ORES
Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 004.62

ВЛИЯНИЕ РАЗЛИЧНЫХ СТРАТЕГИЙ МАСКИРОВКИ ПАРСИНГА НА ВОЗМОЖНОСТЬ СБОРА ДАННЫХ

Аскеров Салех Теймур оглы,
Студент группы ИУК5-21М
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
askerovst1@student.bmstu.ru

Буракова Мария Сергеевна,
Ассистент кафедры ИУК5 КФ
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
m.burakova@bmstu.ru

Романовский Илья Олегович,
Студент группы ИУК5-21М
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
romanovskiyio@student.bmstu.ru

Аннотация

В данной статье проводится сравнительный анализ стратегий маскировки, применяемых при автоматизированном сборе данных с веб-ресурсов. Исследование фокусируется на использовании headless-режима, рандомизации заголовков User-Agent и прокси с динамической ротацией. Оценка проводится по таким критериям, как время отклика и устойчивость системы к блокировкам.

Ключевые слова: маскировка парсинга, обход блокировок, парсинг данных, автоматизация.

THE IMPACT OF VARIOUS PARSING MASKING STRATEGIES ON THE DATA COLLECTION CAPABILITY

Saleh T. Askerov,
Student of group IUK5-21M
Bauman Moscow State Technical University (Kaluga Branch)
askerovst1@student.bmstu.ru

Maria S. Burakova,
Assistant of the Department of IUK5 CF

Kaluga Branch of the Bauman Moscow State Technical University
m.burakova@bmstu.ru

Ilya O. Romanovskiy,
Student of group IUK5-21M
Bauman Moscow State Technical University (Kaluga Branch)
romanovskiyio@student.bmstu.ru

ABSTRACT

This article provides a comparative analysis of masking strategies used in automated data collection from web resources. The research focuses on the use of headless mode, randomization of User-Agent headers and proxies with dynamic rotation. The evaluation is based on criteria such as response time and system lock tolerance.

Keywords: masking parsing, bypassing locks, data parsing, automation.

Введение

В условиях бурного развития цифровых технологий и стремительного роста объема информации, размещаемой в интернете, автоматизированный сбор данных становится важной составляющей современных информационных систем. Веб-ресурсы постоянно обновляются, а их содержимое часто защищено мерами, препятствующими массовому извлечению данных. Для эффективного мониторинга и анализа информации с таких сайтов автоматизированный парсинг является необходимым инструментом, однако традиционные методы парсинга сталкиваются с трудностями из-за усиления защитных механизмов на целевых ресурсах. Особое внимание уделяется стратегиям маскировки парсинга – методам, позволяющим имитировать действия реального пользователя, обходить ограничения доступа и снижать риск блокировки. Цель данного исследования заключается в сравнительном анализе различных стратегий маскировки парсинга с целью определения оптимальных методов для повышения эффективности автоматизированного сбора данных.

Обзор технологий маскировки автоматизированных запросов

Использование ротируемых прокси

Ротируемые прокси представляют собой группу серверов, обеспечивающих динамическую смену IP-адресов при совершении запросов к целевым ресурсам. В отличие от статических прокси, где один и тот же IP-адрес используется постоянно, ротируемые прокси автоматически переключаются между несколькими адресами. Такой подход позволяет уменьшить вероятность блокировки, так как запросы выглядят как поступающие с разных источников [2].

Основные преимущества использования ротируемых прокси включают:

Динамическая смена IP: Каждый запрос или сеанс может отправляться с другого IP-адреса, что затрудняет механизмам защиты на сайте идентифицировать и заблокировать автоматизированные запросы.

Географическая гибкость: Ротируемые прокси часто предоставляют возможность выбора IP-адресов из различных регионов, что помогает обходить географические ограничения и адаптировать запросы под специфические требования целевого сайта [5].

При использовании ротируемых прокси важно учитывать, что:

Скорость ответа может варьироваться в зависимости от качества выбранных прокси-серверов и их загруженности.

Автоматизация смены IP требует правильной настройки и мониторинга, чтобы избежать чрезмерных задержек или сбоев в работе системы.

Ротация user-agent заголовков

Ротация заголовков User-Agent представляет собой метод динамической смены идентификатора браузера, который передаётся серверу при выполнении HTTP-запроса. Обычно строка User-Agent содержит сведения о типе устройства, операционной системе и версии браузера, что позволяет серверу оптимизировать выдачу контента [6]. При автоматизированном сборе данных использование одного фиксированного заголовка быстро становится потенциальным признаком автоматизации, что может привести к блокировкам или ограничению доступа со стороны целевого веб-ресурса.

Рандомизация User-Agent позволяет имитировать запросы от различных типов клиентов (десктопов, мобильных устройств, планшетов), что помогает обеспечить более естественное поведение автоматизированного парсера и снизить вероятность обнаружения его работы [4].

Метод ротации достаточно легко реализуется: можно использовать заранее сформированный список User-Agent строк и при каждом новом запросе выбирать случайное значение. Этот подход не требует значительных дополнительных вычислительных ресурсов, но существенно повышает уровень анонимности запросов. Важно следить за качеством и актуальностью выбранных строк, так как неподходящий или устаревший User-Agent может негативно сказаться на корректном отображении или получении данных.

Использование headless-браузеров

Headless-браузеры, представляющие собой специализированные версии традиционных браузеров, функционируют без графического интерфейса, выполняя все операции в фоновом режиме. Их ключевая особенность заключается в способности обрабатывать веб-контент — загружать страницы, исполнять JavaScript и извлекать данные — без визуализации элементов на экране [1]. Это делает их незаменимым инструментом для задач, где важны скорость, эффективность и минимальное потребление ресурсов.

Одним из главных преимуществ headless-режима является повышенная производительность. Отсутствие необходимости рендерить графические элементы сокращает нагрузку на процессор и оперативную память, что особенно критично при масштабных операциях, таких как массовый парсинг или параллельное тестирование веб-приложений. Благодаря этому процессы автоматизации становятся быстрее и экономичнее, а интеграция с облачными сервисами и серверными средами упрощается.

Важным аспектом является способность этих инструментов работать с динамическим контентом [3]. Они корректно исполняют JavaScript, что позволяет извлекать данные с современных веб-платформ, где информация генерируется на стороне клиента.

Сравнение стратегий маскировки парсинга

Для проведения эксперимента была выбрана одна url-ссылка сайта объявлений, количество страниц равна одной. Каждый тест состоит из 10 итераций. Всего проведено 4 теста, сравнивались 2 критерия: быстродействие и успешность сбора данных.

Использование всех вышеперечисленных стратегий

В рамках первого эксперимента система была сконфигурирована с использованием всех трёх методов маскировки. Для каждого запроса применялись динамическая смена IP посредством прокси, случайная ротация заголовков User-Agent (значение выбирается случайным образом из текстового файла) и запуск браузера в headless-режиме.

При такой конфигурации система продемонстрировала наилучшие показатели – все 10 итераций завершились успешно. Среднее время выполнения одной итерации составило 19,06 секунды. Наблюдалось, что благодаря headless-режиму страница загружалась быстрее, а использование прокси и динамически меняемого User-Agent существенно снижало риск обнаружения автоматизированных запросов, что способствовало стабильной работе парсера без блокировок.

```
Результаты теста:
Количество итераций: 10
Успешных итераций: 10
Среднее время выполнения одной итерации: 19.06 сек
Процент успешных итераций: 100.0%
```

Рисунок 1 – Скриншот выполнения 1 теста [разработано авторами]

Использование User-Agent заголовков и headless-браузера

Во втором эксперименте система была настроена так, что для каждого запроса применялась ротация заголовков User-Agent и использовался headless-режим, но не применялись прокси-серверы. Это означало, что все запросы отправлялись с одного и того же исходного IP-адреса, что потенциально могло увеличить вероятность блокировок со стороны целевого ресурса.

В ходе проведения теста результаты оказались менее успешными по сравнению с первым экспериментом. Из 10 итераций только 7 завершились успешно, остальная часть запросов столкнулась с блокировками. Среднее время выполнения одной итерации составило 19,50 секунд. Более низкий процент успешных итераций свидетельствует о том, что отсутствие смены IP оказывает существенное влияние на эффективность обхода защитных механизмов, несмотря на использование динамической смены User-Agent и headless-режима.

```
Результаты теста:
Количество итераций: 10
Успешных итераций: 7
Среднее время выполнения одной итерации: 19.50 сек
Процент успешных итераций: 70.0%
```

Рисунок 2 – Скриншот выполнения 2 теста [разработано авторами]

Использование только headless-браузера

В рамках третьего эксперимента был протестирован наименее защищённый сценарий автоматизированного парсинга. Для всех 10 итераций использовались одни и те же параметры: отсутствие прокси-серверов и неизменный заголовок User-Agent. Парсинг выполнялся в headless-режиме, что позволяло ускорить процесс за счёт отсутствия графического интерфейса браузера.

Сценарий подразумевал, что каждый запрос выглядел идентично предыдущему — с одинаковым User-Agent и IP-адресом. Это резко повышало вероятность того, что сайт воспримет активность как подозрительную. Несмотря на это, парсеру удалось завершить 5 итераций из 10 успешно. Остальные 5 итераций завершились блокировкой по IP. Среднее время выполнения успешных итераций составило 19,33 секунды.

Результаты данного теста наглядно показывают, что при отсутствии как ротации User-Agent, так и прокси, устойчивость автоматизированной системы существенно снижается. Эффективность ограничивается 50% успешных попыток, и без дополнительных мер маскировки такой подход может быть применим лишь для кратковременного или тестового парсинга.

```
Результаты теста:
Количество итераций: 10
Успешных итераций: 5
Среднее время выполнения одной итерации: 19.33 сек
Процент успешных итераций: 50.0%
```

Рисунок 3 – Скриншот выполнения 3 теста [разработано авторами]

Использование headed-браузера без маскировки

Четвёртый тест был проведён в режиме с графическим интерфейсом (headed-режим), где браузер функционировал как обычное приложение с полной визуализацией страниц, но без применения каких-либо инструментов маскировки.

Из 10 выполненных итераций успешными оказались лишь 6, в то время как остальные 4 завершились сбоями. Среднее время выполнения запроса составило 25.11 секунды.

Отсутствие ротации IP и однотипные параметры User-Agent значительно повысили вероятность блокировки. Даже несмотря на естественный вид взаимодействия через графический интерфейс, сайт быстро распознал автоматизированный характер действий.

Результаты теста демонстрируют, что применение headed-режима без дополнительных мер маскировки не обеспечивает устойчивость парсинга при регулярном или массовом сборе данных. Такой подход может быть оправдан лишь на этапе отладки скриптов или для единичных ручных операций, где критичен визуальный контроль. Для промышленного использования требуется интеграция с инструментами, скрывающими цифровые следы и имитирующими человеческое поведение.

```
Результаты теста:
Количество итераций: 10
Успешных итераций: 6
Среднее время выполнения одной итерации: 25.11 сек
Процент успешных итераций: 60.0%
```

Рисунок 4 – Скриншот выполнения 4 теста [разработано авторами]

Результаты сравнения

В Таблице 1 представлены результаты тестов.

Таблица 1 – Сравнительные результаты тестов

№ теста	Среднее время выполнения, с	Успешность, %
1	19.06	100
2	19.50	70
3	19.33	50
4	25.11	60

Заключение

В ходе проведённого исследования были протестированы различные методы маскировки автоматизированных запросов при парсинге данных с сайта объявлений. Эксперименты показали, что для полноценной и стабильной работы парсера требуется комплексное применение всех перечисленных техник маскировки — использование headless-браузера, ротируемых прокси и динамической ротации заголовков User-Agent.

Отказ от одного или нескольких методов значительно снижает процент успешных итераций: чем проще конфигурация, тем выше вероятность блокировок и ошибок в процессе выполнения парсинга.

Также было выявлено, что время выполнения итераций зависит только от режима запуска браузера. Headless-режим обеспечивает меньшую нагрузку на систему и быстрее обрабатывает страницы по сравнению с полноэкранным режимом. Прочие методы маскировки не влияют на продолжительность одной итерации, но значительно повышают устойчивость к защите со стороны сайта.

Таким образом, при разработке надёжного парсера для защищённых ресурсов рекомендуется использовать все доступные методы маскировки в совокупности — это позволяет добиться максимальной эффективности, стабильности и минимального риска блокировок.

Список литературы:

1. Barth A., Jackson C., Mitchell J. C. Robust Defenses for Cross-Site Request Forgery // Proceedings of the 15th ACM Conference on Computer and Communications Security. – 2008. – Pp. 75–88. – DOI 10.1145/1455770.1455782.
2. Nikiforakis N., Kapravelos A., Joosen W. et al. Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting // Proceedings of the IEEE Symposium on Security and Privacy. – 2013. – Pp. 541–555. – DOI 10.1109/SP.2013.43.
3. Englehardt S., Narayanan A. Online Tracking: A 1-million-site Measurement and Analysis // Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. – 2016. – Pp. 1388–1401. – DOI 10.1145/2976749.2978313.
4. Ferrara E., Varol O., Davis C. et al. The Rise of Social Bots // Communications of the ACM. – 2016. – Vol. 59, No. 7. – Pp. 96–104. – DOI 10.1145/2818717.
5. Acar G., Eubank C., Englehardt S. et al. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild // Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. – 2014. – Pp. 674–689. – DOI 10.1145/2660267.2660347.
6. Olston C., Najork M. Web Crawling // Foundations and Trends in Information Retrieval. – 2010. – Vol. 4, No. 3. – Pp. 175–246. – DOI 10.1561/1500000017.

References:

1. Barth A., Jackson C., Mitchell J. C. Robust Defenses for Cross-Site Request Forgery // Proceedings of the 15th ACM Conference on Computer and Communications Security. – 2008. – Pp. 75–88. – DOI 10.1145/1455770.1455782.
2. Nikiforakis N., Kapravelos A., Joosen W. et al. Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting // Proceedings of the IEEE Symposium on Security and Privacy. – 2013. – Pp. 541–555. – DOI 10.1109/SP.2013.43.
3. Englehardt S., Narayanan A. Online Tracking: A 1-million-site Measurement and Analysis // Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. – 2016. – Pp. 1388–1401. – DOI 10.1145/2976749.2978313.
4. Ferrara E., Varol O., Davis C. et al. The Rise of Social Bots // Communications of the ACM. – 2016. – Vol. 59, No. 7. – Pp. 96–104. – DOI 10.1145/2818717.

5. Acar G., Eubank C., Englehardt S. et al. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild // Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. – 2014. – Pp. 674–689. – DOI 10.1145/2660267.2660347.
6. Olston C., Najork M. Web Crawling // Foundations and Trends in Information Retrieval. – 2010. – Vol. 4, No. 3. – Pp. 175–246. – DOI 10.1561/1500000017.