

Portfolio

이담호 (Damho Lee)

목차

Contents

소개

경력 사항

연구 배경

연구 분야

1. 프로그램 분석
2. 프로그램 보안 (공격 및 방어)
3. 빅데이터

관심 분야

1. 머신러닝
2. 홈 네트워크

교육 이수

수상 경력

소개

Who am I?



이담호 (Damho Lee)

1989.11.04

홍익대학교 컴퓨터공학과
컴파일러 연구실 박사 수료

와이브레인 연구원 (10년차)

Phone : 010-2734-5798

Email : damho1104@gmail.com

LinkedIn : <https://www.linkedin.com/in/damho1104>

Keyword

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

경력 사항

RUVA Tech (2011 ~ 2012)

- RUVA Tech 위치 참조 센서 연동 UI

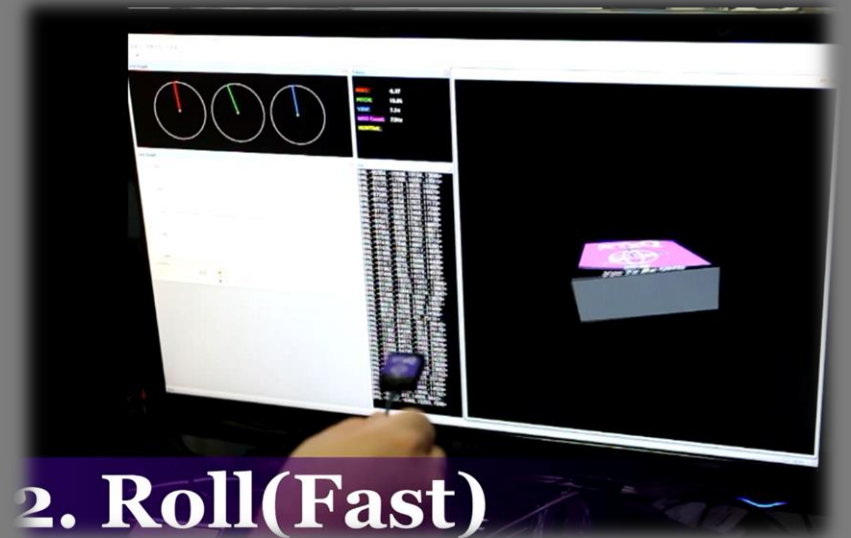
- OpenGL 을 사용한 그래픽 표현
 - 오일러 각/쿼터니언 표현 방식 사용
 - 현재의 센서 값을 그래픽으로 표현
 - 마우스와 연동하여 게임 테스트

- 복수의 센서를 사용한 모션 캡처 시스템
 - 센서 값을 인체 모형 그래픽에 표현

- 결과물

- 논문 1건

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



경력 사항

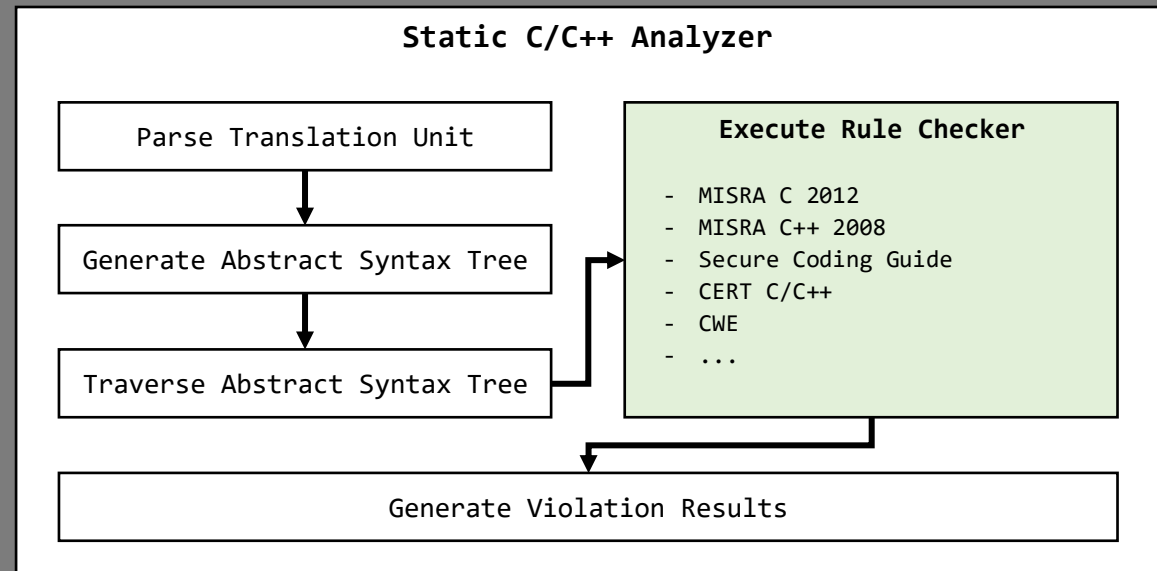
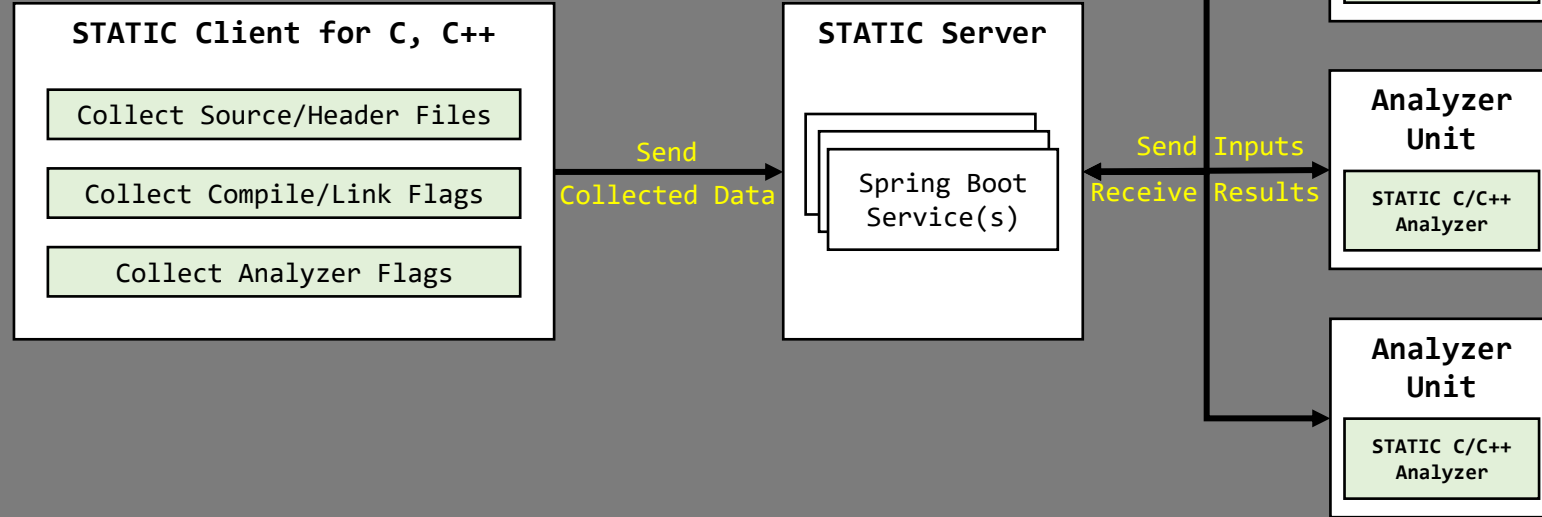
슈어소프트테크
(2019 ~)

- **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

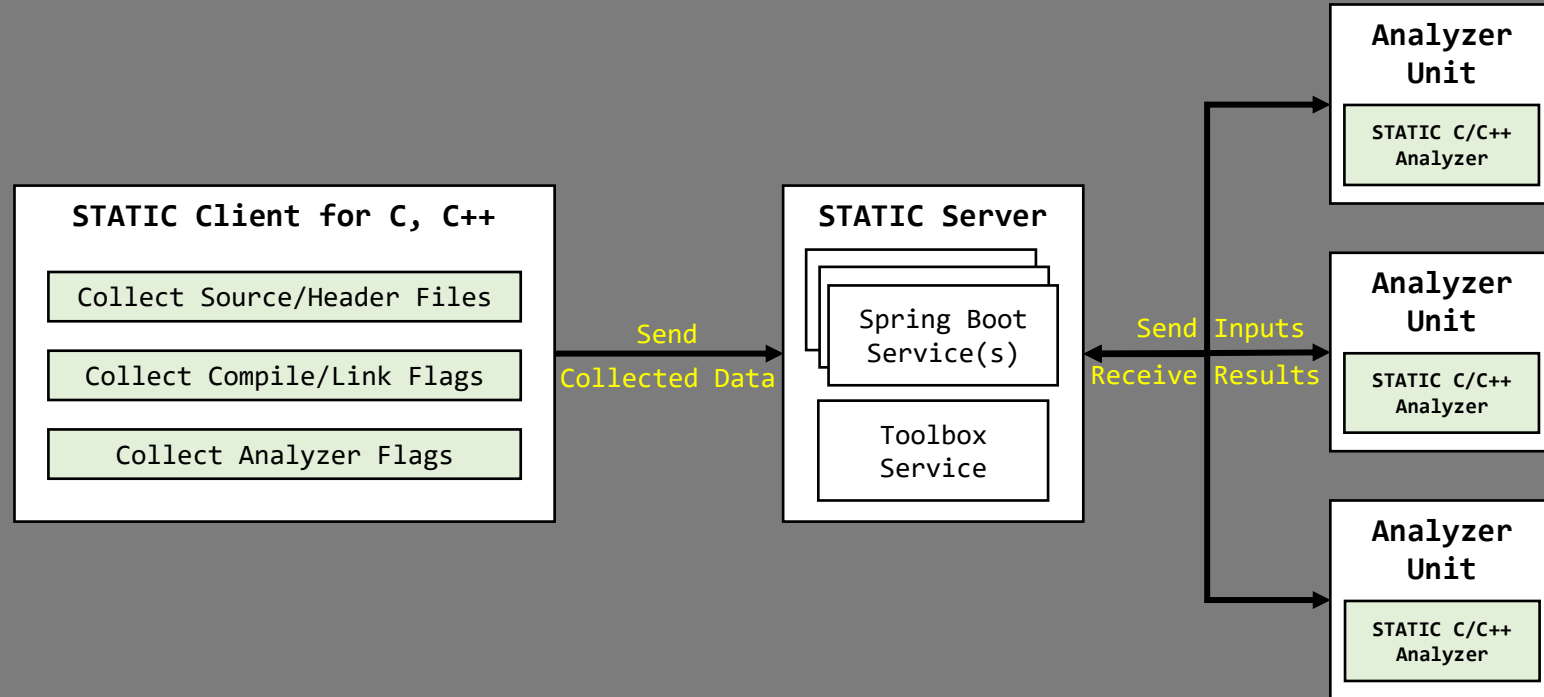
슈어소프트테크 (2019 ~)

• STATIC Analysis Service

- Spring Boot 기반 웹 백엔드 서비스
- 엔터프라이즈급 웹 기반 정적 분석 도구 STATIC의 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
- STATIC을 이용하는 다수의 사용자들의 패키지 버전을 관리하는 서비스
- 사용 기술
 - Python
 - Flask



경력 사항

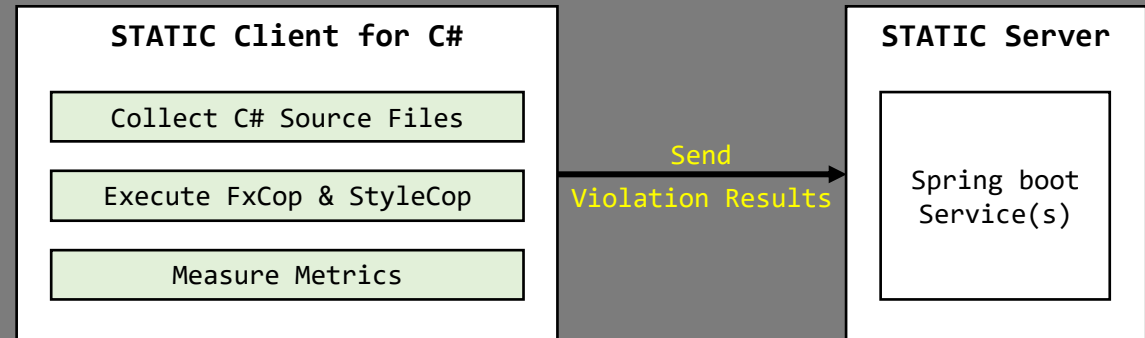
슈어소프트테크
(2019 ~ 2022)

- **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**



경력 사항

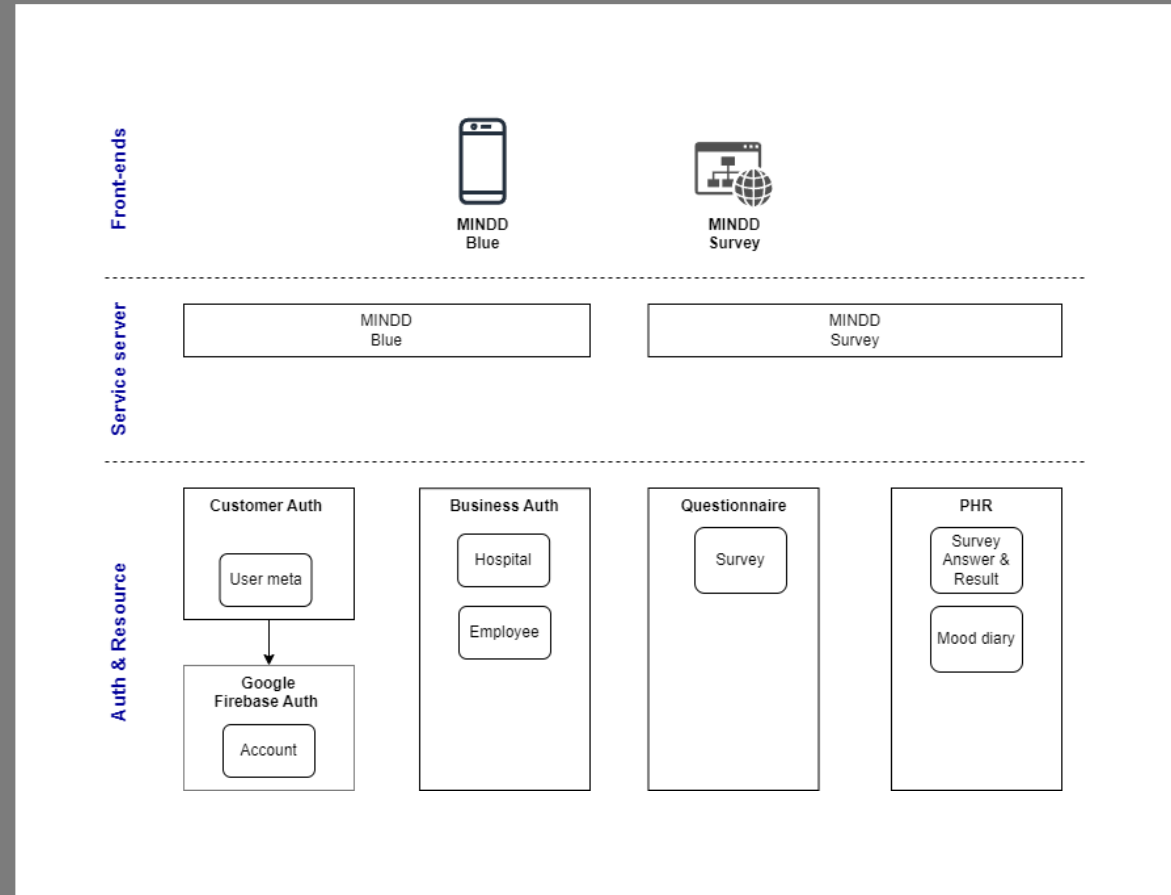
와이브레인 (2022 ~)

• 정신 건강 관리 서비스 MINDD

- 온라인 척도검사를 위한 웹 백엔드 서비스 MINDD Survey 개발
 - 기존 병원에서 종이로 척도검사를 진행했던 방식을 온라인으로 가져와서 진행하는 서비스
 - B2B (병원 대상)
 - 담당: API 백엔드 서비스 및 기록 저장 서비스 개발

• 개인 정신 건강 관리를 위한 웹 백엔드 서비스 MINDD Blue 개발

- 개인 정신건강을 일부 온라인 척도검사와 일기를 통해 관리, 일상 루틴을 등록하고 관리
- B2C
- 담당: API 백엔드 서비스 및 기록 저장 서비스 개발
- 사용 기술
 - MSA
 - Springboot Webflux
 - MongoDB
 - Kotlin

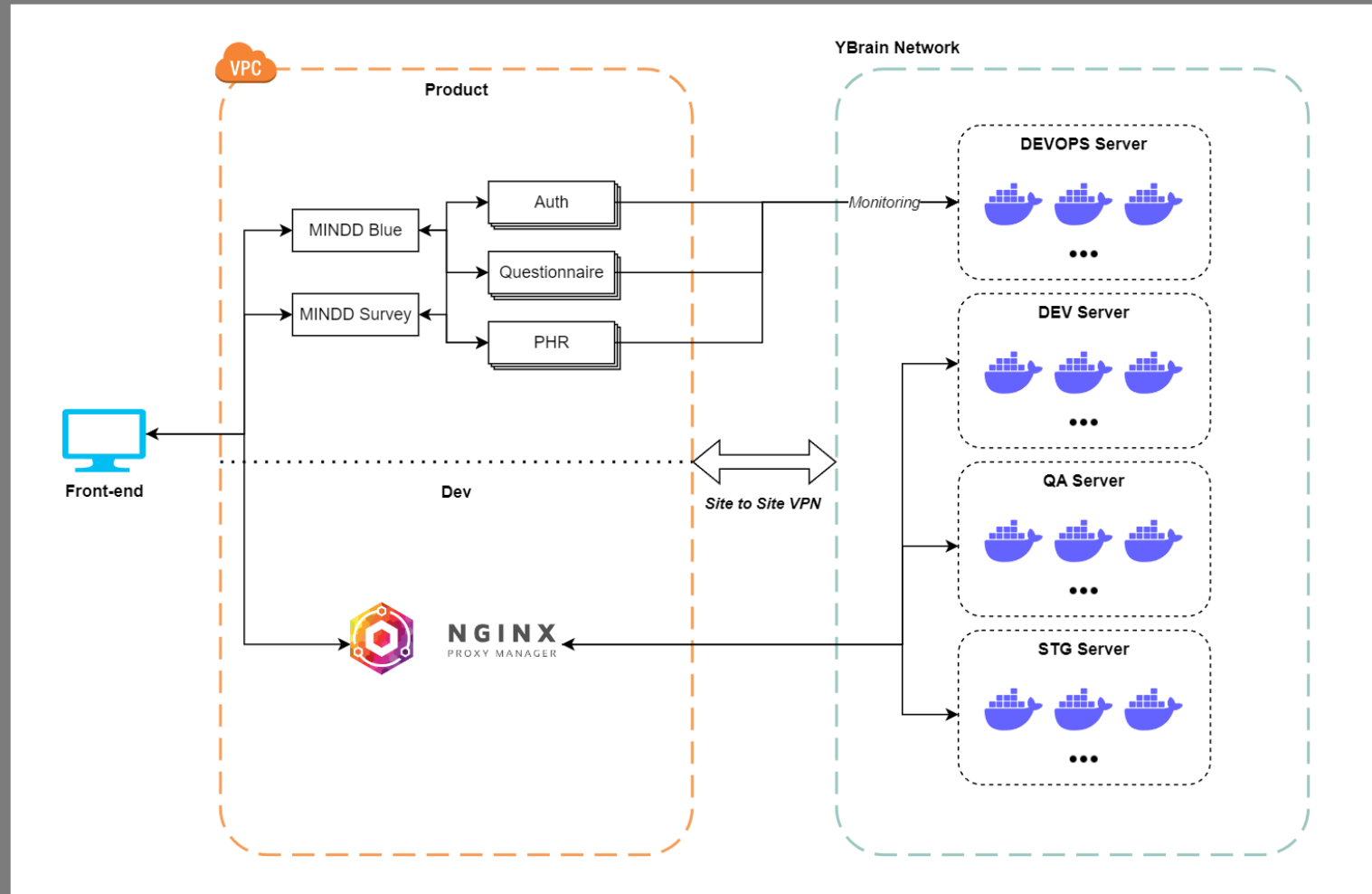


경력 사항

와이브레인 (2022 ~)

• 인프라 관리

- 원격 접속을 위한 VPN 서버 구축
 - Docker 기반 Wireguard VPN 서버
- 사내 개발 환경 백엔드 서비스 인프라 구축
 - Docker 기반 Nginx Proxy Server
 - Docker 기반 로그 모니터링 도구
- APM 도구 Signoz 구축 및 관리
- AWS
 - EC2
 - Lambda
 - API Gateway
 - Beanstalk
 - WAF



연구 배경

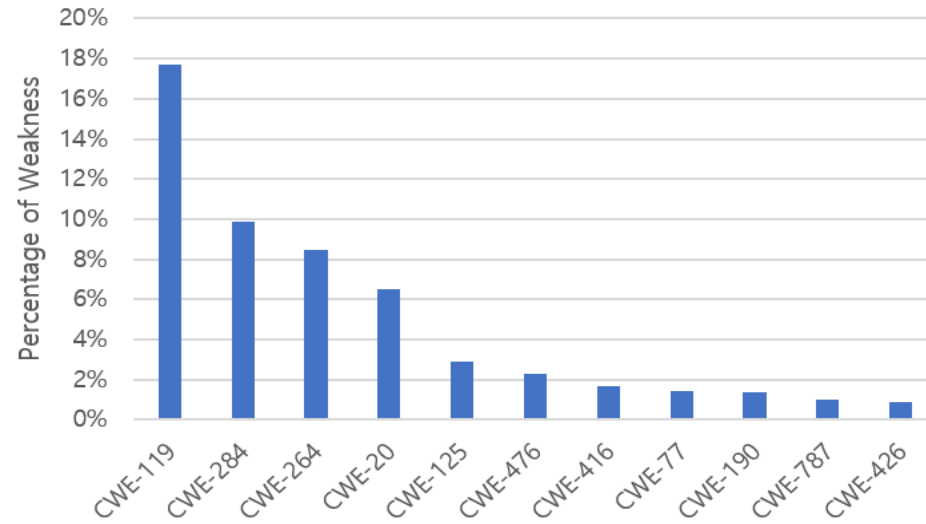
Background

• 시스템 개발 언어의 문제

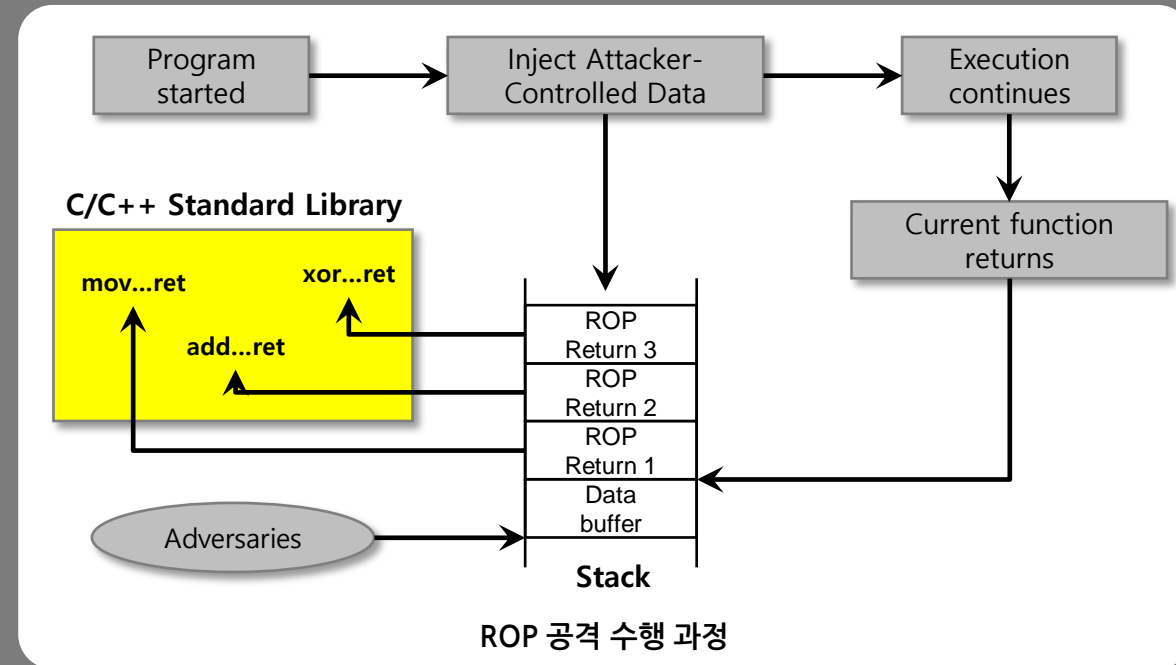
- C/C++ 언어 문제
 - 약한 타입시스템
 - 포인터의 무제한적 사용
 - 통제되지 않는 타입 변환
 - 배열의 불완전 처리
- 언어의 문제로 인해 보안약점 발생
예: 버퍼 오버플로우

• 메모리 안전성 부재로 인한 공격 존재

- 메모리 안전성 부재로 인해 제어흐름 데이터 변조 가능
 - 제어흐름 데이터 예: 코드 포인터
- 제어흐름데이터변조를 시작으로 프로그램공격수행
 - 코드 재사용 공격 (Code Reuse Attack)
 - Return-Oriented Programming (ROP)



2017년 CVE에 등록된 취약점 기준 보안약점 비율



연구 분야 1

프로그램 분석

- C/C++ 시큐어코딩 연구

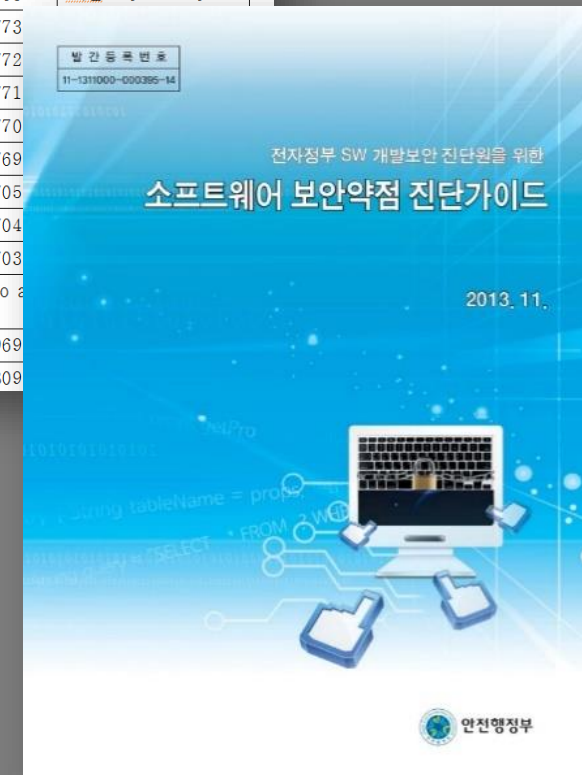
- 47개의 보안약점에 대응되는 3가지 계층, 14가지 규칙 분류
- 시큐어코딩 가이드를 통해 보안 지식 학습 가능

- 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 ('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 Profiling을 통한 실행되지 않은 기본 블록 수집
- Framework

LLVM

Compiler-RT

• Unreachable Region Analysis based on Input Conditions

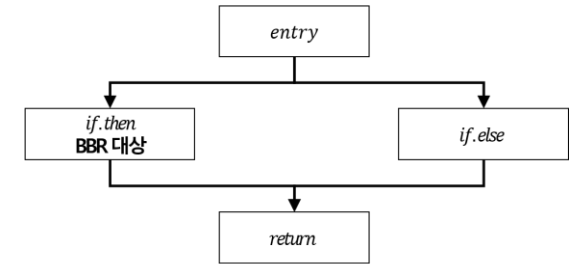
- 미리 지정한 입력 조건에 따라 실행되지 않을 Basic Block 탐색 방법
- Framework

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

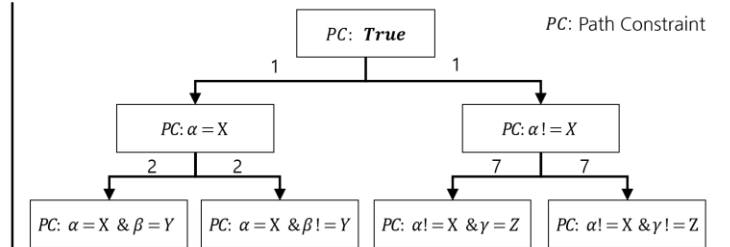


(b) 제어흐름 그래프

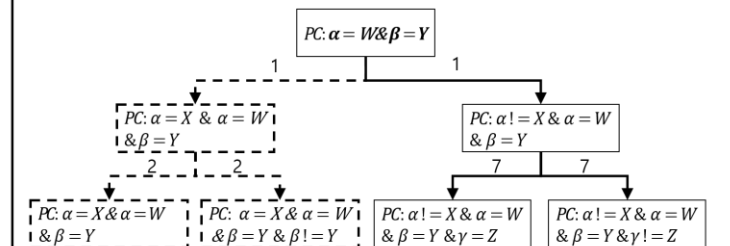
Unreachable Region Analysis based on Input Conditions

```
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.kscsi.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 the 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 interpretation weaknesses, and diagnostic information accordingly, would static analyzers that minimizes false positives and negatives.

• Keyword : Security, Weakness, Static Analysis, Diagnostic

1. Introduction

현재 정보 시스템 개발 사업을 중심으로 한 정부 지원 개발 사업에서 시작되어 최근 개발보안에 대한 요구가 확대되고 있는 상황이다. 이러한 환경 변화에 따라 소프트웨어 개발자 측면에서는 사용자 코딩 규칙을 준수하여 개발하여야 한다는 인식이 확산되었고, 관리 측면에서는 이를 감시하기 위한 도구 사용이 확대되었다.

이와 관련하여 소프트웨어 보안약점이 존재하는지 여부를 검사하기 위한 정적분석 도구에 대한 연구가 활발하게 진행되고 있다. 소프트웨어에서 나타나는 보안 취약점은 지속적으로 변화하고 있고 그에 따라 검사 대상 보안약점도 변화하고 있으므로, 정적분석 도구가 이에 따라 지속적으로 변화하여야 한다. 따라서 보안약점에 대하여 진단 방법을 기술하여 추가하는 방법을 제공하는 도구를 연구한다. 이를 기술하기 위해서는 진단 방법 명세를 위한 표현 방법과 더불어 보안약점과 진단 방법에 대한 지식이 필요하다.

First Author: Kyungsook Han, Corresponding Author: Changwoo
*Kyungsook Han(khan@kpu.ac.kr) Dept. of Computer Engineering
**Damho Lee(damho104@gmail.com) Dept. of Computer Engineering
**Changwoo Pyo(pyoo@hongik.ac.kr) Dept. of Computer Engineering
[Received: 2017. 03. 09, Revised: 2017. 03. 14, Accepted: 2017. 03. 14]
*This work was supported by the core technology R&D project (UD160013ED)

이 논문에서는 입력 조건에 따라 제어흐름이 도달하지 않는 영역을 분석하는 방법을 제안한다. 입력 조건은 기호 식으로 입력 조건으로 추가된 기호 실행의 결과는 제어흐름이 된다. Coreutils를 대상으로 유용성을 평가하기 위해 특정 입력과, ROP 기법의 개수를 분석하였다. 이 영역들을 제거할 경우 42%가 줄어든다. 실행의 결과로부터 제어흐름이 도달하지 않는 경우가 없으며, 보안성 향상에도 효과가 있음을 확인하였다.

1. 서론

프로그램은 입력에 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 $S_1; S_2; S_3$ 에 대하여 처리되는 입력 데이터가 조건 C에 대하여 항상 거짓으로 평가된다면 코드 조각을 $S_1; S_2; S_3$ 로 변환해도 결과가 같게 된다.

제거되는 영역에 코드 재사용 문제[1]에 사용되는 가젯(gadget)이 있거나 비제어 데이터 공격[2]을 통해 프로그램의 특성을 탈취할 수 있는 코드 조각을 포함한다면, 프로그램 보안 수준이 향상된다. 위의 예에서 S_2 에 setuid 시스템 호출이 있다면 외부에서 C가 참이 되게 하는 입력 값을 공급하여 권한 상승 공격을 시도해 볼 수 있는데, 즉가 제거되어 가능성이 차단된다.

제어흐름이 도달하지 않는 영역을 탐색하기 위한 방법은 정적 분석, 동적 분석, 기호 실행(symbolic execution)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 정해진다. 정적 분석으로는 조건이 참인지 거짓인

* 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2018R1A1A1A0107711).

입력 조건에 따라 도달하지 않는 영역을 기호 실행을 활용한 분석*

박영관¹ 김태환² 이담호² 표창우²
홍익대학교 컴퓨터공학과
(byg0102, everstar, damho1104)@mail.hongik.ac.kr

Analyzing Unreachable Regions Using Symbolic

YoungGwan Park¹ Taehwan Kim² Damho Lee²
Department of Computer Engineering,

요약

이 논문에서는 입력 조건에 따라 제어흐름이 도달하지 않는 영역을 분석하는 방법을 제안한다. 입력 조건은 기호 식으로 입력 조건으로 추가된 기호 실행의 결과는 제어흐름이 된다. Coreutils를 대상으로 유용성을 평가하기 위해 특정 입력과, ROP 기법의 개수를 분석하였다. 이 영역들을 제거할 경우 42%가 줄어든다. 실행의 결과로부터 제어흐름이 도달하지 않는 경우가 없으며, 보안성 향상에도 효과가 있음을 확인하였다.

계층화된 문맥 자유 문법을 사용한 입력 데이터 기술*

서현지¹ 김태환² 이담호² 표창우²
홍익대학교 컴퓨터공학과
(slori1128, everstar, damho1104)@mail.hongik.ac.kr, pyo@hongik.ac.kr

Input Data Description using Stratified Context-Free Grammars

Hyunji Seo¹ Taehwan Kim² Damho Lee² Changwoo Pyo²
Department of Computer Engineering, Hongik University

요약

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 유용성을 위해 단일 입력 장치와 문법 생성, 기호 실행(Symbolic Execution) 방식의 입력 생성 체계와 비교 평가하였다. 평가 결과를 생성 시간 대비 기본 블록 실행 횟수와 실행 명령을 확인하는 유닛 입력 개수 모두 DDL을 사용한 결과가 다른 방식을 사용한 결과보다 높게 나타났음을 확인할 수 있었다. 기본 블록 실행 횟수 측면에서는 DDL을 사용한 결과가 기호 실행 방식보다 평균적으로 1.31배, 문맥 자유 문법 사용 방식보다는 1.03배 높게 나타났다. 유닛 입력 개수는 기호 실행을 사용한 방식보다 2.8배, 문맥 자유 문법 사용 방식보다 1.3배 더 많은 생성 결과를 보이며, 입력 생성 시간은 기호 실행과 문맥 자유 문법 사용 방식 각각 146.6배, 1.9배 더 짧은 시간을 보였다.

1. 서론

퍼징(fuzzing)은 자동 소프트웨어 테스트와 취약성 분석에 널리 사용되는 방법으로서 동적 프로그램 분석에 사용되는 입력 데이터를 생성할 때에도 사용될 수 있다. [1] 과정에 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 특징을 기술하여 입력 데이터 생성에 활용하면 유효하지 않은 데이터를 사용함으로써 인해 발생하는 동적 분석의 비효율을 방지할 수 있다.

문맥 자유 문법(Context-Free Grammar, 이하 CFG로 표시)은 입력 데이터의 구조 및 값의 범위와 같은 문법을 기술하기에 적합한 형식언어 체계이다. CFG의 생성 규칙을 하향식으로 적용하여 문법을 생성하는 기법을 사용하면 기술된 특징을 만족하는 임의의 입력 데이터를 생성할 수 있다. CFG를 사용한 입력 생성은 자바스크립트 데이터의 보안약점을 분석할 때 사용한 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 받아들이며 맵 문자를 처리하는 프로그램으로, 'unexpand -t 12 file.txt'와 같은 명령어 입력이 가능하다. 입력 문자 중 명령어에 적힌 나열은 '-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 * 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2018R1A1A1A0107711).

2. 계층 문법의 구조

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

연결자는 하나의 문법에서 단말 기호처럼 사용 가능한 하나의 입력이 들어오는 장치 정보가 함께 표시된다는 특징이 있다. DDL로 기술한 unexpand 입력은 그림 1에 보여 주고 있다. 1행에서 1행은 문법 생성을 기술하는 CFG의 생성 규칙이며, 10행에서 11행은 명령어에 포함된 파일 입력의 내부 데이터를 기술하는 CFG의 생성 규칙에 해당한다. 두 문법은 1행에 있는 연결자에 의하여 연결

연구 분야 2-1

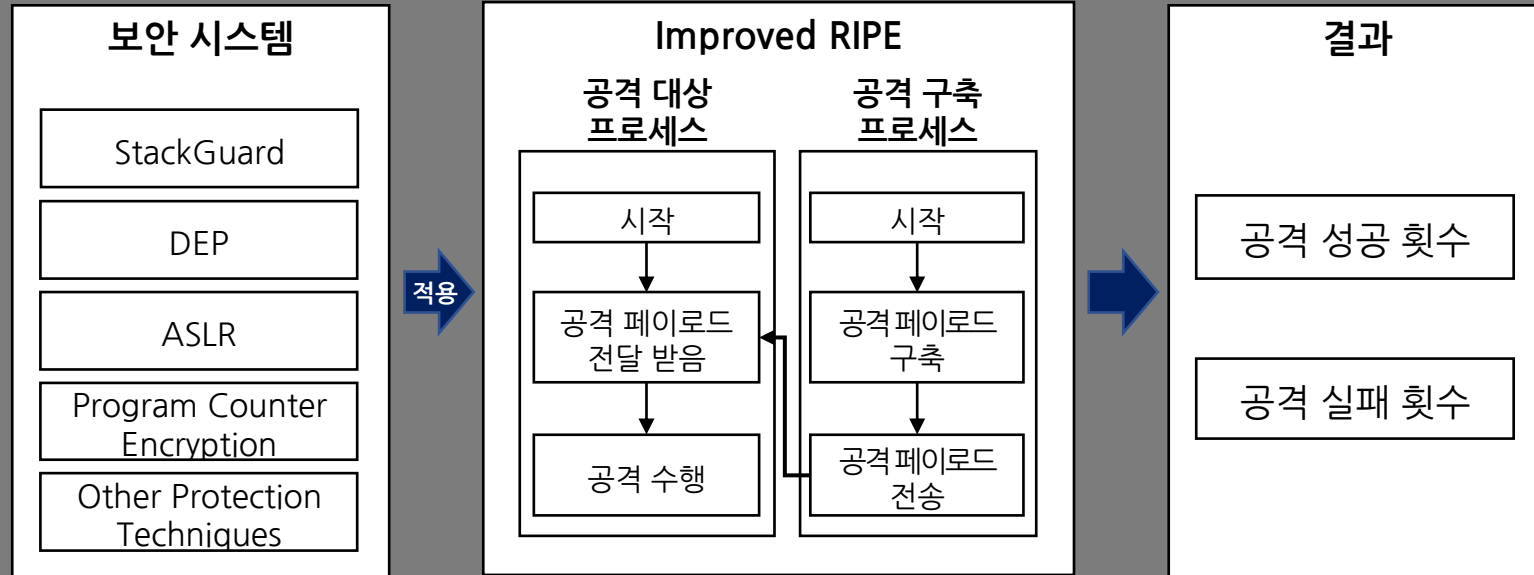
프로그램 보안(공격)

• 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).

실행시간 침입 방지 평가 프로그램 RIPE의 개선과 확장*

이담호¹, 이현규²

¹damho1104@gmail.com, ²lhg11234@naver.com

Improvement and Expansion of Runtime Intrusion Prevention Program

Damho Lee¹, Hyungyu Lee²

비버 오버플로우 기법은 비버의 경계 값을 복사되는 현상을 의미한다. 2011년 John Wilander는 이 공격을 방지하기 위해 RIPE를 소개하였다. 이 논문에서는 RIPE의 공격 코드, 위치 등의 공격을 종합하여 만들어진 비버 오버플로우 공격으로 운영체제 커널에서 제공하는 방어 기법들을 평가하는 도구를 소개한다. RIPE 프로그램 보안 기법 평가 도구는

1. 서론

프로그램 공격은 공격자가 프로그램 제어 흐름을 변경할 수 있는 데이터를 원하는 값으로 변조하면서 실행된다. 프로그램 공격 기법의 대표적인 사례는 비버 오버플로우[1]이다. 비버 오버플로우는 비버의 경계를 넘지 않는 함수나 프로그램을 통해 비버의 크기 이상 데이터가 복사되는 현상이다. 비버의 경계 밖으로 데이터가 복사되면 함수의 코드 포인터와 같은 제어 관련된 데이터가 변조되어 프로그램은 공격자가 원하는 흐름으로 변경된다. 이러한 공격을 예방하기 위해 ProPolice[2]와 StackShield[3] 등의 방어 기법이 도입되었고 또한 방어 기법들을 평가하는 도구들도 만들어졌다.

2011년에 John Wilander는 비버 오버플로우 공격에 대한 프로그램 보안 기법 평가 도구인 RIPE를 발표하였다. RIPE는 공격 코드, 위치 등의 공격을 종합하여 만들어진 비버 오버플로우 공격으로 운영체제 커널에서 제공하는 방어 기법들을 평가하는 도구로 소개한다. RIPE 프로그램 보안 기법 평가 도구는

- * 본 연구는 미래창조과학부 및 정보통신산업진흥원의 연구센터특성 지원사업/IT융합고급인력양성지원사업의 결과로 수행되었음 (NIPA-2014-H0301-14-1023)
- * 본 연구는 국방과학연구소에서 수행중인 "사이버 공격 방어 기술 개발"을 통해 지원되는 연구로 수행되었음 (UD140009ED)

홍익대학교 (P. 223)

ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 42, No. 8, pp. 1049-1056, 2015. 8
http://dx.doi.org/10.5626/JOK.2015.42.8.1049

실행시간 침입 방지 평가 프로그램(RIPE)의 개선

(Improvement of Runtime Intrusion Prevention Program)

이현규^{*} 이담호^{*}
(Hyungyu Lee) (Damho Lee)

이상훈^{***} 김훈^{*}
(Sanghoon Lee) (Hoonky)

요약 2011년에 발표된 RIPE는 프로그램 공격과 비버 오버플로우 기반 공격에 대한 완화 기법들을 평가 실행시간에 구성되며, RIPE가 실행될 때에는 공격할 수 밖에 없게 된다. 그 결과 공격 루틴은 방어 기법이다. 이 논문에서는 RIPE의 공격과 방어 루틴에 대해 난독화와 같은 기법들에 근거한 방어 기법에 대한 방어 능력을 실험할 수 있도록 실험 모드 분석을 공격을 수행하도록 38 가지 패턴의 공격을 도입하고, 보호 효과 평가의 정확성을 높였다.
키워드: RIPE, 비버 오버플로우, 주소 공간 난독

Abstract Runtime Intrusion Prevention Evaluation for evaluating mitigation techniques against RSI. RIPE is built as a single process, defense and at address space layouts when RIPE is tested. As a for defense routines without restriction. We sepa

- * 이 논문은 2014년도 홍익대학교 학술연구진흥사업에 의하여 지원되었음
- * 본 연구는 미래창조과학부 및 정보통신산업진흥원의 연구센터특성 지원사업/IT융합고급인력양성지원사업의 결과로 수행되었음 (NIPA-2014-H0301-14-1023)
- * 본 연구는 국방과학연구소에서 수행중인 "사이버 공격 방어 기술 개발"을 통해 지원되는 연구로 수행되었음 (UD140009ED)

* 비 회 원 : 홍익대학교 컴퓨터공학과
lhg11234@naver.com
damho1104@gmail.com, hongik.ac.kr

** 학생회원 : 홍익대학교 컴퓨터공학과
kimsooh2@naver.com

*** 비 회 원 : 국방과학연구소 제2기술연구소 국방사이버기술센터
shh101@add.ac.kr
hunk@add.ac.kr

**** 총신회원 : 홍익대학교 컴퓨터공학과 교수(Hongik Univ.)
pvo@hongik.ac.kr
(Corresponding author)

홍익대학교 (P. 223.194.69.44)

제 C-2014-022811 호

프 로 그 램 등 록 증

1. 프로그램의 제호: 개선된 실행시간 방어 기법 평가 프로그램 (명칭)
2. 저작자 성명 (법인명): 고려대학교 산학협력단 서울특별시 성북구 안암로
3. 생년월일 (법인등록번호): 114471-0002565
4. 창작연월일: 2014년05월
5. 공표연월일: 2014년06월25일
6. 등록사항: 저작자 : 고려대학교 산학협력단
창작 : 2014.05, 공표 : 2014.06.25
7. 등록연월일: 2014년09월16일

「저작권법」 제53조에 따라 위와 같이 등록되었음을 증명합니다.



2014년 09월 17일
한국저작권위원회



연구 분야 2-2

프로그램 보안(방어)

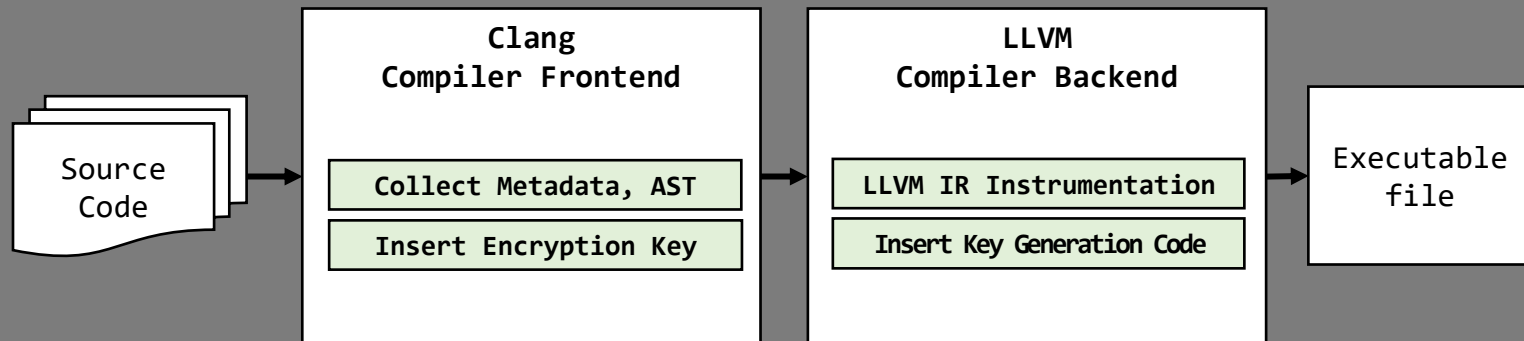
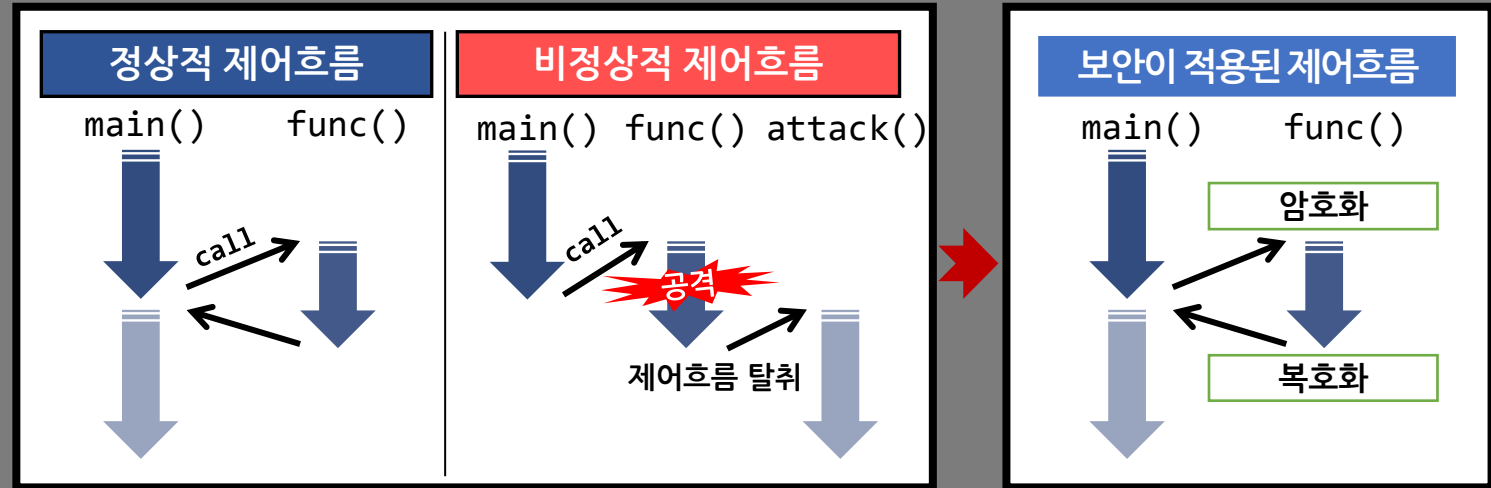
- Control-Flow Data Protection

- 실행시간에 제어흐름 데이터 암호화, 제어흐름 데이터 사용시 복호화
- 암호 키는 프로그램 실행시간에 변경
- 많은 업무

Collect Metadata in AST phase
LLVM IR Instrumentation(LLVM IR Pass)

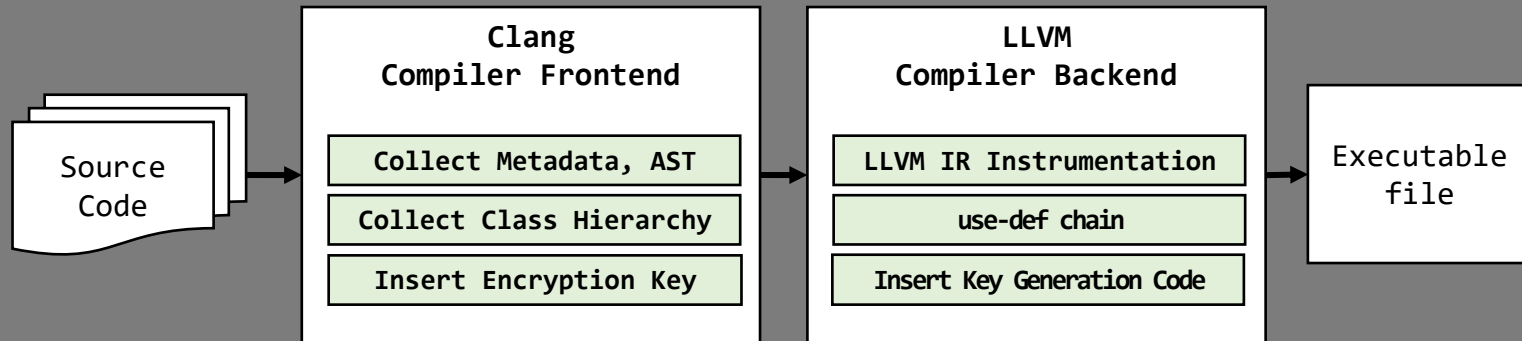
- Framework

Clang
LLVM
Compiler-RT
LibTooling



연구 분야 2-2

프로그램 보안(방어)



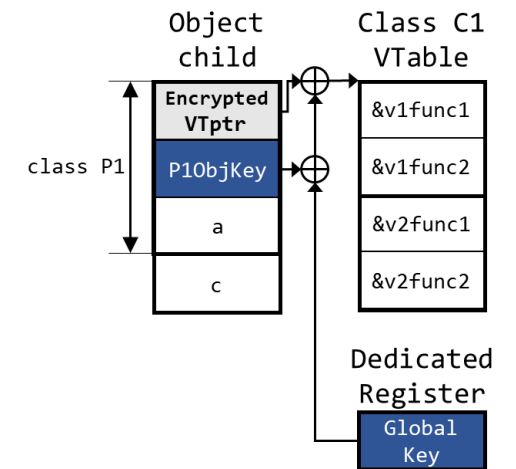
• VTable Pointer Protection

- 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 답 호

2016년 2월

출원번호통지서

페이지 1/4

관인생략

출원번호통지서

출원일자 2016.12.30
특기사항 심사청구(무) 공개신청(무)
출원번호 10-2016-0183304 (출원번호 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

특허청장

<< 안내 >>

1. 귀하의 출원은 위와 같이 정상적으로 접수되었으며, 이후의 심사 진행사항은 출원번호를 통해 확인하실 수 있습니다.
2. 출원에 따른 수수료는 접수일로부터 다음날까지 통편된 납입영수증에 성명, 납부자번호 등을 기재하여 가까운 우체국 또는 은행에 납부하여야 합니다.
※ 납부자번호 : 0131(기공코드) + 접수번호
3. 귀하의 주소, 연락처 등의 변경사항이 있을 경우, 즉시 (특허고려번호 정보변경(장청), 장청신고서)를 제출하여야 하며 출원 이후의 각종 통지서를 정상적으로 받을 수 있습니다.
※ 특허로(patent.go.kr) 접속 > 민원서비스(다문로드) > 특허법 시행규칙 별지 제5호 서식
4. 특허(실용신안등록)출원은 명세서 또는 도면의 보장이 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(특히, 실용신안이나 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> 특허(실용신안)PCT/마드리드
※ 우선권 인정기간 : 특허, 실용신안 12개월, 상표 디자인은 6개월 이내
※ 미국특허상표청의 선출원을 기초로 우리나라에 우선권주장출원 시, 선출원이 미공개상태이면, 우선일로부터 16개월 이내에 미국특허상표청에 (전자적으로한가서(PTOSB/39)를 제출하거나 우리나라에 우선권 주장서류를 제출하여야 합니다.

제 C-2015-025903 호

프로그램 등록증

1. 프로그램의 제호 가상 테이블 포인터 인코딩 기법을 적용한 LLVM(엘엘브이엠) 컴파일러 (명칭)
2. 저작자 성명 고려대학교 산학협력단 서울특별시 성북구 안암로 (법인명칭)
3. 생년월일 114471-0002565 (법인등록번호)

4. 창작연월일 2015년 10월 20일

5. 공표연월일 2015년 10월 28일

6. 등록사항 저작자 : 고려대학교 산학협력단, 창작 : 2015. 10. 20, 공표 : 2015. 10. 28

7. 등록연월일 2015년 11월 04일

한국저작권위원회
KOREA COPYRIGHT COMMISSION

「저작권법」 제53조에 따라 위와 같이 등록되었음을 증명합니다.

2015년 11월 05일

제 C-2016-022293 호

프로그램 등록증

1. 프로그램의 제호 개선된 가상 함수 테이블 포인터 인코딩을 적용한 LLVM(엘엘브이엠) 컴파일러 (명칭)
2. 저작자 성명 홍익대학교 산학협력단 서울특별시 마포구 파주산로 (법인명칭)
3. 생년월일 270171-0011448 (법인등록번호)

4. 창작연월일 2016년 08월 24일

5. 공표연월일 -

6. 등록사항 저작자 : 홍익대학교 산학협력단, 창작 : 2016. 08. 24

7. 등록연월일 2016년 09월 19일

한국저작권위원회
KOREA COPYRIGHT COMMISSION

「저작권법」 제53조에 따라 위와 같이 등록되었음을 증명합니다.

2016년 09월 20일

한국저작권위원회



연구 분야 3

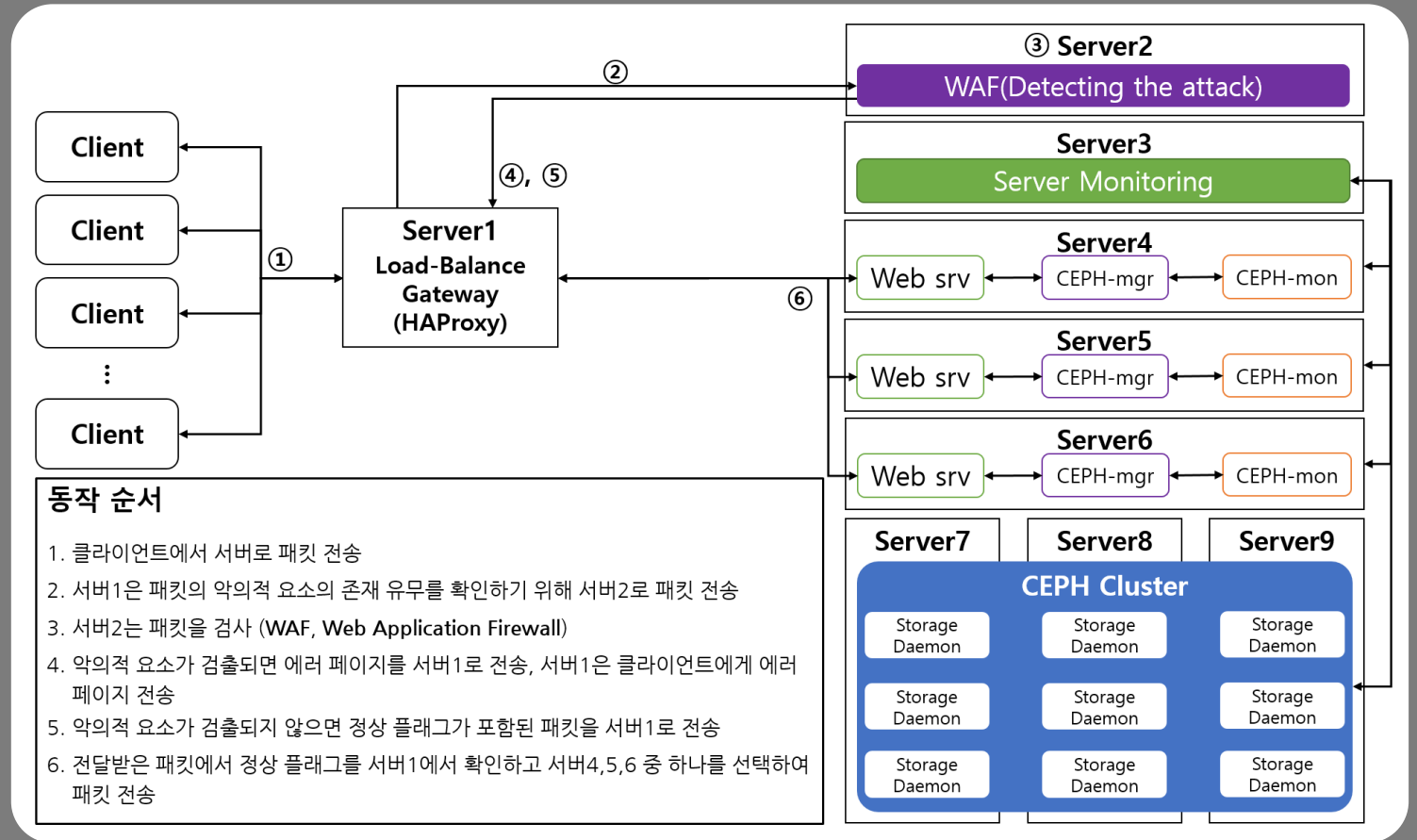
빅데이터

• 빅데이터 저장을 위한 서버 구축

- Distributed Object Storage System
- Compatible Amazon S3 Protocol 사용
- Framework

CEPH

HAProxy (Load-Balance Gateway)

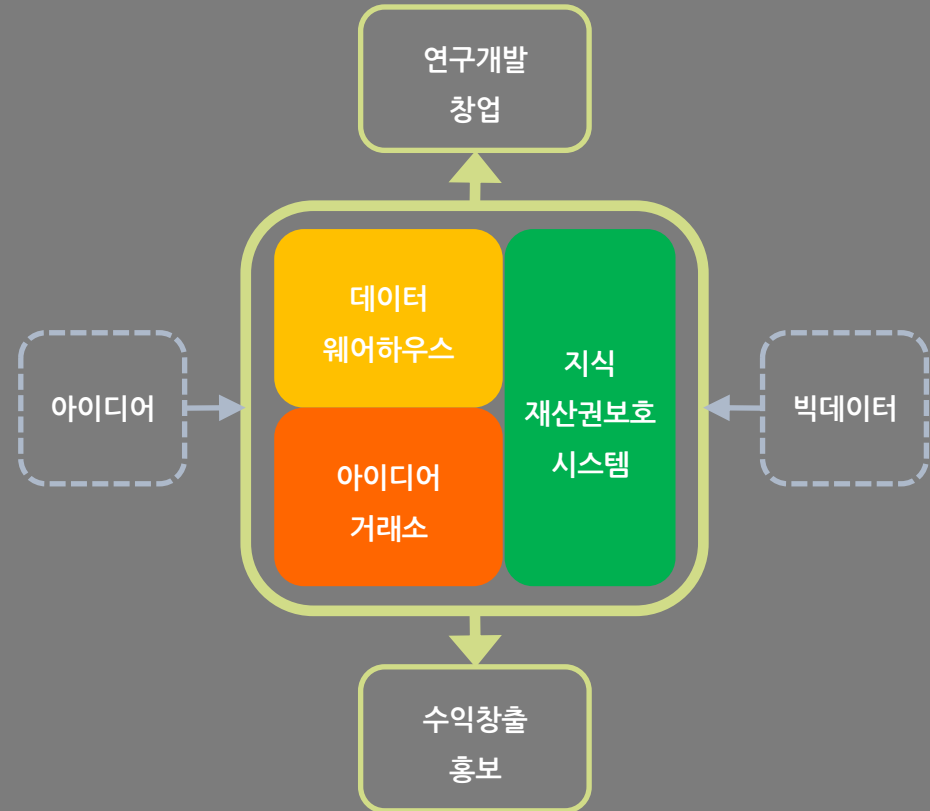
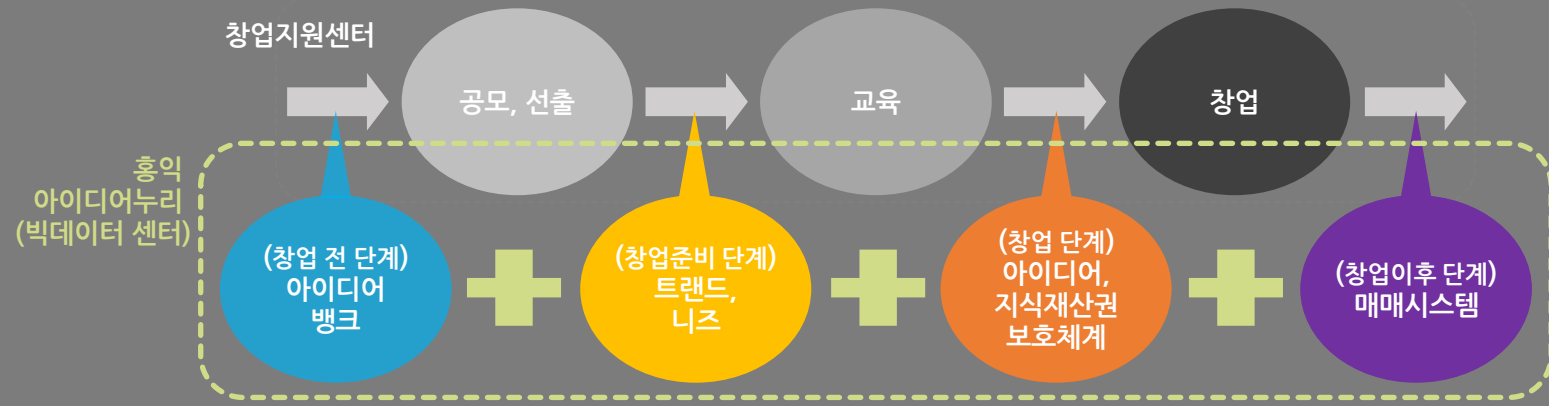


연구 분야 3

빅데이터

- 참여 프로젝트

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



관심 분야 1

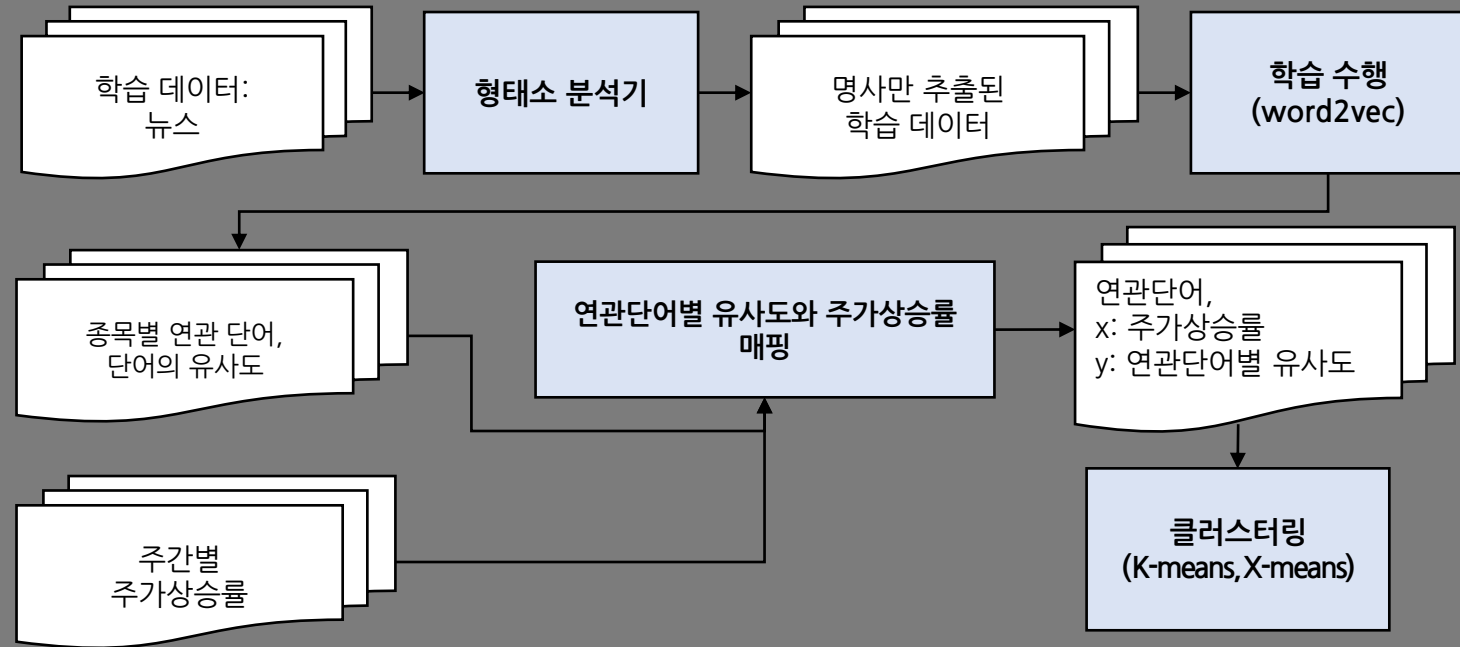
머신러닝

• 종목별 연관 단어를 활용한 주가 예측

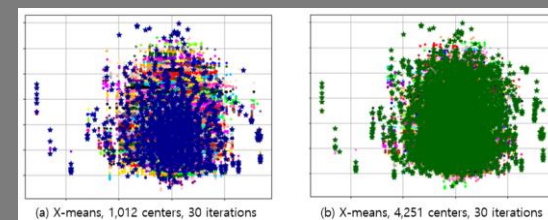
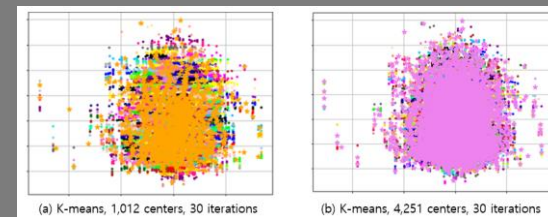
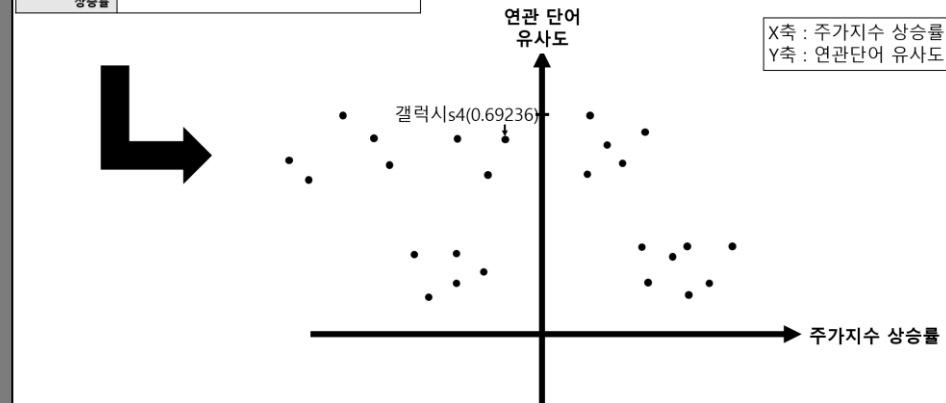
- 3년간(2012~2014) 주간별 뉴스 데이터를 수집 후 주식 종목별 연관 단어 추출
- 연관 단어 유사도와 주가상승률을 매핑 후 클러스터링
- 클러스터링된 단어가 2015년도 뉴스에 나오는 경우 해당 종목의 주가상승률의 +/- 예측
- 결과

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



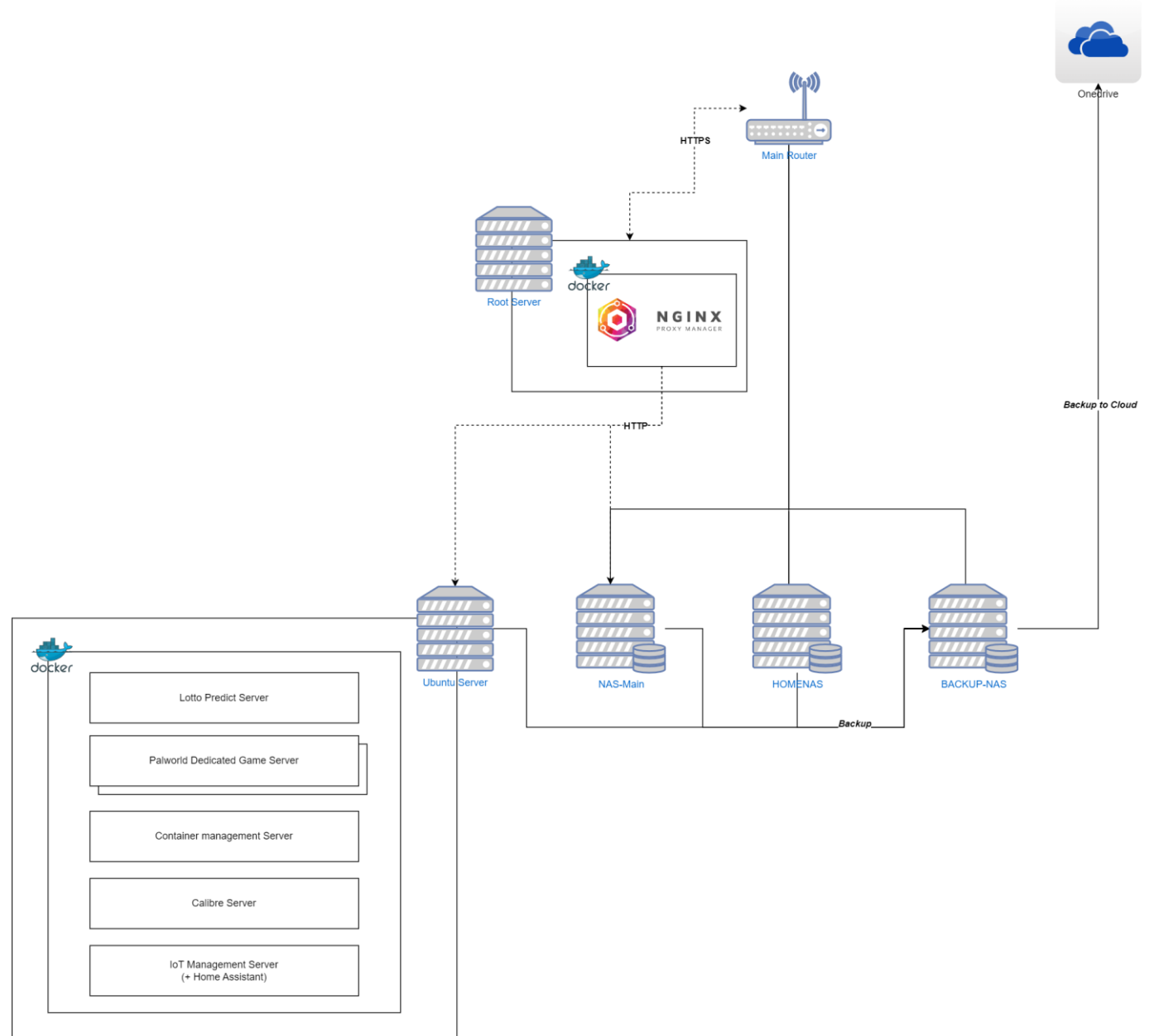
날짜	2012-01-02~2012-01-08			...
종목	삼성전자			...
연관 단어	갤럭시s4	IT	스마트카메라	...
유사도	0.69236	0.542123	0.491278	...
주가지수 상승률	+0.12%			...



관심 분야 1

홈 네트워크

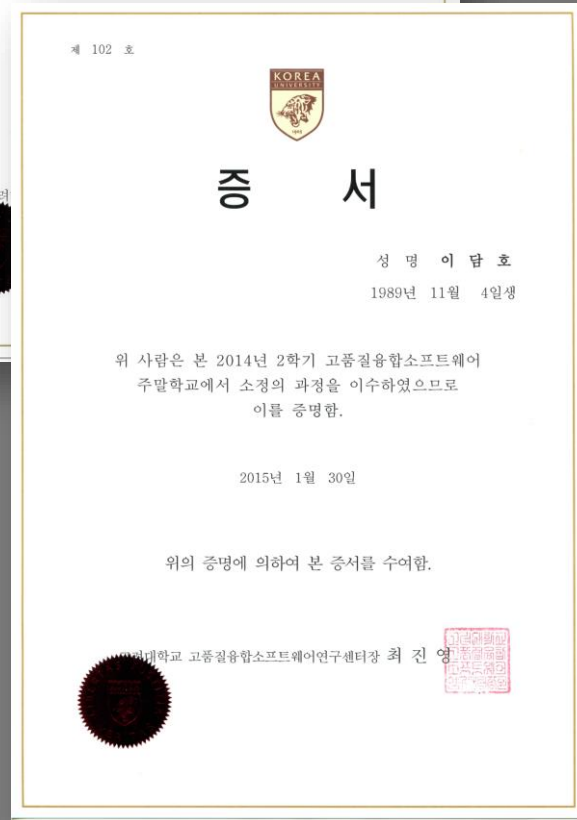
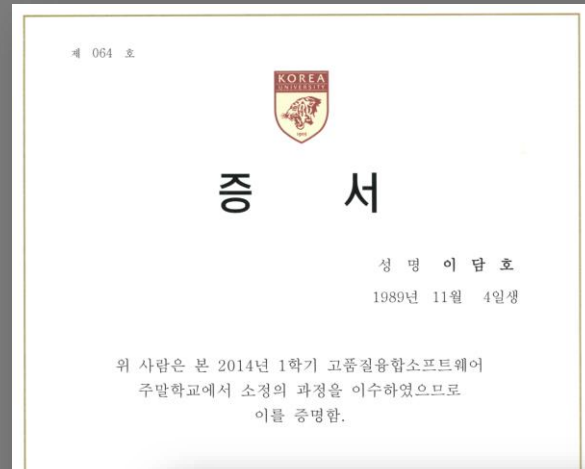
- 홈 네트워크 구축
 - 사이드 프로젝트
 - 로또 번호 예측 Telegram 챗봇 서버
- 운영
 - 팰월드 게임 서버
 - E-book 관리 서버(Calibre Server)
 - Nginx Proxy 서버
 - 영상 관리 Plex 서버
 - IoT 관리 서버
 - 중요 데이터(가족 영상, 사진, ...) 백업 시스템
- 사용 기술
 - Docker
 - Python
 - Bash Shell Script
 - HTTP
 - DNS



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 테스트
 - 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

Yun Ho Yu

Ph.D. Yun Ho Yu, Facilitator

Pyoung Hee Choi

Pyoung Hee Choi

Managing Director of Venture CEO Mentoring Co.

관인생략

출원번호통지서

출원일자 2014.10.02
특기사항 심사청구(유) 공개신청(무)
출원번호 10-2014-0132974 (접수번호 1-1-2014-0943561-85)
출원인명칭 고려대학교 산학협력단(2-2004-017068-0)
대리인성명 김동윤(9-2012-000021-4)
발명자성명 강미영 김광원 강현재 박무규 기영준 연광훈 김현민 아노 이현
박 김문학 이담호 김정민 임지은
발명의명칭 리뷰 등록 서버 및 리뷰 등록 방법

특허청장

<< 안내 >>

1. 귀하의 출원은 위와 같이 정상적으로 접수되었으며, 이후의 심사 진행상황은 출원번호를 통해 확인하실 수 있습니다.
2. 출원에 따른 수수료는 접수일로부터 다음날까지 통보된 납입영수증에 성명, 납부자번호 등을 기재하여 가까운 우체국 또는 은행에 납부하여야 합니다.
※ 납부자번호 : 0031(가과과코드) + 접수번호
3. 귀하의 주소, 연락처 등의 변경사항이 있을 경우, 즉시 [출원인코드 정보변경(경정), 정정신고서]를 제출하여야 출원 이후의 각종 통지서를 정상적으로 받을 수 있습니다.
양 특허청(patent.go.kr) 좌측 > 마우스 클릭 우클릭 > 특허법 시행규칙 별지 제5호 서식
4. 특허(실용신안 등) 출원은 명세서 또는 도면의 보장이 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.
양 제도 안내 : <http://www.kipo.go.kr> 특허마포 PCT마드리드
※ 우선권 인정기간 : 특허-실용신안 12개월, 상표-디자인 6개월 이내
※ 미국특허심판원의 선택특허 기호상 무인나중에 우선권주장출원 시, 선택특허 미공개상태이며, 우선권주장부터 16개월 이내에 미국특허심판원에 [가자적고허가서(PFO58B39)]를 제출하거나 무인나중에 우선권 주장을 제출하여야 합니다.

<http://www.patent.go.kr/jsp/kiponet/ir/receipt/online/appNoOficAct.do>

2014-10-02

교육 이수

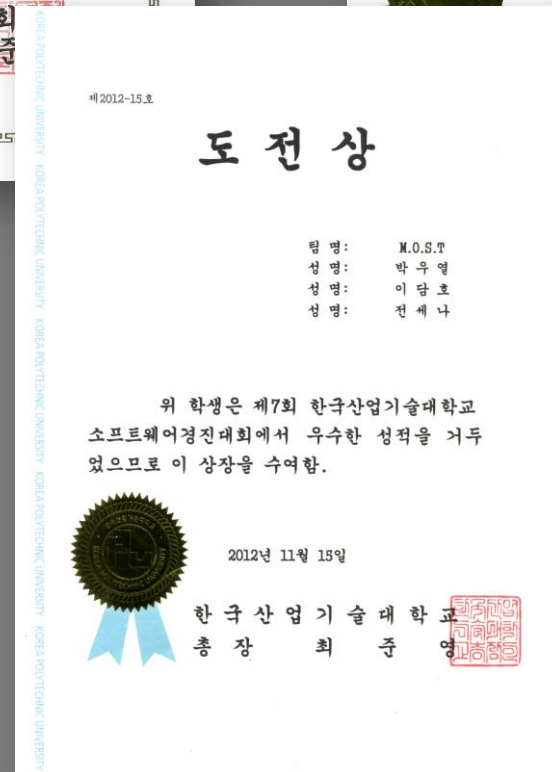
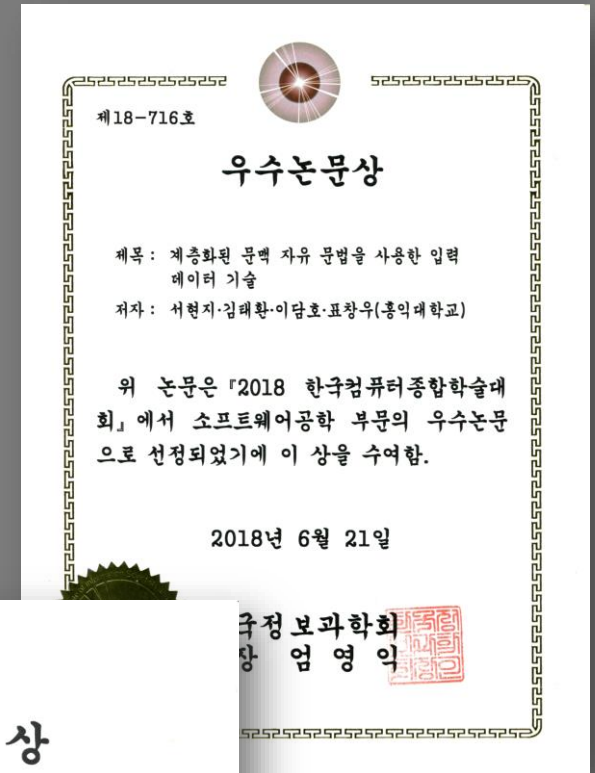
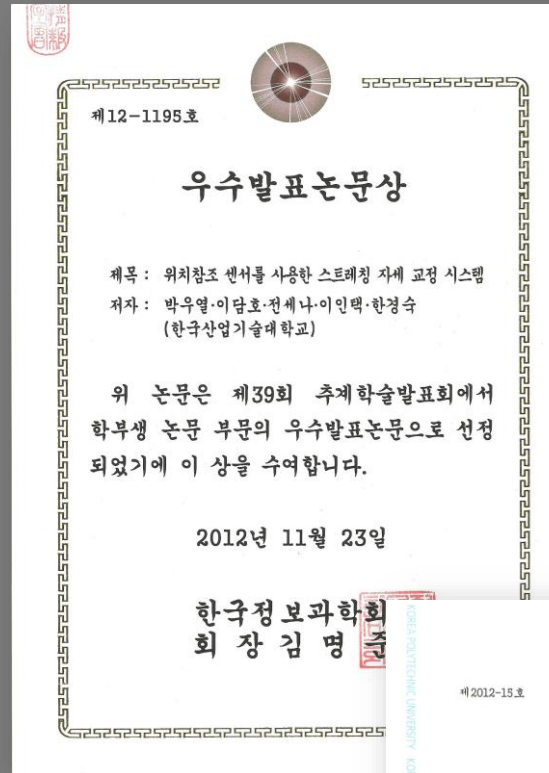
Oracle Certified Professional

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



수상 경력

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



수상 경력

- 슈어소프트테크 슈퍼히어로(우수사원상)
 - 2년 연속 수상
 - 2022
 - 2021

