PPI: Towards Precise Page Identification for Encrypted Web-browsing Traffic

Zhenlong Yuan*, Yibo Xue† and Wei Xia‡
*Department of Automation, Tsinghua University, China
†Tsinghua National Lab for Information Science and Technology, China
‡School of Control and Computer Engineering, North China Electric Power University, China
yuanzl11@mails.tsinghua.edu.cn, yiboxue@tsinghua.edu.cn, xiawei@ncepu.edu.cn

ABSTRACT

Precise Web page identification has always been a research hotspot in the areas of network management and security. However, previous works generally focused on statistical or probabilistic approaches and could not exactly calculate the length of encrypted data under different conditions, which makes them hardly cover all the cases. In this poster, we propose an exact fingerprint derivation method for encrypted Web-browsing traffic and thereby implementing a prototype system for precise page identification (PPI). Our experiments show that PPI not only can be employed for early page identification at individual-flow level but also can achieve very high accuracy at aggregate-traffic level.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations

Keywords

Page Identification, Fingerprint Derivation, PPI System

1. INTRODUCTION

The World Wide Web has become the largest resource for information retrieval and thus causes great concerns about the possibility to identify certain pages from Web-browsing traffic. However, due to the rise of cloud computing platforms such as Windows Azure and Amazon EC2, more and more Web sites are tending to support HTTPS access and render the same IP group for providing various Web services. As a result, it is not able to distinguish the specific Web pages based on string matching or IP addresses any more.

Several projects [1–4] have focused on using the size of page objects to match encrypted traffic through statistical or probabilistic approaches. However, different conditions will result in different negotiable parameters (e.g. cipher suite, SSL/TLS version number and compression algorithm) and these approaches actually can not cover all the cases. In this poster, different from them, we propose a novel fingerprint derivation method that can derive the exact data length for encrypted Web-browsing traffic. Furthermore, based on that method, a prototype system for precise page identification (PPI) is implemented. The experimental results show that the PPI system not only can be employed for early page identification at individual-flow level but also can achieve very high accuracy at aggregate-traffic level.

2. METHOD

Figure 1 shows the fingerprint derivation process for HTTPS traffic. Firstly, assuming the target Web pages are publicly available, thus we can obtain the actual size of every target object in the page, including HTML code and other embedded objects.

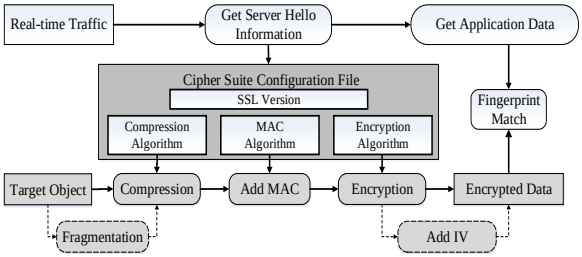

Figure 1: Fingerprint Derivation Process

Then during the handshake, as the compression algorithm negotiated can be obtained from the Server Hello message in the cleartext, so we can get the exact size of compressed data by calling the corresponding compression algorithm such as GZIP to process the target object data. In a similar way, the specified SSL/TLS version number, MAC and symmetric algorithms can also be obtained from the Server Hello message. Suppose L_{comp} is the length (in bytes) of the object raw data (include the server response header) compressed using the compression algorithm defined in a session, L_{mac} is the length of MAC, and L_{block} is the block size of the block cipher defined in the cipher suite. Note that we have prepared a list of data values (L_{mac} and L_{block}) for all the cipher suites, thus we are able to derive the encrypted data length of target object as follows:

For SSL2.0, SSL3.0 or TLS1.0:

$$L_{target} = \begin{cases} L_{comp} + L_{mac} & Stream\ cipher \\ \left\lceil \frac{L_{comp}+L_{mac}+1}{L_{block}} \right\rceil * L_{block} & Block\ cipher \end{cases}$$

For TLS1.1 or TLS1.2:

$$L_{target} = \begin{cases} L_{comp} + L_{mac} & Stream\ cipher \\ (\left\lceil \frac{L_{comp}+L_{mac}+1}{L_{block}} \right\rceil + 1) * L_{block} & Block\ cipher \end{cases}$$

The value of derived data length are considered as the fingerprint of target Web page. Nevertheless, there may be a fragmentation process in SSL/TLS record protocol, so

the encrypted data might be divided into multiple records. In this case, the fingerprint of target page changes to a sequence length that indicates the lengths of consecutive records. During the application message exchange, we can compare the fingerprint with the captured SSL/TLS encrypted data length.

The approach of fingerprint derivation can be utilized to identify the target Web page which has a "unique" object size. More than that, if we take the HTML code as the target object to derive the fingerprint, then the method would have the following advantages: i) Since HTML code is downloaded earlier than any other embedded object, we could have an early identification, which is vital for timely blocking harmful information. ii) Our approach can identify without a deep packet inspection, which is essential for efficient real-time page identification. iii) Our approach can deal with unidirectional 5-tuple flow, which is crucial to an intermediate router that are hard to gather the aggregate-traffic.

3. ARCHITECTURE

The PPI system is designed as shown in Figure 2. The hierarchical structure is introduced for step-by-step classification and thus reduce the pressure on the next module that is responsible for more precise identification.

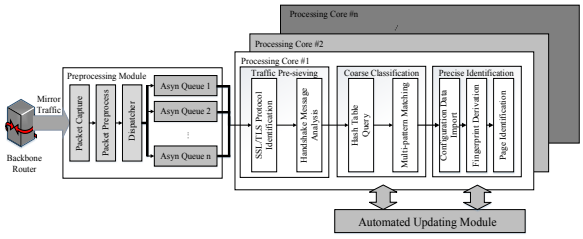

Figure 2: The Architecture of PPI System

The proposed system includes five modules: i) Preprocessing module. This module captures and preprocesses packets, and dispatches preprocessed packets into asynchronous queues for parallel processing. ii) Traffic pre-sieving. As we know, HTTPS uses SSL/TLS for its transport, thus overall HTTPS traffic are mixed in common SSL/TLS traffic. In light of that, we firstly consider picking all the SSL/TLS traffic out from the background traffic. iii) Coarse classification. This module is aiming at classifying the incoming SSL/TLS traffic into different Web Service providers (e.g. Amazon and Wikipedia) at a coarse-grained level according to their IP address or server certificate. iv) Precise identification. This module is used to identify encrypted Web-browsing traffic based on the fingerprint derivation method proposed in Section 2. v) Automated updating. This module is responsible for updating the fingerprints of Web pages in our databases.

4. EVALUATION

For the experiments, we took 100000 pages from Wikipedia. These pages are randomly selected using the special hyperlink: https://en.wikipedia.org/wiki/Special:Random. Figure 3 shows the identification accuracy of the 100000 Wikipedia pages at two different levels. Particularly, in the aggregate-traffic level experiments, the fingerprint is derived

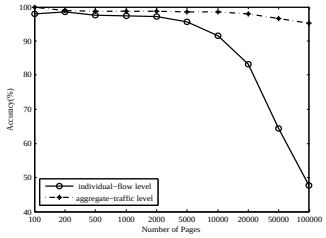

Figure 3: Wikipedia Page Identification

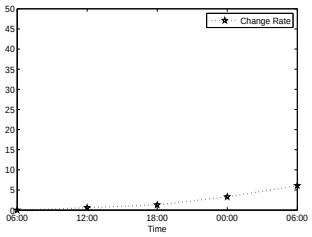

Figure 4: Rate of Pages Change over Time

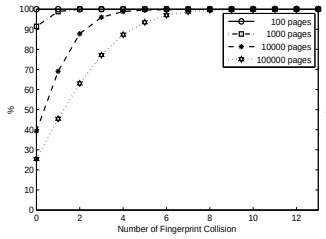

Figure 5: CDF of Fingerprint Collision at Individual-flow Level

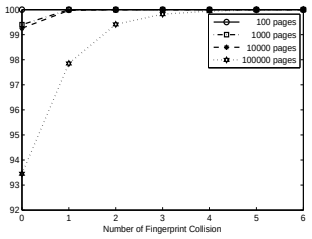

Figure 6: CDF of Fingerprint Collision at Aggregate-traffic Level

by both the HTML size and the total length of the embedded objects, while in the individual-flow level experiments, we only use the HTML size to derive the fingerprint. We can see that the accuracy always maintains at a high level while dealing with aggregated traffic. Figure 4 shows the change rate for 10000 Wikipedia pages during 24 hours. It is noticeable that the HTML size of the tested Web pages is almost exactly the same as time goes by, suggesting that the fingerprint might be robust enough to identify the Web pages for a comparatively long period of time. Figure 5 and Figure 6 respectively show the Cumulative Distribution Function (CDF) of fingerprint collision for Wikipedia pages at different levels. As we can see from the two figures, the extent of fingerprint collision is not so serious, especially for the aggregate-traffic level or Web pages in a small scale, which has reconfirmed the effectiveness of the PPI system.

5. ACKNOWLEDGMENT

This work was supported by the National Key Technology R&D Program of China under Grant No.2012BAH46B04.

6. REFERENCES

[1] Heyning Cheng and Ron Avnur. Traffic analysis of ssl encrypted web browsing. *URL citeseer. ist. psu. edu/656522. html*, 1998.

[2] Qixiang Sun, Daniel R Simon, Yi-Min Wang, Wilf Russell, Venkata N Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings of IEEE Symposium on Security and Privacy (S&P'02)*, 2002.

[3] Liming Lu, Ee-Chien Chang, and Mun Choon Chan. Website fingerprinting and identification using ordered feature sequences. In *Computer Security–ESORICS 2010*, pages 199–214. Springer, 2010.

[4] George Danezis. Traffic analysis of the http protocol over tls. *Unpublished draft*, 2010.