Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.3, Jun. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.3.429

Office Open XML 문서 기반 악성코드 분석 및 탐지 방법에 대한 연구

이 덕 규,[1†] 이 상 진[2‡]
[1,2]고려대학교 정보보호대학원 (대학원생, 교수)

A Study of Office Open XML Document-Based Malicious Code Analysis and Detection Methods

Deokkyu Lee,[1†] Sangjin Lee[2‡]
[1,2]Institute of Cyber Security & Privacy (ICSP), Korea University
(Graduate student, Professor)

요 약

최근 침해사고에서 오피스 문서를 통한 공격 비중이 높아지고 있다. 오피스 문서 어플리케이션의 보안이 점차 강화되어왔음에도 불구하고 공격기술의 고도화, 사회공학 기법의 복합적 사용으로 현재도 오피스 문서를 통한 공격이 유효하다. 본 논문에서는 악성 OOXML(Office Open XML) 문서 탐지 방법과 탐지를 위한 프레임워크를 제안한다. 이를 위해 공격에 사용된 악성파일과 정상파일을 악성코드 저장소와 검색엔진에서 수집하였다. 수집한 파일들의 악성코드 유형을 분석하여 문서 내 악성 여부를 판단하는데 유의미한 의심 개체요소 6가지를 구분하였으며, 악성코드 유형별 개체요소 탐지 방법을 제안한다. 또한, 탐지 방법을 바탕으로 OOXML 문서 기반 악성코드 탐지 프레임워크를 구현하여 수집된 파일을 분류한 결과 악성 파일셋 중 98.45%에 대해 탐지함을 확인하였다.

ABSTRACT

The proportion of attacks via office documents is increasing in recent incidents. Although the security of office applications has been strengthened gradually, the attacks through the office documents are still effective due to the sophisticated use of social engineering techniques and advanced attack techniques. In this paper, we propose a method for detecting malicious OOXML(Office Open XML) documents and a framework for detection. To do this, malicious files used in the attack and benign files were collected from the malicious code repository and the search engine. By analyzing the malicious code types of collected files, we identified six "suspicious object" elements that are meaningful in determining whether they are malicious in a document. In addition, we implemented an OOXML document-based malware detection framework based on the detection method to classify the collected files and found that 98.45% of malicious filesets were detected.

Keywords: OOXML, Microsoft Office, documents, malware

I. 서 론

최근 침해사고는 다양한 기법들을 복합적으로 사용하여 쉽게 탐지하기 어려운 상황이다. 일반적으로 알려진 APT(Advanced Persistence Threat) 공격은 정보 수집 후 내부 네트워크로 진입하기 위한

Received(03. 12. 2020), Modified(05. 18. 2020), Accepted(05. 18. 2020)

† 주저자, dk_lee@korea.ac.kr
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

방법 중 하나로 스피어피싱 기법을 사용한다. 주로 메일에 악성 파일을 첨부한 공격이며, Microsoft Office 문서가 큰 비중을 차지한다. 대표적인 형태로서 업무메일을 가장하여 문서파일을 첨부하고, 이 파일을 사용자가 열면, 문서 내 삽입된 코드가 악의적인 행위를 수행하게 된다.

매년 발간하는 보안 업체 카스퍼스키의 리포트 Kaspersky Security Bulletin[1](Fig. 1)에 따르면 Microsoft Office 취약점을 이용한 공격의 비중이 비약적으로 높아져 2018년부터 과반을 나타냈다. 최근 이슈가 되고 있는 랜섬웨어 유포에서도 이메일, SNS 등에 악성 Office 파일을 담아 공격하는 방법이 사용되고 있다[2].

이러한 상황을 인지하여 이전보다 보안이 상당히 강화되었다. 현재는 상위버전의 오피스 문서 어플리케이션에서는 보안경고가 뜨고, 동의하면 Macro가 실행되는 보안정책이 기본으로 설정되어있다. 그럼에도 통계에서 알 수 있듯이 공격자 입장에서 Microsoft Office를 통한 공격이 충분히 유용하다.

문서 파일을 통한 공격은 실행파일 공격에 비해서 정상행위와 비정상행위를 구분하기가 쉽지 않아 행위 탐지에 어려움이 있을 수 있다[6]. 예를 들면 Microsoft Word 형식의 파일(*.docx 또는 *.doc)을 접근할 경우 해당 파일의 연결 프로그램인 winword.exe가 실행되어 프로세스 자체가 신뢰할 수 있는 정상 프로세스로 인식되고 이에 대한 악성행위와 정상행위를 구분하기 어렵다.

OOXML(Office Open XML) 문서는 구조적으로 오피스 내의 기능을 이용한 형태와 파일 내부에 포함 가능한 외부개체를 이용한 형태, 실행 시 동적으로 참조되어 다운로드되는 개체를 이용한 형태가 있다. 본 논문에서는 이에 발생할 수 있는 문제점에 대해서 다루고, 악성 OOXML 문서 파일을 분석하여 공격을 위해 포함된 개체에 따라 사용된 공격 기법을 분류하여 해당 악성 개체를 기반으로 한 탐지 방법 및 탐지 프레임워크를 제안한다.

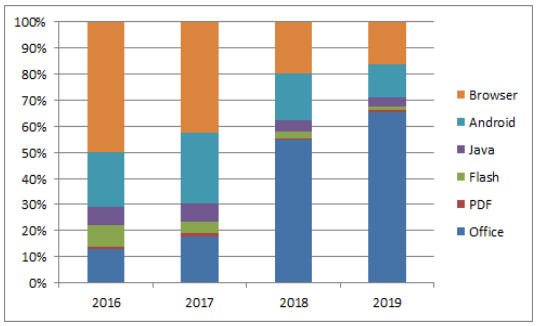

Fig. 1. Distribution of exploits used in cyber attacks, by type of application attacked

II. 관련 연구

문서 기반 악성코드는 해당 문서 포맷에 종속적이다. 이와 관련하여 OOXML 포맷의 Microsoft Office 2007 에서 Data hiding을 위한 unknown parts와 unknown relationship의 탐지기법[3] 및 OOXML 문서 내에서 Ignorable Attribute, Custom XML Data에 대한 탐지기법이 제안되었다[4]. 표준으로 정의하지 않은 unknown 영역에 대해서 OOXML 구조적 관점에서 오피스 문서 어플리케이션의 파싱 에러 등의 노출 없이 데이터를 은닉할 수 있음에 의의가 있다. 다만 찾아진 unknown 데이터가 어떤 성격을 띠는지를 설명하지 않는 데 한계가 있다.

포렌식 관점에서 진행된 연구에서는 오피스 문서의 구조와 오피스 포맷에 대해 forensic method로서 정보 은닉을 위해 중복되는 파일, 압축 알고리즘, Revision Identifier, Macro, XML 코멘트 등이며, 수사관점에서 코멘트등 임의로 숨겨놓은 메시지 등이 존재할 때 이를 발견하기 위한 방법이 제안되었다[5].

Compound File Binary(CFB) 포맷의 Microsoft Office 문서에서 파일 내 비정상 요소를 탐지하기 위한 연구[6]가 있었으며, 비정상을 판단하기 위해 Macro, Data Record Area, OLE, Unused Area, Stream Storage 등으로 영역을 나누어 탐지방법을 제안했다. 이 연구에서는 공격자가 해당 영역에 데이터 삽입 시 어떤 비정상 행위가 일어날 수 있는지 명세하고 비정상 요소 탐지 가능성을 찾는 것에 의의가 있다.

이 밖에도 PDF 문서에 관한 연구[7]는 Key 요소가 되는 메타데이터를 추출하여 Reference Chain, Hex Digit, 압축데이터 등 비정상 요소를 식별하고 악성코드를 탐지하기 위한 프레임워크를 제안하였다. OOXML 기반의 오피스 문서와 유사하게 구조적 문서 포맷을 가진 PDF에 대해서 구조적 특성 정보를 추출하고 비정상 요소를 탐지하는 데 의의가 있다.

또한, 악성 오피스 문서 탐지를 머신러닝에 적용한

연구[8]에서는 Microsoft Office Word에 대해서 OOXML 구조에서 추출한 메타데이터를 탐지하는 프레임워크를 제안하였다. 이 연구에서 오피스 문서의 구조적 특성 정보가 악성 여부를 결정하는 데 효과적이며, 머신러닝이 악성 오피스 문서 파일 탐지에 적용될 수 있음을 나타내었다.

III. Office Open XML 파일의 구조

3.1 Office Open XML 파일 포맷

Office Open XML은 XML을 기반으로 한 파일 포맷으로서, Microsoft가 추진하여 ECMA-376[9], ISO/IEC 29500[10]으로 승인된 표준 포맷이다. 이 문서 규격은 Word processing documents, Presentation, Spreadsheet를 포함하고 있다. 확장자는 기존의 확장자에서 "x"의 postfix가 추가된 형태로 .docx, pptx, xlsx으로 통일되어있다. 포맷 자체의 특성으로 PKZIP 파일 포맷으로 압축하여 패키징한다.

이 외에 속성에 따른 확장자 포맷은 다음과 같다.
- Word: .docx, .docm, .dotx, .dotm
- Excel: .xlsx, .xlsm, .xltx, .xltm, .xlsb, .xlam
- PowerPoint: .pptx, .pptm, .ppsx, .ppsm

OOXML 문서는 Open Packaging Convention (OPC) 패키지 안에 저장되어 있으며, 이는 PKZIP 압축 포맷 내에 XML, 다른 데이터 파일과 그와 관련된 파일들로 이루어져 있다. 이는 편집 시점에 자동으로 unzip 되어 내부에 있는 파일들을 참조하게 된다.

3.2 일반적인 OOXML 파일의 구성 요소

탐지 방법에 대한 이해를 위해서 OOXML 포맷 구조에 대해 개략적으로 언급하고자 한다. 어플리케이션 별로 다양한 디렉터리 구조를 가질 수 있으며, OOXML 포맷에 필수적으로 포함되어야 하는 개체와 선택적으로 포함되면서 악용 가능성이 있는 개체에 대해서 오피스 문서 포맷 아래 상대 경로를 기준

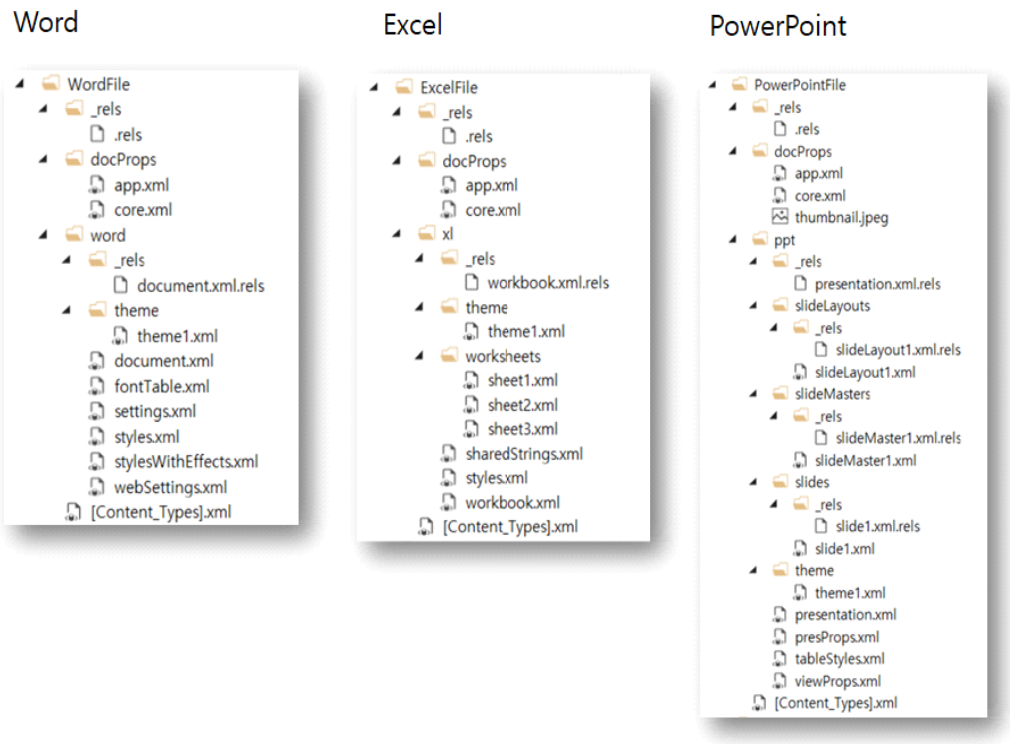

Fig. 2. Common format of OOXML Microsoft Office

으로 설명한다.

/[Content_Types].xml OOXML 포맷 파일에 필수적으로 포함되는 이 파일은 특정 파일 확장명에 대한 기본값과 IRI에서 지정한 부분에 대한 재정의를 사용하여 패키지 일부에 MIME 형식 정보를 제공한다.

/_rels 모든 패키지는 다른파트들과 패키지 밖의 리소스들 사이에 관계를 정의하는 관계파트를 포함한다. 이 디렉터리에서 패키지 내의 파일에 대한 관계를 명시하여 나타낸다. 특정 파일에 대한 관계를 찾으려면 파일에 연관되는 _rels 디렉터리를 찾은 다음 원본 파일 이름에 .rels가 추가된 파일을 찾는다. 예를 들어 document.xml에 참조 관계가 있으면 _rels 디렉터리 내에 document.xml.rels라는 파일이 존재하게 된다. 참조 형태는 암시적 참조와 명시적 참조 형태가 있다. *.xml.rels에는 참조관계가 표현되어있으며, "Relationship" 속성으로 이를 나타낸다. "TargetMode" 속성값이 "Internal"인 경우 문서 파일 내에서 참조하고, "External"인 경우 문서 밖에서 대상을 참조하게 된다. 문서 외부에서 참조하는 경우를 "External-Ref" 유형(Table 3)으로 분류하고 뒤에서 다루도록 한다.

/_rels/.rels 이 파일은 패키지 관계를 나타내는 필수적인 부분이다. 오피스 문서 구조에서 각 섹션들의 관계를 간략하게 나타낸다.

/docProps/core.xml 이 파일에는 Office Open XML 문서의 메타데이터 등 핵심 속성이 포함되어 있다.

/[word|ppt|xl]/document.xml 이 파일은 전체 문서의 주요 부분으로서 문서의 본문을 구성하는 텍스트와 이를 오피스 문서 내에서 표현하기 위한 각 본문 속성에 맞는 XML과 Object 참조에 대한 내용을 담고 있다.

/[word|ppt|xl]/[header[digits]|footer[digits]].xml 머리말, 꼬리말을 구성하는 텍스트와 이에 대한 참조속성이 있으며, 텍스트 표현 및 개체 참조에 대한 XML 구조가 본문과 동일하다.

/[word|xl|ppt]/vbaProject.bin Macro가 존재하는 경우에 이 개체가 포함된다. Macro는 내장된 프로그래밍언어인 VBA(Visual Basic for Applications)로 작성되어 "vbaProject.bin" 개체로서 문서에 포함된다. Macro로 작성된 코드는 오피스 문서 어플리케이션에서 정상적으로 지원하는 기능으로서 시스템 및 파일 접근 권한을 가지므로, Macro 작성자의 의도에 따라 악용 가능성이 있다[11][12].

/[word|xl|ppt]/activeX/activeX[digits].bin ActiveX 기능을 사용하여 바이너리를 오피스 문서 내에서 참조 실행하기 위해 포함된다. ActiveX 형태로 참조된 바이너리를 삽입하여 공격 벡터로 사용할 수 있다.

/[word|xl|ppt]/embeddings/oleObject[digits].bin 문서 내에서 데이터 또는 다른 포맷의 파일을 삽입하여 참조하기 위해 포함된다. 이미지, 미디어 포맷과 다양한 형식의 raw binary 파일을 참조할 수 있다. 바이너리를 삽입하여 공격 벡터로 사용할 수 있다.

IV. 악성 파일 유형별 분류

4.1 데이터셋

실제로 위협이 되고 있는 OOXML 포맷 문서 악성코드를 분석하고 나아가 탐지 프레임워크를 제안하고 평가하기 위해 데이터셋을 생성하였다.

Anti-Virus 엔진을 이용한 악성코드 탐지 서비스를 제공하는 VirustTotal(virustotal.com) 악성코드 저장소의 오피스 문서 중에서 OOXML 포맷 문서 3,780개의 파일을 수집하였고, Google 검색엔진에서 OOXML 포맷의 문서를 임의로 검색하여 600개의 정상 파일을 수집하였다(Table 1).

수집된 파일들에 대해서 분석 및 검증, 평가를 위해서 각 파일에 대한 Label(Malicious, Benign) Data를 생성하였다. 이를 위해서 파일에 대한 Anti-Virus 제품의 탐지 결과와 VirusTotal 및 분석가의 정밀 분석을 통해 데이터를 구성하였다.

제품을 통한 탐지 데이터 검증에는 Anti-Virus 평가기관 "AVTEST"에 등록된 Major Vendor 중에서 임의로 선정한 Ahnlab, Kaspersky, Avast 등 9개 사의 탐지 정보를 참고하였다. 참고한 제품

Table 1. The Source of Our Data-Set

Data Source	Files
VirusTotal Repository	3,780
Google Search Engine	600
Total	4,380

의 탐지 결과 및 VirusTotal 결과의 에러를 줄이고 좀 더 명확한 Label Data를 만들기 위해서 9개 Anti-Virus 제품 탐지 결과 중 2개 이상의 제품에서 탐지하는 파일을 Malicious로 분류하고, 제품에서 탐지하지 않는 파일을 Benign으로 분류하였으며, 나머지는 오탐의 여지가 있어 데이터셋에서 제외하였다.

이후 모호한 부분에 대해서 실제 의심 개체가 존재하는지 여부 확인 및 CVE 등 분석자료를 바탕으로 파일을 분석, 검증하였다. 검증 결과 4,380개 파일 중 Malicious 파일 2,066개와 Benign 파일 2,184개로 구성하였으며, 손상된 파일 등을 포함하여 130개 파일은 제외하였다(Table 2). 본 논문에서는 Malicious 파일셋에 초점을 맞추어 분석하고, Benign 파일셋은 탐지 평가를 위해서 사용한다.

Table 2. Class of Data-Set

Class of Data-Set	Count
Malicious	2,066
Benign	2,184
ETC (damaged, FP, ...)	130

4.2 악성 개체요소

데이터셋의 악성코드 분류셋에 대해서 파일 분석을 수행하였으며, 공격자가 악성 행위를 수행하기 위해 문서 악성코드 생성 시 OOXML 포맷 문서 파일 내에서 의도적으로 포함하거나 기능을 위해 자동으로 포함되는 의심 개체요소를 구분하였다. 악성코드 여부를 판단하는데 유의미하다고 판단되는 개체의 요소별로 전체 데이터셋을 분류(Table 3)하였으며 해당 개체요소에 대한 설명과 데이터셋에 포함되는 개수, 비율을 나타내었다.

Table 4. Vulnerability Files Classification

CVE number	word	excel	ppt	total
CVE-2017-11882	6	212	0	218
CVE-2017-0199	183	0	6	189
CVE-2016-7262	0	33	0	33
CVE-2017-8570	0	9	8	17
CVE-2017-8759	1	1	10	12
CVE-2018-0802	1	4	0	5
CVE-2014-6352	0	0	2	2
CVE-2018-4878	1	2	0	3
CVE-2012-1856	1	0	0	1
CVE-2018-8414	1	0	0	1
CVE-2015-2545	1	0	0	1
CVE-2018-15982	1	0	0	1

분류 결과, 한 파일에서 여러 개체요소를 포함하거나, 하나도 포함하지 않는 경우도 존재함을 확인하였다. 전체 악성 데이터셋에서 적어도 하나 이상의 의심 개체요소를 포함하는 파일의 전체대비 비율은 99.66%로 탐지대상 파일에서 하나 이상의 개체요소를 가지고 있음을 알 수 있다.

분류된 악성 파일 중 취약점을 활용한 악성코드의 비중과 유형의 파악을 위해 OOXML 포맷의 문서 파일들에 대한 VirusTotal 및 Anti-Virus 제품들의 탐지결과를 구했으며, 탐지명에 CVE(Common Vulnerabilities and Exposure) 식별자가 부여된 것을 바탕으로 분류하였다. 실제 공격에 사용되는 취약점은 Table 4와 같이 한정적임을 알 수 있다.

Table 3. Classification of Office Files by Malicious Object Type

Element	Description	Malicious Set		Benign Set	
		Files	Per.	Files	Per.
Macro	A "VbaProject.bin" exists in a document	873	42.26%	84	3.85%
OLE Object	"oleObject*.bin" exist in a document	541	26.19%	70	3.21%
DDE	DDE keywords exist in XML files	104	5.03%	2	0.09%
ActiveX	"activeX*.bin" exist in a document	30	1.45%	18	0.82%
EPS	EPS objects exist in a document	14	0.68%	1	0.05%
External-Ref.	An external reference object exists in "relationships" properties	706	34.17%	306	14.01%

이 중 본 논문에서 제안하는 개체요소의 구분을 바탕으로 탐지모델을 구현하여 해당 취약점 유형들을 탐지할 수 있도록 고려하였다.

4.3 악성코드 유형별 분석 및 탐지 방법

악성 개체요소 분석을 통해 문서 내 악의적인 의도로 사용 가능한 개체에 대한 탐지영역을 식별 및 분류하였다. 이에 대한 유형별 분석 및 탐지 방법을 제시한다.

Macro(VBA). VBA 스크립트 언어로 문서 내 "vbaProject.bin" 개체에 포함되어 있는 형태이다. 오피스 문서 프로그램 내에서뿐만 아니라 시스템을 제어하는 행위를 수행할 수 있다. 일반적인 프로그래밍 언어의 속성을 갖고 있어 언어의 자유도에 따라 악성코드 스크립트의 형태가 다양하고, 스크립트 형태의 악성코드는 보안도구에 탐지되지 않기 위해 대부분 난독화 되어 있는 상태여서 탐지 패턴을 찾기가 쉽지 않다. 이에 Macro에서 자동실행 전략을 수행하기 위해서 "DocumentOpen", "Auto_Open" 등의 자동실행 함수 키워드[13]와 사용자의 행위를 유도하여 특정 ActiveX Control 실행 시 발동되는 기법[14]들을 탐지전략으로 사용하였다. 이와 함께 VBA digital signature가 없는 파일 중 자동실행을 위해서 해당 함수 키워드가 삽입된 파일을 탐지되도록 조건을 설정하였다.

또한, 시스템 제어기능을 활용 할 수 있는 쉘명령과 네트워크 소켓을 사용하는 경우 탐지대상으로 설정하였다.

OLE Object. 다양한 포맷의 데이터를 오피스 문서 어플리케이션에서 지원하기 위해 OLE 개체로 변환되어 문서 내 개체로 삽입된 "oleObject*.bin"에 대해서 삽입된 데이터가 바이너리 포맷의 형식이 자유로워 공격자가 보안 기능을 우회하거나 탐지를 회피하여 코드를 실행시키는 데 유리하게 활용할 수 있다.

OLE Object 내 취약점을 유발하는 프로그램 실행 또는 라이브러리를 로드하는 바이너리 코드를 포함하고 있다. 오피스 문서 프로그램 자체로는 DEP, ASLR 등 보안이 강화되고 기존 취약점이 보완되면서 이보다는 좀 더 취약한 라이브러리를 로드하거나 타 프로세스를 실행하는 방식을 사용한다.

CVE-2017-11882, CVE-2018-0802 취약점(Fig. 3)을 사용한 악성코드는 취약점을 유발하는

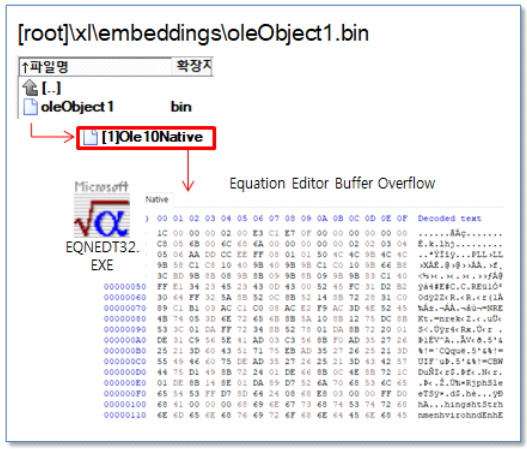

Fig. 3. OLE Object with Equation Editor

수식을 문서 내에 삽입하여 문서 실행 시 수식을 표현하기 위해 취약한 버전의 수식편집기가 실행되고 수식편집기 코드가 Exploit을 발생시켜 공격자의 시스템 제어행위를 수행하게 된다. 현재 수식편집기는 2018년 1월 Microsoft 업데이트에서 제외되었으며, 이와 같은 악성코드와 유사한 방법으로 oleObject 내에서 악의적인 의도로 Equation Editor 개체를 참조하는 Object가 존재하는 것을 악성 개체로 판단하고 탐지하였다.

CVE-2014-6352 취약점을 사용하는 악성코드는 취약한 버전의 packager.dll를 통해 파일의 OLE 개체 내 명시되어있는 INF 파일을 사용자의 인지 없이 원격에서 다운로드하고 실행한다. 이를 탐지하기 위해서 PPT 형태의 오피스 파일이 취약한 버전의 packager.dll 을 악성코드와 유사한 방식으로 로드하고 특정 커맨드를 실행하려고 할 때 악성 개체로 보고 탐지되도록 하였다.

CVE-2018-4878 취약점을 사용하는 악성코드의 경우 내부에 Adobe Flash 파일을 포함하여 실행 시 Flash 취약점을 발생시키도록 하며, CVE-2018-4878과 유사한 방식으로 flash 취약점을 유도하는 유형이 탐지되도록 하였다.

CVE-2018-8414 취약점을 사용하는 악성코드의 경우 Settingcontent-MS(윈도우 설정 바로가기 파일)에서 경로값 검증이 미흡하여 발생하는 원격코드를 실행할 수 있으며, oleObject*.bin 에서 추출한 데이터 안에서 해당 취약점을 사용하기 위해 CLSID를 참조하는 경우에 탐지되도록 하였다.

OLE Object 개체 내부에 사용자의 편의를 위해

서 다양한 포맷의 문서 또는 스크립트, 바이너리들이 포함될 수 있다. 악의적인 목적으로 개체가 포함된 문서가 실행되면, 문서 편집 시점에 본래 포맷으로 출력된 쉘스크립트, 실행 파일 등이 시스템 제어 등 공격자가 의도한 동작을 수행하게 된다.

문서 내에 JAR 파일을 포함하고 있는 유형(Fig. 4)의 경우 사용자가 문서 파일 실행 시 문서 내부에 있는 악성코드가 파일 형태로 시스템에 저장 및 Java 백도어가 실행되도록 한다. 해당 유형은 공격자가 탐지 회피를 위해 JAR 파일을 오피스 문서 파일의 OLE Object 내부에 포함시켜 은폐하였다. 문서 내 아이콘 형태의 파일로 보이며 사용자가 더블클릭하여 실행 시 발동하여 해당 파일이 실행된다. Object 내 JAR 파일을 활용하는 것은 Anti-Virus를 우회하기 위해 자주 쓰이는 방법이며, 이 자체를 악성 개체로 본다.

공격자의 코드를 실행하는 다른 형태로서 oleObject*.bin 내에 쉘스크립트가 포함된 유형이 있다. 그 중 VBS를 포함하고 있는 유형(Fig. 5)의 경우 실행되면, 일반적으로 공격자의 추가적인 기능 수행을 위해서 유포지로부터 2차 악성 파일 다운로드를 시도한다. 같은 이유에서 악성 위험성을 고려하여 Object 내 포함된 쉘스크립트 유형을 악성 의심 개체로 탐지되도록 하였다.

이를 포함하여 공격자가 임의로 은폐하기 위해 포함시킨 것으로 판단되는 유형의 파일이 포함된 경우

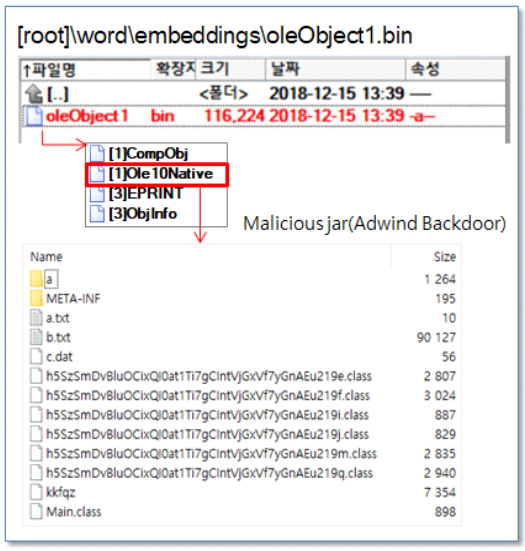

Fig. 4. OLE Object with JAR

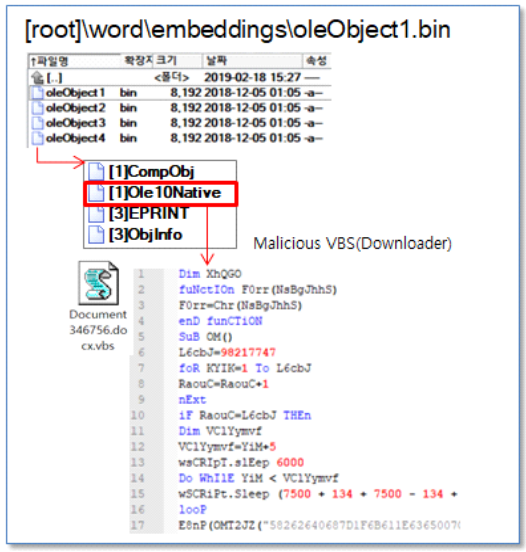

Fig. 5. OLE Object with VBS

탐지하였으며, 대상이 되는 파일 유형의 확장자는 exe, scr, com, pif, jar, vbs, vbe, js, jse, lnk, swf, rar, 7z, bat, cmd 이다.

DDE(Dynamic Data Exchange). 일반적으로 "DDE" 또는 "DDEAUTO" 키워드를 사용하여 해당 기능을 호출한다. DDE 기능은 원격지에 있는 제3자에게 정보가 공유되도록 하거나 프로그램 간 편의성 있는 통신을 목적으로 한다. 이를 통해 공격자는 임의의 코드 실행 또는 외부 개체를 참조하여 코드 또는 바이너리를 실행할 수 있다. 본래 "DDEAUTO" 키워드를 쓰면 자동으로 DDE 기능이 활성화되었으나 17년 하반기 Microsoft 패치 이후에 "DDEAUTO" 키워드를 쓰더라도 자동으로 로드되지 않도록 수정되었으며, 동적 로딩 시 오피스 문서 어플리케이션 기본 보안 설정에 따라 경고 문구(Fig. 6)가 뜨게 되며, 사용자가 "예"를 누를 경우 동적으로 개체가 다운로드되어 참조된다.

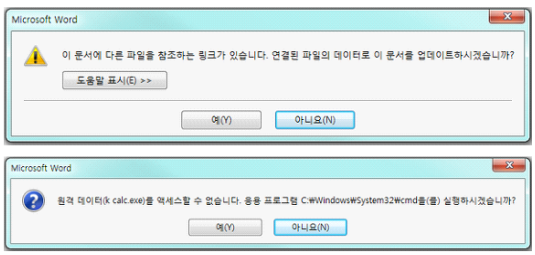

Fig. 6. External referral warning Messages

DDE 키워드와 명령 구문(Fig. 7)을 통해서 간단한 형태로도 시스템을 제어하는 명령을 수행하게 할 수 있으며, 수식 편집기 등 지원하는 객체 컨트롤 내에서 이와 관련한 기능을 활성화 시켜 호출할 수 있다.

```
DDEAUTO c:\\windows\\system32\\cmd.exe "/k calc.exe"
```

Fig. 7. DDE edit selected formula example

일반적으로 해당 DDE 명령코드를 본문에 삽입하며, 본문 내 DDE 기능을 삽입한 악성코드 유형의 경우 DDEAUTO 명령코드가 "document.xml" 파일 내에 해당 키워드 문장의 형태(Fig. 8)로 존재한다. 본문 이외에도, 머리말과 꼬리말에 삽입할 수 있으며, 이 경우 "header*.xml", "footer*.xml" 에 DDE 명령코드가 존재한다.

탐지 관점에서는 해당 XML 구조 안에서 악성코드가 사용하는 DDE 명령코드가 삽입된 경우를 조건으로 하였다.

[root]\word\document.xml

```
<?xml ...(생략)...
<w:instrText>DDEAUTO
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
 "-w hidden -noexit
[Reflection.Assembly]::Load([Convert]::FromBase64String((
New-Object Net.WebClient).DownloadString('
https://pastebin.com/raw/xWM3HuvP
')))).EntryPoint.Invoke($Null,$Null)"</w:instrText>
...(생략)...
```

Fig. 8. DDEAUTO keyword code

ddeLink Class[15]는 'externalLink*.xml' 안에 존재하게 되며, XML 개체 내 ddeService, ddeLink, ddeItem, 속성값이 존재한다(Fig. 9). 이를 통해 호출하고자 하는 어플리케이션과 파라미터로 실행될 기능을 명세한다.

[root]\xl\externalLinks\externalLink1.xml

```
<?xml...(생략)...
<ddeLink xmlns:r="
http://schemas.openxmlformats.org/officeDocument/2006/rel
ationships" ddeService="MSEXCEL" ddeTopic=
"\..\..\..\Windows\System32\cmd.exe /c powershell.exe
-nop -w ...(생략)...
```

Fig. 9. DDE Service with powershell

CVE-2016-7262 취약점을 사용하는 악성코드는 Excel에서 사용자가 특정 셀을 클릭했을 시 링크를 업데이트하는 과정에서 보안 기능을 우회하고 취약점을 발생시키며 이때 DDELink 기능을 사용하여 임의의 명령코드를 동작시키게 된다.

이 기능을 호출하기 위해서는 반드시 DDE 키워드가 있어야 하므로, XML 내에서 의심 개체가 특정 위치에서 악성코드가 사용하는 기법으로 DDE 기능을 사용하면 악성으로 탐지한다.

ActiveX. 오피스 문서 어플리케이션의 ActiveX Control 기능을 위해 문서 내 삽입된 개체 "activeX*.bin"(Fig. 10)을 통해 악성 Macro 등의 공격자 코드가 실행되거나, 취약점을 유발하는 adobe flash 파일을 삽입하여 ActiveX object 바이너리 구문 분석 시 오피스 문서 어플리케이션에서 예외처리가 되지 않는 문제를 이용하여 임의코드 실행 취약점을 발생시켜 의도된 공격을 수행할 수 있다.

[root]\xl\activeX\activeX1.bin

파일명	확장자	↓크기	날짜	속성
[..]		<폴더>	2019-02-18 18:53	——
[_rels]		<폴더>	2018-12-04 17:38	——
activeX1	bin	117,371	2018-12-05 01:05	-a--
activeX1	xml	302	2018-12-05 01:05	-a--

Fig. 10. ActiveX Control Object

악성코드가 ActiveX object 내부에 flash 파일로 숨겨진 경우도 탐지될 수 있도록 하였다. ActiveX 바이너리 파싱 시 발생하는 오피스 문서 어플리케이션 취약점에 대해서는 heap spray를 위해 바이너리가 비정상적으로 많거나 큰 형태를 탐지할 수 있도록 하였다.

EPS(Encapsulated PostScript). 그래픽 이미지를 표현하는 목적의 Adobe 스크립트 언어기반의 EPS는 오피스 문서 어플리케이션에서 지원한다.

문서 내 "*.eps" 개체 파일로 존재하며, 주로 "image*.eps"의 이름을 갖는다. 외부 모듈을 통해 실행기능을 호출하므로, 해당 모듈의 취약점에 영향을 받는다. EPS는 공격자들이 악용하고 있어서 현재는 최신 버전의 오피스 문서 어플리케이션에서 지원되지 않는다. 그만큼 많은 공격자들이 EPS 취약점을 활용하여 공격에 사용하였으며, 문서 파일이 EPS를 포함하고 있는 경우 악성코드로 의심할 수 있다.

CVE-2015-2545 취약점을 사용하는 악성코드는

문서 파일을 열 경우 EPS 스크립트에 작성된 내용으로 인해 EPS 파일을 처리하는 모듈인 "EPSIMP32.FLT"에서 UAF(Use After Free) 등의 취약점을 발생시킨다. 개체 내 삽입된 스크립트는 가독성 있는 문자열의 형태로 패턴이 일정하므로 이 개체에 정의된 특정 행위에 대한 코드를 탐지하기 위해 해당 기능을 호출하는 형태를 탐지하면 된다. 언어의 자유도로 인해서 특정 행위가 바이너리 문자열 형태(Fig. 11)로 표현되거나 난독화 된 유형(Fig. 12)의 경우 이를 탐지하는 것이 어렵다. 탐지 관점에서는 취약점을 유발하기 위한 쉘코드 바이너리 형태의 패턴 또는 연속되는 바이너리 패턴이 비정상적으로 큰 형태를 탐지하도록 하였다.

External-Ref. 문서 내에서 "*.xml.rels" XML 개체 내 Relationship 속성으로 Object 참조 시 문서에 포함된 Object뿐만 아니라 URI 주소를 지정함으로써 파일 외부에 있는 Object도 참조할 수 있다. 이 외부 개체 참조 기능을 통해서 시스템 외부 개체를 실행 시점에 다운로드 하여 원래 파일 안에 포함된 것처럼 참조할 수 있다. 참조되는 개체가 어떤 것인지에 따라 이를 이용한 취약점 등 다양한 방법으로 공격을 수행할 수 있다.

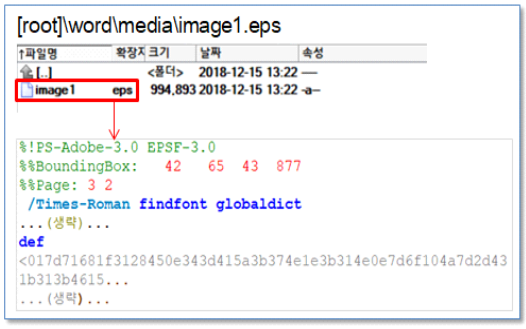

Fig. 11. EPS binary included script

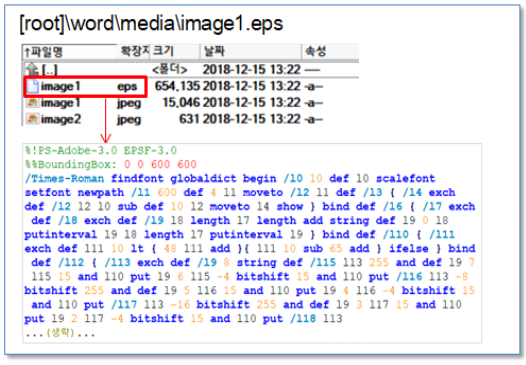

Fig. 12. EPS obfuscated script

공격자가 악성 행위를 위해서 External-Ref 코드를 삽입한 경우 참조되는 유형의 속성값이 "oleObject", "attachedTemplate", "frame", "externalLinkPath", "hyperlink"로 한정적인 것으로 확인하였으며, 속성값 중 하이퍼링크는 정상 사용 여지가 많아 False Positive가 예상되는 "externalLinkPath", "hyperlink"를 제외하고 탐지되도록 하였다.

이와 함께 악성코드를 다운받아 실행하는 경우가 일반적이어서 추가적인 악성 문서 또는 실행 파일을 다운받는 형태로 나타나며, Word 확장자와 실행 파일 확장자의 파일들이 있는 경우 탐지되도록 하였다.

CVE-2017-0199 취약점을 사용하는 악성코드는 URL Moniker와 OLE를 이용한 HTA(HTML Application) 파일을 삽입하고 로드한다. 이를 탐지하기 위해 악성코드와 같은 방식으로 외부 HTA 파일을 참조하는 경우와 취약점에서 명세하는 OLE Link 방식으로 외부참조하는 형태를 범용적으로 탐지할 수 있도록 하였다.

CVE-2017-8570과 CVE-2017-8759 취약점을 사용하는 악성코드도 그 동작 형태가 유사하다. CVE-2017-8570을 사용하는 악성코드(Fig. 13)는 Script Moniker 개체에 대해 초기화를 하는 과정에서 발생하는 취약점을 활용하기 때문에 외부에서 sct script(Windows Script Component)를 참조하며, 이를 탐지할 수 있도록 한다.

CVE-2017-8759의 경우 SOAP WSDLParser 코드 취약점을 사용했으며, SOAP Moniker에 전달된 코드에서 CRLF 유효성을 검사하지 않아 발생하는 취약점이다. 이와 같은 형태로 악성코드에서 soap:WSDL을 통해 XML 문서를 참조하려고 할 때 탐지하도록 하였다.

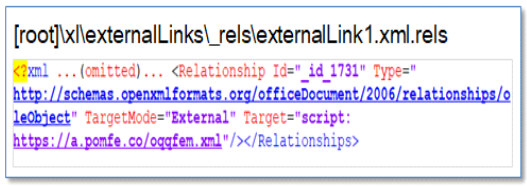

Fig. 13. External OLE Object

V. 구현 및 성능 평가

5.1 탐지 도구 구현

본 논문에서 제시한 OOXML 포맷의 문서 악성코드 탐지기법을 바탕으로 탐지 도구를 개발하였다. 추출된 메타데이터로부터 6가지 개체유형으로 분류하였으며 각각 Macro, OLE Object, ActiveX, DDE, EPS, External-Ref이다. OOXML 포맷은 PKZIP 형태로 패키징 되어있으므로, 압축해제하여 메타데이터를 추출하고 OOXML 포맷 여부를 체크하여 대상 여부를 판단한다. 그다음 개체와 XML 파일들에 대해 오피스 문서 유형(Word, Excel, PPT)에 따라서 분류한다. 추출한 메타데이터들은 각 탐지 모듈로 보내어 검사(Fig. 14)를 수행한다. 개체별 탐지 모듈에서는 정의한 악성코드에서 사용하는 개체요소 존재 여부를 판단한다. 이때 의심 개체

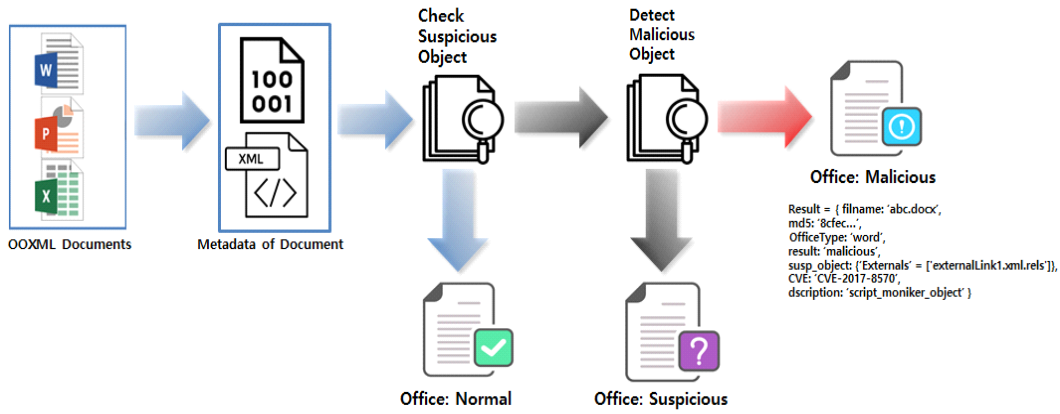

Fig. 14. Model of OOXML document detection framework

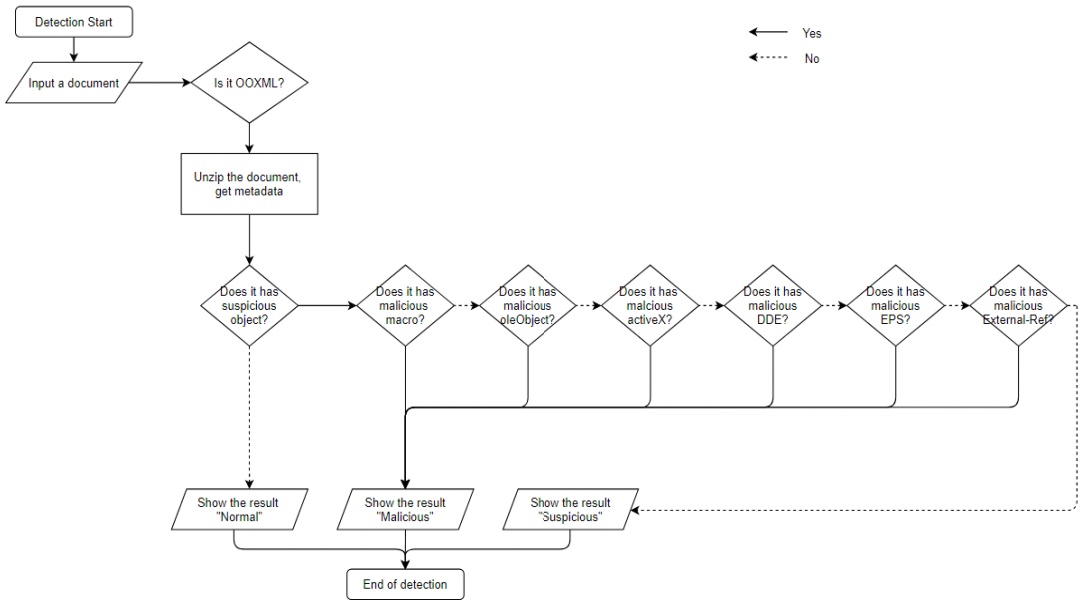

Fig. 15. Detection Process of Malicious OOXML Document

가 존재하지 않는 경우 더 이상의 진행 없이 해당 파일의 결과를 Normal로 분류한다. 개체요소가 존재하는 경우 해당 개체요소에 해당하는 각 악성 탐지 모듈로 해당 정보를 전달한다. 탐지 모듈에서는 개체요소별 악성코드 특성 및 취약점을 바탕으로 구현된 탐지 프로세스(Fig. 15)를 통해서 악성 여부를 판단한다. 취약점 사용 문서 파일에 대해서는 사용자가 탐지 결과에서 CVE number 값을 출력하여 어떤 취약점을 사용한 악성코드인지 알 수 있게 하였다. 의심 개체 검사 순서는 악성 파일에서 사용되는 빈도와 그 의심 개체가 시스템에 미치는 영향을 고려하여 정하였다. 검사에 탐지될 경우 Malicious로 분류한다. 검사에서 탐지되지는 않았지만, 개체가 존재하여 악성 가능성이 있는 파일은 Suspicious로 분류하여, 사용자가 악성코드의 가능성이 있음을 인지하게 하고, 수동으로 확인할 수 있도록 구분한다.

일부 악성코드의 경우에는 특정 오피스 문서에서 거의 사용되지 않는 기능이 악성코드에서 악용되어 유포된 경우가 있고, 특히 취약점 유형별 오피스 문서 분포가 특정 유형으로 치우치고 있으므로 이를 종류로 구분하는 것이 유의미하며, 오피스 문서 유형(word, excel, ppt)을 구분하여 탐지함으로써 False Positive의 가능성을 줄였다.

이어서, 악성/의심/정상 여부를 바탕으로 탐지유형과 탐지된 개체, 오피스유형 등 관련정보를 출력 및 저장한다.

구현된 악성 OOXML 문서 탐지 프레임워크는 https://github.com/sharpduckk/OOXML-MalClassifier 에서 확인할 수 있다.

5.2 성능 평가

본 논문에서 제시한 탐지기법을 적용한 탐지 도구 프레임워크에 대한 성능 평가를 수행하기 위해 상용 Anti-Virus 4개 제품군과 비교하였다. 총 4,250개 파일(Malicious 2,066개, Benign 2,184개)에 대해서 탐지성능을 측정하였다. 탐지기법을 적용한 탐지 도구와 Anti-Virus 제품에 대한 성능 비교 결과는 Table 5와 같다.

본 논문에서 제시한 탐지기법을 통해 탐지성능을 측정한 결과 악성 데이터셋에 대해서 98.45% 탐지(True Positive) 성능을 보였다. 한편 정상 데이터셋에 대해서는 1.19%의 오탐(False Positive) 결과를 보여 어느 정도 오류의 여지가 있음이 확인되었다. Anti-Virus 제품들의 탐지명을 기준으로 분류한 취약점 사용 악성 문서 데이터셋에 대해서는 100% 탐지(True Positive)를 확인하였다(Table 6). 대부분의 취약점 문서 파일에 대해서 CVE number를 분류하였으며, 일부 파일은 탐지는 되었으나 CVE number가 Label Data와 상이하였다. 이와 관련하여 CVE-2018-0802는 CVE-2017-11882와 유사한 취약점 유형으로 동일한 특징을 탐지대상으로 하였기 때문에 모두 CVE-2017-11882로 분류되었다. CVE-2018-4878는 악성 Adobe Flash 파일이 "oleObject*.bin" 개체 내 포함된 것을 탐지하기 때문에 정확한 CVE number를 특정하지 않아 CVE-2018-4878가 분류에 포함되지 않았다. CVE-2015-2545는 EPS 난독화로 정확하게 CVE number를 특정하기 어렵다. CVE-2018-15982는 탐지 도구의 특성상 UAF 취약점을 구분하기 어려워 해당 유형 또한 CVE number로 분류되지 않았다.

Table 5. Comparison between the Anti-Virus and the suggested method

Product	TP	FP	Performance
Our Method	98.45%	1.19%	8m 24s
Ahnlab	76.08%	0.41%	1m 44s
Avast	69.07%	1.09%	4m 16s
Kaspersky	91.48%	0.59%	11m 43s
Symantec	80.73%	0.64%	17m 25s

다른 제품과 비교해봤을 때, 탐지성능이 뛰어나지만, 상대적으로 오탐의 비율이 높으며 이는 실시간으로 탐지해야 하는 Anti-Virus 제품과 비교해서 실시간탐지는 요구되지 않지만, 사용자가 정확히 악성코드의 존재 여부와 위험성을 인지할 필요가 있는 탐지 도구의 특성을 잘 적용한 결과로 보인다. 탐지 도구의 악성 파일 분류인 "Malicious" 탐지율이 98.45%이며, "Suspicious" 결과까지 포함한 경우 전체 탐지율은 99.66% 가 되어 대부분의 악성 의심 파일을 커버할 수 있다. 탐지 못 한 일부 파일을 살펴본 결과 문서 본문에 URL 주소가 나열된 형태의 파일과 단순문서 파일이 첨부된 형태가 있었다. 이는 악성코드의 한 유형으로 보기 어려워 정적 탐지 도구에서 적용하기 어려운 한계로 보인다.

비교한 제품은 Ahnlab V3, Avast Free

Table 6. CVE-Files Detection Result

CVE number	Files	TP	CVE Match
CVE-2017-11882	218	218	218
CVE-2017-0199	189	189	189
CVE-2016-7262	33	33	33
CVE-2017-8570	17	17	17
CVE-2017-8759	12	12	11
CVE-2018-0802	5	5	0
CVE-2014-6352	2	2	2
CVE-2018-4878	3	3	0
CVE-2012-1856	1	1	1
CVE-2018-8414	1	1	1
CVE-2015-2545	1	1	0
CVE-2018-15982	1	1	0

Antivirus, Kaspersky Free Antivirus, Symantec Norton Security이다. 테스트 시 네트워크 연결 없이 진행하였으며, 각 제품의 업데이트 시점은 2019.12.26 23:00이다. 테스트 환경은 CPU Intel 2.5GHz Dual Core, RAM 12GB 조건에서 진행하였다.

VI. 결 론

본 논문에서 OOXML 문서 파일들에 대해 개체 유형별 개체 악성 요소를 분류하여 문제점을 확인하고 악성 문서 파일 탐지를 위한 프레임워크를 제안하였다. 공격자의 OOXML 문서 악성코드 유형은 사용 가능한 기법의 다양성 등으로 파악하기 어렵지만, 유형별 분류 및 탐지 방법을 제안함으로써 이를 이해하고, 침해사고 발생 시 개인 또는 조직이 좀 더 정확한 판단을 할 수 있을 것으로 기대된다.

그러나, 공격으로부터 여전히 위협에 노출되어 있음을 인지해야 하며 업무 성격과 그 공간을 분리함으로써 상대적인 위협으로부터 발생할 수 있는 문제를 예방해야 한다. 또, 관련 작업이 필요한 부서를 제외하고는 본 논문에서 의심 개체요소로 분류한 확장 기능들을 제한할 필요가 있다.

다른 한편으로는 해당 업무 담당자의 시스템만 허용하는 방법과 꼭 확인이 필요한 경우에는 해당 오피스 문서 어플리케이션을 실행하기 위한 가상환경을 구성하는 방법이 있다. 이런 대응 방법은 업무 환경에서 시간과 비용 소요 측면에서 적용하기 어려운 방법이지만 군/기관의 특성상 외부공격을 받은 적이 있거나 위협에 노출된 경우 이에 따른 대비책이 필요하며, 시간과 비용을 감수하더라도 반영할 여지가 있다고 판단된다.

앞에서 언급한 바와 같이 외부 개체 참조를 통해 동적으로 파일을 받아와 로드 시에는 이를 정적으로 탐지하기는 거의 불가능하다. 그런데 공급업체의 정책상 기능은 유지되고 공격자는 Anti-Virus 제품에 대한 우회 용이성으로 꾸준히 사용할 것으로 예상되므로, 대응 방안에 대해 우선적으로 고려되어야 할 것이다.

References

[1] Kaspersky, "Kaspersky Security Bulletin STATISTICS 2016-2019", https://securelist.com, March. 2020.

[2] J. Hurtuk, M. Chovanec, M. Kicina and R. Billik, "Case Study of Ransomware Malware Hiding Using Obfuscation Methods," International Conference on Emerging eLearning Technologies and Applications, pp. 216-217, 2018.

[3] Bora Park, Jungheum Park and Sangjin Lee, "Data concealment and detection in Microsoft Office 2007 files," Digital Investigation, vol. 5(3-4), pp. 149-152, 2009.

[4] Muhammad Ali Raffay et al. "Data Hiding and Detection in Office Open XML (OOXML) Documents," University of Ontario Institute of Technology, pp. 52-57, 2011.

[5] Fu Z., Sun X., Zhou L. and Shu J. "New Forensic Methods for OOXML Format Documents," Digital-Forensics and Watermarking, IWDW, pp. 507-511, 2013.

[6] SungHye Cho and SangJin Lee, "A Research of Anomaly Detection Method in MS Office Document," KIPS Transactions on Computer and Communication Systems, 6(2), pp. 87

- 94, Feb. 2017.
〔7〕 M. Li, Y. Liu, M. Yu, G. Li, Y. Wang and C. Liu, "FEPDF: A Robust Feature Extractor for Malicious PDF Detection," IEEE Trustcom/BigDataSE/ICESS, NSW, pp. 218-224, 2017.
〔8〕 N. Nissim, A. Cohen and Y. Elovici, "ALDOCX: Detection of Unknown Malicious Microsoft Office Documents Using Designated Active Learning Methods Based on New Structural Feature Extraction Methodology," IEEE Transactions on Information Forensics and Security, vol. 12, no. 3, pp 631-646, 2017.
〔9〕 ECMA-376, "Office Open XML File Formats", https://www.ecma-international.org/publications/standards/Ecma-376.htm, Feb. 2020.
〔10〕 ISO/IEC 29500, "Office Open XML File Formats", https://www.iso.org/standard/71691.html, Feb. 2020.
〔11〕 Australian Cyber Security Centre(ACSC), "Microsoft Office Macro Security" Security Report, https://www.cyber.gov.au/publications/microsoft-office-macro-security, Oct, 2019.
〔12〕 Australian Cyber Security Centre(ACSC), "Hardening Microsoft Office 365 ProPlus, Office 2019 and Office 2016" Security Report, https://www.cyber.gov.au/publications/hardening-microsoft-office-365-proplus-office-2019-and-office-2016, Oct, 2019.
〔13〕 GreyHatHacker.NET, "Detecting Malicious Microsoft Office Macro Documents", www.greyhathacker.net/?p=872, Oct, 2019.
〔14〕 GreyHatHacker.NET, "Running Macros via ActiveX Controls", www.greyhathacker.net/?p=948, Oct, 2019.
〔15〕 Microsoft Open Specifications, "2.1.839 Part 1 Section 18.14.4, ddeLink (DDE Connection)", https://docs.microsoft.com/en-us/dotnet/api/documentformat.openxml.spreadsheet.ddelink?view=openxml-2.8.1, Oct, 2019.

〈저자소개〉

이 덕 규 (Deokkyu Lee) 정회원
2014년 2월: 충북대학교 컴퓨터공학부 졸업
2017년 9월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 악성코드, 역공학, 디지털 포렌식

이 상 진 (Sangjin Lee) 종신회원
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~현재: 고려대학교 정보보호대학원 교수
2008년 3월~현재: 고려대학교 디지털포렌식연구센터 센터장
2017년 3월~현재: 고려대학교 정보보호대학원 원장
〈관심분야〉 디지털 포렌식, 심층암호, 해쉬함수