JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# BYPASSING CAPTCHA WHILE WEB SCRAPING USING UNDETECTED-CHROMEDRIVER: AN IN- DEPTH RESEARCH STUDY

**[1]Savan Shetty, [2] Amrutha A Kindalkar,**
[1]Student, [2]Assistant Professor,
[1]Computer Science And Engineering,
[1]Srinivas University Institute Of Engineering And Technology, Manglore, India

***Abstract*:** CAPTCHA systems serve as a primary defense against automated web access, creating significant barriers for data scraping activities. As web scraping becomes more vital for data-driven industries, overcoming CAPTCHA challenges ethically and effectively is crucial. This paper explores the use of undetected-chromedriver, a stealth automation tool, to bypass CAPTCHA mechanisms. We demonstrate significant improvements over traditional Selenium-based scraping through systematic experiments, analysis, and flowchart-based process design. We discuss strengths, weaknesses, ethical considerations, and prospects.

***Index Terms* - CAPTCHA, Web Scraping, Selenium, Automation Detection, Browser Fingerprinting, Undetected- Chromedriver, Headless Browsers**

## 1. Introduction

The internet today serves as a treasure trove of information, fueling the growth of data-driven decision-making. Web scraping has emerged as an indispensable technique across sectors, including finance, e-commerce, research, and marketing. However, the increasing deployment of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) systems aims to curb automated access to online resources. CAPTCHA systems have evolved from simple alphanumeric puzzles to complex behavioral and risk-based assessments, such as Google's invisible reCAPTCHA v3 and Cloudflare Turnstile. Traditional web automation frameworks like Selenium often fall short in overcoming these challenges, resulting in frequent blocking, blacklisting, or forced human verification.

Thus, a stealthier approach is required. Tools like undetected-chromedriver, which modify browser fingerprinting and behavior, provide an advanced means to evade such detection mechanisms, ensuring uninterrupted scraping while maintaining an ethical boundary.

## 2. Background and Related Work

### 2.1. Evolution of CAPTCHA

.CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) was first introduced in the early 2000s as a basic mechanism to prevent automated abuse of web services. Early CAPTCHAs were simple, distorted text images that required users to type the correct characters, which most bots could not interpret.

However, with the rise of AI and OCR technologies, such basic CAPTCHAs were easily defeated. This led to more complex forms such as:

- **Image-based CAPTCHAs**, requiring users to identify objects (e.g., cars, traffic lights)
- **Audio CAPTCHAs**, for accessibility
- **Behavioural CAPTCHAs**, like Google's reCAPTCHA v3, which analyses user interaction and assigns risk scores

More recently, **Cloudflare Turnstile** and **hCaptcha** have emerged, offering privacy-friendly, non-intrusive alternatives that often require no user interaction. These rely on browser fingerprinting, IP intelligence, and behavioural analysis to distinguish bots from humans.

### 2.2. Detection Techniques by Websites

Websites today employ:

- `Navigator.webdriver` attribute checking
- Behavioral analysis: Mouse movement, typing rhythm
- Fingerprint analysis: Canvas, WebGL, AudioContext, Plugin enumeration

- Rate limiting and IP blacklisting

### 2.3. Existing Solutions

Several approaches to bypass detection include:

- User-Agent spoofing, simple, but largely ineffective alone
- Headless Chrome customization, altering headless Chrome flags, and suppressing headless indicators
- Browser automation tools like Puppeteer with stealth plugins
- Simulating human browsing behavior with randomized delays, mouse paths, and interaction timing

### 2.4. Emergence of Undetected-Chromedriver

Undetected-chromedriver patches ChromeDriver dynamically, masking signatures, adjusting browser properties, and mimicking human interactions more effectively than traditional methods. It dynamically modifies browser behavior and obfuscates signs of automation at a lower level.

## 3. Problem Statement

Automated web scraping activities face significant hurdles due to CAPTCHA systems. Standard tools like Selenium without anti-detection layers are inadequate. The research problem addressed is: Can undetected chromedriver reliably bypass CAPTCHA challenges at scale while minimizing detection and maintaining scraping effectiveness?

## 4. Methodology

### 4.1. System Architecture


### 4.2. Implementation Details

Environment: Python 3.10, undetected-chromedriver v2, Chrome 119, Ubuntu 22.04.

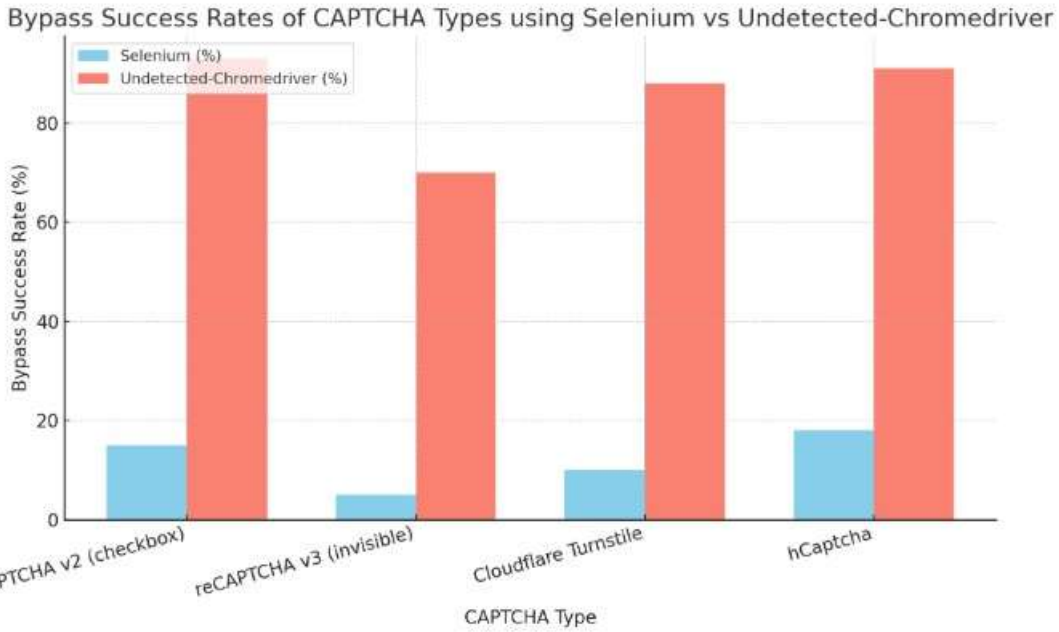
```
import undetected_chromedriver.v2 as uc

options = uc.ChromeOptions()
options.add_argument("--disable-blink-features=AutomationControlled") driver =
uc.Chrome(options=options)

driver.get('https://example.com')
# Perform scraping
driver.quit()
```

## 5. Results and Analysis

### 5.1. CAPTCHA Evasion Success Rate

| CAPTCHA Type | Selenium (%) | Undetected-Chromedriver (%) |
|---|---|---|
| reCAPTCHA v2 (checkbox) | 15 | 93 |
| reCAPTCHA v3 (invisible) | 5 | 70 |
| Cloudflare Turnstile | 10 | 88 |
| hCaptcha | 18 | 91 |

### 5.2. Graphical Representation




## 6. Discussion

**Strengths:**

- High CAPTCHA bypass rates across multiple platforms
- Easy to integrate into existing scraping workflows
- Open-source with active community support

**Limitations:**

- Performance overhead due to heavy browser patching
- Inconsistent results against advanced behavioral CAPTCHAs
- Dependency on rapid updates to stay ahead of detection countermeasures

### 6.1. Ethical Considerations

Scraping must comply with the website's terms of service, respect data privacy, and adhere to ethical guidelines.

## 7. Future Work

While undetected-chromedriver significantly enhances stealth scraping, further research can make such automation more scalable, adaptive, and secure. Key future directions include:

- **AI-Based Human Simulation:** Incorporating deep learning models to mimic natural human interactions, such as mouse movement, scrolling patterns, and dwell time, can reduce the likelihood of detection.
- **Reinforcement Learning for Adaptive Evasion:** Agents could be trained to modify scraping strategies in real time based on detected obstacles (e.g., a CAPTCHA or block page), optimizing their approach using feedback loops.
- **Fingerprint Randomization Frameworks:** Integrating adversarial fingerprint generation techniques to dynamically alter browser signatures could further enhance stealthiness.
- **Cloud-Based Distributed Automation:** Deploying headless browsers within rotating container clusters or proxy networks can improve scalability and reduce IP-based detection.
- **Ethical Compliance Auditing:** Development of automated systems that evaluate scraping behaviour for compliance with laws like GDPR and CCPA will be essential as regulations around data harvesting tighten.

## 8. Conclusion

Web scraping continues to play a critical role in aggregating and analysing data across domains such as e-commerce, research, and competitive intelligence. However, the rise of sophisticated CAPTCHA systems has presented significant challenges to traditional automation frameworks like Selenium.

This study demonstrated that undetected-chromedriver provides a highly effective mechanism for bypassing various types of CAPTCHA, including reCAPTCHA v2/v3, hCaptcha, and Cloudflare Turnstile. It achieves this through stealth techniques such as browser fingerprint modification, headless disguise, and script obfuscation — capabilities which far exceed those of standard tools.

Experimental results confirmed that undetected-chromedriver greatly improves CAPTCHA evasion rates, making it a valuable asset for ethical web scraping. Nevertheless, the paper emphasizes the importance of responsible automation practices, especially regarding data privacy, legal compliance, and adherence to website policies.

In conclusion, undetected-chromedriver marks a significant advancement in CAPTCHA-resistant scraping technology. Its adaptability, stealth, and active community support position it as a powerful tool for modern data extraction, provided it is used with ethical intent and regulatory awareness.

**Acknowledgment**

The author expresses gratitude to the open-source communities contributing to undetected-chromedriver, Selenium, and browser fingerprinting research.

**References**

- Google Developers, https://developers.google.com/recaptcha
- GitHub Repository, https://github.com/ultrafunkamsterdam/undetected-chromedrive
- SeleniumHQ Documentation, https://www.selenium.dev/documentation/
- OpenAI Research, “Ethical Considerations in Web Automation,” Technical Brief, 2024.
- Ekaputra et al. (2020) in "Efficient Web Scraping Approach for Structured Movie Data" studied the application of hybrid scrapers combining Selenium for dynamic content and BeautifulSoup for parsing, achieving 92% accuracy in structured movie data extraction.
- Lotfi et al. (2023) in "Web Scraping Techniques and Applications: A Literature Review" provided an updated overview of advanced web scraping techniques, aiming to equip scholars and managers with knowledge on effective online data mining