Analysis of header usage patterns of HTTP request messages

Maria Carla Calzarossa
Dipartimento di Ingegneria Industriale e Informazione
Università di Pavia
via Ferrata 5 – I-27100 Pavia, Italy
mcc@unipv.it

Luisa Massari
Dipartimento di Ingegneria Industriale e Informazione
Università di Pavia
via Ferrata 5 – I-27100 Pavia, Italy
massari@unipv.it

***Abstract*—The headers used by the various network protocols represent a good source of information for understanding the behavior and the properties of the traffic and detecting potential security attacks. In this paper we present a comprehensive analysis of the usage patterns of the headers included in the HTTP request messages of the clients. Our study shows that message headers vary significantly in terms of number, field names and field values. In general, Web browsers tend to specify in detail the client preferences by including in the request messages a large and variable number of headers. The usage patterns of these headers together with their field values provide some useful hints for website customization. On the contrary, the requests issued by Web robots are characterized by some distinctive patterns specific of the individual robots. These one-to-one correspondences between patterns and Web robots can then be used for their identification.**

I. Introduction

The increased pervasiveness of Web technologies has led to a considerable growth of the HTTP traffic that now represents a significant portion of the total Internet traffic. Web content has become more complex in terms of number of objects embedded in the pages, their size and the client-side interactions. To cope with this complexity and reduce the client latency, Web browsers have increased the number of simultaneous connections per Web server. Similarly, Web robots are deploying some more aggressive crawling policies, thus, making the traffic appear more burstier. In addition, the growth of the HTTP traffic is coupled with an increase of security threats. Websites have become the favorite targets of hackers who inject various types of malware to exploit vulnerabilities and crawl the sites to extract their business intelligence or to steal sensitive information.

In this paper we present a comprehensive analysis of the Web traffic in terms of the HTTP request messages issued by the clients. Our objective is to characterize the usage of the various message headers, discover their patterns and discuss their role in the customization of websites and in the development of Web robot detection techniques.

Let us recall that headers are used in a variable number to alter and describe the HTTP requests [9], [12]. In our study, instead of resorting to the information available in the standard log files stored by the Web servers, we capture the entire HTTP request messages by means of a simple network sniffer. This allows us to study in details the properties and characteristics of the headers included in each request message.

The paper is organized as follows. After the review of the work related to Web traffic analysis, given in Section II, the monitoring approach adopted in our study is described in Section III. The overall characteristics of the HTTP request messages collected by the sniffer are presented in Section IV. Section V introduces and discusses the usage patterns of the message headers. Finally, Section VI presents some concluding remarks.

II. Related work

The characteristics of Web traffic and its evolution have been addressed by several papers since its inception (see, e.g., [2], [4], [10], [14], [15], [17], [18]). Most of these studies are experimental, that is, they rely on measurements collected on the servers being deployed. Passive monitoring techniques are usually employed for this purpose. In particular, the headers associated with the various network protocols are a good source of information for characterizing Web traffic and detecting security attacks. For example, the TCP/IP protocol headers are captured and used in [19] to infer the characteristics of the Web traffic.

The HTTP request and response headers stored by Web servers in their access log files have been extensively used to discover the properties of the traffic of Web servers and the behavior of their clients (see, e.g., [5]). In addition, Web logs are used to characterize the activities of the Web robots as well as for their classification and detection (see, e.g., [6], [7], [8], [13], [20]).

An approach based on inspection and validation of HTTP headers, such as, `Referer` and `Origin`, is applied in [3] to implement defense techniques against Cross-Site Request Forgery attacks. In [16] headers are used to analyze malware samples and identify how malware makes use of the HTTP protocol. The study detects a significant number of mispelled or non-standard headers and some requests without any header at all.

The contribution of our paper to the analysis of Web traffic is two-fold. The detailed data collected for each HTTP

request message provide a novel perspective of the traffic from the client side. Moreover, the analysis of the headers included in these messages highlights usage patterns specific of the various types of Web traffic. These results could be used to detect the presence of Web robots and especially of malicious robots that camouflage themselves by forging the values of the `User-Agent` header field.

III. Monitoring approach

The Web traffic considered in this study consists of the HTTP request messages generated by the clients towards two Web servers hosted at our University. The monitoring approach adopted to collect these messages relies on a simple network sniffer that allows us to capture all packets flowing on the network segment that connects the servers and retain the packets referring to the HTTP requests only. This approach is motivated by the compelling need to avoid any perturbation to the regular operations performed by the Web servers. We emphasize that we could have gathered the same type of information by enabling on the servers the Apache module that provides for forensic logging of client requests and server responses [1]. Nevertheless, this type of logging is seldom activated on Web servers because it could affect their performance due to log files growing extremely large. On the contrary, sniffers do not affect server performance as they usually run on dedicated hosts that share the network segment of the servers.

The sniffer relies on *libpcap*, an open source library that provides a portable framework for network monitoring [11]. In particular, using some *libpcap* functions, we implement a filter that specifies to capture the packets carrying HTTP request messages, that is, packets whose IP destination address corresponds to the IP address of one of the Web servers and whose TCP destination port is 80. We also set the maximum number of bytes to be captured to the maximum Ethernet frame size, that is, 1,518 bytes. Once the filter expressions are compiled and applied to the sniffer, the sniffer enters its packet capturing loop running in the kernel space and storing the captured packets in a buffer. A callback function is finally called to periodically copy the content of the buffer from the kernel space to the user space. Let us recall that to intercept all packets flowing on the network segment, the network interface card of the host needs to be set into promiscuous mode.

The information stored for each packet includes the time stamp of the packet, the IP address of the source host, that is, the IP address of the client issuing the HTTP request, and the entire HTTP request message, that is, the request line followed by the corresponding headers.

Figure 1 shows one of the request messages captured by the sniffer. The first line refers to the request line, whereas the remaining lines correspond to the headers included in the message. In detail, this request includes seven headers, each consisting of a field name (e.g., `Accept-Language`), followed by a colon and the corresponding field value (e.g., `ru, uk;q=0.8, be;q=0.8, en;q=0.7, *;q=0.01`).

```
GET / HTTP/1.1
Host: myserver
Connection: Keep-Alive
Accept: text/html
Accept-Encoding: gzip,deflate
Accept-Language: ru, uk;q=0.8, be;q=0.8, en;q=0.7,
*;q=0.01
User-Agent: Mozilla/5.0 (compatible; YandexBot/3.0;
+http://yandex.com/bots)
From: email@crawler-xxx.com
```

Figure 1. Example of an HTTP request message captured by the sniffer.

IV. Overall characteristics of request messages

The dataset analyzed in this study consists of some 315,000 packets referring to the HTTP request messages with a valid request line captured by the sniffer. This data was collected during a monitoring interval of about three months in the spring 2013. The requests, originating from approximately 6,100 clients, are not evenly distributed across the clients: most clients generate a small number of requests each, whereas seven clients are responsible of about two third of the traffic. In addition, the request line of the vast majority of the requests refers to the `GET` method, that is, the method that notifies the server to fetch the resource identified by the Uniform Resource Identifier embedded in the request line. Similarly, very many request lines include `HTTP/1.1` as the protocol version in use, whereas only about 4% refers to `HTTP/1.0`.

The analysis of the request messages shows that the headers that are part of the messages vary significantly in terms of field names and values as well as number. Even though most headers are optional, Web browsers and other software agents used to generate the requests include them for specific reasons. In particular, the headers of request messages are used to express preferences on the nature of the response, to include additional information with the request or to specify a constraint on the server in handling the request.

Some messages do not include any header at all, whereas others include as many as 14 headers. The average number of headers per request message is equal to 6.34. The details of the corresponding distribution are shown in Figure 2. We notice a peak of about 143,000 request messages, that is, 46% of the messages, including six headers each. In addition, less than a thousand requests do not include any header and some 10,000 requests include at least nine headers. In general, as we will discuss in more details in Section V, the number of headers within HTTP request messages varies with the device and the software agent used

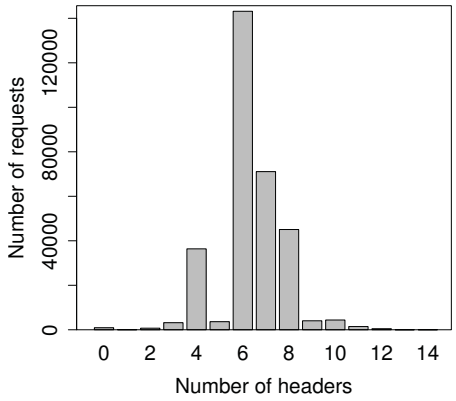

Figure 2. Distribution of the number of headers of the HTTP request messages analyzed in our study.

by the clients to send their requests, whereas the dependence on the IP address of the clients is less evident. This could be due to the dynamic assignment of the IP addresses of the clients often adopted by providers as well as by the presence of clients behind firewalls or proxy servers.

By analyzing the request messages stored in our dataset, we notice a large variety of header field names, including, among the others, some mispelled names and some non-standard names with the `X-` prefix, such as, `X-OperaMini-Features`, `X-Requested-With`. More specifically, we detect 60 unique names whose popularity varies significantly (see Figure 3). We notice some names

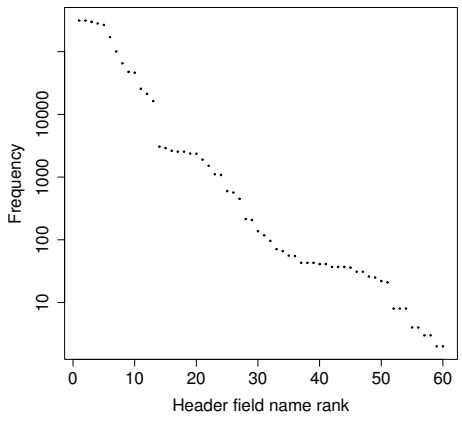

Figure 3. Popularity of the unique header field names detected in our dataset.

appearing in almost every request message. This is the case, for example, of the `Host` and the `User-Agent` headers, used to specify the host that can provide the resource being required, and the Web browser or agent originating the request. These field names appear in about 99.8% and 99.3% of the messages, respectively. Let us remark that when using `HTTP/1.1`, the `Host` header is mandatory to cope with the implementation of the virtual hosting mechanisms. Nevertheless, not every message includes it.

Moreover, the `Connection` general header, denoting the options desired by the client with respect to the TCP connection with the Web server, and the `Accept` header, specifying the acceptable media types for the HTTP response, are part of the majority of the messages. In a good number of messages, namely, 170,148, we also find the `From` header, used to denote the email address of the person responsible of the requesting user agent. The reverse DNS lookup of the corresponding IP addresses shows that the clients generating these request messages mainly refer to organizations operating major Web robots, such as, Googlebot, Bingbot, YandexBot. In our dataset we also discover a non-standard header field name, that is, `Botname`, used by some very specific Web robots to identify themselves.

The values associated with the field names vary considerably as a function of the name and of the preferences expressed by the clients generating the requests. Some headers, e.g., `Connection`, have a limited number of values, whereas the values of others, e.g., `If-modified-since`, `Referer`, can change at every request.

In the case of the `User-Agent` header, the value usually consists of a long string of characters specifying the details of the agent or Web browser being used, such as, its name and version, as well as the operating system and sometimes the device where it runs (see, e.g., Fig. 1). In total, for this header we detect some 1,350 values. Through a more detailed analysis, we discover about 1,100 values referring to Web browsers, such as, Apple Safari, Google Chrome, Microsoft Internet Explorer, Mozilla Firefox, that appear under many different versions and configurations. Some 150 values denote software agents used by Web robots, such as, Baiduspider, Bingbot, Exabot, Googlebot, Ocelli, Sogou web spider, YandexBot, and other open source scripts, such as, `curl`, `libwww-perl`, `wget`, used to crawl specific Web pages or entire websites for legitimate or malicious purposes. The remaining values are a sort of miscellanea that cannot be classified in any specific category and are used by only 3,000 request messages.

Tables I and II present the characteristics of the main header field names used in the request messages of Web robots and browsers, respectively. We notice that the headers used by Web robots are characterized by a limited number of values. For example, the value of the `Accept` header specified by most robots is `*/*`, that is, they accept for the responses all media types and subtypes. In addition, some headers are used by robots mainly operating in specific

Web robot	Number of values				Requests
	User-agent	Accept	Accept-language	From	
Baiduspider	4	1	4	-	5,104
Bingbot	3	1	-	1	11,507
Fast Enterprise	1	5	-	-	82,606
Googlebot	9	4	-	1	120,875
Ichiro	3	1	1	1	3,507
Psbot	1	1	-	1	2,190
YandexBot	3	3	2	1	4,920

Table I
NUMBER OF VALUES ASSOCIATED WITH THE MAIN HEADER FIELD NAMES AS A FUNCTION OF THE WEB ROBOT GENERATING THE REQUESTS. THE DASH SIGN DENOTES THE LACK OF THE CORRESPONDING HEADER.

countries, such as, Russia, China and Japan. For example, the language preferences of Baiduspider, that is, the values associated with the `Accept-language` header, include Chinese, i.e., `zh-cn` and `zh-tw`, Japanese, i.e., `ja-JP`, as well as English, i.e., `en-US`. We also detect that all robots, but Baiduspider and FAST Enterprise, associate with the `From` header one email address.

Browser	Number of values				Requests
	User-agent	Accept	Accept-language	Cache-control	
Chrome	220	9	55	8	11,843
Firefox	258	13	53	15	13,213
MSIE	405	66	73	15	10,242

Table II
NUMBER OF VALUES ASSOCIATED WITH THE MAIN HEADER FIELD NAMES AS A FUNCTION OF THE WEB BROWSER GENERATING THE REQUESTS.

Despite Web robots, the headers of the request messages generated by Web browsers are characterized by a much larger variety of values. For example, the Mozilla Firefox browser appears in the `User-agent` header under 258 different configurations, running under operating systems, such as, Android, Linux, Windows, Mac OS. Similarly, we detect more than 50 strings describing the language preferences of these browsers. The values of the `Accept-language` header include, among the others, Hungarian, Finnish, Russian, Portuguese, Dutch, Slovak and Polish, as well as many combinations of two or more languages, e.g., German and English, English US and English GB. We also observe that Web browsers specify caching directives using a limited number of values in the corresponding `Cache-control` header. It is worth noting that all these details could be very useful for website customization according to the preferences specified by the clients in their request messages.

Finally, it is important to point out that the header field names as well as their values can be easily forged. This is the case of mispelled header names. In addition, the values of the `User-Agent` header are often forged to make the request appearing to the Web server as generated by a different agent. Hence, to validate and ensure the authenticity of these strings and especially of those referring to Web robot agents, it is advisable to analyze in detail the usage patterns of the headers and possibly combine it with a reverse DNS lookup of the IP address of the corresponding clients.

V. HEADER USAGE PATTERNS

As pointed out in the previous section, the headers that are part of the HTTP request messages vary in terms of numbers, field names and field values. It is then important to study the usage of these headers to discover their patterns and assess how common each specific pattern is and whether there is any relationship with the clients and agents used to issue the requests.

For a good number of the clients, namely, about 4,400, the number of headers included in their request messages does not vary across messages. On the contrary, for the remaining 1,700 clients, the number changes and for some clients it does change even significantly. These results are somehow expected and in line with the assignment of IP addresses previously discussed.

Another interesting difference in the usage of the headers emerges from the analysis of the request messages generated by two main categories of user agents, that is, Web browsers and software agents employed by Web robots. In general, Web browsers tend to include in their requests a larger and more variable number of headers than Web robot agents do. On average messages include 7.1 and 6.3 headers, respectively. In addition, about 24% of the messages generated by Web browsers include at least nine headers. On the contrary, in the case of Web robot agents, the corresponding percentage does not reach 1%. It is also worth mentioning that messages generated by Web browsers running on mobile devices tend to include a larger number of headers, that is, on average 7.74 headers per message.

A more detailed analysis of the usage patterns of the headers is performed by focusing on the header field names and studying their usage in the request messages. Out of the 60 field names detected in our dataset, we identify 545 unique patterns, that is, 545 groups of messages characterized by the same set of field names. The diagrams of Figure 4 summarize the composition and popularity of some of these patterns. Each row represents a pattern. The light green and dark green areas denote the presence and the lack of a given header field name in the pattern, respectively. In addition, the patterns displayed in the diagrams are sorted according to their popularity.

The diagrams clearly highlight the similarities and differences among patterns. We identify a limited number of very common patterns grouping the majority of the request messages and a very large number of less popular patterns. The ten patterns shown in Fig. 4 (a) account for 81% of the messages. In particular, about one third of the messages

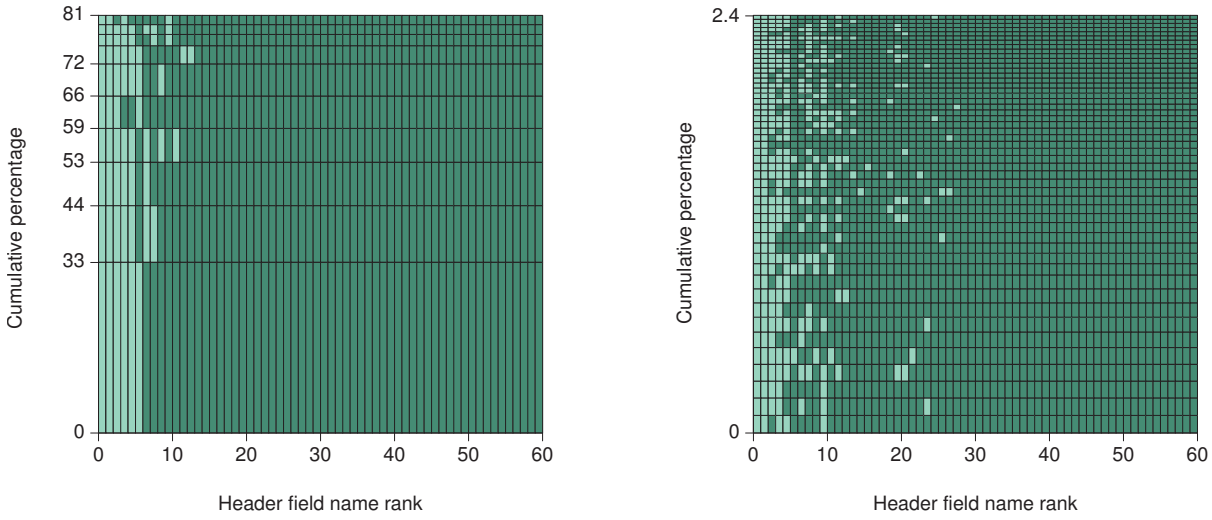

Figure 4. Composition and popularity of the usage patterns as a function of the 60 header field names detected in our dataset.

shares the most popular pattern. On the contrary, the 50 patterns of Fig. 4 (b) group only the 2.4% of the messages, i.e., on average 152 messages per pattern.

By looking at the composition of these patterns, we notice light green areas on the left hand side of both diagrams. These areas correspond to very popular header field names that are included in almost all request messages. On the contrary, the large dark green areas on the right hand side of the diagrams correspond to header names, such as, non-standard headers, seldom included in the messages. Note that these headers are often part of the requests generated by mobile devices, such as, Blackberry, Nokia and Samsung smartphones.

In the figure, we can also observe that patterns are characterized by a variable number of headers. For example, the most popular pattern (see Fig. 4 (a)) consists of six headers, that is, `Host`, `User-Agent`, `Accept`, `Connection`, `Accept-Encoding`, `From`. The analysis of the `User-Agent` values and the IP addresses of the clients shows that this pattern mainly refers to requests issued by the various clients used by Google for its crawling activities.

In general, we notice that the requests generated by Web robots are characterized some distinctive patterns that are specific of the individual robots. This is the case of FAST Enterprise Crawler, whose requests follow three different patterns that are not used by any other robot. Similarly, we detect two patterns corresponding to the requests generated by YandexBot, and one pattern to the requests generated by Binbgbot. We emphasize that this one-to-one correspondence between patterns and Web robots could be very useful for robot identification. In addition, it is important to point out that the patterns corresponding to `User-Agent` strings forged to make the requests appearing as generated by a Web robot, usually consist of three headers only, that is, `Host`, `Connection` and, of course, `User-Agent`.

Figure 5 shows the distribution of the number of headers of the 545 patterns identified in our dataset. We can observe

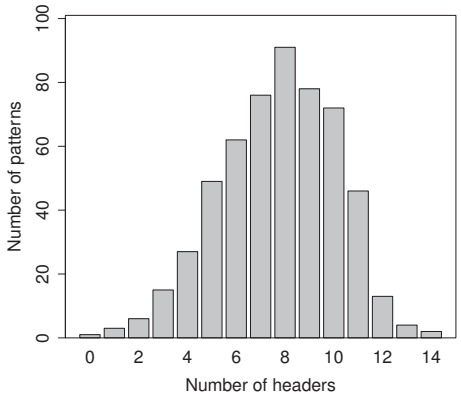

Figure 5. Distribution of the number of headers of the usage patterns identified in our dataset.

a sort of right-skewed bell shaped distribution whose highest peaks correspond to a number of headers ranging from seven up to ten. These peaks globally account for more than 300 patterns, mainly referring to requests generated by Web browsers. It is worth noting that even though the requests of the Web browsers do not represent the majority of the overall

traffic considered in our study, they are spread across many different patterns.

The analysis of the usage patterns is further specialized by considering the values associated with the various header field names. Note that in building these patterns, we do not consider the values of headers that can change in every message, such as, headers used for conditional requests, e.g., `If-match`, or for sending specific information, e.g., `Referer`. For these field names, we simply take into account their presence or lack in the request message.

From the set of values used by the headers, we obtain about 5,100 unique patterns, each including on average 62 requests. The most popular pattern groups about 75,000 request. We also detect some 2,000 patterns with one request only. Moreover, as expected, most of these patterns refer to Web browsers, and very few, namely, less than 250, to Web robots. Nevertheless, it is important to study these patterns as they can drive the identification of Web robots.

Table III presents the behavior of some Web robots by summarizing the main characteristics of their usage patterns. From the table, we notice that the requests of individual robots are grouped in a limited number of patterns even though some of them refer to specific agents employed by the organizations operating the Web robots for crawling content, such as, images, content for mobile devices. This is, for example, the case of Google, whose 17 patterns include agents, such as, plain Googlebot, Googlebot-Image, Googlebot-Mobile, Google-Site-Verification.

Web robot	Headers per pattern	Requests per pattern	Clients per pattern	Number of patterns
Baiduspider	6.10	510.40	60.20	10
Bingbot	8.17	1,917.83	60.00	6
Fast Enterprise	7.27	7,509.64	1.45	11
Googlebot	5.94	7,710.29	19.76	17
Ichiro	3.70	501.00	1.42	7
Psbot	5.50	547.50	1.75	4
YandexBot	6.45	447.27	2.63	11

Table III
MAIN CHARACTERISTICS OF THE USAGE PATTERNS OF SOME WEB ROBOTS.

It is then evident that the requests of the various Web robots are characterized by a good degree of similarity in terms of both header field names and values. In addition, for each individual robot the number of requests does not significantly vary across patterns. Moreover, the patterns identified for Web robots are rather stable, that is, they have been used without any significant variation across the entire monitoring interval.

Despite these similarities, the behavior and the usage patterns of Web robots with respect to the clients involved in the crawling activities vary. For some robots, there is almost a one-to-one correspondence between client and usage pattern; for others, multiple cooperating clients share the same pattern. This is the case of Baiduspider and Bingbot whose patterns are shared on average by some 60 clients each. Hence, the detection of Web robots cannot rely only on the analysis of the IP addresses of the clients; the usage patterns of the headers have to be taken into account.

Our analysis also shows that field names, such as, `TE`, used to specify the preferred transfer encoding, often denote forgeries. Indeed, the corresponding messages contain forged values of the `User-Agent` header or request resources, such as, `zboard.php`, `xmlrpc.php`, aimed at exploiting website vulnerabilities. Let us remark that the `TE` header is almost always used in combination with a `Connection` header whose value is the string `TE, close`.

In general, headers and their usage patterns represent a good source of information for website customization and Web server tuning in that they provide a detailed overview of the preferences of the clients as well as of the behavior of Web robots.

VI. CONCLUSIONS

The analysis of Web traffic usually relies on the information stored by Web servers into their log files. Nevertheless, these files often include a small subset of the headers that are part of the HTTP request messages.

Our study shows that the message headers and their usage patterns provide some valuable information for understanding the behavior and the preferences of the clients and for detecting the presence of Web robots and forgeries. An important finding is that the headers of the messages generated by the software agents used by Web robots are very specific and distinctive of the robots themselves. On the contrary, requests generated by Web browsers include headers whose number, field names and values vary significantly. In general, Web browsers tend to provide many details about the preferences of the clients. This usually leads to an increase of the number of bytes being transmitted over the network. Hence, software agents have to take into account the tradeoff between the information included in their request messages and what is actually necessary and used by Web servers.

An additional interesting finding deals with the request messages that include forged values of the `User-agent` header. Our analysis shows that many of the agents used to generate these requests do not bother to add headers, other than the `Host` mandatory header and the `Connection` header. Hence, these requests can be easily identified from the usage patterns of the headers.

The results presented in this paper could be used for the customization of websites that cope with the preferences expressed by the clients in terms, for example, of languages and content types supported. Moreover, the patterns identified for Web robots could be incorporated in automatic traffic regulation mechanisms aimed at avoiding server overload and limiting bandwidth usage.

REFERENCES

[1] Apache Software Foundation. Apache Forensis Logging. http://httpd.apache.org/docs/2.2/mod/mod_log_forensic.html.

[2] M.F. Arlitt and C.L. Williamson. Internet Web servers: workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 5(5):631–645, 1997.

[3] A. Barth, C. Jackson, and J.C. Mitchell. Robust defenses for cross-site request forgery. In *Proc. of the 15th ACM Conf. on Computer and Communications Security - CCS '08*, pages 75–88. ACM, 2008.

[4] T. Callahan, M. Allman, and V. Paxson. A Longitudinal View of HTTP Traffic. In A. Krishnamurthy and B. Plattner, editors, *Passive and Active Measurement*, volume 6032 of *Lecture Notes in Computer Science*, pages 222–231. Springer, 2010.

[5] M. Calzarossa and L. Massari. Analysis of Web logs: Challenges and findings. In K.A. Hummel, H. Hlavacs, and W. Gansterer, editors, *Performance Evaluation of Computer and Communication Systems - Milestones and Future Challenges*, volume 6821 of *Lecture Notes in Computer Science*, pages 227–239. Springer, 2011.

[6] M. Calzarossa, L. Massari, and D. Tessera. An extensive study of Web robots traffic. In *Proc. of the 15th Int. Conf. on Information Integration and Web-based Applications and Services - iiWAS2013*, pages 410–417. ACM, 2013.

[7] M.D. Dikaiakos, A. Stassopoulou, and L. Papageorgiou. An investigation of web crawler behavior: characterization and metrics. *Computer Communications*, 28(8):880 – 897, 2005.

[8] D. Doran and S.S. Gokhale. Web Robot Detection Techniques: Overview and Limitations. *Data Mining Knowledge Discovery*, 22(1-2):183–210, 2011.

[9] R.T. Fielding et al. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), 1999. http://www.ietf.org/rfc/rfc2616.txt.

[10] S. Ihm and V.S. Pai. Towards understanding modern web traffic. In *Proc. of the 2011 ACM SIGCOMM Conf. on Internet measurement - IMC '11*, pages 295–312. ACM, 2011.

[11] V. Jacobson, C. Leres, and S. McCanne. Libpcap. Lawrence Berkeley Laboratory, University of California, 1996.

[12] B. Krishnamurthy and J. Rexford. *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*. Addison-Wesley, 2001.

[13] J. Lee, S. Cha, D. Lee, and H. Lee. Classification of web robots: An empirical study based on over one billion requests. *Computers & Security*, 28(8):795 – 802, 2009.

[14] B. Newton, K. Jeffay, and J. Aikat. The Continued Evolution of Web Traffic. In *Proc. of the IEEE 21st Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems - MASCOTS 2013*, pages 80–89. IEEE Computer Society Press, 2013.

[15] J.E. Pitkow. Summary of WWW Characterizations. *World Wide Web*, 2(1-2):3–13, 1999.

[16] C. Rossow, C.J. Dietrich, H. Bos, L. Cavallaro, M. van Steen, F.C. Freiling, and N. Pohlmann. Sandnet: network traffic analysis of malicious software. In *Proc. of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security - BADGERS '11*, pages 78–88. ACM, 2011.

[17] R. Sadre and B.R.H.M. Haverkort. Changes in the Web from 2000 to 2007. In *Proc. 19th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, volume 5273 of *Lecture Notes in Computer Science*, pages 136–148. Springer Verlag, 2008.

[18] F. Schneider, B. Ager, G. Maier, A. Feldmann, and S. Uhlig. Pitfalls in HTTP Traffic Measurements and Analysis. In N. Taft and F. Ricciato, editors, *Passive and Active Measurement*, volume 7192 of *Lecture Notes in Computer Science*, pages 242–251. Springer, 2012.

[19] F.D. Smith, F.H. Campos, K. Jeffay, and D. Ott. What TCP/IP protocol headers can tell us about the web. In *Proc. of the 2001 ACM SIGMETRICS Int. Conf. on Measurement and Modeling of Computer Systems*, pages 245–256. ACM, 2001.

[20] A. Stassopoulou and M.D. Dikaiakos. Web Robot Detection: A Probabilistic Reasoning Approach. *Computer Networks*, 53(3):265–278, 2009.