ARISTOTLE UNIVERSITY OF THESSALONIKI

Latex Lab 2.3

by

Uhtverov Matvey - Senutin Sergey

A thesis submitted in partial fulfillment for the
Undergraduate degree

in the
Faculty of Sciences
School of Informatics

ARISTOTLE UNIVERSITY OF THESSALONIKI

Abstract

Faculty of Sciences
School of Informatics

Undergraduate Degree

by Uhtverov Matvey - Senutin Sergey

In the Internet age, malware has posed serious and evolving security threats to Internet users. To protect legitimate users from these threats, anti-malware software products from different companies provide the major defense against malware. In this article, we first provide a brief overview on malware as well as the anti-malware industry, and present the industrial needs on malware detection. We then survey intelligent malware detection methods. In these methods, the process of detection is usually divided into two stages: feature extraction and classification/clustering. The performance of such intelligent malware detection approaches critically depend on the extracted features and the methods for classification/clustering. We provide a comprehensive investigation on both the feature extraction and the classification/clustering techniques. We also discuss the additional issues and the challenges of malware detection using data mining techniques and finally forecast the trends of malware development.

Contents

0.1 Overview of malware and anti-malware industry

0.1.1 Types of Malware

1. Self-Replicating Malware:
    - Viruses.
    - Worms.
2. Spyware and Information Stealing Malware:
    - Trojans.
    - Spyware.

0.1.2 Anti-malware industry

We will now revisit the anti-malware industry, which is used in the later part of this paper.

- Cloud-based malware detection.
- Data mining techniques.
- Hybrid Analysis.
- Dynamic analysis techniques.
- Feature extraction method.
- Signature-based detection.

0.2 Feature selection

From the above, the FS (Fk) is essentially the ratio of the average inter-class distance to the average intra-class distance. Thus, higher values of Fk imply that members belonging to different classes are further separated using the k-th feature, while members in the same class are closer together. The discrimination

ability for the k-th feature increases with increasing values of Fk. For the malware detection, the issue is often a two-class problem—positive (malicious) class or negative (benign) class. The FS then reduces to a simple form

0.2.1 Max-Relevance Algorithm

To minimize the classification error, feature selection often requires that the target class c has maximal statistical dependency on the selected features. One approach to realize maximal dependency (MaxDependency) is maximal relevance (Max-Relevance) feature selection

$$I(a_i, c) = \iint p(a_i, c) \log \left(\frac{p(a_i, c)}{p(a_i)p(c)} \right) d(a_i)\, d(c) \tag{1}$$

where $p(a_i), p(c)$ - mutual information defined using their appearance frequencies.

$$SI(S, A) = -\,|V| \sum_{v=1}^{|S|} \frac{|S_v|}{|S|} \times \log_2 \frac{|S_v|}{|S|}, \text{ where } SI(S, A) \text{ is the entropy of } S \tag{2}$$

$$IGR(S, A) = \frac{IG(S, A)}{SI(S, A)} \tag{3}$$

To classify any unknown file, which could be either benign or malicious, the classification process can be divided into two consecutive steps: model construction and model usage. In the first step, training samples including malware and benign files are provided to the system. Then, each sample is parsed to extract the features representing its underlying characteristics. The extracted features are then converted to vectors in the training set.

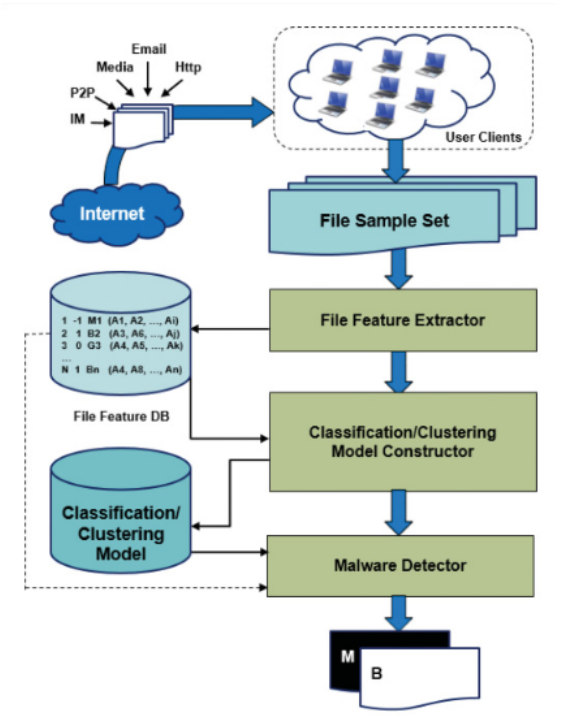

FIGURE 1: The overall process of malware detection using data mining techniques.

Signature-based Malware Detection To protect legitimate users from malware threats.

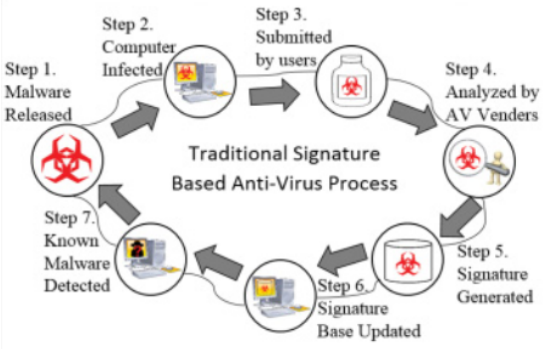

FIGURE 2: Process of traditional signature-based malware detection.

TABLE 1: Summary of Some Typical Feature Extraction Methods in Malware Detection

Survey	**Static analysis**	**Dynamic analysis**	**Other analysis**
Schultz et al. [2001]	DLL call information, strings, and byte sequences	X	
Kolter and Maloof [2004]	Binary n-grams	X	
Henchiri and Japkowicz [2006b]	16-byte sequences	X	
Wang et al. [2006b]	DLLs and APIs	Modifications upon system files, registries, and network activities	
Anderson et al. [2011]	X	Graphs constructed from dynamically collected instruction traces	
Ye et al. [2011]	X	X	File content combining file relations
Anderson et al. [2012]	2-gram byte sequences, disassembled OpCodes, control flow graph, and miscellaneous file information	Instruction traces, system call traces	