Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Owner based malware discrimination

Lansheng Han [a,*], Songsong Liu [a], Shuxia Han [b], Wenjing Jia [c], Jingwei Lei [d]

[a] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[b] School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan, China
[c] Faculty of Engineering and IT, University of Technology Sydney, Sydney, Australia
[d] Tencent Technology (Shenzhen) Co Ltd, China

HIGHLIGHTS

- Current malware discrimination is on URM that cannot discriminate targeted malware.
- URMO (Unlimited Register Machine of Owner) based malware discrimination is proposed.
- URMO is theoretically proved be able to discriminate the targeted malware.
- Malware samples and practical statistics are presented to show URMO is effective.

ARTICLE INFO

Article history:
Received 1 September 2015
Received in revised form
20 May 2016
Accepted 22 May 2016
Available online xxxx

Keywords:
Malware discrimination
Computability
Relativity of discrimination
Owner

ABSTRACT

A piece of malware code can be harmful in one's system but totally harmless in another's. In this paper, we point out that the detection of malicious code or software is actually a matter of discrimination which depends on the owners of the computer systems. We propose an owner based malicious software discrimination model, named as Unlimited Register Machine of Owners (URMO). First, we characterize and analyze the limitations of existing discrimination techniques in theory by using the discrimination model of Unlimited Register Machine (URM) and then move on to construct the URMO discrimination model by giving the two important elements of malicious behavior: an operation and the object of the operation. The relationship between an operation and the object of the operation is fundamental to solving the relativity of the discrimination problem about malice, which is also the advantage of the URMO model. Finally, by applying the model to discriminate real-world malware and comparing it with existing popular antivirus software, we demonstrate the effectiveness and superior performance of the URMO model.

1. Introduction

1.1. Technical difficulties

Malware, short for malicious software, is a kind of program code which purpose is to violate the target system's security policy, and as a direct consequence leads to information leakage, resource abuse and damage to the integrity and availability of the targeted system [1]. In general, malware includes viruses, worms and Trojans [2], each containing a large number of variants.

In 1980s, Fred Cohen, who pointed out the dangers of malware and limitations of defense, gave the famous assertion that there was no algorithm that could perfectly detect all viruses [3]. Defense against malware should focus on the specific signatures of malware since we cannot possibly find a general way to identify all malware. Researchers have gone great length to study malware detection [4,5] and have developed some useful detection techniques, which have somewhat succeeded in preventing the spread of malware. Currently, the common malware discrimination techniques [4] include Virus-signature [5] used to check the malware-identified signature code in targeted objects, Integrity [6] to check whether software is tempered or infected by malware, Behavior Block [7] to block some operation of software taking the present rules into consideration, and Heuristic Analysis [8] to analyze whether software contains malicious intention.

Although many techniques have been developed to detect malware, we still cannot cope with the rapid development of malware effectively. The reason is that new malware has relatively higher pertinence, and can be very similar to applications [9]. All

* Corresponding author.
E-mail addresses: hanlansheng@hust.edu.cn (L. Han), 1998010259@hust.edu.cn (S. Han), Wenjing.Jia@uts.edu.au (W. Jia), 1302855531@qq.com (J. Lei).

http://dx.doi.org/10.1016/j.future.2016.05.020
0167-739X/

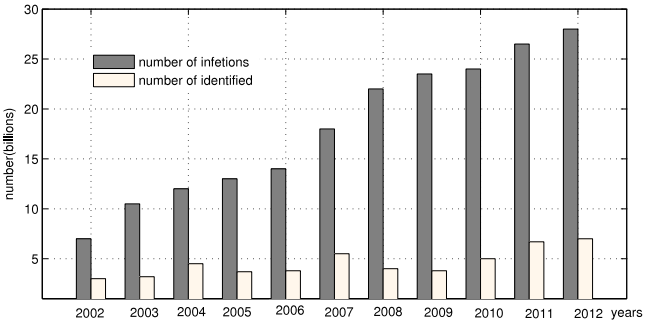

Fig. 1. Comparison between the number of infections and the number of identified malware from 2002 to 2012 in mainland China.

these make it harder to classify and extract signatures from them. For instance, Iran's nuclear facilities were attacked by Stuxnet [10,11], which was created just to damage those specific targets. According to the CERT—National Computer Network Emergency Response Technical Team/Coordination Center of China, over 90% of system failures are caused by malware or malware related incidents [12]. Fig. 1 shows the comparison between the number of infections and the number of identifications in mainland China for a period of nearly ten years. Here we can see the predicament of the current discrimination model.

Meanwhile, antivirus software is confronted with the problem of false positives and false negatives. For instance, some antivirus software that scan users' data package has been considered as Trojan by another antivirus software. So, we need a breakthrough on malware discrimination theory.

1.2. Relativity of discrimination theory

The first step of malware identification is malware discrimination [13]. However, there is a lack of reference and basis of discrimination. Different environments will magnify the problem of false positives and false negatives.

The basis of discrimination is a function f to calculate the received parameter $\boldsymbol{c}$ [14]. If $f(\boldsymbol{c})$ can give the result in expected range D, i.e., $f(\boldsymbol{c}) \in D$, we can take this procedure as a valid discrimination; otherwise, this procedure becomes invalid if there is no result or the result falls outside of D.

Nowadays, traditional antivirus software has come to a development bottleneck although all the known techniques have been utilized to detect malware [4], which means the composition of discrimination $f(\boldsymbol{c}) \in D$ is relatively stable. In order to break the bottleneck, we need to introduce a new reference to solve the relativity problem of discrimination that the same signature can have different impact on different users. Thus, it is important for the discrimination to introduce the relationship between the owner and software. In this paper, we propose an owner based malicious software discrimination model, named as Unlimited Register Machine of Owners (URMO).

1.3. Organization of this paper

The rest of this paper is organized as follows: Section 1 mainly talks about the predicament and challenge of current malware discrimination and introduces the idea of owner based discrimination. Section 2 depicts the discrimination mode of traditional antivirus software in theory and explains the origin of its limitations. Section 3 presents our owner based discrimination model URMO, and gives its theoretical proof. In Section 4, a real case is used to illustrate the validity of this model and its potential is also discussed. Section 5 concludes the paper and outlines the directions of future work.

Table 1
Notations used in this paper.

Notations	Explanations
Γ	Discrimination system consisting of programs
f	Discrimination function
$\boldsymbol{C}$	Signature sequence
x_i	Signature
$\aleph$	Discrimination rule
URM	Unlimited register machine
URMO	Unlimited register machine of owner
b	Result of discrimination, Boolean
R_i	Register
$a_1, a_2, \ldots, a_n$	Natural number sequence of signature
P	Discrimination program
$\rightarrow$	Give the result
$\downarrow$	Computation stops with the result
$\uparrow$	Computation never stops
$\Rightarrow$	Value tend to

2. Discrimination model of malicious software and its relativity

2.1. Abstract of malware discrimination mode

Irrespective of whether static or dynamic analysis is in use, the essence of detection is to input a series of signature of targeted software into a discrimination system Γ to calculate. Here, we use *min-* to indicate the unit object received by the system, and the signature discussed in this paper is just the parameter unit, which gives system Γ a present value of b. As long as partial or even total signatures of targeted software match the rule $\aleph$ in system Γ, the discrimination system Γ can take it as malware.

Discrimination systems are supposed to receive infinite software signature x_i, which type or mode is decided by the input object. In practice, however, the signature sequence fed into a system Γ is finite.

Suppose the software signature sequence received by a discrimination system is $\boldsymbol{C}$, and

$$\boldsymbol{C} = \{x_1, x_2, \ldots, x_i\}$$

where x_i represents the signature of the targeted software, and $i \in \mathbb{N}$. Here, we suppose this signature sequence is linear and each unit is independent, or we can always find a transformation $\boldsymbol{A}$ [15]:

$$(x_1, x_2, \ldots, x_i) \cdot \boldsymbol{A} = \boldsymbol{C} \cdot \boldsymbol{A} = (x'_1, x'_2, \ldots, x'_i)$$

so that $(x'_1, x'_2, \ldots, x'_i)$ is linear and mutually independent.

Based on this, the abstract model of discrimination system is constructed by introducing URM. In Table 1, we list the commonly used notations and their meanings in this paper.

2.2. The URM model

The current discrimination system takes a linear sequence of software signature as input, and outputs the deduction as to whether the software is malicious or not. Thus, we use the Unlimited Register Machine (URM) [16] to characterize this model in this paper.

According to the URM model, all the symbols handled by URM should be encoded into natural numbers and stored in its register [16,17]. The URM has an infinite number of registers labeled as:

$$R_1, R_2, \ldots, R_n, \ldots,$$

to store the input signature parameters (natural numbers), which are denoted by

$$r_1, r_2, \ldots, r_n, \ldots$$

where each R_i contains a r_i respectively. Thus, the URM can be illustrated as Fig. 2.

Please cite this article in press as: L. Han, et al., Owner based malware discrimination, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.05.020