Locating Tor hidden services through an interval-based traffic-correlation attack

Juan A. Elices
University of New Mexico
jelices@ece.unm.edu

Fernando Pérez-González
University of Vigo
fperez@gts.uvigo.es

I. Introduction

Lately, research on low-latency communication systems has mainly focused on sender anonymity in order to bypass censorship. But we cannot disregard server anonymity, as it is a necessity to guarantee freedom-of-speech in many countries. For instance, the Electronic Frontier Foundation and Reporters Without Borders advise the use of Tor hidden services (HSes) to protect the safety of dissidents.

Unfortunately, anonymous servers can also be used for less ethical purposes, such as distributing child pornography, drug traffic, and supporting terrorism. Tracking the location of these HSes has become a serious concern for law enforcement agencies. For example, Silk Road (an online black market) is known to be operative since February 2011, but as of today, is still functional.

Most of the previous work to locate HSes, [1], [2], [3], needs the attacker (AT) to become the first hop of the rendezvous circuit established by the HS. Therefore, these attacks can be easily bypassed by the HS controlling its entry guards, (relays that can be used as entry points), and even under the default entry guard selection they need a large amount of time.

Zander and Murdoch [4] proposed an attack based on the drift of the HS clock depending on the amount of requests. This method has two drawbacks: the need of another application on the HS to measure timestamps, and being detectable. To the best of our knowledge, the only method that directly correlates TCP traffic is [5] that looks for a packet around a predicted time, using just one packet per HTTP request.

In this paper, we propose a method to locate HSes by correlating the increment of TCP traffic that suffers the link between the HS and its entry guard.

II. Proposed Method

A. Threat Model

We have a client-server architecture, where the server is a HS. The primary goal of the attacker (AT) is to link a pseudonym (.onion) to the operator's real IP address.

We assume that the HS is really concerned about its anonymity, so it selects just one trusted entry relay (ER). The HS also monitors the traffic, looking for deanonymizing attacks, in case it notices an attempt it stops providing the service. Hence, the attacker has to completely replicate a common user's behavior.

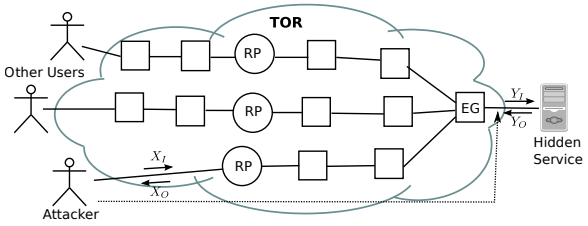

Fig. 1. SystemModel

We assume that the AT is able to obtain the time and size of the packets sent between the HS and the ER. This can be a reasonable assumption if the AT is an ISP on the path between the HS and the ER as most of their networking equipment is able to monitor the traffic, for instance, using Netflow.

B. System Model

The system model is shown in Figure 1, where an AT connects to the HS through a one-hop Tor circuit to the rendezvous point (RP), to reduce the latency variability.

Our application deals with two bidirectional flows: the flow from the AT to RP with timing information $X_Z = \{X_{Z,1}, \ldots, X_{Z,M_{X_Z}}\}$, where Z denotes the direction that can be I (Input to the HS) or O (Output from the HS). Similarly, we have the suspect flow with timing information $Y_Z = \{Y_{Z,1}, \ldots, Y_{Z,M_{Y_Z}}\}$. We convert both packet flows into cell flows by considering a cell every 586 bytes of TCP traffic.

The goal of the attacker is to correctly decide whether Y carries the flow X. Formally, we can define the following statistical hypotheses: H_0: Y does not carry the data cells of X and H_1: Y carries the data cells of X.

We consider the following intervals: $PC_{Z,j} = [\alpha_Z + (j-1)T_Z, \alpha_Z + j \cdot T_Z],\ j = 1, \ldots, N_Z$, used at AT side, and $PS_{Z,j} = [\beta_Z + (j-1)T_Z, \beta_Z + j \cdot T_Z],\ j = 1, \ldots, N_Z$, used at HS side, where α_Z denotes the instant we start counting at AT, β_Z the moment we start counting at the HS, N_Z represents the number of considered intervals, and T_Z the interval length.

We define the sequences C_Z and S_Z as the number of cells that appear in each interval, as follows:

$$C_{Z,j} = \sum_{k=1}^{M_{X_Z}} \mathbb{1}_{PC_{Z,j}}(X_{Z,k}) \text{ and} \tag{1}$$

$$S_{Z,j} = \sum_{k=1}^{M_{Y_Z}} \mathbb{1}_{PS_{Z,j}}(Y_{Z,k}),\ j = 1, \ldots, N_Z \tag{2}$$

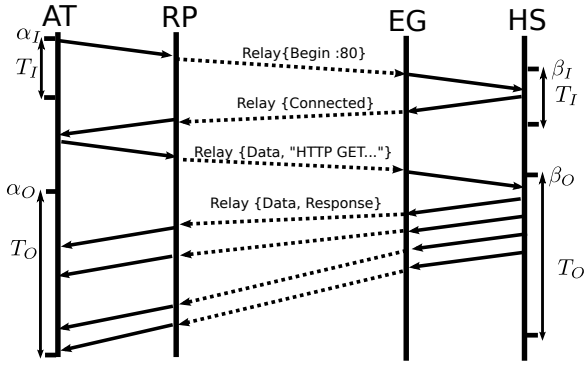

Fig. 2. Cell Sequence of an HTTP request-response to a HS

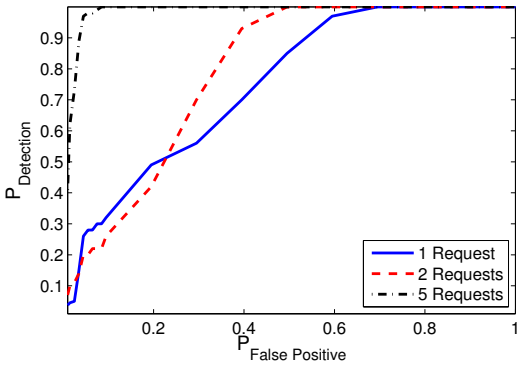

Fig. 3. Experiment Performance

where $\mathbb{1}_A(x)$ is the indicator function of $x \in A$, and HS traffic can have either one of two sources: AT or different flows. Therefore, $S_Z = C'_Z + R_Z$ where C'_Z represents the number of packets in each interval that come from AT and R_Z from other flows.

C. Detector

In order to obtain the best possible performance, we base our detector in the likelihood ratio test. Hence, we decide that the eavesdropped flow correspond to the HS (H_1) if

$$\Lambda(X,Y) = \sum_{j=1}^{N_I} log\left(\frac{f_{S_{I,j}|C_I,H_1}(s_{I,j}|c_I)}{f_R(r_i)}\right) + \sum_{j=1}^{N_O} log\left(\frac{f_{S_{O,j}|C_O,H_1}(s_{O,j}|c_O)}{f_R(r_O)}\right) > \eta \quad (3)$$

and H_0 in the opposite case, where η is a threshold to achieve a certain probability of false positive.

O_Z can be modelled as a negative binomial distribution with parameters that can obtained through maximum likelihood estimation using the HS traffic prior to AT sending its request. Empirically we obtain the best results using the previous 180 seconds. We model C' as a sum of binomial distributions as follows: $C'_{Z,j} \sim \sum_{k=0}^{N_Z} \text{Bin}(C_{Z,j-k}, \Pr(S_{j-k}))$, where S_n represents the event that a cell shifts n intervals.

III. Web Server Location Implementation

When a client demands a HS web page, before sending any data, AT and HS build each a circuit to the RP, as explained in [6]. After these two circuits are connected the cell sequence is shown in Figure 2. First, AT sends a relay begin cell that HS responds with a relay connected cell, afterwards AT and HS communicate according to the HTTP protocol.

Due to the bulk traffic nature of HTTP, the cell delays in each direction are very different. Cells from AT to HS (I) suffer in general little queuing delays, however cells from HS to AT (O) suffer a queuing time that varies within the position inside the burst. This implies that delays of O are not identically distributed. To avoid the modelling problem that this generates, we choose T_O the time to receive the whole page, i.e. $N_O = 1$, so no shift is possible. The rest of parameters are chosen as follows: α_I the moment where the relay begin cell is sent, $\beta_I = \alpha_I + \min(\text{RTT})/2$, where RTT is the round trip time, $\min(\text{RTT}) \approx 400$ ms, $\alpha_O = \alpha_I + 2\min(\text{RTT})$ and $\beta_O = \alpha_I + 1.5\min(\text{RTT})$. The interval length for the I direction, i.e. T_I, is fixed to the standard deviation of RTT under no burst traffic, that is approximately 270 ms.

In order to validate our proposal, we carried out an experiment on the live Tor network. We launched a hidden web server that replicates web pages from Silk Road. Besides CL, 10 different machines are requesting the web page according to a Poisson model with rate 50 web requests per hour each one. We captured the traffic on both ends with `tcpdump`. We repeated the experiment 100 times obtaining the results depicted in Figure 3, where we can observe that 5 web page accesses is enough to accurately locate a HS.

IV. Future Work

We have proposed a method to locate a hidden server that only accesses to TCP traffic, carrying out a real experiment using intuitive parameters to prove the feasibility of the method. We have not studied the influence of the parameters, leaving it for future work. Also, since the ISP (above all tier-1 ISPs) may store only sampled packet data, due to the immense amount of information they transmit, the study of this method under it will be addressed in the future.

References

[1] L. Øverlier and P. Syverson, "Locating hidden servers," in *Security and Privacy, 2006 IEEE Symposium on*, may 2006, pp. 15 pp. –114.

[2] A. Biryukov, I. Pustogarov, and R.-P. Weinmann, "Trawling for tor hidden services: Detection, measurement, deanonymization," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, ser. SP '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 80–94.

[3] Z. Ling, J. Luo, K. Wu, and X. Fu, "Protocol-level hidden server discovery," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1043–1051.

[4] S. Zander and S. J. Murdoch, "An improved clock-skew measurement technique for revealing hidden services," in *Proceedings of the 17th conference on Security symposium*, ser. SS'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 211–225.

[5] J. A. Elices and F. Perez-Gonzalez, "Fingerprinting a flow of messages to an anonymous server," in *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*. IEEE, 2012, pp. 97–102.

[6] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13*, ser. SSYM'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 21–21.