Journal of Knowledge Information Technology and Systems(JKITS), Vol. 14, No. 3, pp. 211~219, June 2019
https://doi.org/10.34163/jkits.2019.14.3.001

Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

http://www.kkits.or.kr

Malicious Code Neutralizing Method Using Image Format Transforming Based on Nonlinear Transfer Function

Dong-Seob Jung[1], **Sang-Joon Lee**[*2]

[1]*HUNESION Co., Ltd.*
[2]*School of Business Administration, Chonnam National University*

ABSTRACT

Various bypass techniques have been developed to hide malicious code in image files among non-executable files. It is difficult to detect by reputation or signature-based antivirus methods when unknown malware is hidden. In this paper, we proposed a neutralizing method of hidden malicious code to analyze the structure of the original image file format and disable the malicious code through image data area conversion even if there is no prior information about the signatures or characteristics of malicious codes. The proposed method consists of image file extraction phase, file format analysis phase, file transformation phase, and management phase of transformation image file. In the image file transformation phase, header information transformation, specific string filtering transformation for additional information, and image pixel data transformation using nonlinear transfer function are performed. In order to prove the effectiveness of the proposed method, 10 malicious code - hidden image files among 48,220 of the latest malicious code (paid API) purchased from Virus Total Company were used in the experiment. After the file extraction phase, the format analysis phase, and the image file conversion phase for the neutralizing method, the experimental results show that the virus detection amount is reduced quantitatively and thus the effectiveness of the proposed method is verified. In addition, by using the non-linear transfer function, the converted image file was able to neutralize the malicious code while maintaining the same quality as the original image, which could not be distinguished by the naked eye.

KEYWORDS Neutralization, Malicious code images, Steganography, Antivirus, Image transform

ARTICLE INFO: Received 15 March 2019, Revised 28 April 2019, Accepted 7 June 2019.

[*]Corresponding author is with school of business administration, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju, 61186, KOREA.
E-mail address: s-lee@chonnam.ac.kr

1. 서 론

글로벌 백신 조사단체인 'AV-Test.org' 에 따르면 매일 350,000개의 새로운 악성코드가 생겨나고 있고, 2015년 470만개 였던 것이 2018년에는 856만개로 갯수가 급속도로 증가되고 있다[1]. 이처럼 점점 지능화 되고 있는 사이버 공격과 보안사고 급증에 따른 대책으로 공공기관 및 금융권을 중심으로 망분리가 구축돼 왔고, 망분리의 보안성을 유지하면서 업무 편의성를 향상시키기 위하여 망간자료전송 솔루션이 도입되어 운영되고 있다[2].

망간자료 전송시 보안 정책에 따라 악성코드 검사를 필수적으로 수행하게 한다. 기존의 안티 바이러스 백신은 알려진 정보에 의한 평판이나 시그니처를 사용한다. 콘텐츠 무해화 및 재조합 기술에서는 문서형 파일 내 Macro, JavaScript와 같은 액티브 콘텐츠를 제거할 수 있다. 또한, DLP/개인정보 탐지 기술을 이용하여 파일 내 기밀 문서나 개인정보 유출을 방지하고 있다. 최근에는 머신러닝 기술을 적용하여 알려지지 않은 악성코드 파일을 분류하기도 한다.

비실행형 파일 중에서 BMP, JPG, PNG 등과 같은 이미지 파일에 악성코드를 은닉시키는 다양한 우회 기법 연구가 진행되어 왔다. 망분리 환경에서 반출입하는 파일이나 바이러스토탈과 같은 글로벌 바이러스 분석 서비스에서도 이미지 파일에 악성스크립트가 포함된 경우가 다수 발견되고 있다[3, 4]. 알려진 악성코드 은닉형 이미지 파일에 안티바이러스 백신에서는 평판 및 시그니처를 활용하여 탐지하고 보안문제를 처리하도록 하고 있다. 하지만, 잘 알려지지 않은 악성코드가 숨겨진 이미지 파일은 평판 정보나 시그니처가 없어서 안티바이러스 기술로는 탐지되지 않는다. 특히, 스테가노그래피 방식은 악성코드를 이미지 파일에 은닉시켜 전파(Stegosploit, Shellcode hiding 등)하거나 기밀정보를 이미지 파일에 은닉시켜 의도적인 정보 유출에 악용될 수 있다. 하지만 존의 기술로는 다양한 은닉 알고리즘이 사용된 스테가노그래피 공격을 탐지하기가 매우 어렵다. 이미지 전체의 그룹 인텍스를 랜덤으로 재처리한 정보 은닉 방지 방법이나, 스테가노그래피 부호화 및 복호화 결과를 비교하여 판별하는 방법도, 정보 은닉 여부를 정확히 탐지하고 판별할 수 있는 능력에 한계가 있어서, 보다 효과적으로 대응할 수 있는 새로운 접근방식의 연구가 필요하다.

본 논문에서는 악성코드의 평판이나 시그니처와 같은 사전 정보가 없더라도, 원본 이미지 파일 포맷의 구조를 분석하고, 비선형 전달함수를 이용하여 이미지 파일의 데이터 영역을 변환하는 방법을 통해 이미지 파일에 숨겨진 악성코드를 무력화시키는 방안을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서 이미지 파일 내 악성코드 은닉 기법과 이미지 파일내 악성코드 방지 및 판별기법들을 살펴보고, 3장에서는 이미지 악성코드 은닉 유형과 이미지 악성코드 은닉을 무력화하는 방법을 기술한다. 4장에서는 실험 및 평가를 진행하고, 5장에서 결론 내용을 기술한다.

2. 관련연구

2.1 이미지 파일 악성코드 은닉 기법

이미지 파일에 은닉된 악성코드 데이터가 어떻게 존재하는지 <그림 1>을 통해 살펴보면, 일반적인 이미지 파일(BMP, JPG, PNG등)은 파일 첫 부분에 헤더가 있고, 그 이후 부분에 실제 데이터가 있다. 이런 이미지 파일구조를 이용해 이미지 포맷은 정상적으로 구성되어 있지만 이미지 데이터의 끝에 악성코드 바이너리나 악성코드 스크립트 등을 추가하는 기법, 이미지 파일 포맷의 부가 정보영역

에 악성코드 스크립트를 삽입하고, 이미지 데이터 영역에 악성코드를 은닉시키는 방법 등의 은닉기법이 있다.

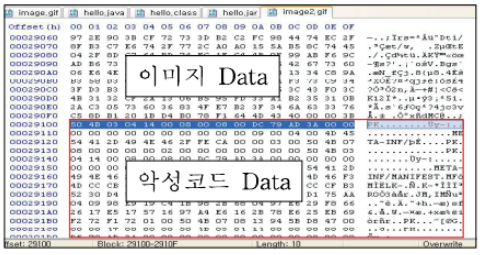

그림 1. 이미지 파일내 은닉된 악성코드 데이터
Figure 1. Hidden Malicious Code in Images Data

만약, 실제 데이터가 위치하고 있는 정상 이미지 파일의 맨 마지막 위치에다 악성코드를 삽입하면 원본 이미지의 손상 없이 악성코드 은닉이 가능하다[5].

이미지 쉘코드 은닉 기법을 활용한 선행 연구가 있다[6]. 이 연구에서는 24Bit BMP 이미지에 쉘코드를 은닉하여, 기존의 악성코드 탐지 기법에 발각되지 않음을 확인하였다. 이 기법을 적용하기 위해서 이미지 디코더 저장소와 세 개의 모듈(스캐닝, 결정, 숨김)을 구성하였다. 동작 순서는 디코더 저장소로부터 24Bit 이미지를 디코더로 입력받은 후, 이미지를 검색하여 결정 모듈과 통신을 통해 결정 작업을 반복한 후, 더미코드나 점프코드를 생성한다. 이렇게 생성된 이미지는 원본 이미지와 분간이 어렵고, 시그니처가 없기 때문에 시그니처 기반 방법으로 탐지가 안되고, 에뮬레이션 탐지방법으로는 에뮬레이션 시간이 길어서 실시간 탐지가 어렵게 된다.

최근에는 연구되는 Stegosploit 기술이 있다. Stegosploit은 Steganography와 Exploit가 결합된 용어이다. 스테가노그래피가 사용된 이미지 파일을 실행하면, 이미지 안에 숨겨진 스크립트가 실행되어, 다양한 익스플로잇 공격이 시도될 수 있다. 스크립트에 숨겨진 익스플로잇의 유형과 사용자 환경에 따라 다양한 피해가 발생 할 수 있다. 네트워크 트래픽 관점으로 봤을 때 Stegosploit는 단지 이미지 파일이지만, 픽셀에 스크립트가 숨겨져 있으며, 외관상으로는 유해여부를 구분하기 어렵고, 단지 이미지를 보기만하여도 삽입된 스크립트가 실행되는 특징을 갖고 있다. 최근에는 스테가노그래피 기술을 CCTV카메라, 스마트TV, 스마트 홈과 같은 IoT 환경, 모바일앱, VoIP, P2P 파일공유시스템, DNS, IPv6 프로토콜에 적용한 지능화된 공격으로 진화해 가고 있다[7, 8]. 이러한 공격은 이미지, 오디오, 비디오 등의 파일에 악성 정보를 은닉시켜 위험을 주고 있다.

2.2 이미지 파일 악성코드 은닉 방지 및 판별 기법

이미지 파일 악성코드 은닉 방지 기법으로 이미지 인덱스를 무작위로 섞은후 재처리하는 방식이 있다[9]. 이 방법에서는 이미지 전체를 16개 그룹으로 나누고, 재처리된 이미지 인덱스를 저장한다. 명암의 분포에 특정 규칙을 주지 않는 방법을 이용하며, 온라인 이미지 등에 정보를 은닉하고, 나중에는 역으로 추출하는 정보은닉을 막을 수 있다.

스테가노그래피 은닉에서는 이미지에 악성코드가 삽입되면 일반적으로 픽셀 값의 퍼짐성 효과가 나타난다. 원본 이미지에 비해 은닉자료가 5%에서 10% 사이의 용량의 코드를 삽입하고 있다[10]. 만약 메시지 용량이 충분하다면 통계적 기법인 카이스퀘어 테스팅을 통해 은닉코드 존재를 감지할 수 있다. 그러나 원본 이미지에 비해 은닉자료가 매우 작게 혼합되어 있을 때 기존의 이웃한 2개의 픽셀 값을 이용한 카이스퀘어 적합도 검증을 통해 은닉자료를 감지하고 위치를 찾아낼 수 있다.

이미지 파일 악성코드 은닉을 판별하는 일반적인 기법에서는 숨기려는 정보를 미디어 데이터에 교묘하게 삽입한 스테고(stego) 영상을 직접 분석하여 부호화 여부를 판별하는 방식을 사용하였다. 최근에는 부호화 라이브러리에서 선택한 랜덤 방식으로 은닉 영상 정보를 복원해서, 올바르게 복원된 영상 정보인지를 판별함으로써, 복호화와 은닉을 함께 판별하는 방법이 연구되었다[11]. 이 방법은 스테고영상의 엔트로피 특성과 픽셀 값의 분산에 대한 상관관계, DCT(Discrete Cosine Transform) 계수의 분포, 영상의 분산 특성과 같은 정보를 학습한 판별 기법을 통해, 영상 스테가노그래피로 부호화된 은닉 정보를 자동으로 검출할 수 있다.

스테가노그래피나 악성코드의 존재를 탐지하기 위해 블라인드 방식 탐지 기술을 사용하여 시각적, 구조적, 통계적으로 분석해 왔다[12]. 시그니처 검색 방법, 파일 레지스트리와 같은 주요 정보를 분석하는 방법, 휴리스틱 방식으로도 악성코드의 존재 여부를 탐지하도록 연구했다[13]. 하지만 이런 연구로는 스테가노그래피를 감지할 수 있는 상용 도구나 서비스로 전통적인 안티바이러스를 사용하기는 어렵고, 기존의 탐지 방식들은 악성코드의 잘못된 탐지나 탐지 못하는 문제점을 갖고 있다[14].

3. 이미지 파일 내 악성코드 무력화

본 논문에서는 이미지 파일 내 악성코드와 은닉 정보를 무력화하기 위한 프로세스로 <그림 2>를 제안하였다. 이미지 파일 추출 단계는 다양한 파일 유입 경로를 통해 수집된 파일들 중에서 파일 확장자나 파일 헤더 정보에서 이미지 파일 식별자를 확인하여 이미지 파일만을 추출한다. 또한, 유입된 문서 포맷 파일로부터 각 문서 포맷의 구조를 파악하여 특정 위치로부터 이미지 객체를 추출한다.

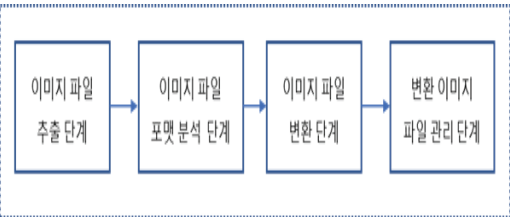

그림 2. 이미지 파일의 악성코드 은닉정보 무력화 프로세스
Figure 2. Malicious Cod Hidden Images Neutralizing Process

두 번째 이미지 파일 포맷 분석 단계에서는, 추출한 이미지 파일의 포맷 구조를 분석하기 위해 파일 헤더 정보로부터 이미지의 종류를 파악하고, 각 이미지 종류에 따른 이미지 파일 포맷 구조 정보를 기준으로 정상적인 구조인지 판별한다. 세 번째 이미지 파일 포맷 변환 단계에서는 이미지 파일의 각 영역별 변환 방법을 적용한다. 마지막 이미지 파일 포맷관리 단계는 동일한 원본 이미지 파일을 반복적으로 변환하는 과정으로 인한 처리 성능의 감소를 해결하기 위해 해쉬값 정보를 주기적으로 갱신한다.

3.1 악성코드 은닉 유형 분석

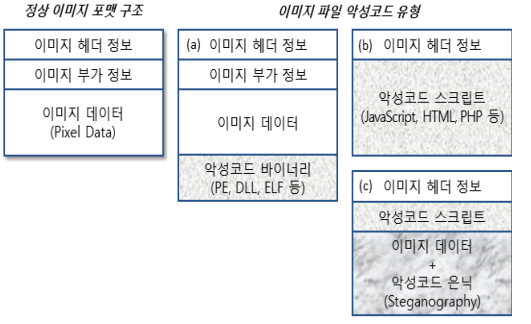

그림 3. 이미지 파일 악성코드 구조유형
Figure 3. Image file malware Structure type

악성코드 은닉 유형 분석에서는 이미지 파일 구조에서 어느 영역에 악성코드가 삽입되는지를 파

악한다. 정상 이미지 파일은 헤더정보, 부가정보, 실제 이미지 데이터 이렇게 세 영역으로 구성되어 있다. 이미지 파일 악성코드 구조유형은 <그림 3>과 같다. 정상적인 이미지 파일 포맷 구조에 악성코드를 은닉하는 방법은 세 영역에 이들 악성코드를 삽입 또는 변조 하는 방식이며 다음과 같이 몇 가지 유형으로 대변할 수 있다.

첫 번째 유형(a)은 이미지 포맷은 정상적으로 구성되어 있지만 이미지 데이터의 끝에 PE, DLL, ELF 등의 악성코드 바이너리나 악성코드 스크립트, 기밀 정보(기업 정보, 개인정보 등) 등을 추가하는 유형이다. 이는 이미지 뷰어 애플리케이션들이 이미지 끝(EOI)까지만 처리하고 그 이후 영역은 무시한다는 것을 악용한 것으로, 만약 악성코드가 DBD(Drive by Download) 유형인 경우에는 이미 다운로드 시점에 이미지 파일로 가장하여 유입될 수 있다.

두 번째 유형(b)은 이미지 헤더 정보에 각 이미지 종류에 대한 파일 식별 시그니처만을 표시하고, 나머지 영역에는 JavaScript, HTML, PHP 등으로 작성된 악성코드 스크립트를 삽입해 놓은 유형이다. 이것은 많은 애플리케이션들이 파일의 헤더 정보만으로 MIME(Multipurpose Internet Mail Extensions) Type을 판단한다는 점을 악용하여, 유입이 허용된 MIME Type의 이미지 파일인 경우에는 이들 악성코드가 탐지되거나 차단되지 않는다.

세 번째 유형(c)은 이미지 파일 포맷의 부가 정보 영역에 악성코드 스크립트를 삽입하고, 이미지 데이터 영역(실제 이미지 픽셀 정보)에도 악성코드를 은닉시키는 형태이다. 악성코드 스크립트에 악성코드 은닉을 위한 다양한 방식의 암호화 및 난독화 알고리즘을 사용하거나 스테가노그래피 알고리즘을 사용할 경우 대단히 어려운 난제로 대두된다. 이런 경우 시그니처 기반 안티 바이러스나 평판 기반 탐지 기법을 우회할 수 있고, 머신 러닝 기반 기법에서도 탐지하고 분석하는게 어렵다. 마찬가지로, 이미지 데이터 영역에서 픽셀정보에 다양한 스테가노그래피 알고리즘이나 도구를 악용하여 기밀 정보를 은닉시켰을 경우 악성코드의 탐지와 분석이 매우 어렵다.

3.2 이미지 파일 악성코드 은닉 무력화 기법

본 논문은 악성코드가 숨겨진 이미지 파일을 구성하는 세 가지 구조 영역을 변환할 때, 비선형 전달함수(Nonlinear Transfer Function)를 사용함으로서, 악성 코드를 제거하면서도 이미지 품질의 손실이 없는 방안을 제시한다.

<그림 4>는 이미지 파일 추출 단계 및 포맷분석 단계를 거쳐 이미지 파일의 각 영역별로 변환하는 방법을 나타낸 것이다. 이미지 헤더 정보 변환 단계(TF1)는 변환 이미지 포맷의 식별 시그니처로 바꾸고, 이미지 부가 정보 변환 단계(TF2)는 특정 스트링 필터링 변환 방법을 적용한다. 그리고 이미지 픽셀 데이터 변환 단계(TF3)는 특정 범위 값의 비선형 전달 함수를 적용하여 원본 이미지의 속성 값을 변환한다.

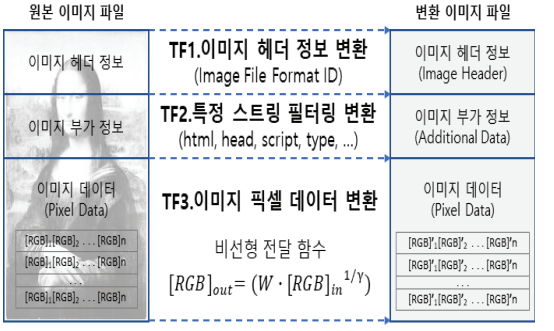

그림 4. 이미지 파일의 각 영역별 변환 방법
Figure 4. To convert by each region of an image file

이미지 파일의 픽셀 데이터 변환 방법에서 사용한 비선형 전달 함수는 아래 식 1과 같이 정의하여 사용한다.

$$[RGB]_{out} = (W \cdot [RGB]_{in}^{1/\gamma}) \quad (1)$$

여기서, γ는 비선형 전달 함수의 특성을 갖게 하는 특정 범위 값을 의미하고, W는 알파 채널의 적용을 의미하며 알파 채널이 적용되어 있는 이미지 포맷인 경우에 한해서만 계산된다.

<그림 5>은 비선형 전달 함수를 적용한 픽셀데이타의 RGB 색상 변화를 예로 보여주고 있는데, 상기 수식 (1)로 정의한 비선형 전달 함수에서 γ의 값을 특정한 범위 값으로 한정함으로써 계산된 변환 값이 각 픽셀의 하위 비트(Leat Significant Bit)가 1~4 비트 이하에서만 변화되도록 제한한다.

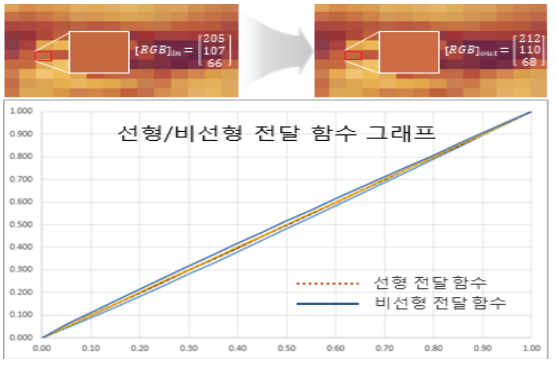

그림 5. 비선형 전달 함수 적용의 이미지 RGB 색상 변화
Figure 5. Image RGB color change of nonlinear transfer function application

4. 실험 및 평가

4.1 실험 개요

본 논문에서는 바이러스토탈(VirusTotal)[15]사로부터 구매한 최신 악성코드(유료API) 48,220개 중 악성코드가 은닉된 이미지 파일 10개를 추출하여 3장에서 제시한 이미지 파일 내 악성코드 은닉 무력화 프로세스 중 이미지 파일 추출단계와 포맷분석단계 이미지 파일 변환단계를 적용한 실험을 통하여 그 효과성을 검증하였다. 바이러스토탈은 구글의 자회사로서 안티바이러스 회사 약70개와 악성코드 정보를 공유하고 있어서, 실험하고 싶은 악성코드를 클라우드 서비스로 접속하여 70개의 안티바이러스 엔진으로 검사해 볼 수 있다.

4.2 실험 결과

8,220개의 악성코드 데이터 중 이미지에 악성코드가 포함된 형태의 파일 10개를 선별하여 바이러스토탈과 국내 5개 백신엔진을 사용하여 탐지여부를 테스트 하였다.

표 1. 악성코드가 은닉된 이미지 파일에 대한 탐지여부
Table 1. Detecion of Hidden image files by Malicious code

종류	분석파일명	파일 크기	VT 탐지	국산 AV
Dropper	63_photograph.octer-stream	1,946	30/59	0/5
Trojan	bgcolor.gif	354	27/59	1/5
	343_s.gif	13,667	34/56	1/5
	694_s.gif	14,163	35/60	1/5
	898_s.gif	10,559	36/59	1/5
	logo.gif	6,546	34/60	1/5
	0f0005AaGAW9UHTlkItEws.gif	20,102	27/59	1/5
	wso.php.gif	66,152	26/60	0/5
Ransomware	242881.png	103,593	13/56	0/5
	bild.png	76,986	19/60	0/5

<표 1>은 악성코드가 은닉된 이미지 파일 10개에 대하여 바이러스토탈 및 국산 안티바이러스 엔진 5개에서 탐지한 결과를 보여 주고 있다. 결과에서 보는 바와 같이 악성코드가 은닉된 이미지 파일 10개에 대해 평균 47.7%의 백신만이 탐지하는

결과를 보였으며, 국산의 경우 평균 12%의 저조한 탐지율을 보였다. 이 결과는 이미지에 포함된 악성코드의 경우 탐지에 어려움을 보여주고 있다.

<그림 6>은 상기 악성코드 이미지 샘플에 대해 무력화 실험을 진행한 결과로, 원본 이미지 파일(GIF)의 악성코드가 변환된 이미지 파일(JPG)에서는 다른 데이터 정보로 변환되면서 악성코드의 특성이 사라졌음을 바이러스토탈을 통해 재확인하였다.

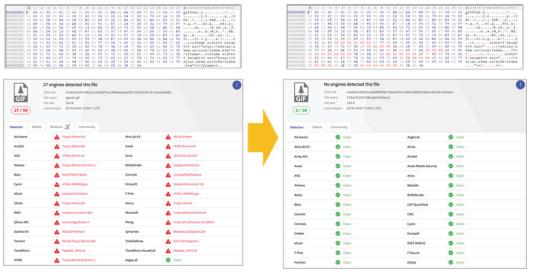

그림 6. 악성코드 이미지 파일의 무력화 실험결과
Figure 6. Result of neutralization of malicious code image file

표 2. 무력화 처리 후, 악성코드 탐지결과 비교
Table 2. Compare Malware Detecion Results

종류	분석파일명	파일 크기	VT 탐지	국산 AV
Dropper	63_photograph.octer-stream	1,946	3/58	0/5
Trojan	bgcolor.gif	354	0/58	0/5
	343_s.gif	13,667	0/58	0/5
	694_s.gif	14,163	0/58	0/5
	898_s.gif	10,559	0/58	0/5
	logo.gif	6,546	0/58	0/5
	0f0005AaGAW9UHTlkItEws.gif	20,102	0/58	0/5
	wso.php.gif	66,152	2/58	0/5
Ransomware	242881.png	103,593	2/57	0/5
	bild.png	76,986	0/58	0/5

<표 2>는 10개의 악성코드 은닉 이미지 파일에 대하여 무력화 방안 수행 후, 바이러스토탈과 국산 안티바이러스 엔진을 활용하여 악성코드 탐지 결과를 나타낸 것이다. 일부 파일을 제외하고는 대부분이 악성코드가 없는 것으로 나타나고 있다.

<표 1>과 <표 2>의 VT(바이러스토탈) 탐지 컬럼 값을 일대일로 비교해보면, 제안한 무력화 기법이 적용된 <표 2>의 VT 탐지값이 원 자료값인 <표 1>의 VT 탐지값보다 현저하게 낮아져 있어서 제안한 무력화 기법의 효과성을 증명할 수 있다. 예를 들어, <표 1>에서 bgcolor.gif 파일의 경우 iframe 유형의 악성코드가 은닉된 파일로서, 이 파일을 바이러스토탈을 이용해 악성파일 유무를 탐지한 결과 59개 탐지엔진 중 27개에서 bgcolor.gif 파일이 악성파일이다는 것을 밝혀낼 수 있었다. 본 논문에서 제안한 무력화 방안을 bgcolor.gif 파일에 적용하여, textString HEX값(http, iframe, htm에 매칭되는 HEX값)을 변경하여 이미지파일의 부가정보 영역에 있는 String 값을 삭제하여 실험한 <표 2>의 결과를 보면, 악성코드 무력화가 적용된 bgcolor.gif 파일이 바이러스토탈을 통한 악성파일 유무 탐지 결과 58개 모든 탐지엔진에서 악성파일이 아니다는 탐지 결과를 제공하였다. <표 1>과 같은 악성코드가 내재된 10개 시험 파일에, 제안한 악성코드 무력화 방안을 적용한 후 악성코드를 탐지한 <표 2>의 결과를 비교할 때, 제안한 악성코드 무력화 방안의 효과를 확인할 수 있다. <표 2>의 일부 파일에서 바이러스토탈에서 사용하는 소수의 안티바이러스 엔진에 의해 악성코드로 인지되는 경우는 특정 스트링으로 인한 잘못된 탐지일 가능성이 높은 것으로 해석할 수 있으며, 악성코드 및 악성코드를 실행하게 하는 부분들이 제거되었기 때문에 실제 정상적인 활동은 못할 것으로 판단된다.

5. 결 론

본 논문에서는 이미지 파일의 구조를 조사보고, 이미지 파일에서 은닉되는 악성코드 은닉유형을

분석한 후, 악성코드를 무력화 시키는 기법을 제안하였다. 기존의 시그니처 기반 안티 바이러스나 평판분석 방법이나, 콘텐츠 무해화 및 재조립 기술과는 다른 접근 방식이다. 원본 이미지 파일에 숨겨진 악성코드를 어떤 분석이나 탐지 기법을 적용하지 않았고, 대신에 이미지 파일 포맷의 구조를 분석하고 픽셀 데이터를 비선형 전달 함수로 변환하여 악성코드 동작을 무력화시킬 수 있었다.

논문의 타당성은 실험을 통해 효율성과 이미지 품질로서 평가하였다. 이 기법은 알려진 또는 알려지지 않은 악성코드의 시그니처나 특징에 대한 어떠한 사전 정보 없이도 악성코드 동작을 사전에 무력화시키는 효과가 있다. 또한, 원본 이미지 파일에 포함된 악성코드를 무력화시킨 변환된 이미지 파일은 육안으로 구분하지 못할 정도의 원본 이미지 품질과 유사한 수준으로 제공해 주는 효과도 있다. 추가로 이 기법은 이미지 파일에 기밀 정보를 은닉시켜 유출하는 보안 위협을 방지하는데도 활용할 수 있다.

향후 연구에서는 악성코드 이미지뿐만 아니라 동영상, 오디오 등 멀티미디어 파일에 은닉되는 악성코드를 효과적으로 무력화시키는 연구를 수행할 필요가 있다.

References

[1] Malware statistics, https://www.av-test.org/en/statistics/malware/html, Feb. 2019.

[2] Financial services commission, *Financial information network separation guide*, Financial supervisory service, 2013. 09.

[3] S. Shah, Stegosploit: *Hacking with pictures*, Hack In The Box Security Conference, May 2015.

[4] D. S. Jung, S. J. Lee, and D. J. Ryu, *A study on the correspondence malicious traffic between the network data transfer systems in a network isolation environment*, Winter Proceedings of Journal of Communications and Networks, pp. 1152-1153, 2016.

[5] I. H. Park, and W. S. Seoung, *A study on concealing malicious code using a image file : method and countermeasure*, Proceedings of the Institute of Electronics Engineers of Korea, pp. 235-236, 2009.

[6] Y. J. Keum, H. J. Choi, and H. K. Kim, *Hiding shellcode in the 24Bit BMP image*, Journal of The Korea Institute of Information Security and Cryptology, Vol. 22, No. 3, pp. 691-705. 2012.

[7] K. Cabaj, L. Caviglione, W. Mazurczyk, S. Wendzel, A. Woodward, and S. Zander, *The new threats of information hiding : the road ahead*, IT Professional, Vol. 20, No. 3, pp. 31-39, 2018.

[8] G. suarez-Tangil, J. E. Tapiador, and P. Peris-Lopez, *Stegomalware: playing hide and seek with malicious components in smartphone Apps*, Lecture Notes in Computer Science, Vol. 8957, pp. 496-515, 2014.

[9] S. D. Cha, K. H. Park, and H. G. Lee, *A solution to the On-Line image downgrading problem*, Journal of The Korea Institute of Information Security and Cryptology, Vol. 6, No. 2, pp. 23-32, 1996.

[10] S. S. Ji, *Locating and searching hidden messages in Stego-Images*, Journal of the Korea industrial information systems society, Vol. 14, No. 3, pp. 37-43, 2009.

[11] J. H. Lee, C. L. Kim, S. H. Lee, and J. I. Park, *Image steganography and its discrimination*, Journal of broadcast engineering, Vol. 23, No. 4, pp. 462-473, 2018.

[12] J. E. Wingate, G. D. Watt, M. Kurtz, C. W.

Davis, and R. Lipscomb, *Defending against insider use of digital steganography*, Proceedings of the Conference on Digital Forensics, Security and Law, pp. 175-184, 2007.

[13] C. T. Do, N. H. Tran, C. Hong, C. A. Kamhoua, K. A. Kwiat, E. Blasch, S. Ren, N. Pissinou, and S. Iyengar, *Game theory for cyber security and privacy*, ACM Computing Surveys, Vol. 50, No. 20, pp. 1-37, 2017.

[14] S. R. Wiseman, *Poison Pixels - combatting image steganography in cybercrime*, RSA Conference 2018, HTA-W02, 2018.

[15] VirusTotal, https://en.wikipedia.org/wiki/VirusTotal, Jan. 2019.

비선형 전달함수 기반의 이미지 포맷 변환을 이용한 악성코드 무력화 기법

정동섭[1], 이상준[2]
[1]*㈜휴네시온 대표*
[2]*전남대학교 경영학부 교수*

요 약

비실행형 파일 중에서 이미지 파일에 악성코드를 은닉시키는 다양한 우회 기법 연구가 진행되어 왔다. 알려지지 않은 악성코드가 은닉되어 있는 경우 평판이나 시그니처 기반의 안티바이러스 방법에 의해 탐지하기 어렵다. 본 논문에서는 악성코드의 시그니처나 특징에 대한 어떠한 사전 정보가 없더라도 원본 이미지 파일 포맷의 구조를 분석하고 이미지 데이터 영역 변환 통해 이미지 파일에 숨겨진 악성코드를 무력화시키는 방안을 제안하였다. 제안한 방법은 이미지 파일 추출 단계, 이미지 파일 포맷 분석 단계, 이미지 파일 변환단계, 변환 이미지 파일 관리 단계로 구성되어 있다. 이미지 파일 변환 단계에서는 원본 이미지의 각 영역별로 헤더 정보 변환, 부가 정보에 대한 특정 스트링 필터링 변환, 비선형 전달함수를 사용한 이미지 픽셀 데이터 변환이 이루어진다. 제안된 방법의 효과성을 증명하기 위하여 바이러스토탈(Virus total)사로부터 구매한 최신 악성코드(유료API) 48,220개 중 악성코드가 은닉된 이미지 파일 10개를 실험에 사용하였다. 무력화 기법을 위한 파일 추출단계와 포맷분석단계, 이미지 파일 변환단계를 적용 후, 실험한 결과는 정량적으로 바이러스 검출량이 현저히 줄어듬을 보임으로서 제안한 방법의 효과성을 검증하였다. 또한, 비선형 전달함수를 이용하여, 변환된 이미지 파일은 육안으로 구분하지 못할 정도의 원본 이미지 같은 품질을 유지하면서도 악성코드를 무력화시킬 수 있었다.

감사의 글

본 논문은 정동섭 전남대학교 석사학위 논문을 보완하여 재구성하였음.

Dong-Seob Jung received the bachelor's degree at KAIST in 1993 and Master's degree in Interdisciplinary Program of Information Security at Chonnam National University. His current research interests include security solutions, network connection systems, and security consulting services.
E-mail address: dsjung@hunesion.com

Sang-Joon Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Statistics from Chonnam National University in 1991, 1993 and 1999, respectively. From 1995 to 2006, he was in Seonam University and Shingyeong University as an assistant professor. Since 2007, He has been with Chonnam National University as a professor in the school of business administration. His current research interests include Management Information Systems, Software Engineering, IT Service, Information Security and Ubiquitous Business. He is a life member of the KKITS.
E-mail address: s-lee@chonnam.ac.kr