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 a server behavior model to identify potential proxies within the observed traffic.

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, MySpace and Youtube 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 analyzes based on 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 rigid. Several studies have been conducted in which employees self-reported behavior that could be considered as Internet abuse [2]. Unfortunately, self-reporting is neither reliable nor effective. Johnson and Chalmers [2] 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.

Traditionally, URL or IP filtering have been adopted to enforce acceptable Internet use policies [3]. Unfortunately, these techniques can be easily circumvented with the use of anonymous Web proxies, especially when the traffic is encrypted using HTTPS.

Web proxies are special sites that allow the users to browse anonymously other Web sites. In most settings, access to these sites is not restricted. Despite these sites giving the opportunity for unconstrained Internet misuse, they might exist for more malicious reasons, e.g., 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 (e.g., glype, PHProxy). 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 wishes 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 go through the proxy server. In addition, some proxies also include a new navigation bar (with the URL input box) and advertisements in the final HTML. Clearly, this rewrite process introduces a delay, albeit small, which sums to the delay of making the request through the proxy. 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.

We carried out a trial of several proxy servers. From a well known proxy list (`http://www.kortaz.com/`), we picked the top 10 proxies by popularity. Unsurprisingly, most of them run the same HTTP proxy script, glype (`http://www.glype.org/`). We found that the total delay is typically in the order of several seconds which is obviously noticeable and affects the browsing experience. Of course, it also depends on the server load an location. Many proxies use local caches to reduce the delay of previously fetched contents, if these are cacheable.

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

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

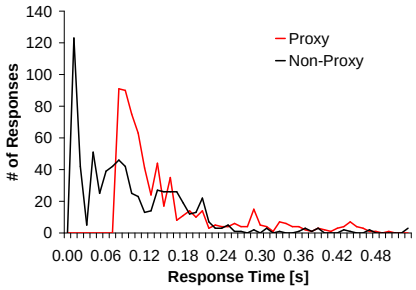

Figure 1: Histogram of the first response time for proxies and normal sites.

During our trial, we recorded packet traces. The analysis of these traces shows that the browser first makes a POST request to which the proxy servers respond with the HTTP temporary redirection code. Only the second browser requests actually triggers the remote procedure that fetches the requested page from the target site, introducing another delay. However, it is possible that also the target site uses browser redirection, typically for POST requests or for serving certain dynamic contents in Web 2.0 applications. In response to these redirections, we observe that the proxies simply generate another redirect response for the user browser and slow down the communication by another round trip.

One factor that further complicates the identification of these proxies is that the specific mechanics of the communication between browsers and proxy servers appear to depend on the implementation (or the HTTP server configuration): in some cases, all the HTTP requests to access a single URL reuse the same TCP connections. Conversely, for other proxies, the server closes the connection after each request during the redirection procedure. Therefore, multiple TCP connections are established to fetch the page. We envision these can be correlated (e.g., using the approach in [4]) so that more meaningful information than the single connection information can be provided to the identification method.

3. METHOD

The underlying idea of our approach is to create a server behavior model using recorded network traffic containing both proxy and normal HTTP server activities. In essence, we want to train a classifier that embeds the knowledge of the normal server behavior contrasted with the proxy server behavior. The classifier can then be used to identify proxy servers in live network traffic through passive measurements.

Each server profile is derived from certain flow features measured from all flows toward the server. The features are based on the characteristics of the packets that make up the flows, without looking at the payload content as it might be encrypted. Drawing inspiration from previous works that focus on flow classification (e.g., [5, 6]), we consider the packet sizes and the inter-packet arrival times from the first few packets of each flow (e.g., 10). This feature set allows us to determine size, duration and inter-time of the HTTP requests and responses which, as observed before, have a specific behavior for proxy servers. For instance, Figure 1 illustrates the difference of the first response time between proxies and normal sites.

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 observe, but we can exclude those with small significance (i.e., only use servers with at least N flows) as we assume a proxy will be the destination of a sufficiently high number of user requests. During the offline training phase, a number of machine learning algorithms (both supervised and unsupervised) can be used to build server behavior models that are able to classify servers into proxy or non proxy. In our case, we firstly opt for an unsupervised technique: the K-means algorithm.

Once a server behavior model has been created, it is deployed in a network probe located on the Internet access link. The probe monitors HTTP traffic and collects the features from the observed flows. When the number of flows for a certain server reaches a threshold, it computes the server profile and uses the model to identify whether this server is a proxy. To establish with higher confidence the nature of a given server, one might consider to test a server multiple times before taking the final decision. However, this involves measuring multiple independent profiles of the same server which might require to wait for a comparatively longer period. The result can then be sent off to the network administrator for validation or can trigger an automatic update on the firewall to immediately terminate all flows involving the proxy and prevent further connections.

Given that only packet headers are used, the network probe can be built on commodity hardware as this is sufficient to cope with current access link speeds, even up to 1 Gbps [7]. Further, the approach is easily scalable by distributing the load on multiple network probes using flow hashing techniques [8].

4. CONCLUSION

We have introduced a novel method for identifying anonymous Web proxies that relies on the specific server behavior rather than payload inspection. We have started to collect training data using free proxies available on the Internet. Some preliminary experiments have given us promising results and we plan to include them in the final poster.

5. REFERENCES

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

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

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

[4] J. Kannan, J. Jung, V. Paxson, and C. E. Koksal. Semi-automated discovery of application session structure. In *Proceedings of IMC'06*, 2006.

[5] L. Bernaille, R. Teixeira, and K. Salamatian. Early application identification. In *Proceedings of CoNEXT'06*, pages 1–12, Dec 2006.

[6] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic classification through simple statistical fingerprinting. *SIGCOMM Comp. Comm. Rev.*, 37(1):5–16, Jan 2007.

[7] L. Deri. Passively monitoring networks at gigabit speeds using commodity hardware and open source software. In *Proceedings of PAM'03*, 2003.

[8] F. Schneider, J. Wallerich, and A. Feldmann. Packet capture in 10-gigabit ethernet environments using contemporary commodity hardware. In *Proceedings of PAM'07*, Apr 2007.