Toward the Identification of Anonymous Web Proxies

Marco Canini
DIST
University of Genoa, Italy

Wei Li
Computer Laboratory
University of Cambridge, UK

Andrew W. Moore
Computer Laboratory
University of Cambridge, UK

ABSTRACT

Anonymous proxies have recently emerged as very effective tools for Internet misuse ranging from *Activity and Online Information Abuse* to *Criminal and Cybersexual Internet Abuse*. The ease with which existing proxies can be found and accessed, and new ones can be quickly set up poses an increasing difficulty to identify them. The traditional solution relies on URL filtering approach based on keyword databases. However, such approach cannot keep up with hundreds of new proxies created each day and more importantly the growing adoption of encrypted connections.

This work introduces a new methodology that uses flow features to create server profiles and then identifies potential proxies within the observed servers. Finally, we present early experimental results of this methodology.

1. INTRODUCTION

The misuse of Internet is highly undesirable in environments such as corporate and educational networks. Employee's productivity, legal liability, security risks, and bandwidth drain are potential concerns for many companies.

To understand the severity of the problem, we turn to the case of online social networks. Sites such as Facebook and MySpace have quickly gained popularity and are now widespread, each having over one hundred million subscribers. Recent reports in the popular media indicate that these sites are potentially costing corporations several billions of dollars annually, according to pools carried out amongst office workers[1].

However, quantifying Internet abuse is difficult. Firstly the form of misuse may vary[2], and secondly the line that separates between misuse and legitimate use is not a sharp one. Several studies have been conducted in which employees self-reported behavior that could be considered as Internet abuse. Unfortunately, in this context, self-reporting is subject to critism as it might not be sincere. Johnson and Chalmers [3] took a different approach to study employee Internet abuse: they analyzed the firewall log file of a large company with offices in several countries. They conclude that much of the employee Internet activity may have constituted inappropriate use of the company's time and IT resources.

[1]For example, `http://www.gss.co.uk/press/?&id=17`
[2]In [2], Griffiths offers a complete taxonomy.

PAM2009, April 1-3, 2009, Seoul, Korea.

Traditionally, URL or IP filtering have been adopted to enforce acceptable Internet use policies [4]. Unfortunately, these techniques can be easily circumvented with the use of anonymous Web proxies, especially when the traffic is encrypted (HTTPS), as in most settings these proxies appear as unrestricted Web sites. Despite these sites giving the opportunity for unconstrained Internet misuse, they might exist for more malicious reasons, for example, harvesting login credentials or disseminating malware.

The number of anonymous proxy sites has grown significantly in the past few years, especially through widespread installations of home-based proxies as many open source implementations of web proxies exist. Such a vast and dynamic deployment of proxies makes their effective identification challenging.

In this paper, we propose a method to detect Web proxies. Our method, based upon measurement of simple flow characteristics and server profiling, does not rely on packet payload inspection.

2. WEB PROXIES EXPLAINED

An anonymous Web proxy is a special form of the normal innocent "proxy server": a mediator between a client and a server which forwards every client request to the server and delivers the server response back to the client.

Firstly, the user log on to the proxy's home page and enters the URL he whishes to access. The browser sends this URL to the proxy server via a standard HTTP request. The proxy then fetches the requested page and, before returning it to the user's browser, it rewrites all the URLs contained in the original HTML page to point to the proxy server. In addition, some proxies also include a new navigation bar (with the URL input box) and advertisements in the final HTML. Effectively, the page and all its content are obtained through the proxy without any direct communication between the user's browser and the target Web site. Clearly, this rewrite process introduces a delay, albeit small, which sums to the delay of making the request through the proxy. We found that the total delay is obviously noticeable and affects the browsing experience.

If possible, some proxies use local caches to address this problem. Upon the user's browser request, proxies redirect the browser few times meanwhile the page is either fetched from the server or retrieved from the cache. The way the communication with the client is handled depends on the implementation. In some cases, all the HTTP requests to access a single URL reuse the same TCP connections. Conversely, for other proxies, the server terminates the connec-

tion after each request. Therefore, multiple requests are needed to fetch the URL, but we consider that it is possible to correlate successive requests and determine whether a request is caused by a previous one.

3. METHODOLOGY

The underlying idea of our approach is to consider the skewness in the server response time (by consider packets' inter-arrival times) and the statistics of the payload sizes for the first few packets of a flow. An explanation of the rationale behind this is now given.

Consider a single proxy which is used as a mean to browse the Web. Reasonably, the RTT between the user's browser and the proxy server can be assumed to be lowly variant. On the other hand, the RTT between the proxy and the destination server will greatly depend on what the destination is. Therefore, we expect the proxy response time to be variant, and we set to exploit this characteristic for identifying proxies. The exception might consist of a single user using a proxy for a single website. However, this case is arguably of little interests as it cannot represent one of severe Internet abuse. Further, we consider the statistics of the payload sizes for the first few packets of the flows. The main idea here is that we can take advantage of the peculiar way in which proxies use HTTP. In particular, it is informative the fact that the proxy server redirects the client several times, and that, because of the intrinsic delay in the operation (i.e., the server has to fetch the page and rewrite the HTML), pure TCP acknowledgments are sent in a distinctive way.

Our methodology consists of four stages: **(a)** service identification, **(b)** server profiling, **(c)** proxy identification, and **(d)** host cache management.

The first stage aims at identifying HTTP services for both the standard and secure flavors. This is needed as the proxy server might be running on a non standard port number (allowed by firewall). Existing techniques can be leveraged for this task (e.g., those in [1]).

Within the second stage we focus on creating a profile for each identified HTTP service. The profile is derived from the statistics of flow features measured from each flow towards the service. The features are based on the packets' inter-arrival times and payload sizes. A profile consists of the average and standard deviation of these features. The significance of a profile clearly depends on the number of flows that we can observe, but we assume a proxy will be the destination of many user requests.

The third stage is the classification of the services into proxy or non proxy. A number of ML algorithms (both supervised and unsupervised) can be applied to this problem. In our case, we firstly opt for an unsupervised technique: the K-means algorithm.

Finally, there is a practical need of storing, updating and deleting service profiles. This is the scope of the fourth stage (omitted due to page limit).

4. EARLY RESULTS

We experimented with two proxies: *guardster.com* and *anonymouse.org*. We recorded the traffic to browse a dozen of popular websites using direct connection and through the proxies[3], reaching a total of 81 servers. Then we extracted the flow features and computed the server profiles for every server with at least 5 flows. Figure 1 compares the distribution of the first response time within each connection for proxies against websites. We applied the K-means algorithm to group the profiles into two clusters, using the mean and standard deviation of request sizes, and standard deviation of response times. This simple technique is already leading to promising results: the profiles of the 6 proxy servers are clustered together (100% accuracy).

[3]1 server for anonymouse and 5 servers for guardster.

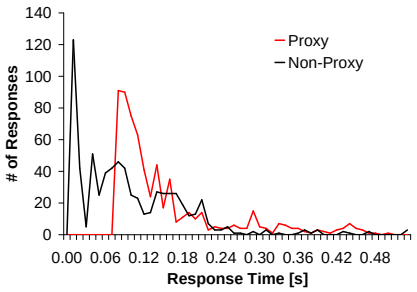

Figure 1: Distributions of the fit response time for proxies and websites.

5. CONCLUSIONS

We have introduced a novel methodology for identifying proxies and shown interesting preliminary results. We recognize more data is needed in order to generalize. However, we plan to include more results in the final version of this work.

6. REFERENCES

[1] M. Canini, W. Li, A. W. Moore, and R. Bolla. GTVS: Boosting the collection of application traffic ground truth. In *PAM*, 2009. In submission.

[2] M. Griffiths. Internet abuse in the workplace: Issues and concerns for employers and employment counselors. *Journal of Employment Counseling*, 2003.

[3] J. J. Johnson and K. W. Chalmers. Identifying employee internet abuse. In *Hawaii International Conference on System Sciences*, 2007.

[4] K. Siau, F. F. Nah, and L. Teng. Acceptable internet use policy. *Commun. ACM*, Jan. 2002.