논문 18-43-12-14 The Journal of Korean Institute of Communications and Information Sciences '18-12 Vol.43 No.12
https://doi.org/10.7840/kics.2018.43.12.2089

# 2-Dimension 정적 Feature Set이 적용된 Convolutional Neural Network 기반의 악성코드 패킹 분석

황 준 호◆, 이 태 진°

# Malware Packing Analysis Based on Convolutional Neural Network with 2-Dimension Static Feature Set

Jun-ho Hwang◆, Tae-jin Lee°

요 약

근래의 악성코드 발생량의 폭발적인 증가와 더불어 악성코드는 지능화/고도화되고 있다. 이러한 경향 중 하나로 대부분의 악성코드에 분석을 어렵게 만드는 패킹 기법이 적용되어 유포되는데, 대량의 악성코드에 실시간으로 대응하기 위해 시스템의 성능 제약이 적은 정적분석을 채택한 악성코드 자동화 분석 시스템들의 경우에 이러한 패킹 기법으로 인해 분석 성능이 저하되고 있는 실정이다. 이에 다양한 패커 식별 연구가 진행되어 왔지만 대부분 복잡한 매커니즘으로 인해 실제 운용환경에 적용하기에 성능제약이 크기 때문에 대량의 악성코드에 대응하기에 한계가 있다. 본 논문에서는 분석성능을 유지하면서 보다 단순하고 가볍게 구축할 수 있는 경량화된 패킹 분석 시스템을 제안한다. 이는 고속의 분석 성능을 가진 정적 분석 기법을 기반으로 고차원의 feature 조합과 유사 그룹분류에 탁월한 Convolutional Neural Network를 연계함으로서 가능하다. 본 연구 결과물은 독립적인 모듈로서 동작 가능하며, 향후 지속적인 연구를 통해 기존의 안티바이러스나 자동화된 악성코드 분석 시스템과 연계하여 패커그룹을 우선적으로 식별해주는 pre-filter 시스템으로 동작 가능할 것이라 기대한다.

**Key Words** : Malware, Packing, Convolutional Neural Network, Static Analysis

ABSTRACT

Along with the recent explosion of malicious code, malicious code is becoming intelligent / advanced. One of these trends is that most malicious code apply to the packing technique makes analysis difficult. In the case of malicious code automated analysis systems adopting static analysis with low system performance constraints in order to cope with a large amount of malicious codes in real time, analysis performance is deteriorated due to such a packing technique. Various packer identification studies have been carried out. However, due to the complicated mechanism, it is difficult cope with a large number of malicious codes. In this paper, we propose a lightweight packing analysis system that can simplify and lightly construct while maintaining analytical performance. This is possible by linking a static analysis technique with high-speed analysis capability to a high-dimensional feature combination and Convolutional Neural Network with excellent performance similar group classification. The results of this study can be operated as an independent module and it will be possible to operate as a pre-filter system that identifies the packer group in advance by linking with existing antivirus or automated malicious code analysis system through continuous research.

※ 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. 2018-0-00276, 딥러닝 기반 악성코드 패턴룰셋 생성 자동화 원천 기술 개발)

◆ First Author : (ORCID:0000-0002-3440-7218)Hoseo University Department of Information Security, hwangso93@gmail.com, 학생회원
° Corresponding Author : (ORCID:0000-0003-3078-3459)Hoseo University Department of Information Security, kinjecs@hoseo.edu, 정회원
논문번호 : 201806-B-017-RN, Received May 30, 2018; Revised September 28, 2018; Accepted October 23, 2018

## Ⅰ. 서 론

최근 악성코드 발생량의 폭발적인 증가와 더불어 다양한 디바이스를 타겟으로 전방위적 보안 위협을 가하는 악성코드, 딥웹/토르 등을 이용한 악성코드 유포지 추적 회피, 악성코드 분석을 어렵게 만드는 파일 압축/암호화 등의 난독화 기법 적용 악성코드, 수많은 악성코드 그룹별 기능 및 이로 인한 일관성 없는 악성코드 특징 등 지능화/고도화된 악성코드들로 인해 사회 전반에 막대한 피해를 야기하고 있다. 특히, 난독화 기법이 적용된 악성코드와 수많은 악성코드 그룹은 분석을 어렵게 하는 주요 원인이다. 이러한 지능화된 대량의 악성코드에 대응하기 위해서는 자동화된 악성코드 분석기술이 중요한데, 그 중 하나인 자동화된 악성코드의 패킹 분석 연구는 지속적으로 연구되어왔지만 여전히 정적, 동적 분석 간의 이해상충 관계가 존재한다. 행위 기반의 동적 분석의 경우에는 파일 난독화, 그룹분류 관점에서 상대적으로 뛰어난 성능을 보이지만 성능제약이 여전히 큰 상황이고 정적 분석의 경우에는 경량화 된 시스템 구성이 강점이지만 파일 난독화, 다수의 악성코드 그룹분류에 견고하지 못한 점이 존재한다. 따라서, 본 논문에서는 정적분석 및 유사도 검색에 우수한 CNN(Convolutional Neural Network)기법을 활용하여 파일 난독화에 견고한 악성코드의 패킹 분석 기술을 제안한다. 제안 기술은 정적분석으로 도출되는 악성코드 특징을 CNN 기법과 연계 가능한 2-Dimensional 배열로 가공하는 동시에, 컴퓨팅적 관점의 머신러닝 연산으로 경량화된 악성코드 분석 시스템을 구축한다. 이는 기존의 DNN(Deep Neural Network)를 활용한 악성코드 분석 시스템과 같이 다수의 feature를 전수 분석 및 연계해석 소요가 적은 2-Dimensional 배열 제안기법을 통해 악성코드의 locality 특징을 고차원으로 표현함에 따라 동작한다. 따라서, 본 연구 결과물은 악성코드에 대한 패커 분석 시스템으로 동작하게 되며 악성코드 자동화 분석 시스템 관점에서 pre-filter 시스템으로 동작하여 향후 악성코드 그룹분류 및 악성여부 분석 시스템과 연계하여 고도화된 분석 성능을 나타낼 것으로 판단된다. 또, CNN 기법 연산에서 옵션으로 제공되는 악성코드의 이미지 등을 이용하여 분석 전문가에게 유의미한 정보로 제공 가능할 것이라 판단된다.

다음으로 2장에서는 악성코드의 패킹 분석의 다양한 방법에 대한 연구들과 악성코드 그룹분류를 시각화기법 및 CNN 분류기 등을 연계하는 방법에 대한 연구들을 기술하고, 3장에서는 본 논문에서 제안하는 악성코드 정적특징 기반 2-Dimensional 악성코드 패킹분석 기법과 바이트시퀀스 기법의 개선모델을 제안한다. 4장에서는 제안하는 모델과 기존의 기법의 개선모델에 대한 성능분석 결과를 보이고 각 기법의 장단점을 기술한다. 5장에서는 제안하는 악성코드 패킹분석 기법의 유의미성과 향후 개선 방향 및 결론 등을 기술한다.

## Ⅱ. 관련연구

근래의 악성코드는 다양한 변종 악성코드를 통해 시그니처 기반 분석 작업을 어렵게 함에 더불어 각종 난독화 기법을 적용하여 분석을 어렵게 하는데, 이러한 지능화된 악성코드의 경우 대부분 변종 악성코드에 패킹 기법을 연계해 그 특징을 모호하게 만드는 경향이 있다. 이에 대응하기 위해 악성코드 그룹분류, 패킹 분석에 대한 다양한 연구가 이루어지고 있는데, M. Bat-Erdene의 패커별 entropy 패턴을 이용하여 Multi-Layer 패킹 분석 모델을 제안하는[1], Wright. C.S.의 주요 패커 중 하나인 NsPack을 debugging tool을 이용하여 심층 분석한 결과를 제시하는[2], Yan, W의 패커들의 스택 레벨의 restore 특징에 기반한 스택 포인터 룰, 컴파일러의 entry 시그니처 등을 이용한 시그니처 룰, windows API들을 이용하는 행위기반 룰 등을 이용한 언패킹 기법 연구[3], Al-Anezi의 entropy 분석, 압축해제 속도에 최적화된 lossless 데이터 압축 알고리즘 LZO(Lempel–Ziv–Oberhumer), LZ77과 Huffman 코딩의 조합을 이용하는 Deflate 데이터 압축 알고리즘 등에 기반한 언패킹 기법[4] 등이 패킹 분석 연구로 진행되었다. Fig. 1.은 M. Bat-Erdene가 제시한 복잡한 수준의 패킹에 따른 바이너리 내부 데이터 변화를 나타내는데, 앞서 기술한 것과 같이 바이너리의 패킹 수준에 따라 내부의 데이

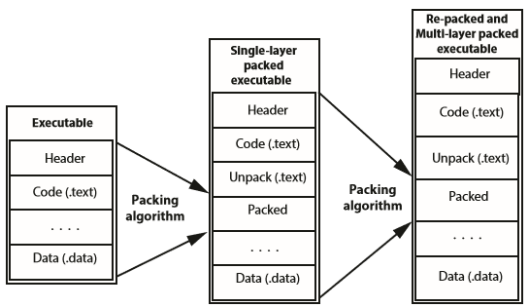



그림 1. 패킹 시 내부 바이너리 데이터 변화 예시
Fig. 1. Example of changing internal binary data when packing

터의 구조 또는 값이 크게 변화하여 악성코드와 정상 파일간의 특징을 특정 짓기 어려워진다는 것을 나타낸다. 따라서, M. Bat-Erdene는 상용환경에서 유포되는 언패킹, single-패킹, multi-layer 패킹 등을 분석 대상으로 지정하여 패킹 수준에 따른 분석 매커니즘을 제시하지만 이러한 기존의 패킹분석 연구의 경우 복잡한 분석 매커니즘과 그에 따른 제약조건으로 인해 실 운용이 어려운 부분이 존재한다.

반면에, 악성코드 그룹분류 연구로는 S. Yue의 악성코드 그룹별 이미지를 이용한 VGG 모델별 성능측정, Scaling 파라미터 효과 및 악성코드에서 추출되는 feature를 활용한 class별 feature map 연구[5], L. Nataraj의 바이너리 파일의 grayscale 이미지 시각화를 통한 악성코드 변종그룹 식별 연구[6], Seonhee Seok의 바이너리 파일의 바이트시퀀스를 이용한 8비트 grayscale 이미지 시각화와 CNN 연계를 통한 악성코드 그룹분류 연구[7] 등이 있다. Fig. 2.는 L. Nataraj가 제안한 일반적인 시각화된 이미지를 만드는 방법이고 Fig. 3.은 이러한 방법으로 시각화된 악성코드와 PE 구조에 대한 예시를 나타내는데 이러한 바이너리 파일의 비트, 바이트 단위의 이미지화를 통해 그룹을 분류하는 방법의 경우에는 단순하면서 효율적인 장점이 있지만 파일의 난독화 수준에 따라 이미지의 형태가 크게 변경되는 경향이 존재한다. 이에 S. Yue는 CNN 연산 과정에서 사용되는 특징들을 이용하여 악성코드 그룹별 feature map을 제시하지만 모든 feature들의 결합으로 1차원 feature matrix를 형성하기 때문에 선행 연구와 동일한 문제점이 존재한다.

본 논문에서는 이러한 선행 연구들의 제안 방식을 연계하여 악성코드 시각화와 CNN 기반의 패킹 분석 기법을 제안하는데, 해당 기법은 패킹 여부 및 패커의 그룹을 식별한다. 또, 제안 기법은 기존 연구들의 문제점들을 개선하기 위하여 단순하면서 고속의 처리 성능을 가진 악성코드 패킹 분석 시스템으로 구현되었는데, 다음 장에서 제안 매커니즘 전반에 대해 자세히 기술한다.

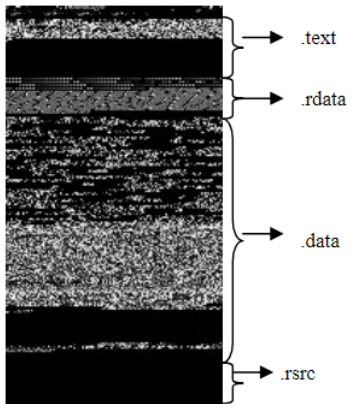



그림 3. 바이너리 시각화 시 섹션 구조 예시
Fig. 3. Example of section structure in binary visualization

## Ⅲ. 제안모델

### 3.1 바이트 시퀀스 기반 악성코드 분석

CNN 기반의 악성코드 분석 연구에서 바이너리 파일의 바이트 시퀀스를 이용한 기법들이 다수 제안되었는데 해당 기법은 악성코드 정적 분석의 일종으로 난독화 기법이 적용된 파일 분석에 일반적으로 효과적이지는 않다. 그럼에도 불구하고 간단한 동작 방식과 일정 수준의 분석 성능으로 인해 활발히 연구되고 있고 유의미한 연구 결과를 나타내고 있다. 하지만, 기존의 CNN 기반의 악성코드 분석 연구들의 경우에 바이너리 파일의 이미지화 프로세스가 수반되고, 상용환경의 바이너리 파일들의 크기가 일정하지 않아 크기가 큰 파일의 경우에는 바이트 시퀀스의 일부만 적용하는 등 일관성 있는 이미지 처리 정책을 적용하고 있지 않은 상황이다. 이는 분석 시스템의 성능 및 신뢰성을 저하시키는 문제로 직결된다. 따라서 본 논문에서는 적절한 이미지 처리 정책을 제시하고 컴퓨팅 관점에서의 머신러닝 연산 기법을 제안함으로써, 일관

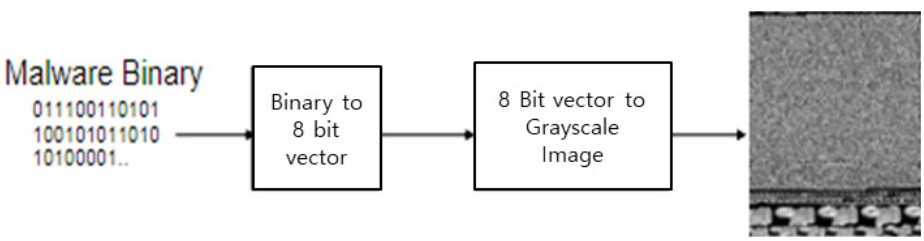



그림 2. 일반적인 바이너리 시각화 처리 과정 예시
Fig. 2. Typical Binary Visualization Process Example

성 있고 경량화 된 바이트 시퀀스 기반의 패킹 분류 시스템을 제안한다.

#### 3.1.1 바이너리 파일 이미지 변환

바이너리 파일의 데이터는 파일 내부에 일정한 포맷으로 구성되게 되는데, 이와 별개로 데이터 자체는 일반적으로 비트 단위로 구성되게 된다. 우선, 바이너리 파일을 이미지파일로 변환하기 위해 제안하는 방법은 8비트로 표현되는 최댓값이 0xFF(255)이라는 것과 컴퓨터에서 이미지를 표현할 때 사용하는 RGB Color의 최댓값이 0xFF(255)라는 점에 착안하였다. 따라서, 바이너리 파일의 8비트로 구성된 1바이트 값 자체를 RGB Color의 파라미터로 볼 수 있다. 즉, 바이너리 파일의 첫 번째 바이트와 마지막 바이트까지 RGB Color의 파라미터로 사용하면 바이너리 전체에 대해 바이트 단위의 이미지를 생성할 수 있다. Fig. 4.는 위의 과정을 도식화 한 것이다.

#### 3.1.2 이미지 압축/패딩 기법

CNN 기반의 분석 시스템의 성능을 감안하면 이미지 파일들은 고정 크기 값으로 연산되어져야 하고, 제안하는 정책은 다양한 파일의 크기에 대해 일관성 있게 변환할 수 있다. ILSVRC(Imagenet Large Scale Visual Recognition Challenge)에서 제안된 VGG, Inception와 같은 CNN 모델에서는 299x299 크기의 이미지를 제안하지만 일반적으로 너무 작은 크기의 이미지만 아니라면 이미지 크기 자체에 대한 성능 제약사항은 크게 없는 상황이고, 바이트 시퀀스 기반 기법의 핵심은 바이너리 파일의 데이터를 온전히 이미지로 표현하는 것이므로 본 논문에서는 128x128 크기의 이미지에 대해 바이너리 파일 전체에 대한 압축/패딩 기법을 제안한다. 압축 기법은 128x128(16,384) 바이트 이상의 바이너리 파일이 처리 대상으로, 파일 크기에 대해서 16,384 바이트의 배수를 올림으로 구하여 해당 배수만큼의 바이트 시퀀스의 RGB 평균값을 한 픽셀로 사용한다. 패딩 기법의 경우에는 16,384 바이트 미만의 바이너리 파일이 처리 대상으로, 16,384 바이트가 될 때 까지 제로패딩 한다. Fig. 5.는 위의 과정을 도식화 한 것이다.

#### 3.1.3 바이트 시퀀스 분석 시스템 구조

3.1.1절 및 3.1.2절에서 제안하는 기법들을 사용하면 바이너리 파일에 대해서 일관성 있는 이미지 생성이 가능하다. 본 논문에서는 이러한 이미지와 CNN을 연계하여 악성코드 분석 시스템을 제안하는데, 기존의 연구결과와는 다르게 명확한 이미지 생성 정책과 컴퓨팅 관점에서의 머신러닝 연산 기법을 통해 기존 시스템을 개선하였다. 제안하는 악성코드 분석 시스템은

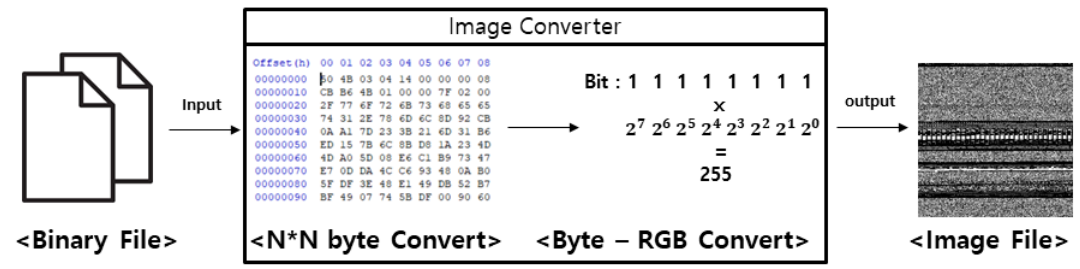



그림 4. 바이너리 파일 이미지 변환 프로세스
Fig. 4. Binary file image conversion process

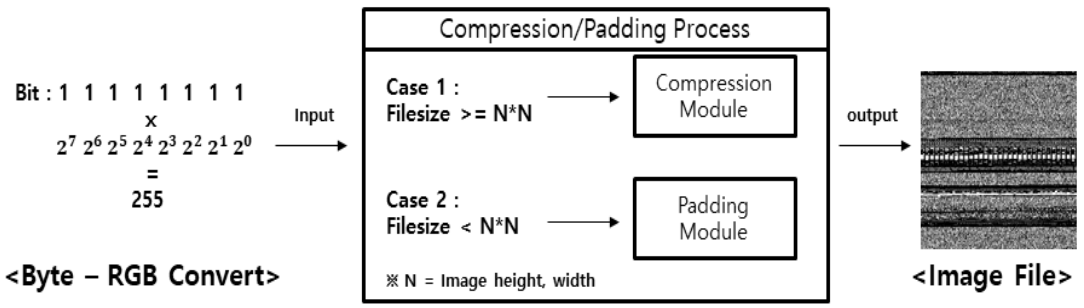



그림 5. 이미지 압축/패딩 프로세스
Fig. 5. Image compression / padding process

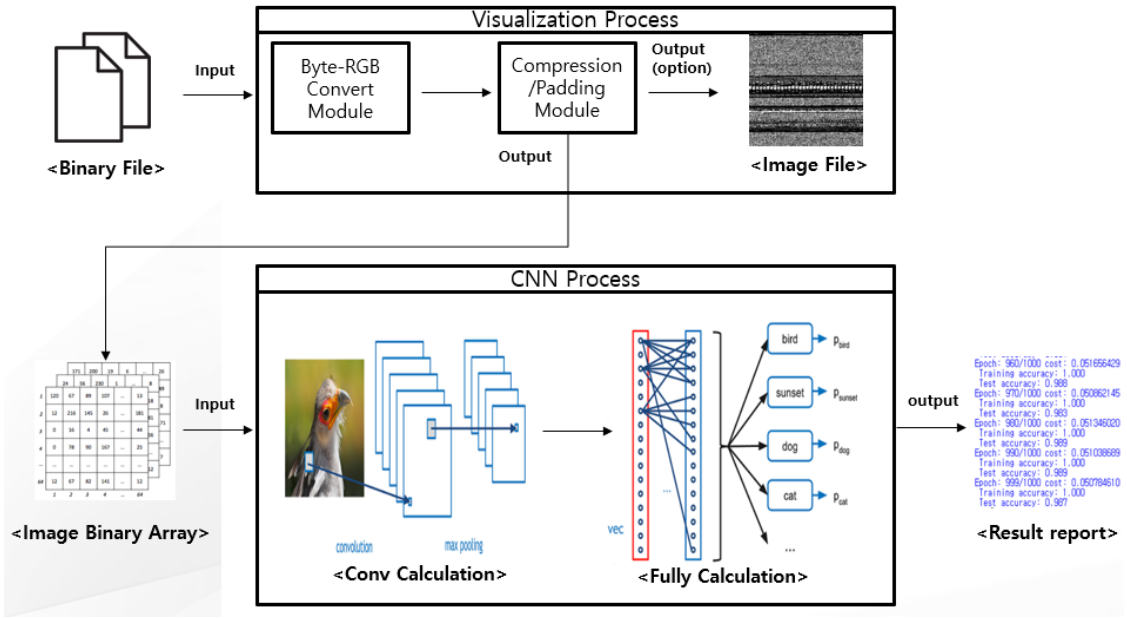



그림 6. 바이트시퀀스 기반 분석 시스템 구조
Fig. 6. Structure of byte sequence-based analysis system

Fig. 6.와 같다.

앞서 기술한바와 같이 우선 바이너리 파일의 바이트 시퀀스를 이용하여 연속적인 RGB 값의 배열로 나타낸다. 다음으로, 파일의 크기에 따라 일관성 있는 처리를 하기 위하여 해당 배열에 압축/패딩 처리를 진행하여 최종적으로 CNN 연산에 사용될 수 있는 입력 형태로 가공되게 되는데, 본 논문에서 제안하는 시스템은 컴퓨팅적 관점에서의 CNN 연산을 통해 성능 향상을 기대한다. 해당 방법의 경우에는 기존 CNN 기법의 경우에 이미지가 입력 포맷이지만 내부 연산은 이미지 각각의 픽셀 값에 대한 배열로서 진행되기 때문에 이미지 처리 과정이 생략하고 바이너리 데이터에 대한 배열 값을 직접적으로 CNN에 입력함으로서 가능하다. 이미지 처리의 경우에는 옵션으로 제공하여 악성코드 분석 전문가 등이 필요로 할 경우 유의미한 정보로 제공 가능할 수 있도록 구성하였다.

### 3.2 정적 특징 값 기반 2-Dimension 악성코드 분석

악성코드 정적 분석의 성능은 일반적으로 파일의 난독화 정도에 의존적이다. 하지만 정적 분석의 고속처리와 낮은 시스템 자원 소모 등의 장점으로 인해 여전히 활발하게 연구 중 인데, 본 논문에서 제안하는 분석기술은 이러한 정적 특징 값을 기반으로 악성코드를 분류한다. 제안기법은 바이너리 파일에서 도출되는 정적 특징 값이 악성코드와 정상파일별로 locality 한 특징을 가지고 있다는 점에 기반하여, 그 특징들이 1차원의 feature 나열이 아니라 상호 연계하여 고차원으로 표현될수록 고도화된 특징으로 나타난다는 것을 가정한다. 특히, 앞서 제시한 바이너리 파일의 데이터 전체를 온전히 이미지화 하는 것이 중요한 바이트 시퀀스 기법에 비하여 이미지 크기가 작고, 그에따라 고속의 분류 속도를 기대할 수 있다. 따라서, 이번 절에서는 정적 특징 값의 2차원 표현 기법으로 정적 분석의 한계와 바이트 시퀀스 기법을 단점을 보완한 악성코드 패킹 분석 기법을 제안한다.

바이너리 파일 내의 정적 특징과 머신러닝의 연계를 통한 기존의 악성코드 분석 기법의 경우에, 각 정적 특징들이 악성코드와 정상파일을 구분 지을 수 있는 특징 값이 일정수준 존재하고 이를 연계 해석하여 feature의 조합으로 악성코드를 분석 및 분류한다. 또, 난독화 관점에서의 feature 조합을 통한 악성코드 분석 연구 등도 존재한다. 하지만, 해당 기법들의 경우에는 기본적으로 다량의 feature를 전수 분석하여야 하고, 단순한 1차원의 feature 나열 값을 머신러닝의 연산으로 가용하기에, 시스템의 학습 환경에 따라 분석 성능이 크게 상이한 경우가 존재한다. 이러한 문제들은 본질적으로 파일 난독화 및 파일 그룹별 locality 한 특성이 원인이다.

Table 1.과 같이 악성코드의 정적 분석결과로 도출

표 1. 바이너리의 정적 feature 예시
Table 1. Examples of static features of binaries

| Feature Group | Description |
|---|---|
| Section | Separate data regions into sections |
| Characteristic | Characteristic of the file, such as read and write permissions |
| API | A set of subroutine definitions, protocols, and tools for building application software. |
| DLL | Microsoft's implementation of the shared library |
| Entropy | Measurement of the randomness |

되는 feature들은 다양하게 존재한다. 기존의 악성코드 정적 feature 연구들의 경우에는 다수의 feature와 이들의 연계해석을 통한 분석 기법이 대부분인데, 본 논문에서는 이러한 특징들 중 section, characteristic 특징만을 이용한 분석 기법을 제안한다. section은 파일을 안정성 있게 사용하기 위해 설계된 PE 파일 구조 중 하나로, 일반적으로 파일의 데이터, 리소스 등을 각각의 다른 section에 분할하여 저장하는 특징이 있고, characteristic은 section의 특징을 나타내는 값으로 execute, read, write와 같은 액세스 권한이나 section이 코드를 포함하고 있는지에 대한 code 값 등이 총 6개 존재한다.

제안 기법은 앞서 설명한 특징을 이용해 2-Dimension 처리를 수행하여 악성코드를 이미지화 하는데, section name을 이용하여 길이 16의 x축을, characteristic의 각 특징을 이용하여 길이 6의 y축을 통해 바이너리 파일의 section, characteristic 특징을 2차원 평면에 맵핑한다. section name의 경우에는 파일 제작자가 임의로 수정할 수 있고 section의 개수 또한 특정하게 지정되어 있지 않은 속성이기 때문에 modulo 연산을 이용해 길이 16으로 지정한다. 다음 Fig. 7.은 section의 맵핑에 대한 처리 방법을 도식화 한 것이다.

section name의 MD5 해쉬 값을 10진수로 변경한 뒤 modulo 16 연산을 통해 길이 16의 x축, characteristic의 특징값 6개를 길이 6의 y축으로 사용하면 다음 그림과 같은 2차원 평면에 이미지화 가능하다. Fig. 8.은 UPX 패커의 section name 중 하나인 UPX0가 characteristic 특징 Write, Read, Execute, Uninitialized data를 나타낼 때, 앞서 기술한 맵핑 처리를 통해 2차원 평면에 맵핑 되는 예시를 나타낸다. 해당 평면은 16x6 크기의 이미지로 간주할 수 있는데, CNN은 일반적으로 정사각형 이미지를 사용하므로 y축은 제로패딩한다. 결론적으로 해당 평면을 16x16 크기의 이미지로 간주하여 3.1절의 바이트 시퀀스 기반 분석시스템과 동일하게 해당 평면의 배열 값들을 CNN 기반의 악성코드 유사도 분석 시스템으로 연산 가능하며, 마찬가지로 이미지 변환은 옵션으로 제공하여 경량화된 연산 및 이미지 정보 지원이 가능하다. 다음 Fig. 9.는 제안하는 2-Dimension 분석 기법을 도식화 한 것이다.

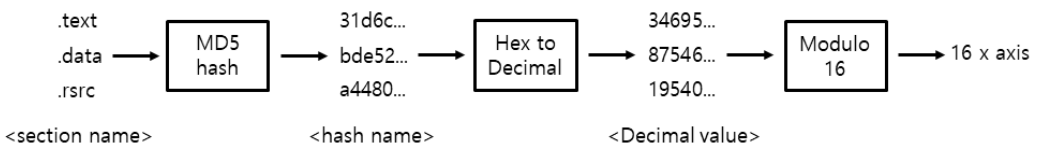



그림 7. 정적 feature 값의 2차원 표현 프로세스
Fig. 7. 2D representation process of static feature values

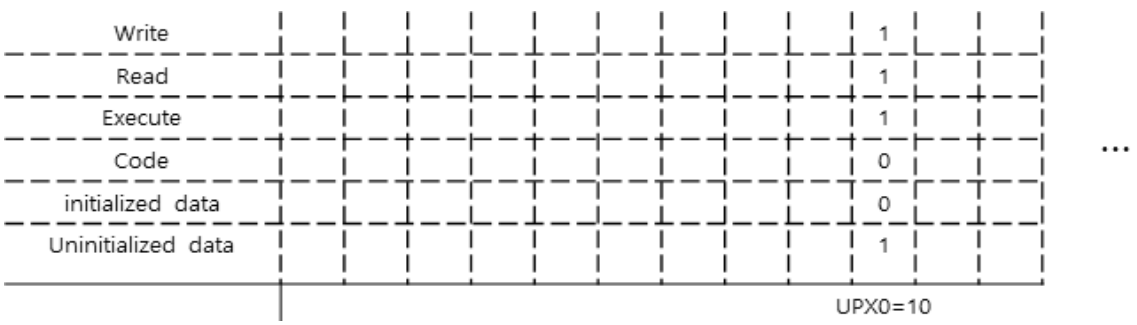



그림 8. 정적 feature 값의 2-Dimensional 표현 예시
Fig. 8. Example of 2-Dimensional representation of static feature values

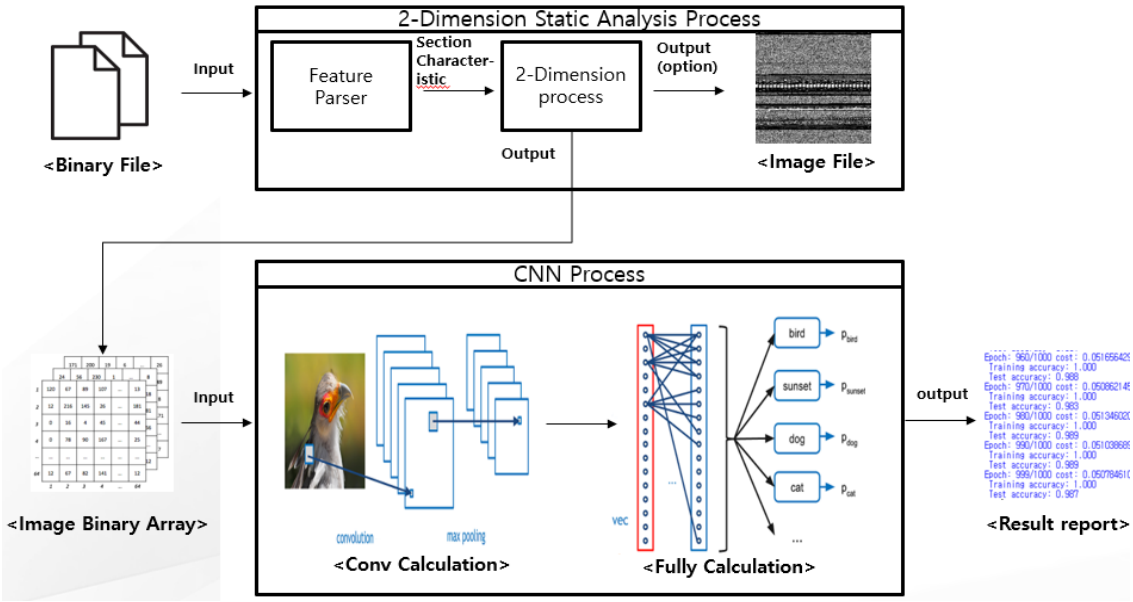



그림 9. 2-Dimension 기반 분석 시스템 구조도
Fig. 9. 2-Dimension-based analysis system structure diagram

## Ⅳ. 실험결과

이번 절에서는 본 논문에서 제안하는 기법들의 분석성능을 기술하도록 한다. 성능측정은 실제 상용환경에서의 시스템 운용을 가정하여 일반적인 PC 환경인 2.30GHz 듀얼코어 CPU, 8GB ram과 windows 10 환경에서 진행되었다. 4.1절에서는 성능측정을 위한 데이터 및 CNN 구조 세부사항에 대해 기술하고, 4.2절에서는 제안 시스템의 성능측정 결과에 대해 제시한다.

### 4.1 데이터 셋 및 CNN 구조 세부사항

제안 시스템은 성능측정을 위하여 상용환경의 악성코드 및 정상파일을 트레이닝 셋으로 4481개, 테스트 셋으로 1121개 사용하여 총 5602개의 파일로 실험을 진행하였다. 제안 시스템으로 분류할 패커 그룹의 경우에는 pedump로 데이터셋에 적용된 패커를 우선적으로 식별하여 사용빈도 상위 4개 패커 및 borland, visual C 컴파일러 그룹, 언패킹 그룹과 해당 그룹에 속하지 않은 etc 그룹으로 지정하였고, 시스템으로 분류할 8개 class와 각각의 개수는 Table 2.와 같다.

CNN 구조의 경우에는 2개의 convolution layer와 2개의 fully connected layer를 사용하게 되는데 시험 시스템의 경우에는 convolution layer 연산 후에 max pooling, drop out 을 적용한다. 세부 파라미터는 Table 3.과 같다.

표 2. 데이터셋의 패커 및 컴파일러 구성
Table 2. Configuring Packers and Compilers in Datasets

| Class | Number of File |
|---|---|
| unpack | 141 |
| upx | 2,082 |
| aspack | 158 |
| nspack | 241 |
| pecompact | 1,341 |
| msvisualC | 195 |
| borland | 26 |
| etc | 1,418 |

표 3. CNN 파라미터 표
Table 3. CNN parameter table

| Parameter | Constant |
|---|---|
| Number of Filter | 128 |
| Filter Size | 3 |
| Stride | 2 |

### 4.2 패킹 분석 기법 시험결과

CNN 기반의 패킹 분석 기법 시험결과로는 앞서 기술한 바이트시퀀스 및 2-Dimension 기법에 대한 속도측정 및 분류정확도를 도출하였다. 이때, 속도측정의 경우에는 해당 기법이 동작하기 위한 전 과정이 포함된 시간을 의미한다. 분류정확도의 경우에는 데이터에 대한 실제값과 알고리즘에서의 측정값 간의 차이

를 의미하는 것으로 다음 수식(1)과 같이 정의할 수 있다.

CNN 기반의 패킹 분석 기법 시험결과로는 앞서 기술한 바이트시퀀스 및 2-Dimension 기법에 대한 속도측정 및 분류정확도를 도출하였다. 이때, 속도측정의 경우에는 해당 기법이 동작하기 위한 전 과정이 포함된 시간을 의미한다. 분류정확도의 경우에는 데이터에 대한 실제값과 알고리즘에서의 측정값 간의 차이를 의미하는 것으로 다음 수식(1)과 같이 정의할 수 있다.

Accuracy=(TruePositive+TrueNegative)/(TruePositive+FalsePositive+TrueNegative+FalseNegative) (1)

동일한 시스템 환경에서 학습횟수 1,000번으로 훈련 및 테스트 셋 모두에 대해 측정한 시험 결과는 Table 4.와 같다. 또한, Fig. 10.과 Fig. 11.은 학습횟수 1,000번 동안의 바이트시퀀스, 2-Dimension 각각의 훈련 및 테스트 셋에 대한 분류정확도를 나타낸 것이다. 30epoch 이후로의 데이터는 이전과 동일한 양상의 반복적인 분류정확도로 수렴한다고 판단 가능하므로 관련 그림 표현상의 문제로 생략하였다.

제안 기법의 경우에 바이트시퀀스 기법에 비해 정확도 성능은 3% 정도 향상되었고 속도측면에서 약 10배 우수한 성능을 나타낸다. 해당 수치는 제안 기법이 패커 분류기로 바이트시퀀스와 동일하게 CNN을 분류기로서 사용하지만, 이미지 처리를 하지 않기 때문에 해당 기능에 대한 별도의 연산이 필요없고 데이터 자체도 128x128 크기의 바이트시퀀스에 비해 16x16 크기의 배열을 사용하기 때문에 연산 비용이 저렴한 것으로 판단된다.

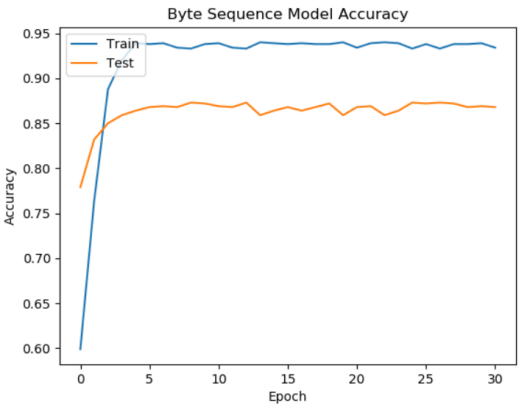



그림 10. 바이트시퀀스 모델 분류정확도
Fig. 10. Byte Sequence model accuracy

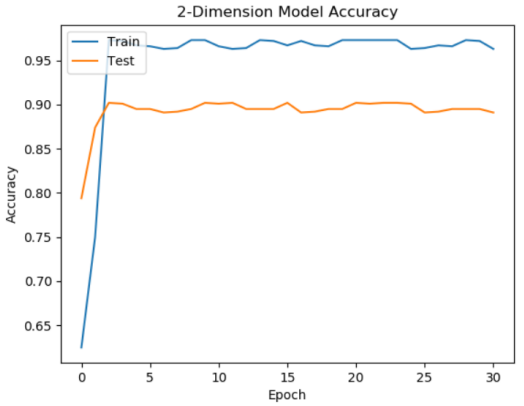



그림 11. 2-Dimension 모델 분류정확도
Fig. 11. 2-Dimension model accuracy

Table 5.는 두 기법에서 옵션으로 제공되는 이미지화 기능을 사용하여 랜덤하게 그룹별 이미지를 3장씩 도출한 결과이다.

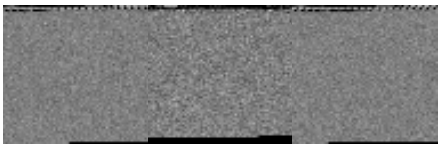
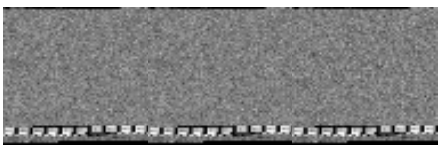
표 4. CNN 기반의 패킹 분석 기법 테스트 결과
Table 4. Test results of CNN-based packing analysis technique

| | Byte Sequence | 2-Dimension |
|---|---|---|
| Accuracy(Training) | 93.8% | 96.9% |
| Accuracy(Test) | 87.0% | 89.5% |
| Speed (File/Second) | 10.2s | 1.07s |

## V. 결 론

본 논문에서는 보안위협의 주요 원인인 악성코드에 적용되는 분석 및 탐지를 어렵게 만들고 회피하기 위해 적용되는 난독화 기법 중 하나인 패킹 기법에 대응하기 위한 기법을 다루었다. 특히, 패킹기법의 경우에는 악성코드에 적용이 간단하기 때문에 근래에 폭발적으로 출현하는 악성코드에 대부분 적용되어 있다. 이러한 경향은 다량의 악성코드에 대응하기 위해서 악성코드 자동화 분석 시스템의 중요성이 대두되고 상황에서 운용 시스템의 성능 제약조건으로 인해 널리 사용되는 정적 기반 분석 시스템에 막대한 성능 저하를 유발시킨다. 본 연구는 이러한 악성코드 패킹에 대응하기 위해 기존에 악성코드의 그룹분류에 주로 가용되던 CNN 모델을 패커 분류기로 채택하여 이를 연계한 2-Dimension 분석 기법을 본문에서 제안하였다. 2-Dimension 기법의 경우에는 기존의 정적 특성

표 5. 패킹 분석 기법 처리 중 도출된 이미지 예시
Table 5. Examples of image by packing analysis technique

| | Byte Sequence | 2-Dimension |
|---|---|---|
| unpack |  |  |
| upx |  |  |
| aspack |  |  |
| nspack |  |  |
| pecompact |  |  |
| msvisualC |  |  |
| borland |  |  |
| etc |  |  |

을 1차원 벡터의 조합으로 분석 성능을 고도화 시키는 DNN과 달리 극소수의 feature 그룹의 조합으로 유의미한 성능을 나타내었고, 이는 일반적으로 고차원의 feature 조합이 해당 군집의 locality한 특징들을 잘 나타내는 경향이 있어 군집간 경계가 좀 더 명확한 것과 유사하다. 또, 제안기법은 이러한 특징과 더불어 본문에서 제시한 컴퓨팅적 관점의 시스템 처리 및 파일당 16x16 바이트 크기의 적은 양의 데이터를 사용함으로써 바이트시퀀스 기법과 비교하였을 때 보다 적은양의 연산으로 개선된 성능을 나타낼 수 있었다. 반면에, 시험에 사용된 데이터 양이 적어 추후 대량의 데이터를 사용하여 보다 객관적인 성능측정이 필요하며 CNN 구조 또한 기본적인 layer 구조만 가지고 있어 머신러닝 관점에서의 시스템 개선 또한 필요하다.

향후 이를 보완하고, 고성능/고차원의 정적 feature 조합의 연구를 병행한다면 제안 기법은 자동화된 악성코드 탐지 시스템에 패커를 우선적으로 식별해주는 pre-filter 시스템으로 동작할 수 있을 것이라 기대하며 악성코드 그룹분류 또한 제안하는 매커니즘으로 우수한 성능을 나타낼 수 있다고 판단한다.

## References


[1] M. Bat-Erdene, T. Kim, H. Park, and H. Lee, "Packer detection for multi-layer executables using entropy analysis," *Entropy*, vol. 19, no. 3, 2017.

[2] C. S. Wright, "Packer analysis report debugging and unpacking the NsPack 3.4 and 3.7 packer," *SANS Reading Room*, Aug. 2010.

[3] W. Yan, Z. Zhang, and N. Ansari, "Revealing packed malware," *IEEE Secur. & Privacy*, vol. 6, no. 5, 2008.

[4] M. M. K. Al-Anezi, "Generic packing detection using several complexity analysis for accurate malware detection," *Int. J. Advanced Comput. Sci. and Appl.*, vol. 5, no. 1, 2014.

[5] S. Yue, "*Imbalanced Malware Images Classification: a CNN based Approach*," arXiv preprint arXiv:1708.08042, 2017.

[6] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proc. VizSec '11*, Pittsburgh, Pennsylvania, USA, Jul. 2011.

[7] S. Seok and H. Kim, "Visualized malware classification based-on convolutional neural network," *J. Korea Inst. Inf. Secur. & Cryptol.*, vol. 26, no. 1, pp. 197-208, 2016.

[8] W. Jung, S. Kim, and S. Choi, "Poster: Deep learning for zero-day flash malware detection," *36th IEEE Symp. Secur. and Privacy*, 2015.

[9] E. Gandotra, D. Bansal, and S. Sofat, "Malware analysis and classification: A survey," *J. Inf. Secur.*, pp. 56-64, 2014.

[10] J. Z. Kolter and M. A. Maloof, "Learning to detect and classify malicious executables in the wild," *J. Machine Learning Res.*, pp. 2721-2744, 2006.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Inf. Process. Syst.*, 2012.

[12] P. Sukumar, R. K. Gnanamurthy, "Computer aided detection of cervical cancer using pap smear images based on adaptive neuro fuzzy inference system classifier," *J. Med. Imaging and Health Informatics*, vol. 6, no. 2, pp. 312-319, 2016.

[13] Y. Prayudi and I. Riadi, "Implementation of malware analysis using static and dynamic analysis method," *Int. J. Comput. Appl.*, vol. 117, no. 6, 2015.

[14] A. Pfeffer, et al., "Malware analysis and attribution using genetic information," *IEEE MALWARE*, pp. 39-45, Fajardo, PR, USA, Oct. 2012.

[15] K. Kancherla and S. Mukkamala, "Image visualization based malware detection," *IEEE CICS*, pp. 40-44, Singapore, Apr. 2013.

[16] D. Steyrl, R. Scherer, J. Faller, and G. R. Müller-Putz, "Random forests in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier," *Biomed. Eng.*, vol. 61, no. 1, pp. 77-86, 2016.

[17] S. Z. M. Shaid and M. A. Maarof, "Malware behavior image for malware variant identification," *IEEE ISBAST*, 2014.

[18] D. Kong and G. D. Yan, "Malware distance

learning on structural information for automated malware classification," in *Proc. ACM SIGMETRICS/ Int. Conf. Measurement and Modeling of Comput. Syst.*, pp. 347-348, 2013.

[19] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based vision systems for place and object recognition," in *Proc. ICCV*, 2003.

[20] J. Jang, D. Brumley, and S. Venkataraman, "Bitshred: Feature hashing malware for scalable triage and semantic analysis," in *Proc. 18th ACM Conf. Computer and Commun. Secur., ACM*, pp. 309-320, Oct. 2011.

**황 준 호** (Jun-ho Hwang)

2018년 : 호서대학교 정보보호학과(공학사)

2018년~현재 : 호서대학교 일반대학원(정보보호학) 석사과정

<관심분야> 악성코드 분석, 파일분석, 정보보호

**이 태 진** (Tae-jin Lee)

2003년~2017년 : 한국인터넷진흥원 팀장

2017년~현재 : 호서대학교 정보보호학과 교수

<관심분야> 시스템 보안, 악성코드 분석, 침해사고 대응