Understanding Website Behavior based on User Agent

Kien Pham
Computer Science and Engineering
New York University
kien.pham@nyu.edu

Aécio Santos
Computer Science and Engineering
New York University
aecio.santos@nyu.edu

Juliana Freire
Computer Science and Engineering
New York University
juliana.freire@nyu.edu

ABSTRACT

Web sites have adopted a variety of adversarial techniques to prevent web crawlers from retrieving their content. While it is possible to simulate users behavior using a browser to crawl such sites, this approach is not scalable. Therefore, understanding existing adversarial techniques is important to design crawling strategies that can adapt to retrieve the content as efficiently as possible. Ideally, a web crawler should detect the nature of the adversarial policies and select the most cost-effective means to defeat them. In this paper, we discuss the results of a large-scale study of web site behavior based on their responses to different user-agents. We issued over 9 million HTTP GET requests to 1.3 million unique web sites from DMOZ using six different user-agents and the TOR network as an anonymous proxy. We observed that web sites do change their responses depending on user-agents and IP addresses. This suggests that probing sites for these features can be an effective means to detect adversarial techniques.

Keywords

User-agent String, Web Crawler Detection, Web Cloaking, Stealth Crawling

1. INTRODUCTION

There has been a proliferation of crawlers that roam the Web. In addition to crawler agents from major search engines, there are shopping bots that gather product prices, email harvesters that collect address for marketing companies and spammers, and malicious crawlers attempting to obtain information for cyber-criminals. These crawlers can overwhelm web sites and degrade their performance. In addition, they affect log statistics leading to an overestimation of user traffic. There are also sites connected to illicit activities, such as selling illegal drugs and human trafficking, that want to avoid crawlers to minimize their exposure and eventual detection by law enforcement agents. A number of strategies can be adopted to prevent crawler access. Sites can use the Robot Exclusion Protocol (REP) to regulate what web crawlers are allowed to crawl. But since the REP is not enforced, crawlers may ignore the rules and access the forbidden information. Web sites may also choose what content to return based on the client identification included in the user-agent field of the HTTP protocol. However, this string is not reliable since the identification is not secure and one can easily spoof this information in the HTTP requests. More robust detection methods have been developed, including the use of web server logs to build classification models that learn the navigational patterns of web crawlers [7] and the adoption of mechanisms that detect human activity (e.g., embedding JavaScript code in pages to obtain evidence of mouse movement) [5].

From a crawler's perspective, this raises a new question: how to find and retrieve content from sites that adopt adversarial techniques. It is possible to automate a browser to simulate human actions, but this approach is not scalable. Since pages need to be rendered and the crawler must simulate a user's click behavior, crawler throughput would be significantly hampered. Ideally, this expensive technique should only be applied for sites that require it, provided that the crawler can identify adversarial techniques and adapt its behavior accordingly.

In this paper, we take a first step in this direction by studying how web sites respond to different web crawlers and user-agents. In addition, we are also interested in determining whether the response patterns are associated to specific topics or site types. We issued over 9M HTTP GET requests to more than 1.3M unique web sites (obtained from DMOZ) using different user agent. User agents included a web browser, different types of web crawlers (e.g., search engine providers, well-known and less-known crawlers), and an invalid user-agent string. We also issued requests in which we masked our IP address using proxies from the TOR network. To reduce the risk that sites can identify our experiment and to reduce the chance the content changes in between requests, requests with different user-agents were sent from independent machines (with different IP addresses) and concurrently. As we discuss in Section 3, the analysis of the responses uncovered many interesting facts and insights that are useful for designing adversarial crawling strategies at scale. For example, we observed that requests from less-known crawlers have a higher chance of success. In contrast, when a TOR proxy is used, not only are most requests unsuccessful, but there is also a large number of exceptions. Another important finding is that response patterns vary for different topics – sensitive topics result in a larger number of 403 (forbidden) responses.

SIGIR '16, July 17-21, 2016, Pisa, Italy
© 2016 ACM. ISBN 978-1-4503-4069-4/16/07. . . $15.00
DOI: http://dx.doi.org/10.1145/2911451.2914757

The URLs used in the experiment, source code and response headers of all requests are available at https://github.com/ViDA-NYU/user-agent-study.

2. RELATED WORK

Many techniques have been proposed to detect web crawlers, for example log analysis, heuristic-based learning techniques, traffic pattern analysis and human-detection tests [1, 5, 6, 7]. Although relying on user-agent string is a naïve approach, our study shows that it is still used by web sites.

To regulate the behavior of crawlers, Web sites can implement the Robots Exclusion Protocol (REP) in a file called *robots.txt*. There have been studies that measured both REP adoption and the extent to which crawlers respect the rules set forth by sites [4, 2]. These studies found that while REP adoption is high, crawler compliance is low.

Cloaking is a technique whereby sites serve different content or URLs to humans and search engines. While there are legitimate uses for cloaking, this technique is widely used for spamming and presents a challenge for search engines [9, 8]. Cloaking detection usually requires a crawler to acquire at least two copies of the web pages: from a browser's view and from web crawler's view. In this paper, we also acquire multiple versions of pages returned to different user-agents.

3. EXPERIMENTS

3.1 Data Collection

We used DMOZ[1], a human-created directory of web sites, to collect URLs for our experiment. First, we selected URLs from all topics in DMOZ except the topic World, a non-English version URLs of DMOZ. This resulted in 1.9M URLs. Then, we grouped URLs by their web sites so that only one URL per web site is kept for the experiment. We do this to avoid hitting a web site multiple times, since this may lead a site to detect and potentially block our prober. After filtering, the original list was reduced to 1.3M URLs that were used in the experiment. Note that since each URL is associated with a DMOZ topic, we can analyze the behavior of web sites in particular topics.

3.2 Methodology

We spoofed the identity of our prober by changing the user-agent field in the header of an HTTP request. Table 1 shows 6 different user-agents corresponding to 5 different types of web crawlers. Besides well-known user-agents such as Chrome Web Browser, Google, and Bing, we used Nutch, ACHE and Empty (i.e., HTTP requests with an empty user agent). Nutch[2] is widely-adopted open-source web crawler. ACHE is focused crawler[3] and represents less-known system.

For each user-agent, we constructed an HTTP header with the corresponding user-agent string. The header was sent together with the HTTP GET requests for all URLs in our collection. We made roughly 7.8M requests for 1.3M URLs. Each URL received 6 requests (one per user-agent). Multiple requests to a URL with the same user-agent are only issued when an exception is raised (e.g., ConnectionError). To reduce the risk that sites can identify our experiment and to reduce the chance the content changes in between requests, these requests were sent simultaneously from machines with different IP addresses. For each request, we stored the entire response header and exception type.

Table 1: User-agent strings used in our experiments

ACHE	Ache
Bing	Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
Google	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Browser	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36
Nutch	Nutch
Empty	

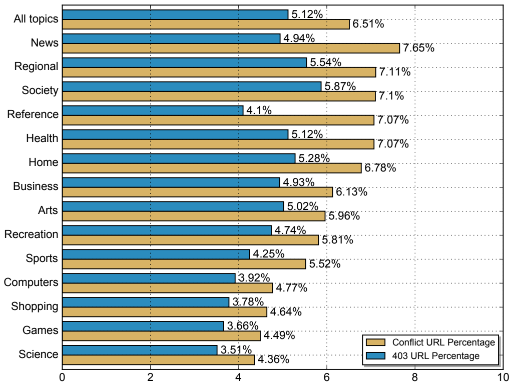

Figure 1: Percentage of URLs with conflicts and status code 403 for each topic

We ran a second experiment in which our prober used a TOR[4] proxy. The HTTP requests were routed through the TOR network, and thus, the sites could not identify our IP address, although they could detect that the request was issued from within the TOR network. In this experiment, we used ACHE as the user-agent.

If a request fails and raises an exception, i.e., HTTPConnectionError or Timeout, we repeat the request at most 5 times until the web site responds. We set a delay of 20 minutes between each request to avoid web site detecting our actions by looking at request frequency.

3.3 Response Analysis

We analyzed the response headers received from all requests and DMOZ topics associated with each URL. In what follows, we present the results of this analysis.

Conflicting URLs. We call a URL *conflicting* if the status codes returned for the different user-agents are inconsistent, i.e., at least one status code that is different from the others. We discovered 85K conflicting URLs which represents 6.5% of our URL collection. This finding provides a partial answer to our first question, whether web sites behave differently to different user-agents. Figure 1 shows the distribution of conflicting URLs grouped by high-level topics. The topic *News* has the highest number of conflicting URLs (7.65%), while *Science* has the smallest number (4.36 %). The percentage of conflicting URLs is an indicator of how strictly crawler detection based on user agent is enforced by web sites in different topics.

[1] http://www.dmoz.org

[2] http://nutch.apache.org/

[3] https://github.com/ViDA-NYU/ache

[4] https://www.torproject.org

Table 2: Number of exceptions by agents

	Connection Error	Timeout	Too Many Redirects
Google	46278	2768	165
Nutch	26494	1325	877
Bing	46410	2779	144
ACHE	45777	2581	135
Empty	47543	2238	292
Browser	45757	2335	129

Exceptions. Table 2 shows the number of exceptions raised by different user-agents. ConnectionError is the most frequent, accounting for 94.16% of all exceptions. Since this is a general exception, it could be caused either by the client or server due to network problems. Furthermore, ConnectionError exceptions happen at the network layers, therefore it does not reflect the web sites behavior on user-agents, which are handled at application layer. We have observed a few occurrences of Too-Many-Redirects exceptions, but most happen only for Nutch. This suggests that more sites block the Nutch crawler.

Status Codes. While the total number of distinct returned status codes is 56, in Table 3 we only show ones that present notable differences among our 6 user-agents. The definitions for these codes are given in Table 4[5]. Also, since most exceptions are not raised by application layer, we only take into account URLs that do not cause any exception to compute the status code statistics.

Note that the Nutch user-agent returns the highest number of status code 429 (Too Many Requests). Since Nutch is open source and has been widely adopted, it is possible that it is being misused, leading sites to block it. The Empty user-agent causes 407, 400, 403, 500, 503 which is reasonable because Empty is not a recognized user-agent string. (Service Unavailable) and 451 (Unavailable For Legal Reasons). A possible explanation is that some web sites do not want to be indexed in popular search engine, so they block Google. Another possibility is that the web sites do not respond with a 200 code because it detects spoofing in user-agent. Since Google publishes the IP addresses for its crawlers, it is easy for web sites to detect that our prober does not come for a Google address. Another interesting finding is that the ACHE user-agent returns the least number of 403 (Forbidden) codes. This suggests that sites are more permissive for less popular crawlers.

Overall, ACHE and Browser produce smaller numbers of unsuccessful status codes than Bing and Google. At a first glance, this is counter-intuitive. One possible reason is that web sites can easily detect spoofing if clients identify themselves as Google or Bing by verifying if the client's IP address is within the IP address range published by search engines. This observation suggests that to conduct a large-scale crawl, it might be a good idea to use a less popular crawler as the user-agent.

Finally, our results reveal that status codes may not provide the precise reasons for why web sites refuse access to their content. If a service is unavailable, it should return 503 to all requests, however in our experiment, the return codes differ for different user agents.

403 (Forbidden) Status Code. We found 66,911 URLs that returned 403 to at least one user-agent, we call these *403-URLs*. This represents 5.12% of all collected URLs. One question we investigated was whether there was a pattern for these URLs associated to the topics they belong to.

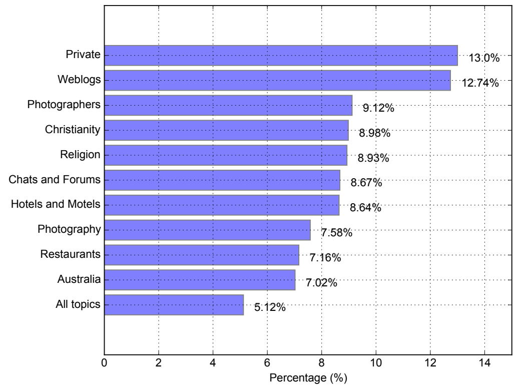

Figure 2: Topic distributions of 403-returned URLs

Since DMOZ topics are organized as tree, each URL is associated with a list of hierarchically connected topics. We grouped the 403-URLs by both lowest-level topics and highest-level topics separately. Each grouping provides different view: lowest-level topics tend to be specific while highest-level topics are more general. Figure 2 shows the top-10 lowest-level topics that have highest percentage of 403 URLs. We can easily see that topics which are more sensitive tend to return higher percentage of 403, e.g., Private, Weblogs, Religion or Chats and Forums. Note that we only take into account topics that consist of at least 100 URLs, so that the percentage values is more statistically representative. Figure 1 shows the percentage of 403 URLs over all highest-level topics. The pattern in this figure is not clear; the deviations from average percentage is not as significant as we observed in Figure 2. This is understandable since URLs from high-level topic are much more diverse than those from specific topics.

Using TOR as a Proxy. When we used TOR as routing proxy and ACHE as user-agent, we only got responses from 270K web sites out of 1.3M (20.7%); all other requests raised exceptions. This experiment indicates the ineffectiveness of using TOR to crawl anonymously. Furthermore, TOR seems to be overused for crawling, resulting in many web sites refusing access. Among the successful requests, only 29.07% of them got a 200 (OK) status code, the majority (69.17%) got a 503 (Service Unavailable) status code.

Content Difference. We analyzed the returned content by examining the content-length in the request response header. Although content-length does not show the precise differences in page content, it is cost-effective strategy to detect blocked pages [3]. Ideally, if a web site returns OK status code to all user-agents, we expect the returned content is the same as well as the content length in response header. Surprisingly, we discovered a significant difference in the returned content. We gathered 717K URLs with 200 status code returned to all user-agents. Out of these, 201K URLs, representing 28.96%, return content with different lengths. We computed the content length difference returned by each URL by subtracting length of the maximum content and that of minimum one. Figure 3 shows the histogram of content length difference, i.e., how many URLs return content with significantly different length. In the result, the length

[5] Source: https://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html

Table 3: Number of URLs returning non-OK status code

	404	**451**	**407**	**429**	**406**	**403**	**503**	**400**	**416**	**500**	**999**	**463**	**Total (excluding 200)**
Google	24173	95	0	10	21	15365	6242	452	144	2239	1	0	49142
Bing	24734	11	0	39	72	16483	1161	383	145	2250	0	0	45275
Nutch	24319	11	0	8555	234	39147	883	450	164	2354	34	1	77168
ACHE	23583	11	0	12	14	2127	837	367	0	2192	0	0	29519
Empty	23636	11	466	10	297	16378	1921	9675	169	6976	36	418	60566
Browser	23552	11	0	17	13	4646	844	368	2	2108	0	0	31916

Table 4: Status code definitions

Status Code	**Definitions**
451	Unavailable For Legal Reasons
407	Proxy Authentication Required
429	Too Many Requests
406	Not Acceptable
403	Forbidden
503	Service Unavailable
400	Bad Request
416	Requested Range Not Satisfiable
500	Internal Server Error
999, 463	Undefined

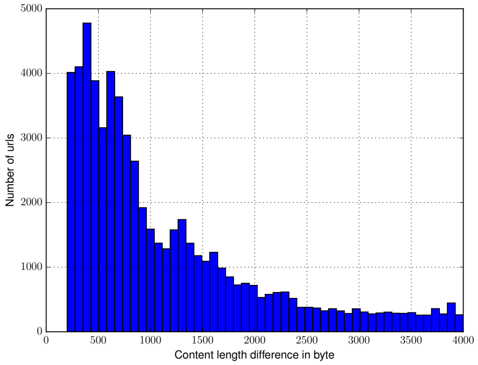

Figure 3: Content length difference in bytes

difference ranges from 1 byte to 3.6 MB, however in this figure, we omit the long tail since it is less informative. Also, we ignore cases where difference is less than 200 bytes. Although content length difference does not represent the semantic difference, it shows that web sites behaves differently to different user-agents.

4. CONCLUSIONS AND FUTURE WORK

We have carried out the first large-scale study of web site behavior based on their responses to different user-agents and to TOR proxies. The study shows that web sites respond with different content and status code to different user-agents and IP addresses. In addition, response patterns vary for different topics. The results also provide insights into potential strategies for adversarial crawling, including the use of a less popular user-agent.

There are several avenues we intend to pursue in future work. In particular, we plan to study in more detail the actual differences in the content returned to different agents as well as other aspects sites take into account for blocking crawlers (e.g., violation of the rules in robots.txt). Ultimately, our goal is to design effective techniques for adversarial crawling.

5. ACKNOWLEDGMENTS

This work was supported in part by NSF (awards CNS-1229185 and CI-EN-1405927) and the Defense Advanced Research Projects Agency (DARPA) MEMEX program award FA8750-14-2-023. We specially thank to Dr. Ari Juels for valuable feedback in the design of the experiment.

6. REFERENCES

[1] D. Doran and S. S. Gokhale. Web robot detection techniques: Overview and limitations. *Data Mining Knowledge Discovery*, pages 183–210, Jan. 2011.

[2] C. L. Giles, Y. Sun, and I. G. Councill. Measuring the web crawler ethics. In *roceedings of the 19th International Conference on World Wide Web*, pages 1101–1102, 2010.

[3] B. Jones, T.-W. Lee, N. Feamster, and P. Gill. Automated detection and fingerprinting of censorship block pages. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 299–304, New York, NY, USA, 2014. ACM.

[4] S. Kolay, P. D'Alberto, A. Dasdan, and A. Bhattacharjee. A larger scale study of robots.txt. In *Proceedings of the 17th International Conference on World Wide Web*, pages 1171–1172, 2008.

[5] K. Park, V. S. Pai, K.-W. Lee, and S. Calo. Securing web service by automatic robot detection. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference*, pages 23–23, 2006.

[6] A. Stassopoulou and M. D. Dikaiakos. Web robot detection: A probabilistic reasoning approach. *Computer Networks*, pages 265–278, 2009.

[7] P.-N. Tan and V. Kumar. Discovery of web robot sessions based on their navigational patterns. *Data Mining Knowledge Discovery*, pages 9–35, 2002.

[8] D. Y. Wang, S. Savage, and G. M. Voelker. Cloak and dagger: Dynamics of web search cloaking. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, pages 477–490, 2011.

[9] B. Wu and B. D. Davison. Cloaking and redirection: A preliminary study. In *Proceedings of the first International Workshop on Adversarial Information Retrieval on the Web*, Chiba, Japan, 2005.