Detecting Malware-infected Hosts Using Templates of Multiple HTTP Requests

Taiga Hokaguchi[1], Yuichi Ohsita[1], Toshiki Shibahara[1,2], Daiki Chiba[2], Mitsuaki Akiyama[2], Masayuki Murata[1]
[1]Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
Email: {t-hokaguchi, y-ohsita, murata}@ist.osaka-u.ac.jp
[2] NTT Secure Platform Laboratories, Tokyo, Japan
Email: toshiki.shibahara.de@hco.ntt.co.jp, {daiki.chiba, akiyama}@ieee.org

***Abstract*—In this paper, we propose a method for detecting malware-infected hosts with a high rate of detection and a low rate of false positives without using any data on benign communication. Based on the fact that many malware-infected hosts generate multiple HTTP requests, we propose a method using the templates of sets of those HTTP requests. For each malware, this method generates a template that comprises the set of templates of the HTTP requests that the malware generates. We call the set of templates *group template*. It then detects malware-infected hosts by comparing the set of monitored HTTP requests with the group templates.**

***Index Terms*—Malware, Detection, Bot, Template**

I. Introduction

Malware-infected hosts constitute one of the most serious threats in network services. An attacker controls many malware-infected hosts via command and control (C&C) servers to carry out cyber attacks. One approach to detecting malware-infected hosts is to use blacklist the of known malicious domains including C&C servers. However, attackers frequently change the domains of the C&C servers to avoid detection by the blacklist.

Methods have also been proposed that use templates of C&C communication instead of blacklists [1]. Many botnets use the Hypertext Transfer Protocol (HTTP) as their C&C protocol to avoid being blocked by a firewall. Consequently, these methods generate templates of the HTTP requests of malware-infected hosts and then monitor the HTTP requests sent from the monitored network to detect any C&C communication based on template similarity. Of course, some benign HTTP requests are similar to the templates of the HTTP requests of malware-infected hosts. To consider such benign requests, researchers proposed methods using the benign requests monitored in the environment where the methods are deployed. The current methods for avoiding the misdetection of benign traffic require a sufficient amount of benign traffic to have been monitored. That is, the current methods cannot work accurately soon after deployment, and it takes time to achieve accurate detection.

In this paper, we propose a method that detects malware-infected hosts with a high rate of detection and a low rate of false positives without using any data on benign communication.

II. Detection of Malware-infected Hosts Using Templates of Multiple HTTP Requests

Based on the fact that most malware-infected hosts generate multiple HTTP requests, our method is based on templates for sets of multiple HTTP requests. For each malware, this method generates a template (called the group template) that comprises the set of templates of the HTTP requests that the malware generates. The multiple benign requests sent within a short time period do rarely match the HTTP requests sent by malware. That is, the probability that the set of benign traffic matches the group templates is low. In our system, a *group template T* is defined by the set of *single templates U_T* that match the HTTP requests sent by the malware-infected host and the number of matched HTTP requests for each single template t_i ($i \in U_T$).

A. Generating the Group Template

Using the malware traffic captured in the sandbox running malware-infected hosts, our method begins by generating the single template in the same way as does an existing method. To generate the single template, any template generation method can be used.

Then, based on the generated single template, we generate the group template by using the same malware traffic. A group template is generated by taking the following steps for each malware-infected host. First, for each HTTP request sent by the host, we select the single template whose similarity with the request is the highest by using the similarity metric defined for the single template. The selected templates are added to U_T, whereupon we count the number of matched HTTP request t_i for $i \in U_T$.

B. Detection

1) Generating Groups of HTTP Requests: We begin by dividing the time series of HTTP requests into HTTP request groups, which are compared with the group templates. In this paper, we divide the HTTP requests based on time.

2) Single-template Matching: Before using the group template, we use the single template to detect malware-infected hosts based on the matching score $Score(h, t)$, which is defined based on the used single template generation method. If $Score(h, t) \leq \theta_L$ for all HTTP requests in an HTTP request group, then that group is deemed benign because none of the HTTP requests match any single template. However, if $Score(h, t) \geq \theta_H$ for any of the HTTP requests in an HTTP request group, the group is deemed malicious because there are HTTP requests that exactly match the features of the HTTP requests generated by the malware-infected hosts. Otherwise, we perform group-template matching.

3) Group-template Matching: We detect the malware-infected hosts by comparing the HTTP request groups with the group templates. To do so, we define the matching score $S(D, T)$ between HTTP request group D and group template T, and malware-infected hosts are detected when $S(D, T)$ for one of the group templates exceeds the predefined threshold θ_G.

Although $S(D, T)$ could be defined more sophisticatedly, we define $S(D, T)$ simply as

$$S\left(D, T\right) = 1 - \frac{1}{|U_T|} \sum_{i \in U_T} s\left(d_i, t_i\right), \tag{1}$$

$$s\left(d_i, t_i\right) = \begin{cases} \alpha & \left(d_i = 0\right), \\ \frac{\beta(t_i - d_i)}{t_i} & \left(0 < d_i \leq t_i\right), \\ 0 & \left(d_i > t_i\right). \end{cases} \tag{2}$$

Here, D is an HTTP request group, T is a group template, d_i is the number of HTTP requests in D whose score $Score(h, t)$ for single template i exceeds the threshold θ_L, and α and β are fixed parameters with $\alpha >> \beta$. In our evaluation, we set α to 0.8, and β to 0.5. As more single templates in group template T match the HTTP requests in HTTP request group D, so $S(D, T)$ increases.

III. Evaluation

A. Implementation of generation of single templates

Any template generation methods can be used for single template generation in our method. In this evaluation, we implemented our method using the BotProfiler [1] as the method to generate single templates.

B. Compared methods

In this evaluation, we compare the performance of the following methods.

Our method: Our method detects malware-infected hosts based on group templates.

BotProfiler: In this evaluation, we use the BotProfiler [1] as a method using only single templates. Though the BotProfiler uses the benign requests to avoid misdetections and cannot work in the case without any data on benign requests unlike our method, we show the performance of the BotProfiler in addition to our method to show the accuracy achieved by the existing method in this section.

BotProfiler without rarity profiling: The BotProfier requires a sufficient amount of benign traffic to have been monitored. We aim to achieve a high rate of detection and a low rate of false positives without using any data on benign communication. Therefore, we also compare our method with the BotProfiler when no benign traffic is monitored.

TABLE I: Dataset.

Label	Training		Testing	
	Period	# HTTP requests	Period	# HTTP requests
Malicious	2017/8/1 - 2017/12/31	656,714	2018/1/1 - 2018/3/31	442,532
Benign	2018/12/1 - 2018/12/31	291,343	2018/1/1 - 2018/3/31	876,778

TABLE II: TPR when parameters are set to make FPR less than 3%.

		TPR
BotProfiler without RP		86.18%
BotProfiler		87.17%
Our method	$\theta_H = 0.95, \theta_L = 0.40$	93.22%
	$\theta_H = 0.95, \theta_L = 0.80$	87.49%
	$\theta_H = 0.90, \theta_L = 0.80$	-

C. Data

Malware traffic was captured from the sandbox system [2] running malware samples. The sandbox supports executable files only in Microsoft Windows environments. We use the malware samples including PUP obtained from VirusTotal [3]. Benign traffic was captured in a university. We divide the malware samples and benign traffic into training data and testing data according to the date on which the sample was collected. Table I gives the numbers of malicious and benign HTTP requests. Note that the benign HTTP requests in the training data set are used only by BotProfiler and not by our method or BotProfiler without RP. In our method, HTTP requests are divided into HTTP request groups. In this section, we simply group the HTTP requests sent within 30 s from the first request into the same HTTP request group.

D. Results

As a simple way of comparing our method with BotProfiler and BotProfiler without RP, in Table II we compare the TPRs when the thresholds are set to make the FPR less than 3%. Our method with $\theta_L = 0.80$ and $\theta_H = 0.90$ cannot achieve an FPR less than 3%. Table II indicates that our method with $\theta_L = 0.40$ and $\theta_H = 0.95$ achieves the highest TPR.

IV. Conclusion

We proposed a method that detects malware-infected hosts with a high rate of detection and a low rate of false positives without using any data on benign communication. We implemented our method and evaluated it using real traffic data. However, the TPR and FPR both depend on the chosen parameter values. In future work, we will seek a method for setting suitable parameter values.

References

[1] D. Chiba et. al., "BotProfiler: Detecting malware-infected hosts by profiling variability of malicious infrastructure," *IEICE Transactions on Communications*, vol. 99, no. 5, pp. 1012–1023, 2016.

[2] K. Aoki et. al., "Controlling malware HTTP communications in dynamic analysis system using search engine," in *Proc. Third International Workshop on Cyberspace Safety and Security*, pp. 1–6, IEEE, 2011.

[3] "Virustotal-free online virus, malware and URL scanner," *Online: https://www.virustotal.com/*.