Malware Analysis by Combining Multiple Detectors and Observation Windows

Massimo Ficco

IEEE TRANSACTIONS ON COMPUTERS, VOL. 71, NO. 6, JUNE 2022.

Problem addressed in this paper:

Malware developers continually produce new malware variants and use several evasive techniques to hide the malicious behaviour in legitimate applications and delay the detection process. According to report, the number of malware produces grows rapidly every year makes difficult for their detection process. Each new variant of malware could evade signature-based malware detection methods used by most of todays Anti-Virus software. In last year, several dynamics analysis detector have been proposed that are resiliant to adversarial evasion because deterministic classifier can always be reverse-engineered. The use of multiple diverse detectors and stochastic mechanisms for unpredictably switching between them makes the ensemble detector more resilient to both reverse-engineering and evasion.

This paper proposes an ensemble detector that exploits diversity in detection algorithm and it presents an extensive experimental campaign that shows the detection performance as a function of the incorporated detectors and the length of the observation time window.

- Malware can be described from multiple dimensions. For each of them, the most significant detection algorithm is considered.
- Switching between different sets of specialized and generic detectors during the analysis process can improve the unpredictability of the detection technique.
- An alpha-count mechanism is proposed and used to explore how different observation window and multiple computation periods can affect the detection accuracy and speed of different combination of detectors.

Motivation for the Problem:

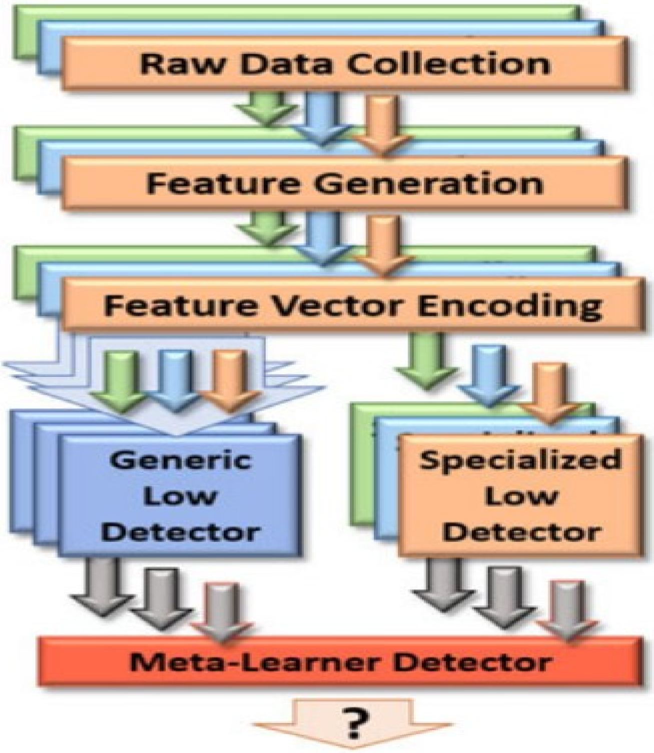
Deterministics classifiers can easily reverse-engineered and prone to adversarial invasion which result in bypassing of malware detection process. Each new variant of malware could evade detection methods used by most of today’s Anti-Virus software.

Motivation for the Solution:

If we add rondomness by adding multiple detectors and stochastic mechanisms for unpredictabillity. switching between them makes the ensemble detector more resilient to reverse-engineering and evasion. The resilience of the detection technique increases with the number of diversities introduced. Based on these assumption, they exploits diversity in detection algorithm by combining various feature, switching between specialized and generic detectors and by changing the length of observation window.

Theory:

They adopted the ensemble learning technique for malware analysis consists in combining multiple independent detectors that are each trained to detect any type of malware. Specialized detectors trained for specific malware types that could perform better compare to a generic detector in classifying malware. Generic detectors are trained for all type of malware. Their result shows that *ensemble detector composed only by specialized detectors does not always produce greater sensitivity than* *generic detectors, especially in presense of malware evolution.* Therefor they uses both specialized and generic detecotrs. The stacking approach has been considered as a descision fusion function and used to combine the specialized and generic detectors. Collected malware features are separately processed in specialized and generic first-level neural networks, called ‘low detectors’. The first-level detectors are connected to a second-level neural network called ‘meta-learner detector’ for producing final classification.



Each low detector is first trained then the meta-learner detector is separately trained with the output of the previously learned initial networks. Finally a validation dataset different by one used for training the low detectors is used for a meta-learner test. This learning strategy can enhance the overall accuracy of the model, reduce the training frequency needed to face malware evolution, as well as support stochastic switching between different low-detectors needed to increase the unpredictability of the detection strategy. Following steps are followed:

- All malware and benign apps are executed in a controlled environment and behaviour is extracted.
- Features that are derived from the processed apps: the API call frequency, the API calls pattern, the probabilities of transitioning between two API calls, the memory data dump, and the network messages.
- A multimodal deep neural network is trained to fit the produced features.

Observations of the dynamic behaviour of malware:

- Malicious software tends to use a greater number of critical API calls than benign apps.
- Within the same family, variants of the malware invoke critical API calls by following similar patterns.
- Use of harware features in addressing the malware evasion problem during dynamic analysis.
- Malware generated network traffic more frequent than normal traffic.

Five main algorithms proposed in the paper have been compared and implemented to extract the considered features of the malware behaviour.

1. API Call Frequency Detector (AFD) : it list out the API calls that are more frequent in the malware than in the benign set.
2. API Sequence Alignment Detector (ASD) : match the common sequence of API using MSA algorithm.
3. Markov Chain Detector (MCD) : sequence of state transitions perfored by the analysed app is represented as a markov chain.
4. Memory Dump Detector (MDD) : it monitor the malware process memory and convert the memory data dump binary files into grayscale images.

5. Network Traffic Detector (NTD) : it analyse the network traffic generated by apps at the network access point.

Implementation:

The dataset used to identify the best performing combinations of low detectors is composed of different malware families, collected from the Drebin dataset which includes 5.56k malware samples grouped in 179 different families. The final dataset consist of 27 families and 4960 samples. 1.2k benign apps have been obtained by google play store using the GooglePlayApi tool. Specialized detectors are trained using only one single malware type at a time and then evaluated in order to identify the best performing specialized detector for each malware type. Generic detector are trained using all the malware types. The whole dataset have been divided into three subsets, training (60%), validation (20%), and testing (20%).

Best performing combinations of low detectors should be selected to reduce the number of detectors. The approach consisits of :

- identifying the best performing specialized detectors for each malware type,
- comparing the selected specialized detectors against each generic detector with respect to the same malware type that the specialized detectros have been designed for,
- selecting the best performing combinations of low detectors,
- evaluating the performance rate variation of the final ensemble-detector within the observation time window.

The considered features are extracted by using the Cuckoo sandbox, the analysed samples are executed and invoked API call sequences are collected and recorded. The *'ma command'* is used to extract the dump data file, which includes the entire virtual address space of the monitored process. Tcpdump is used to capture the network traffic from the phone.

Critique:

The proposed works enhance the unpredictability of the detection process therefore, it is too hard to reverse-engineer the detection process and ineffective agianst adversarial invasion. They use multiple dynamic features collected from diverse dimensions of a malware. Based on these features different detectors have been trained and evaluated. These features are not specific to hardware therefore collecting all these features is extra-overhead to the system. A secure framework should be built to periodically update as malware evolve and automatically switch during the detection process.

Regarding Related Works:

- *DroidAPIMiner: Mining API-level features for robust malware detection in android* – generated the set of APIs used by a dataset of benign and malicious apps, and performed a frequency analysis to list out the API calls that were more frequent in the malware than in the benign set.
- *A novel approach to detect malware based on API call sequence analysis* – showed that certain API call patterns are commonly included in malware even in different families, they can be used to detect new unknown malware.
- *Polymorphic attacks against sequence based software birthmarks* – change the specific position of malicious application by add, delete or replace API calls in order to produce new sequence that could reduce the effectiveness of API sequence matching detection process.
- *Comparing API call sequence algorithms for malware detection* – they adopted the markov chain model, in which the vector of each app is represented by the probabilites of transitioning from on API call to another in the chain. This approach can be more resilient against evasion technique.

- *A malware classification method based on memory dump grayscale image* - proposed a method based on memory dump data, which converted malware memory data dump binary files into greyscale images, and used multilayer perceptron to classify them.
- *Detecting application update attack on mobile devices through network features – i*t capture traffic data generated by malware samples in a real internet evironment. They capture DNS and HTTP network traffic in the first 5 minutes, and analysed the major compositions of the traffic data.