딥러닝을 이용한 모바일 환경에서 변종 악성코드 탐지 알고리즘

우성희[1] · 조영복[2]

[1]한국교통대학교 · [2]대전대학교

Algorithm for Detecting Malicious Code in Mobile Environment Using Deep Learning

Sung-hee Woo[1] · Young-bok Cho[2]

*Dept of Medical IT Engineering, Korea National University of Transportation,

E-mail : shwoo@ut.ac.kr

요 약

제안 논문은 딥러닝 알고리즘을 이용해 모바일 환경에서 변종 악성코드 탐지 알고리즘을 제안한다. 안드로이드 기반의 행위 기반의 악성코드 탐지 방법의 문제점을 해결하기 위해 시그니처 기반 악성코드 탐지방법과 머신 러닝(Machine Learning)기법을 활용한 실시간 악성파일 탐지 알고리즘을 통해 높은 탐지율을 증명하였다.

ABSTRACT

This paper proposes a variant malicious code detection algorithm in a mobile environment using a deep learning algorithm. In order to solve the problem of malicious code detection method based on Android, we have proved high detection rate through signature based malicious code detection method and realtime malicious file detection algorithm using machine learning method.

키워드

딥러닝 알고리즘, 악성코드, 시그니터 기반 악성코드 탐지, 악성코드 이미지화

I. 서론

IT 기술 및 다양한 보안 제품과 기술이 발전함에 따라 악성코드에서 사용하는 기술 또한 정교해지고 있다. 특히 악성코드 제작자는 자신이 유포하려는 악성코드가 보안 장비나 제품에서 탐지되지 않도록 다양한 탐지 회피 기법을 적극적으로 활용하고 있다. 회피 기법 중 하나가 바로 파일의 외형은 가급적 정상 파일과 비슷하게 유지하면서, 악성 기능을 하는 코드를 내부에 은닉하는 방법이다. 파일 외부에서 확인 시 악성 기능에 해당하는 코드● 데이터가 바로 발견되지 않도록 인코딩 등의 과정을 거쳐 은닉하는데, 악성코드 종류에 따라 매우 다양한 데이터 은닉 기법이 발견되고 있다.법을 사용하고, 잡음 영상에서 정보 손실이 나타나는 드랍아웃과 같은 영역에서 분할의 오류를 개선하기 위해 곡선적합을 이용하여 분할의 잘못된 결과를 보정함으로써 경계선의 완만성을 향상시켰다. 또한 잡음 영상의 특성과 방향성을 고려한 영상 특징을 사용하여 경계선 분할을 통해 영상 획득 시 발생하는 다양한 잡음의 종류와 방향성을 갖는 대상 객체의 경계선에서 강건하고 정확한 분할 결과를 제공한다. 아울러, 대상 구조의 지역적 응집성 크기와 적분 영상을 적용함으로써 영상 특징 추출에 소요되는 계산시간을 최소화하였다. 악성 데이터를 비트맵 이미지(BMP) 형태로 위장한 형태의 악성코드를 알아보기에 앞서 우선 비트맵 이미지에 대한 기본적인 이해가 필요하다. 비트맵(BITMAP)은, 비트(BIT)와 맵(MAP)의 합성어로, 일련의 비트가 나열된 데이터의 집합을 의미한다. 비트는 컴퓨터에서 데이터를 표현하는 최소 단위이며 ‘0’과 ‘1’로 구성된다. ‘비트맵’으로는 단순한 데이터를 저장할 수도 있지만, 이미지 데이터를 저장할 경우에는 ‘비트맵 이미지’로 해석될 수 있다.

비트맵 이미지 구조는 [그림 1]에서 보듯 비트맵 파일헤더(BitmapFileHeader)와 비트맵인포헤더

(BitmapInfoHeader), 픽셀어레이(PixelArray)로 이루어진다. 비트맵파일헤더와 피트맵 인포헤더는 운영체제나 응용 프로그램에서 비트맵 이미지를 제대로 해석하기 위해 필요한 정보로, 운영체제에 따라 일부 차이가 있다. 이 글에서는 윈도우(Windows)를 기준으로 설명한다.

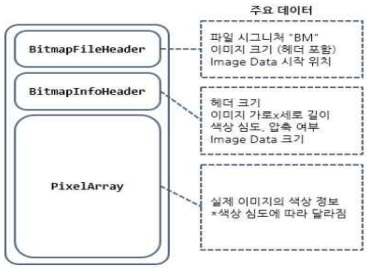


Fig. 1. 비트맵 이미지 구조

Ⅱ. 딥러닝을 이용한 모바일 환경에서 변종 악성코드 탐지 알고리즘

본 논문에서는 딥러닝을 이용한 모바일 환경에서 변종 악성코드를 탐지하는 알고리즘을 제안한다. 따라서 비트맵 이미지 구조는 그림 1에서 보듯 비트맵 파일헤더와 비트맵인포헤더, 픽셀어레이로 구성되고 비트맵파일헤더와 피트맵인포헤더는 운영체제나 응용 프로그램에서 비트맵 이미지를 제대로 해석하기 위해 필요한 정보로, 운영체제에 따라 일부 차이가 있다. 이 글에서는 윈도우를 기준으로 설명한다.

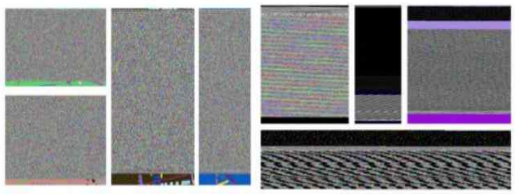
2.1 악성코드 내부의 비트맵 이미지

비트맵 이미지에 대한 구조와 이미지의 색상이 표현되는 방법을 바탕으로 악성코드는 다음과 같이 두 가지 유형으로 분류가 가능하다.

유형 1 - 실행 파일 리소스 내부에 비트맵 이미지를 포함하는 경우

유형 2 - 비주얼베이직폼 내부에 비트맵 이미지를 포함하는 경우

각 유형의 내부에 포함된 비트맵 이미지만 확인해보면 그림 2와 같다. 어떠한 형태를 표현하는 것이 아니라 여러 가지 색상의 점으로만 이루어져 있다. 이러한 특징이 나타나는 이유는 픽셀어레이의 데이터가 색상을 표시하기 위한 값이 아니라 인코딩된 데이터의 일부이기 때문이다. 즉, 색상값이 아닌 데이터를 비트맵 이미지의 픽셀어레이에 포함시켰기 때문에 데이터가 색상값으로 강제 해석되면서 [그림 2]와 같이 표현된 것이다.



유형1 유형2

Fig. 2 유형 1 악성코드와 유형 2 악성코드 내부 비트맵 이미지 예

유형 1과 유형 2는 다양한 색의 점으로 구성됐다는 점에서는 유사한 형태라고 볼 수 있다. 하지만 자세히 살펴보면 몇 가지 차이점이 존재한다.

유형 1 악성코드의 특징은 섹션의 개수, 진입점(EntryPoint) 부분, 코드의 흐름 등 구조적으로 비주얼 스튜디오(Visual Studio)로 컴파일된 실행 파일의 특징을 보인다. 총 4개의 섹션 중 리소스(.rsrc) 섹션에 다양한 리소스 엔트리를 포함하고 있다. 또한 비트맵 리소스 엔트리 내부에 랜덤 픽셀 이미지를 포함한다. 같은 유형의 악성코드일지라도 파일간에 차이가 존재한다. 비트맵 리소스 엔트리 내부에 한 개의 이미지만 포함하는 경우도 있고, 다수의 이미지를 포함하는 경우도 있다. 유형 1 악성코드의 주요 특징은 비트맵 리소스 엔트리 내에 랜덤 픽셀 이미지가 한 개는 반드시 존재한다는 점이다.

2.2 유사한 비트맵 이미지를 포함하지만 랜덤 픽셀 이미지는 없는 경우

유형 1과 거의 비슷하지만, 결정적으로 랜덤 픽셀 이미지가 존재하지 않는 경우다. 인코딩된 악성 데이터는 리소스 영역의 비트맵 엔트리 대신 임의의 이름을 가진 엔트리로 존재하며, 해당 데이터가 이미지로 해석되지 않는다. 이런 유형의 악성코드에서는 그림 3에 나타난 것과 같이 유형 1과 유사하게 다양한 색이 조합된 이미지 파일을 확인할 수 있다.

Fig. 3 덤 픽셀 이미지가 없는 유형1의 변종 악성코드

콜백 함수를 통한 코드 실행을 하지만 랜덤 픽셀 이미지가 없는 경우 비주얼베이직 외형인 점과 CallWindowProcW 등의 API를 사용하여 콜백 방식으로 악의적인 코드를 실행하는 기법은 동일하지만, 랜덤 픽셀 이미지를 내부에 포함하지 않는 점

이 특징인 경우다. 해당 기법을 사용하는 악성 파일 내의 코드 일부를 나타낼 수 있다.

III. 결론

PC와 인터넷의 급속한 발전은 우리에게 편리함을 줌과 동시에 수많은 악성코드들을 만들어 냈으며, 해마다 그 악성코드의 양은 크게 증가하고 있다. 시그니쳐(Signature) 기반 탐지 방법은 악성코드 파일을 유니크(Unique)하게 식별하기 위해 사용되는 방법으로, 시그니쳐는 백신 프로그램이 파일들을 스캔할 때 해당 파일을 유일하게 식별할 수 있도록 사용되는 특정 데이터 부분을 말한다. 여기서 시그니쳐는 패턴이라는 이름으로 불리기도 한다 일부 변종들이 주로 코드부분은 동일하지만 데이터부분의 데이터들을 조금씩 바꾸는 경우가 많은 것에 착안하여 PE 구조의 코드 섹션 영역만을 해쉬하여 시그니쳐로 사용하기도 한다. 탐지의 속도 및 정확성을 증가시키기 위해 파일의 특정 위치로부터 특정 범위까지의 해쉬값을 시그니쳐로 사용하는 방식은 국내 백신업체들이 가장 많이 사용하고 있는 방식으로 주로 파일이 실행될 때 코드의 시작지점인 엔트리 포인트를 기준으로 특정 위치를 지정하고, 그 위치로부터 특정 범위의 값을 CRC, MD5 등과 같은 해쉬값으로 계산하여 시그니쳐를 생성하는 방식이다. 시그니쳐를 사용하여 정확하게 탐지되지 않는 악성코드는 정의된 의심스러운 기준들의 휴리스틱 룰셋과 비교하여 탐지하게 된다. 특정한 코딩 기법의 사용, 의심스러운 것으로 생각되는 행위와 구문들과 그러한 의심스러운 행위들의 조합 등을 통해 해당 파일에 대한 위험을 정의하고 변종 및 유사한 악성코드들을 딥러닝기반으로 탐지한다.

기존의 바이러스 백신이나 안티스파이웨어 등의 보안 프로그램은 이미 알려진 형태의 악성 프로그램에 대한 정보(예를 들면, 이진 코드 중의 특정 부분에 대한 패턴 정보)를 기초로 해당 악성 프로그램을 검출하거나 그 실행을 차단하는 방식으로 작동 한다.본 논문에서는 악성코드를 탐지하기 위한 비시그니처 기반 머신 러닝 방법을 제안한다. 제안한 방법은 실행파일에서 추출된 각 특징을 학습하고 변종유무를 판단한다. 악성코드의 패 반복 영역을 단순히 시그니터로 추출하여 생성함으로 정확도를 향상시켰다.

References

[1] E.K. Kamundala and C. H. Kim, “CNN model to classify malware using image feature.” 정보과학회 컴퓨팅의 실제 논문지, 제 24권 5호, pp.256-261, 2018.

[2] Y.B. Cho "The malware detection using deep learning based R-CNN."한국디지털콘텐츠학회 논문지, vol 19, no6. pp. 1177-1183, 2018

[3] J. Bai, J. Wang, and G. Zou, "A Malware Detection Scheme Based on Mining Format Information", The Scientific World Journal, pp.1-11, May 2014.

[4] ENISA Threat Landscape Report 2016, https://www.enisa.europa.eu/publications/enisa-threat-l andscape-report-2016, 2017, Feb. 8.