Website using DataDome gets captcha blocked while scraping using Selenium and Python

Asked 4 years, 6 months ago Modified 1 year, 7 months ago Viewed 17k times

9

I'm actually trying to scrape some car datas from different websites, i've been using selenium with chromebrowser but some websites actually block selenium with captcha validation(example: https://www.leboncoin.fr/), and this in just 1 or 2 requests. I tried changing $_cdc in the chromebrowser but this didn't resolve the problem, and I've been using those options for the chromebrowser

```
user_agent = 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.97 Safari/537.36'
options = webdriver.ChromeOptions()
options.add_argument(f'user-agent={user_agent}')
options.add_argument('start-maximized')
options.add_argument('disable-infobars')
options.add_argument('--profile-directory=Default')
options.add_argument("--incognito")
options.add_argument("--disable-plugins-discovery")
options.add_experimental_option("excludeSwitches", ["ignore-certificate-errors", "safebrowsing-disable-download-protection", "safebrowsing-disable-auto-update", "disable-client-side-phishing-detection"])
options.add_argument('--disable-extensions')
browser = webdriver.Chrome(chrome_options=options)

browser.delete_all_cookies()

browser.set_window_size(800,800)

browser.set_window_position(0,0)
```

The website I'm trying to scrape uses DataDome for bot security, any clue ?

python selenium google-chrome web-scraping botdetect

Share
Improve this question
Follow

edited Jun 14, 2020 at 17:21

undetected Selenium
193k ● 44 ● 300 ● 374

asked Jun 4, 2020 at 16:07

Mikycid
111 ● 1 ● 1 ● 4

5 Answers

Sorted by: Highest score (default)

To avoid anti-web scraping services like Datadome, we first should understand how they work, which really boils down to 3 categories of detection:

1. IP address
2. Javascript Fingerprint
3. Request details

Services like Datadome use these tools to calculate a trust score for every visitor. A low score means you're likely to be a bot, so you'll either be requested to solve a captcha or denied access entirely. So, how do we get a high score?

IP Addresses / Proxies

For IP addresses, we want to distribute our load through proxies, and there are several kinds of IP addresses:

- **Datacenter**: addresses assigned to big corporations like Google Cloud, AWS, etc.
  These are **awful for your bot score** and should be avoided.
- **Residential**: addresses assigned to living spaces.
  These are great for your bot score.
- **Mobile**: addresses assigned to phone cell towers.
  These are just as good as residential or sometimes even better.

So, to maintain a high trust score, our scraper should rotate through a pool of residential or mobile proxies.

Javascript Fingerprint

This topic is way too big for a StackOverflow question, though let's do a quick summary

Websites can use Javascript to fingerprint the connecting client (the scraper) as javascript leaks an enormous amount of data about the client: operating system, support fonts, visual rendering capabilities, etc.

So, for example: if Datadome sees a bunch of Linux clients connecting through 1280x720 windows, then it can simply deduce that this sort of setup is likely a bot and gives everyone with these fingerprint details low trust scores.

If you're using Selenium to bypass Datadome, you need to patch many of these holes to get out of the low trust zone. This can be done by patching the browser itself to fake fingerprinted details like operating system etc.

For more on this, see my blog [How to Avoid Web Scraping Blocking: Javascript]

Request Details

Finally, even if we have loads of IP addresses and patch our browser from leaking key fingerprint details, Datadome can still give us low trust scores if our connection patterns are unusual.

To get around this, our scraper should scrape in non-obvious patterns. It should connect to non-target pages like the website's homepage once in a while to appear more human-like.

Now that we understand how our scraper is being detected, we can start researching how to get around that. Selenium has a big community and the keyword to look for here is "stealth". For example, selenium-stealth (and its forks) is a good starting point to patching Selenium fingerprint leaks.

Unfortunately, this scraping area is not very transparent, as Datadome can simply collect publicly known patches and adjust their service accordingly. This means you have to figure out a lot of stuff yourself or use a web scraping API to do that for you to scrape protected websites past the first few requests.

I've fitted as much as I can into this answer so for more information see my series of blog articles on this issue How to Scrape Without Getting Blocked

Share Improve this answer Follow

answered Aug 26, 2022 at 7:37

Granitosaurus
21.4k ● 5 ● 63 ● 86

3

A bit more details about your usecase on scraping car datas from different websites or from https://www.leboncoin.fr/ would have helped us to construct a more canonical answer. However, I was able to access the *Page Source* using Selenium as follows:

- Code Block:

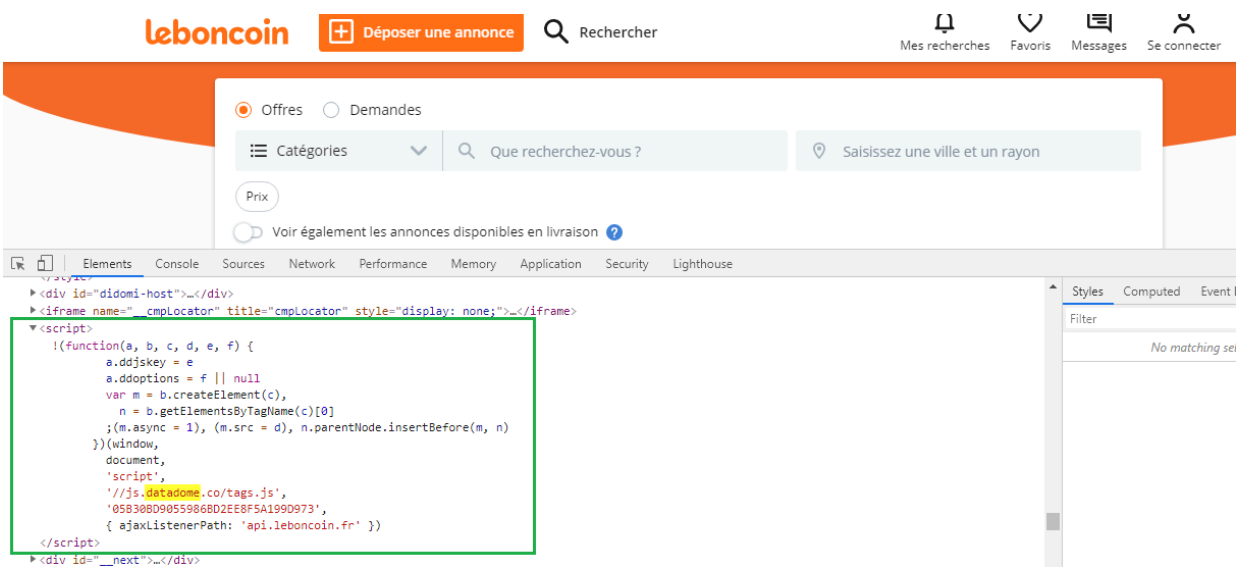
```
  from selenium import webdriver

  options = webdriver.ChromeOptions()
  options.add_argument("start-maximized")
  options.add_experimental_option("excludeSwitches", ["enable-automation"])
  options.add_experimental_option('useAutomationExtension', False)
  driver = webdriver.Chrome(options=options,
executable_path=r'C:\WebDrivers\chromedriver.exe')
  driver.get('https://www.leboncoin.fr/')
  print(driver.page_source)
```

- Console Output:

```
  <html class="gServer"><head><link rel="preconnect"
href="//fonts.googleapis.com" crossorigin=""><link rel="preload"
href="https://fonts.googleapis.com/css2?
family=Open+Sans:wght@400;600;700&amp;display=swap" crossorigin="" as="style
<link rel="stylesheet" href="https://fonts.googleapis.com/css2?
family=Open+Sans:wght@400;600;700&amp;display=swap" crossorigin=""><style da
emotion-css=""></style><meta charset="utf-8"><link rel="manifest"
href="/manifest.json"><link type="application/opensearchdescription+xml"
rel="search" href="/opensearch.xml"><meta name="theme-color" content="#ff6e1
<meta property="og:locale" content="fr_FR"><meta property="og:site_name"
content="leboncoin"><meta name="twitter:site" content="leboncoin"><meta http
equiv="P3P" content="CP=&quot;This is not a P3P policy&quot;"><meta
name="viewport" content="initial-scale=1.0, width=device-width, maximum-
scale=1.0, user-scalable=0"><script type="text/javascript" async=""
src="https://www.googleadservices.com/pagead/conversion_async.js"></script>
<script type="text/javascript" async=""
src="https://tp.realytics.io/sync/se/cnktbDNiMG5jb3xyeV83NTFGRUQwMy1CMDdGLTR
ct=1&amp;rt=1&amp;u=https%3A%2F%2Fwww.leboncoin.fr%2F&amp;r=&amp;ts=15913060
</script><script type="text/javascript" async=""
src="https://www.googleadservices.com/pagead/conversion_async.js"></script>
<script type="text/javascript" async=""
src="https://www.googleadservices.com/pagead/conversion_async.js"></script>
<script type="text/javascript" async=""
src="https://www.googletagmanager.com/gtag/js?id=AW-
766292687&amp;l=dataLayer&amp;cx=c"></script><script type="text/javascript"
async="" src="https://www.googletagmanager.com/gtag/js?id=AW-
667462656&amp;l=dataLayer&amp;cx=c"></script><script type="text/javascript"
async="" src="https://cdn-eu.realytics.net/realytics-1.2.min.js"></script>
<script type="text/javascript" async="" src="https://i.realytics.io/tc.js?
cb=1591306047755"></script><script type="text/javascript" async=""
src="https://www.googletagmanager.com/gtag/js?id=DC-
4167650&amp;l=dataLayer&amp;cx=c"></script><script type="text/javascript"
async="" src="https://www.googletagmanager.com/gtag/js?id=AW-
744431185&amp;l=dataLayer&amp;cx=c"></script><script type="text/javascript"
async="" charset="utf-8"
src="//www.googleadservices.com/pagead/conversion_async.js" id="utag_82">
</script><script type="text/javascript" async="" charset="utf-8"
src="//sdk.mpianalytics.com/pulse.min.js" id="utag_47"></script><script
async="true" type="text/javascript" src="https://sslwidget.criteo.com/event?
a=50103&amp;v=5.5.0&amp;p0=e%3Dexd%26site_type%3Dd&amp;p1=e%3Dvh&amp;p2=e%3D
data-owner="criteo-tag"></script><script type="text/javascript"
src="//try.abtasty.com/09643a1c5bc909059579da8aac99e8f1.js"></script>
<script>window.dataLayer = window.dataLayer || [];
  .
  .
  .
  <iframe height="1" width="1" style="display:none"
src="//4167650.fls.doubleclick.net/activityi;src=4167650;type=slbc01;cat=all
site;u1=homepage;ord=9979622847645.51?" id="utag_179_iframe"></iframe></body
</html>
```

However, it's quite evident from the DOM Tree that the website is protected from *Bad Bots* through **DataDome** as in:



DataDome

The key features are as follows:

- DataDome is the only bot protection solution delivered as-a-service.
- DataDome requires no architecture changes or DNS rerouting.
- DataDome's bot detection engine compares every request to the website with a massive in-memory pattern database, and uses a blend of AI and machine learning to decide in less than 2 milliseconds whether access to your pages should be granted or not.
- DataDome detects and identifies 100% of OWASP automated threats.
- DataDome's Custom Rules function can even allow you to block human traffic from countries you are not selling to, or to allow partner bots to access your site only in specific circumstances.

Outro

Documentation on DataDome can be found at:

- Bot detection
- Server-side bot detection is not enough

Share

Improve this answer

Follow

edited May 23, 2023 at 22:53

Gilles Quénot

184k ● 43 ● 230 ● 229

answered Jun 4, 2020 at 22:34

undetected Selenium

193k ● 44 ● 300 ● 374

It could be happening due to a myriad of reasons. Try going through the answer here that gives someway in you can prevent this problem.

0

A simple solution that worked for me sometimes is to use `Waits` / `Sleep` calls in selenium, see here from the docs about Waits. Or sleep calls can be done like so

```
Import time
time.sleep(2)
```

Share Improve this answer Follow

answered Jun 4, 2020 at 21:07

AzyCrw4282

7,724 ● 5 ● 22 ● 39

I use sleeps every request already, it's not about this as I get blocked in so few requests, I could even go browsing this website on chromedriver that in less than 3 requests I would get to validate the captcha, their javascript is able to detect I'm using an emulated chrome for some reason – Mikycid Jun 5, 2020 at 14:37

0

What's the problem with captcha? You can solve it with cheap service like Anti Captcha or others. Here's an example with NodeJS: https://github.com/MoterHaker/bypass-captcha-examples/blob/main/geo.captcha-delivery.com.js

Share Improve this answer Follow

answered Dec 29, 2020 at 13:05

MotaHaker

47 ● 1

0

I'm in the Web Scraping industry for years and my experience with Datadome suggests the following, at the moment.

1. Use a more evolved automation library like Playwright to scrape Datadome-protected websites.
2. Use a headful browser (but not Chrome). Playwright has bundled also Firefox, I've tried also Brave Browser and it works.
3. If you have some budget, you can consider also paid solutions like Zyte API or Bright Data Web Unlocker. They both work well against Datadome.

I've recently tested these solutions and they work on another Datadome-protected website and results may differ from case to case. An example of scraper with Playwright and Firefox is the following:

```
import time
from playwright.sync_api import sync_playwright
import asyncio
```

```
from gologin import GoLogin
import asyncio
import time
import csv
from random import randrange
from scrapy.http import HtmlResponse


with sync_playwright() as p:
    browser = p.firefox.launch_persistent_context(user_data_dir='./userdata/',
headless=False,slow_mo=200)
    page = browser.new_page()
    page.goto('https://www.footlocker.it/', timeout=0)
    interval=randrange(3,10)
    time.sleep(interval)
    try:
        page.locator("xpath=//button[@id='onetrust-accept-btn-
handler']").click()
        interval=randrange(10)
        time.sleep(interval)
    except:
        pass

    try:
        page.locator("xpath=//div[@class='col
HeaderNavigation']/div/button[contains(text(), 'Uomo')]").click()
        interval=randrange(10)
        time.sleep(interval)
    except:
        pass
    try:
        page.locator("xpath=//li[@class='MegaMenu-link']/a[contains(text(),
'Tutte le scarpe da uomo')]").click()
        interval=randrange(10)
        time.sleep(interval)
    except:
        pass

    html_page=page.content()
    response_sel = HtmlResponse(url="my HTML string", body=html_page,
encoding='utf-8')
    product_urls = response_sel.xpath('//a[@class= "ProductCard-link
ProductCard-content"]/@href').extract()
    for url in product_urls:
        page.goto('https://www.footlocker.it/'+url, timeout=0)
        interval=randrange(3,10)
        time.sleep(interval)
```

Run code snippet Expand snippet

Share Improve this answer Follow

answered Apr 14, 2023 at 9:49

Pierluigi Vinciguerra

104 • 7

Is there something that playwright chrominum / firefox exposes that makes it different to an actual chrome/firefox browser? I tried this snippet of yours and somehow `footlocker.it` is

still detecting the requests as a bot. – mang4521 Jun 18 at 23:59