# In Plain Sight

How Developer Tools Enable Invalid Traffic

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Summary</td>
<td>1</td>
</tr>
<tr>
<td>Puppeteer Features</td>
<td>2</td>
</tr>
<tr>
<td>Tools Used to Circumvent Detection</td>
<td>2</td>
</tr>
<tr>
<td>Puppeteer Stealth</td>
<td>2</td>
</tr>
<tr>
<td>Developer Discussions</td>
<td>4</td>
</tr>
<tr>
<td>Captcha Solvers</td>
<td>5</td>
</tr>
<tr>
<td>Playwright</td>
<td>8</td>
</tr>
<tr>
<td>Amazon Elastic Compute Cloud</td>
<td>8</td>
</tr>
<tr>
<td>No-Code Browser Automation Tools</td>
<td>9</td>
</tr>
<tr>
<td>Companies' Responses are Insufficient</td>
<td>11</td>
</tr>
<tr>
<td>Google/YouTube</td>
<td>11</td>
</tr>
<tr>
<td>Google, Bing, and Yahoo Search</td>
<td>12</td>
</tr>
<tr>
<td>Facebook</td>
<td>14</td>
</tr>
<tr>
<td>Corporate Landing Pages</td>
<td>18</td>
</tr>
<tr>
<td>Conclusion</td>
<td>18</td>
</tr>
<tr>
<td>About MMI</td>
<td>19</td>
</tr>
<tr>
<td>References</td>
<td>20</td>
</tr>
</tbody>
</table>

Method Media Intelligence
[https://methodmi.com](https://methodmi.com)
Contact: [info@methodmi.com](mailto:info@methodmi.com)
Executive Summary

Bot traffic is any online activity performed by an automated process rather than human action. Bots are often associated with cybercriminals and malware, but our research confirms 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 assumption 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. Benign bots use these technologies to monitor websites, aggregate online content, and automate repetitive tasks. But the same technology is also used to simulate human activity online at the expense of large and small businesses. We are observing a rise in add-on tools specifically designed to evade bot detection. These sophisticated bots can emulate human behaviors and signals, are available free of cost, and can be scaled up inexpensively. While the case can be made for bot creation for testing purposes, it is hard to justify detection evasion being necessary in most standard testing workflows.

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. The intention of this report is not to shame individual advertising platforms or brands, but to demonstrate that the technological and economic barriers to commit ad fraud are very low.
Puppeteer Features

Puppeteer is a framework developed by Google that enables automated control over a Chrome or Firefox browser. Puppeteer has pre-built features which can emulate dozens of human characteristics. Google has provided detailed documentation for all Puppeteer features online.

Puppeteer capabilities include:

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. Many demonstrations and code samples are available online that enable developers to quickly construct functioning bots.

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.

<table>
<thead>
<tr>
<th>package</th>
<th>downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>puppeteer-extra-plugin-stealth</td>
<td>5,015,808</td>
</tr>
</tbody>
</table>

Puppeteer Stealth has been downloaded more than 5 million since inception. (Source)

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
Implementation of this plugin to an existing Puppeteer script requires installing the plugin and adding only 2 lines of code:

```javascript
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](https://methodmi.com).

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Agent</td>
<td>Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15.5) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/83.0.4103.60 Safari/537.36</td>
</tr>
<tr>
<td>WebDriver</td>
<td>present (faked)</td>
</tr>
<tr>
<td>WebDriver Advanced</td>
<td>passed</td>
</tr>
<tr>
<td>Chrome</td>
<td>missing (faked)</td>
</tr>
<tr>
<td>Permissions</td>
<td>prompt</td>
</tr>
<tr>
<td>Plugins Length</td>
<td>0</td>
</tr>
<tr>
<td>Plugins has type PluginArray</td>
<td>failed</td>
</tr>
<tr>
<td>Languages</td>
<td>en-US</td>
</tr>
<tr>
<td>WebGL Vendor</td>
<td>Google Inc.</td>
</tr>
<tr>
<td>WebGL Renderer</td>
<td>Google WebGL/Shader</td>
</tr>
<tr>
<td>Harware Feature</td>
<td>missing</td>
</tr>
<tr>
<td>Broken Image Dimensions</td>
<td>60 pixels</td>
</tr>
</tbody>
</table>

Screenshot taken with standard Puppeteer bot.
Upon learning of the existence of Puppeteer Stealth, MMI updated its 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 browser is working as intended, and that the existence of Puppeteer Stealth does not require action on their part.

Developer Discussions

Puppeteer Stealth is a 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.

Developer comments on the power of bad actors using Puppeteer on HackerNews.

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
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. The developers and contributors to this plugin work voluntarily to improve the availability of sophisticated evasion techniques. This necessitates a correspondingly dedicated effort to continue detecting these bots.

How to give fake GPU info to site

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 question 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. The user received a detailed description of how to solve the problem from a volunteer.

Captcha Solvers

Captcha solvers are a common method used to validate if a browser is controlled by a human or a bot. However, incorporating a Captcha solving feature into a browser bot is a solved problem and is relatively inexpensive.
Example of a Captcha on Google Search.

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. "\_(ツ)_/"  
The lead developer’s reasoning and comments on the plugin they created.

Method Media Intelligence  
https://methodmi.com  
Contact: info@methodmi.com
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.

<table>
<thead>
<tr>
<th>Rate</th>
<th>Solving speed</th>
<th>Service load</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.77</td>
<td>20 s</td>
<td>35%</td>
</tr>
<tr>
<td>$2.99</td>
<td>46 s</td>
<td>57%</td>
</tr>
</tbody>
</table>

Pricing for two of the common vendors is:
2captcha - $0.77 / 1,000 Captchas, $2.99 / 1,000 ReCaptchas. Over 2,000 workers online.
Deathbycaptcha.com - $1.39 / 1,000 Captchas

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
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.

Puppeteer is still far more popular than Playwright, with 1.5M weekly downloads vs. 65k for Playwright. An equivalent “stealth add-on” is in progress for Playwright from the same developers as the Puppeteer version. 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 them. 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](https://methodmi.com). 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.
No-Code Browser Automation Tools

Tools like Puppeteer require the execution of JavaScript code and a basic understanding of computer programming. More recently, no-code browser automation tools have been developed which allow those with little to no programming experience to create browser bots. Examples include Axiom, TestCraft, Usetrace, and Perfecto.

Axiom enables the configuration of automated browsers without writing any code. Their product is a Chrome extension that prompts the user for step-by-step instructions for the browser to carry out. Axiom is intended for auto filling forms and web scraping. A free tool like this significantly lowers the barrier to non-technical users running headless browsers.

Axiom has the option to run the browser in headless mode.

Axiom launches a Chromium browser which performs the specified automated task. Currently the browser does not attempt to hide that it is a bot, as it does not rewrite its self-identifying characteristics (user-agent, etc.). Unsophisticated detection solutions can detect this type of non-malicious bot. Currently Axiom runs only on a local device and cannot be used to run a large scale bot operation in a data center environment.

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
Axiom has pre-built functionality to automate social media activity.

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
Companies’ Responses are Insufficient

Google/YouTube

In some cases, Google’s own ad platform serves their own ads to their own automated browser. This demonstrates a clear gap in control of the digital advertising environment. It is clear that their ad platform cannot detect all implementations of automated Chrome.

A Google Fi advertisement served to an automated Chrome browser on iplocation.com.

Similar results are visible on YouTube, where automated Chrome browsers are presented with ads.

Method Media Intelligence
https://methodmi.com
Contact: 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.

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. This result holds when using Playwright (Microsoft) on Bing.

Bots can click through these ads to reach advertisers’ websites.

We created search ad campaigns on Google AdWords, Bing, and Yahoo and sent our own bot to click on our ads. In each case, we were able to consume hundreds of dollars of budget in minutes with a bot. Clicks from the even the least sophisticated Puppeteer and Playwright bots are not filtered from Google, Bing, and Yahoo search ads billing.

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.

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
Examples of search engines showing ads to automated browsers.
Facebook

Facebook’s terms of service prohibits product access or collection of data through automated means.

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
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

MMI video showing an automated browser attempting to reset an account password:  
https://youtu.be/2FcfnJjR3f_4

Method Media Intelligence  
https://methodmi.com  
Contact: info@methodmi.com
This screenshot was taken with a headless Chrome bot. Three ads are visible on the page.
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 capabilities and has successfully implemented bot blocking methods on at least one part of its platform.

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
We shared this information with Facebook’s security team. Their response was that they are aware of the characteristics of automated browsers, and have a mechanism in place to catch this if it were done as a mass exploit.

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:

<table>
<thead>
<tr>
<th></th>
<th>Indifferent</th>
<th>Failing</th>
<th>Succeeding</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>107</td>
<td>17</td>
<td>6</td>
<td>130</td>
</tr>
<tr>
<td>Percentage</td>
<td>82.3%</td>
<td>13.1%</td>
<td>4.6%</td>
<td>-</td>
</tr>
</tbody>
</table>

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](https://methodmi.com).

* As of August 21st, Puppeteer Stealth has been updated with new evasions and now passes detection on all 130 sites listed in the report linked above.

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 huge scale on the internet.

It is intuitive that web testing tools designed to improve developer productivity will have the ability to replicate human action. These tools have many legitimate uses that reduce the need for repetitive tasks.

Given the widespread nature of online bots, companies purchasing human attention online are at high risk of wasting their budgets on invalid traffic. Ad platforms often claim no bot traffic is allowed on their platform, but evidence shows their measures to prevent the monetization of invalid traffic are ineffective. The industry narrative is that ad fraud is declining, but as bots become less distinguishable from human operated browsers, the difficulty of addressing will continue to increase.

Method Media Intelligence
[https://methodmi.com](https://methodmi.com)
Contact: [info@methodmi.com](mailto:info@methodmi.com)
About MMI

Method Media Intelligence provides solutions to prevent undesired bot activity online. Founded by Shailin Dhar and Praneet Sharma in 2017, we verify the integrity of digital media and other online assets. Our patented bot detection technology detects headless browsers and data center traffic activity on the web. We are based in New York City with offices in the San Francisco Bay Area, and London. MMI is backed by General Motors Ventures.

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
References

1. Official Puppeteer documentation: https://pptr.dev/
2. List of Puppeteer demonstrations of browser bots: https://www.youtube.com/results?search_query=puppeteer+browser+demo
3. Code samples of Puppeteer bots: https://github.com/checkly/puppeteer-examples
7. One of many tutorials on how to avoid detection with Puppeteer: https://javascriptwebscrapingguy.com/avoid-being-blocked-with-puppeteer/
8. A demonstration of what common bot detection methods are used, frequently used to assess the effectiveness of evasion techniques: https://bot.sannysoft.com/
11. Stackoverflow user getting help on how to add evasion capabilities to a browser bot: https://stackoverflow.com/questions/55955203/how-to-give-fake-gpu-info-to-site
12. Youtube tutorial on how to integrate Captcha solving capabilities into a Puppeteer bot: https://www.youtube.com/watch?v=wsDRkAD6lPs
15. 2Captcha charges ~$0.77 to solve 1,000 Captchas: https://2captcha.com/public_statistics
16. Playwright browser automation framework from Microsoft: https://github.com/microsoft/playwright
17. Progress toward a stealth add on for Playwright: https://github.com/berstend/puppeteer-extra/pull/303
18. AWS Elastic Compute Cloud is used to rent servers for high performance computing: https://aws.amazon.com/ec2/
19. Google Compute is the equivalent of EC2 for Google Cloud: https://cloud.google.com/docs/compare/aws/compute
20. Microsoft Virtual Machines is the equivalent of EC2 for Microsoft Azure: https://azure.microsoft.com/en-us/services/virtual-machines/
21. EC2 offers pre-configured instances with browser automation software pre-installed: https://aws.amazon.com/marketplace/pp/Michael-Fleck-Selenium-Webdriver-on-Headless-Ubunt/B07NVNJF63
22. The cost of renting a server equivalent to a high-end laptop is $0.09/hr on Amazon EC2: https://aws.amazon.com/ec2/spot/pricing/

Method Media Intelligence
https://methodmi.com
Contact: info@methodmi.com
23. More on headless browsers, which run in the background and do not display on a screen:
   https://en.wikipedia.org/wiki/Headless_browser
24. MMI demonstration video on creating web traffic from Amazon EC2 and Puppeteer:
   https://youtu.be/og2KhwlhxzA
25. Luminati provides access to residential IPs from around the world, enabling bot users to mask the
origin of their activity: https://luminati.io/proxy-types/static-residential-ips
26. Axiom, a no-code browser automation tool that allows users with no programming knowledge to
create bots: https://axiom.ai/
27. Google, Bing, and Yahoo are 96% of the worldwide search engine market:
   https://gs.statcounter.com/search-engine-market-share
28. Facebook’s Terms of Service prohibits the use of their products by automated means:
   https://www.facebook.com/terms.php
29. The Navigator.webdriver was intended to be used to identify bots on the web. However, it can be
easily overwritten by bots: https://developer.mozilla.org/en-US/docs/Web/API/Navigator/webdriver
30. MMI demonstration video showing a bot logging into Facebook and viewing ads nonstop:
    https://www.youtube.com/watch?v=5OteU0s8gps
31. MMI demonstration video showing a bot attempting 21 password reset codes on a Facebook
account: https://youtu.be/2FcfNjR3f_4
32. Facebook documentation stating that advertisers can be billed for both ad views (CPM) and ad
33. MMI report on how differently bot traffic is handled by a variety of corporate websites:
    https://docs.google.com/document/d/1BnumC-IknCLxaTQUM_SuYsCttEq20X_imncPWoDKUWy/edit?usp=sharing
34. The regular update frequency of Puppeteer stealth means bot detection solutions will constantly
    need to adapt:
    https://github.com/berstend/puppeteer-extra/commits/master/packages/puppeteer-extra-plugin-stealth
35. Link to Method Media Intelligence website: https://www.methodmi.com/