



In Plain Sight

Executive Summary	1
Puppeteer Features	2
Tools Used to Circumvent Detection	2
Puppeteer Stealth	2
Developer Discussions	4
Captcha Solvers	5
Playwright	7
Amazon Elastic Compute Cloud	8
Companies' Responses are Insufficient	8
Google Display Ads	8
Google, Bing, and Yahoo Search	9
Facebook	12
Corporate Landing Pages	17
Conclusion	18



Executive Summary

Bot traffic is any online activity performed by an automated process rather than human action. Bots are often associated with cybercrime and malware, but it is far easier to generate robotic activity with developer tools such as headless Chrome and Amazon Web Services.

Programmatically controlled web browsers can imitate even the most sophisticated human activity online. This includes visiting websites, consuming online media, writing social media posts, clicking on advertisements, installing applications, and filling out lead forms.

Advertisers purchase engagement in the form of views, clicks, likes, app installs, etc. under the impression that they are receiving the attention of real human users. Due to the sophistication of browser automation technologies and the lack of implementation of anti-bot measures, advertisers routinely end up paying for robotic (fraudulent) activity.

Browser bots use automation tools such as Selenium, Puppeteer, and Playwright. These tools are developed and maintained by Google and Microsoft, and can be deployed at enormous scale using cloud computing technologies provided by Amazon (AWS), Google (GCP), and Microsoft (Azure). Puppeteer has been downloaded over 100 million times.

These tools were created to help developers streamline web testing, but are also used to simulate human activity online at the expense of large and small businesses. Sophisticated bots which emulate human behaviors and technical characteristics are available free of cost and can be scaled inexpensively.

Commentary on the size and scale of ad fraud is not novel. Most estimates place the annual cost of ad fraud in the tens of billions of dollars. What MMI sees is lacking is the understanding and discussion of the tools that are used to perpetrate fraud and inauthentic activity.

Major advertising platforms and corporate websites have responded to bot traffic in a manner that is detrimental to their customers, users, and their own businesses. In this report, we describe what developer tools are capable of, show how they can be used to generate robotic traffic, and provide examples of how bots are inadequately handled by companies like Google, Microsoft, Facebook, and Yahoo.

MMI provides patented solutions to detect bot traffic for the purpose of verifying online advertising, web traffic, and application security.

Method Media Intelligence

<https://methodmi.com>

info@methodmi.com



Puppeteer Features

Puppeteer has prebuilt features which can emulate dozens of human characteristics. Google has provided detailed documentation for all Puppeteer features [online](#).

1. A [function](#) to emulate 73 different [types](#) of devices. This feature enables a single computer or data center to appear as dozens of different device types (iPhone 6/7/8/10, Android devices, Blackberry, iPad, etc.).
2. A [function](#) to move, click, and [hover](#) with the mouse pointer.
3. A [function](#) to press any key and type any text on the keyboard.
4. [Ability](#) to set a new user agent for each page. This can be used to pass false operating system types/versions, browser types/versions, etc.
5. Ability to set an artificial [geolocation](#) for each page, including latitude and longitude.
6. Ability to set the [viewport](#) (screen resolution) of the browser on each page.
7. Ability to set [cookies](#) and specify urls, values, expiry, etc.
8. Runs in headless mode by default, meaning no graphical interface is displayed.

When used together, these features allow even novice programmers to construct bots which exhibit human-like activity online.

Tools Used to Circumvent Detection

Puppeteer Stealth

Puppeteer Stealth is a plugin for Puppeteer created by third-party developers, hosted on [NPM](#) and [Github](#) (both owned by Microsoft). This plugin was released on May 29th, 2018 and currently includes support for 12 different methods to evade bot detection.

Total number of downloads between 2018-05-29 and 2020-07-23:

package	downloads
puppeteer-extra-plugin-stealth	4,926,672

Puppeteer stealth has been downloaded nearly 5 million since inception. ([Source](#))

Total number of downloads between 2018-05-29 and 2020-07-23:

package	downloads
puppeteer	106,016,687

Method Media Intelligence

<https://methodmi.com>

info@methodmi.com



Puppeteer has been downloaded 106 million times in that same period. ([Source](#))

Implementation of this plugin to an existing Puppeteer script requires installing the plugin and adding only 2 lines of code:

```
const puppeteer = require('puppeteer-extra');
puppeteer.use(require('puppeteer-extra-plugin-stealth')())
```

The relative ease of implementation and regular updates has led to the plugin being used widely by web scrapers. Many enterprise sites including Zillow either cannot [detect](#) Puppeteer Stealth or rely on Captchas, which create an obstacle for human users.

A demonstration of the changes that Puppeteer stealth makes can be viewed [here](#).

Intoli.com tests + additions

Test Name		Result
User Agent	(Old)	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/83.0.4103.0 Safari/537.36
WebDriver	(New)	present (failed)
WebDriver Advanced		passed
Chrome	(New)	missing (failed)
Permissions	(New)	prompt
Plugins Length	(Old)	0
Plugins is of type PluginArray		failed
Languages	(Old)	en-US
WebGL Vendor		Google Inc.
WebGL Renderer		Google SwiftShader
Hairline Feature		missing
Broken Image Dimensions		16x16

Screenshot taken with standard Puppeteer bot.

Intoli.com tests + additions

Test Name		Result
User Agent	(Old)	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.2403.157 Safari/537.36
WebDriver	(New)	missing (passed)
WebDriver Advanced		passed
Chrome	(New)	present (passed)
Permissions	(New)	denied
Plugins Length	(Old)	3
Plugins is of type PluginArray		passed
Languages	(Old)	en-US,en
WebGL Vendor		Intel Inc.
WebGL Renderer		Intel Iris OpenGL
Hairline Feature		missing
Broken Image Dimensions		16x16

Method Media Intelligence

<https://methodmi.com>

info@methodmi.com



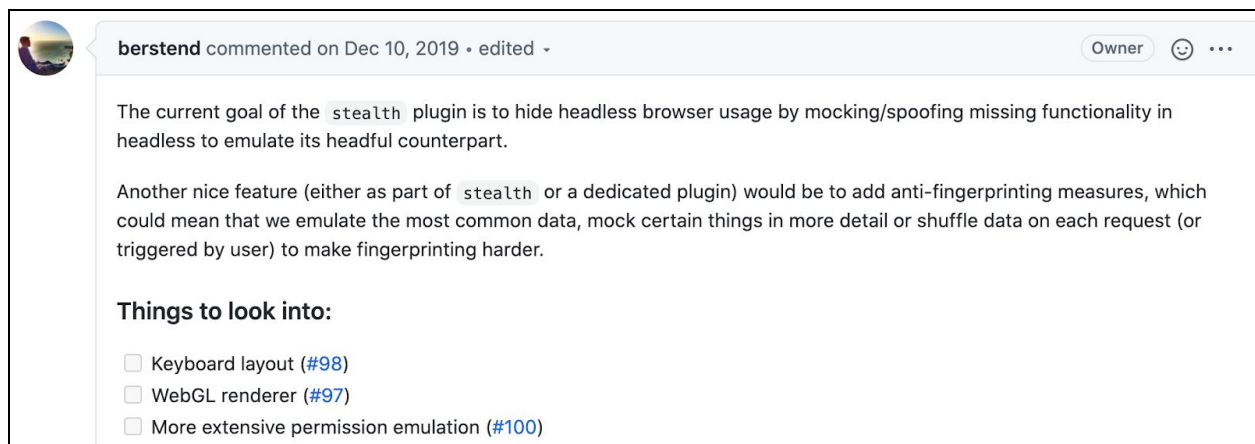
Screenshot taken with Puppeteer Stealth bot.

Upon learning of the existence of Puppeteer Stealth, MMI updated our bot detection technology. We can now identify all versions of headless Chrome on Puppeteer, including implementations that use the stealth plugin and attempt to patch native values.

MMI also informed Google's Chromium team of the existence of the stealth plugin and the clear attempt to use Chrome to bypass existing security tools. The Chrome team maintained that the software is working as intended, and that the existence of Puppeteer Stealth does not require action on their part.

Developer Discussions

Puppeteer Stealth is one small part of the larger cat and mouse game between bot detection companies and bot operators. Software development discussion forums such as StackOverflow and Github are rich with discussion and guidance on how to use Puppeteer to evade existing bot detection measures.



The creator of Puppeteer stealth clearly [states](#) the purpose of this plugin is to make headless Chrome imitate its headful (human operable) counterpart.

Puppeteer Stealth's Github [page](#) is one of the most active forums, where users and developers regularly discuss how to evade bot detection techniques.

How to give fake GPU info to site

Asked 1 year, 2 months ago Active 1 year, 2 months ago Viewed 589 times



I need to create web browser using CefSharp.Wpf with ability to give fake data to site for example CPU cores, browser plugins, platform name etc. There are site that can retrieve all this info:
<https://www.deviceinfo.me/>



My quesiton is: How to hide GPU info from this site? Using javascript or CefSharp functionality

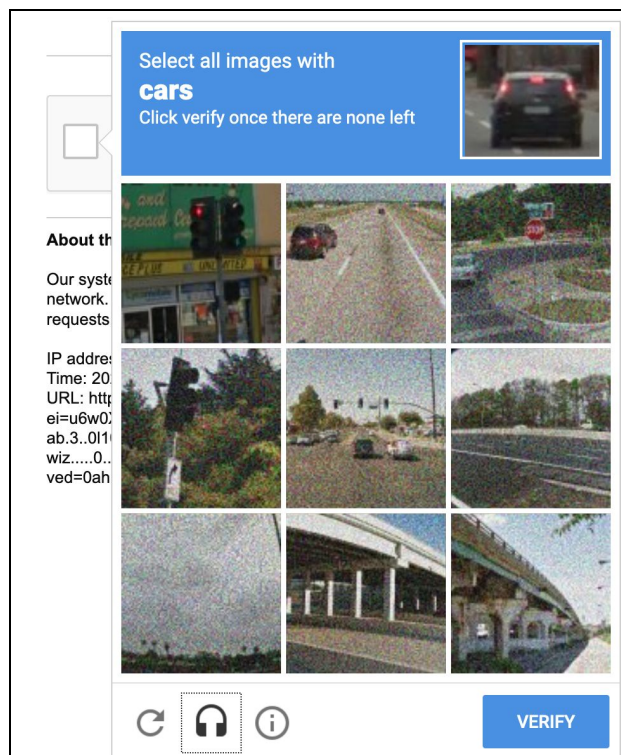


I have tried to redefine WebGLRenderingContext.getParameter method, which gives an info about GPU renderer and vendor:

A user on Stack Overflow [asking](#) how to pass fake information to a site through a browser. They received a detailed description of how to solve the problem.

Captcha Solvers

Captchas are a common method used to validate if a browser is controlled by a human or a bot. However, incorporating Captcha solving into a browser bot is a solved problem and is relatively inexpensive.



Example of a Captcha on Google Search.

Method Media Intelligence

<https://methodmi.com>

info@methodmi.com



This [video](#) explains how a captcha service can be used to solve captchas and automate the creation of Reddit accounts. These services are powered by workers manually solving captchas, and returning the results to end-users via an API.

A Puppeteer [extension](#) created by an unaffiliated developer integrates with captcha solving services to add additional evasion capabilities to bots.

My anarchist contribution to this discussion is to demonstrate this absurdity, with a plugin for robots with which **a single line of code is all it takes to bypass reCAPTCHAs on any site.**

I thought about having the plugin solve captchas directly (e.g. using the **audio challenge** and speech-to-text APIs), but external solution providers are so cheap and reliable that there is really no benefit in doing that. `\\_(\`)/_`

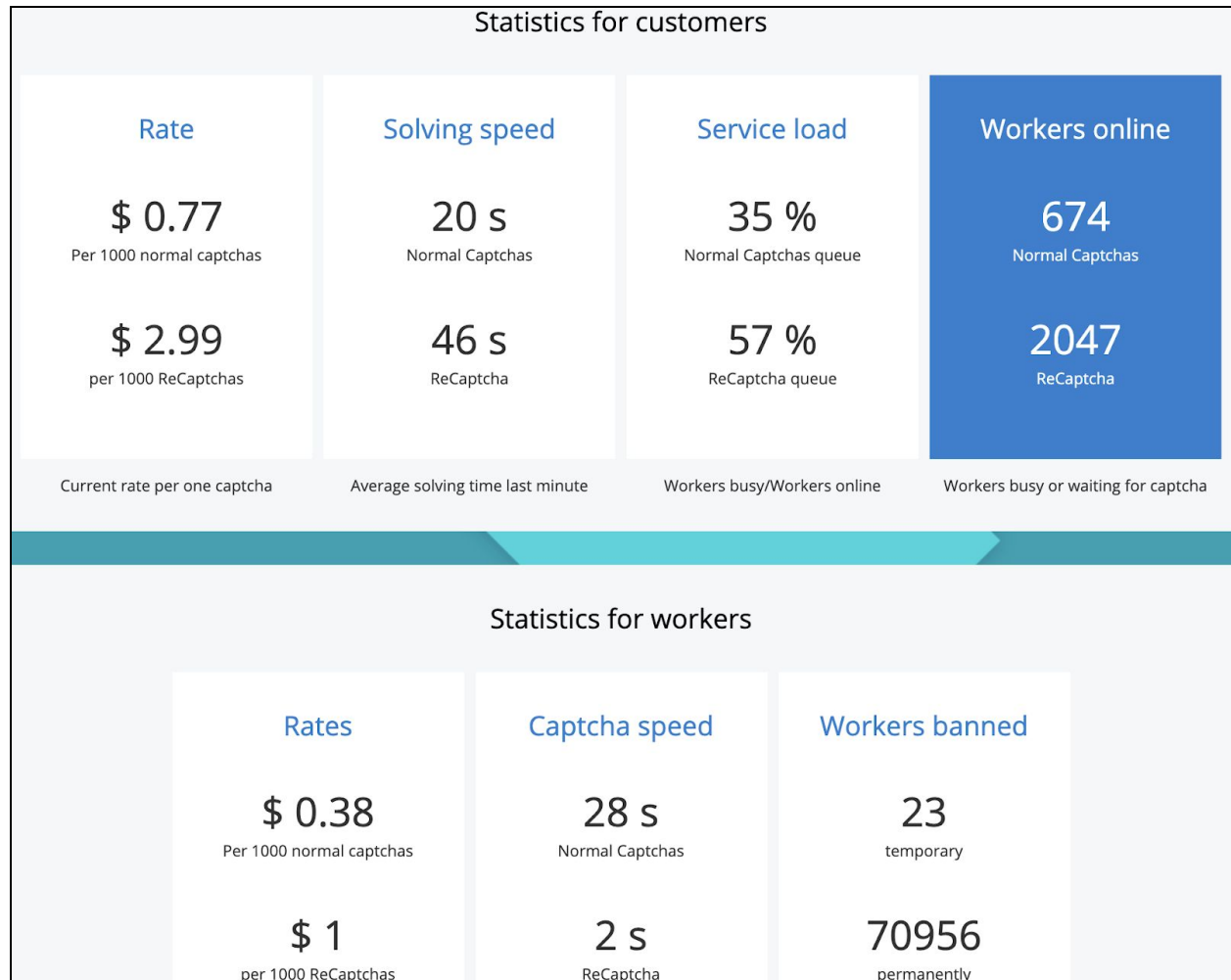
Berstend's reasoning and comments on the plugin he created.

Total number of downloads between 2019-01-01 and 2020-08-11:

package	downloads
puppeteer-extra-plugin-recaptcha	75,157

The Puppeteer Recaptcha plugin has been downloaded over 75,000 times since inception ([Source](#)).

As demonstrated in the YouTube video, using these services in combination with a headless browser does not result in bot detection on Reddit. Given the relatively low cost of using the Captcha solving service, Captchas do not pose a significant hurdle to bot operators.



Pricing for two of the common vendors is:

[2captcha](#) - \$0.77 / 1,000 Captchas, \$2.99 / 1,000 ReCaptchas. 674 workers online to solve Captchas.

[Deathbycaptcha.com](#) - \$1.39 / 1,000 Captchas

Playwright

[Playwright](#) is a new browser automation platform by Microsoft. It was released on January 31st, 2020. Two contributing Google engineers from the Puppeteer project moved to Microsoft and are the key developers for the project. Playwright is different from Puppeteer because it has cross-browser (Chromium, Firefox, WebKit) support. Playwright has some features with increased functionality relative to Puppeteer, including improved device simulation.

Method Media Intelligence

<https://methodmi.com>

info@methodmi.com



Puppeteer is still far more popular than Playwright, with 1.5M weekly downloads vs. 65k for Playwright. As Playwright developers continue to improve the platform, bot evasion modules will grow in a similar fashion to Puppeteer.

Amazon Elastic Compute Cloud

A single laptop or desktop can run 10-20 browsers before running into a bottleneck. This inherently limits the ability of a lone operator to scale-up robotic browsers. Purchasing hundreds of laptops to run thousands of browsers is expensive compared to renting the servers from a cloud provider.

A preferable method to control thousands of browsers is to use cloud computing (data center) services such as [Amazon Elastic Compute Cloud](#), [Google Compute](#), or [Microsoft Virtual Machines](#). These provide rentable, inexpensive access to servers that are capable of running hundreds of browsers each. Data centers have the high bandwidth internet connections and even have [versions](#) with browser automation tools pre-installed. The [cost](#) of renting a server with comparable power to a MacBook Pro is about \$0.09/hr on AWS.

Browsers run in the more efficient “[headless mode](#)” on data centers, as the servers are screenless and do not need to render graphics on a display. As a result, these browsers have no way to display any content on a screen, and humans have no opportunity to view it. For this reason, bot detection solutions (attempt to) classify data center traffic as invalid.

A video demonstration of how Puppeteer and AWS can be used to generate web traffic is [here](#). This bot uses a residential proxy service to mask its IP address. These proxy services make IP-based detection methods obsolete, and are available from many suppliers at a low cost.

Companies’ Responses are Insufficient

Google Display Ads

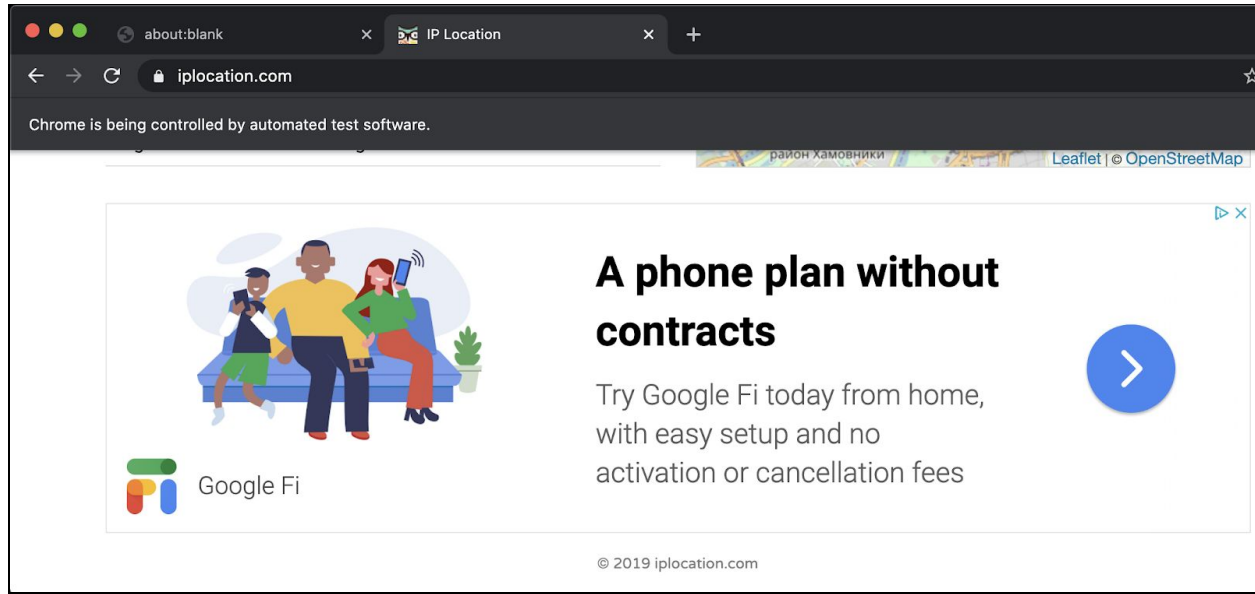
Google’s own ad platform serves their own ads to their own automated browser. This demonstrates a clear gap in control and understanding of the digital advertising environment.

While they don’t have to pay for their own ads, they could have generated revenue from selling that ad space to another advertiser, representing a lost opportunity. It’s clear that their ad platform cannot detect all implementations of automated Chrome.

Method Media Intelligence

<https://methodmi.com>

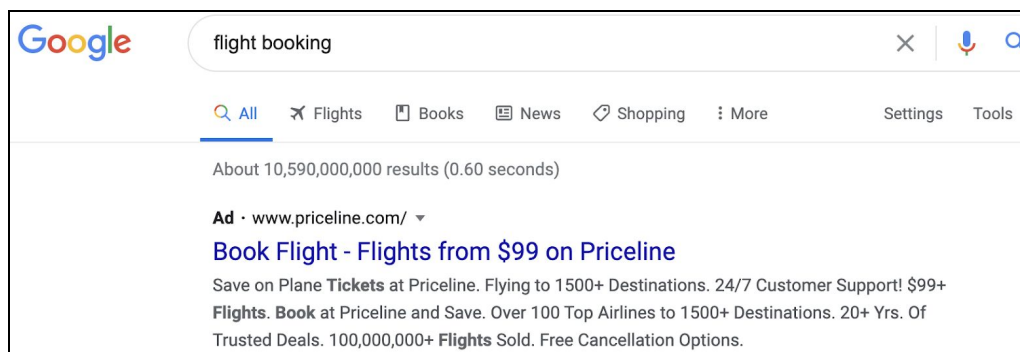
info@methodmi.com



Google, Bing, and Yahoo Search

Google, Bing (Microsoft), and Yahoo are the three largest search engines worldwide by market share, with 96% of the worldwide [market](#) and 98% of the United States [market](#). Search engine marketing can be a highly effective and targeted method to reach new customers. Advertisers can target specific keywords or search terms to show their site as a top search result. Basic search campaigns can have a cost-per-click (CPC) of \$1 – 5, but the CPC for certain high value keywords can run up to \$1,000 per [click](#).

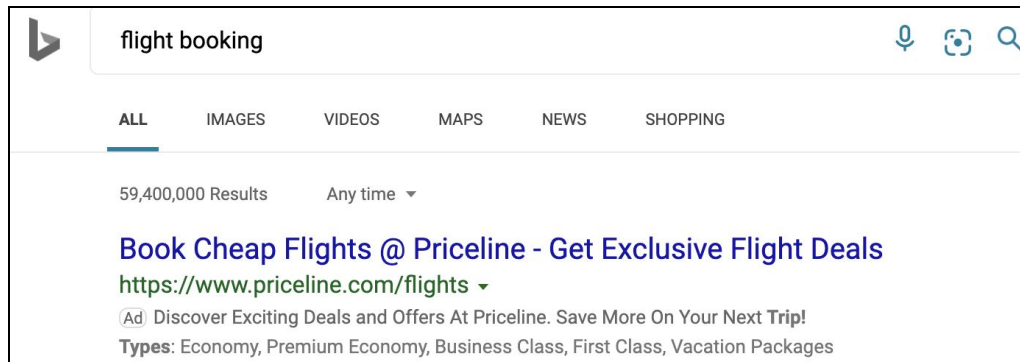
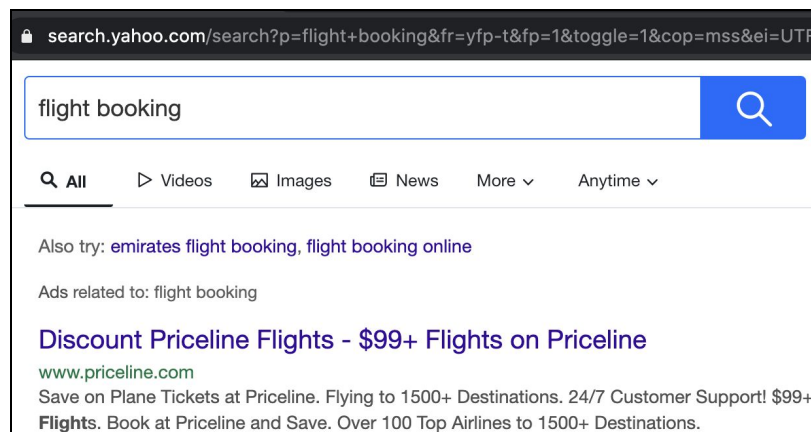
Search results for terms such as “flight booking”, “car rental”, “home furniture”, “athletic shoes”, “luxury clothing” have become a battleground for brands where the highest bidder gets the first result on the page.



Method Media Intelligence

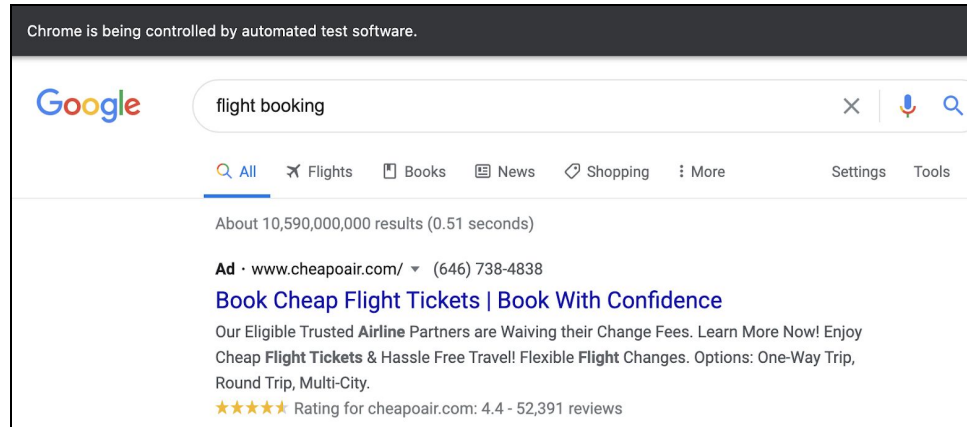
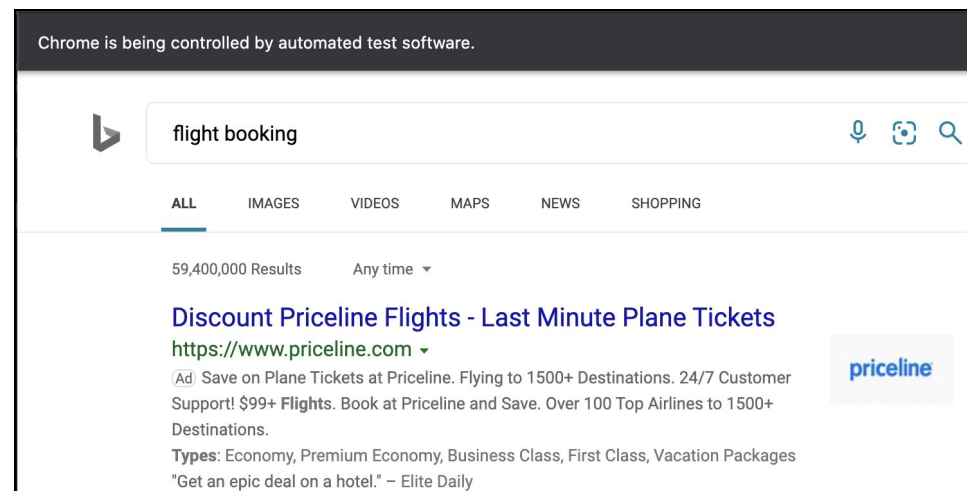
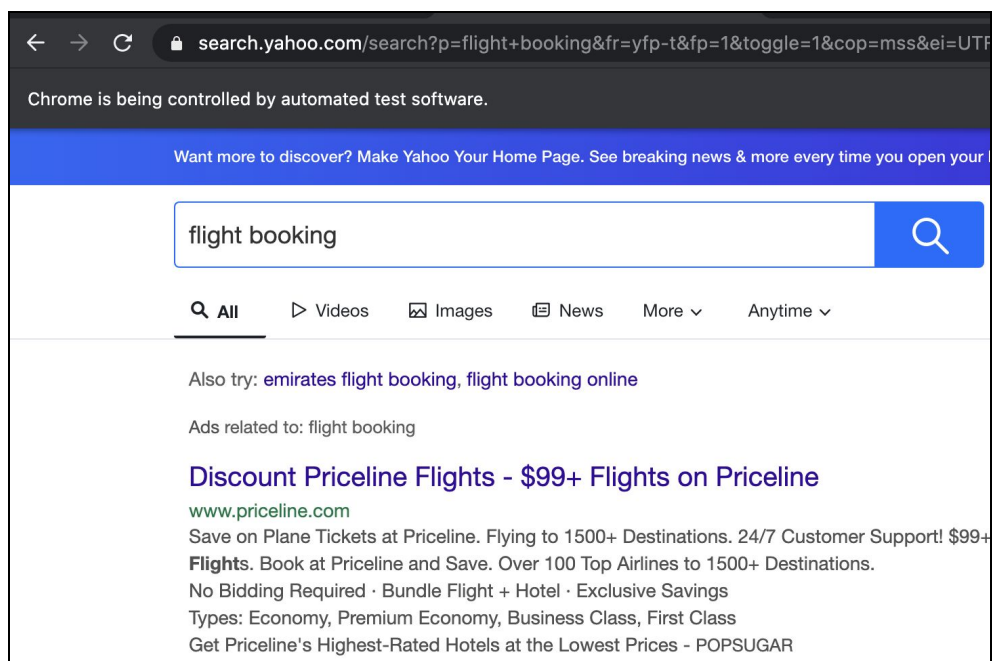
<https://methodmi.com>

info@methodmi.com



Searches for the term "flight booking" return ads on all three search engines.

Brands purchasing paid clicks from search engines expect that users are humans, not robots, and that those users will reach their website after clicking the ad. Unfortunately for advertisers, the major search engines do a poor job of preventing bots from receiving and clicking on ads. For example, an automated version of Chrome (Puppeteer) still gets search ads on Google, Bing, and Yahoo.



Examples of search engines showing ads to automated browsers.

Method Media Intelligence

<https://methodmi.com>

info@methodmi.com



Bots can click through these ads to reach advertisers' websites. Unless an advertiser has their own bot detection code on their site to detect invalid clicks, they are dependent on the search engine provider to detect robotic activity and not bill them for it.

We contacted Google (July 21st), Microsoft (July 16th), and Yahoo (July 20th) about this issue and did not receive a response.

Facebook

Facebook's [terms of service](#) prohibits product access or collection of data through automated means.

2. What you can share and do on Facebook

We want people to use Facebook to express themselves and to share content that is important to them, but not at the expense of the safety and well-being of others or the integrity of our community. You therefore agree not to engage in the conduct described below (or to facilitate or support others in doing so):

1. You may not use our Products to do or share anything:
 - That violates these Terms, our [Community Standards](#), and [other terms and policies](#) that apply to your use of Facebook.
 - That is unlawful, misleading, discriminatory or fraudulent.
 - That infringes or violates someone else's rights, including their intellectual property rights.
2. You may not upload viruses or malicious code or do anything that could disable, overburden, or impair the proper working or appearance of our Products.
3. You may not access or collect data from our Products using automated means (without our prior permission) or attempt to access data you do not have permission to access.

We can remove or restrict access to content that is in violation of these provisions.

Facebook terms of service, August 5th 2020.

Facebook's terms of service make sense because of the significant consequences of allowing bot traffic to go unchecked on the platform. These include:

1. Accurately counting active users
2. Data security
3. Detecting coordinated inauthentic behavior
4. Protecting the integrity of advertising on the platform

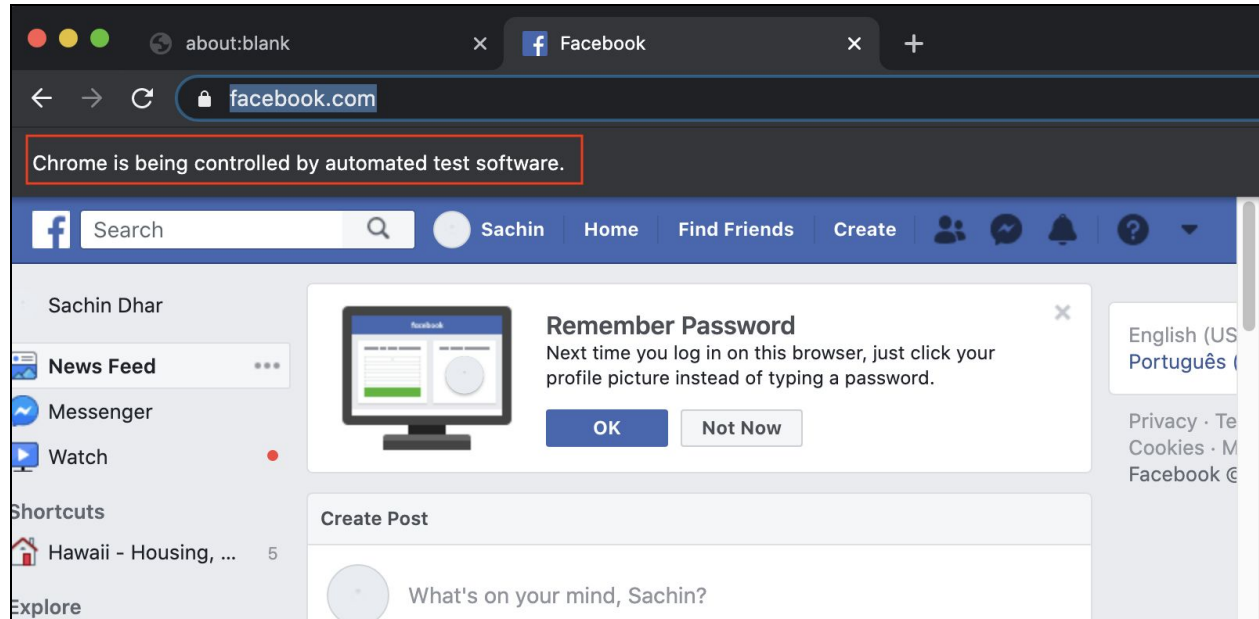
Method Media Intelligence

<https://methodmi.com>

info@methodmi.com

5. Unnecessary bandwidth costs of serving content to bots

However, Facebook's technology does not align with their terms of service. Their site does not block even self declared bots from logging in, viewing content, and engaging with ads.



Facebook allows automated browsers to login to the platform

Unsophisticated automated browsers such as the one shown above can be easily detected by several means, each of which Facebook has the full ability to use.

1. User-agent detection: these browsers contain "Headless Chrome" in their user-agent instead of "Chrome".
2. [Navigator.webdriver](#) is set to "true", instead of "false" or "undefined" in human controlled browsers.

Both of these properties have been designated by Chrome to indicate a programmatically controlled browser.

More advanced bots modify these two (and many other) properties to hide the fact that they are automated. Currently, these bots are able to login to Facebook and interact with the page.

MMI video demonstration showing news feed ads being shown to a bot:

<https://youtu.be/5OteU0s8gps>

Method Media Intelligence

<https://methodmi.com>

info@methodmi.com

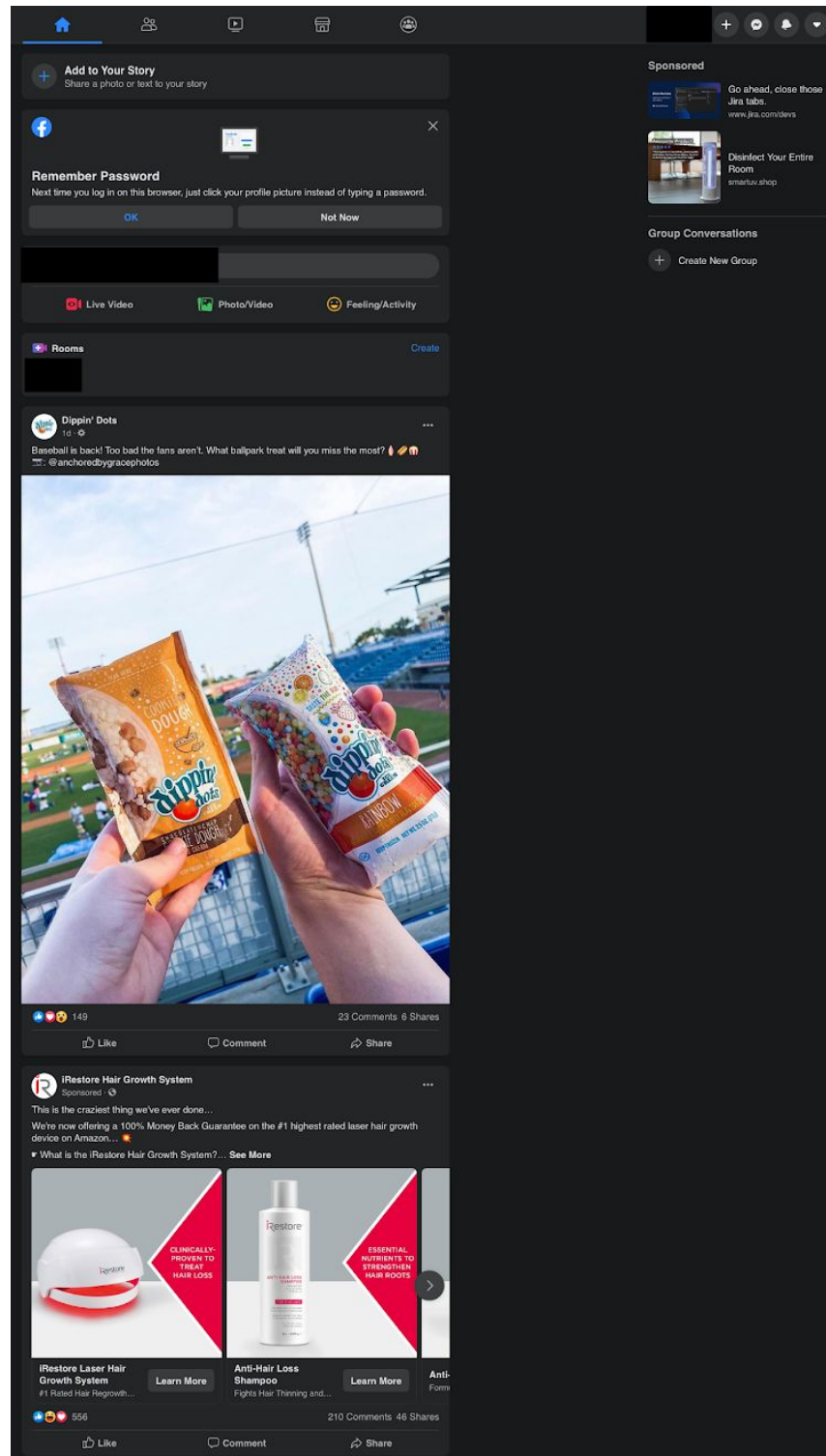


MMI video showing an automated browser attempting to reset an account password:

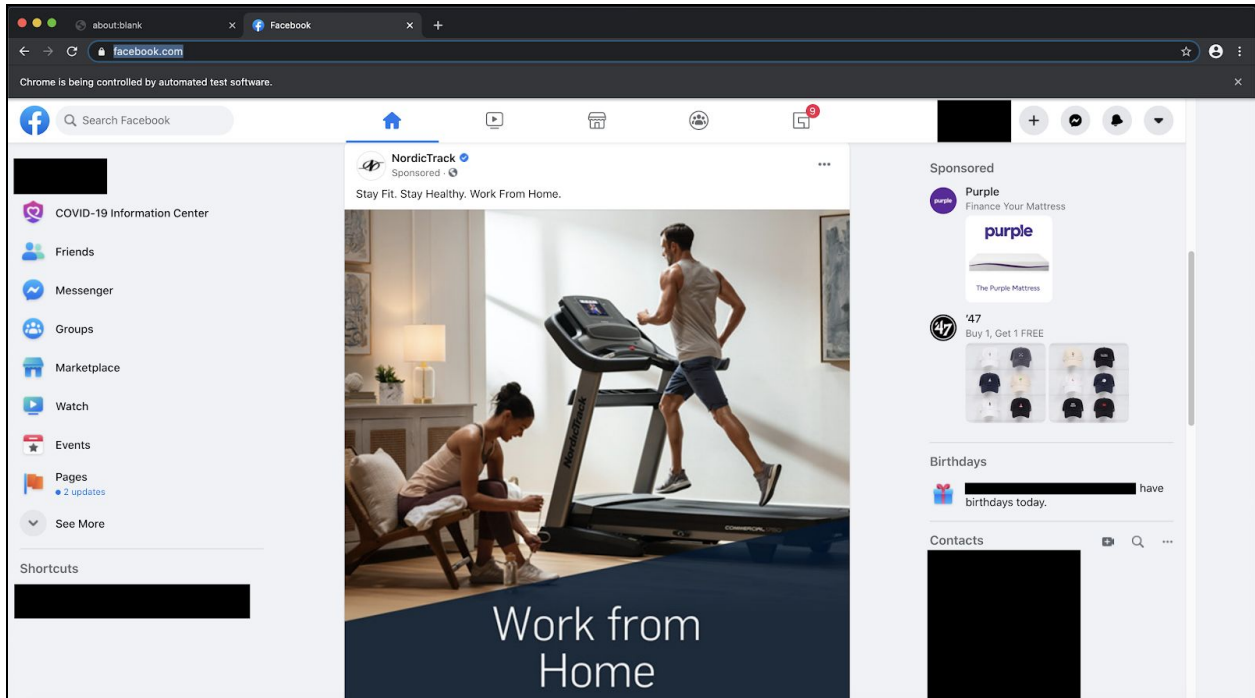
https://youtu.be/2FcfNjR3f_4

Facebook's practice of not blocking bot behavior is significant for the following reasons:

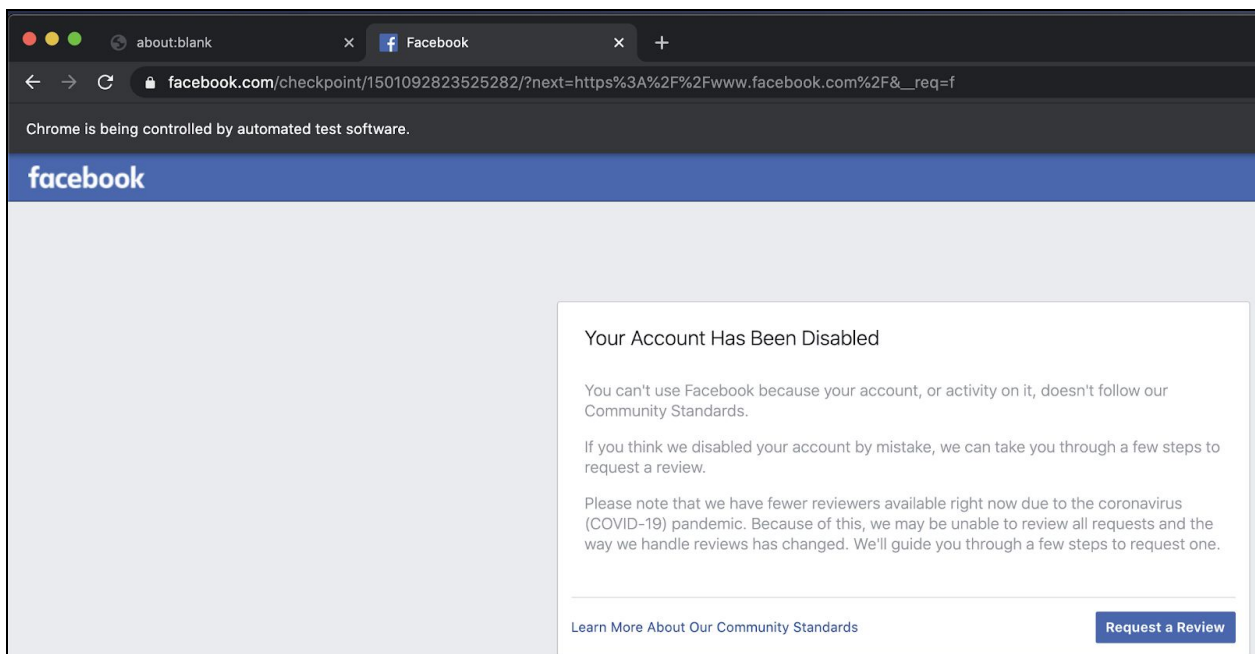
1. Facebook's user metrics have long been used in its financial statements. They are very likely inflated as a result of not blocking bot traffic.
2. Users can still gather data from the platform in an automated fashion. Even if this violates the terms of service, the data has already been exfiltrated.
3. Coordinated inauthentic behavior scales best when the activity is automated. Blocking or shadowbanning bots would greatly increase the hurdle to bad actors on their platform.
4. Facebook [bills](#) advertisers based on both ad impressions (views) or clicks. Because Facebook does not block bots from accessing the platform, Facebook's customers are paying for robotic traffic. The following screenshots and video support that conclusion.
5. Bots consume computing resources unnecessarily, increasing Facebook's server costs.



This screenshot was taken with a headless Chrome bot. Three ads are visible on the page.



A Chrome bot in automated, headful mode receiving multiple ads.



Facebook does prevent automated bots, including Puppeteer Stealth from registering new accounts on their platform. All of our attempts to use an automated browser to create a Facebook account were unsuccessful. This result shows that Facebook has researched bot

Method Media Intelligence

<https://methodmi.com>

info@methodmi.com



capabilities and has successfully implemented bot blocking methods on at least one part of its platform.

Corporate Landing Pages

Many companies try to block bots from accessing their site to protect their data, keep analytics clean, reduce IT costs, etc. We checked standard Puppeteer and stealth Puppeteer against 130 sites and found the following results:

	Indifferent	Failing	Succeeding	Sum
Count	107	17	6	130
Percentage	82.3%	13.1%	4.6%	-

82% of sites were indifferent to bots, they did not block standard or stealth Puppeteer. 13% of sites block standard Puppeteer but fail to detect and block stealth Puppeteer. 5% of sites successfully identified and block both standard and stealth Puppeteer. A full report on this is available [here](#).



Conclusion

Bot activity is a natural consequence of the increasingly complex capabilities of developer tools such as Puppeteer. The ease of use and low cost of cloud computing platforms like Amazon Web Services allow even inexperienced and bootstrapped programmers to quickly deploy bots at scale on the internet.

It is intuitive that web testing tools designed to improve developer productivity have the ability to replicate human action.

Insert sentence on how dev tools have many legitimate uses, but companies who are paying for human activity should be able to tell the difference.

However, as these bots become less distinguishable from human operated browsers, the difficulty of addressing them increases.

Comment on how ad platforms all claim no bot traffic is allowed on their platform, but they all show ads to bots. It makes more sense to prevent monetization of bot traffic than to allow it and reconcile it after the fact by providing a credit or refund to advertisers.

Possible comment on how Apple does not allow headless mode on Safari, which greatly reduces the possibility of monetization of robotic activity.

MMI provides bot management solutions to prevent undesired bot activity online. Our [patented](#) bot detection technology detects headless browsers/data center traffic and blocks their activity on the web. We use this technology to protect the integrity of digital media and other online assets.