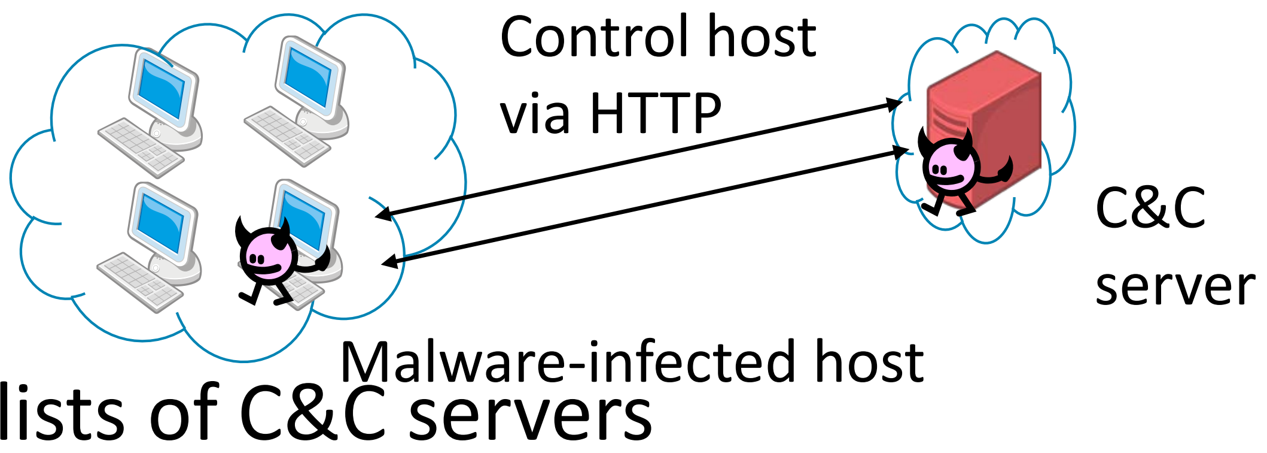
Detecting Malware-infected Hosts Using Templates of Multiple HTTP Requests

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
[2] NTT Secure Platform Laboratories, Tokyo, Japan

Detection of Malware Infected Hosts

Detecting malware-infected hosts by detecting a traffic from/to C&C server

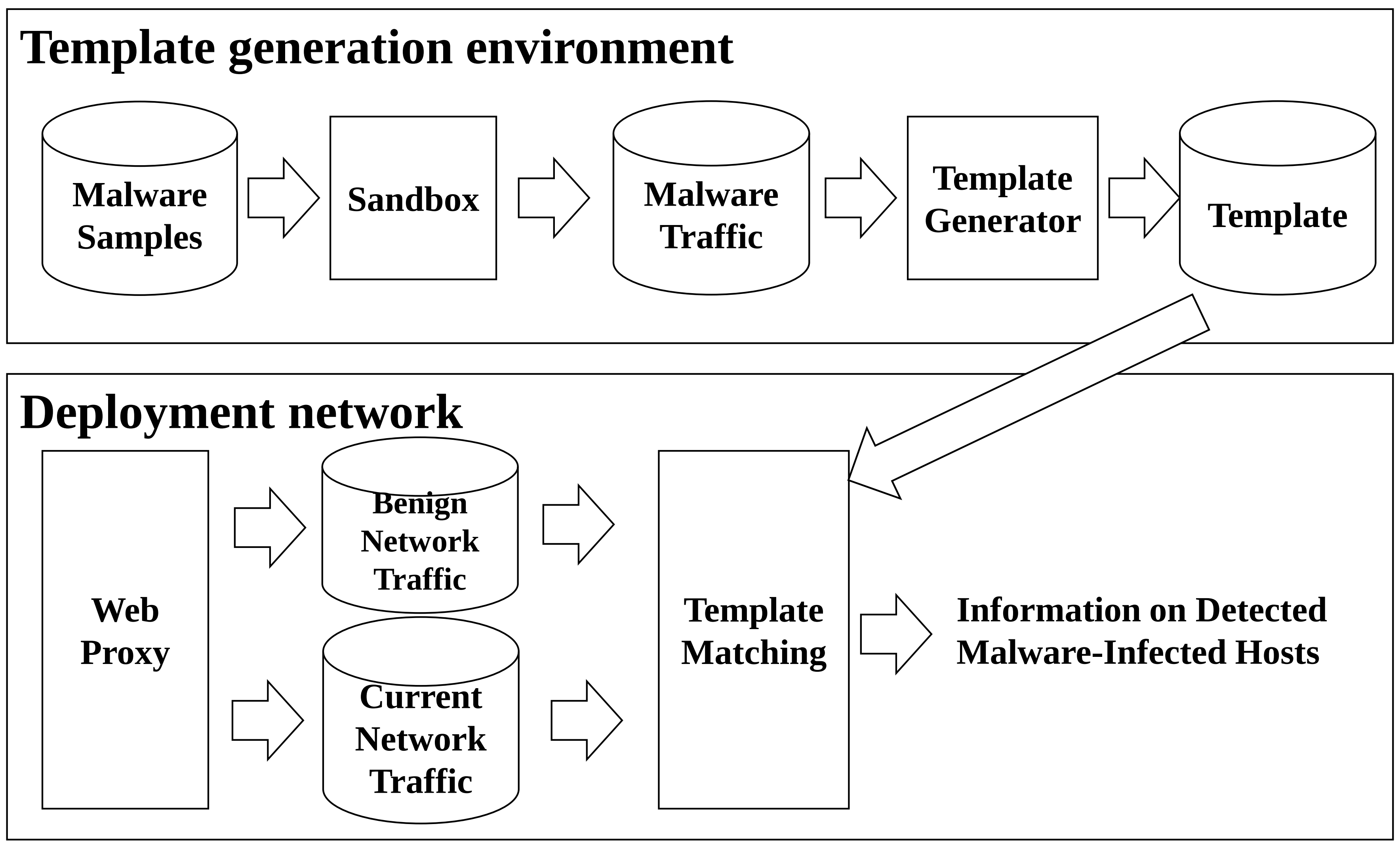


Approach1: Using blacklists of C&C servers
→Attackers frequently change the domain of C&C servers

Approach2: Using templates of C&C communication

Detection based on templates

Templates: Template of HTTP request send by malware infected hosts



Problem: A sufficient amount of benign traffic is required to avoid misdetection
→ This method cannot accurately detect malware-infected hosts soon after deployment

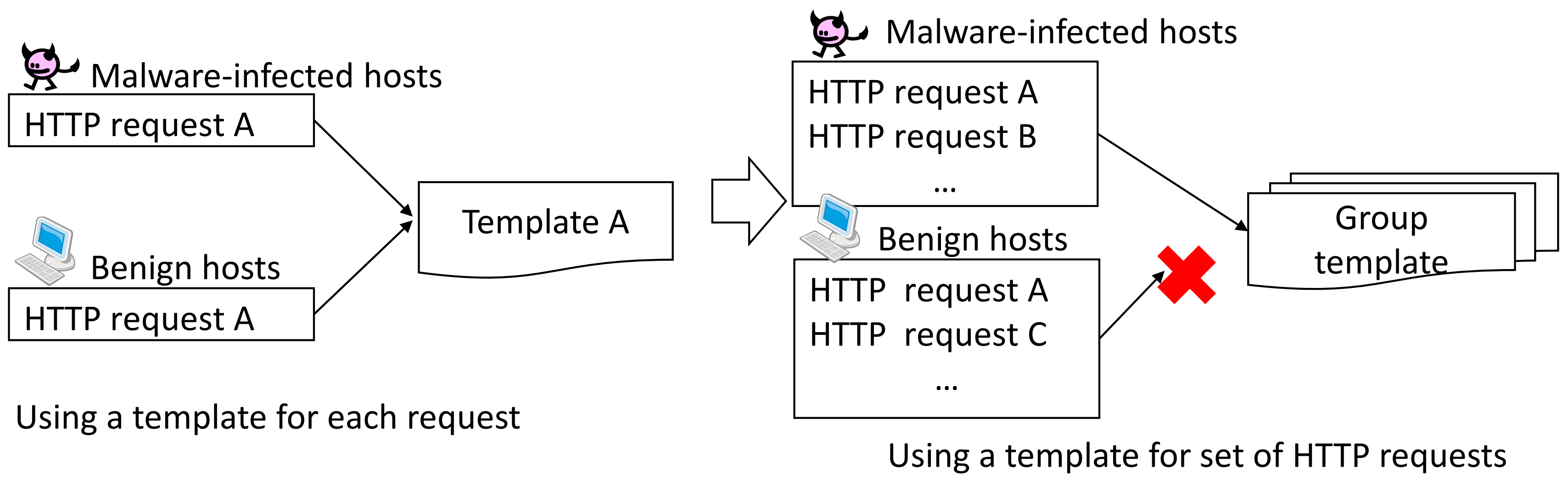
Goal

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.

Approach

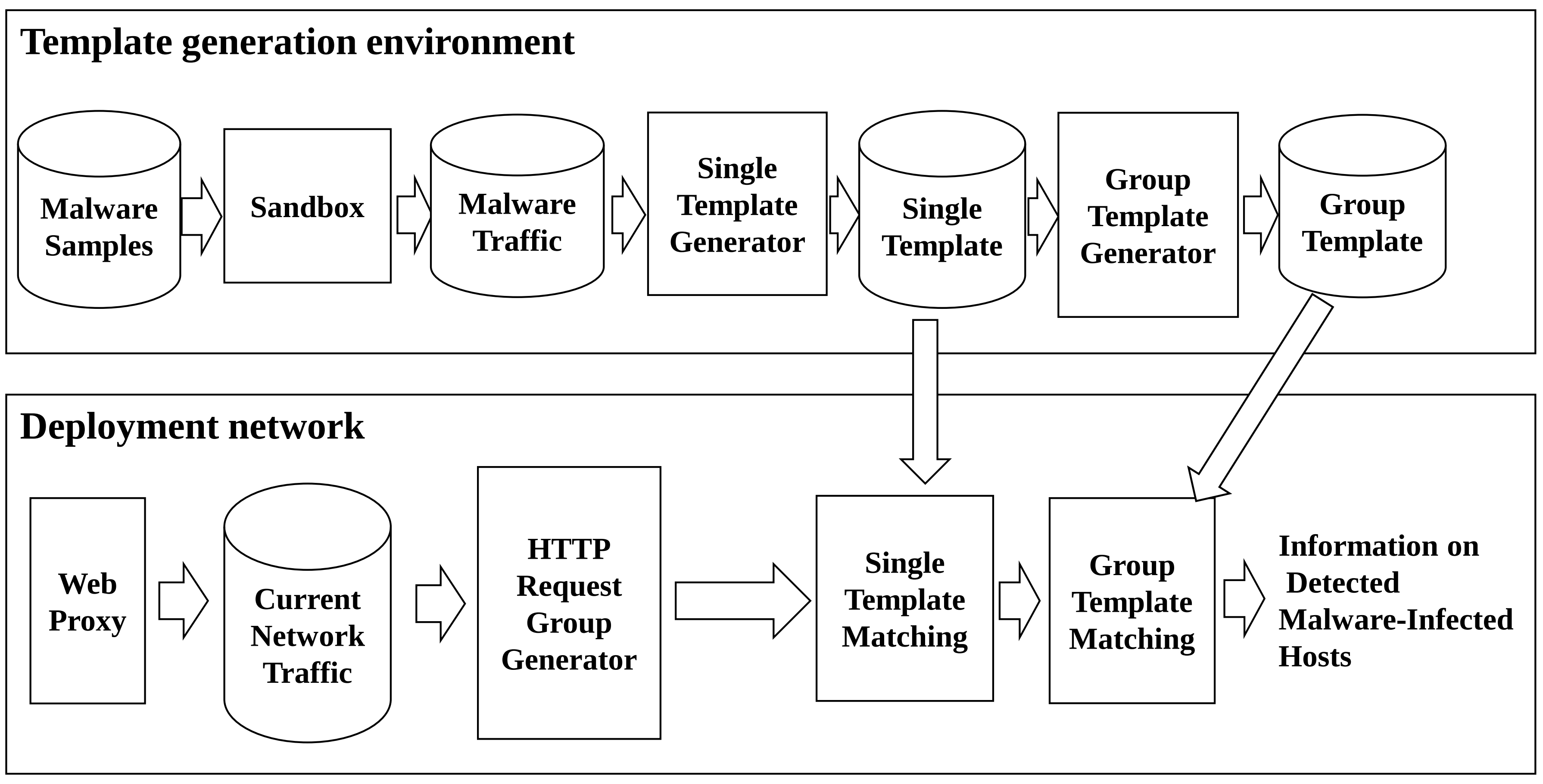
Use Templates for sets of multiple HTTP requests

- Most malware-infected hosts generate multiple HTTP requests
- multiple benign requests sent within a short time period do rarely match the HTTP requests sent by malware.



Our method

Detection using templates of multiple HTTP requests



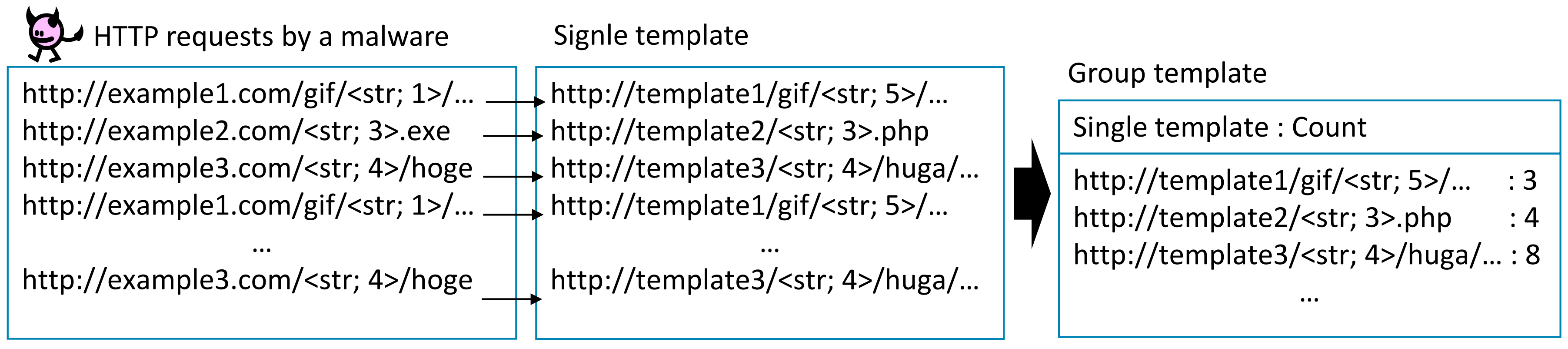
Generation of Single template

Any method to generate single template can be used

Generation of Group templates

Generate a group template for each malware

- For each HTTP request sent by the host, select the single template whose similarity with the request is the highest
- The selected templates are added to U_T, whereupon count the number of matched HTTP request t_i for $i \in U_T$.


HTTP requests by a malware
http://example1.com/gif/<str; 1>/…
http://example2.com/<str; 3>.exe
http://example3.com/<str; 4>/hoge
http://example1.com/gif/<str; 1>/…
…
http://example3.com/<str; 4>/hoge
Signle template
http://template1/gif/<str; 5>/…
http://template2/<str; 3>.php
http://template3/<str; 4>/huga/…
http://template1/gif/<str; 5>/…
…
http://template3/<str; 4>/huga/…
Group template
Single template : Count
http://template1/gif/<str; 5>/… : 3
http://template2/<str; 3>.php : 4
http://template3/<str; 4>/huga/… : 8
…

Detection

1. Single template matching:

If $Score(h,t) \leq \theta_L$ for all HTTP requests → benign
If $Score(h,t) \geq \theta_H$ for any of the HTTP requests→malicious
Otherwise, perform group-template matching.
*Any method to calculate $Score(h,t)$ can be used

2. Group template matching:

Matching score $S(D,T)$ between HTTP request group D and group template T exceeds the predefined threshold θ_G
→ malicious

Definition of $S(D,T)$:

$$\mathrm{S(D,T)} = 1 - \frac{1}{|\mathrm{U_T}|}\sum_{\mathrm{i \in U_T}} \mathrm{s(d_i,t_i)}, \qquad \mathrm{s(d_i,t_i)} = \begin{cases} \alpha & (d_i = 0) \\ \dfrac{\beta(t_i - d_i)}{t_i} & (0 < d_i \leq t_i) \\ 0 & (d_i > t_i) \end{cases}$$

Evaluation

Dataset:

Malware traffic: captured from the sandbox system running malware samples obtained from VirusTotal
Benign traffic: captured in a university

Result:

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%
	$\theta_H = 0.95, \theta_L = 0.80$	n/a