Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.2, Apr. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.2.327

Peach 퍼징을 위한 파일 분석 데이터 자동 생성 모델

김 민 호,[1†] 박 성 빈,[1] 윤 지 노,[1] 김 민 수,[2‡] 노 봉 남[1]
[1]전남대학교 시스템보안연구센터, [2]목포대학교

File Analysis Data Auto-Creation Model For Peach Fuzzing

Minho Kim,[1†] Seongbin Park,[1] Jino Yoon,[1] Minsoo Kim,[2‡] Bong-Nam Noh[1]
[1]Chonnam National University SSRC, [2]Mokpo National University

요 약

소프트웨어가 산업 및 사회전반으로 저변을 확대함에 따라 소프트웨어의 취약점으로 인한 위험이 증가하고 있으며, 소프트웨어의 취약점을 악용하는 사례도 빈번하게 나타나고 있다. 스마트 퍼징은 이러한 소프트웨어 취약점을 발견하기 위한 자동화된 방법이다. 그러나 스마트 퍼징을 위해서는 우선 퍼징을 수행하고자 하는 대상 소프트웨어에 대한 데이터 모델을 생성해야하며, 데이터 파일 및 소프트웨어 자체에 대한 분석이 필요하기 때문에 많은 자원이 소모된다. 따라서 효율적인 스마트 퍼징을 위해서 데이터 모델을 자동으로 생성하기 위한 방법이 필요하다. 본 논문에서는 데이터 입력 파일에 대한 분석을 통해서 스마트 퍼징을 위한 데이터 모델을 자동으로 생성하기 위한 프레임워크를 제안하고 이를 구현함으로서 소프트웨어 취약점 발견에 도움을 줄 것으로 기대한다.

ABSTRACT

The rapid expansion of the software industry has brought a serious security threat and vulnerability. Many softwares are constantly attacked by exploit codes using security vulnerabilities. Smart fuzzing is automated method to find software vulnerabilities. However, Many resources are consumed in fuzzing, because the fuzzing needs to create data model for target software and to analyze a data file and software binary. Therefore, The automated method for efficient smart fuzzing is needed to develop the automated data model. In this paper, through analysing the input file format and optimizing the data structure, we propose an efficient data modeling framework for smart fuzzing and implement the framework for detect software vulnerabilities.

Keywords: Smart fuzzing, Data Analysis, Data Model

I. 서 론

오늘날 소프트웨어 이용 범위가 확대되고 중요성이 높아짐에 따라 소프트웨어의 안전성에 대한 문제가 대두 되었다. 실제로 소프트웨어의 취약점은 지속적으로 발견되고 있으며, 이를 이용한 사이버 공격이 증가하고 있다[1]. 2012년 한해 5,291개의 새로운 취약점이 발견되는 등 소프트웨어 취약점에 대한 위험성이 부각되고 있다[2][3]. 소프트웨어 테스팅 분야의 많은 연구자들에 의해 소프트웨어의 취약점을 사전에 발견하여 제거하는 방법으로 소프트웨어의 안전성을 높이려는 많은 연구가 진행되고 있다. 소프트웨어 테스팅 분야 중 하나인 퍼징은 자동화된 메커니즘을 이용해 소프트웨어의 취약점을 발견할 수 있는 방법이다. 소프트웨어 개발사에서는 자사의 소프트웨어에 대한 취약점을 발견하여 패치하기 위해 퍼징 기법을 이용하

접수일(2013년 10월 29일), 수정일(2014년 1월 21일), 게재확정일(2014년 2월 5일)
† 주저자, linz@lsrc.jnu.ac.kr
‡ 교신저자, phoenix@mokpo.ac.kr(Corresponding author)

고 있다[4].

퍼징은 블랙박스 테스팅인 덤 퍼징(Dumb Fuzzing)과 그레이박스 테스팅인 스마트 퍼징(Smart Fuzzing)으로 분류된다. 덤 퍼징은 대상 소프트웨어에 대한 이해 없이 퍼징을 수행한다. 스마트 퍼징은 대상 소프트웨어에 대해서 사전에 분석된 정보 즉, 데이터 모델을 바탕으로 퍼징을 수행한다. 스마트 퍼징은 대상 소프트웨어에 대한 이해를 바탕으로 퍼징을 수행하기 때문에 보다 정밀한 퍼징이 가능하다는 장점이 있다. 반면에 스마트 퍼징을 위해서는 우선 퍼징을 수행하고자 하는 대상 소프트웨어에 대한 데이터 모델을 생성해야하며, 데이터 파일 및 소프트웨어 자체에 대한 분석이 필요하기 때문에 많은 노력이 요구된다. 따라서 효율적인 스마트 퍼징을 위해서 데이터 모델을 자동으로 생성하기 위한 방법이 필요하다.

본 논문에서는 데이터 입력 파일에 대한 분석을 통해서 스마트 퍼징을 위한 데이터 모델을 자동으로 생성하기 위한 프레임워크를 제안한다. 제안하는 방법은 입력 파일을 분석하여 파일 포맷에 의존적인 데이터 모델을 자동으로 생성하는 방법으로서, 샘플 파일에 의존적인 데이터 모델을 자동으로 생성함으로써 파일 포맷을 분석하는 시간과 데이터 모델 생성을 위한 자원을 줄여 기존보다 효율적으로 퍼징을 수행하는 것이 가능하다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구를 통해 기존 퍼징 시스템과 데이터 모델 생성에 있어서 한계성을, 3장에서는 소프트웨어에 대한 입력 데이터 분석에 따른 자동화된 데이터 모델 생성 메커니즘을, 4장에서는 데이터 모델 자동 생성 프레임워크 개발 결과를, 5장에서는 프레임워크를 이용하여 특정 소프트웨어에 대한 스마트 퍼징을 수행한 결과를 소개한다.

II. 관련 연구

소프트웨어 퍼징 기술은 분석 대상 소프트웨어에 대해 입력 값을 변형하면서 소프트웨어 오류 및 결함을 찾아가는 기술이다. 대상 소프트웨어 이해 유무에 따라 덤 퍼징, 스마트 퍼징으로 분류되며, 최근 퍼징 기술에서는 기존의 퍼징 연구에서의 자원적 문제를 해결하기 위해 메모리 퍼징, 코드 커버리지와 같은 기술을 접목하여 연구를 수행하고 있다[5].

덤 퍼징은 분석 대상 소프트웨어에 대한 기반 지식 없이 랜덤하게 생성된 입력 값을 이용하여 퍼징을 수행하는 기법이다. 스마트 퍼징은 취약점을 탐색하기 위한 대상 소프트웨어에 대한 분석을 선행하고, 이에 최적화된 데이터를 입력하여 퍼징을 수행하는 방식으로 데이터 생성 방법에 따라 생성기반 퍼징(Generation based Fuzzing)과 변이기반 퍼징(Mutation based Fuzzing)으로 나누어진다[6]. 생성기반 퍼징은 데이터 모델을 기반으로 소프트웨어의 입력 데이터를 생성하고 이를 입력으로 소프트웨어를 테스팅하는 방법이다. 변이기반 퍼징은 데이터 모델을 기반으로 특정 입력 데이터의 일부분을 변조하고 이를 입력으로 소프트웨어를 테스팅하는 방법이다.

2.1 소프트웨어 취약점 분석 과정

파일 퍼징은 입력 파일이 존재하는 소프트웨어에 대한 취약점을 탐지하기 위한 방법으로서, 입력 파일 선정 및 분석, 데이터 처리, 퍼징 데이터 출력의 단계로 수행된다. 일반적으로 다수의 샘플 파일을 수집하고, 이에 대한 파일 포맷 분석을 수행한 다음 데이터 모델을 생성하고 이를 이용하여 퍼징을 수행한다. 그림 1은 기존 파일 퍼징을 통한 취약성 분석 과정을 나타낸다.

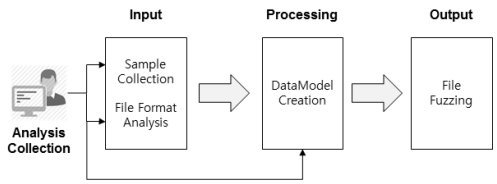


Fig.1. Vulnerability analysis process using file fuzzing

샘플 파일의 수집 이후 포맷 분석과 데이터 모델 생성을 분석자가 직접적으로 처리하기 때문에 많은 시간 및 자원이 소모된다.

2.2 스마트 퍼징 도구 및 기존 접근 방법

소프트웨어 퍼징을 수행하기 위한 대표적인 기존 도구들로는 filefuzz, Ioctlfuzzer, bestorm, peach 등이 있다. filefuzz는 닷넷 기반 파일 퍼징 도구로서 퍼징 대상 프로그램에 인자 값으로 입력될 변수가 퍼징의 대상이 되며, 퍼징 수행 시 입력되는 데이터의 크기 및 범위를 조절할 수 있다[7]. 단순히 파일 퍼징만을 수행하기 때문에 입력 데이터에 대한 세밀한 조정이 불가능하여 효율적이지 못하다.

Ioctlfuzzer은 윈도우 드라이버의 취약점을 찾기 위해 개발된 퍼징 도구이다. NtDeviceIoControl File 함수를 후킹 하여 디바이스 드라이버의 IOCTL을 조작하여 퍼징을 수행한다[8]. 드라이버에 대한 퍼징에 특화되어 있기 때문에 특정한 함수만을 퍼징하며, 입력 데이터를 다양하게 사용하지 못하는 단점이 존재한다.

bestorm은 파일 및 프로토콜, API 퍼징이 가능한 도구로서 포맷 정보가 저장된 DB를 이용하여 퍼징을 수행한다[9]. 그러나 DB에 저장된 정보만을 이용하여 퍼징을 위한 데이터를 생성하기 때문에 구조가 복잡한 입력 파일은 사용하기 힘든 한계가 있다.

peach는 데이터 변이를 기반으로 퍼징을 수행하는 도구로서, 피치 핏(pit - XML 형태로 정의된 데이터 모델) 파일을 요구한다. 피치 핏 파일은 피치 플랫폼에서 퍼징을 수행하기 위한 모든 정보를 포함하고 있으며, 피치를 이용하여 퍼징을 수행하기 위해서는 반드시 피치 핏 파일 생성하는 작업을 선행해야 한다[9]. 피치 핏은 퍼징을 수행하기 위한 데이터에 대한 구조, 타입 정보, 관계정보 등이 정의되어 있다. 이를 기반으로 입력 파일을 파싱하고 특정 영역을 변조함으로서 퍼징을 수행한다[10]. 또한 변이 옵션을 설정함으로서 취약점이 발생할 가능성이 높은 영역에 대해서만 데이터 변조를 수행할 수 있다. 따라서 데이터 모델인 피치 핏 파일을 생성하는 과정이 매우 중요하다.

2.3 네트워크 데이터 모델 자동 생성 연구

스마트 퍼징 과정에서 데이터 모델을 생성하는 작업에 대한 인적, 시간적 자원 소모가 크기 때문에 네트워크 기반 퍼징에 한하여 데이터 모델을 자동으로 생성하기 위한 연구가 진행되었다.

Netzob는 문서화되지 않은 네트워크 프로토콜을 분석하여 퍼징을 수행하기 위한 데이터 모델의 생성과정에 활용하기 위한 연구 프로젝트이다. Netzob는 네트워크 패킷을 문법적 추론과 단어적 추론 기법을 통해 분석하여 데이터 모델을 자동으로 생성하고 이를 이용하여 Peach 기반의 퍼징을 수행한다. 단어적인 추론은 패킷에 존재하는 메시지들을 클러스터링 하여 문자열 필드로 구분하는 과정이며, 문법적인 추론은 수집된 필드들의 연관성을 조사하여 데이터들을 프로토콜에 맞게 재구성하는 과정이다. 단어 및 문법 추론을 통해 분석된 네트워크 프로토콜을 기반으로 데이터 모델을 생성하며, Peach의 변이 및 제너레이션 기능을 이용해 테스트 케이스가 생성된다. 패킷 분석을 통한 데이터 모델 생성 과정을 단축시켜 퍼징을 수행하기 위한 시간을 줄일 수 있다는 장점이 있다. 그러나 대상이 네트워크 패킷 프로토콜로 제한되어 있어서 입력 파일을 요구하는 소프트웨어를 대상으로 퍼징을 수행하는 것이 불가능하다. 또한 단순히 문자열의 단어적, 문법적 요소들만을 사용하여 데이터 모델을 생성하는 것은 취약점을 발견하기 위한 퍼징의 효율적인 면에서 한계성이 있다.

III. 파일 분석을 통한 자동화 데이터 모델 생성 메커니즘

파일 분석을 통한 자동화 데이터 모델 생성은 입력 파일을 분석하여 파일 포맷에 의존적인 데이터 모델을 자동으로 생성하는 방법이다. 한 개의 데이터 모델이 모든 샘플 파일을 충족할 필요 없이 자동 생성 기능을 통해 샘플에 의존적인 데이터 모델을 생성함으로써 파일 포맷을 분석하는 시간과 데이터 모델 생성을 위한 자원을 줄여 기존보다 효율적으로 퍼징을 수행하는 것이 가능하다. 또한 퍼징 수행을 통해 발생한 크래시를 이용하여 유효한 취약점이 발생하는 부분에 대한 변이 영역 설정을 통해 전체적인 퍼징 시간을 단축할 수 있다.

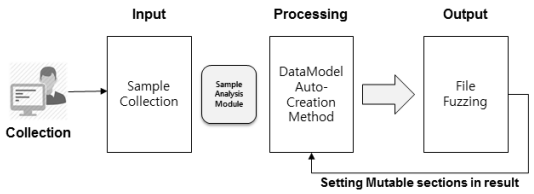


Fig.2. Vulnerability analysis process using auto-creation fuzzing

파일 포맷에 대한 분석을 통하여 퍼징을 수행하기 위한 플랫폼에 맞는 데이터 모델의 자동 생성을 목표로 하며, 본 논문에서는 퍼징을 수행하기 위한 플랫폼으로서 peach를 사용하였다. 파일 분석을 통한 자동화 데이터 모델 생성 메커니즘은 크게 두 가지 영역으로 나눌 수 있는데 각각 자동 데이터 모델 생성과 특정 영역에 특화된 변이 영역이다.

3.1 데이터 모델 자동 생성 메커니즘

자동 데이터 모델 생성은 그림 3과 같은 구조로 이루어진다. 샘플 파일을 로드하여 데이터를 스캔한 이후 주요 요소를 추출한다. 주요 요소는 토큰으로 사용할 수 있는 문자열 데이터나 고정적인 시그니처 등이 될 수 있다. 추출이 완료되면 데이터 타입을 확인하여 문자열일 경우 String 데이터 타입을 생성하고 나머지 데이터 형에 대해서는 NULL 문자의 존재 여부와 크기 비교를 통해 Number 데이터 타입 또는 Blob 데이터 타입을 생성한다. Number와 Blob 데이터 타입의 분류 기준은 4바이트와 8바이트를 읽어 각 바이트의 끝 부분에 NULL 문자의 존재 여부에 따라 결정된다. 이때 추출된 데이터의 위치를 고정적으로 사용하기 위해 Blob와 문자열 데이터 타입의 크기를 설정한다. Number 데이터 타입의 경우 4바이트와 8바이트로 고정적이다. 이후 데이터 모델에 대한 생성이 완료되면 생성된 데이터 모델을 이용하여 Peach를 통해 퍼징을 수행한다.

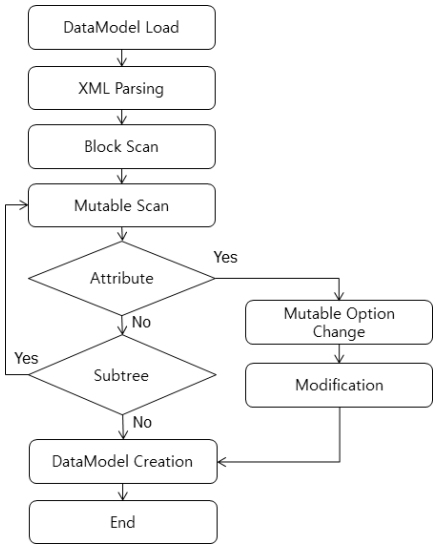


Fig.3. Datamodel auto-creation method

3.2 특정 영역에 특화된 변이 메커니즘

특정 영역에 특화된 변이 메커니즘은 그림 4와 같은 구조로 이루어지며, 퍼징을 수행하는 과정에서 입력 데이터를 사용하여 취약점이 발생할 가능성이 높은 특정 영역에 대한 변이를 수행하기 위한 데이터 모델을 생성하는 방법이다.

데이터 모델을 로드하여 XML 파싱을 수행하고 블록을 스캔하여 하위 노드에 데이터 타입의 변이 속성을 스캔한다. 속성을 발견하면 사용자의 설정에 따라 변이 설정을 변경하고 하위 모델을 스캔하여 모든 하위 노드에 대한 변이 설정을 동시에 수정한다.

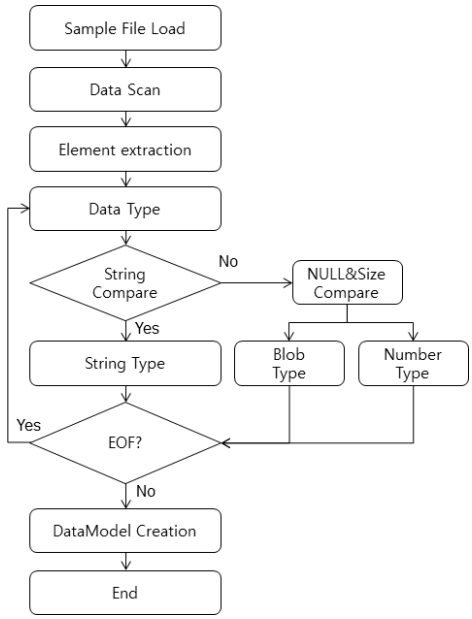


Fig.4. Mutation method for specific section

IV. 데이터 모델 자동 생성 프레임워크

본 논문에서는 데이터 입력 파일에 대한 분석을 통해서 스마트 퍼징을 위한 데이터 모델을 자동으로 생성하기 위한 프레임워크를 제안한다. 프레임워크는 변이 모듈, 데이터 모델 자동 생성 및 수정, 입력 샘플 파일 분석에 대한 기능을 포함하고 있다.

4.1 데이터 모델 자동 생성 프레임워크

본 논문에서 제안하는 데이터 모델 자동 생성 프레임워크는 그림 5와 같은 구조로 이루어진다.

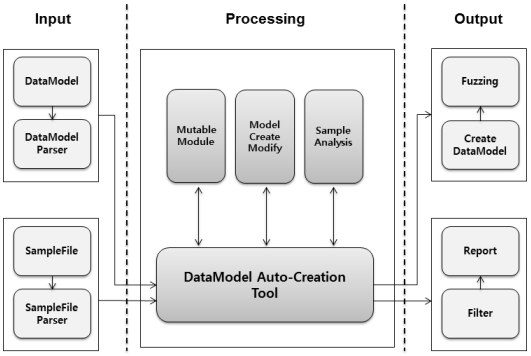


Fig.5. Datamodel auto-creation framework

데이터 모델 자동 생성 프레임워크는 기능에 따른 3가지 모듈이 존재한다. 변이성 모듈은 데이터 모델에서 변이성을 설정하고 적용하기 위한 모듈이며, 데이터 모델 생성 및 수정하기 위한 모듈과 샘플 파일을 분석하고 샘플에 맞는 데이터 모델을 생성하기 위한 모듈이 존재한다. 데이터 모델은 데이터 모델 파서를 통해 데이터 모델 생성기에 전달되며 변이성 모듈이나 데이터 모델 생성 및 수정 모듈을 통해 새로운 데이터 모델을 생성하고 이를 이용하여 퍼징을 수행한다. 샘플 파일의 경우 샘플 파일 파서를 통해 샘플 파일의 주요 정보를 추출하며 샘플 분석을 통해 데이터 모델을 생성하기 위한 요소를 선정한다. 이후 데이터 모델 자동 생성기를 통해 샘플에 특화된 데이터 모델을 생성하고 이를 이용하여 Peach를 통해 퍼징을 수행할 수 있다.

Peach 데이터 모델은 XML 파일 구조로 이루어져있으며, 데이터 모델, 상태 모델, 에이전트, 테스트에 대한 정보가 포함되어 있다. 기본적으로 XML에 대한 파싱이 수행되며, 각각의 데이터 모델에 대한 인식 기능이 포함되어 있다. 또한 입력 파일을 이용하여 데이터 모델을 자동 생성할 경우에도 마찬가지로 Peach 기반 XML 데이터 모델을 생성한다.

4.2 구현 결과 및 성능 평가

본 논문에서 제안한 데이터 모델 자동 생성 프레임워크를 구현하여 실험하였다. 구현한 프로그램의 실행 화면은 그림 6과 같다.

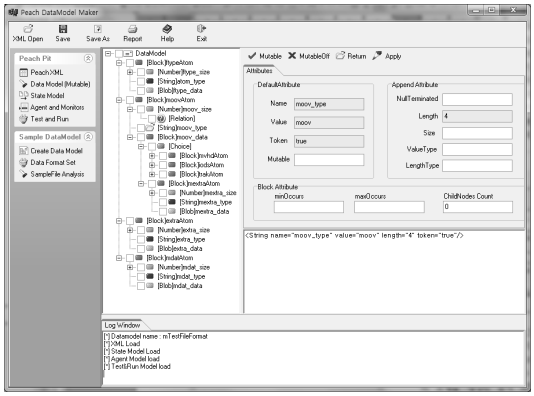


Fig.6. Datamodel auto-creation framework GUI

데이터 모델 자동 생성 프레임워크는 데이터 모델에 대한 뷰 기능을 제공하며, 입력 파일을 분석하여 퍼징에 최적화된 데이터 모델을 생성한다. 또한 특정 영역에 대한 변이를 수행하기 위한 변이성 영역 자동 생성 기능이 존재하며 직접 데이터 모델을 수정하는 것 또한 가능하다. 입력 데이터 파일을 분석하여 자동으로 데이터 모델을 생성하며, 특정 영역에 특화하여 변이를 수행하는 것이 가능하기 때문에 스마트 퍼징을 수행하는데 있어서 유용한 도구가 될 것으로 기대된다.

실험 환경은 다음과 같다. 총 9대의 i7-3770 @3.40Ghz PC를 이용하여 vmware를 통한 Windows XP 서비스팩3 환경에서 퍼징을 수행하였다. 총 103개의 입력 파일 샘플을 수집하여 이를 퍼징에 이용하였으며, 퍼징 실험은 48시간을 기준으로 진행되었다.

구현된 도구를 이용하여 A사 소프트웨어에 대한 퍼징을 수행한 결과 표 1과 같은 결과를 얻을 수 있었다. 기본 퍼징은 데이터 모델에서 파일에 대한 세부적인 데이터 타입을 명시하지 않고 단순히 데이터 모델만 생성하여 퍼징을 수행하였으며, 분석 퍼징의 경우 미리 입력 파일에 대한 분석을 선행하였으며, 이 결과를 바탕으로 데이터 모델을 만들어 퍼징을 수행한 결과이다. 따라서 기본 퍼징과 개발 도구의 경우 입력 파일에 대한 수동 분석 과정이 존재하지 않는다.

Table 1. Result of fuzz testing

Crash Type		Peach method		Frame work
		def.	anls.	auto.
Probably Exploitable	ReadAV	0	17	15
UnKnown	TaintedData Control	54	684	547
	ReadAV	2	15	15
	TaintedData Passed ToFunction	0	17	13
Total		56	733	590

퍼징 결과로 PROBABLY_EXPLOITABLE 크래시 가 발견되었으며 크래시 발생 횟수는 15회이다. Unknown 크래시는 3종류가 발견되었으며 크래시 발생횟수는 575회이다. 수동 분석 과정을 통해 데이터 모델을 생성한 분석 퍼징 과정과 비교했을 때 80.49%의 효율을 보이는 것을 확인할 수 있다. 반면, 기존 퍼징 도구에서 파일 포맷 분석을 통해 데이터 모델을 생성하지 않는 기본 퍼징의 경우 동일한 시간적 자원이 소모되었음에도 불구하고 7.63%의 효율을 보였다. 즉, 퍼징을 수행하는 부분에 있어서 각각의 샘플 파일에 대한 분석 과정을 생략할 수 있기 때문에 시간적인 부분이나 자원 소모 면에서 효율적이다 할 수 있다. 발견된 크래시의 경우 추가 분석을 통해 코드 실행 취약점 여부를 판별 할 수 있다.

V. 결 론

본 논문에서는 퍼징을 수행하기 위한 입력 파일의 분석을 통해 데이터 모델을 자동으로 생성하기 위한 프레임워크를 제안하였다. 스마트 퍼징을 위해서는 우선 퍼징을 수행하고자 하는 대상 소프트웨어에 대한 데이터 모델을 생성해야하며, 데이터 파일 및 소프트웨어 자체에 대한 분석이 필요하기 때문에 많은 자원이 소모된다. 따라서 효율적인 스마트 퍼징을 위해서 데이터 모델을 자동으로 생성하기 위한 방법이 필요하다. 이에 효율적인 퍼징 수행을 위해 데이터 입력 파일에 대한 분석을 통해서 스마트 퍼징을 위한 데이터 모델을 자동으로 생성하고, 특정 영역에 대한 변이를 수행하기 위한 기능을 추가함으로서, 분석자가 특정 부분에 특화된 퍼징을 수행할 수 있도록 하였다.

특히, 소프트웨어에 대한 입력 파일에서 퍼징 요소가 특정한 데이터 형태로 나누어진다는 점에 착안하여, 이러한 파일 분석을 통해 데이터 타입을 자동으로 생성하기 위한 방법을 연구하였다. 그 결과로 데이터 모델 자동 생성 프레임워크를 제안하였다. 이 프레임워크를 바탕으로 데이터 모델 생성 도구를 개발하였다. 또한 이 도구를 이용하여 피치 기반 퍼징을 수행한 결과 590개의 크래시를 발견하였고, 파일 포맷을 분석하고 데이터 모델을 생성하여 퍼징을 수행한 결과의 80% 효율을 보였다. 이를 통하여 소프트웨어 취약점 연구에 도움을 줄 것이며, 더 나아가 사이버 공격 및 범죄 예방에 기여할 수 있을 것이다.

References

[1] Min-ho Kim, Minsoo Kim, and Bong-nam Noh, "The Framework for Malware Analysis using Statistical Information of Registry," Journal of the KITS, 10(9), pp. 97-104, Sep. 2012.

[2] Symantec, "2013 Internet Security Threat Report, Volume 18," 2013.

[3] IBM, "IBM X-Force 2013 Mid-Year Trend and Risk Report," 2013.

[4] A. Takanen, J. DeMott, and C. Miller, Fuzzing for software security testing and quality assurance, Artech House Publishers, 2008.

[5] Zhao Zhang, Qiao-Yan Wen, and Wen Tang, "An Efficient Mutation-Based Fuzz Testing Approach for Detecting Flaws of Network Protocol", Computer Science & Service System International Conference, pp. 814-817, Aug. 2012.

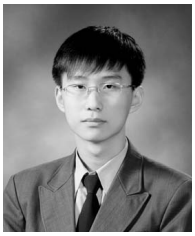
[6] Sofia Bekrar, Chaouki Bekrar, and Roland Groz, "A Taint Based Approach for Smart Fuzzing," Software Testing and Verification and Validation (ICST) IEEE Fifth International Conference, pp. 818-825, Apr. 2012.

[7] Michael Sutton, "filefuzz", http://packetstormsecurity.com/files/39626/FileFuzz.zip.html.

[8] ioctlfuzzer, https://code.google.com/p/ioctlfuzzer/.

〔9〕 Michael Eddington, Demystifying Fuzzers, 2009 Black Hat USA, Leviathan Security Group Inc, Jul. 2009.
〔10〕 Peach Fuzzing Platform, http://old.peachfuzzer.com/v3/DataModel.html.

〈저자소개〉



김 민 호 (Min-ho Kim) 학생회원
2011년 : 전남대학교 전자컴퓨터공학부(공학사)
2013년 : 전남대학교 정보보안협동과정(이학석사)
2013년 ~ 현재 : 전남대학교 정보보안협동과정 박사과정
〈관심분야〉 디지털 포렌식, 시스템 보안, 악성코드 탐지, 취약점 분석

박 성 빈 (Seongbin Park) 학생회원
2011년 : 전남대학교 전자컴퓨터공학부 공학사
2013년 : 전남대학교 정보보안협동과정 이학석사
2013년 ~ 현재 : 전남대학교 정보보안협동과정 박사과정
〈관심분야〉 취약점 분석, 악성코드 탐지, 시스템 보안

윤 지 노 (Ji-No Yun) 학생회원
2012년 : 전남대학교 전자컴퓨터공학부(공학사)
2012년 ~ 현재 : 전남대학교 정보보안협동과정 석사과정
〈관심분야〉 취약점 탐지, 시스템 보안, 악성코드

김 민 수 (Minsoo Kim) 종신회원
1993년 : 전남대학교 전산통계학과 (이학사)
1995년 : 전남대학교 전산통계학과 (이학석사)
2000년 : 전남대학교 전산통계학과 (이학박사)
2000년 ~ 2001년 : 한국인터넷진흥원 선임연구원
2001년 ~ 2004년 : 전남대학교 연구교수
2005년 ~ 현재 : 목포대학교 정보보호학과 부교수
〈관심분야〉 침입탐지, 디지털 포렌식스, 데이터마이닝, 악성코드 분석

노 봉 남 (Bong-Nam Noh) 종신회원
1987년 : 전남대학교 수학교육과 (이학사)
1982년 : KAIST 전산학과 (이학석사)
1994년 : 전북대학교 전산과 (이학박사)
1983년 ~ 현재 : 전남대학교 전자컴퓨터공학부 교수
2000년 ~ 현재 : 시스템보안연구센터 소장
〈관심분야〉 디지털 포렌식, 시스템 및 네트워크 보안, 정보사회와 사이버 윤리