Как парсить сайты с защитой - современные технологии и методы обхода блокировок в 2025 году

Краткое содержание

Процесс извлечения данных из интернета в 2025 году стал значительно труднее из-за внедрения интеллектуальных систем защиты, таких как Cloudflare, DataDome и Akamai. Эти системы анализируют не только количество запросов, но и цифровые отпечатки браузера, сетевые параметры и даже манеру движения мыши. В данном материале подробно разбираются методы маскировки под реального пользователя, использование специализированных библиотек вроде curl-impersonate и Camoufox, а также стратегии управления прокси-серверами. Вы узнаете, как работают современные алгоритмы оценки риска, почему стандартные инструменты автоматизации больше не справляются и какие технические решения позволяют стабильно собирать информацию в условиях жесткой защиты. Информация будет полезна как опытным разработчикам, так и тем, кто только начинает разбираться в технической части сбора данных.

Современные механизмы обнаружения автоматизированного трафика

Процесс автоматического сбора информации претерпел серьезные изменения. Если десять лет назад для блокировки хватало простого отслеживания количества запросов с одного адреса, то сегодня защита строится на комплексном анализе десятков параметров.[1] Компании, занимающиеся защитой от ботов, используют алгоритмы машинного обучения для формирования оценки доверия к каждому посетителю. Этот показатель определяет, получит ли пользователь доступ к контенту или столкнется с проверкой.

Системы защиты теперь работают на разных уровнях. На сетевом уровне они проверяют параметры шифрования. На уровне браузера они выполняют невидимые скрипты для сбора характеристик устройства. На уровне поведения они следят за тем, как быстро человек перемещается по страницам и как он взаимодействует с интерфейсом.[3] Такая многослойная преграда делает обычные скрипты на базе стандартных библиотек бесполезными.

Уровень проверки	Что анализируется	Инструменты защиты

Сетевой	Отпечатки TLS (JA3/JA4), параметры HTTP/2, репутация IP [5]	Cloudflare, Akamai
Браузерный	Canvas, WebGL, аудио-отпечатки, наличие WebDriver [7]	DataDome, PerimeterX
Поведенческий	Движения мыши, скорость набора, паттерны переходов [9]	HUMAN Security, Kasada
Аппаратный	Модель GPU, количество ядер процессора, состояние батареи [11]	Все современные системы

Статистика показывает, что трафик ботов, созданных с помощью искусственного интеллекта, в 2025 году вырос в четыре раза.[13] Это заставляет владельцев сайтов внедрять еще более жесткие методы фильтрации. Только около 2.8% крупных ресурсов остаются полностью незащищенными от автоматизированного сбора.[13]

Сетевой уровень и технологии имитации отпечатков tls

Одной из самых частых причин мгновенного бана является несоответствие параметров сетевого соединения. Когда программа пытается подключиться к серверу по протоколу HTTPS, происходит обмен данными для установки защищенного канала. Браузеры Chrome, Firefox и Safari делают это по-разному, используя разные наборы алгоритмов шифрования и расширений.[5]

Системы защиты используют технологию JA3 для создания цифрового следа этого процесса. Если программа представляется браузером Chrome через заголовок User-Agent, но ее сетевой отпечаток соответствует стандартной библиотеке Python, сервер поймет, что перед ним бот.[6] В 2025 году на смену JA3 пришел стандарт JA4, который более устойчив к случайным изменениям в поведении браузеров и учитывает параметры протокола HTTP/2.[6]

Использование библиотеки curl_cffi для имитации браузера

Для решения преград на сетевом уровне была создана библиотека curl_cffi. Она представляет собой обертку над модифицированной версией cURL, которая умеет полностью копировать поведение реальных браузеров при установке соединения.[15] В

отличие от обычных библиотек, она позволяет передать параметр имитации конкретной версии браузера.

Python

```
from curl_cffi import requests

# Пример выполнения запроса с имитацией последней версии Chrome
response = requests.get(
    "https://www.walmart.com",
    impersonate="chrome124"
)

# Вывод статуса и части контента
print(f"Статус ответа: {response.status_code}")
print(f"Заголовок страницы: {response.text[:100]}")
```

Этот подход значительно быстрее использования полноценных браузеров, так как он не требует отрисовки графики и выполнения всего кода JavaScript на странице.[16] Это оптимальный вариант для сайтов, которые блокируют запросы на этапе проверки сетевых отпечатков, но не требуют сложного взаимодействия с интерфейсом.

Особенности стандарта ja4 и его важность

Стандарт JA4 разделяет отпечаток на несколько частей. Часть A описывает протокол и основные метаданные, часть B содержит хеш алгоритмов шифрования, а часть C - расширения и алгоритмы подписи.[6] Современные системы защиты, такие как Cloudflare, используют эти данные для мгновенной фильтрации трафика от простых скриптов.

Параметр JA4	Что означает	Почему важно
t13d1516h2	Протокол TCP, TLS 1.3, наличие SNI, 15 шифров, 16 расширений, HTTP/2	Позволяет отличить Chrome от автоматизированного запроса cURL [6]
8daaf6152771	Хеш отсортированного списка алгоритмов	Препятствует обходу через простую подмену

	шифрования	порядка данных
e5627efa2ab1	Хеш расширений и алгоритмов подписи	Выявляет использование библиотек, которые не поддерживают современные расширения

Для успешного сбора данных необходимо следить за тем, чтобы выбранный профиль имитации соответствовал заголовку User-Agent.[17] Если вы заявляете, что используете Windows, но сетевые параметры указывают на Linux, защита посчитает это признаком бота.

Браузерная автоматизация и скрытие признаков управления

Когда сайт активно использует JavaScript для проверки окружения, простыми HTTP-запросами обойтись не получится. В таких случаях применяются инструменты автоматизации, такие как Playwright или Puppeteer. Однако стандартные версии этих программ содержат множество признаков, которые легко обнаруживаются системами защиты.[8]

Проблема переменной navigator.webdriver

Самым простым признаком автоматизации является флаг navigator.webdriver. В обычном браузере он отсутствует или равен значению false, но при управлении через специальные протоколы он автоматически принимает значение true.[3] Сайты проверяют этот флаг с помощью простого скрипта: if (navigator.webdriver) { block_bot(); }.

Для скрытия этого признака в Playwright можно использовать внедрение скрипта при инициализации страницы:

Python

```
from playwright.sync_api import sync_playwright

with sync_playwright() as p:
    browser = p.chromium.launch(headless=False) # Headed режим менее подозрителен
    context = browser.new_context()
```

```
    # Удаляем признак автоматизации перед загрузкой страницы
    context.add_init_script("""
        Object.defineProperty(navigator, 'webdriver', {
            get: () => undefined
        })
    """)

    page = context.new_page()
    page.goto("https://www.browserscan.net/bot-detection")
    # Далее следует логика сбора данных [18]
```

Технология camoufox для максимальной скрытности

Camoufox представляет собой современное решение в сфере автоматизации. Это не просто библиотека, а полноценная модифицированная сборка браузера Firefox, в которой защита от обнаружения внедрена на уровне исходного кода C++.[19] Это делает невозможным обнаружение бота через стандартные проверки JavaScript.

Основные технические решения Camoufox:

- Изоляция кода управления. Все команды автоматизации выполняются в отдельной среде, к которой у скриптов сайта нет доступа.[20]
- Подмена отпечатков на низком уровне. Браузер имитирует параметры видеокарты (WebGL) и отрисовки (Canvas) без внесения искусственных помех, которые могут быть замечены.[19]
- Защита от утечек через WebRTC. Система блокирует возможность узнать реальный IP-адрес пользователя, даже если он использует прокси.[20]
- Использование Juggler вместо CDP. Для управления Firefox используется протокол Juggler, который сложнее обнаружить, чем стандартный протокол управления Chrome.[20]

Исследование аппаратных и системных отпечатков

Современные системы защиты собирают данные о "железе" вашего устройства. Это позволяет им создать уникальный идентификатор, который сохраняется даже при смене IP или очистке куки. Этот процесс называется фингерпринтингом.[7]

Анализ графической подсистемы

Отрисовка графики уникальна для каждой комбинации видеокарты и драйвера. Скрипты защиты просят браузер нарисовать скрытую фигуру или текст на холсте (Canvas) или через WebGL. Полученный результат превращается в короткую строку - хеш. У ботов, работающих в облачных сервисах, этот хеш часто указывает на использование

программного отрисовщика SwiftShader, что является явным признаком автоматизации.[7]

Для защиты от этого метода используются два подхода:

1. Добавление небольшого "шума" в результат отрисовки. Это меняет хеш, но при частом использовании может выглядеть подозрительно.
2. Использование реальных профилей оборудования. Программы вроде Camoufox подставляют параметры реальных видеокарт, чтобы результат отрисовки выглядел естественно.[19]

Другие важные параметры системы

Помимо графики, защитные скрипты анализируют множество других данных:

- Шрифты. Список установленных в системе шрифтов может многое сказать о пользователе. Боты часто имеют ограниченный набор стандартных шрифтов.[11]
- Аудио. Проверка того, как звуковая карта обрабатывает сигнал, также позволяет создать уникальный след.[12]
- Состояние батареи. У обычных ноутбуков уровень заряда постоянно меняется. Если значение всегда равно 100% или данные отсутствуют, это вызывает подозрение.[7]
- Часовой пояс и язык. Эти параметры должны совпадать с местоположением вашего IP-адреса.[19]

Ниже приведена таблица сравнения популярных анти-детект решений для автоматизации.

Инструмент	**Базовый движок**	**Метод маскировки**	**Плюсы**
Camoufox	Firefox (Custom)	Патчи в ядре C++ [20]	Невозможно обнаружить через JS, высокая скорость
Playwright Stealth	Chromium	Инъекция JS-скриптов [22]	Легко настроить, работает со стандартным Playwright
Undetected ChromeDriver	Chrome	Модификация бинарного файла [21]	Хорошо обходит базовые проверки

			на базе Chrome
SeleniumBase	Chrome	Комплексные патчи и CDP [23]	Готовое решение "из коробки" для сложных задач

Поведенческий анализ и имитация действий человека

Даже если ваш браузер выглядит как настоящий, ваше поведение может выдать программу. Боты обычно перемещаются по сайту по кратчайшему пути, кликают в центр кнопок и делают это мгновенно после появления элемента.[3] Люди же ведут себя хаотично: они долго думают, двигают мышью по кривым траекториям и иногда ошибаются.

Алгоритмы движения мыши

Библиотеки вроде HumanCursor используют сложные математические модели для генерации траекторий. Вместо того чтобы переместить курсор из точки А в точку Б по прямой, они создают изогнутую линию с фазами ускорения и замедления.[9] Это имитирует физическую работу руки человека.

Python

```
from humancursor import WebCursor

# Использование HumanCursor вместе с Selenium
cursor = WebCursor(driver)
element = driver.find_element(By.ID, "submit-button")

# Курсор переместится к элементу по естественной траектории
cursor.click_on(element)
```

В Playwright для этих же целей используется библиотека ghost-cursor. Она позволяет не только двигать мышь, но и имитировать наведение (hover) перед кликом, что является важным сигналом для систем вроде PerimeterX.[10]

Паттерны навигации и задержки

Важно соблюдать временные интервалы. Если программа открывает 10 страниц товаров за одну секунду, она будет заблокирована. Необходимо внедрять случайные паузы между действиями. Также полезно "прогревать" сессию: перед тем как перейти к нужному товару, стоит зайти на главную страницу, прокрутить ее, имитируя чтение, и только потом совершать целевое действие.[21]

Управление прокси-серверами и репутация адресов

IP-адрес является первым рубежом защиты. Если ваш адрес принадлежит крупному дата-центру (например, AWS или Google Cloud), доверие к нему будет минимальным.[21] Большинство серьезных сайтов сразу выдают проверку капчей таким пользователям.

Типы прокси и их эффективность в 2025 году

Для качественного сбора данных используются три основных вида прокси:

1. Резидентные прокси. Это адреса обычных домашних пользователей. Они имеют высокий уровень доверия, так как за ними обычно стоят реальные люди.[26]
2. Мобильные прокси. Самый надежный вариант. Благодаря технологии CGNAT, тысячи людей могут использовать один и тот же мобильный IP. Блокировка такого адреса может лишить доступа множество реальных клиентов мобильного оператора, поэтому сайты относятся к ним очень лояльно.[25]
3. Серверные прокси. Дешевые и быстрые, но легко обнаруживаются. Подходят только для простых сайтов без продвинутой защиты.

Тип	**Уровень доверия**	**Стоимость**	**Основная сфера применения**
Резидентные	Высокий	Средняя	Массовый сбор данных, обход Cloudflare [26]
Мобильные	Очень высокий	Высокая	Работа с соцсетями, обход DataDome и Akamai [25]
Дата-центры	Низкий	Низкая	Тестирование, сбор данных с простых ресурсов

			[29]

Стратегия использования прокси

Эффективный подход заключается в гибридной маршрутизации. Для простых страниц можно использовать дешевые серверные адреса, а при обнаружении блокировки автоматически переключаться на резидентные или мобильные.[25] Важно сохранять "липкие" (sticky) сессии: в рамках одного сеанса работы с сайтом IP-адрес не должен меняться, иначе это вызовет подозрение у системы безопасности.[17]

Автоматизация решения капчи

Если защита все же выдала капчу, ее нужно решить автоматически. В 2025 году это чаще всего Cloudflare Turnstile или reCAPTCHA v2/v3. Эти задачи решаются либо через специализированные API-сервисы, либо с помощью локальных моделей машинного обучения.

Использование сервисов решения капчи

Сервисы вроде CapSolver или 2Captcha позволяют получить токен решения, который затем подставляется в форму на сайте. Процесс выглядит так:

- Скрипт находит ключ сайта (sitekey) в исходном коде страницы.
- Параметры отправляются на сервер сервиса.
- Через некоторое время сервис возвращает длинную строку - токен.
- Скрипт вставляет этот токен в скрытое поле на странице и отправляет форму.[30]

Локальное решение через модели yolo

Для текстовых или графических капч (например, где нужно выбрать изображения с определенным объектом) можно использовать нейронные сети. Модели семейства YOLO (You Only Look Once) отлично справляются с обнаружением объектов на картинках в режиме реального времени.[32] Это позволяет значительно сэкономить на услугах сторонних сервисов при больших объемах работы.

Python

```
# Концептуальный пример использования модели для поиска объектов на капче
import torch
from PIL import Image
```

```
# Загрузка предобученной модели
model = torch.hub.load('ultralytics/yolov5', 'custom', path='captcha_model.pt')

# Распознавание объектов
img = Image.open('captcha.png')
results = model(img)

# Получение координат объектов для имитации клика
print(results.xyxy)
```

Исследования показывают, что модели версии nano (YOLOv8n или YOLOv10n) обеспечивают наилучшую скорость работы, что критично для автоматизации.[33]

Реальный пример: обход системы datadome на сайте ритейлера

Рассмотрим процесс сбора данных о ценах на кроссовки с сайта, защищенного DataDome. Это одна из самых агрессивных систем, которая блокирует доступ при малейшем подозрении.

Шаг 1. Инициализация окружения. Используется Camoufox через Playwright. Выбирается профиль реального пользователя Windows с актуальной версией Chrome. Настраивается подключение через мобильный прокси Великобритании.[10]

Шаг 2. Обход первичной проверки. При первом переходе DataDome собирает отпечатки. Благодаря Camoufox, все проверки Canvas и WebGL проходят успешно, так как браузер выдает параметры реального графического процессора.[20]

Шаг 3. Имитация поведения. Для перехода в раздел "Мужская обувь" используется библиотека ghost-cursor. Мышь плавно перемещается к меню, задерживается на нем (имитируя hover), и только потом происходит клик. Это позволяет избежать блокировки на основе поведенческих алгоритмов.[10]

Шаг 4. Сбор и ротация. После извлечения данных с пяти страниц скрипт закрывает браузер, меняет IP на прокси и создает новый профиль пользователя с другими характеристиками экрана и шрифтов. Это предотвращает связывание различных сессий в одну цепочку.[10]

Резюме и лучшие практики

Для построения стабильной системы сбора данных необходимо придерживаться следующих правил:

- Всегда обеспечивайте соответствие между всеми уровнями отпечатков. User-Agent

должен совпадать с версией TLS и аппаратными характеристиками.[17]
- Используйте инструменты, которые вносят изменения на уровне ядра браузера (как Camoufox), а не просто подменяют переменные через JavaScript.[20]
- Отдавайте предпочтение резидентным и мобильным прокси. Это значительно снижает риск появления капчи.[25]
- Внедряйте механизмы имитации человеческого поведения: случайные задержки, плавные движения мыши и реалистичные паттерны прокрутки страниц.[9]
- Регулярно тестируйте свои скрипты на детекторах вроде CreepJS, чтобы вовремя заметить новые методы обнаружения ботов.[34]

Мини-faq по техническим вопросам

1. В чем разница между JA3 и JA4? JA3 - это старый стандарт, который чувствителен к порядку данных в сетевом пакете. Современные браузеры специально меняют этот порядок, чтобы защититься от слежки, что ломает JA3. JA4 решает эту проблему, сортируя данные перед хешированием и добавляя информацию о протоколе HTTP/2.[6]

2. Можно ли использовать headless режим в 2025 году? Стандартный headless режим в Chrome легко обнаруживается по специфическим параметрам отрисовки и отсутствию некоторых API. Для успешной работы лучше использовать "новый" headless режим или специальные патчи вроде тех, что есть в Camoufox и Undetected ChromeDriver.[20]

3. Помогают ли бесплатные прокси?

Почти никогда. Бесплатные прокси быстро попадают в списки блокировок всех систем защиты. Их использование приведет к тому, что вы будете получать ошибку 403 или капчу на каждом запросе.

4. Зачем нужна библиотека curl_cffi, если есть Playwright? Playwright потребляет много ресурсов процессора и памяти, так как запускает целый браузер. Если сайт защищен только на сетевом уровне (TLS), то curl_cffi позволит собирать данные в десятки раз быстрее и дешевле.[16]

5. Как система понимает, что я использую Selenium? Selenium добавляет во внутренние структуры браузера специфические переменные и свойства (например, $cdc_... в Chrome). Системы защиты ищут эти следы. Чтобы этого избежать, нужно использовать модифицированные драйверы, такие как Undetected ChromeDriver.[21]

6. Что такое WebRTC утечка? Это ситуация, когда через специальный протокол для видеосвязи сайт может узнать ваш реальный IP-адрес, даже если вы используете прокси. Чтобы этого не произошло, WebRTC нужно отключать в настройках браузера.[12]

7. Как часто нужно менять цифровой отпечаток? Рекомендуется создавать новый

профиль (отпечаток) для каждой новой сессии или после выполнения определенного объема задач. Это не позволяет системе защиты накопить достаточно данных для того, чтобы пометить ваш профиль как подозрительный.[21]

Источники

1. Web Scraping Challenges & Compliance in 2025 | Market Insights - GroupBWT, дата последнего обращения: февраля 17, 2026, https://groupbwt.com/blog/challenges-in-web-scraping/
2. How to Bypass Akamai when Web Scraping in 2026 - Scrapfly, дата последнего обращения: февраля 17, 2026, https://scrapfly.io/blog/posts/how-to-bypass-akamai-anti-scraping
3. Modern Anti-Bot Systems and How to Bypass Them | by Harim Choi, дата последнего обращения: февраля 17, 2026, https://python.plainenglish.io/modern-anti-bot-systems-and-how-to-bypass-them-4d28475522d1
4. The Ultimate Guide to Web Scraping Antibot Systems (2025) - WebAutomation, дата последнего обращения: февраля 17, 2026, https://webautomation.io/blog/ultimate-guide-to-web-scraping-antibot-and-blocking-systems-and-how-to-bypass-them/
5. TLS Fingerprinting: How It Works & How to Bypass It (2025) - Browserless, дата последнего обращения: февраля 17, 2026, https://www.browserless.io/blog/tls-fingerprinting-explanation-detection-and-bypassing-it-in-playwright-and-puppeteer
6. JA3/JA4 TLS Fingerprint - Detect Browser TLS/SSL Fingerprinting - Scrapfly, дата последнего обращения: февраля 17, 2026, https://scrapfly.io/web-scraping-tools/ja3-fingerprint
7. BrowserLeaks: Browser Fingerprint & Privacy Testing Tool, дата последнего обращения: февраля 17, 2026, https://datadome.co/anti-detect-tools/browserleaks/
8. How to Bypass DataDome: Complete Guide 2026 - ZenRows, дата последнего обращения: февраля 17, 2026, https://www.zenrows.com/blog/datadome-bypass
9. How to Mimic Real Human Interactions with HumanCursor During Scraping - ZenRows, дата последнего обращения: февраля 17, 2026, https://www.zenrows.com/blog/humancursor
10. Guide to Bypassing DataDome in 2025 - Kameleo, дата последнего обращения: февраля 17, 2026, https://kameleo.io/blog/guide-to-bypassing-datadome
11. Browserleaks - Check your browser for privacy leaks, дата последнего обращения: февраля 17, 2026, https://browserleaks.com/
12. Building Ethical Anti-Detect Browsers: Techniques & Insights - BrowserCat, дата последнего обращения: февраля 17, 2026, https://www.browsercat.com/post/ethical-anti-detect-browser-techniques
13. DataDome's 2025 Global Bot Security Report Exposes the AI Traffic Crisis, дата последнего обращения: февраля 17, 2026, https://datadome.co/press/datadomes-2025-global-bot-security-report-exposes

-the-ai-traffic-crisis/
14. What is TLS Fingerprint and How to Bypass it in 2025 - Roundproxies, дата последнего обращения: февраля 17, 2026, https://roundproxies.com/blog/what-is-tls-fingerprint/
15. Web Scraping With curl_cffi and Python in 2026 - Bright Data, дата последнего обращения: февраля 17, 2026, https://brightdata.com/blog/web-data/web-scraping-with-curl-cffi
16. How to Bypass Datadome in 2026: 6 working methods, дата последнего обращения: февраля 17, 2026, https://roundproxies.com/blog/bypass-datadome/
17. How to Bypass Cloudflare Challenge While Web Scraping in 2026, дата последнего обращения: февраля 17, 2026, https://www.capsolver.com/blog/Cloudflare/bypass-cloudflare-challenge-2025
18. Avoid Bot Detection With Playwright Stealth: 9 Solutions for 2025, дата последнего обращения: февраля 17, 2026, https://www.scrapeless.com/en/blog/avoid-bot-detection-with-playwright-stealth
19. How to Scrape With Camoufox to Bypass Antibot Technology - ScrapingBee, дата последнего обращения: февраля 17, 2026, https://www.scrapingbee.com/blog/how-to-scrape-with-camoufox-to-bypass-antibot-technology/
20. Stealth Overview | Camoufox, дата последнего обращения: февраля 17, 2026, https://camoufox.com/stealth/
21. How to Bypass Datadome Anti Scraping in 2026 - Scrapfly, дата последнего обращения: февраля 17, 2026, https://scrapfly.io/blog/posts/how-to-bypass-datadome-anti-scraping
22. Bypass Cloudflare with Playwright BQL 2025 Guide - Browserless, дата последнего обращения: февраля 17, 2026, https://www.browserless.io/blog/bypass-cloudflare-with-playwright
23. Stealthy Playwright Mode: Bypass CAPTCHAs and Bot-Detection! - SeleniumBase, дата последнего обращения: февраля 17, 2026, https://seleniumbase.com/stealthy-playwright-mode-bypass-captchas-and-bot-detection/
24. riflosnake/HumanCursor: Simulate Human Cursor Movement for Automated Scripts - GitHub, дата последнего обращения: февраля 17, 2026, https://github.com/riflosnake/HumanCursor
25. What is a Mobile Proxy? Compared With Residential & Datacenter - PromptCloud, дата последнего обращения: февраля 17, 2026, https://www.promptcloud.com/blog/mobile-proxy-vs-datacenter-for-scraping/
26. Mobile vs. Residential Proxies: 4 Key Differences - anyIP, дата последнего обращения: февраля 17, 2026, https://anyip.io/blog/mobile-proxies-vs-residential-proxies
27. Safe Web Scraping in 2025: Proxies, User-Agent, Mobile vs Residential for Marketing, дата последнего обращения: февраля 17, 2026, https://mobileproxy.space/en/pages/safe-web-scraping-in-2025-proxies-user-agent-mobile-vs-residential-for-marketing.html

28. Mobile Proxies vs Residential Proxies | Which Is Best? - Proxyrack, дата последнего обращения: февраля 17, 2026, https://www.proxyrack.com/blog/mobile-proxies-vs-residential-proxies/
29. Mobile vs Residential Proxy vs Datacenter Proxies - NodeMaven, дата последнего обращения: февраля 17, 2026, https://nodemaven.com/blog/mobile-vs-residential-vs-datacenter-proxies-whats-the-difference/
30. How to Bypass reCAPTCHA and Turnstile in Crawlee with CapSolver - DEV Community, дата последнего обращения: февраля 17, 2026, https://dev.to/luisgustvo/how-to-bypass-recaptcha-and-turnstile-in-crawlee-with-capsolver-307p
31. How to Solve Cloudflare Turnstile Captcha Automatically with CaptchaAI in 2025, дата последнего обращения: февраля 17, 2026, https://captchaai.com/blog-item/how-to-solve-cloudflare-turnstile-captcha-automatically-with-captchaai-in-2025-best-cloudflare-turnstile-solver
32. Breaking a CAPTCHA system with Machine Learning - GeeksforGeeks, дата последнего обращения: февраля 17, 2026, https://www.geeksforgeeks.org/machine-learning/breaking-a-captcha-system-with-machine-learning/
33. Benchmarking of Different YOLO Models for CAPTCHAs Detection and Classification - arXiv, дата последнего обращения: февраля 17, 2026, https://arxiv.org/html/2502.13740v1
34. abrahamjuliot/creepjs: Creepy device and browser ... - GitHub, дата последнего обращения: февраля 17, 2026, https://github.com/abrahamjuliot/creepjs
35. WebRTC Is the Silent IP Leak Living in Your Browser | by KeyboardSamurai | Medium, дата последнего обращения: февраля 17, 2026, https://medium.com/@keyboardsamurai007/webrtc-is-the-silent-ip-leak-living-in-your-browser-cb72c46641cb