерывное обновление и проактивный мониторинг формируют фундаментальный принцип обеспечения безопасности информационных систем, включая веб-ресурсы. Поддержание устойчивости к динамичному ландшафту киберугроз требует реализации системного подхода, базирующегося на двух взаимодополняющих компонентах. Первый компонент - это упреждающее управление жизненным циклом безопасности, заключающееся в своевременной установке обновлений для всех элементов программного стека и переходе на поддерживаемые версии ПО с целью устранения известных уязвимостей. Второй компонент включает регулярное проведение аудитов безопасности и тестирования на проникновение, направленных на выявление как известных, так и потенциальных уязвимостей, ошибок конфигурации и архитектурных слабостей посредством статического и динамического анализа. Интеграция данных практик создает основу для построения адаптивной системы защиты, способной не только к реактивному отражению атак, но и к прогнозированию новых векторов угроз и оперативной модификации оборонительных стратегий. Следовательно, безопасность трансформируется из периодического мероприятия в непрерывный

циклический процесс, неотъемлемо встроенный в жизненный цикл разработки и эксплуатации цифрового продукта [3].

Правовые аспекты как способ защиты

Владельцы интернет-ресурсов могут применять правовые механизмы для защиты информационных активов. Данный подход включает два ключевых аспекта: превентивный и реактивный:

Нормативный аспект: Формальное закрепление запрета на автоматизированный сбор данных без явного разрешения в публичном договоре - Пользовательском соглашении или Политике использования роботов (Robots Exclusion Protocol, реализуемой через файл robots.txt). Эти документы устанавливают правовые рамки легитимного взаимодействия с ресурсом.

Процессуальный аспект: Внедрение технических средств логирования и мониторинга, позволяющих документально фиксировать факты нарушения установленных правил. Собранные доказательства формируют основу для подачи искового заявления с требованиями о прекращении нарушения, возмещении ущерба и компенсации убытков.

Однако эффективность правовых мер ограничена рядом факторов. Юрисдикционные барьеры, в частности если нарушитель действует с территории государства, чье законодательство не предусматривает ответственности за подобные действия, или между странами отсутствуют договоры о правовой помощи, реализация судебного решения становится крайне затруднительной или невозможной. Анонимность нарушителя, через использование технологий сокрытия личности (прокси-серверы, VPN, сеть Tor) и фиктивных регистрационных данных существенно осложняет, а часто делает невыполнимой задачу идентификации ответчика для привлечения к судебной ответственности. Правовые инструменты являются необходимым, но не самодостаточным элементом комплексной защиты. Их действенность напрямую зависит от возможности идентификации субъекта нарушения и наличия действующих механизмов трансграничного правоприменения [4].

Реализация примера защиты от парсинга на примере сервера и базы данных

Пример реализации защиты от парсинга при обнаружении подозрительной активности со стороны пользователя.

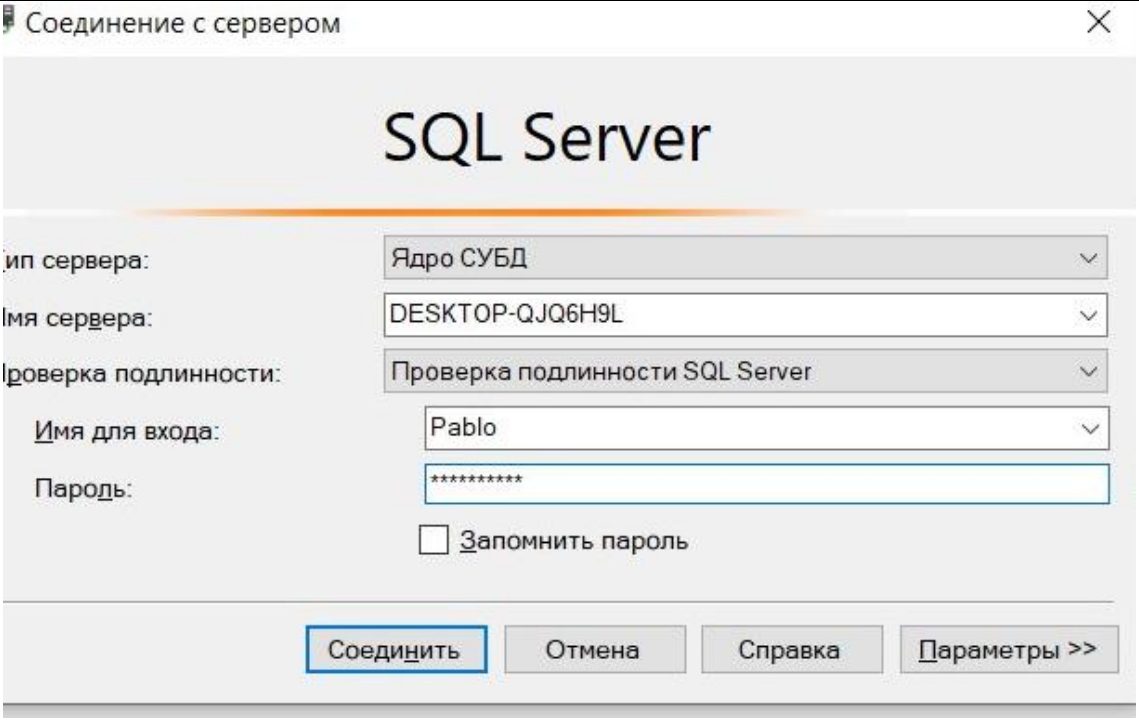

Рисунок 3 - Недоброжелательный пользователь Pablo

На Рисунке 3 представлена авторизация, пользователя (Pablo), обладающего легитимным доступом к серверу, который инициирует аномальную и деструктивную активность. Его действия характеризуются значительной нагрузкой на вычислительные ресурсы сервера, приводящей к снижению производительности и потенциальным отказам в обслуживании. А также становлением чрезмерного количества одновременных соединений, что свидетельствует о попытке автоматизированного доступа.

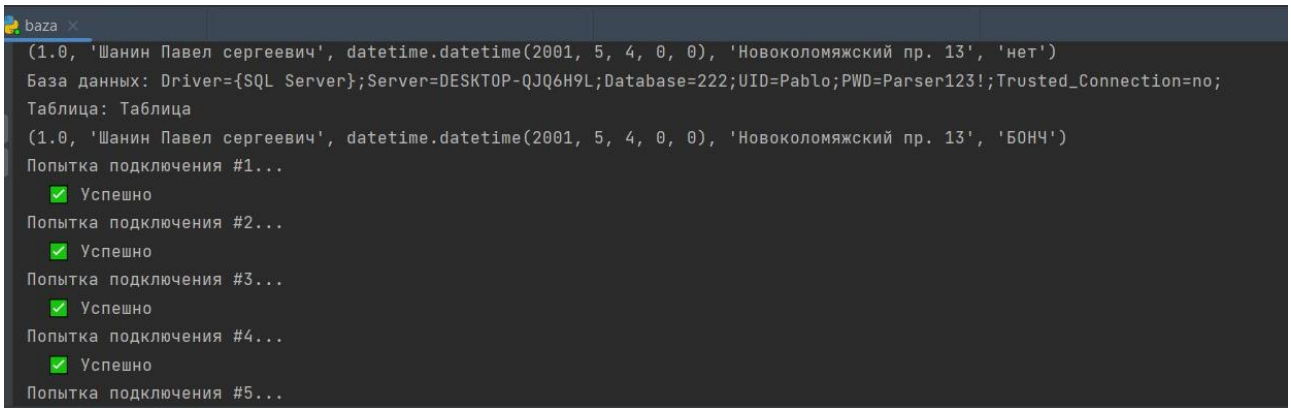

Рисунок 4 - Активность пользователя. Полученные данные

100 %

Результаты | Сообщения

	session_id	login_time	status	host_name	program_name
1	64	2025-12-06 20:33:24.447	sleeping	DESKTOP-QJQ6H9L	Python
2	60	2025-12-06 20:33:18.290	sleeping	DESKTOP-QJQ6H9L	Python
3	59	2025-12-06 20:33:10.877	sleeping	DESKTOP-QJQ6H9L	Python
4	57	2025-12-06 20:33:10.850	sleeping	DESKTOP-QJQ6H9L	Python
5	56	2025-12-06 20:33:10.820	sleeping	DESKTOP-QJQ6H9L	Python
6	55	2025-12-06 20:33:10.790	sleeping	DESKTOP-QJQ6H9L	Python
7	54	2025-12-06 20:33:10.760	sleeping	DESKTOP-QJQ6H9L	Python
8	51	2025-12-06 20:33:10.730	sleeping	DESKTOP-QJQ6H9L	Python
9	61	2025-12-06 20:32:56.900	sleeping	DESKTOP-QJQ6H9L	Microsoft SQL Server Management Studio

Рисунок 5 - Активность пользователя внутри сервера. Работа парсера.

Анализ сетевого трафика и логируемых событий позволяет идентифицировать данную активность как работу скраппера, нацеленного на несанкционированное извлечение больших объемов данных, что дополнительно усугубляет нагрузку на инфраструктуру.

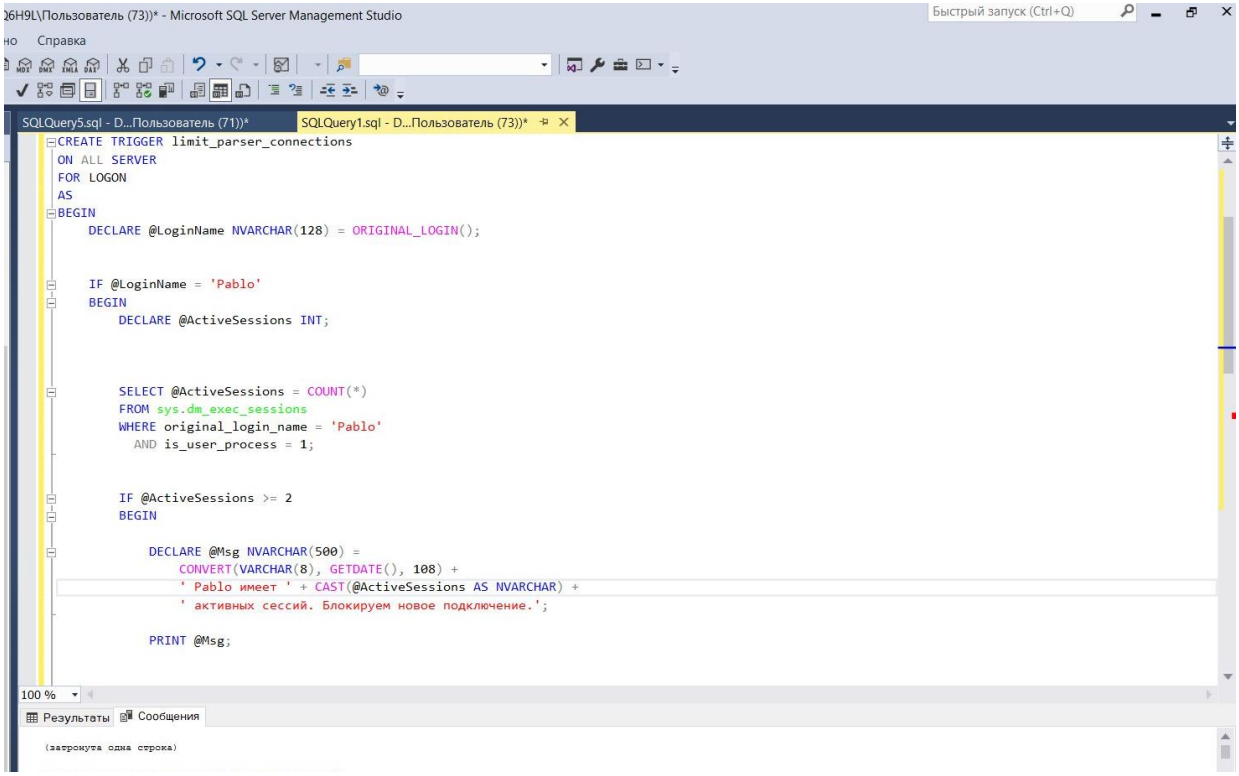

Рисунок 6 - Установленный Триггер

В результате оперативного мониторинга была проведена классификация пользователя как злоумышленника, чья деятельность представляет собой прямую угрозу целостности и доступности системы. В качестве первоочередной меры противодействия был реализован защитный триггер. Триггер, активируемый при превышении порогового значения количества соединений с одного IP-адреса за единицу времени, автоматически ограничивает последующие попытки подключения. Данная мера, эффективно блокирует работу простых парсеров, основанных на простом переборе запросов, лишая их возможности массового сбора данных.

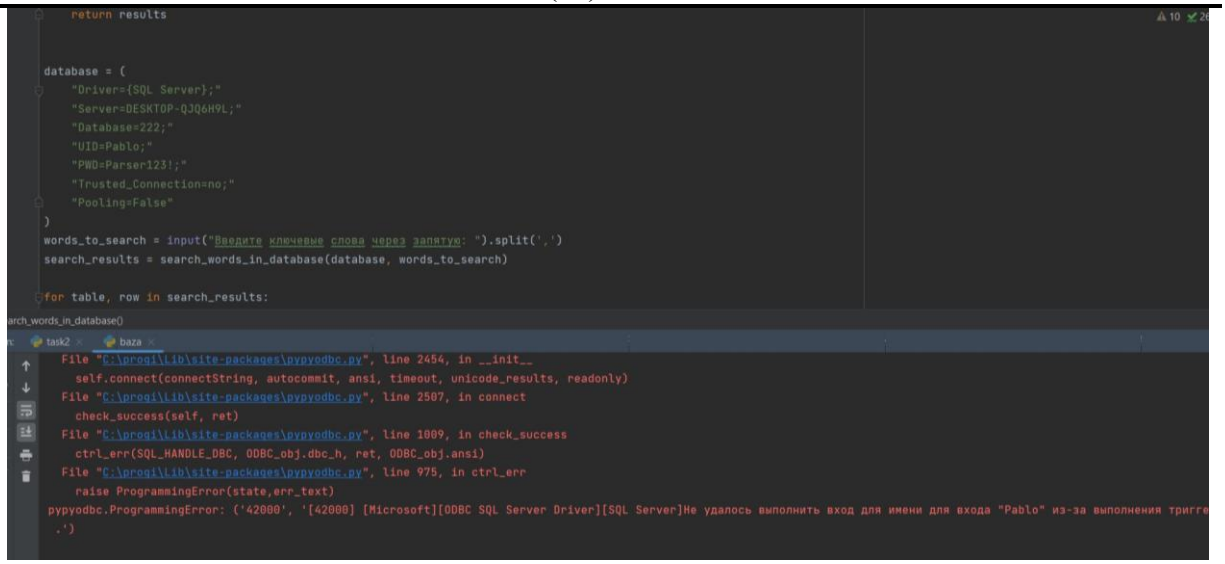

Рисунок 7 - Нейтрализация пользователя

Несмотря на успешную нейтрализацию рассмотренного инцидента, он представляет собой лишь частный случай атаки. Эффективная стратегия защиты должна основываться на комплексном многоуровневом подходе. Комплекс должен обеспечивать прозрачность для добросовестных пользователей, но сохранять устойчивость к эволюции угроз, включая возможность эскалации атаки, привлечения распределённых средств (DDoS) или применения более изощрённых методов обхода защиты. [5]

Заключение

Парсинг представляет собой многогранную угрозу, которая приводит к серьезным последствиям как для владельцев сайтов, баз данных и прочих открытых или закрытых источников информации, так и для пользователей. Утечка конфиденциальных данных, использование парсинга для конкурентного анализа, технические проблемы, распространение недостоверной информации, нарушения авторских прав и несоответствие законам о защите данных - все это примеры связанных рисков.

Важно принятие мер для защиты данных и систем, внедряя современные технологии и методы, осведомляясь о новых угрозах и тенденцииях в области кибербезопасности. Защита от парсинга требует комплексного подхода и постоянного внимания, чтобы обеспечить безопасность как для бизнеса, так и для пользователей. Как и любой аспект кибербезопасности, ни один метод защиты в той или иной мере может обеспечить только частичную защиту. Комплексный подход обеспечения безопасности и даже открытой информации имеет остро необходим в нынешней ситуации с постоянно открывающимися все новыми возможностями как для злоумышленника, так и для обычного пользователя. Неправильно настроенный парсер, приносящий технические сложности, может быть столь же опасен как и целенапралено созданный инструмент хищения данных.

Список литературы

1. Липатников, В.А. Проактивное управление информационной безопасностью автоматизированной системы радиоконтроля / В.А.Липатников, А.А. Шевченко // Информационные системы и технологии. – 2019. – №4(114). – С. 112-121.

2. Липатников, В.А. Методика проактивного управления информационной безопасностью распределенной информационной системы на основе интеллектуальных технологий / В.А. Липатников, А.А. Шевченко //Информационные системы и технологии. – 2022. – № 2(130). – С. 107-115.
3. Липатников, В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя / В.А. Липатников, А.А. Шевченко,К.В. Мелехов, В.А. Задбоев // Информационно-управляющие системы. – 2025. – №2(135). – С. 37-49.
4. Simonova S. V. Обработка данных пользователей цифровых платформ: актуальные вопросы совершенствования законодательства и практик // Вестник ЯрГУ. Серия Гуманитарные науки. – 2022. – Т. 16. – №. 4. – С. 642-649. URL: http://j.uniyar.ac.ru/index.php/vyrgu/article/view/1347
5. Байгушкина Е. А. Исследование алгоритма парсинга структур баз данных // Энигма. – 2021. – №. 29-2. – С. 132-136. URL: https://elibrary.ru/item.asp?id=44593329
6. Власенко А. В., Дзьобан П. И., Жук Р. В. Обзор инструментов машинного обучения и их применения в области кибербезопасности // Прикаспийский журнал: управление и высокие технологии. – 2020. – №. 1 (49). – С. 144-155. URL: https://cyberleninka.ru/article/n/obzor-instrumentov- mashinnogo-obucheniya-i-ih-primeneniya-v-oblasti-kiberbezopasnosti
7. Воскресенский А. С., Семёнова-Тян-Шанская В. А. СОВРЕМЕННЫЙ ХАКИНГ: ПАРСИНГ САЙТОВ // Современные технологии в теории и практике программирования. – 2020. – С. 61-62. URL: https://elibrary.ru/item.asp?id=42830131
8. Григорьев К. А. Принципы парсинга html-страниц // Сборник статей по итогам Международной научно-практической конференции 29 декабря 2017 г. – 2017. – С. 30. URL: https://ami.im/sbornik/MNPK-174-3.pdf#page=30.
9. Демина Р. Ю., Ажмухамедов И. М. Защита web-контента от нелегитимного роботизированного копирования // Вестник ГГНТУ. Технические науки. – 2022. – Т. 18. – №. 1. – С. 27. URL: https://gstou.ru/files/nauka/works_ggntu/2022/teh-1/Вестник техн 2022-1_ Демина, Ажмухамедов.pdf

References

1. Lipatnikov, V.A. Proactive Management of Information Security in Automated Radio Monitoring Systems / V.A. Lipatnikov, A.A. Shevchenko // Information Systems and Technologies. – 2019. – No. 4(114). – pp. 112-121.
2. Lipatnikov, V.A. Methodology for Proactive Management of Information Security in Distributed Information Systems Based on Intelligent Technologies / V.A. Lipatnikov, A.A. Shevchenko // Information Systems and Technologies. – 2022. – No. 2(130). – pp. 107-115.
3. Lipatnikov, V.A. Active Protection Method for Critical Information Infrastructure from Cyber Attacks Based on Interrupting the Intruder's Impact Process / V.A. Lipatnikov, A.A. Shevchenko, K.V. Melekhov, V.A. Zadboev // Information and Control Systems. – 2025. – No. 2(135). – pp. 37-49.
4. Simonova, S.V. Processing User Data on Digital Platforms: Current Issues in Improving Legislation and Practices // Bulletin of Yaroslavl State University. Humanities Series. – 2022.

– Vol. 16. – No. 4. – pp. 642-649. URL: http://j.uniyar.ac.ru/index.php/vyrgu/article/view/1347

5. Baigushkina, E.A. Research on the Algorithm for Parsing Database Structures // Enigma. – 2021. – No. 29-2. – pp. 132-136. URL: https://elibrary.ru/item.asp?id=44593329
6. Vlasenko, A.V., Dzyoban, P.I., Zhuk, R.V. Overview of Machine Learning Tools and Their Applications in Cybersecurity // Caspian Journal: Management and High Technologies. – 2020. – No. 1 (49). – pp. 144-155. URL: https://cyberleninka.ru/article/n/obzor-instrumentov-mashinnogo-obucheniya-i-ih-primeneniya-v-oblasti-kiberbezopasnosti
7. Voskresensky, A.S., Semyonova-Tyan-Shanskaya, V.A. MODERN HACKING: WEBSITE PARSING // Modern Technologies in Programming Theory and Practice. – 2020. – pp. 61-62. URL: https://elibrary.ru/item.asp?id=42830131
8. Grigoryev, K.A. Principles of Parsing HTML Pages // Collection of Articles from the International Scientific and Practical Conference, December 29, 2017. – 2017. – p. 30. URL: https://ami.im/sbornik/MNPK-174-3.pdf#page=30.
9. Demina, R.Yu., Azhmukhamedov, I.M. Protecting Web Content from Illegitimate Robotic Copying // Bulletin of GGTU. Technical Sciences. – 2022. – Vol. 18. – No. 1. – p. 27. URL: https://gstou.ru/files/nauka/works_ggntu/2022/teh-1/Вестник техн 2022-1 Демина, Ажмухамедов.pdf