Feature Selection 기법을 이용한 북한 ODT 파일 퍼징 테스트케이스 분석

남지희*, 강동수+
국방대학교 컴퓨터공학전공
e-mail: *namdi9012@gmail.com, +greatkoko@kndu.ac.kr

Analysis of ODT File Fuzzing Testcase in North Korea using Feature Selection Method

JiHee Nam*, Dongsu Kang+
* +Dept. of Computer Science & Engineering, Korea National Defense University

요 약

소프트웨어의 비정상적인 작동인 크래시는 보안 취약점의 원인이 된다. 이러한 크래시로부터 야기되는 취약점을 예방하기 위해 다양한 테스트케이스를 생성하고 크래시를 발견 및 분석하는 연구가 지속되고 있다. 본 논문에서는 북한 소프트웨어 서광사무처리체계에서 사용하는 국제 사무용 전자문서 형식인 Open Document Format for Office Application (ODF)의 워드프로세스 문서 형태인 ODT파일의 효과적인 보안 테스트케이스를 찾기 위해 먼저 테스트케이스를 도출한다. 도출된 테스트케이스를 데이터 전처리한 후 Feature Selection 기법을 적용하여 의미 있는 속성들을 분류한다. 마지막으로 ODT 파일 내에 크래시를 유발하는 유의미한 속성들을 확인하고 퍼징 테스트케이스 작성 시 메트릭으로 활용할 수 있다.

1. 서론

크래시는 소프트웨어가 적절하게 기능하지 못하고 중단된 상태로, 크래시의 발생은 SW 취약점의 원인이 될 수 있다. 이러한 크래시와 취약점을 예방하기 위해서 설계 단계에서부터 다양한 SW 보안 테스트를 진행하고 있다[1]. 여러 SW 보안 테스팅 기법 중에서 퍼징테스트는 SW에 결함을 주입하는 것으로, 취약점을 찾고자 하는 소프트웨어에 임의의 비정상적인 값을 주입하여 발생하는 비정상적인 상태를 유도하고 그 결과로부터 취약점을 찾아내는 것이다[2]. 퍼징테스트를 이용하여 취약점을 발견하려는 연구가 지속적으로 진행되고 있으며[3-4], 실제로 여러 취약점이 발견되었다[5]. 최근에는 퍼징테스트의 효율성을 높이기 위해서 머신러닝 기법과 퍼징테스트를 결합하는 연구도 다양하게 진행되고 있다.

본 논문에서는 북한 소프트웨어 서광사무처리체계에서 사용하는 국제 사무용 전자문서 형식인 Open Document Format for Office Application (ODF)의 워드프로세스 문서 형태인 Open Document Text (ODT) 파일 퍼징테스트에서 효과적인 테스트케이스를 찾기 위해 머신러닝의 한 종류인 속성선택기법을 적용하는 연구를 진행한다.

논문의 구성은 2장에서 북한 서광사무처리체계, ODT와 속성선택기법에 대해 알아보고, 3장에서는 ODT 파일 테스트케이스 분석 기법을 제시한다. 4장에서는 퍼징테스트에 적용할 효율적인 테스트케이스 생성을 위해 속성선택기법과 관련지어 실험 및 분석을 진행하고, 마지막으로 5장에서는 연구내용을 정리하고, 향후 방향을 제시한다.

2. 관련연구

2.1 북한 서광사무처리체계와 ODT

북한은 리눅스 기반의 자체 개발 운영체제인 붉은별(Red Star)을 사용하고 있다[6]. 붉은별 내에는 다양한 응용프로그램들이 설치 가능하며, 그 중 한컴 Office와 유사한 서광사무처리체계가 대표적인 프로그램이다. 서광사무처리체계는 평양인쇄공업대학에서 개발한 패키지 프로그램으로 (그림 1)과 같이 문서처리체계, 자료표체계와 연시물체계 등으로 구성되어 있다.

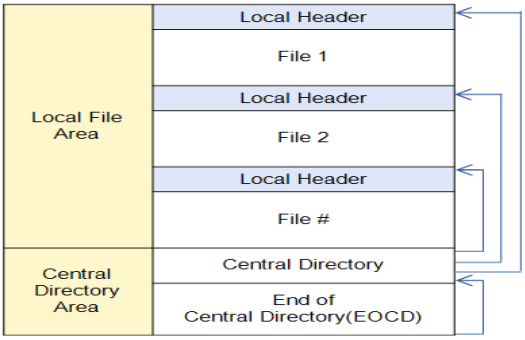
(그림 1) 서광사무처리체계 구동화면

+ 교신저자: greatkoko@kndu.ac.kr

서광사무처리체계의 확장자는 개방형 문서표준 포맷으로 XML기반이며, 특정 어플리케이션이나 플랫폼에 종속되지 않는 개방 문서 형식이다. ODF는 2006년 ISO/IEC 26300:2006에서 발표되었으며 OASIS (Organization for the Advancement of Structured Information Standard)에서 표준화하였다.

ODF는 투명하고 개방적인 프로세스로 전 세계의 사용자가 공개된 커뮤니티 등을 통해 파일 포맷이나 형식에 관한 의견을 공유하고 수렴하여 개발되는 특징이 있다. 영국과 독일 등 일부 국가에서는 정부 문서에 ODF를 의무적으로 채택하여 사용하고 있다.

ODT는 ODF의 워드프로세스 형식으로 파일의 구조는 ZIP 파일과 유사하다[7]. 크게 로컬파일 영역과 센트럴 디렉터리 영역으로 구분되며, 세부 구조는 (그림 2)와 같다.



(그림 2) ODT 파일 구조

로컬파일 영역은 최소 1개 이상의 로컬파일들로 구성되었으며, XML 파일 형식이다. 로컬파일 영역은 다시 Header 영역과 Data 영역으로 구분되는데 Header 영역에는 파일의 압축 정보와 같은 메타데이터들이 저장되고, Data 영역에는 압축된 실제 데이터들이 저장된다[8]. 로컬파일 내에는 Content.xml, Meta.xml, Setting.xml, Style.xml, Manifest.xml로 구성된 5개의 XML이 있으며, 이 XML 파일들은 DOM(Document Object Model) 표준을 사용하고 있다.

2.2 속성선택기법

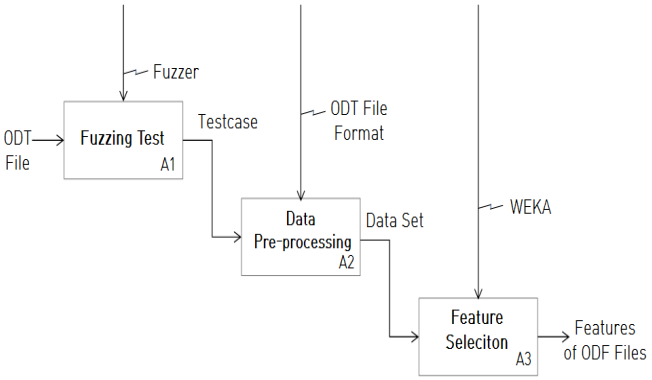
속성선택(Feature Selection)은 기계학습의 하나로 많은 데이터 중에서 원본데이터로부터 가장 좋은 성능을 나타내는 데이터의 부분 집합을 찾기 위해 주요 속성들을 찾아내는 것이다[9]. 속성선택기법은 특징 부분 집합을 어떻게 생성할지, 어떤 과정으로 평가할지에 따라 크게 Filter methods, Wrapper methods와 Embedded methods로 구분된다. Feature Selection 기법 분류는 <표 1>과 같다.

<표 1> Feature Selection 기법 분류

구 분	내 용
Filter methods	· 결과 변수와의 상관관계 분석 · 통계적인 방법으로 스코어를 매겨 높은 순서대로 특징선택
Wrapper methods	· 가장 분류 정확도가 높은 특징 조합 도출 · 속성의 하위집단 검색 후 선택한 특성 평가
Embedded methods	· Filter와 Wrapper methods 결합 · 특징선택과 분류를 동시에 수행

3. ODT 파일 테스트케이스 분석 기법

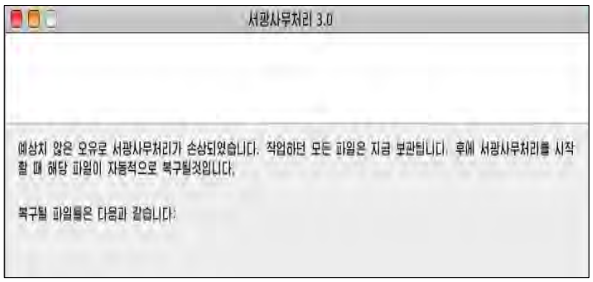
제안하는 ODT 파일 테스트케이스 분석 절차는 (그림 3)과 같다. 먼저, ODT 파일에 대한 퍼징테스트를 통해 테스트케이스를 추출하고, 데이터 전처리 과정을 통해 속성선택기법 분석을 위한 데이터 세트를 구축한다. 마지막으로, 속성선택기법을 적용하여 유의미한 속성을 도출해내는 프로세스이다.



(그림 3) ODT 파일 테스트케이스 분석 방법

3.1 퍼징테스트

퍼징테스트 단계에서는 ODT 파일에서 특정 영역을 선택한 뒤 임의의 값을 변화시켜 크래시를 발생시키는 변이를 진행한다. 입력 데이터의 16진수 바이너리 값 중 상단 최우측 2바이트를 최대값 FF FF로 변경한다. 그 결과 (그림 4)와 같이 서광사무처리체계 ODT 파일에 오류가 발생하여 파일의 정상적인 작동이 제한되는 상태들을 발견할 수 있다.



(그림 4) 퍼징테스트 후 ODT 파일 오류

3.2 데이터 전처리

ODT 파일에서 입력 가능한 항목들과 실제 ODT 파일에 퍼징테스트 후 크래시가 발생하였을 때 확인 가능한 항목들을 도출하여 실험 데이터를 만든다. 크래시가 발생한 파일에서 <표 2>와 같이 ODT파일 속성을 9개 도출할 수 있다.

<표 2> ODT 파일 속성

1	2	3	4
Filename	Line	Size	Tag

5	6	7	8	9
Image	Table	Figure	Text	Crash

3.3 속성선택

데이터 전처리에서 도출된 속성 중 유의미한 속성을 선택하기 위해 뉴질랜드의 Waikato 대학에서 개발한 기계학습 알고리즘 도구인 'WEKA'를 사용한다. 특정 상황에서 종속적인 Wrapper methods와 Embedded methods를 제외하고 Filter methods를 사용한다[10]. Filter method의 대표적인 알고리즘은 Feature Evaluator와 Search Method가 있고, 세부 알고리즘은 <표 3>과 같으며[11], 알고리즘별 특징을 기술하면 다음과 같다.

<표 3> Feature Evaluator와 Search Method 알고리즘

구분	Search Method	Feature Evaluator
알고리즘	BestFirst	CfsSubsetEval
	Greedy Stepwise	
	Ranker	GainRatioAttributeEval
		SymmetricalUncertEval
		InfoGainAttributeEval
		CorrelationAttributeEval

- **BestFirst:** 휴리스틱에 따라서, 최근의 모든 경로들을 순서화하여 깊이 우선 탐색을 최적화하는 탐색 알고리즘
- **GreedyStepwise:** 속성 하위 집합의 공간을 통해 정방향 또는 역방향 검색을 수행, 최적해를 구하는데 사용
- **Ranker:** 개별 평가에 따라 속성의 순위를 나타내는 알고리즘
- **CfsSubsetEval:** 각 기능의 개별 예측 가능성과 그 사이의 중복 정도를 고려하여 속성 하위 집합의 가치를 평가
- **GainRatio AttributeEval:** 클래스에 대한 이득 비율을 측정하여 특성의 가치를 평가
- **SymmetricalUncertAttributeEval:** 클래스와 관련하여 대칭 불확실성을 측정하여 속성의 가치를 평가
- **InformationGainAttributeEval:** 클래스와 관련된 정보 획득을 측정하여 속성의 가치를 평가
- **CorrelationAttributeEval:** 클래스와 클래스간의 상관관계를 측정하여 속성의 가치를 평가

4. 실험 및 분석

3장에서 제시한 방법을 분석하기 위해 실험환경을 구축하고, ODT 파일에 퍼징을 통한 테스트케이스를 추출, 데이터 전처리와 속성선택기법을 적용하여 속성을 도출해낸다.

4.1 환경구성 및 실험

실험을 위한 환경구성은 <표 4>와 같으며, 먼저 기존 파일 퍼징 기법을 적용하여 테스트케이스를 산출한다. 도출된 테스트 케이스들에서 파일 속성에 따른 값을 산출하고 WEKA를 이용해 관련 있는 속성을 선택한다. 실험을 위해 일부 속성에 따른 데이터 값은 크래시가 발견되는 파일을 참조해 6개의 데이터 세트로 가정하여 생성하였다. 데이터 세트 A와 B는 크래시 발생률이 50%이며, C는 20%, D는 80%, E는 30%, F는 70%로 구성하였다.

<표 4> 실험환경

구 분		내 용
장비	CPU	Intel(R) Core(TM) i5-8250U CPU @ 1.6GHz
	RAM	16GB
	SSD	256GB
	OS	Windows 10 Home
SW	서광사무처리체계	Ver 3.0
도구	VMware	Workstation 14 player
		Memory: 2GB 저장용량: 20GB Processor: 2 운용체계: 붉은별 3.0
	HxD(헥사에디터)	Ver 1.7.7
	Notepad++	Ver 7.5.8

4.2 분석결과

WEKA의 속성연관 규칙 분류 알고리즘에 적용한 결과는 <표 5>와 같다. 먼저 BestFirst와 Greedy Stepwise는 최상위 3가지 속성만 선택되고, Ranker 알고리즘은 9개의 속성이 순서대로 선택된다.

<표 5> Feature Selection 실험 결과

데이터	알고리즘	결과		
A	BestFirst, CfsSubsetEval	1	5	7
	GreedyStepwise, CfsSubsetEval	1	5	7
	Ranker, GainRatioAttributeEval	1	5	6
	Ranker, SymmetricalUncertEval	1	5	6
	Ranker, InfoGainAttributeEval	1	5	6
	Ranker, CorrelationAttributeEval	1	4	5
B	BestFirst, CfsSubsetEval	3	5	6
	GreedyStepwise, CfsSubsetEval	3	5	6
	Ranker, GainRatioAttributeEval	3	5	6
	Ranker, SymmetricalUncertEval	3	5	6
	Ranker, InfoGainAttributeEval	3	5	6
	Ranker, CorrelationAttributeEval	4	5	6
C	BestFirst, CfsSubsetEval	3	5	8
	GreedyStepwise, CfsSubsetEval	3	5	8
	Ranker, GainRatioAttributeEval	3	5	8
	Ranker, SymmetricalUncertEval	3	5	8
	Ranker, InfoGainAttributeEval	3	5	8
	Ranker, CorrelationAttributeEval	4	5	8
D	BestFirst, CfsSubsetEval	3	5	6
	GreedyStepwise, CfsSubsetEval	5	6	-
	Ranker, GainRatioAttributeEval	3	5	6
	Ranker, SymmetricalUncertEval	3	5	6
	Ranker, InfoGainAttributeEval	3	5	6
	Ranker, CorrelationAttributeEval	2	4	5
E	BestFirst, CfsSubsetEval	3	5	-
	GreedyStepwise, CfsSubsetEval	3	5	-
	Ranker, GainRatioAttributeEval	3	5	6
	Ranker, SymmetricalUncertEval	3	5	6
	Ranker, InfoGainAttributeEval	3	5	6
	Ranker, CorrelationAttributeEval	1	4	5
F	BestFirst, CfsSubsetEval	3	5	-
	GreedyStepwise, CfsSubsetEval	3	5	-
	Ranker, GainRatioAttributeEval	3	5	6
	Ranker, SymmetricalUncertEval	3	5	6
	Ranker, InfoGainAttributeEval	3	5	6
	Ranker, CorrelationAttributeEval	1	4	5

실험 결과 총 9개의 속성 중 분류 Class로 지정한 Crash 속성을 제외하고 Image, Size, Table, Filename, Tag, Text, Figure, Line의 순으로 속성이 추출되었다. 추출된 수가 많은 상위 4개의 속성인 Image, Size, Table, Filename은 ODT 파일 테스트케이스 생성 시 활용이 가능한 속성이라는 특징이 있다.

5. 결론 및 향후 연구

크래시를 발견하는 것은 SW의 보안성을 향상시키는데 있어 중요한 부분이다. 본 논문에서는 크래시를 발견하는 방법의 하나인 퍼징테스트의 효율적인 테스트케이스를 도출하기 위해 Feature selection 기법을 적용하여 의미 있는 특징들을 선별하였다.

향후 연구에서는 ODT 파일에 대해서 추가적인 퍼징테스트를 진행한 후 더 많은 실험 데이터를 확보하여 속성 선택기법에 의한 결과의 타당성을 확보할 예정이다. 이를 통해 ODT 파일 내에 크래시를 유발하는 효과적인 속성들을 추출함으로써 효율적인 보안 테스트 수행이 가능하다.

참고문헌

[1] R.shirey, RFC 2828 - Internet Security Glossary, 2007.

[2] H Tahbildar, B Kalita, “Automated software test data Generation: Direction of Research,” International Journal of Computer Science and Engineering Survey, Vol.2, No.1, pp.99-120, 2011.

[3] P. Uhley, “Advanced Persistent Responses,” CanSecWest, 2012.

[4] Sangsu Kim, Dongsu Kang, “Fuzzing-based Test Case Generation Technique for Multimedia File Vulnerability Analysis,” Journal of Security Engineering, Vol.14, No.6, pp.441-458, 2017.

[5] A. Manion, M. Orlando, “Fuzz Testing for Dummies,” Industrial Control Systems Joint Working Group Spring Meeting, 2011.

[6] Jongseon Kim, Lee Choongeun, “Analysis and Cooperation of North Korea's IT Technology in Uniform Preparation,” Science and Technology Policy Institute, 2014.

[7] Chanju Park, Dongsu Kang, “Analysis of file structure about Red Star's SeoKwang Document Processing System for security vulnerability analysis,” Proceedings of the Korea Information Processing Society, Vol.25, No.1, pp.110-112, 2018.

[8] Byungjoon Jung, et al. “A Method of Recovery for Damaged ZIP Files,” Journal of The Korea Institute of Information Security & Crypto logy, Vol.27, No.5, 2017.

[9] Jungdong Li, et al. “Feature selection : A data perspective,” ACM Computing Surveys(CSUR) 50.6(2017): 94, 2017.

[10] Binita Kumari, Tripti Swarnkar, “Filter versus wrapper feature subset selection in large dimensionality micro array: A review,” 2011.

[11] Hojin Lee, et al. “FEATURE SELECTION PRACTICE FOR UNSUPERVISED LAERNING OF CREDIT CARD FRAUD DETECTION,” Journal of Theoretical & Applied Information Technology 96.2, 2018.