# HWP 문서형 악성코드 위협인자 추출 및 분석 연구*

최민지○ 신강식 정동재
KAIST 사이버보안연구센터
chlalswl0928@kaist.ac.kr, ksshin90@kaist.ac.kr, jjp1018@kaist.ac.kr

# Research on Extracting and Analyzing Threat Factors of HWP Malware

Minji Choe○, KangSik Shin, DongJae Jung
KAIST Cyber Security Research Center

**요 약**

디지털 기기 및 인터넷 기술이 발전됨에 따라 이메일을 활용한 문서형 악성코드가 유포되면서 침해사고가 증가하고 있는 추세이다. 다양한 문서형 악성코드 중 특히 HWP 문서는 주로 국내 공공기관, 일반 기업, 금융기관, 학교 등에서 가장 많이 사용하고 있는 문서 중 하나이다. 악의적인 목적을 가진 사용자들은 HWP 문서의 취약점 및 정상 기능들을 매개체로 이용하여 공격하며, HWP 문서의 미사용 영역 내에 악성코드, 악성 행위를 수행하는 코드를 삽입하는 등의 다양한 공격 형태로 진화되어 가고 있다. 이에 따라 사이버 보안 관점에서 악성 HWP 문서를 탐지하고 분석하기 위해서는 HWP 문서 내 위협인자를 추출하는 기술이 필요하다. 본 논문에서는 파일 포맷 구조 분석을 통해 HWP 파서를 개발하였고 파서로부터 문서 내 위협인자를 추출하는 방법에 대해 설명한다.

## 1. 서 론

디지털 기기 및 인터넷 기술이 발전함과 동시에 악성코드를 이용한 침해사고도 지속적으로 증가하고 있으며[1], 이중 이메일로 유포되는 악성코드의 경우 대부분이 PDF, HWP, MS Office 문서를 이용하여 첨부파일 형태로 유포된다[2].

HWP 문서형 악성코드는 지난 2010년 이후 본격화되기 시작하였는데, 2014년 한수원 관련된 침해사고 이후부터 현재까지 다양한 제목 및 내용으로 악성 HWP 문서가 유포되었다[3]. HWP 문서형 악성코드의 공격 형태는 공격 기술의 변화에 따라 달라지는데 과거에는 윈도우 실행 파일인 PE(Portable Executable) 파일을 바이너리 형태로 문서 파일에 직접 삽입하여 공격하였다. 하지만 최근에는 매크로, 객체 연결 삽입 등과 같은 HWP 문서에서 제공하는 다양한 정상 기능들을 악용하거나 문서의 취약점을 이용하여 공격하는 방식으로 변화되고 있는 추세이다.

악성 HWP 문서는 이메일, 웹사이트와 같은 다양한 경로를 통하여 사용자가 다운로드 받을 수 있어 보안 솔루션을 쉽게 우회할 수 있는 특징을 가지고 있다. 또한 공격자가 사회공학 기법 및 코드 난독화 등 다양한 공격기법을 이용하여 공격을 수행하기 때문에 사이버 보안 관점에서 탐지 및 대응에 있어 매우 어려운 문제가 발생한다. 따라서 HWP 문서를 효과적으로 분석하여 악성행위가 포함되어 있는지 여부를 판단하기 위해서는 선제적으로 위협인자를 추출하는 파서를 통해 HWP 문서의 안정성 식별하는 연구가 필요한다.

본 논문에서는 HWP 파일의 포맷 구조 분석을 통해 악성코드가 삽입될 수 있는 영역을 정의하고, HWP의 안전성을 식별하기 위해 파서를 이용하여 영역 내 비정상적인 위협인자를 추출하는 기법 및 도구를 제안하고자 한다.

* 본 논문은 과학기술정보통신부 글로벌사이버보안기술연구(과제고유번호: 1711125408) 사업의 지원을 받아 수행된 연구임.

## 2. HWP 내부 구조

### 2.1 OLE 파일 포맷 구조

OLE 파일 포맷은 '복합 파일 이진 구조' 라고도 불리며 MS Office 2007 이전 버전과 HWP 5.0 버전에서 사용된다[4]. OLE 파일 구조는 파일 시스템 중 마이크로소프트에서 제작한 FAT(File Allocation Table) 파일 시스템과 유사하며 스토리지(Storage)와 스트림(Stream)으로 이루어진 계층 구조이다. 여기서 설명하는 스토리지는 폴더, 스트림은 파일의 관계와 유사하며 문서 내 객체가 독립적으로 존재하기 때문에 호환성이 뛰어난 장점이 있다. OLE 구조는 그림 1과 같이 HEADER 영역과 DATA 영역으로 나뉜다. HEADER 영역에는 OLE 파일을 구성하는 중요 정보들이 저장되며 DATA 영역에는 스토리지 및 스트림 정보, 스트림 데이터, 테이블 정보(BBAT, SBAT) 등이 저장된다.

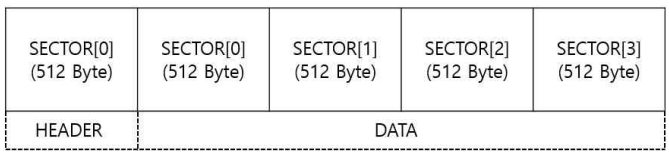


그림 1. OLE 파일 포맷 구조

### 2.2 HWP 문서 구조

HWP 파일 구조는 그림 2처럼 파일 인식 정보, 문서 정보, 본문, 문서 요약, 바이너리 데이터, 미리보기 텍스트 및 이미지, 문서 옵션, 스크립트, 문서 이력 관리, XML 템플릿 정보들로 구성된다[4]. HWP 문서는 파일 내에 존재하는 모든 스트림 및 스토리지를 포함하는 최상위 폴더인 Root Entry를 참조하여 데이터를 얻고

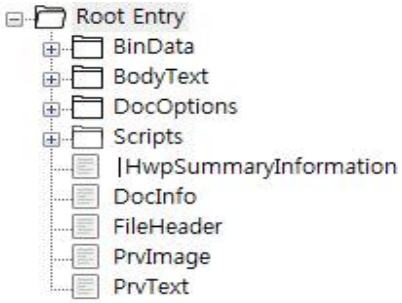


그림 2. MS Office Document File Structure

각 스트림 특성에 따라 레코드 구조 여부나 압축 여부가 결정된다. 레코드 구조란 논리적으로 연관된 데이터들을 헤더 정보와 함께 저장하는 방식으로 문서 정보, 본문, 문서 이력 관리에 사용된다. 압축 여부는 파일 인식 정보 스트림의 압축 플래그를 통해 확인할 수 있으며 압축이 되어 있을 경우 압축을 해제해야 한다.

### 2.3 악성 위협인자 정의

HWP 문서의 경우 복합 파일 이진 구조 형식으로 문서의 미사용 영역 내에 비정상적인 바이너리 데이터 및 스트림 형식으로 악성 위협인자들이 삽입될 수 있다[5]. 악성 위협인자는 표 1과 같이 총 4가지로 구분하여 정의할 수 있다.

**(Scripts)** HWP 문서는 자바스크립트 언어를 사용하여 매크로 작성 및 편집 기능을 제공하는데 Scripts의 DefaultJScript 스트림에는 자바스크립트 코드가 삽입된다. 악의적인 목적을 가진 사용자는 매크로 기능을 활용하여 다른 악성코드를 다운받을 수 있는 웹 주소로 연결시키거나 사용자 PC에 감염시킬 실행 파일들을 임베디드 하여 제작하고 있다.

**(BinData)** BinData 스토리지에는 문서에 첨부된 바이너리 데이터가 각각의 스트림으로 저장된다. 각종 고화질 벡터 이미지를 표현하는 EPS(Encapsulated PostScript) 파일과 하이퍼링크과 같은 자료 연결 기능들이 .EPS, .OLE 스트림 형식으로 저장된다. EPS 파일에는 PE 파일, 원형 또는 XOR로 암호화된 쉘코드, 익스플로잇 코드들을 삽입하고 자료 연결 기능에도 PE 파일, VBS 코드를 삽입하여 공격한다.

**(ViewText)** ViewText 스토리지는 배포용 문서를 사용할 경우 나타나는 스토리지이다. 배포용 문서란 일반 HWP 문서와 다르게 수정 및 편집이 불가능한 읽기 전용 문서로 주로 공공기관에서 중요한 공문을 안내할 때 사용하는 문서 형식이다. ViewText의 Section 스트림에는 그림 3처럼 실제 악성행위를 하는 쉘코드 영역이 삽입되어 존재한다.

**(BodyText)** BodyText 스토리지에는 문서에서 사용되는 문단, 표, 그리기 개체와 같은 내용이 스토리지 하위에 Section%d 스트림으로 구분되어 저장된다. 해당 스토리지는 압축 및 암호화가 적용 가능한 영역으로 레코드로 구성되어 있으며 레코드에 따라 분석하고 데이터를 추출하면 스토리지 내 문단 텍스트를 의미하는 HWPTAG_PARA_TEXT 태그에 비정상적인 값이 삽입되어 있다. 이는 쉘코드 영역이 삽입되어 있음을 의미한다.

표 1. 악성 위협인자 정의

| 스토리지 | 스트림 | 임베드 방식 |
|---|---|---|
| Scripts | DefaultJScript | 파일 |
| BinData | EPS, OLE | 파일, 쉘코드 |
| ViewText | Section | 쉘코드 |
| BodyText | Section | 쉘코드 |

```
/Section1_dec.txt[('[0x0080010C]',), 71, 1, '0x10', 16]
[('[0x008000FA]',), 75, 2, '0xe', 14]
[('[0x008000E8]',), 75, 2, '0xe', 14]
[('[0x008000D6]',), 75, 2, '0xe', 14]
[('[0x008000B6]',), 74, 2, '0x1c', 28]
[('[0x00800096]',), 74, 2, '0x1c', 28]
[('[0x0080006A]',), 73, 2, '0x28', 40]
[('[0x00800040]',), 71, 1, '0x26', 38]
[('[0x00800034]',), 68, 1, '0x8', 8]
[('[0x0000001A]',), 67, 1, '0x800012', 8388626]
[('[0x00000000]',), 66, 0, '0x16', 22]
```

그림 3. BodyText의 문단 텍스트 태그에 삽입된 비정상적인 값

## 3. HWP 문서형 악성코드 위협인자 추출 도구 구현

### 3.1 HWP 문서형 악성코드 위협인자 추출 아키텍처

HWP 문서형 악성코드를 효과적으로 탐지하고 분석하기 위해서는 문서 내에 악성 위협인자를 추출하는 파서가 필요하다. 그림 4는 본 논문에서 설명하는 HWP 문서형 악성코드 위협인자 추출 도구의 아키텍처로 해당 도구는 3개의 영역으로 구성되어 있으며 아키텍처 모듈별 주요 단계 및 기능은 다음과 같다.

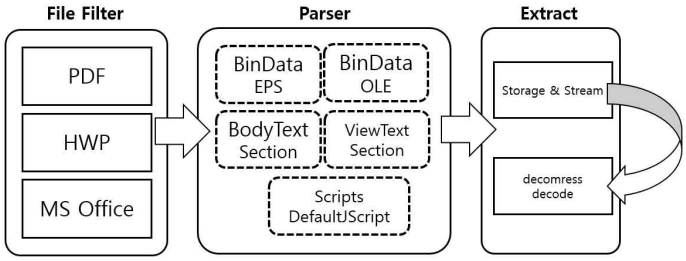


그림 4. HWP 문서형 악성코드 위협인자 추출 아키텍처

- 1단계 다양한 문서 중 HWP 문서인지 판별하기 위해 각 문서를 식별할 수 있는 시그니처 즉, 매직 넘버(Magic number)를 활용한 HWP 문서 판별
- 2단계 HWP 문서 판별 이후 파서를 통해 악성 위협인자로 정의한 BinData의 .EPS 또는 .OLE, Scripts의 DefaultJScript, BodyText와 ViewText의 Section 스트림 추출
- 3단계 HWP 문서 압축시 사용되는 zlib 라이브러리를 이용하여 2단계에서 추출한 악성 위협인자(스트림) 압축 해제

### 3.2 HWP 문서형 악성코드 위협인자 추출

그림 5는 HWP 문서형 악성코드 위협인자를 추출하는 순서도이며 위협인자 추출 방식의 원리는 다음과 같다. ① Input으로 파일 또는 디렉토리 옵션 전달, ② 매직넘버 식별을 통한 HWP 문서 확인, Storage 및 Stream 형식으로 구성되어 있는지 확인하여 HWP 문서 파싱, ③ 기에 악성 위협인자라고 정의한 요소들이 존재하는지 확인하고 존재한다면 악성 위협인자(EPS, OLE, JavaScript, Section) 추출, ④ 추출한 데이터들의 압축을 해제한 한 후 "스트림명.dec 파일명"으로 로그 생성

### 3.3 추출된 악성 위협인자 분석

제안한 위협인자 추출 도구를 이용하여 실제 100개의 HWP 문서형 악성코드 샘플을 대상으로 악성 위협인자를 추출하고 추출된 데이터를 수기로 분석하는 실험을 진행하였다. 샘플 데

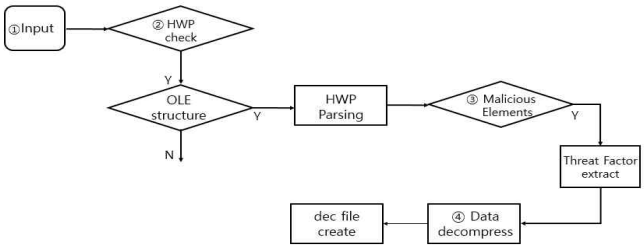


그림 5. 위협인자 추출 순서도

이터에서 정상적으로 악성 위협인자가 추출되었으며, 악성 Scripts 4개, 악성 BinData 75개, 악성 ViewText 2개, 악성 BodyText 19개와 같은 결과를 얻을 수 있었다. 추출된 악성 위협인자를 수기로 분석해 보았더니 실제 100개의 악성코드에는 PE 파일이 바이너리 형태로 삽입되거나, 실제 악성 행위를 하는 쉘코드 및 취약점을 유발하는 익스플로잇 코드들이 삽입되어 있음을 확인하였다.

## 4. 결론 및 향후 방안

본 논문에서는 국내에서 가장 많이 사용되는 HWP 문서를 대상으로 HWP 문서 내에 삽입될 수 있는 악성 위협인자 특징 및 영역에 대해 정의하였고, 이를 실제 100개의 HWP 문서형 악성코드 샘플을 통해 개발한 악성 위협인자 추출 도구로부터 악성 HWP 문서에 존재하는 다양한 악성 위협인자(스토리지 및 스트림)를 추출하는 실험하여 제안한 위협인자 추출 도구의 유효성을 입증하였다. 향후 연구로는 HWP 문서로부터 추출한 악성 위협인자들을 세부적으로 분석하여 공격에 이용되는 분석 기준을 정의하고 분석 기준을 기반으로 악성 HWP 탐지 룰셋 정의 후 탐지도구를 개발하여 문서형 악성코드에 대한 분석 및 탐지 시스템으로 연구를 확장할 계획이다.

## 5. 참고 문헌

[1] KISA, “20201년 상반기 사이버 위협 동향보고서”, 2021, July.

[2] Financial Security Institute(금융보안원), “한글 문서를 이용하는 악성코드 프로파일링”, 2018 사이버 위협 인텔리전스 보고서, 2018

[3] Financial Security Institute(금융보안원), “코로나 19 금융부문 사이버 위협동향”, 2020, May.

[4] https://www.hancom.com/etc/hwpDownload.do

[5] Kang, A. R., Jeong, Y. S., Kim, S. L., Kim, J., Woo, J.,& Choi, S., “Detection of malicious pdf based on document structure features and stream objects”. Journal of The Korea Society of Computer and Information, 23(11), 85-93, 2018.

[6] Cho, Sung Hye, and Sang Jin Lee. “A Research of Anomaly Detection Method in MS Office Document.“ KIPS Transactions on Computer and Communication Systems 6.2, 87-94. 2017.

[7] Lee, Deokkyu, and Sangjin Lee. “A Study of Office Open XML Document-Based Malicious Code Analysis and Detection Methods.“ Journal of the Korea Institute of Information Security & Cryptology 30.3, 429-442, 2020.

[8] 서민정, et al. “오픈소스 기반 문서형 악성코드 차단 프로그램의 개발”, 한국정보처리학회 학술대회논문집, 27.2: 424-427, 2020