IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

# Anti-Scraping Techniques

**Shubham Sandeep Zujam[1] and Prof. Bhanudas Satam[2]**
Student, Department of MCA[1]
Mentor, Department of MCA[2]
Late Bhausaheb Hiray S. S. Trust's Institute of Computer Application, Mumbai, India

**Abstract:** *Web scraping refers to a software program that mimics human web surfing behavior by pointing to a website and collecting large amounts of data that would otherwise be difficult for a human to extract. A typical program will extract both unstructured and semi-structured data, as well as images, and convert the data into a structured format. The activity is deemed illegal, but the change in legality has not stopped people from doing the same. This paper aims to list challenges and proposes techniques to developing anti-scraping application.*

**Keywords:** Web scraping

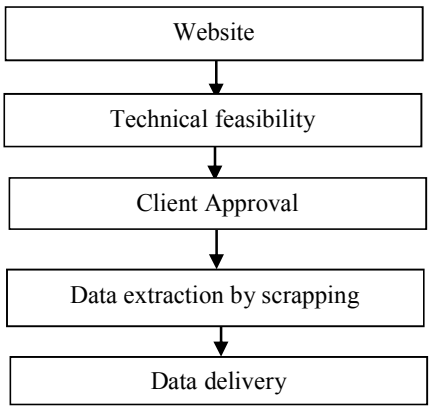
## I. INTRODUCTION

Web scraping is not initially developed for research of social science, as a effect, analysts using this method may incorporate unknown suppositions into their own, because web scraping will not usually require direct contact among the analyst and those who were formerly collecting the information and inserting it online, data analysis issues may simply arise. Research teams using web scraping techniques as an information gathering method still have to be acquainted with the accuracy and correct analysis of the details retrieved from the website. The major challenges that are faced by a Software as a Product (SaaP) anti-scraping application are mostly related to its environment: the fact that it shares the same resources as the website itself. The database server is the same for the application and the website and so is the hosting server. Subsequently, the techniques used by the SaaP model are subject to the user's server capabilities. Added to this is the fact that the website is a data-rich one, and thus, the database is already under a lot of load. This paper aims to shed light on the restrictions faced by anti-scraping applications and offers some ways to mitigate their effect on the same.

### 1.1 What is Web Scraping?

Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. However, most web pages are designed for human end-users and not for ease of automated use. Because of this, tool kits that scrape web content were created. A web scraper is an API or tool to extract data from a web site. Companies like Amazon AWS and Google provide web scraping tools, services and public data available free of cost to end users. Newer forms of web scraping involve listening to data feeds from web servers. For example, JSON is commonly used as a transport storage mechanism between the client and the web server.

## II. DATA FLOW DIAGRAM





DOI: 10.48175/IJARSCT-5419

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

## II. ANTI-SCRAPING TECHNIQUES

### 2.1 Login

Many websites, especially social media platforms only show you information after you log in to the website. In order to crawl sites like these, the crawlers would need to simulate the logging steps as well. After logging into the website, the crawler needs to save the cookies. A cookie is a small piece of data that stores browsing data for users. Without the cookies, the website would forget that you have already logged in and would ask you to log in again. Moreover, some websites with strict scraping mechanisms may only allow partial access to the data, such as 1000 lines of data every day even after log-in.

You need to know how to log-in:

1. Simulate keyboard and mouse operations. The crawler should simulate the log-in process, which includes steps like clicking the text box and "log in" buttons with the mouse or typing in account and password info with the keyboard.
2. Log in first and then save the cookies. For websites that allow cookies, they would remember the users by saving their cookies. With these cookies, there is no need to log in again to the website in the short term. Thanks to this mechanism, your crawler could avoid tedious login steps and scrape the information you need.
3. If you, unfortunately, encounter the above strict scaping mechanisms, you could schedule your crawler to monitor the website at a fixed frequency, like once a day. Schedule the crawler to scrape the newest 1000 lines of data in periods and accumulate the newest data.

### 2.2 IP

One of the easiest ways for a website to detect web scraping activities is through IP tracking. The website could identify whether the IP is a robot based on its behaviors. when a website finds out that an overwhelming number of requests had been sent from one single IP address periodically or within a short period of time**,** there is a good chance the IP would be blocked because it is suspected to be a bot. In this case, what really matters for building an anti-scraping crawler is the number and frequency of visits per unit of time. Here are some scenarios you may encounter.

**Case #1**: Making multiple visits within seconds. There's no way a real human can browse that fast. So, if your crawler sends frequent requests to a website, the website would definitely block the IP for identifying it as a robot.
**Solution**: Slow down the scraping speed. Setting up a delay time (e.g. "sleep" function) before executing or increasing the waiting time between two steps would always work.

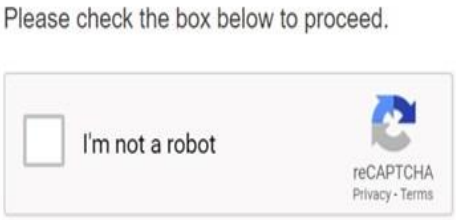
**Case #2**: Visiting a website at the exact same pace. Real human does not repeat the same behavioral patterns over and over again. Some websites monitor the request frequency and if the requests are sent periodically with the exact same pattern, like once per second, the anti-scraping mechanism would very likely be activated.
**Solution**: Set a random delay time for every step of your crawler. With a random scraping speed, the crawler would behave more like how humans browse a website.

**Case #3**: Some high-level anti-scraping techniques would incorporate complex algorithms to track the requests from different IPs and analyze their average requests. If the request of an IP is unusual, such as sending the same amount of requests or visiting the same website at the same time every day, it would be blocked.
**Solution**: Change your IP periodically. Most VPN services, cloud servers, and proxy services could provide rotated IPs. When requests are being sent through these rotated IPs, the crawler behaves less like a bot, which could decrease the risk of being blocked.

1. Black list: list of IPs to block
2. Gray list : IPs which might be scrapers and have shown suspicious activity. IPs in this list have to, for some days, repeatedly solve CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) every time they start a session on the site.
3. White list: normal IPs/IP sources having legitimate traffic.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

### 2.3 Interval Analysis




Another method is interval analysis which is a long- term strategy. If the interval of visits is very similar, it results in the gray-listing of the traffic. As is already noted earlier, transition from gray-lists to any other list takes place through bot-differentiating techniques. Popular examples of the same include the various forms of CAPTCHA.
CAPTCHA tests are based on open and hard problems in AI , which include recognition of highly distorted images, answering problems that require semantic knowledge , without giving a learning set to the bot. CAPTCHA itself has been developed into many variants by various individuals and organizations like Google and Microsoft . reCAPTCHA aims to convert images of old books and manuscripts to text alongside a normal CAPTCHA image etc.

### 2.4 Regular Markup Changes

The markup of a site can be regularly changed in an automated manner, although not in a very major way. Most web scrapers rely on class names and id of various tags to identify the section to scrape. Automated regular changing of just the class and id names of the various sections will prove to be very effective in breaking some scraping bots. However, changing the position of various sections might prove rather annoying to actual users who are used to a particular layout and this might result in decrease of genuine traffic on the site. A pseudocode for markup randomizer is given in Listing 1 while entire process is shown.

### 2.5 Information as an Image

Another method to prevent scraping is to present information as part of images, and not raw text. Normal text information can be converted to images on the server-side and then can be presented to the user. This force scrapers to use Optical Character Recognition(OCR) software to actually get the text from the images. However, this method is not very effective as nowadays, OCR libraries have become very advanced. Deliberate smudging of text in the image or otherwise will lead to annoyance from the genuine users and might affect the user experience on the site. Data can also be presented as flash snippets or HTML5 canvas instead of images.

### 2.6 Frequency Analysis

One of the most important levels of defense against scraping traffic is checking for frequency of the visits. This is a rather medium-term technique, as the average hits per user has to be determined first (or has to be supplied to the Anti-Scraping Application periodically). Some Anti-Scraping Applications use bell curves to determine hits that are over the top. Again, multiple lists such as gray list, blacklist and white list are used.
Before any kind of analysis, hits need to be recorded for some time, around four to seven days, depending on the user. Multiple variations of this technique can exist. One could be to gray list traffic the hits of which per day exist substantially more than the average normal traffic for a day. The benchmarks can be designed to vary. The period could also vary: just take the average number of hits per day over a month; or refer to different averages on different week days, in order to lessen the number of false positives. Another thing that follows immediately is that we have to check for highly substantial differences from the mode of the data (i.e. hits data) range in question, rather than the average, because a particular traffic source recording a large number of hits on a particular time interval, and lesser traffic in the other equivalent time intervals is likely to be not a scrape-bot.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

### 2.7 User Agent

It is a header for the website to identify how the user visits. It contains information such as the operating system and its version, CPU type, browser, and its version, browser language, a browser plug-in, etc.

An example UA: *Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.56 Safari/535.11*

When scraping a website, if your crawler contains no headers, it would only identify itself as a script (e.g. if using python to build the crawler, it would state itself like a python script). Websites would definitely block the request from a script. In this case, the crawler has to pretend itself as a browser with a UA header so that websites could provide access to it.

Sometimes website shows different pages or information to different browsers or different versions even if you enter the site with the same URL. Chances are the information that is compatible with one browser while the other browsers are blocked. Therefore, to make sure you can get into the right page, multiple browsers and versions would be required.

**Switch between different UA's to avoid getting blocked**. Change UA information until you find the right one.

Some sensitive websites which apply complex anti-scraping techniques may even block access if using the same UA for a long time. In this case, you would need to change the UA information periodically.

### 2.8 URL Analysis

The analysis of URLs visited could also be used, wherein, the application checks for the pages which the traffic visits. If it is only the data-rich pages, the URLs of which are provided by the user i.e., the person in charge of the website, the application takes-action after a particular benchmark defined by the user is crossed.

Of course, the ratio of number of non-data rich pages to the ratio of data-rich pages in the website needs to be substantially high for this method to work properly.

If this ratio is low, this method is going to throw up a whole lot of false positives. As is evident, this needs a lot of database storage space, plus, it is generally CPU intensive, based on the number of average hits per user per time interval. This, again, is a medium term (bimonthly, usually) technique. Also, an analysis of the order in which pages are visited is a possible modification to this method. When the same order is repeated over a long period of time, it may also point out the existence of a scraper bot. Modified data mining algorithms like Apriori-algorithm , and other frequent-itemset and association rule mining algorithms like ECLAT can be implemented here, provided that the temporal data(time) of the requests is also stored.

### 2.9 AJAX

Today more websites are developed with AJAX instead of traditional web development techniques. AJAX stands for *Asynchronous JavaScript and XML*, which is a technique to update the website asynchronously. Briefly speaking, the whole website doesn't need to reload when only small changes take place inside the page.

So how could you know whether a website applies AJAX?

- **A website without AJAX**: The whole page would be refreshed even if you only make a small change on the website. Usually, a loading sign would appear, and the URL would change. For these websites, we could take advantage of the mechanism and try to find the pattern of how the URLs would change. Then you could generate URLs in batches and directly extract information through these URLs instead of teaching your crawler how to navigate websites like humans.
- **A website with AJAX**: Only the place you click will be changed and no loading sign would appear. Usually, the web URL would not change so the crawler has to deal with it in a straightforward way.

For some complex websites developed by AJAX, special techniques would be needed to find out unique encrypted ways on those websites and extract the encrypted data. Solving this problem could be time-consuming because the encrypted ways vary on different pages. If you could find a browser with build-in JS operations, then it could automatically decrypt the website and extract data.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

### 2.10 Honeynets

Some companies which were already in a networking- related business for example, Amazon Cloud Front and Cloud Flare uses a network of honeypots, also known as honeynets or honey farms around the world to capture bot information, and relay this information to their applications around the world via regular updates or other methods. Such honeypots are traditionally different from the normal ones in the fact that they mostly compromise of some seemingly highly data-rich pages. However, they are effective mostly when the scraper is not limited to a niche. Small honeypot-pages can be deployed on the website for detecting completely automated crawler-scrapers too.

## III. CONCLUSION

The popularity of web scraping is growing at such an accelerated pace these days. Nowadays not everyone has technical knowledge of web scraping and they use APIs like news API to fetch news, blog APIs to fetch blog-related data, etc.

As web scraping is growing, it would be almost impossible not to get cross answers when the big question arises: is it legal?

Using a web scraper to collect data from the Internet is not a criminal act in and of itself. Many times, scraping a website is perfectly legal, but the way you intend to use that data may be illegal.

There are many anti-scraping solutions available from many vendors. However, the small to medium scale websites have smaller and less valuable data sets, and therefore can implement an effective anti-scraping measures locally without buying any professional anti-scraping solution. On such sites the scrapers realizes that the extra effort for data theft outweighs the value of the scraped data. This leads to reduction in site scraping attack on those sites. There is a fundamental working principle of the security world at work here - the cost of getting a piece of data, or getting access to a certain environment should be higher than the perceived value of the prize.

This paper presents the survey of Anti-Web scraping technology incorporating what it is, how it works, the popular tools and technologies of Anti-web scraping.

## ACKNOWLEDGEMENT

We would like to acknowledge the University of Mumbai, Mumbai, India to give us the opportunity to do the research work under the title “Docker and how Micro-Services interacts: Case Study”. Also, we would like to acknowledge the college L.B.H.S.S. T’s ICA Bandra East, Mumbai, India to support us during the research process.

## REFERENCES

**[1].** Renita Crystal Pereira and Vanitha T, “Web Scraping of Social Networks,” International Journal of Innovative Research in Computer and Communication Engineering, pp. 237-240, Vol. 3, 2015.

**[2].** Kaushal Parikh, Dilip Singh, Dinesh Yadav and Mansingh Rathod, “Detection of web scraping using machine learning,” Open access international journal of Science and Engineering, pp.114-118, Vol. 3, 2018.

**[3].** Sameer Padghan, Satish Chigle and Rahul Handoo, “Web Scraping-Data Extraction Using Java Application and Visual Basics Macros,” Journal of Advances and Scholarly Researches in Allied Education, pp. 691-695, Vol.15, 2018.

**[4].** T. W.Bell, “Internet law,” http://www.tomwbell.com/NetLaw/ Ch06/eBay.html, [Online; accessed 15-Feb-2015].

**[5].** P. Vixie, “cron - daemon to execute scheduled commands (vixie cron),” Unix and Linux Forums, 2010, [Online; accessed 15- Feb-2015]. [Online]. Available: http://www.unix.com/man-page/ linux/8/cron/

**[6].** Wikipedia, “Discrete fourier transform — wikipedia, the free encyclopedia,” http://en.wikipedia.org/w/index.php?title=Discrete Fourier transform&oldid=650317831, 2015, [Online; accessed 15-March-2015].

**[7].** R. Agrawal, T. Imielinski, and A. Swami, “Mining association ´ rules between sets of items in large databases,” in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD ’93. New York, NY, USA: ACM, 1993, pp. 207–216. [Online]. Available: http://doi.acm.org/10.1145/170035.170072