Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

On the adoption of anomaly detection for packed executable filtering

Xabier Ugarte-Pedrero, Igor Santos, Iván García-Ferreira, Sergio Huerta, Borja Sanz, Pablo G. Bringas*

S3Lab, DeustoTech – Computing, Deusto Institute of Technology, University of Deusto, Avenida de las Universidades 24, 48007, Bilbao, Spain

ARTICLE INFO

Article history:
Received 23 January 2014
Received in revised form
12 March 2014
Accepted 28 March 2014

Keywords:
Malware
Packer
Anomaly detection
Machine-learning
Computer security

ABSTRACT

Malware packing is a common technique employed to hide malicious code and to avoid static analysis. In order to fully inspect the contents of the executable, unpacking techniques must be applied. Unfortunately, generic unpacking is computationally expensive. For this reason, it is important to filter binaries in order to correctly handle them. In previous work, we proposed the adoption of anomaly detection for the classification of packed and not packed binaries using features based on the Portable Executable structure. In this paper, we extend this work and thoroughly evaluate the method with a different dataset and two different feature sets, rendering new conclusions. While anomaly detection is reaffirmed as a sound method for the discrimination of packed and not packed binaries, Portable Executable structure based features present limitations to distinguish custom packed files from not packed files.

Introduction

Malware is the term used to designate software that was coded with malicious intentions, such as damaging computers or networks or even obtaining economic benefit in an illegitimate way. Security products such as Anti-Virus solutions and operating systems have evolved in order to detect and prevent the infection and execution of this kind of software. Consequently, malware writers have developed new techniques to evade detection. A very common technique is software packing, which consists of compressing or encrypting the malicious code, impeding the disassembly of the protected code. This content is then decrypted at runtime, prior to its execution. Some reports claim that up to the 80% of the malware analysed is packed (McAfee, 2009). Packed malware can be analysed with traditional automated dynamic execution techniques that explore the real functionality of the binary. Nevertheless, these techniques usually do not cover every possible execution path. In fact, many malware samples present complex functionality that cannot be easily triggered in an automated execution. In these cases, the code must be statically analysed in order to discover all its functionality, making necessary to unpack the sample. When the packer used to protect the sample is known, specific unpacking routines can be applied to extract the original code. On the contrary, for unknown packers it is necessary to generically unpack the code (according to Morgenstern and Pilz, 2010, 35% of malware is packed by a custom packer). A

* *Corresponding author*. Tel.: +34 944139000.
E-mail addresses: xabier.ugarte@deusto.es, xabiugarte@gmail.com (X. Ugarte-Pedrero), isantos@deusto.es (I. Santos), ivan.garcia.ferreira@deusto.es (I. García-Ferreira), shuerta@deusto.es (S. Huerta), borja.sanz@deusto.es (B. Sanz), pablo.garcia.bringas@deusto.es (P. G. Bringas).

http://dx.doi.org/10.1016/j.cose.2014.03.012
0167-4048/© 2014 Elsevier Ltd. All rights reserved.

correct classification of samples can help the analyst to correctly handle binaries.

Previous approaches have applied supervised machine-learning techniques for the classification of packed and not packed binaries using different heuristics (Perdisci et al., 2008b). Nevertheless, supervised approaches learn from both classes: packed and not packed files. Alternatively, anomaly detection methods can be applied in cases in which it is not reliable to model one of the classes. In this context, we consider that it is more realistic to collect a representative dataset of not packed samples, considering that packed binaries can present a higher variability. On the one hand, new packers are developed continuously. Malware creators can also employ modified versions of existing packers, or even custom made protection engines. On the other hand, common compilers normally follow standard conventions to form the resulting binaries. Following this intuition, we propose the application of a distance-based anomaly detection approach to classify packed and not packed binaries. More concretely, we evaluate two different feature sets, based on the Portable Executable structure and operational code frequency, and we apply a data reduction approach and evaluate different distance measures and distance selection rules.

In order to conduct this study, we define the following research questions:

- Which is the feature set that best discriminates packed from not packed files?
- What is the impact of the data reduction approach over the results obtained?
- What is the impact on the results of the different distance measures evaluated?
- What is the impact on the results of the different distance selection rules?
- Does our anomaly detection approach present sound results for the classification of packed and not packed files?

Finally, we discuss how these findings can be useful for the deployment of a binary filtering system in different contexts.

In previous work (Ugarte-Pedrero et al., 2011, 2012) we proposed a similar method for the classification of packed binaries. Nevertheless, this paper extends this work in several manners.

- We measure the appropriateness of different groups of features based on the Portable Executable structure for the classification of packed and not packed binaries. To this end, we test the performance for several common supervised machine-learning algorithms.
- We present a new threshold selection approach and a new normalization process for the anomaly detection method proposed in previous work, in order to avoid considering any data regarding packed samples for the classification.
- We extend the experiments by considering two different approaches for the data reduction approach (discarding or including outliers).
- We evaluate our approach over two different feature sets. In previous work, we tested a Portable Executable structure based feature set. In these experiments, we have considered a new feature set based on operational code frequency.
- We evaluate the method over a new dataset that (i) has been sanitized and (ii) includes custom packed binaries.

The remainder of this paper is structured as follows. Section Dataset selection describes the process followed to select the dataset. Section Feature selection details the feature sets employed for classification. Section Distance-based anomaly detection describes the anomaly detection method proposed. Section Evaluation presents the results obtained for the different experiments conducted. Section Conclusions and discussion describes and discusses the conclusions obtained from the experiments, and outlines avenues for future work. Finally, Section Related work compares this work with most related publications.

Dataset selection

In order to evaluate the adoption of anomaly detection for the classification of packed binaries, we configured a set of 4000 binaries.

The possible biases and limitations of the dataset were thoroughly studied and discussed. Nevertheless, the intrinsic nature of packers, the efforts of malware creators to evade detection, and the limitations of already existing tools make difficult to discriminate packed and not packed files. Actually, Royal et al. (2006) formulated the task of determining the existence of an unpack-execute process as an undecidable problem.

The possible risks to the validity of the experiment were reduced to the extent possible by defining a methodology for binary selection and labelling.

In this way, the dataset must fulfil several requirements. First, it must contain both goodware and malware for both packed and not packed classes, in order to ensure that the model discriminates packed files from not packed ones avoiding possible biases. In addition, the variability must be ensured to guarantee the inclusion of samples from different origins (e.g., system files, common tools ...), generated by different compilers. Secondly, different types of packers must be considered. On the one hand, off-the-shelf packers use very different techniques to protect samples. While some of them are simple compressors, others employ encryption and anti-analysis techniques or even instruction-set virtualization. On the other hand, current malware also employs custom packers (i.e., custom made protection), using a legitimate file as a carrier, making detection more difficult.

In order to ensure that all kinds of packers are represented in the dataset, different kinds of commercial packers and custom packers must be included. Finally, all the samples included in the dataset should be correctly labelled. This is usually the most difficult task when creating a dataset, and sometimes it is necessary to assume the existence of noise in it. In order to minimise possible errors in the labelling process, it is important to consider the actual limitations of the tools employed for the analysis.

Several tools, such as PEiD, identify known packer signatures by searching for common fingerprints in the headers and the unpacking stub of the packer. Nevertheless, malware writers sometimes modify their samples to evade signature-