Portfolio

이담호 (Damho Lee)

목차

Contents

소개

Who am I?

이담호 (Damho Lee)

1989.11.04
홍익대학교 컴퓨터공학과 박사 수료
슈어소프트테크 책임 (9년차)

Phone : 010-2734-5798
Email : damho1104@gmail.com
LinkedIn : https://www.linkedin.com/in/damho1104

Keyword

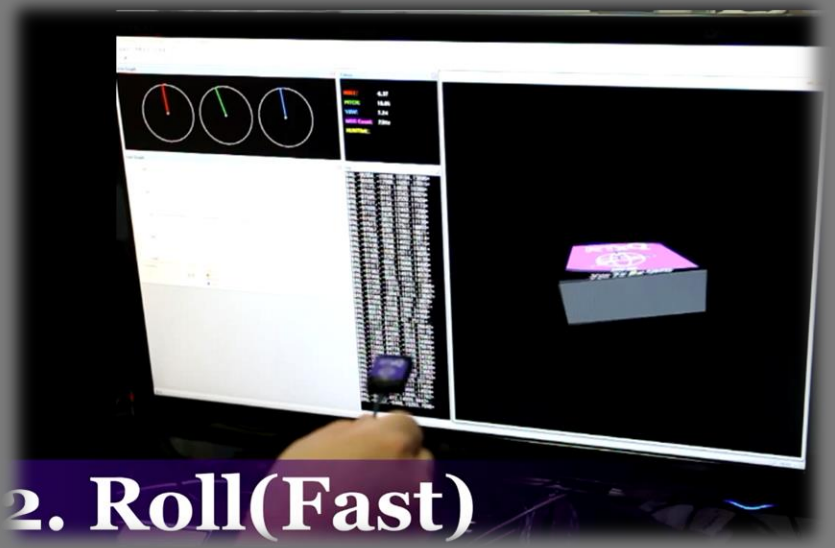
- C / C++, Java
- Python, Shell script, Docker
- Flask, SpringBoot
- LLVM, Clang, Static Instrumentation
- Program Analysis (Static / Dynamic)
- Control-flow Hijacking
- Control-flow Protection
- KLEE, Symbolic Execution

경력 사항

RUVA Tech
(2011 ~ 2012)

- **RUVA Tech 위치 참조 센서 연동 UI**
  - OpenGL 을 사용한 그래픽 표현
    - **오일러 각/쿼터니언 표현 방식** 사용
    - 현재의 센서 값을 그래픽으로 표현
    - 마우스와 연동하여 게임 테스트
  - 복수의 센서를 사용한 모션 캡쳐 시스템
    - 센서 값을 **인체 모형 그래픽**에 표현
- **결과물**
  - **논문 1건**

    위치참조 센서를 사용한 스트레칭 자세 교정 시스템, 한국정보과학회 학술발표논문집 2012.

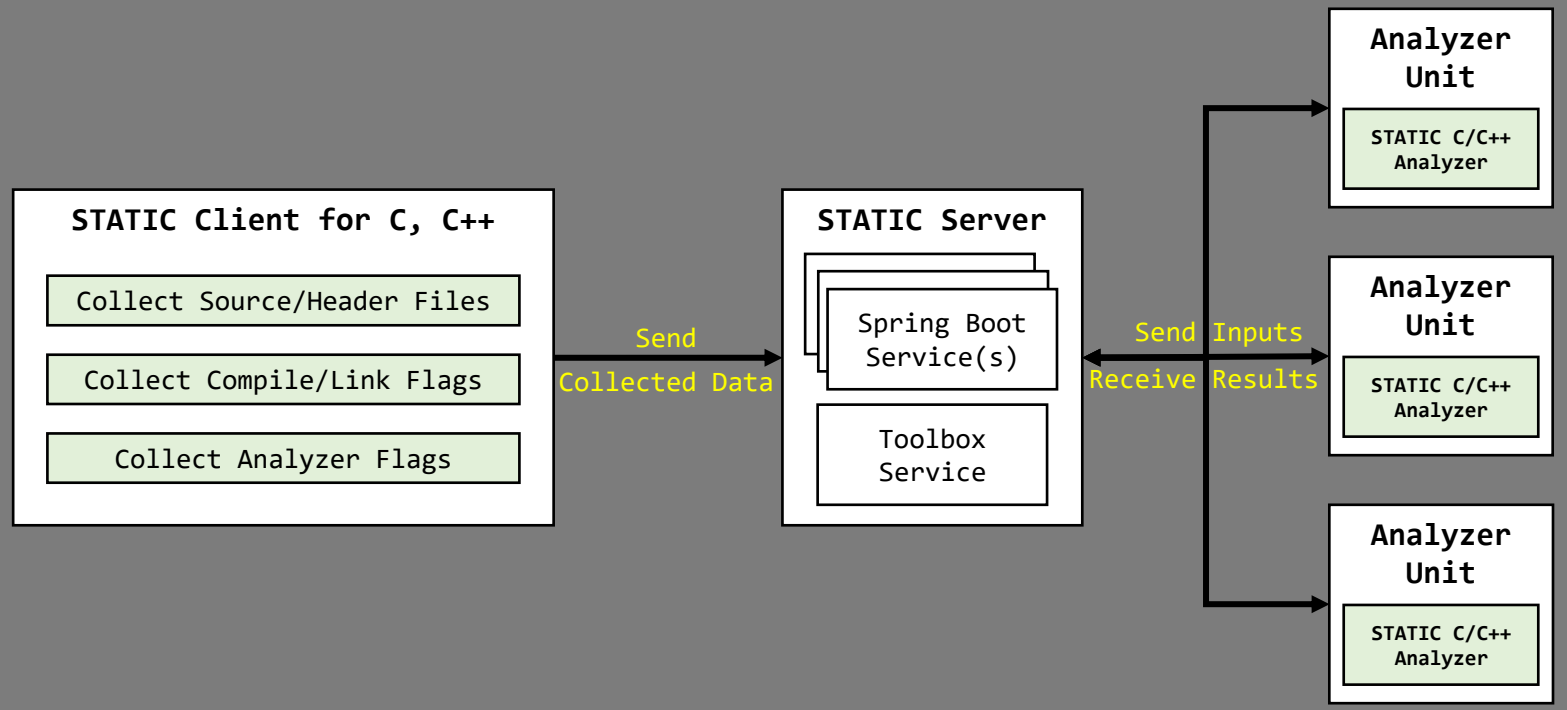


경력 사항

슈어소프트테크
(2019 ~)

- **STATIC Analysis Service**
  - Spring Boot 기반 웹 백엔드 서비스
  - 엔터프라이즈 급 웹 기반 정적 분석 도구 STATIC 의 핵심 서비스
    - 프로젝트 분석 정보 관리 및 분석 결과 모니터링
    - Analyzer Unit 연결 관리
    - Analyzer 패치 관리
    - JPPF 기반 TU/Link 단위 병렬 분석
  - 사용 기술
    - Java
    - Spring Boot
    - JPPF(Java Parallel Programming Framework)

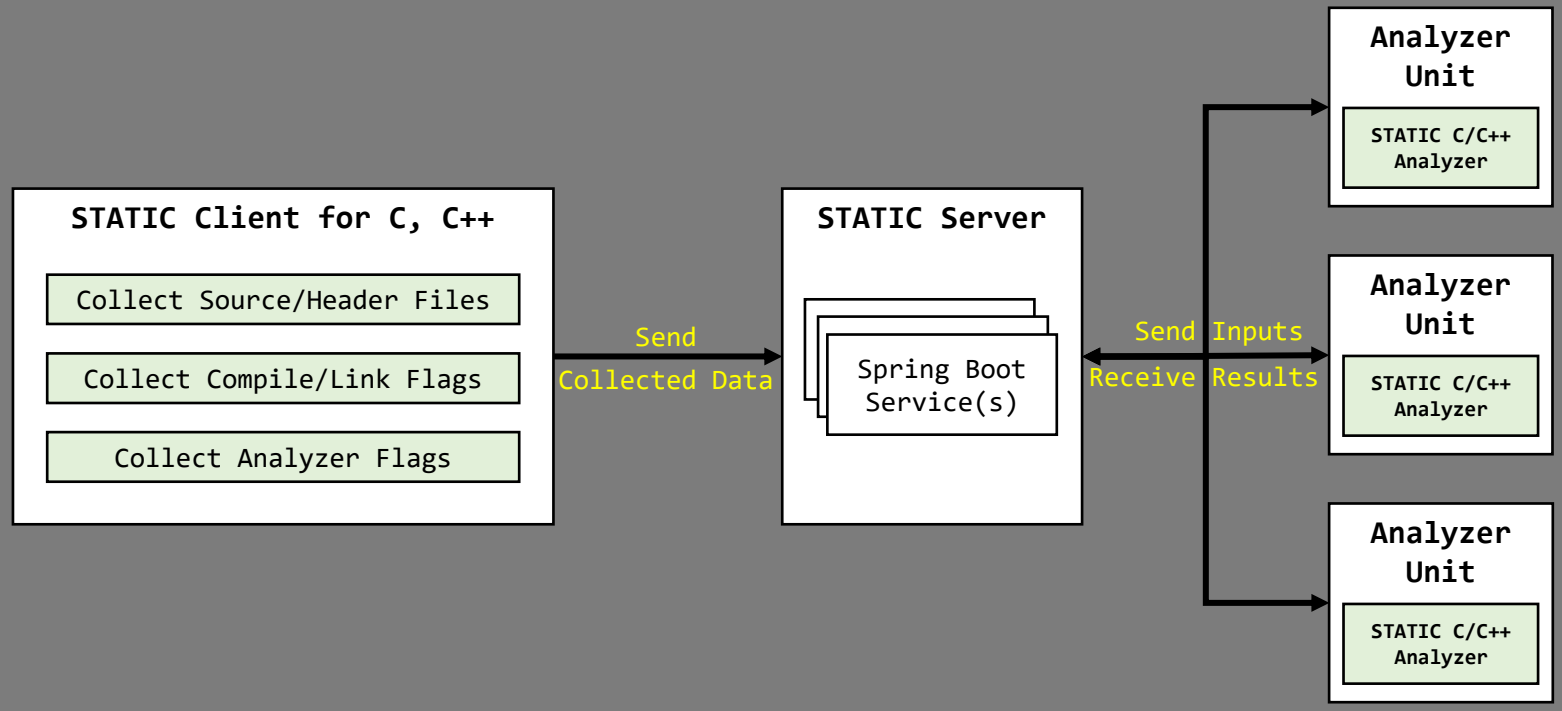
- **STATIC Toolbox Service**
  - Flask 기반 웹 백엔드 서비스
  - STATIC 을 이용하는 다수의 사용자들의 패키지 버전을 관리하는 서비스
  - 사용 기술
    - Python
    - Flask

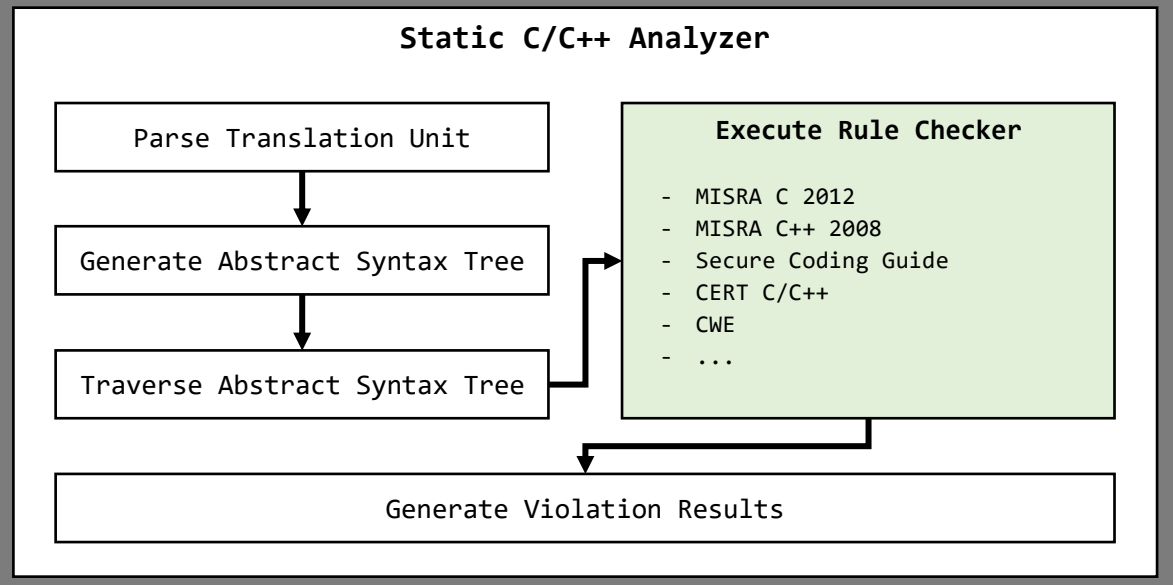


경력 사항

슈어소프트테크
(2019 ~)

- STATIC C/C++ Analyzer
  - Secure Coding 규칙 개발
    - CERT C/C++, CWE 규칙 기반 Suresoft Secure Coding 규칙 개발
    - Abstract Syntax Tree 기반 Checker 형태
- STATIC Client for C, C++
  - STATIC 서버에서 정적 분석을 위한 사용자 프로젝트 소스 및 빌드 정보 추출 도구 개발 및 유지보수
  - 사용 기술
    - DLL Injection
    - API Hooking
    - Compiler Preprocessor
    - Cross Compile Technique



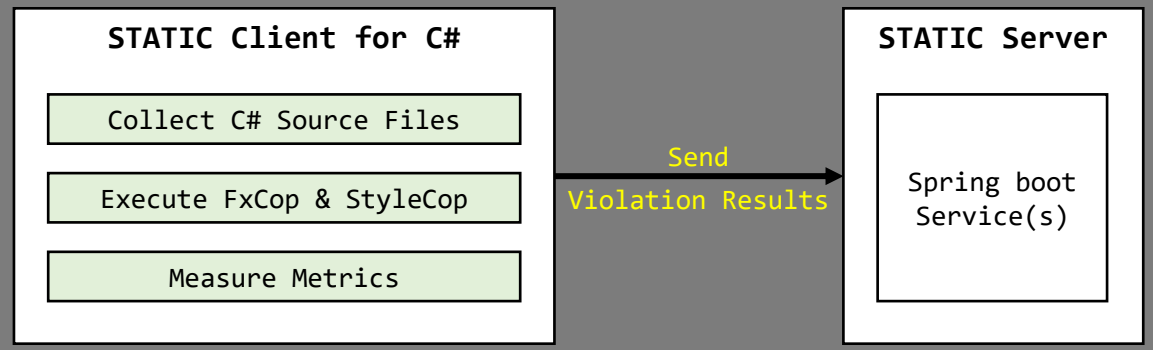


경력 사항

슈어소프트테크
(2019 ~)

- **STATIC Client for C#**
  - C# 프로젝트 소스 및 빌드 정보를 통해 정적 분석 수행하는 클라이언트 개발
    - Roslyn .Net Compiler Platform 라이브러리 사용한 소스 및 빌드 정보 추출
    - FxCop, StyleCop 연동
    - C# Metric Calculator 개발
      PLOC, LOC, Cyclomatic Complexity, Nesting Depth

- **Continuous Integration System**
  - STATIC 서비스 및 클라이언트 빌드, 테스트 및 배포 자동화 시스템, 사내 Code Sign Service 구축 및 관리
    - Framework
      Gitlab CI/CD
      Jenkins
      Flask

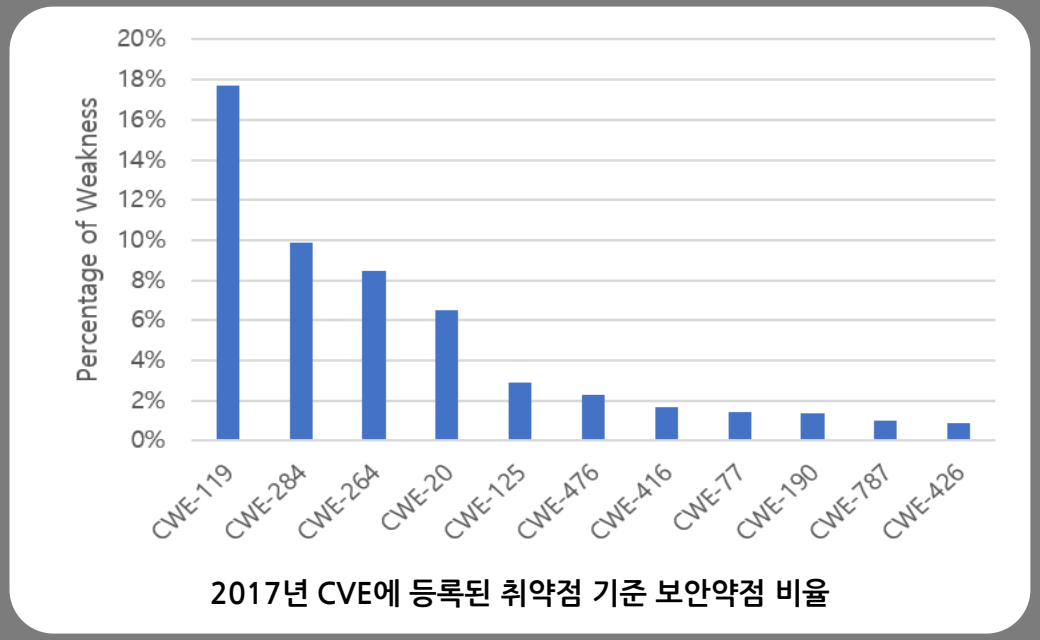


연구 배경

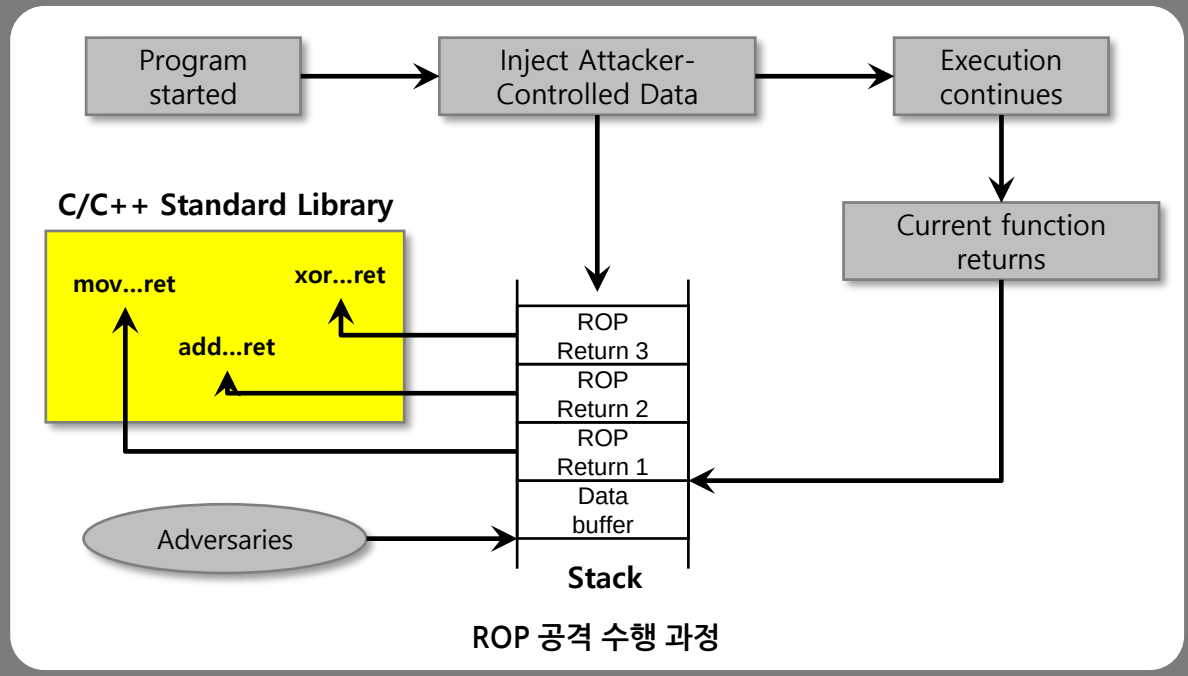
Background

- **시스템 개발 언어의 문제**
  - C/C++ 언어 문제
    - 약한 타입시스템
    - 포인터의 무제한적 사용
    - 통제되지 않는 타입 변환
    - 배열의 불완전 처리
  - 언어의 문제로 인해 보안약점 발생
    예: 버퍼 오버플로우

- **메모리 안전성 부재로 인한 공격 존재**
  - 메모리 안전성 부재로 인해 제어흐름 데이터 변조 가능
    - 제어흐름 데이터 예: 코드 포인터
  - 제어흐름 데이터 변조를 시작으로 프로그램 공격 수행
    - 코드 재사용 공격 (Code Reuse Attack)
    - Return-Oriented Programming (ROP)



2017년 CVE에 등록된 취약점 기준 보안약점 비율



ROP 공격 수행 과정

연구 분야 1

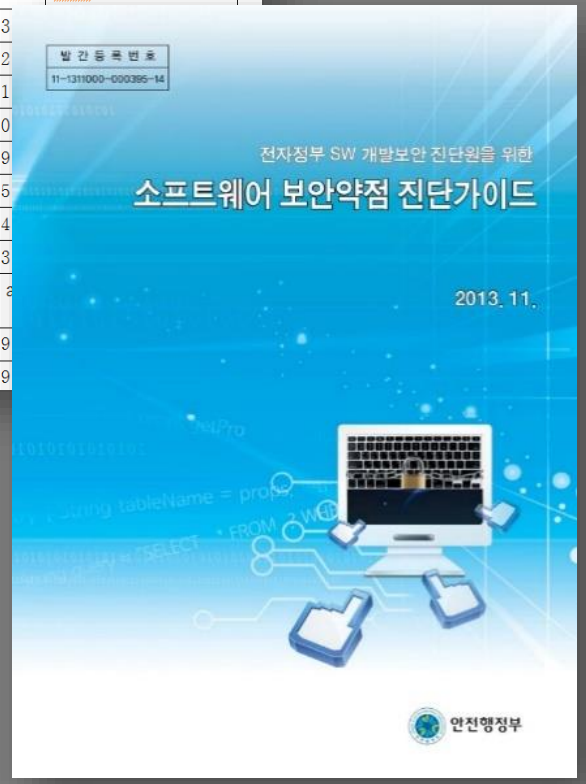
프로그램 분석

- **C/C++ 시큐어코딩 연구**
  - 47개의 보안약점에 대응되는 3가지 계층, 14가지 규칙 분류
  - 시큐어코딩 가이드를 통해 보안 지식 학습 가능

- **보안약점 진단 규칙 설계**
  - C++ 관련 보안약점 하위 85개 항목의 분석
  - 보안약점 탐지 규칙을 기술하기 위한 언어 정의

표 9. 안드로이드 시큐어 코딩 규칙 유형이 완화하는 CWE 종류 및 해당 CVE

규칙	CWE	CWE 설명 / CVE 리스트		
입력 값 검증				
	CWE-471	Modification of Assumed-Immutable Data (MAID)		
		CVE-2014-1977	CVE-2011-4864	CVE-2011-4702
		CVE-2013-2301	CVE-2011-4863	CVE-2011-4701
		CVE-2013-2300	CVE-2011-4773	
		CVE-2013-0720	CVE-2011-4772	
		CVE-2013-0719	CVE-2011-4771	
		CVE-2013-0718	CVE-2011-4770	
		CVE-2012-6422	CVE-2011-4769	
		CVE-2011-4867	CVE-2011-4705	
		CVE-2011-4866	CVE-2011-4704	
		CVE-2011-4865	CVE-2011-4703	
	CWE-22	Improper Limitation of a Pathname to a ('Path Traversal')		
		CVE-2014-1975	CVE-2014-1969	
		CVE-2014-1506	CVE-2014-0809	



연구 분야 1

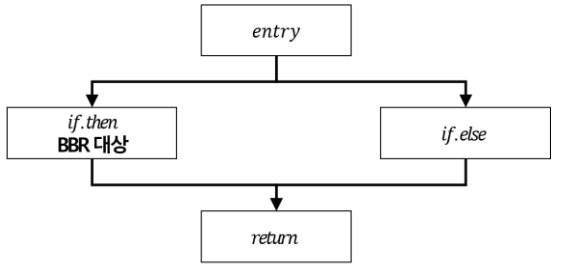
프로그램 분석

- Basic Block Restriction (BBR)
  - 정상적으로 종료되는 입력으로 동적 분석하여 **실행되지 않은 기본 블록들의 접근 제한**
  - **Basic Block Profiling**을 통한 실행되지 않은 기본 블록 수집
  - Framework
    - LLVM
    - Compiler-RT
- Unreachable Region Analysis based on Input Conditions
  - **미리 지정한 입력 조건에 따라 실행되지 않을 Basic Block 탐색 방법**
  - Framework
    - KLEE Symbolic Executor

Basic Block Restriction

```
1 float divide(float a, int b){
2    float c = 0.0;
3    if(b == 0)
4      printf("divide by 0\n");
5    else
6      c = a/b;
7    return c;
8 }
```

(a) 예제 코드

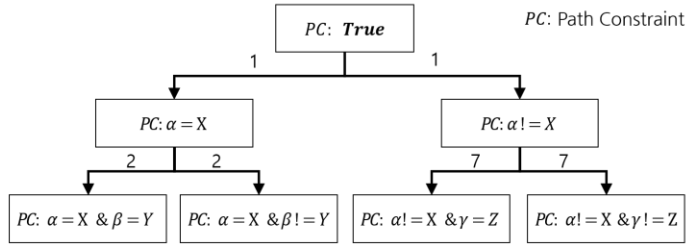


(b) 제어흐름 그래프

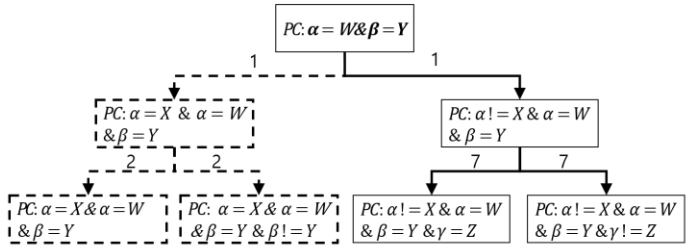
Unreachable Region Analysis based on Input Conditions

```
1  if(argv[1][0]=='X'){    //argv[1][0]:α
2    if(argv[2][0]=='Y')   //argv[2][0]:β
3      printf("arg2=Y\n");
4  }
5
6  else{
7    if(argv[3][0]=='Z')   //argv[3][0]:γ
8      printf("arg3=Z\n");
9  }
```

(a) 예제 코드



(b) 입력의 조건을 추가하지 않은 기호 실행 트리



(c) 입력의 조건을 추가한 기호 실행 트리

연구 분야 1

프로그램 분석

- **참여 프로젝트**
  - **모바일 기기 시스템 계층별 보안 강화 기법 연구**. 국방과학연구소(위탁과제). 2014.03~2015.02
  - **C++로 개발된 SW의 보안약점 탐지를 위한 규칙 설계 및 구현 기술 연구**. 국방과학연구소(위탁과제). 2016.03~2017.03
  - **프로그램 실행 흐름의 처방적 통제 및 응용**. 한국연구재단(교육부 지원과제). 2015.04~2018.10

- **연구 결과물**
  - **논문 3건, 수상 1건**

    C/C++ 프로그램을 위한 진단 정보와 분석 방법의 분류, KSCI 2017.

    계층화된 문맥 자유 문법을 사용한 입력 데이터 기술, KCC 2018 (우수논문상).

    입력 조건에 따라 도달하지 않는 영역을 기호 실행을 활용한 분석, KCC 2018.

Journal of The Korea Society of Computer and Information
Vol. 22 No. 3, pp. 81-88, March 2017

www.ksci.re.kr
https://doi.org/10.9708/jksci.2017.22.03.081

Classification of Diagnostic Information and Analysis Methods for Weaknesses in C/C++ Programs

Kyungsook Han*, Damho Lee**,

Abstract

In this paper, we classified the weaknesses of C/C++ ... diagnostic information produced at each stage of program ... which stages should be responsible for analyzing the ... frameworks for detecting typical weaknesses belonging to th... scheme. For the weaknesses that cannot be analyzed ... separated them as a group that are often detectable by the ... for instance, symbolic execution and abstract interpret... weaknesses, and diagnostic information accordingly, would ... static analyzers that minimizes false positives and negatives.

▸ Keyword :Security, Weakness, Static Analysis, Diagnostic

I. Introduction

입력 조건에 따라 도달하지 않는 영역을 기호 실행을 활용한 분석*

박영관○ 김태환 이담호 표창우
홍익대학교 컴퓨터공학과
{byg0102, evenstar, damho1104}@mail.hong...

Analyzing Unreachable Regions Using Symbolic ...

YoungGwan Park○ TaeHwan Kim Damh...
Department of Computer Engineering, ...

요 약

이 논문에서는 입력 조건에 따라 제어흐름이 도달하지 않는 영역이 발생할 수 있다. 이 영역은 특정 조건의 데이터를 처리할 때 조건 분기에 의해 항상 참이거나 거짓이어서 다른 한쪽은 실행되지 않을 때 발생한다. 이 영역을 제거하면 프로그램의 성능과 보안 수준을 향상시킬 수 있다. 성능이 개선되는 이유는 분기 명령과 해당 분기 명령이 포함하는 계산들을 생략할 수 있기 때문이다. 예를 들어 코드 조각 $S_1;S_2;$if$(C)S_3;S_4;$에 대하여 처리되는 입력 데이터가 조건 C에 대하여 항상 거짓으로 평가된다면 코드 조각을 $S_1;S_2;S_4;$로 변환해도 결과가 같게 된다.

제거되는 영역에 코드 재사용 공격[1]에 사용되는 가젯(gadget)이 있거나 비제어 데이터 공격[2]을 통해 프로그램의 특권을 탈취할 수 있는 코드 조각을 포함한다면, 프로그램 보안 수준이 향상된다. 위의 예에서 S_4에 setuid 시스템 호출이 있다면 외부에서 C가 참이 되게 하는 입력 값을 공급하여 권한 상승 공격을 시도해 볼 수 있는데, S_4가 제거되어 가능성이 차단된다.

제어흐름이 도달하지 않는 영역을 탐색하기 위한 방법은 정적 분석, 동적 분석, 기호 실행(symbolic execution)[3]을 활용할 수 있다. 정적 분석은 프로그램 실행 없이 분석할 수 있지만, 외부 입력에 따라 변하는 값은 정확하게 알 수 없다. 위의 예에서 C의 결과가 외부 입력에 따라 정해진다면, 정적 분석으로는 조건이 참인지 거짓인

* 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2015R1D1A1A01057711).

계층화된 문맥 자유 문법을 사용한 입력 데이터 기술*

서현지○ 김태환 이담호 표창우
홍익대학교 컴퓨터공학과
{siori1128, evenstar, damho1104}@mail.hongik.ac.kr, pyo@hongik.ac.kr

Input Data Description using Stratified Context-Free Grammars

Hyunji Seo○ Taehwan Kim Damho Lee Changwoo Pyo
Department of Computer Engineering, Hongik University

요 약

이 논문에서는 복수의 입력 장치로부터 들어오는 입력의 형태적 특성을 문맥 자유 문법 기반으로 표현하는 데이터 기술 언어(DDL)를 정의하였다. DDL은 여러 개의 독립된 문맥 자유 문법들을 계층적으로 연결할 수 있게 한다. 또한 DDL의 유효성과 입력 생성 성능 확인을 위해 단일 문맥 자유 문법 사용 방식, 기호 실행(Symbolic Execution) 방식의 입력 생성 체계와 비교 평가하였다. 평가 결과로 생성 시간 대비 기본 블록 실행 범위와 실행 범위를 확장하는 유효 입력 개수 모두 DDL을 사용한 결과가 다른 방식을 사용한 결과보다 높게 나타남을 확인할 수 있었다. 기본 블록 실행 범위 측면에서는 DDL을 사용한 결과가 기호 실행 방식보다 평균적으로 1.91배, 문맥 자유 문법 사용 방식보다는 1.03배 높게 나타났다. 유효 입력 개수는 기호 실행을 사용한 방식보다 2.68배, 문맥 자유 문법 사용 방식보다 1.29배 더 많은 생성 결과를 보이며, 입력 생성 시간은 기호 실행과 문맥 자유 문법 사용 방식 각각 146.6배, 1.9배 더 빠른 시간을 보였다.

1. 서 론

퍼징(fuzzing)은 자동 소프트웨어 테스팅과 취약성 분석에 널리 사용되는 방법으로서 동적 프로그램 분석에 사용되는 입력 데이터를 생성할 때에도 사용될 수 있다[1]. 퍼징에 의해 입력 데이터를 만들 때 아무런 제약 조건 없이 무작위로 생성하면 유효하지 않은 입력이 포함될 수 밖에 없다. 특히 '정상적인' 프로그램 실행 중에 동적 분석을 수행하기 위해서는 프로그램에 유효한 입력을 생성해야 한다. 프로그램이 수용 가능한 입력의 특성을 기술하여 입력 데이터 생성에 활용하면 유효하지 않은 데이터를 사용함으로 인해 발생하는 동적 분석의 비효율성을 방지할 수 있다.

문맥 자유 문법(Context-Free Grammar, 이하 CFG로 표시)은 입력 데이터의 구조 및 값의 범위와 같은 특성을 기술하기에 적합한 형식언어 체계이다. CFG의 생성 규칙을 하향식으로 적용하여 문장을 생성하는 기법을 사용하면 기술된 특성을 만족하는 임의의 입력 데이터를 생성할 수 있다[2]. CFG를 사용한 입력 생성은 자바스크립트 해석기의 보안약점을 분석할 때 사용할 입력의 자동 생성에 사용된 사례가 있다[3].

그러나 입력이 2개 이상의 파일 또는 다른 입력 장치로부터 공급될 때에는 하나의 문법 체계로 나타낼 수 없다. 예를 들어 Coreutils[4]의 unexpand는 파일 입력을 받아들여 탭 문자를 처리하는 프로그램으로, 'unexpand -t 12 file.txt'와 같은 명령선 입력이 가능하다. 입력인자 중 명령선에 직접 나타나는 '-t 12 file.txt'는 형태적 특성을 CFG로 기술할 수 있다. 그러나 세 번째 인자인 'file.txt' 파일의 내부 데이터의 형태적 특성은 같은 하나의 문법 내에서 나타낼 수 없다. unexpand의 입력을 나타내기 위해선 하나의 기술(description) 체계 안에서 여러 개의 CFG에 해당하는 입력 기술 체계가 계층적으로 연결되어야 한다.

이 논문에서는 복수의 입력 장치로부터 들어오는 입력의 형태적 특성을 CFG 기반으로 표현하는 데이터 기술 언어(Data Description Language, 이하 DDL로 표시)를 정의하였다. DDL은 여러 개의 독립된 CFG들을 계층적으로 연결할 수 있게 한다. 또한 DDL의 유효성과 입력 생성 성능 확인을 위해 CFG 방식과 기호 실행(symbolic execution)[5] 방식의 입력 생성 체계와 비교하였다. 비교 기준은 기본 블록 실행 범위(basic block coverage)[6]와 유효 입력 개수, 입력의 생성 시간으로 하였다.

이후 DDL의 정의는 2절에서 설명한다. DDL로 기술한 프로그램의 입력 데이터 생성 과정은 3절에서 설명한다. 생성된 입력 데이터의 성능은 4절에서 평가한다. 실험 결과를 토대로 한 결론은 5절에 있다.

2. 계층 문법의 구조

복수의 입력 장치로부터 들어오는 입력을 기술하려면 장치 별 CFG를 표기하고 이들의 연결 관계를 표시할 수 있어야 한다. 이를 위해 장치 이름을 표현하는 단말 기호는 장치로부터 들어오는 입력을 기술하는 CFG의 시작 기호와 연결되어야 한다. 연결에 사용되는 단말 기호를 연결자(connector)라 명한다.

연결자는 하나의 문법에서 단말 기호처럼 사용 가능하나 입력이 들어오는 장치 정보가 함께 표시된다는 특성이 있다. DDL로 기술한 unexpand 입력은 그림 1이 보여주고 있다. 1행에서 8행은 명령선 입력을 기술하는 CFG의 생성 규칙이며, 10행에서 11행은 명령선에 포함된 파일 입력의 내부 데이터를 기술하는 CFG의 생성 규칙에 해당한다. 두 문법은 1행에 있는 연결자에 의하여 연결

* 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2015R1D1A1A01057711).

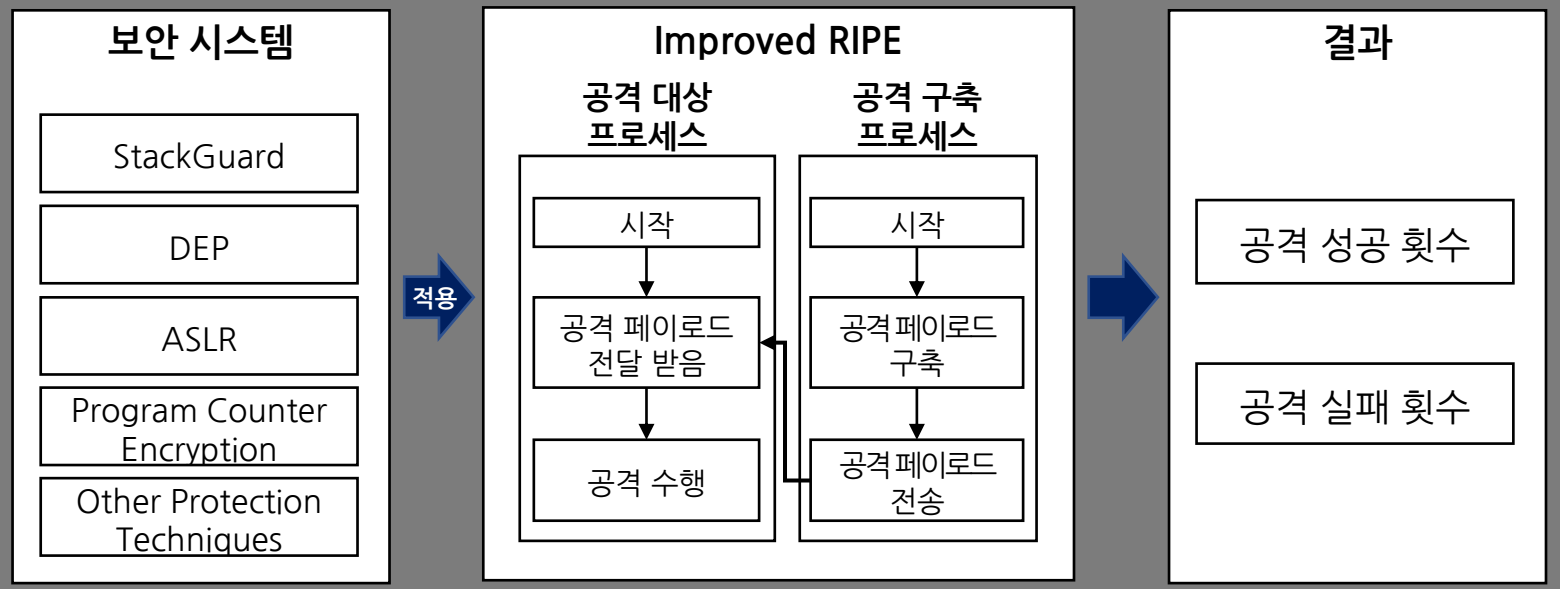
연구 분야 2-1

프로그램 보안(공격)

- **보안 시스템 검증 벤치마크 RIPE 확장**
  - OS 수준, 컴파일러 수준 보안 능력 검증
  - 888 가지 버퍼 오버플로우 기반 제어흐름 탈취 공격
    - RIPE 벤치마크 대비 **38가지 공격 추가**
    - **방어 검증 요소 추가**

```
865. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l bss -f strncat
866. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l bss -f sscanf
867. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l bss -f fscanf
868. [PASS] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l bss -f homebrew
869. [PASS] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f memcpy
870. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f strcpy
871. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f strncpy
872. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f sprintf
873. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f snprintf
874. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f strcat
875. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f strncat
876. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f sscanf
877. [FAIL] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f fscanf
878. [PASS] $./build/ripe_attack_generator -t indirect -i returnintolibc -c baseptr  -l data -f homebrew
879. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f memcpy
880. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f strcpy
881. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f strncpy
882. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f sprintf
883. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f snprintf
884. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f strcat
885. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f strncat
886. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f sscanf
887. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f fscanf
888. [FAIL] $./build/ripe_attack_generator -t direct -i rop -c ret  -l stack -f homebrew

=============== Summary ================
#PASS : 400
#FAIL : 488
#TOTAL: 888
========================================
crl@crl-VirtualBox:~/RIPE/RIPE-revised$
```



연구 분야 2-1

프로그램 보안(공격)

- **참여 프로젝트**
  - **모바일 기기 시스템 계층별 보안 강화 기법 연구**. 국방과학연구소(위탁과제). 2014.03~2015.02
  - **고품질 융합 소프트웨어 개발 지원 도구 연구**. 정보통신기술진흥센터(미래창조과학부 지원과제). 2014.01~2016.12

- **연구 결과물**
  - **논문 2건**

    실행시간 침입 방지 평가 프로그램 RIPE의 개선과 확장, KCC 2014.

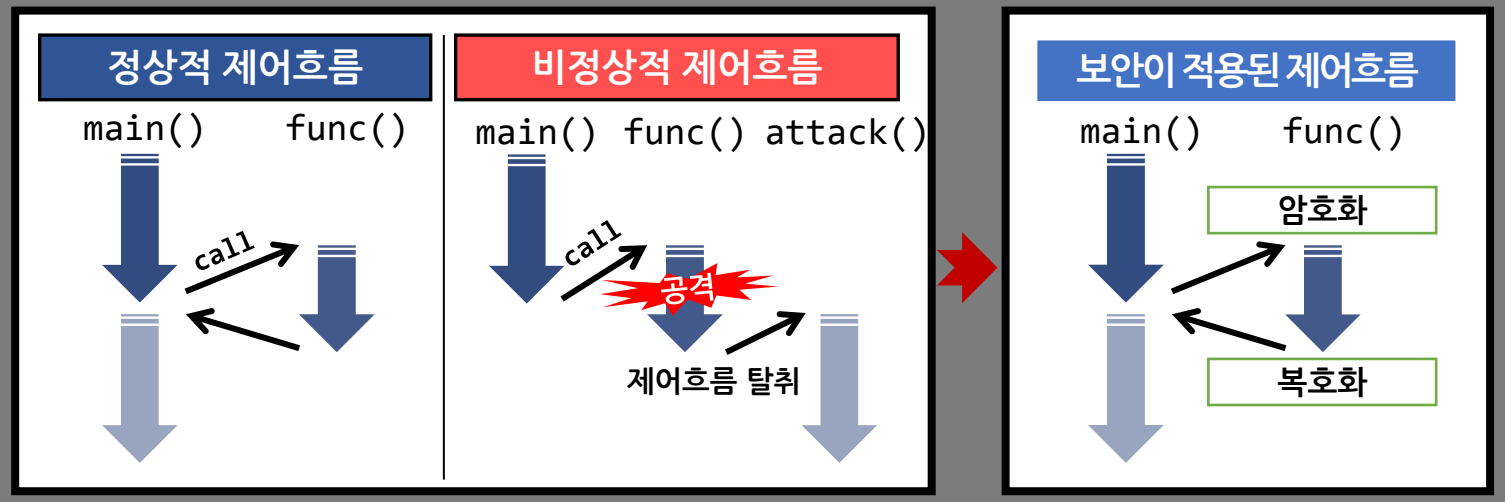
    실행시간 침입 방지 평가 프로그램(RIPE)의 개선, KIISE 2015.
  - **프로그램 등록 1건**

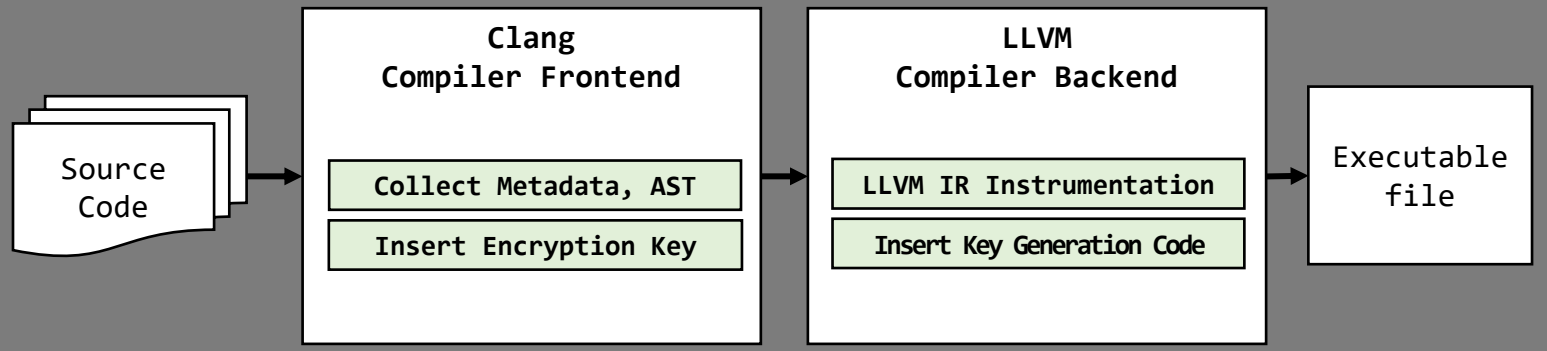
    개선된 실행시간 방어 기법 평가 프로그램, 한국저작권위원회 (C-2014-022811).

연구 분야 2-2

프로그램 보안(방어)

- Control-Flow Data Protection
  - 실행시간에 제어흐름 데이터 암호화, 제어흐름 데이터 사용시 복호화
  - 암호 키는 프로그램 실행시간에 변경
  - 맡은 업무
    - Collect Metadata in AST phase
    - LLVM IR Instrumentation(LLVM IR Pass)
  - Framework
    - Clang
    - LLVM
    - Compiler-RT
    - LibTooling

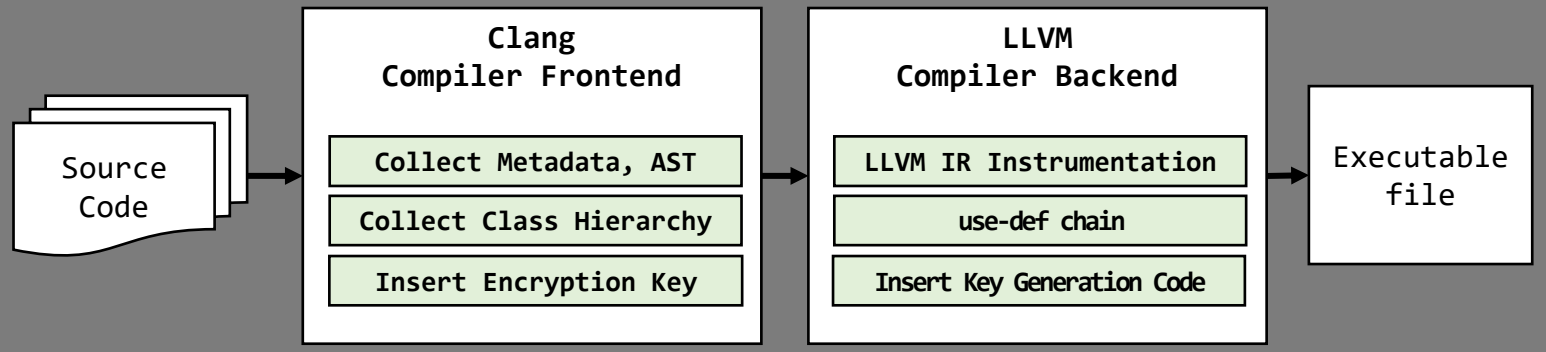




연구 분야 2-2

프로그램 보안(방어)

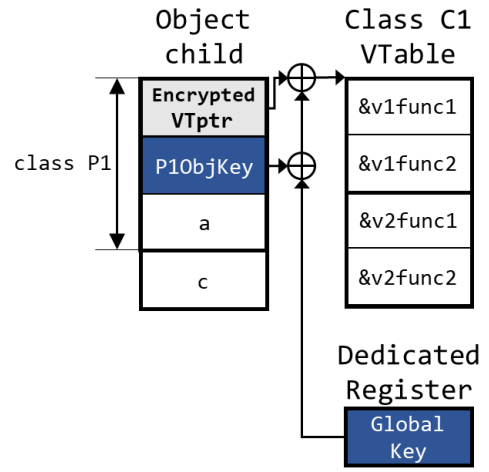
- **VTable Pointer Protection**
  - 실행시간에 VTable Pointer 생성 시 암호화, VTable Pointer 사용 시 복호화
  - 암호 키는 객체마다 unique 한 난수
  - 사용 기술
    - Clang
      - Source Code Rewriter
      - Class Hierarchy Collector
    - LLVM
      - LLVM IR Instrumentation
    - Compiler-RT(Runtime Library)
      - Pseudorandom Number Generator
      - Use Intel rdrand opcode
  - Framework
    - Clang
    - LLVM
    - Compiler-RT
    - LibTooling



Source Code

```
1.  class P1 {
2.      public:
3.      virtual void v1func1();
4.      virtual void v1func2();
5.      int a;
6.      unsigned long long P1ObjKey;
7.  };
8.  class C1 : public P1 {
9.      public:
10.     virtual void v2func1();
11.     virtual void v2func2();
12.     int c;
13. };
14. int main(int argc, char **argv){
15.     C1 *child = new C1();
16.     ...
17.     return 0;
18. }
```

Memory Layout



연구 분야 2-2

프로그램 보안(방어)

- **참여 프로젝트**
  - **프로그램 실행 흐름의 처방적 통제 및 응용.** 한국연구재단(교육부 지원과제). 2015.04~2018.10
  - **고품질 융합 소프트웨어 개발 지원 도구 연구.** 정보통신기술진흥센터(미래창조과학부 지원과제). 2013.04~2016.12

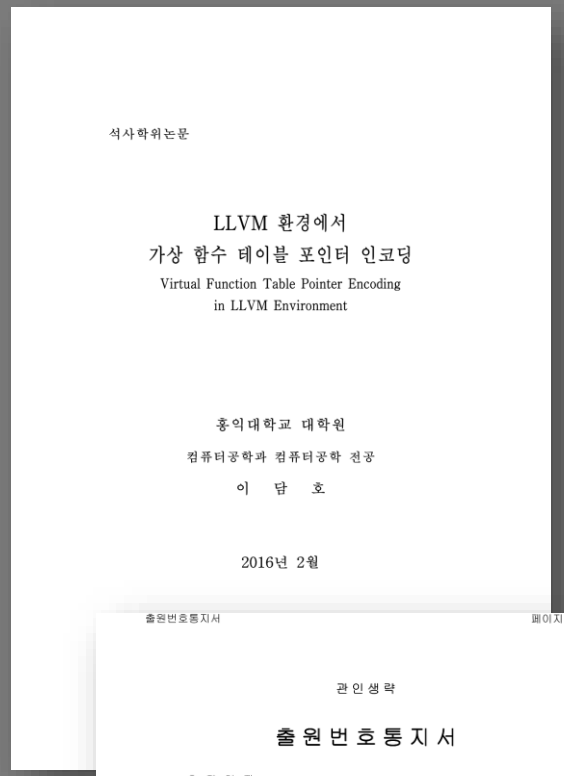
- **연구 결과물**
  - **논문 1건, 프로그램 등록 2건, 특허 출원 1건**

    LLVM 환경에서 가상 함수 테이블 포인터 인코딩, 홍익대학교, 2016.

    가상 테이블 포인터 인코딩 기법을 적용한 LLVM(엘엘브이엠) 컴파일러, 한국저작권위원회 (C-2015-025903).

    개선된 가상 함수 테이블 포인터 인코딩을 적용한 LLVM(엘엘브이엠) 컴파일러, 한국저작권위원회 (C-2016-022293).

    가상 함수 테이블 포인터 암호화 시스템 및 그 방법, 출원번호 10-2016-0183304.

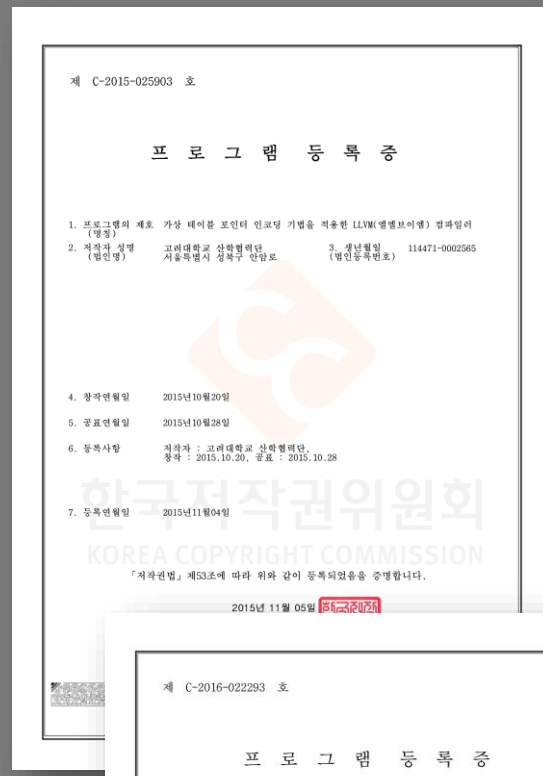


석사학위논문

LLVM 환경에서
가상 함수 테이블 포인터 인코딩
Virtual Function Table Pointer Encoding
in LLVM Environment

홍익대학교 대학원
컴퓨터공학과 컴퓨터공학 전공
이 담 호

2016년 2월



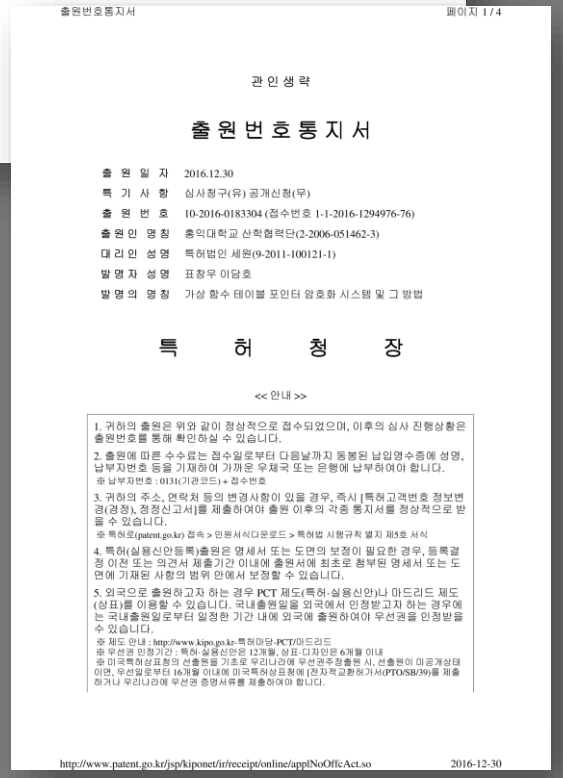
제 C-2015-025903 호

프 로 그 램 등 록 증

1. 프로그램의 제호 (명칭) 가상 테이블 포인터 인코딩 기법을 적용한 LLVM(엘엘브이엠) 컴파일러
2. 저작자 성명 (법인명) 고려대학교 산학협력단, 서울특별시 성북구 안암로 3. 생년월일 (법인등록번호) 114471-0002565
4. 창작연월일 2015년10월20일
5. 공표연월일 2015년10월28일
6. 등록사항 저작자 : 고려대학교 산학협력단, 창작 : 2015.10.20, 공표 : 2015.10.28
7. 등록연월일 2015년11월04일

「저작권법」 제53조에 따라 위와 같이 등록되었음을 증명합니다.

2015년 11월 05일



출원번호통지서 페이지 1 / 4

관인생략

출원번호통지서

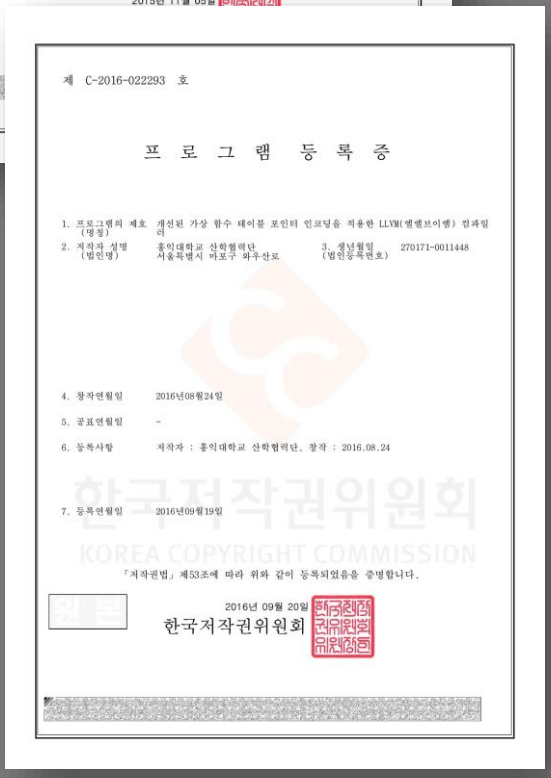
출원일자 2016.12.30
특기사항 심사청구(유) 공개신청(무)
출원번호 10-2016-0183304 (접수번호 1-1-2016-1294976-76)
출원인명칭 홍익대학교 산학협력단(2-2006-051462-3)
대리인성명 특허법인 세원(9-2011-100121-1)
발명자성명 표창우 이담호
발명의명칭 가상 함수 테이블 포인터 암호화 시스템 및 그 방법

특 허 청 장

<< 안내 >>

1. 귀하의 출원은 위와 같이 정상적으로 접수되었으며, 이후의 심사 진행상황은 출원번호를 통해 확인하실 수 있습니다.
2. 출원에 따른 수수료는 접수일로부터 다음날까지 동봉된 납입영수증에 성명, 납부자번호 등을 기재하여 가까운 우체국 또는 은행에 납부하여야 합니다.
※ 납부자번호 : 0131(기관코드) + 접수번호
3. 귀하의 주소, 연락처 등의 변경사항이 있을 경우, 즉시 [특허고객번호 정보변경(경정), 정정신고서]를 제출하여야 출원 이후의 각종 통지서를 정상적으로 받을 수 있습니다.
※ 특허로(patent.go.kr) 접속 > 민원서식다운로드 > 특허법 시행규칙 별지 제5호 서식
4. 특허(실용신안등록)출원은 명세서 또는 도면의 보정이 필요한 경우, 등록결정 이전 또는 의견서 제출기간 이내에 출원서에 최초로 첨부된 명세서 또는 도면에 기재된 사항의 범위 안에서 보정할 수 있습니다.
5. 외국으로 출원하고자 하는 경우 PCT 제도(특허·실용신안)나 마드리드 제도(상표)를 이용할 수 있습니다. 국내출원일을 외국에서 인정받고자 하는 경우에는 국내출원일로부터 일정한 기간 내에 외국에 출원하여야 우선권을 인정받을 수 있습니다.
※ 제도 안내 : http://www.kipo.go.kr-특허마당-PCT/마드리드
※ 우선권 인정기간 : 특허·실용신안은 12개월, 상표·디자인은 6개월 이내
※ 미국특허상표청의 선출원을 기초로 우리나라에 우선권주장출원 시, 선출원이 미공개상태이면, 우선일로부터 16개월 이내에 미국특허상표청에 [전자적교환허가서(PTO/SB/39)]를 제출하거나 우리나라에 우선권 증명서류를 제출하여야 합니다.

http://www.patent.go.kr/jsp/kiponet/ir/receipt/online/applNoOffcAct.so 2016-12-30



제 C-2016-022293 호

프 로 그 램 등 록 증

1. 프로그램의 제호 (명칭) 개선된 가상 함수 테이블 포인터 인코딩을 적용한 LLVM(엘엘브이엠) 컴파일러
2. 저작자 성명 (법인명) 홍익대학교 산학협력단, 서울특별시 마포구 와우산로 3. 생년월일 (법인등록번호) 270171-0011448
4. 창작연월일 2016년08월24일
5. 공표연월일 -
6. 등록사항 저작자 : 홍익대학교 산학협력단, 창작 : 2016.08.24
7. 등록연월일 2016년09월19일

「저작권법」 제53조에 따라 위와 같이 등록되었음을 증명합니다.

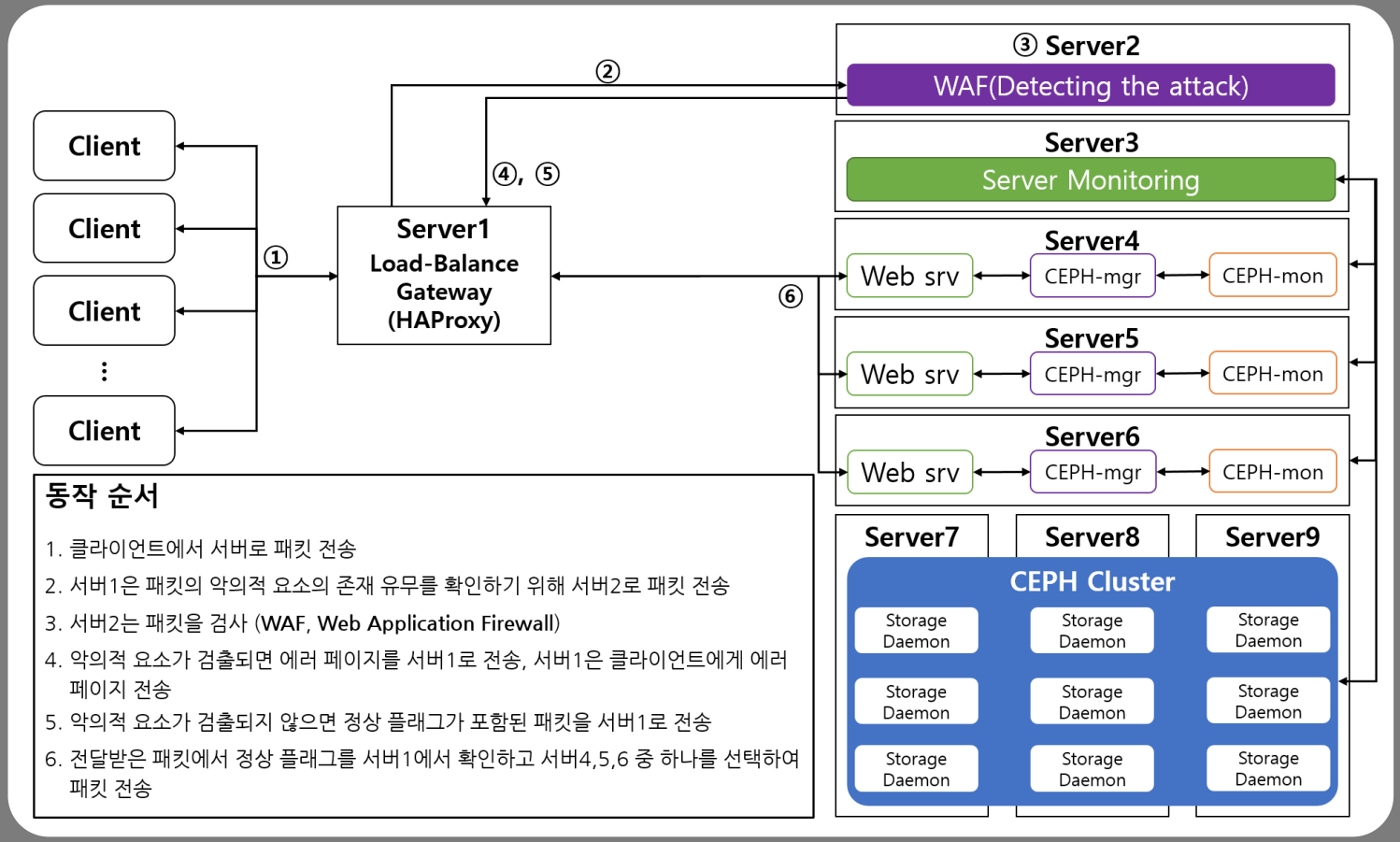
2016년 09월 20일

한국저작권위원회

연구 분야 3

빅데이터

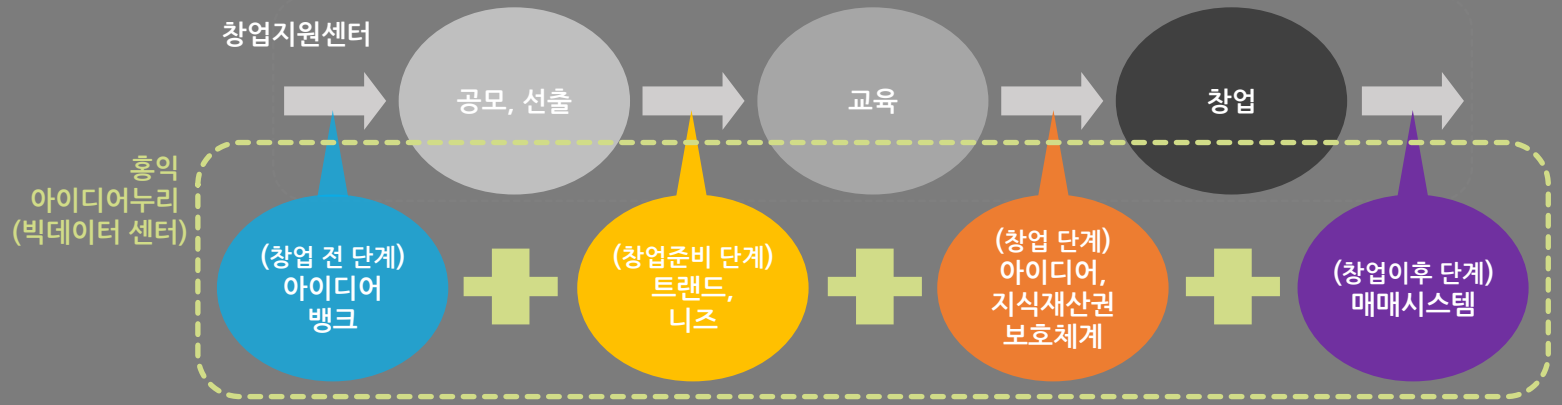
- **빅데이터 저장을 위한 서버 구축**
  - Distributed Object Storage System
  - Compatible Amazon S3 Protocol 사용
  - Framework
    - CEPH
    - HAProxy (Load-Balance Gateway)

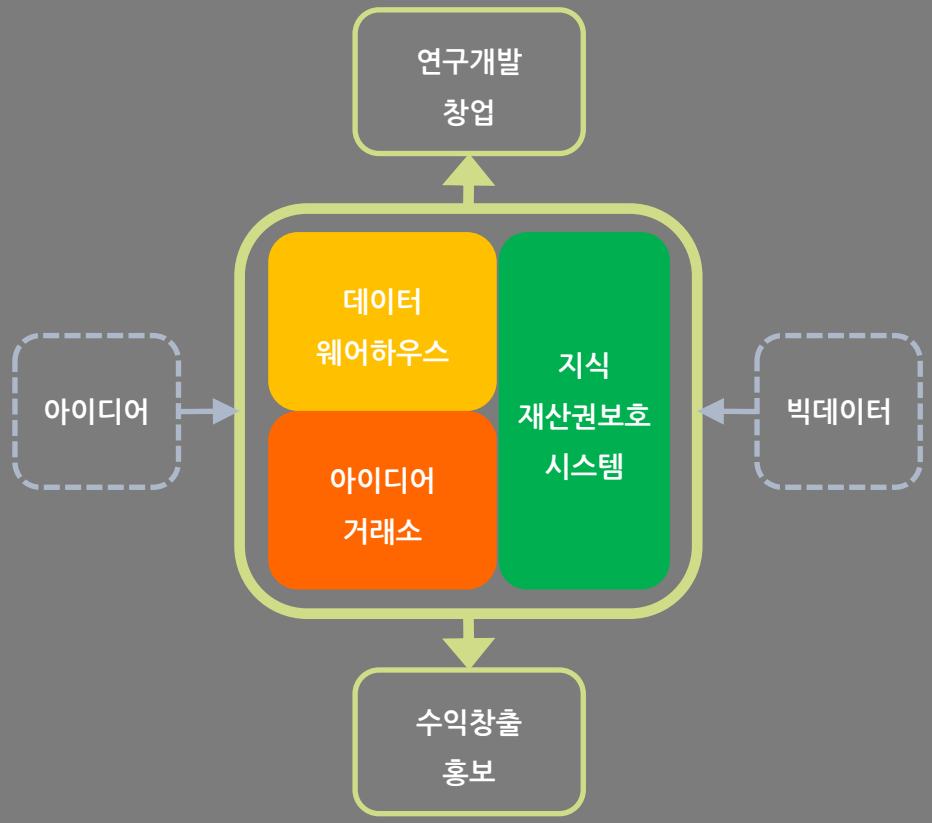


연구 분야 3

빅데이터

- **참여 프로젝트**
  - **상상력을 디자인하는 홍대 캠퍼스타운 조성 사업**. 서울특별시 캠퍼스타운 조성사업. 2017.05.01 ~ 현재





관심 분야 1

머신러닝

- **종목별 연관 단어를 활용한 주가 예측**
  - 3년간(2012~2014) 주간별 뉴스 데이터를 수집 후 주식 종목별 연관 단어 추출
  - 연관 단어 유사도와 주가상승률를 매핑 후 클러스터링
  - 클러스터링된 단어가 2015년도 뉴스에 나오는 경우 해당 종목의 주가상승률의 +/- 예측
  - **결과**

    10%의 연관 단어가 포함

    주가상승률 예측 정확도: **12.8%**
  - Framework

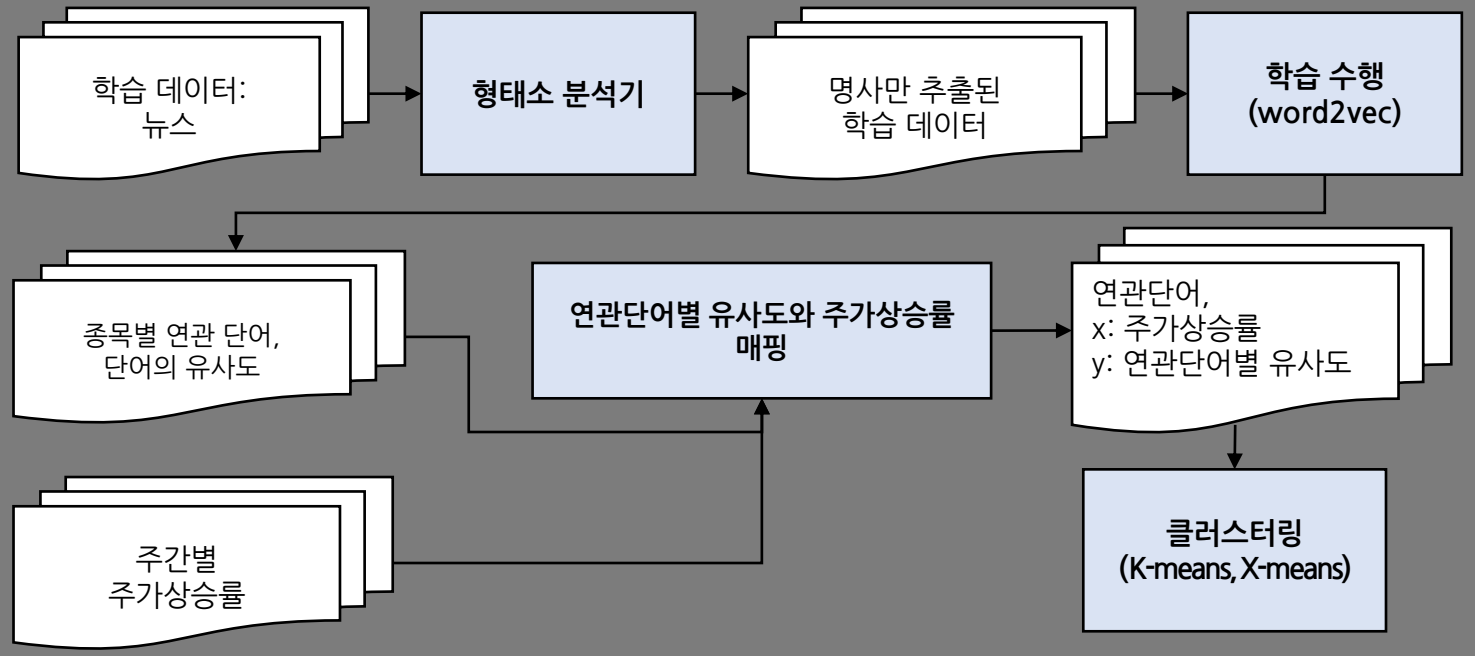
    N2H4 인터넷 뉴스 크롤러

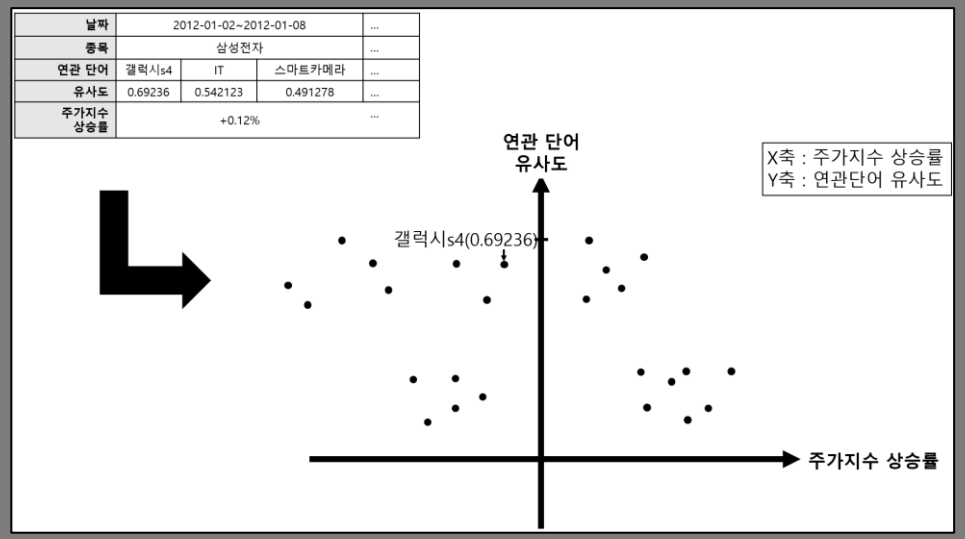
    Google Finance

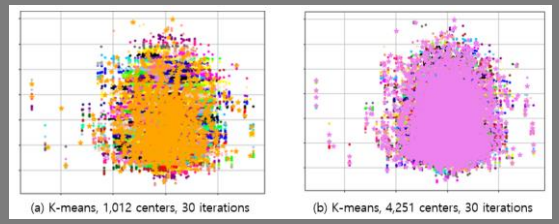
    koNLPy 형태소 분석기

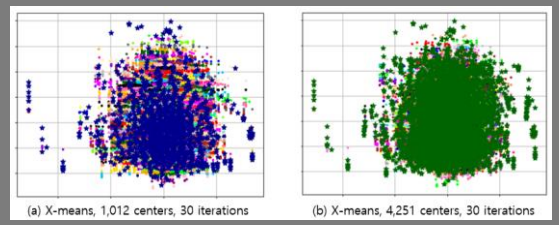
    gensim(Python Library)

    word2vec 알고리즘









교육 이수

고품질 융합소프트웨어 주말학교

- **“고품질 융합 소프트웨어 개발 지원 도구 연구”** 프로젝트에서 개최
- 기간: **2014.03 ~ 2016.01**
- 서로 다른 분야의 기초 과정 교육 이수
  - 프로그램 정적 분석
  - 프로그램 테스팅
  - 프로그램 보안 취약점 탐지 방안

교육 이수

Completion Kauffman FastTrac

- **“고품질 융합 소프트웨어 개발 지원 도구 연구”** 프로젝트에서 진행한 기업가 정신 교육
- 기간: **2014.07.14 ~ 16**
- 기업을 확장하거나 창업을 하는데 있어 필요한 기업가 정신을 전문가에게 지도 및 검증
- **실적**
  - **특허 1건 출원**
    - **리뷰 등록 서버 및 리뷰 등록 방법**, 2014.

KAUFFMAN FASTTRAC

KEF (재)한국청년기업가정신재단 Korea Entrepreneurship Foundation

Certificate of Completion
Kauffman FastTrac®

Presented To:
Damho Lee

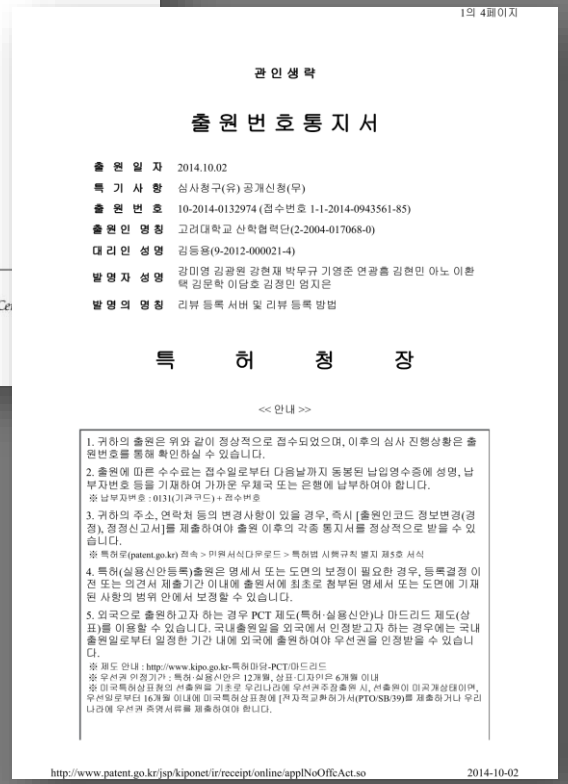
For successfully completing FastTrac® Planning the Entrepreneurial Venture ™ Program

Kauffman FastTrac® - KEF : No. 040

Presented on 16, July, 2014

yuyonho
Ph.D Yon Ho Yu, Facilitator

Pyoung Hee Choi
Managing Director of Venture CEO Mentoring Ce



1의 4페이지

관인생략

출원번호통지서

출원일자	2014.10.02
특기사항	심사청구(유) 공개신청(무)
출원번호	10-2014-0132974 (접수번호 1-1-2014-0943561-85)
출원인 명칭	고려대학교 산학협력단(2-2004-017068-0)
대리인 성명	김등용(9-2012-000021-4)
발명자 성명	강미영 김광원 강현재 박무규 기영준 연광흠 김현민 아노 이환택 김문학 이담호 김정민 엄지은
발명의 명칭	리뷰 등록 서버 및 리뷰 등록 방법

특 허 청 장

<< 안내 >>

1. 귀하의 출원은 위와 같이 정상적으로 접수되었으며, 이후의 심사 진행상황은 출원번호를 통해 확인하실 수 있습니다.
2. 출원에 따른 수수료는 접수일로부터 다음날까지 동봉된 납입영수증에 성명, 납부자번호 등을 기재하여 가까운 우체국 또는 은행에 납부하여야 합니다.
※ 납부자번호 : 0131(기관코드)+접수번호
3. 귀하의 주소, 연락처 등의 변경사항이 있을 경우, 즉시 [출원인코드 정보변경(경정), 정정신고서]를 제출하여야 출원 이후의 각종 통지서를 정상적으로 받을 수 있습니다.
※ 특허로(patent.go.kr) 접속 > 민원서식다운로드 > 특허법 시행규칙 별지 제5호 서식
4. 특허(실용신안등록)출원은 명세서 또는 도면의 보정이 필요한 경우, 등록결정 이전 또는 의견서 제출기간 이내에 출원서에 최초로 첨부된 명세서 또는 도면에 기재된 사항의 범위 안에서 보정할 수 있습니다.
5. 외국으로 출원하고자 하는 경우 PCT 제도(특허·실용신안)나 마드리드 제도(상표)를 이용할 수 있습니다. 국내출원일을 외국에서 인정받고자 하는 경우에는 국내출원일로부터 일정한 기간 내에 외국에 출원하여야 우선권을 인정받을 수 있습니다.
※ 제도 안내 : http://www.kipo.go.kr-특허마당-PCT/마드리드
※ 우선권 인정기간 : 특허·실용신안은 12개월, 상표·디자인은 6개월 이내
※ 미국특허상표청의 선출원을 기초로 우리나라에 우선권주장출원 시, 선출원이 미공개상태이며, 우선일로부터 16개월 이내에 미국특허상표청에 [전자적교환허가서(PTO/SB/39)]를 제출하거나 우리나라에 우선권 증명서류를 제출하여야 합니다.

http://www.patent.go.kr/jsp/kiponet/ir/receipt/online/applNoOffcAct.so 2014-10-02

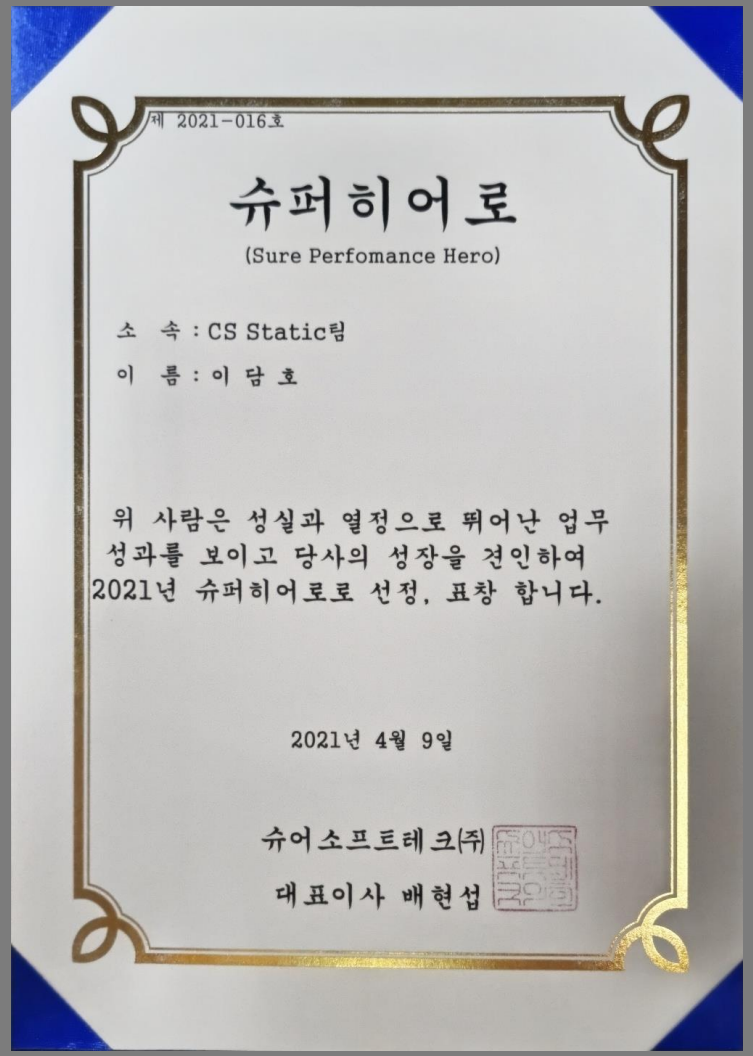
교육 이수

Oracle Certified Professional

- **Oracle Database 10g Administrator** 교육 이수
- 기간: **2013.09 ~ 2013.10**

수상 경력

- 슈어소프트테크 슈퍼히어로
  - 2년 연속 수상
    - 2022
    - 2021



제 2021-016호

슈퍼히어로

(Sure Perfomance Hero)

소 속 : CS Static팀

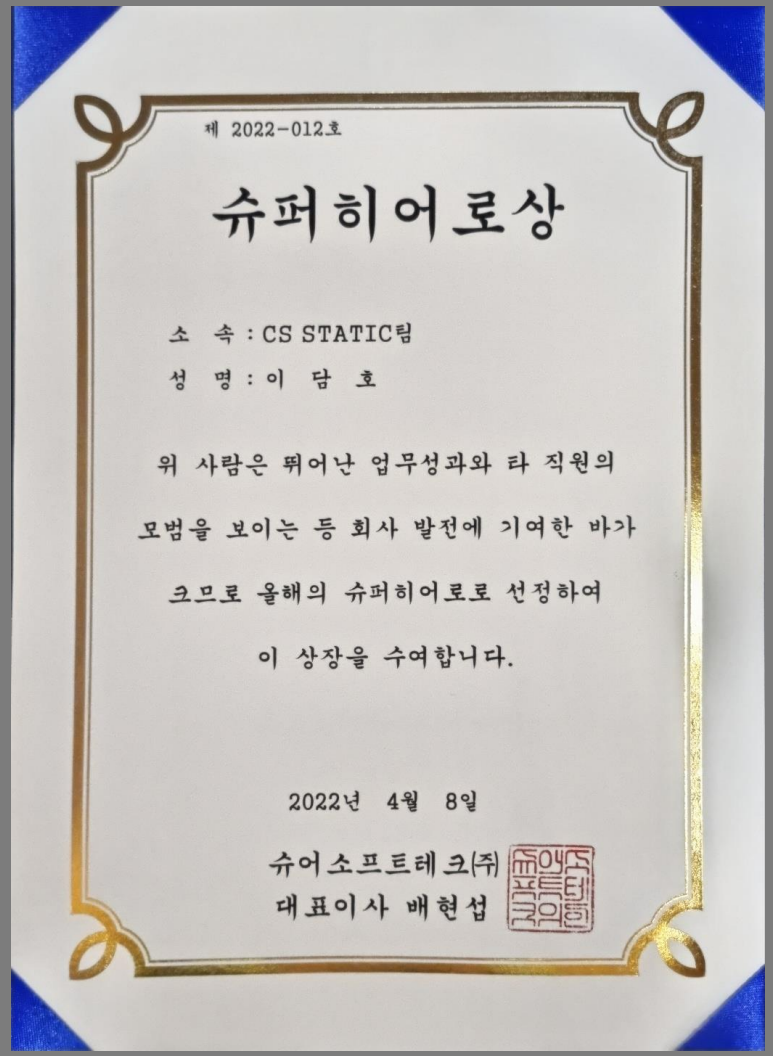
이 름 : 이 담 호

위 사람은 성실과 열정으로 뛰어난 업무 성과를 보이고 당사의 성장을 견인하여 2021년 슈퍼히어로로 선정, 표창 합니다.

2021년 4월 9일

슈어소프트테크㈜

대표이사 배현섭



제 2022-012호

슈퍼히어로상

소 속 : CS STATIC팀

성 명 : 이 담 호

위 사람은 뛰어난 업무성과와 타 직원의 모범을 보이는 등 회사 발전에 기여한 바가 크므로 올해의 슈퍼히어로로 선정하여 이 상장을 수여합니다.

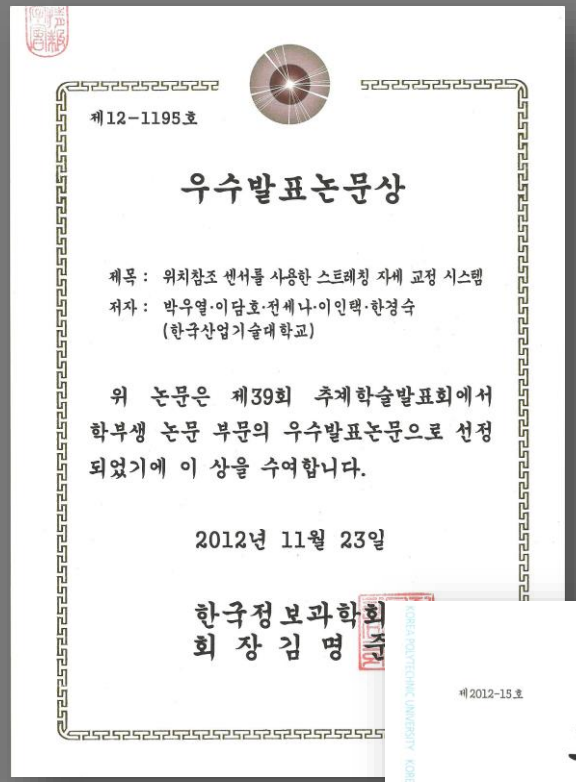
2022년 4월 8일

슈어소프트테크㈜

대표이사 배현섭

수상 경력

- 정보과학회 우수논문상 1건
- 정보과학회 우수발표논문상 1건
- 한국산업기술대학교 소프트웨어경진대회 도전상 1건



제12-1195호

우수발표논문상

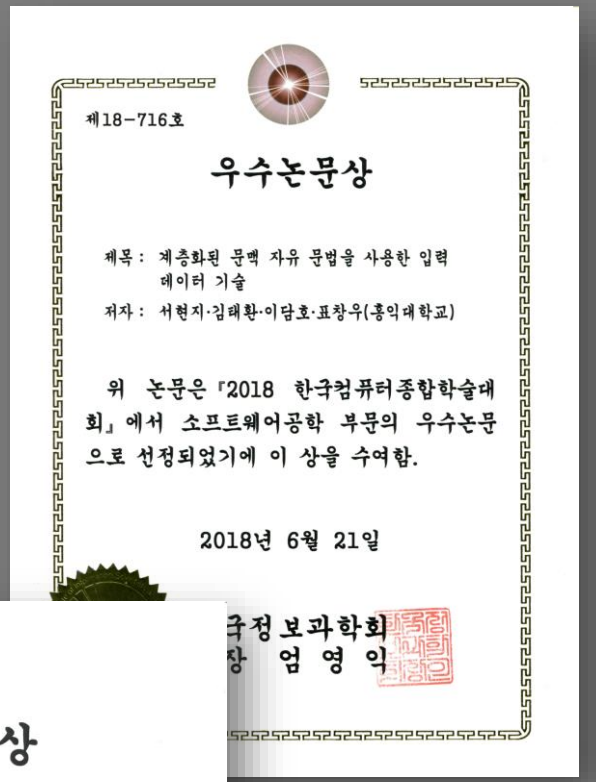
제목 : 위치참조 센서를 사용한 스트레칭 자세 교정 시스템

저자 : 박우열·이담호·전세나·이인택·한경숙
(한국산업기술대학교)

위 논문은 제39회 추계학술발표회에서 학부생 논문 부문의 우수발표논문으로 선정되었기에 이 상을 수여합니다.

2012년 11월 23일

한국정보과학회
회 장 김 명 준



제18-716호

우수논문상

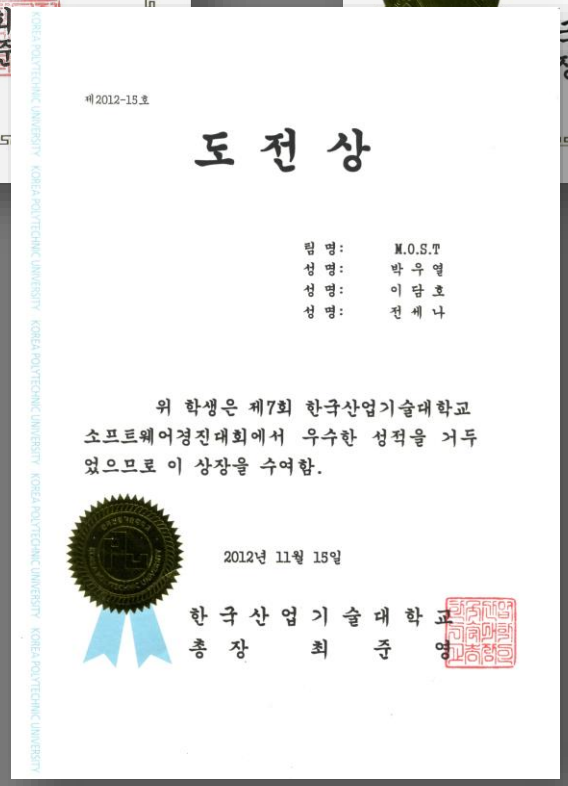
제목 : 계층화된 문맥 자유 문법을 사용한 입력 데이터 기술

저자 : 서현지·김태환·이담호·표창우(홍익대학교)

위 논문은 『2018 한국컴퓨터종합학술대회』에서 소프트웨어공학 부문의 우수논문으로 선정되었기에 이 상을 수여함.

2018년 6월 21일

한국정보과학회
장 엄 영 익



제2012-15호

도 전 상

팀 명: M.O.S.T
성 명: 박 우 열
성 명: 이 담 호
성 명: 전 세 나

위 학생은 제7회 한국산업기술대학교 소프트웨어경진대회에서 우수한 성적을 거두었으므로 이 상장을 수여함.

2012년 11월 15일

한 국 산 업 기 술 대 학 교
총 장 최 준 영