Future Generation Computer Systems 55 (2016) 376–390

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Hybrids of support vector machine wrapper and filter based framework for malware detection

Shamsul Huda [a,*], Jemal Abawajy [b], Mamoun Alazab [c], Mali Abdollalihian [d], Rafiqul Islam [e], John Yearwood [a]

[a] School of SITE, Federation University, Australia
[b] School of Information Technology, Deakin University, Australia
[c] Australian National University, Australia
[d] Mathematical and Geospatial Sciences Department, RMIT University, Australia
[e] Charles Sturt university, Australia

HIGHLIGHTS

- A signature-free malware detection approach has been proposed.
- A hybrid wrapper–Filter based malware feature selection has been proposed.
- Proposed hybrid approach can take advantages from both filter and wrapper.
- Models have also been validated by statistical model selection criteria such as Chi Square and Akaike information criterion (AIC).

ARTICLE INFO

Article history:
Received 19 November 2013
Received in revised form
28 April 2014
Accepted 1 June 2014
Available online 13 August 2014

Keywords:
Malware detection
API call statistics
Hybrid wrapper–filter heuristics

ABSTRACT

Malware replicates itself and produces offspring with the same characteristics but different signatures by using code obfuscation techniques. Current generation Anti-Virus (AV) engines employ a signature-template type detection approach where malware can easily evade existing signatures in the database. This reduces the capability of current AV engines in detecting malware. In this paper we propose a hybrid framework for malware detection by using the hybrids of Support Vector Machines Wrapper, Maximum-Relevance–Minimum-Redundancy Filter heuristics where Application Program Interface (API) call statistics are used as a malware features. The novelty of our hybrid framework is that it injects the filter's ranking score in the wrapper selection process and combines the properties of both wrapper and filters and API call statistics which can detect malware based on the nature of infectious actions instead of signature. To the best of our knowledge, this kind of hybrid approach has not been explored yet in the literature in the context of feature selection and malware detection. Knowledge about the intrinsic characteristics of malicious activities is determined by the API call statistics which is injected as a filter score into the wrapper's backward elimination process in order to find the most significant APIs. While using the most significant APIs in the wrapper classification on both obfuscated and benign types malware datasets, the results show that the proposed hybrid framework clearly surpasses the existing models including the independent filters and wrappers using only a very compact set of significant APIs. The performances of the proposed and existing models have further been compared using binary logistic regression. Various goodness of fit comparison criteria such as Chi Square, Akaike's Information Criterion (AIC) and Receiver Operating Characteristic Curve ROC are deployed to identify the best performing models. Experimental outcomes based on the above criteria also show that the proposed hybrid framework outperforms other existing models of signature types including independent wrapper and filter approaches to identify malware.

* Corresponding author. Tel.: +61 353276217.
E-mail addresses: s.huda@ballarat.edu.au (S. Huda), jemal.abawajy@deakin.edu.au (J. Abawajy), mamoun.alazab@anu.edu.au (M. Alazab), mali.abdollahian@rmit.edu.au (M. Abdollalihian), mislam@csu.edu.au (R. Islam), j.yearwood@federation.edu.au (J. Yearwood).

http://dx.doi.org/10.1016/j.future.2014.06.001
0167-739X/

1. Introduction

Malicious software (Malware) affects the secrecy and integrity of data as well as the control flow and functionality of a computer system which we combat every day [1]. There is no single technique [2–5], but most Anti-Virus (AV) engines use two main approaches: (1) signature-based and (2) anomaly-based approaches for malware detection. The signature-based detection [6,7] methods are very efficient to detect known malware [7]. However, the signature generation process for construction of the database for the AV engine involves manual processing and requires strict code analysis. Most of the malwares [5] have in-built process that can generate new variants each time it is executed and a new signature is generated. Therefore, signature based approaches fail to detect unknown malwares [5] which are not in the database. In contrast, anomaly-based detection approaches [7,8] use API call sequences instead of byte sequence matching through optimal sequence alignment. Although anomaly-based detection approaches use the knowledge of normal behavior patterns and perform better than the signature based approach. But these approaches [7,8] ignore the frequency of API calls in the sequences and suffer from the same problem as normal signature approaches and become similar to signature based approach resulting in a more false positives outcome [9]. Windows Application Program Interface (API) function calls [10–12,10] have been used in statistical N-gram modeling techniques [11,12] for detection. However these approaches [11,12] use simple wrapper classification methods [13] which did not explore the ways of selecting the best set of APIs from a large set of APIs. To find an optimal subset of API that can discriminate malware from benign is essential and difficult which also can be transformed into a feature selection problem. Usually given an m-dimensional API based malware dataset, a detection algorithm needs to find an optimal API subset from the 2^m subsets of the APIs. Therefore finding an optimal API subset is computationally expensive [14] feature selection problem. The performance of a detection algorithm depends on its evaluation criterion as well as search strategies.

The filter based models for best subset selection [15] are computationally cheap due to its evaluation criteria. However, feature subsets selected by filter may result in poor prediction accuracies, since they are independent from the induction algorithm. In contrast, the wrapper models [16] face huge computational overhead due to the use of the induction algorithm's performance criteria as its evaluation criteria. Some researchers have proposed [17] hybrid of genetic algorithm (GA) and filter heuristic where GA framework works as a subset generation process and filter heuristic improves local search. Despite significant researches on evaluation criteria and search strategies, current generation feature selection approaches lack the work that can combine the merits of wrapper and filter approaches. To the best of our knowledge, there is no complete malware literature that reveals with a suitable approach to find the most significant set of APIs from enormous number of API sets and can exploit the merits of both wrapper and filter approaches. This shows a clear and strong motivation for this work in the context of API feature selection for malware detection.

In this paper, we propose a framework that attempts to identify malware by using its malicious activities characterized by the Application Program Interface (API) calls and a novel hybrid wrapper–Filter feature selections techniques. At first, a large number of malware datasets with obfuscated and unknown malware are collected from many sources including the honeynet project, VX heavens [18]. The hybrid frame work proposes a novel automated method to extract the API call behaviors from malware dataset using sophisticated unpacking, disassembling and mapping analysis techniques. Then we propose two hybrid approaches using the hybrids of Support Vector Machines Wrapper heuristic and Maximum-Relevance–Minimum-Redundancy Filter heuristics for malware detection from the API call statistics. The novelty of our proposed malware detection approaches is that these techniques combine the knowledge about the intrinsic nature of malicious activities of the malware with the wrapper score in order to select the most significant set of API features. This is achieved by injecting the filter's ranking score (computed using the intrinsic characteristics from API call statistics) in the wrapper selection process and different search strategies. We have also used binary logistic regression to compare and assess the efficacy of the proposed approaches based on different goodness of fit criteria. Our contribution also includes the following hitherto unreported in the literature:

(1) Development of a fully automated framework for malware detection to compute API call statistics from malware and benign programs.
(2) Development of two novel hybrid API feature selection approaches based on the hybrid of Support Vector machine wrapper heuristics and maximum-relevance–minimum-redundancy filter heuristics that can find an optimal set of APIs in order to detect the malware from their malicious behavior.

The rest of the paper is organized as follows. The next section introduces some related background literature and limitations of current malware detection techniques. Section 3 discusses Filter and wrapper approaches and a mathematical derivation for wrapper heuristic based on Support Vector Machine (SVM). The proposed framework for malware detection using hybrids of Support Vector Machine wrapper heuristics and Maximum-Relevance–Minimum-Redundancy filter heuristics with API Call statistics has been described in Section 4. Section 5 describes the malware datasets. Section 6 presents experimental results, statistical validation and discussion about the results. Conclusions of this study are presented in the last section.

2. Related work

2.1. Code obfuscations and current malware detection approaches

Code obfuscation modifies the program code to produce offspring copies which have the same functionality with different byte sequence so that the new code is not recognized by antivirus scanner. Obfuscation techniques such as, packing [19] is used by malware authors as well as legitimate software developers in order to compress and encrypt the Portable Executable (PE) or Dynamic Link Library (DLL) in secondary memory for changing the byte sequence in the PE. This results different byte sequences in the newly produced packed PE. A second technique, polymorphism [19] uses encryption and data appending/data pre-pending in order to change the body of the malware. It also changes decryption routines from one infection to another as the encryption keys change. Finally, metamorphism [20] is used to transform the code without encryption in order to evade detection by static signature-based virus scanners. Several works [21] propose to use program graph mining techniques for combating (polymorphic) malwares. However, these works either employ subgraph matching or vector-space modeling to learn classifiers for malware detection. These methods are either not scalable (e.g., subgraph matching) or not adaptable to dynamic feature space such as API. Sung et al. [8] present a signature-based malware detection technique, with emphasis on detecting obfuscated (or polymorphic) malware and mutated (or metamorphic) malware. Tian et al. [22] present a method for classifying Trojans based on function lengths, and show that function length plays an important role in classifying malware and if combined with other features may result in a better method of malware classification. Signatures matching techniques in [21,22,8,20] to detect malware requires that signatures to be generated