2022 11th International Conference of Information and Communication Technology (ICTech) | 978-1-6654-9694-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTECH55460.2022.00070

Exploring The Role of Web Crawler and Anti-Crawler Technology in Big Data Era

Fan Zhou[1,*]
Urban Vocational College of Sichuan,
Department of information technology,
Chengdu, Sichuan 610101
e-mail: 563822964@qq.com

Yang Wang[2]
Luzhou Vocational and Technical College
School of Artificial Intelligence and Big Data
Luzhou, Sichuan 646000
e-mail: 1213342548@qq.com

***Abstract*—In the era of big data, with lower costs and higher efficiency, web crawlers access resources and information from the Internet, bringing a lot of convenience to businesses and individuals. Nevertheless, there are two sides to everything, as malicious crawlers bring incalculable threats and losses to websites. In order to prevent web crawlers from being abused or even developing into malicious crawlers, websites usually perform anti-crawler based on techniques such as ip access frequency, browsing page speed, account login, input captcha, js encryption, ajax obfuscation, etc. Anti-crawlers cannot completely block crawlers with a particular technique, but only find ways to increase the cost of crawling for attackers, forcing the catching party to make the right choice after weighing the cost-benefit.**

Keywords-web crawler; anti-crawler; big data; network; attackers

I. INTRODUCTION

With Big Data becoming a trend and a national strategy, research on Big Data has been highly concerned by the industry, academia and even the government. As we all know, there are huge opportunities and values hidden in big data in all industries, making it the new "oil" of the information age. The full exploitation of the potential value of data can be of great help to individuals in their work and life, and even to enterprises in their future development and innovation models. Data can enable people to better grasp market trends, better respond to the market and generate new and rational decisions. Then, where does the data come from? One of the most efficient ways to get data is from the internet. However, manually extracting data from the web is obviously inefficient and difficult to meet the diverse data needs. In order to provide people and businesses with fast access to comprehensive, valid and accurate data, web crawler technology has been developed.

II. WHAT IS WEB CRAWLER TECHNOLOGY IN THE AGE OF BIG DATA

What is web crawler technology and what is its meaning and role? A web crawler is an automated program that automatically browses the web for information, downloads web pages from the internet and extracts the relevant data. There is no difference in nature to a human clicking on a web connection and visiting a web page to obtain data. It is just like an intelligent robot that can automatically simulate a normal human initiating a web request and then acquire the data returned by the web request so as to download the content of the website. Its main task is to download the target web page and parse the information from the web page. The basic process in Figure 1: First send a request to the server through the url, and then the server responds. The server receives Html,Json and binary data including video, image, etc. Then the server parses the data. If it is Html code, the web page parser is used to parse the data. If it is Json data, it is converted to the Json object for parsing. If the data is binary, it can be saved to a file for further processing. Finally, the data can be saved to a local or database.

Figure 1. Basic flow chart of crawler

III. CLASSIFICATION AND DESIGN OF WEB CRAWLER TECHNOLOGY

According to the system architecture and implementation techniques, there are several types of web crawlers: generic web crawlers, focused web crawlers, incremental web crawlers and deep page crawlers[1]. The design of real-life crawler systems is usually a combination of several crawler technologies and is usually implemented using a suitable crawler framework[2]. In Python, for example, the main common crawler frameworks are Scrapy framework, Pyspider framework, Cola framework and so on. In terms of the actual scale of exploitation, large-scale data web crawlers mainly crawl the whole network data, the search engine and crawl speed requirements are high. It generally requires custom development to build web-wide search engines (e.g. Baidu, Google search, etc.) and collects data mainly for portal sites, search engines and large web service providers. Medium-scale data web crawler: with a large data volume, mainly crawling those websites or series of websites data related to pre-defined topics, sensitive to crawling speed, this type of web crawler is usually implemented using the Scrapy library. For small-scale data web crawlers, which have a smaller data volume, mainly crawl newly generated or updated pages, and do not re-download pages that have not changed, and are not sensitive to crawl speed, this type of web crawler is usually implemented using the Requests library.

978-1-6654-9694-0/22/$31.00 ©2022 IEEE
DOI 10.1109/ICTech55460.2022.00070

Authorized licensed use limited to: National PingTung University -. Downloaded on September 13,2022 at 08:31:17 UTC from IEEE Xplore. Restrictions apply.

IV. RESEARCH STATUS AND ANALYSIS OF WEB CRAWLER

A. Bottleneck analysis of network crawler efficiency

The main factors that restrict the efficiency of web crawler are network delay and crawler operation efficiency[3], poor design of crawler system function module, low cooperation efficiency between crawler algorithm and function module, adaptability difference of Web server, etc.

B. Dynamic web page information crawling

Dynamic web pages are generated by updating the website database and passing parameters on the server. Crawler can analyze dynamic web page, process information in web page data, create index database, redefine custom standard interface, and judge web page URL address before crawler captures web page. If it is determined that the dynamic web page meets the custom standard interface, the crawler can download the web page through HTTPS, create and import the database.

C. Update web page

Update by determining whether web page properties were changed when updating the Web database. JavaScript can dynamically add, delete, validate, and change the properties of all objects at any time. You can update captured web page data directly without modifying crawler code.

D. Implementation of JavaScript algorithm

The JavaScript language is an object-based programming language. But JavaScript differs from other object-oriented languages，It has only object probabilities, not classes. It comes from its own internal objects, objects in the host environment, and objects created by users. Crawler builds the object layer, method layer and statement layer of JavaScript program to make use of the data correlation between statements layer by layer. Use functions to control the left and right values of assignment statements that control global variables in JavaScript programs to participate in the influence of statement predicates and object polymorphic inheritance. By using JavaScript to dynamically define objects, the integrated encapsulation of web data is realized.

E. Benefits and potential problems of crawlers

The crawler automatically obtains data from the Internet, and with this data we can do many things. Companies can improve their products and innovate their models by crawling data to analyse user behaviour, the shortcomings of their own products, information about their competitors and so on[4]. Individuals can use crawlers to quickly access the information or resources they need, such as people who need some data support when doing projects, surveys, starting a business, writing papers, etc. They can use web crawlers to collect data from specific websites, thus greatly improving their learning and working efficiency[5].

Although crawler is efficient and fast, they bring a lot of convenience to individuals and businesses. However, there are two sides to everything, as the barrier to entry for web crawlers is low and can be handled by downloading open source crawlers, and the cost of making URL requests to obtain data through the program is also very low. This resulted in a large number of low-quality web crawlers running rampant, with malicious crawlers posing certain potential problems and security risks to security, back-end service and network operational [6].

Service: A large number of visits to the target website by web crawlers lead to huge resource overhead on the web server, which may affect the normal user access experience in a minor case, or lead to unavailability of website services in a major case, and may cause DDOS attacks in the most serious cases. Security: malicious crawlers may have the ability to break through simple access controls and access protected data, resulting in data asset leakage; free mass crawling of searchable resource data by rival companies for competitive analysis, resulting in loss of competitiveness of competing products; scanning of registered users by malicious crawlers; penetration testing, etc. Network operational: malicious crawlers cause click fraud, thus affecting the real interests of the website. Take advertising as an example, crawlers cause clicks that largely dilute the clicks of real users, making the click-through rate of advertising falsely high. This not only raises the cost of clicks on the website, but the unstable number of visits is also not conducive to analysis of the effectiveness of advertising placement, which invariably results in distorted operational statistics. The value of the business output is reduced; the econnoisseur greatly reduces the cost of getting the best deal through web crawlers, etc.

V. COMMON ANTI-CRAWLER TACTICS

Anti-crawler is a way to prevent attackers from obtaining their website information in batches by using certain technical means. Common crawler:

1. IP Restrictions: When the same IP is used to access the server several times frequently, the server will detect that the request is not a normal human action and may be a crawler operation. The administrator can write IP restrictions to prevent further access to the IP or even block the IP directly on the server, but an attacker can counter anti-crawlers by limiting the crawler's crawl frequency or proxy IP in Figure 2.

```
proxies = {"http": "http://42.228.3.155:8080",}
requests.get(url, proxies=proxies)
```

Figure 2. Requests setting proxy IP

2. Access control via the User-Agent: When a browser or crawler makes a web request to the server, it sends a header file containing a field in which the browser "identifies" itself to the server: headers. For crawlers, the most important field to pay attention to is: User-Agent. many websites will detect the User Agent in the header of HTTP requests, and when it is detected that the request header is the default some very

Authorized licensed use limited to: National PingTung University -. Downloaded on September 13,2022 at 08:31:17 UTC from IEEE Xplore. Restrictions apply.

obvious crawler header python-requests/2.18.4 etc., the backend administrator can directly deny access. However, an attacker can resist the anti-crawl by forging a compliant user-agent.

3. Honeypot technology: Deliberately leaving links on web pages that humans can't see or never click on, and that only crawlers are likely to visit. A site that finds an IP access to this link immediately and permanently blocks the IP + user-agent + Mac address and all other information that can be used to identify visitors. Of course, the crawler's crawler track is determined by the attacker, and the honeypot can be avoided by directional crawler.

4. SESSION access restrictions: SESSION is a more effective way of protecting data by requiring users to log in when access to more important or additional data is required. Once logged in, a count can be made in conjunction with the user's unique identifier. When the frequency and number of user visits reach a certain threshold, it can be judged as crawling behaviour and thus be intercepted to restrict the operation rights of the logged-in user. The attacker can register multiple accounts to simulate normal operations to resist the anti-crawler, but the difficulty factor is relatively high.

5. Verification by CAPTCHA：This is a mature but effective anti-crawler strategy.When a particular user has too many visits, the request is automatically made to jump to a verification code page, and access to the site can only continue after the correct code has been entered. captcha include image captcha, SMS captcha, calculation question image captcha, sliding captcha, pattern captcha, marker inverted text captcha, etc. As the computer cannot answer the CAPTCHA question, the user who answers it is considered to be human. CAPTCHA verification can be effective in preventing malicious password cracking, vote scraping, forum spamming, etc.

Now, many websites use third-party captcha services. When the user opens the login page of the target website, the verification code displayed on the login page is loaded from the link provided by a third party (such as Ali Cloud). At this time, we need to take an extra step to grab the verification code from the third-party link provided by the web page when simulating login, and this step often implies traps. Taking the verification code service provided by Ali Cloud as an example, the source code of the login page will show the third-party link provided by Ali Cloud, but when the link is matched for verification code capture, we will find that the verification code is invalid. After careful analysis of the captured request data, it is found that normal browsers will carry an extra TS parameter when they request the verification code. This parameter is generated by the current timestamp, but it is not a complete timestamp, but a string after the timestamp is rounded to nine digits.Of course, attackers can also find ways to access third-party CAPTCHA platforms at great cost to crack a website's CAPTCHA in real time, or use image recognition technology to identify the captcha text.

6. Javascript rendering: The web developer puts important information in the web page but does not write it in the html tags, and the browser automatically renders the js code in the <script> tags to present the information to the browser, while the crawler does not have the ability to execute the js code, so it cannot read out the information generated by the js events. However, an attacker could use Phantomjs to emulate a browser request to return a JS-rendered page to solve the problem.

7. Anti-crawler for dynamic pages: For dynamic web pages, the data we need to crawl is either obtained through Ajax requests or generated through JavaScript. First the network request is analyzed using Firebug or HttpFox. If you can find ajax requests and figure out what the parameters and responses mean, you can use Requests or URllib2 directly to simulate ajax requests and analyze the JSON of the response to get the data you want. It would be nice to be able to retrieve data directly from an Ajax request, but some sites encrypt all the parameters of an Ajax request. There is no way to construct a request for the data we need. In addition to encrypting Ajax parameters, there are some sites that are strictly guarded and encapsulate some basic functions, all calling their own interface, and the interface parameters are encrypted.

When encountering such a site, we can only invoke the browser kernel through the Selenium +phantomJS framework and execute JS using phantomJS to simulate human operations and trigger JS scripts in the page. From filling in the form to clicking the button to scrolling the page, all can be simulated, regardless of the specific request and response process, just a complete simulation of the process of browsing the page to obtain data. You can almost get around most anti-crawlers with this framework, because it's not masquerading as a browser to get data, it's a browser itself, and phantomJS is a browser with no interface, except that it's not controlled by a human.

VI. STRATEGIES FOR IMPROVING ANTI-CRAWLER TECHNOLOGY

A. Speed up the improvement of the front-end restriction programme

In the course of anti-crawler technology solution setting, the importance of front-end restrictions should be concluded, and the rational design of information organisation forms should be flexibly carried out through the use of CSS or HTML tags to improve the rationality of the configuration of technical resources related to front-end settings. The negative effects of factors such as element deviation should be strengthened, and relevant custom fonts should be reasonably applied, so that the construction of the anti-crawler technology solution can have an important positive impact on the front-end restrictions and further meet the innovative application requirements of anti-crawler technology.

In the process of developing the interference factors of anti-crawler strategy, it is very important to strengthen the research on key information (such as image migration), especially in order to regulate the negative effects of key data confusion and make better use of the front-end restriction scheme. The construction of the front-end restriction scheme needs to analyze the general means in the

Authorized licensed use limited to: National PingTung University -. Downloaded on September 13,2022 at 08:31:17 UTC from IEEE Xplore. Restrictions apply.

application process of anti-crawler technology and effectively sample the relevant web pages, so as to gradually improve the test means that affect the application quality of anti-crawler technology. The application of the front-end restriction scheme will have a more positive impact on data replacement. Help increase the value of the corresponding data information on the initial web page. The design of the front-end limitation scheme must also pay due attention to factors such as custom fonts, especially studying the source code characteristics on websites, in order to more accurately identify data offsets associated with CSS files. This facilitates innovative adjustments to web text rendering jobs. The design of the front-end restriction scheme also needs to focus on the application of TTF files and examine the authenticity of the information obtained by copying the Web source code.

B. Improve the reasonableness of the request rulemaking

During the design of anti-crawler technology, an increased focus should be placed on server-side features, with request limits set as the main focus to increase the demand for value exploitation of request rules and provide the necessary support for improvements in malicious data collection schemes. Request rules must be effectively investigated in the specific development process of the constituent features of the anti-crawler strategy, particularly through effective value verification of the various requestability information required by the web server to improve the demand for innovative applications of anti-crawler technology. The role of current request rules, particularly easily identifiable information such as python requests and User-Agents, must be effectively investigated for value during the analysis of attribute and configuration information to ensure that the value of important resources such as data packets is fully exploited to achieve mature and effective improvements to crawling tasks.

In the process of innovative design of request rules, it is very important to pay more attention to crawler coordination. If the site finds that the quality of the mock request header design is inadequate, it should implement a full understanding of the request header attributes. This helps in the efficient formulation of rules related to general access management measures and in the innovation of the main business such as setting properties.

C. Improving the maturity of data encryption technology applications

In exploring the application of data encryption technology, it is important to innovate the basic conditions required for anti-crawler technology, especially to study the information encryption requirements of the website, to fully clarify the data request situation in all aspects, and to meet the needs of the application value development of data resources.

VII. CONCLUSION

The key of crawler lies in batch, which is a way to obtain website information in batch by using any technical means. The key to anti-crawler is batch, which is a way to prevent others from obtaining their website information in batches by using any technical means. Error: In the process of anti-crawler, common users are wrongly identified as crawlers. An anti-crawler strategy with a high rate of friendly fire cannot be used, no matter how effective it is. Interception: Successfully blocking crawler access. However, the higher the interception rate of anti-crawler strategy, the higher the possibility of accidental injury.

Crawler and anti-crawler are mutually antagonistic. No matter how powerful the crawler is, it can be discovered by the complex anti-crawler mechanism. Similarly, no matter how rigorous the anti-crawler mechanism is, it can be broken by the advanced network crawler. The most basic feature of crawler is batch, and the anti-crawler mechanism also makes judgment based on this feature. However, anti-crawler is still a choice of balancing advantages and disadvantages, which requires both a low rate of misfire and a high rate of interception, which is also its vulnerability. So there's a tradeoff between the two[7].

In short, in the information age, through the comprehensive application of the various anti-crawler methods mentioned above, the negative impact of crawlers on websites can be greatly reduced and normal access to websites can be guaranteed. However, web content crawlers and anti-crawlers are destined to be a long-term coexistence situation, where there is a network there are crawlers, and where there are crawlers there is anti-crawling, and anti-web crawlers are always on the way. All we can do is to increase the cost of crawling for attackers and to be more precise about unauthorised crawling behaviour.

REFERENCES

[1] MA Hei Programmer. Python program development case tutorial[M]. China Railway Publishing House,2019:15-18

[2] Lu Zhang. Application of web crawler technology in big data[J]. Cooperative Economy and Technology, 2019(07):190-192.

[3] Tao Fan, Zheng Zhao, Minjuan Liu. Analysis and implementation of a Selenium-based web crawler[J]. Computer programming skills and maintenance, 2019(09):155-156+170.

[4] Guobiao Jiang. Research on the collection and application of audit big data based on web crawler technology[D]. Nanjing Audit University,2019.

[5] Jianwei Hu, Haiyan Qiu. Web crawler technology promotes the integration of big data and CPI surveys[N]. China Information News,2019-09-12(004).

[6] Yaohong Hao. Exploration of web crawling technology and information security strategy in the era of big data[A]. Network Security Bureau of the Ministry of Public Security: Information Network Security, Beijing Editorial Department, 2019:2.

[7] Hao Feng, Yongchang Lao, Lingjie Ye, Qiujie Sun, Taifeng Kang. Research on Big data Intelligent mining Technology based on Web crawler[J]. Electronic Design Engineering, 2019,27(16):161-164+169.

Authorized licensed use limited to: National PingTung University -. Downloaded on September 13,2022 at 08:31:17 UTC from IEEE Xplore. Restrictions apply.