Poster: Exploring Family Features for Classification and Lineage Inference of Packed Malware

Leo Hyun Park, JungBeen Yu, and Taekyoung Kwon
Information Security Lab, Yonsei University, Seoul, 03722, Korea
{dofi, symnoisy, taekyoung}@yonsei.ac.kr

***Abstract*—Both classification and lineage inference are important subjects for handling a tremendous amount of malware variants emerging today. Many previous studies have been done in this respect, classifying variants into families, while they lacked in considering packed malware in lineage inference and applied the common features to all lineages. In this study, with regard to malware lineage inference, we consider packed malware that accounts for the majority of today's malware variants. To improve accuracy, our basic idea is applying each of different features, saying, family features derived through classification and feature selection phases, to identify each malware family. Our experimental study shows that family features are effective and also practical compared to the common features in lineage inference. We also discuss our on-going work and future directions.**

I. Introduction

The amount of malware emerging annually is substantial, but reportedly most of such malware are variants. According to *AV-TEST*, about 110 millions of malware appeared in 2017 but new malware species were only 7.41 millions among them. This means that a tremendous amount of malware are variants derived from the existing ones. Especially, it is estimated that over 80% of malware are packed on distribution [4]. Thus, both classification and lineage inference of malware are very important in that sense, and a large amount of packed malware should be considered very seriously in doing so.

There have been many previous studies regarding classification of packed malware, such as based on static, dynamic, and hybrid features [1], [2], [4], but mostly they did not consider lineage inference in the end. Regarding the lineage inference to trace malware developments, many studies have been done but there still remain challenges. First, it is still a necessary work to deal with packed malware in lineage inference. In the previous studies, some had filtered out packed malware while some could not handle repacked malware [3], [5]. Second, the previous studies applied the common features to different lineages. However, our concern is that applying common features might degrade accuracy in lineage inference because each different family and even its member would behave with different sensitiveness upon the common feature. Our basic insight is that there might exist family features upon which each family member behave with similar sensitiveness.

In this study, we design a new method for classification and lineage inference of a large amount of packed malware. Our main idea is to derive the so-called family features from static and dynamic behaviors of malware variants, and utilize them for more accurate lineage inference upon the packed malware. Our system design is straightforward as illustrated in Figure 1, which proceeds in stepwise for feature extraction and representation, group classification, and lineage inference. We also perform experiments on a large scale dataset, 288 original and 8,640 packed malware samples. We classify them into families using various classification algorithms, and select the most accurate algorithm for our system. We then derive feature sets for each family with a *forward stepwise selection algorithm* and perform lineage inference. We compare the accuracy of lineage inference using family features and common features.

II. System Design

A. Feature Extraction

For malware group classification and lineage inference, we extract both static and dynamic features from malware, one from malware binaries, especially PE header, and the other from a sandbox hardened to deal with anti-VM malware. From sandbox analysis, we extract API call sequences and other behavioral features (e.g., file, registry, network, and mutex).

Our distinguished point is to deal with malware features in two categories, that is, the common features and the family features. In group classification, we only consider the common features, e.g., API call sequence, DLL info., and entropy. We represent API call sequences by 2-gram, reflecting the short sequence pattern and capturing the basic semantics of the program. To improve accuracy in lineage inference, we use hybrid features as family features, derived by the *Forward Stepwise Selection* algorithm described in §II.C.

B. Group Classification

We need to classify malware into families in advance, so as to derive their features it infer lineages more accurately. We compared the accuracy of various classification algorithms, such as random forest, support vector machine, and k-nearest neighbor, by applying 10-fold cross validation. This process is repeated 10 times, and in each process, the dataset is randomly split into 10 pieces, nine for model training, and the other one for classification. As a result, we apply the *random forest*, which has the highest accuracy, to the framework.

C. Lineage Inference

For test sets, we use the *agglomerative clustering* to infer lineage of each family as classified in the previous phase. This algorithm does not require the number of clusters as an initial input. In addition, the result of merging can be useful for generating a dendrogram. Original malware and variants derived from it are grouped together in this phase.

Prior to the lineage inference, we derive a family feature using the *Forward Stepwise Selection* algorithm [1]. First, this

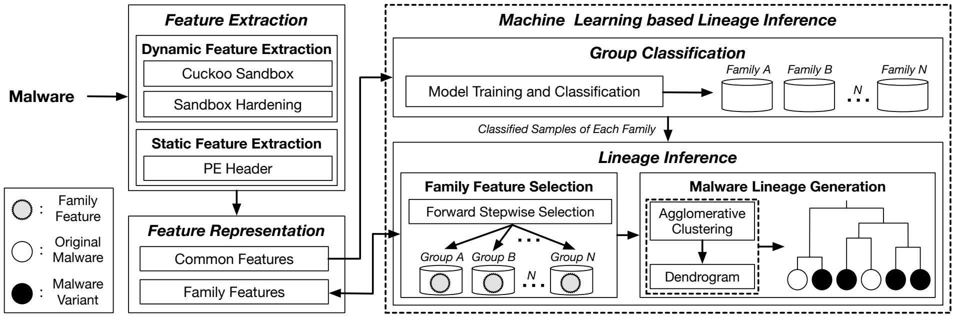


Fig. 1. System Overview

TABLE I. AVERAGE ACCURACY, PRECISION, RECALL, AND F1-SCORE OF GROUP CLASSIFICATION (%)

Feature	Accuracy	Precision	Recall	F1-score
Static Feature	97.67	97.85	97.02	97.43
Dynamic Feature	98.82	98.69	98.33	98.51
Hybrid Feature	99.40	99.36	99.14	99.25

TABLE II. CLUSTERING ACCURACY OF COMMON AND FAMILY FEATURES (%)

Feature	Common				Family
	Static	Dynamic	Hybrid	Selected	
Feature Selection	-	-	-	78.87	96.27
Lineage Inference	20.03	54.58	20.03	73.74	93.56

algorithm selects the most accurate feature among the feature category candidates, and then adds to the empty feature set. In the same way, the remaining candidates are added in order of the highest accuracy. Especially, this algorithm stops when the added feature does not affect the accuracy. This algorithm stops when adding more features does not increase the accuracy. When we analyze new samples of this family, we use the previously selected feature set.

III. Implementation and Evaluation

A. Environment

1) Dataset: We collected malware samples from *VX Heavens*, and filtered out samples that did not match the *VirusTotal* family label for reliable dataset configuration. We also excluded families with too many or too few samples. After then, we selected samples to be packed in the rest of the families, and created the variants through six popular packers (*UPX, ASPack, PECompact, PETite, NSPack, and VMProtect*). As a result, 288 original and 8,640 variants of malware in 15 families were used in our experiment.

2) Framework: Our system was evaluated and calibrated based on the reliable dataset. All of our evaluations were performed on Intel (R) Core (TM) i5-6600 CPU @ 3.30GHz, 32GB RAM and Ubuntu 16.04.2 LTS. We extracted malware features using *Cuckoo Sandbox* and the *pefile* library of *Python*. Especially, we used a new version of *Cuckoo Sandbox* modified by *Spender* to handle anti-VM malware. We adopted libraries provided by *scikit-learn* for both classification and clustering.

B. Experimental Result

1) Classification: Table I shows the accuracy of *Random Forest*. Hybrid features outperform the separate use of dynamic and static features.

2) Lineage Inference: We used a half of the samples for feature selection and then the derived features for lineage inference of the remaining half-samples. Table II shows the clustering accuracy. Clustering based on family features was more accurate than only on the common features.

As for the features, we see that the *compile time* was the most accurate feature in 10 families. In addition, the *accessed files* helped improve the accuracy of those families. On the other hand, the families, which only had the *corrupted compile time*, were accurate with different features, such as *accessed registry keys, deleted files, and network information.*

IV. Summary and Future Work

We study lineage inference of packed malware based on the concept of family features. Our current experiments show that we could effectively deal with a tremendous amount of packed malware for lineage inference. In the future work, we will consider more about actual malware environments. As new malware families are still emerging every year, we need to find them in group classification phase. This can be performed by *outlier detection*. We retrain the classifier to identify outliers and then also apply *forward stepwise selection* to find their features. We will utilize more reliable dataset which consists of malware classified manually by an expert or security vendors.

Acknowledgment

This research was supported by the MSIT, Korea, under the ITRC support program (IITP-2017-2016-0-00304).

References

[1] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Novel feature extraction, selection and fusion for effective malware family classification," in *Proc. ACM CODASPY*, 2016, pp. 183–194.

[2] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "Scalable, behavior-based malware clustering," in *Proc. NDSS*, 2009, pp. 8–11.

[3] M. Graziano, D. Canali, L. Bilge, A. Lanzi, and D. Balzarotti, "Needles in a Haystack: Mining information from public dynamic analysis sandboxes for malware intelligence," in *Proc. USENIX Security Symposium*, 2015, pp. 1057–1072.

[4] X. Hu, K. G. Shin, S. Bhatkar, and K. Griffin, "MutantX-S: Scalable malware clustering based on static features," in *Proc. USENIX ATC*, 2013, pp. 187–198.

[5] M. Lindorfer, A. Di Federico, F. Maggi, P. M. Comparetti, and S. Zanero, "Lines of Malicious Code: Insights into the malicious software industry," in *Proc. ACSAC*, 2012, pp. 349–358.

Poster: Exploring Family Features for Classification and Lineage Inference of Packed Malware

Leo Hyun Park, JungBeen Yu, and Taekyoung Kwon
Information Security Lab, Yonsei University, Seoul, 03722, Korea

Motivation

- Large scale malware variants : According to *the Av-Test's latest Malware Statistics and Trends Report*, about 110 millions of malware appeared in 2017.
- However, the new malware species were only 7.41 millions among them. This means that a tremendous amount of malware are variants derived from the existing ones.
- Especially, it is estimated that over 80% of malware were packed on distribution.
- Related work

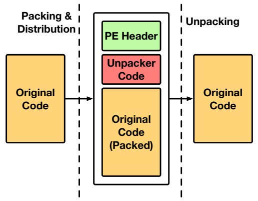
  - `Lineage Inference` refers to tracing the malware development created by the same author and identifying their relationship.

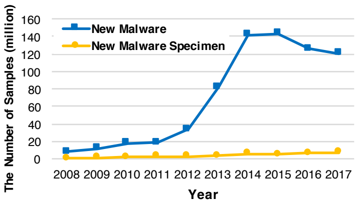
  - Many previous studies have tried to infer lineage of malware (e. g., `M. Lindorfer et al., ACSAC, 2012` and `M. Graziano et al., Usenix Security, 2015`), but there remain challenges.

  - First, it is still a necessary work to deal with packed malware in lineage inference.

  - Second, they applied the common features to different lineages. This can cause degradation of accuracy.

Packing and featuring in lineage inference are still challenging



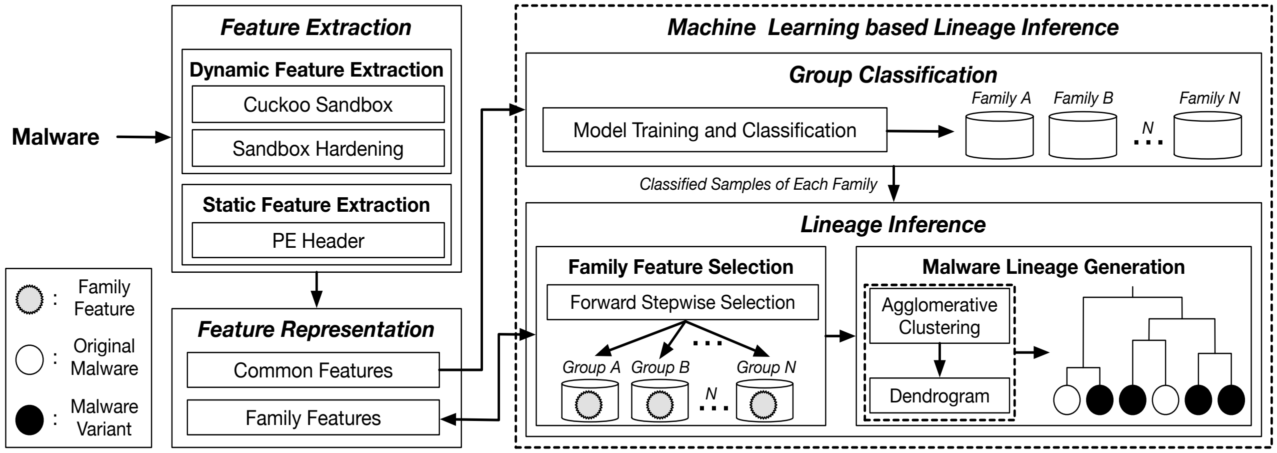


Our Objectives

- Group classification and lineage inference of large scale packed malware based on hybrid features
- Efficient derivation of family features by applying the *Forward Stepwise Selection* Algorithm
- More accurate lineage inference based on family features

Acknowledgement: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2016-0-00304) supervised by the IITP (Institute for Information & communications Technology Promotion).
Corresponding author's email: taekyoung@yonsei.ac.kr

Our System



Malware

Feature Extraction
Dynamic Feature Extraction
Cuckoo Sandbox
Sandbox Hardening
Static Feature Extraction
PE Header

Feature Representation
Common Features
Family Features

○ : Family Feature
○ : Original Malware
● : Malware Variant

Machine Learning based Lineage Inference
Group Classification
Model Training and Classification
Family A
Family B
N
Family N
Classified Samples of Each Family
Lineage Inference
Family Feature Selection
Forward Stepwise Selection
Group A
Group B
N
Group N
Malware Lineage Generation
Agglomerative Clustering
Dendrogram

- Environments : Intel (R) Core (TM) i5-6600 CPU @ 3.30GHz, 32GB RAM and Ubuntu 16.04.2 LTS host operating system.

Group Classification

- Insight: Classify malware into families in advance to deriving features and inferring lineage for each family.
- Methodologies :

  1. `Used Features`

     - Static Features : Section, Entropy, DLL, etc.

     - Dynamic Feature : API Call Sequence

     - We represent API call sequences by 2-gram, reflecting the short sequence pattern and capturing the basic semantics of the program.

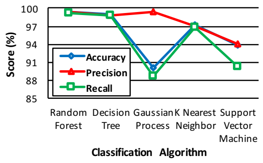
  2. `Classification Algorithm`

     - We compared various classification algorithms by applying *10-fold cross validation*.

     - We applied the *Random Forest*, which has the highest accuracy, to the framework.

     - Hybrid features outperform the separate use of dynamic and static features.

- Evaluation of Classification Algorithms and *Random Forest* Algorithm(%)



Feature	Static	Dynamic	Hybrid
Accuracy	97.67	98.82	99.4
Precision	97.85	98.69	99.36
Recall	97.02	98.33	99.14
F1-score	97.43	98.51	99.25

Lineage Inference

- Insight : For test sets, use *agglomerative clustering* to infer lineage of each family classified in the previous phase.
- Methodologies :

  1. `Family Feature Selection`

     - *Forward Stepwise Selection* : We use this algorithm to derive family features. The feature category candidates are added in the order of highest accuracy. If the number of feature candidates is n, then this requires a total of n^2 calculations (2^n when examining all feature combinations).

     - The *compile time* was the most accurate feature in 10 families. In addition, the *accessed files* helped improve the accuracy of those families.

  2. `Malware Lineage Generation`

     - *Agglomerative Clustering* : Original malware and variants derived from it are grouped together.

     - Dendrogram Analysis : The result of merging can be useful for generating a dendrogram.

- Clustering Accuracy of Common and Family Feature(%)

Feature	Common				Family
	Static	Dynamic	Hybrid	Selected	
Feature Selection	-	-	-	78.87	96.27
Lineage Inference	20.03	54.58	20.03	73.74	93.56