

2025. 11. 04.

졸업 프로젝트 최종발표

하이브리드 환경에서 DevSecOps 구축 및 공개형 Kubernetes 설정 점검 솔루션 개발

: Development of DevSecOps Implementation in a Hybrid Environment
and an Open Kubernetes Configuration Audit Solution

지도교수 : 유승재 교수님

김건희, 김수현, 남지우, 신재형, 어영민, 전유경



Kubee

CONTENT

하이브리드 환경에서 DevSecOps 구축 및 공개형 Kubernetes 설정 점검 솔루션 개발

01 프로젝트 개요

- 팀 소개
- 프로젝트 배경
- 프로젝트 목표

02 프로젝트 진행 결과

- k8s 주요 보안 취약 설정 식별
- Misconfiguration Pipeline
- Kubeskoop Misconfirguration
- LLM 적용
- Kube 시연 영상
- 오픈소스 취약점 연구

03 프로젝트 성과

- Misconfiguration 제보
- 오픈소스 취약점 제보
- 향후 계획

2025

01 프로젝트 개요

- 팀 소개
- 프로젝트 배경
- 프로젝트 목표

팀 소개

| 2025년 졸업 프로젝트 팀원을 소개함.

프로젝트개요

| 프로젝트 진행 결과

| 프로젝트 성과



유승재 교수님



PM
김수현



PA
김건희



PA
남지우



PA
신재형



PA
어영민



PA
전유경

프로젝트 배경

| Kubernetes의 점유율이 지속적으로 증가하고 있음.

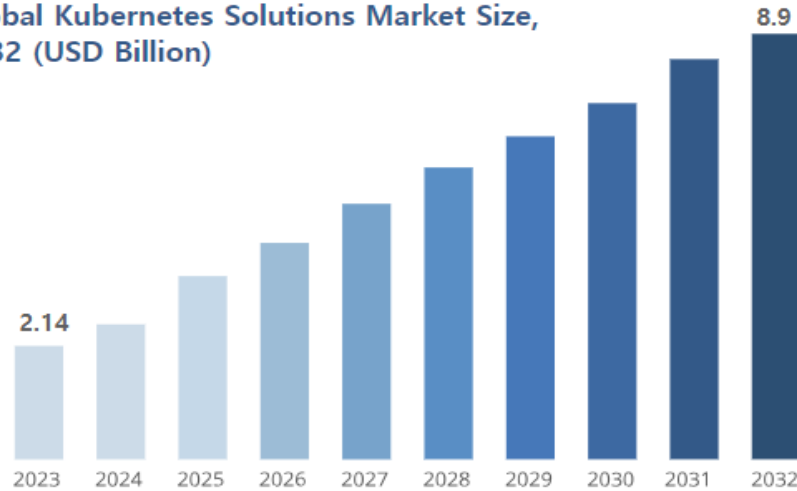
프로젝트 개요

프로젝트 진행 결과

프로젝트 성과

01

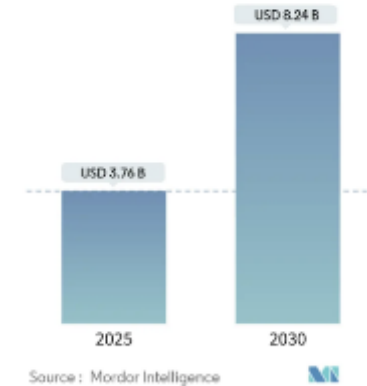
Global Kubernetes Solutions Market Size, 2032 (USD Billion)



2023년 Kubernetes 솔루션 시장 규모는 214억 달러로 평가되었으며, 2032년까지 890억 달러로 성장할 것으로 예상
businessresearchinsights

02

Kubernetes Market
Market Size in USD Billion
CAGR 17.01%



Kubernetes 시장 규모가 2025년에 3억 7,600만 달러에 도달할 것으로 예상되며, 2030년까지 연평균 성장률 17.01%로 성장하여 8억 2,400만 달러에 이를 것으로 전망
mordorintelligence

Kubernetes 점유율 증가

프로젝트 배경

Kubernetes의 기본 설정으로 인해 보안 취약점이 발생할 수 있음.

Pod API 사용	automountServiceAccountToken: true 서비스 계정 토큰이 모든 Pod에 자동으로 마운트되며 기본값으로 설정됨
초기 클러스터 구성	anonymous-auth: true 기본적으로 True 값이 설정되어 있어, 인증되지 않은 익명 요청이 가능함
로그 저장	readOnlyRootFilesystem: false 루트 파일 시스템의 읽기/쓰기 기능 활성화 가능
SetUID 사용 레거시 시스템	allowPrivilegeEscalation: true 컨테이너 내부에서 실행되는 프로세스가 권한 상승을 수행할 수 있음

- Pod 권한을 가진 운영 API
- 익명의 공격자가 'api random'을 호출
- 컨테이너 내부에서 루트 권한 획득
- 컨테이너 내부에 악성 파일 저장

기본값이 낮게 설정되면 악용될 수 있는 보안 취약점이 발생

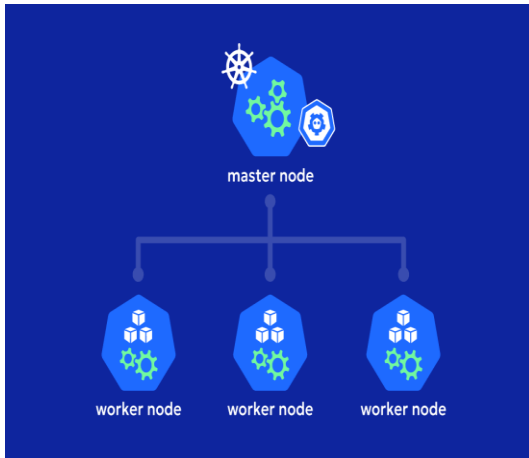
프로젝트 배경

| Kubebench, Kubescape, Polaris 등 기존 설정 점검 오픈소스의 한계점을 도출 함.

프로젝트 개요

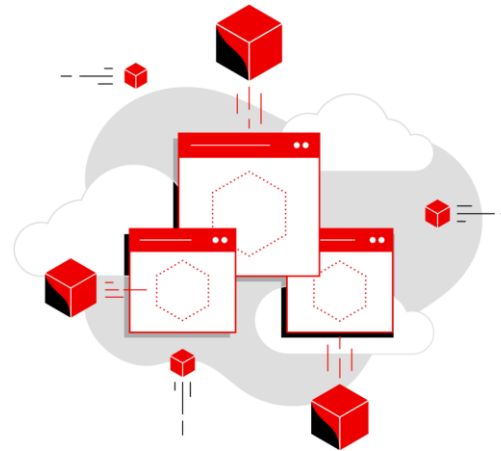
프로젝트 진행 결과

프로젝트 성과



권한 상승 경로를 제공하지 않음

익스플로잇 경로가 표시되지 않아
Pod → Worker Node와 같은 공격 흐름을
명확하게 파악하기 어려움.



익스플로잇 시나리오 부재

익스플로잇 명령어와 실제 공격 시나리오를
알 수 없음.

```
== Summary node ==  
14 checks PASS  
1 checks FAIL  
8 checks WARN  
0 checks INFO  
  
[INFO] 5 Kubernetes Policies  
[INFO] 5.1 RBAC and Service Accounts  
[WARN] 5.1.1 Ensure that the cluster-admin role is only used  
[WARN] 5.1.2 Minimize access to secrets (Manual)  
[WARN] 5.1.3 Minimize wildcard use in Roles and ClusterRoles  
[WARN] 5.1.4 Minimize access to create pods (Manual)
```

분석 결과의 가독성 문제

분석 결과가 JSON, CLI 등 텍스트 기반으로
제공되어 직관적이 파악이 어려움.

프로젝트 목표

더 정확한 분석(LLM)과 더 쉬운 사용(시각화)을 통한 쿠버네티스 설정 점검이 목표임.

프로젝트 개요

프로젝트 진행 결과

프로젝트 성과



쿠버네티스 설정 점검 솔루션 개발

01

쿠버네티스 설정 점검 솔루션 개발

설정 연관성(Context)과 공격 경로(Attack Path)를 종합 분석하여, 기존 오픈소스의 한계를 보완한 쿠버네티스 설정 점검 솔루션을 개발

02

익스플로잇 연계 경로 표시

Pod → Worker Node와 같은 공격경로 흐름 가능성을 LLM RAG 기술을 통해 명확하게 파악

03

사용자 친화적인 시각화

쿠버네티스 비전문가를 위한 사용자 친화적 시각화 구현.

02 프로젝트 진행 결과

- k8s 주요 보안 취약 설정 식별
- Misconfiguration Pipeline
- Kubeskoop Misconfirguration
- LLM 적용
- Kubeeye 시연 영상
- 오픈소스 취약점 연구

k8s 주요 보안 설정 취약점 식별

쿠버네티스 17가지 주요 보안 설정 취약점 조사를 진행했음.

설정	역할	위험
<code>privileged: true</code>	컨테이너가 호스트의 모든 장치에 접근하고, 거의 모든 커널 기능을 사용할 수 있게 해요.	심각: 컨테이너 탈출(Container Escape)이 매우 쉬워지며, 호스트 전체 시스템을 장악할 수 있어요.
<code>hostNetwork: true</code>	컨테이너가 호스트의 네트워크 네임스페이스를 공유해요.	높음: 컨테이너가 호스트의 <code>localhost</code> 서비스에 접근하거나, 호스트 IP로 통신하여 네트워크 보안 정책을 우회할 수 있어요.
<code>hostPID: true</code>	컨테이너가 호스트의 PID 네임스페이스를 공유해요.	높음: 컨테이너 내부에서 호스트의 모든 프로세스를 확인하고, 시그널을 보내거나 종료시킬 수 있어요.
<code>hostIPC: true</code>	컨테이너가 호스트의 IPC 네임스페이스를 공유해요.	높음: 호스트의 공유 메모리 같은 프로세스 간 통신 자원에 접근하여 정보를 탈취하거나 조작할 수 있어요.
<code>allowPrivilegeEscalation: true</code>	컨테이너 내 프로세스가 <code>setuid</code> 등을 통해 권한을 상승시키는 것을 허용해요.	높음: 컨테이너 내부에서 루트(root) 권한을 획득할 수 있는 경로를 제공해요.
<code>runAsNonRoot: false</code> <code>runAsUser: 0</code> <code>runAsGroup: 0</code>	컨테이너가 루트(UID 0) 사용자로 실행되는 것을 허용해요.	높음: 컨테이너가 기본적으로 루트 권한을 가지므로, 다른 보안 설정이 미흡할 경우 호스트에 더 큰 영향을 줄 수 있어요.

k8s 보안에 영향을 미칠 수 있는
17가지 주요 보안 취약점 조사 및
정규 표현식 매칭 스크립트 개발

```
def __init__(self) -> None:
    self.patterns: Dict[str, re.Pattern[str]] = {
        "privileged_true": re.compile(r"\bprivileged\s*:\s*true\b", re.IGNORECASE),
        "hostNetwork_true": re.compile(r"\bhostNetwork\s*:\s*true\b", re.IGNORECASE),
        "hostPID_true": re.compile(r"\bhostPID\s*:\s*true\b", re.IGNORECASE),
        "hostIPC_true": re.compile(r"\bhostIPC\s*:\s*true\b", re.IGNORECASE),
        "allowPrivilegeEscalation_true": re.compile(r"\ballowPrivilegeEscalation\s*:\s*true\b", re.IGNORECASE),
        "runAsNonRoot_false": re.compile(r"\brunAsNonRoot\s*:\s*false\b", re.IGNORECASE),
        "runAsUser_root": re.compile(r"\brunAsUser\s*:\s*\b", re.IGNORECASE),
        "runAsGroup_root": re.compile(r"\brunAsGroup\s*:\s*\b", re.IGNORECASE),
        "readOnlyRootFilesystem_false": re.compile(r"\breadOnlyRootFilesystem\s*:\s*false\b", re.IGNORECASE),
        "automountServiceAccountToken_true": re.compile(r"\bautomountServiceAccountToken\s*:\s*true\b", re.IGNORECASE),
        "serviceAccountName_default": re.compile(r"\bserviceAccountName\s*:\s*default\b", re.IGNORECASE),
        "hostPath_used": re.compile(r"\bhostPath\s*:\s*", re.IGNORECASE),
        "service_type_NodePort": re.compile(r"\btype\s*:\s*NodePort\b", re.IGNORECASE),
        "hostPort_used": re.compile(r"\bhostPort\s*:\s*\d+\b", re.IGNORECASE),
        "shareProcessNamespace_true": re.compile(r"\bshareProcessNamespace\s*:\s*true\b", re.IGNORECASE),
        "dnsPolicy_ClusterFirstWithHostNet": re.compile(r"\bdnsPolicy\s*:\s*ClusterFirstWithHostNet\b", re.IGNORECASE),
        "seccompProfile_Unconfined": re.compile(r"\bseccompProfile\s*:\s*\b", re.IGNORECASE),
        "apparmor_unconfined": re.compile(r"apparmor\.security\.beta\.kubernetes\.io/[^:]*", re.IGNORECASE),
        # Same-line capabilities add list
        "capabilities_add_sysadmin_inline": re.compile(
            r"capabilities\s*:\s*\b*add\s*:\s*[\w|_|\-|\.|\/|:]*\s*(SYS_ADMIN|ALL|NET_ADMIN)[^:]*", re.IGNORECASE,
        ),
        # procMount: Unmasked
        "procMount_Unmasked": re.compile(r"\bprocMount\s*:\s*Unmasked\b", re.IGNORECASE),
    }
```

을 통해 권한을 상승시키는 것을 허용해요.

runAsNonRoot: false
runAsUser: 0
runAsGroup: 0

컨테이너가 루트(UID 0) 사용자로 실행되는 것을 허용해요.

높음: 컨테이너의 권한을 가미할 수 있음
낮음: 컨테이너의 권한을 줄 수 있음

```
def _perform_scan(repo_url: str):
    if not repo_url:
        return jsonify({"error": "repo-url parameter is required"}), HTTPStatus.BAD_REQUEST

    repo_service = RepoService()
    scan_service = ScanService()

    clone_path = None
    try:
        clone_path = repo_service.clone_repo(repo_url)
        scan_result = scan_service.scan_repository(clone_path)

        response = {
            "repoUrl": repo_url,
            "stats": {
                "filesScanned": scan_result["filesScanned"],
                "findings": len(scan_result["findings"]),
            },
            "findings": scan_result["findings"],
        }
        return jsonify(response), HTTPStatus.OK
    except ValueError as exc:
        return jsonify({"error": str(exc)}), HTTPStatus.BAD_REQUEST
    except Exception as exc:
        return jsonify({"error": f"Unexpected error: {exc}"}), HTTPStatus.INTERNAL_SERVER_ERROR
    finally:
        if clone_path:
            try:
                repo_service.cleanup(clone_path)
            except Exception:
                pass
```

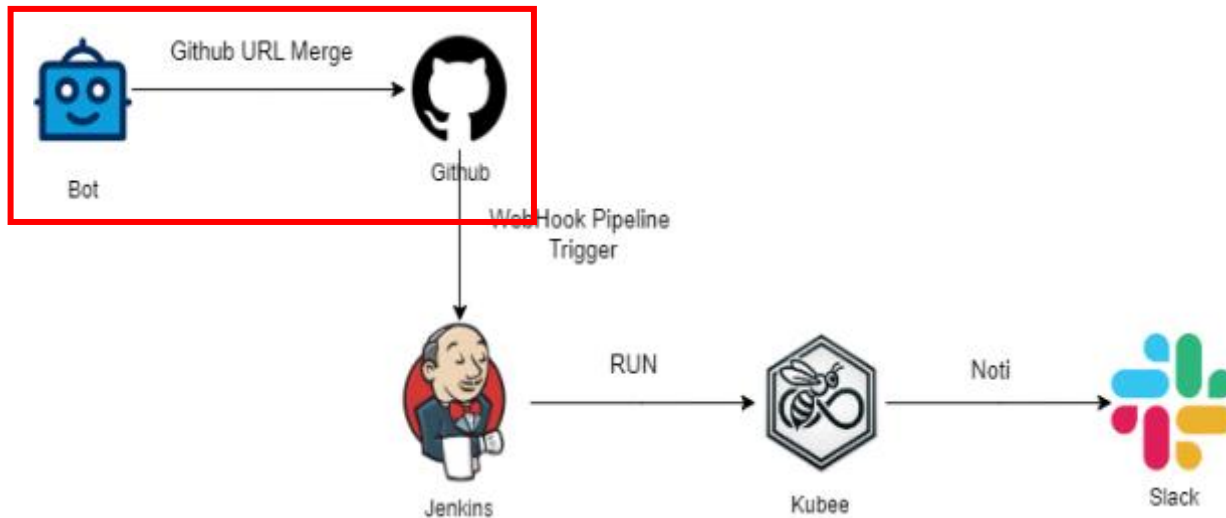
Misconfiguration Pipeline

| 정규표현식 매칭 기반의 취약점 자동 탐지 파이프라인을 구축함.

프로젝트개요

프로젝트진행결과

프로젝트 성과



| Pipeline Flow

STEP 01

타겟 오픈소스 GitHub 리포지토리 병합

STEP 02

GitHub Webhook을 통한 Jenkins 파이프라인 트리거

STEP 03

정규표현식 기반 자동화 스크립트 실행

STEP 04

실행 결과 요약 및 정렬

STEP 05

결과 Slack 알림

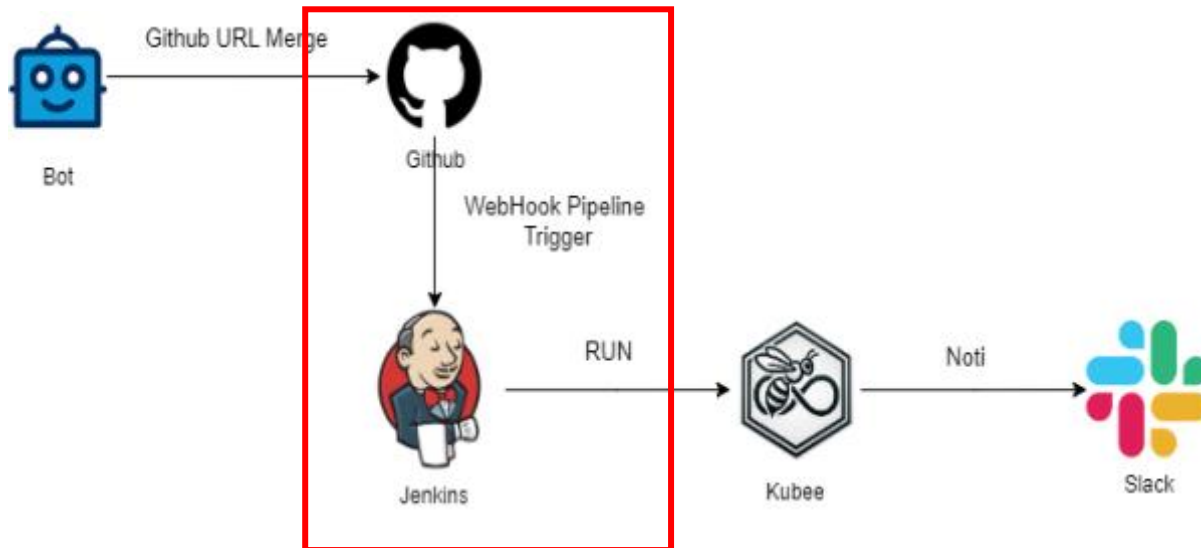
Misconfiguration Pipeline

| 정규표현식 매칭 기반의 취약점 자동 탐지 파이프라인을 구축함.

프로젝트개요

프로젝트진행결과

프로젝트 성과



| Pipeline Flow

STEP 01 ○ 타겟 오픈소스 GitHub 리포지토리 병합

STEP 02 ○ **GitHub Webhook을 통한 Jenkins 파이프라인 트리거**

STEP 03 ○ 정규표현식 기반 자동화 스크립트 실행

STEP 04 ○ 실행 결과 요약 및 정렬

STEP 05 ○ 결과 Slack 알림

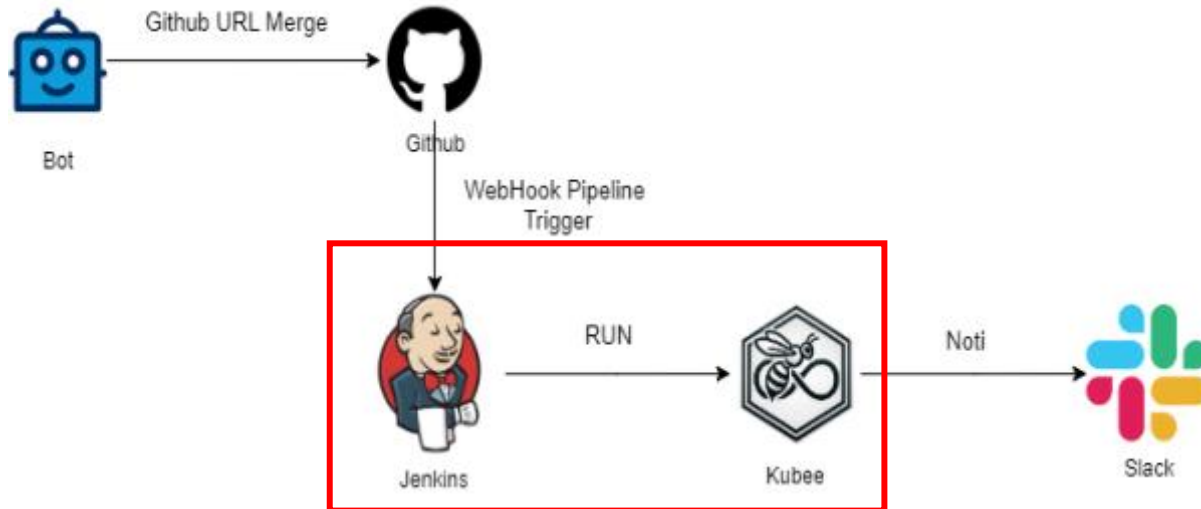
Misconfiguration Pipeline

| 정규표현식 매칭 기반의 취약점 자동 탐지 파이프라인을 구축함.

프로젝트개요

프로젝트진행결과

프로젝트 성과



| Pipeline Flow

- STEP 01 ○ 타겟 오픈소스 GitHub 리포지토리 병합
- STEP 02 ○ GitHub Webhook을 통한 Jenkins 파이프라인 트리거
- STEP 03 ○ 정규표현식 기반 자동화 스크립트 실행
- STEP 04 ○ 실행 결과 요약 및 정렬
- STEP 05 ○ 결과 Slack 알림

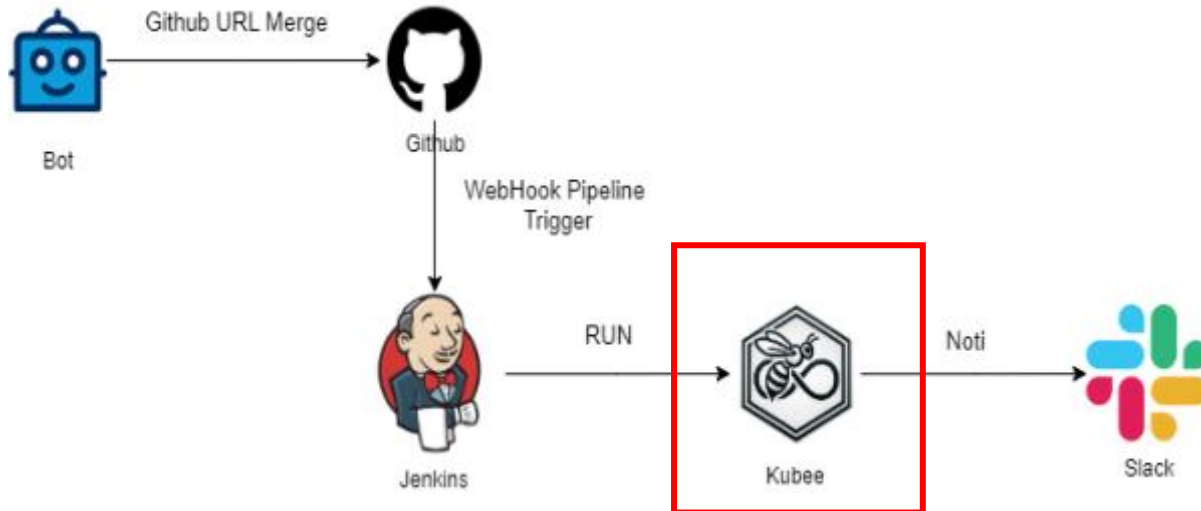
Misconfiguration Pipeline

| 정규표현식 매칭 기반의 취약점 자동 탐지 파이프라인을 구축함.

프로젝트개요

프로젝트진행결과

프로젝트 성과



| Pipeline Flow

- STEP 01 ○ 타겟 오픈소스 GitHub 리포지토리 병합
- STEP 02 ○ GitHub Webhook을 통한 Jenkins 파이프라인 트리거
- STEP 03 ○ 정규표현식 기반 자동화 스크립트 실행
- STEP 04 ○ 실행 결과 요약 및 정렬
- STEP 05 ○ 결과 Slack 알림

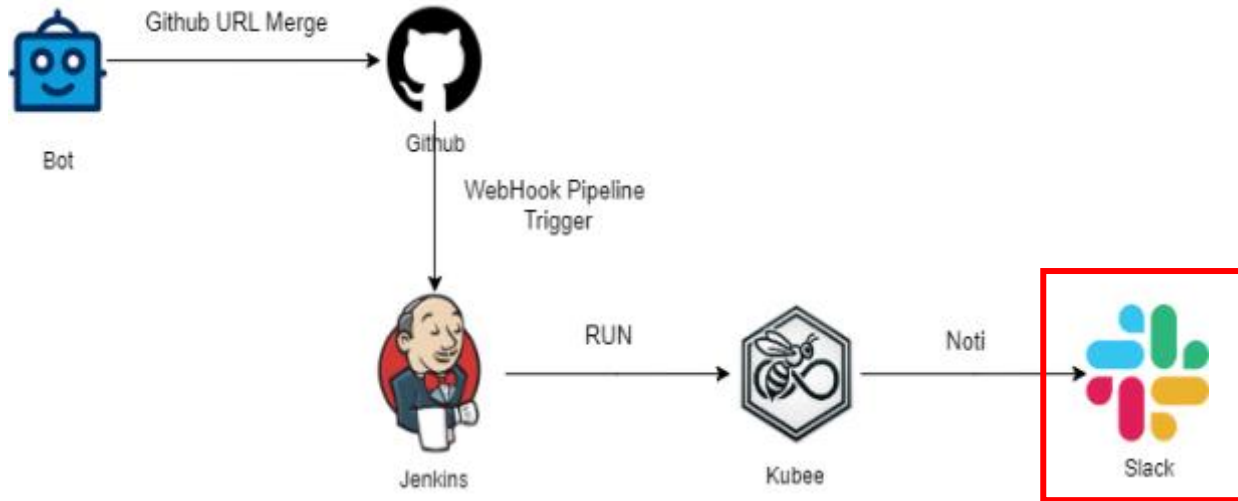
Misconfiguration Pipeline

| 정규표현식 매칭 기반의 취약점 자동 탐지 파이프라인을 구축함.

프로젝트개요

프로젝트진행결과

프로젝트 성과



| Pipeline Flow

- STEP 01 ○ 타겟 오픈소스 GitHub 리포지토리 병합
- STEP 02 ○ GitHub Webhook을 통한 Jenkins 파이프라인 트리거
- STEP 03 ○ 정규표현식 기반 자동화 스크립트 실행
- STEP 04 ○ 실행 결과 요약 및 정렬
- STEP 05 ○ 결과 Slack 알림

메시지 캔버스 추가 파일 +



김수현 오전 10:11

added an integration to this chat. 오늘 ae

김수현 오전 10:13

===== restart =====



kubee 오전 10:13

Kubernetes Misconfiguration

Date:	Time:
2025-04-30	10:13:01
GitHub URL:	Total Matches:
https://github.com/grafana/mimir	466



kubee 오전 10:20

Kubernetes Misconfiguration

Date:	Time:
2025-04-30	10:20:31
GitHub URL:	Total Matches:
https://github.com/grafana/loki	15



kubee 오전 10:26

Kubernetes Misconfiguration

Date:	Time:
2025-04-30	10:26:42
GitHub URL:	Total Matches:
https://github.com/grafana/loki	15

1개의 댓글 오늘 오전 10:26에

B I

#kubee에 메시지 보내기

+ Aa

▶ ▼



kubee 바로 지금

Kubernetes Misconfiguration

Date:	Time:
2025-04-30	10:26:42
GitHub URL:	Total Matches:
https://github.com/grafana/loki	15

1개의 댓글



kubee 바로 지금

scan_result.json ▼

```
1 {
2   "github_url": "https://github.com/grafana/loki",
3   "date": "2025-04-30",
4   "time": "10:26:42",
5   "total_matches": 15,
6   "matches": [
7     {
8       "matched_text": "readOnlyRootFilesystem: false",
9       "file": "loki/operator/config/overlays/openshift/size-
10 calculator/logfile_metric_scc.yaml",
11       "line_number": 25
12     },
13     {
14       "matched_text": "allowPrivilegeEscalation: true",
15       "file": "loki/operator/config/overlays/openshift/size-
16 calculator/logfile_metric_scc.yaml",
17       "line_number": 42
18     },
19     {
20       "matched_text": "privileged: true",
21       "file": "loki/operator/hack/addons_ocp.yaml",
22       "line_number": 99
23     }
24   ]
25 }
```

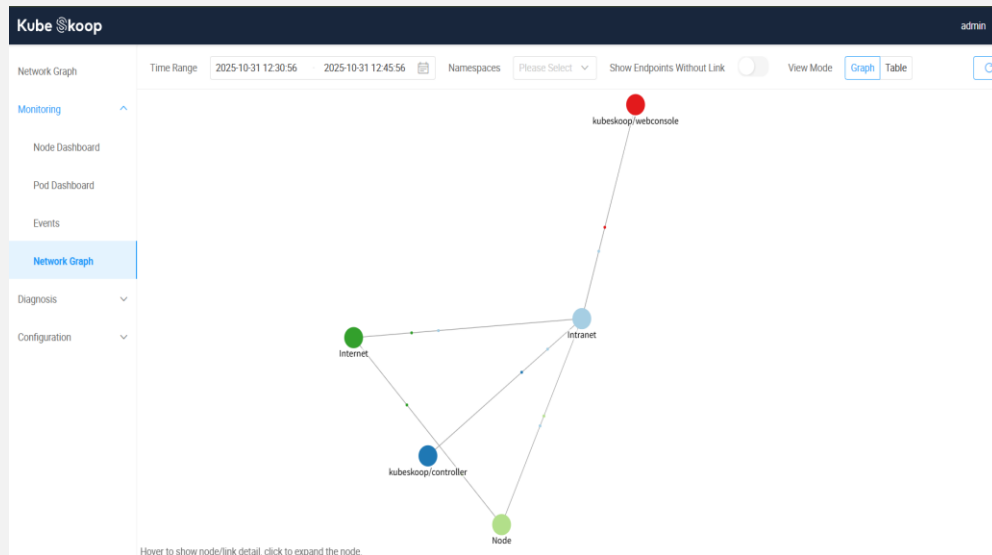
Kubescape Misconfiguration

| Kubescape 외 1개 오픈소스에서 Misconfiguration을 탐지 함.

프로젝트개요

프로젝트진행결과

프로젝트 성과



Kubescape

KubeSkoop은 '쿠버네티스'라는 복잡한 IT 환경에서
네트워크(통신) 문제가 생겼을 때 원인을 찾아주는 도구

시스템 내부의 보이지 않는 통신 흐름을 실시간으로 감시하고
연결 상태를 지도로 그려주어, 어디가 막히거나 잘못되었는지
쉽게 파악하고 고칠 수 있게 시각화

Kubescape Misconfiguration

| 주요 보안 설정 오류: hostIPC 및 AutomountServiceAccountToken을 분석 함.

프로젝트 개요

프로젝트 진행 결과

프로젝트 성과

Kubescape Misconfiguration

hostIPC : True

Pod가 호스트(노드)의 IPC(프로세스 간 통신) 공간을 직접 공유하게 하여 보안 격리를 깨뜨리는 설정

이로 인해 Pod 호스트의 민감한 정보를 엿보거나 다른 프로세스를 방해할 수 있어 매우 위험

AutomountServiceAccountToken

Pod가 생성될 때 서비스 Service Account의 토큰을 자동으로 Pod 내부에 주입할지 여부를 결정하는 옵션

이 옵션의 기본값은 true, 이 때문에 별도로 설정하지 않으면, 모든 Pod는 쿠버네티스 API 서버와 통신할 수 있는 토큰을 자동으로 발급받음

Pod가 공격자에게 탈취 당했을 경우 위험해질 수 있음

Kubescape Misconfiguration

Kubescape Misconfigutarion에 대해 증명 함.

프로젝트개요

프로젝트진행결과

프로젝트 성과

Kube Skoop

admin

Network Graph

Time Range

2025-10-31 14:25:52

2025-10-31 14:40:52



Namespaces

Please Select



Show Endpoints Without Link



View Mode

Graph

Table



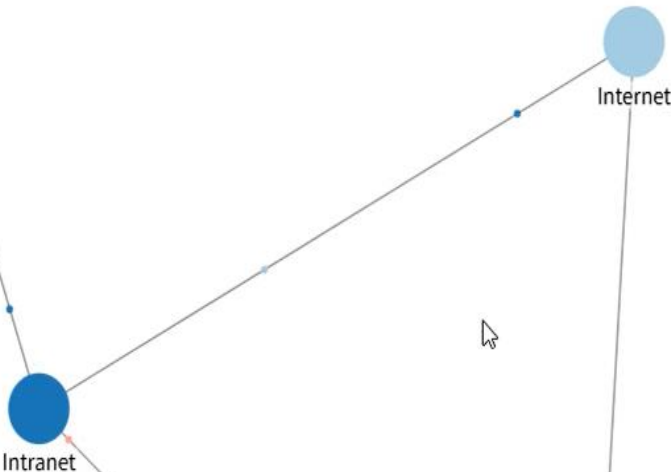
Monitoring



Diagnosis



Configuration



```
root@DESKTOP-FSUSL5A: /  
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /# kubectl get po,svc -n kubekoop  
NAME                                READY    STATUS    RESTARTS    AGE  
pod/controller-76fd7756b9-z68dv     1/1     Running   0            8m32s  
pod/grafana-5bfcb464b8-rk6vg        2/2     Running   0            8m32s  
pod/grafana-loki-58888d64b6-qbvd2   1/1     Running   0            8m32s  
pod/kubekoop-exporter-zcncd          1/1     Running   0            8m33s  
pod/prometheus-deployment-5dc88c6b94-ffz8n 1/1     Running   0            8m33s  
pod/webconsole-698c9695c4-6ppcs      1/1     Running   0            8m32s  
  
NAME                                AGE      TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)  
service/controller                  8m32s    ClusterIP      10.96.198.231 <none>         10263/TCP,1  
service/grafana                      8m32s    NodePort       10.96.116.248 <none>         80:31794/TC  
service/loki-service                 8m33s    ClusterIP      10.96.204.165 <none>         3100/TCP  
service/prometheus-service          8m33s    NodePort       10.96.86.86   <none>         80:30937/TC  
service/webconsole                   8m32s    ClusterIP      10.96.78.221  <none>         80/TCP  
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /# kubectl -n kubekoop port-forward svc/webconsole 8080:80  
Forwarding from 127.0.0.1:8080 -> 8080  
Forwarding from [::1]:8080 -> 8080  
Handling connection for 8080  
Handling connection for 8080  
Handling connection for 8080  
Handling connection for 8080  
|
```

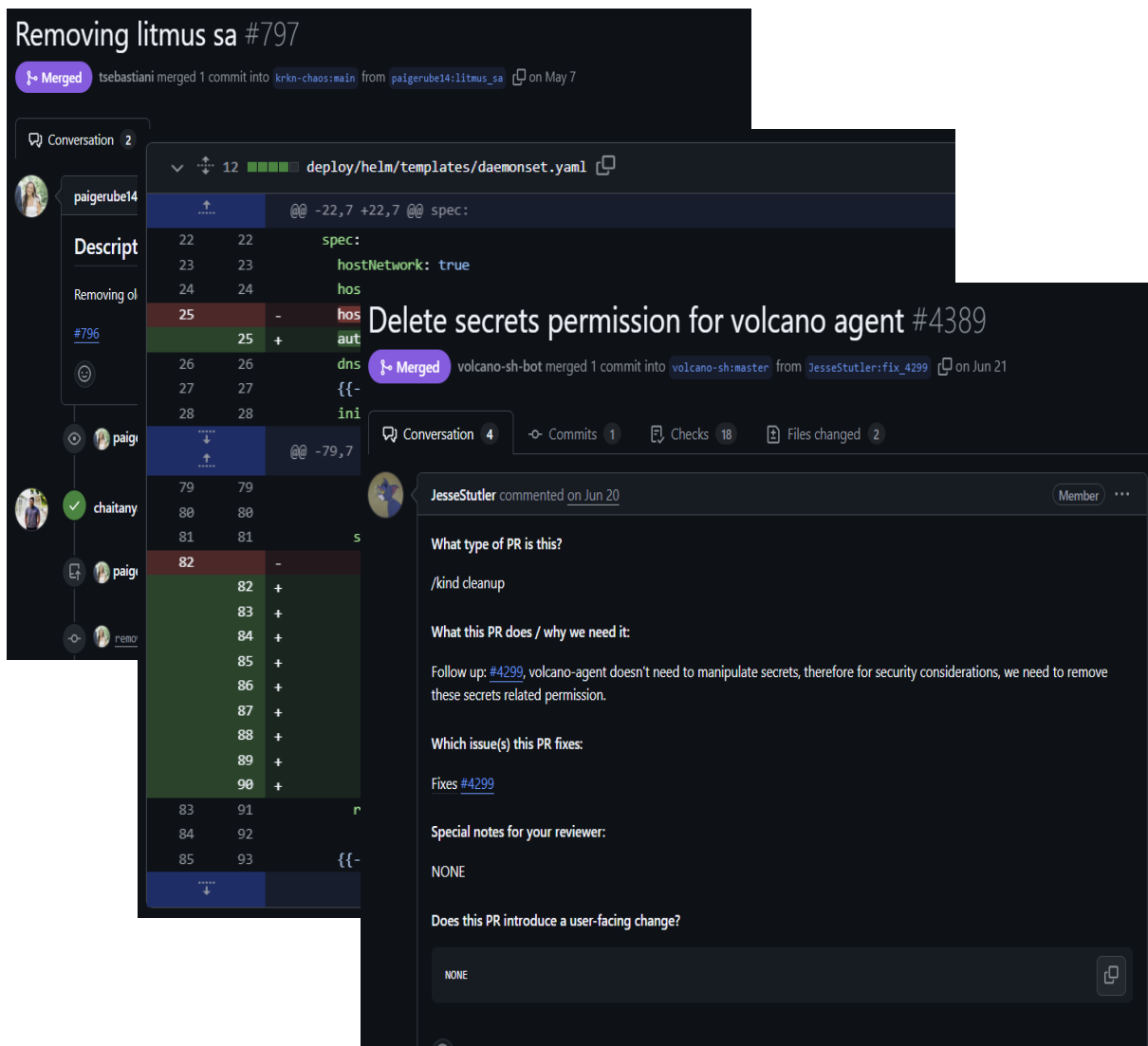
```
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /#
```

Kubescape 외 2개 오픈소스에서 Misconfiguration이 패치 됨.

프로젝트개요

프로젝트 진행 결과

프로젝트 성과



krkn, Kubescop, Volcano

Misconfiguration 제보 및 패치

LLM, RAG 기술을 활용한 취약 설정 탐지

| k8s 보안 지식베이스(RAG)를 이용해서 Misconfiguration를 구현함.

프로젝트 개요

프로젝트 진행 결과

프로젝트 성과

1.2 Security Context 설정
위험도: High 설명: 적절한 Security Context 설정이 없으면 보안 위험이 증가합니다.
권장 설정:
securityContext:
 runAs:
 runAs:
 runAs:
 fsGroup:
 seccompProfile:
 type:
 capabilities:
 drop:
 - ALL
 add:
 - NET_BIND_SERVICE

2. 네트워크 보안
2.1 호스트 네트워크 사용 금지
2.3 서비스 타입 보안
위험도: Critical
설명: NodePort 서비스는 외부에서 직접 접근 가능하므로 주의가 필요합니다.
권장 사항: Kubernetes 보안 지식베이스
• 호스트 네트워크 사용 금지
• 포트 충돌
• 네트워크 노출
1.3 워크로드 보안
위험도: Medium
설명: PodSecurityPolicy (PSP) 설정이 없으면 워크로드의 권한이 과도하게 부여될 수 있습니다.
권장 사항: Kubernetes 보안 지식베이스
• PodSecurityPolicy (PSP) 설정
• NetworkPolicy 설정
• SecurityContext 설정
방어 방법:
spec:
 hostNetwork:
 contain:
 - name:
 ports:
 - containerPort:
 - containerPort:
 - containerPort:

3. 접근 제어
3.1 RBAC 설정
위험도: High
설명: 과도한 권한이 부여된 RoleBinding이 존재하면 보안 위험이 증가합니다.
권장 설정:
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
 name:
rules:
 - apiGroups:
 - ""
 - "rbac.authorization.k8s.io"
 - "extensions"
 resources:
 - "*"
 verbs:
 - "*"
 - "create"
 - "delete"
 - "get"
 - "list"
 - "patch"
 - "update"

1. 컨테이너 보안
1.1 Privileged 컨테이너
위험도: Critical 설명: privileged=true로 설정된 컨테이너는 호스트 시스템의 모든 권한을 가집니다.
위험성:
• 호스트 파일시스템에 직접 접근 가능
• 호스트 네트워크 스택 조작 가능
• 호스트 프로세스 조작 가능
• 컨테이너 탈출 시 호스트 전체 제어 가능
방어 방법:
securityContext:
 privileged: false # 반드시 false로 설정
 runAsNonRoot: true
 runAsUser: 1000
 allowPrivilegeEscalation: false

1.2 Security Context 설정

RAG (Retrieval Augmented Generation) ?

- LLM은 학습한 데이터로만 답하기 때문에 **최신정보는 알지 못함(EX. 사내 정보)**
- LLM이 **'지식 베이스'를 먼저 검색**한 뒤, 그 결과를 바탕으로 답변을 생성하게 하는 기술

k8s Misconfiguration, RAG로 탐지하는 이유

- 각 설정별로 Critical, High 등 **사내에서 정의한 심각도를 기준**으로 답변
- 지식 베이스의 위험도, 설명, 방어 방법 구조 등 LLM이 **일관된 리포팅 형식**으로 답변
- 특정 파일의 **필수 설정을 '예외 처리'** 하여 탐지 정확도를 높이고 오탐(False Positive)을 줄임

RAG 구현에 사용되는 쿠버네티스 설정 보안 지식베이스

Kubee 서비스의 시연 영상을 제작 함.

프로젝트개요

프로젝트 진행 결과

프로젝트 성과



오픈소스 취약점 분석 연구

| k8s Misconfiguration에 국한하지 않고, 오픈소스 전반의 취약점으로 분석 범위 확장

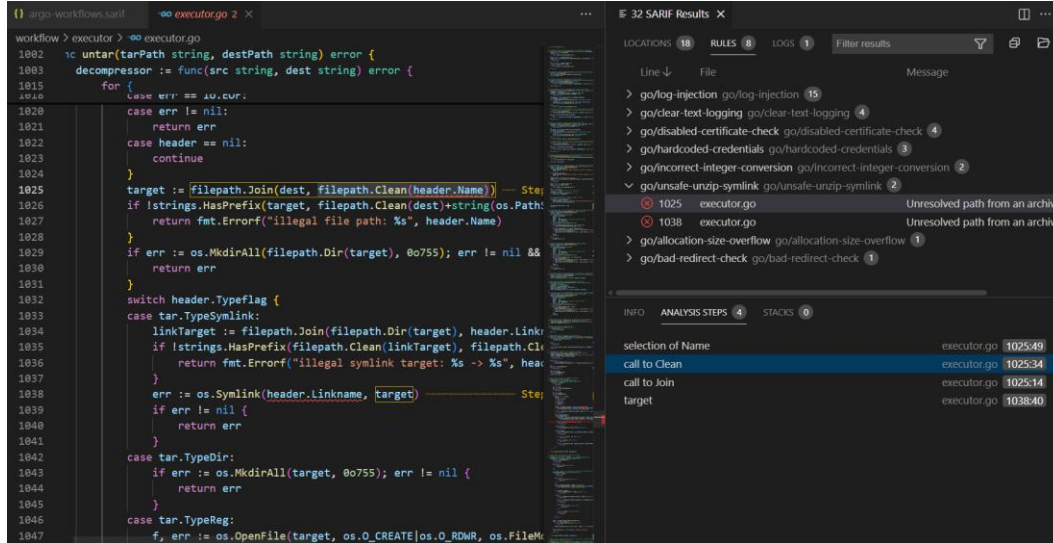
프로젝트 개요

프로젝트 진행 결과

프로젝트 성과



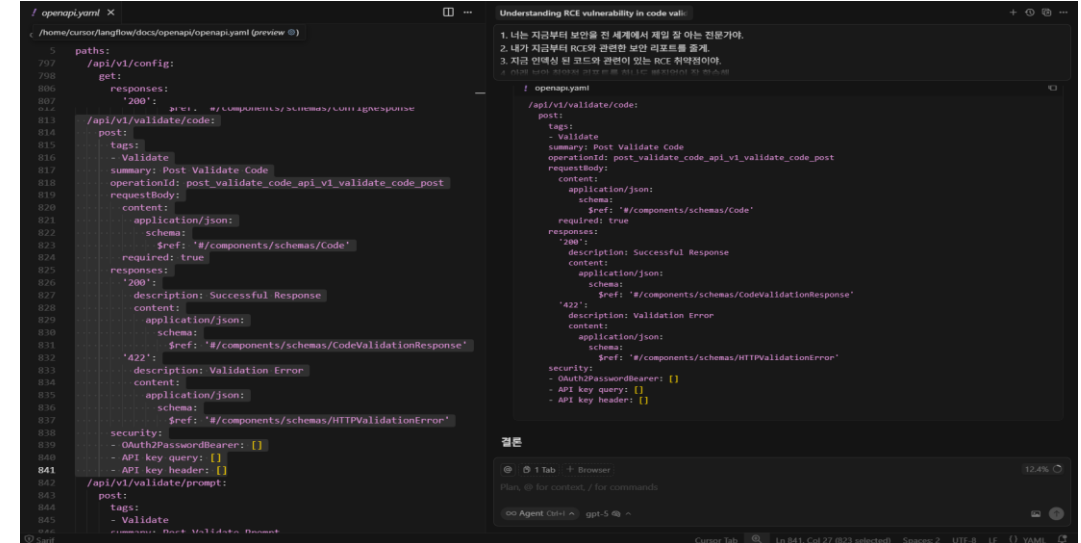
CodeQL



Taint Analysis 데이터 흐름 분석을 쿼리(Query)하여 코드의 잠재적 취약점을 찾는 SAST 도구



Cursor



인덱싱된 코드베이스 전체의 맥락을 LLM에 제공하여 코드 생성 및 편집을 수행하는 AI 네이티브 에디터

오픈소스 취약점 분석 연구

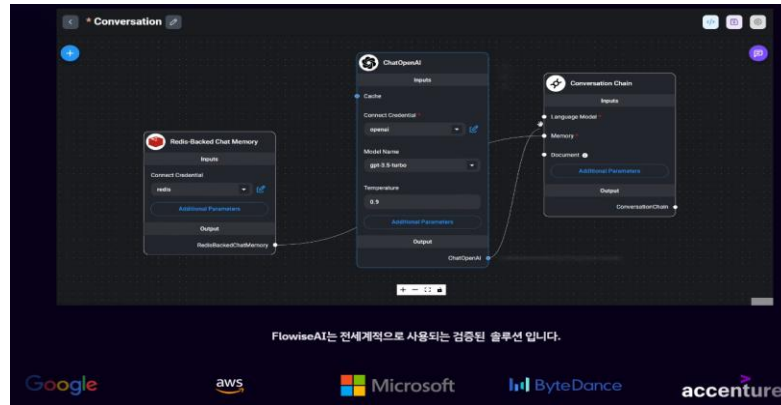
| Flowise 오픈소스에 대해 설명함.

프로젝트개요

프로젝트진행결과

프로젝트 성과

복잡한 AI 워크플로우를 시각적으로 조립 (AI Agent, RAG)



Flowise와 LLM을 활용 에러 분석 자동화 - 토스



Flowise와 LLM을 활용한 에러 분석 자동화

AI Workflow



국내 활용 사례

Remote Code Excuen in Flowise [CVE-2025-59528]



```
function convertToValidJSONString(inputString: string) {  
  try {  
    const jsObject = Function('return ' + inputString)()  
    return JSON.stringify(jsObject, null, 2)  
  } catch (error) {  
    console.error('Error converting to JSON:', error)  
    return ''  
  }  
}
```

오픈소스 취약점 분석 연구

| Flowise RCE 취약점을 PoC 함.

프로젝트개요

프로젝트진행결과

프로젝트 성과

```
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /home/soohyun/cursor/Flowise#  
root@DESKTOP-FSUSL5A: /home/soohyun/cursor/Flowise# pnpm s  
N Unsupported engine: wanted: {"node": ">=18.15.0 <19  
|| ^20"} (current: {"node": "v22.19.0", "pnpm": "10.18.  
owise@3.0.5 start /home/soohyun/cursor/Flowise  
n-script-os  
owise@3.0.5 start:default  
packages/server/bin && ./run start  
-10-31 11:09:28 [INFO]: Starting Flowise...  
-10-31 11:09:28 [INFO]: 🍌 [server]: Data Source init  
zed successfully  
-10-31 11:09:28 [INFO]: 🗄️ [server]: Database migrati  
completed successfully  
-10-31 11:09:28 [INFO]: 🛡️ [server]: Identity Manager  
tialized successfully  
-10-31 11:09:31 [INFO]: 🐳 [server]: Nodes pool initi  
ed successfully  
-10-31 11:09:31 [INFO]: 🛑 [server]: Abort controller  
ol initialized successfully  
-10-31 11:09:31 [INFO]: 🔑 [server]: Encryption key i  
alized successfully  
-10-31 11:09:32 [INFO]: 🚦 [server]: Rate limiters in  
lized successfully  
-10-31 11:09:32 [INFO]: 🗄️ [server]: Cache pool initi  
ed successfully  
-10-31 11:09:32 [INFO]: 🗄️ [server]: Usage cache mana  
initialized successfully  
-10-31 11:09:32 [INFO]: 📧 [server]: Telemetry initia  
d successfully  
-10-31 11:09:32 [INFO]: 🗣️ [server]: SSE Streamer ini  
ized successfully  
-10-31 11:09:32 [INFO]: 🌈 [server]: All initializati  
steps completed successfully!  
-10-31 11:09:32 [INFO]: 🔥 [server]: Flowise Server i  
stening at :3000
```

SERVER

```
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /#
```

REVERSE SHELL

```
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /#  
root@DESKTOP-FSUSL5A: /#
```

RCE TRIGGER

오픈소스 취약점 분석 연구

| SK실더스에서 Lobe Chat SSRF 취약점을 연구함.

프로젝트개요

프로젝트진행결과

프로젝트 성과

SK 실더스 Lobe-Chat SSRF 취약점 연구

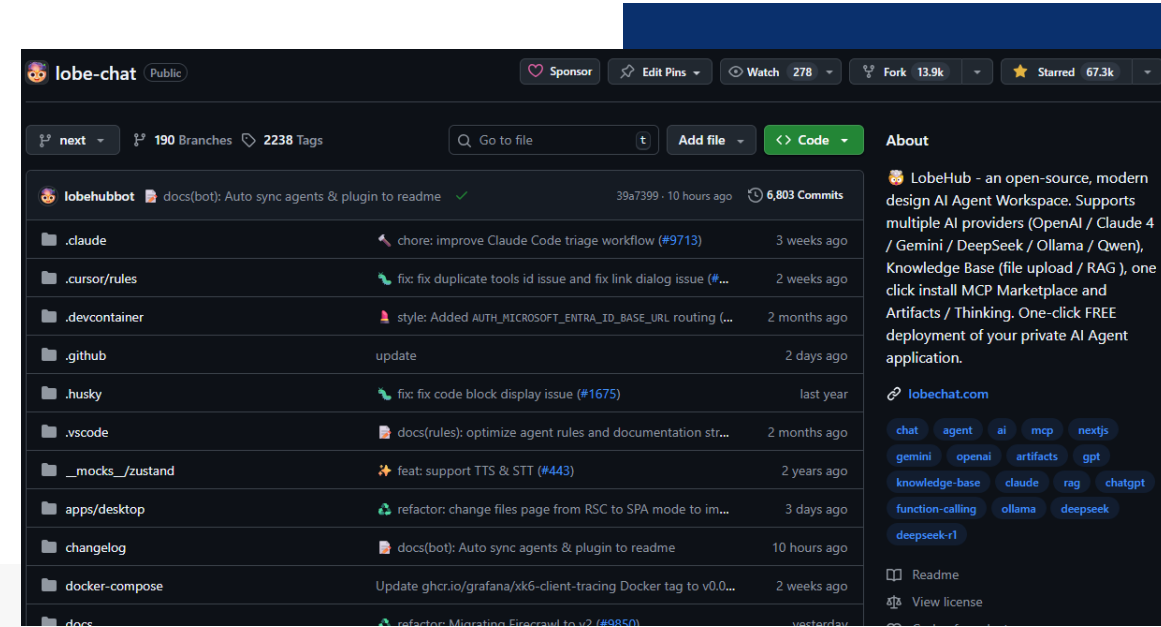
CVE-2024-47066



취약점

[Research & Technique] Lobe Chat SSRF 취약점(CVE-2024-47066)

2024-10-28



오픈소스 챗봇 UI 프레임워크
SATAR 67.3k

오픈소스 취약점 분석 연구

| Lobe-Chat 오픈소스에서 취약점을 찾고 Patch Contribution 함.

프로젝트개요

프로젝트진행결과

프로젝트 성과

SSRF in lobehub/lobe-chat with native web fetch module

Published **Low** arvinxx published GHSA-fgx4-p8xf-qhp9 2 weeks ago · 10 comments

Package	Affected versions	Patched versions
web-crawler (LobeChat)	1.136.1	1.136.2

Severity
Low 3.0 / 10

im-soohyun opened on Sep 15 · edited · ...

Description

Vulnerability Description

Vulnerability Overview

- When the client sends an arbitrary urls array and impls: ["naive"] to the tRPC endpoint tools.search.crawlPages, the server issues outbound HTTP requests directly to those URLs. There is no defensive logic that restricts or validates requests to internal networks (127.0.0.1, localhost, private ranges) or metadata endpoints (169.254.169.254).
- Flow: client input (urls, impls) → service invocation in the tRPC router → the service passes the URLs to Crawler.crawl → the Crawler prioritizes the user-specified impls (naive) → the naive implementation performs a server-side fetch(url) as-is (SSRF) → the server collects responses from internal resources.
- In the dev environment, authentication can be bypassed using the lobe-auth-dev-backend-api: 1 header (production requires a valid token). In the PoC, this was used to successfully retrieve the internal API at localhost:8889 from the server side.

Vulnerable Code

lobe-chat/packages/web-crawler/src/crawlImpl/naive.ts
Lines 39 to 45 in d942a63

```
39 res = await withTimeout(  
40 fetch(url, {
```

CVSS v3 base metrics	
Attack vector	Network
Attack complexity	High
Privileges required	High
User interaction	None
Scope	Changed
Confidentiality	Low
Integrity	None
Availability	None
Learn more about base metrics	

CVSS:3.1/AV:N/AC:H/PR:H/UI:N/SC:C/L:N/H:N/A:N

CVE ID

CVE-2025-62505

Weaknesses

► CWE-918

Credits

im-soohyun

Reporter ✓

```
packages/web-crawler/package.json 100644 → 100755 @@ -12,6 +12,7 @@  
12 12     "@mozilla/readability": "^0.6.0",  
13 13     "happy-dom": "^18.0.0",  
14 14     "node-html-markdown": "^1.3.0",  
15 15 +    "ssrf-safe-fetch": "workspace:*",  
15 16     "query-string": "^9.1.1",  
16 17     "url-join": "^5"  
17 18   }  
...  
  
packages/web-crawler/src/crawlImpl/naive.ts 100644 → 100755 @@ -1,4 +1,5 @@  
1 1   import { CrawlImpl, CrawlSuccessResult } from '../type';  
2 2 + import { ssrfSafeFetch } from 'ssrf-safe-fetch';  
2 3   import { NetworkConnectionError, PageNotFoundError, TimeoutError } from '../utils/errorType';  
3 4   import { htmlToMarkdown } from '../utils/htmlToMarkdown';  
4 5   import { DEFAULT_TIMEOUT, withTimeout } from '../utils/withTimeout';  
...  
@@ -37,7 +38,7 @@ export const naive: CrawlImpl = async (url, { filterOptions }) => {  
37 38  
38 39   try {  
39 40     res = await withTimeout(  
40 40 -     fetch(url, {  
41 41 +     ssrfSafeFetch(url, {  
41 42       headers: mixinHeaders,  
42 43       signal: new AbortController().signal,  
43 44     })),  
...  
+ ...
```

SSRF in Lobe-Chat (CVE-2025-62505)

Lobe-Chat SSRF Patch Contribution

오픈소스 취약점 분석 연구

| HackerOne BugBounty에 참여하여 2건의 취약점을 인정받음.

프로젝트개요

프로젝트진행결과

프로젝트 성과

```
root@DESKTOP-FSUSL5A: /home/soohyun/etc#  
root@DESKTOP-FSUSL5A: /home/soohyun/etc#
```

3. Path Traversal (read): Read `/etc/passwd` by escaping the base path via parent-directory traversal.

<> Code · 376 Bytes

```
1 docker run  
2 fullstorydev  
3 -d '{"indexN  
4 localhost:50  
5 | jq -r '.da
```

#3385934

ZipSlip in weaviate/weaviate

[ADD HACKER SUMMARY](#)

TIMELINE · EXPORT

Image · 79.

```
root@DESKTOP-F  
fullstorydev  
-d '{"indexN  
localhost:50  
| jq -r '.da  
root:x:0:0:root  
bin:x:1:1:bin:  
daemon:x:2:2:d  
lp:x:4:7:lp:/V  
sync:x:5:0:syn  
shutdown:x:6:0  
halt:x:7:0:hal  
mail:x:8:12:ma  
news:x:9:13:ne  
uucp:x:10:14:u  
cron:x:16:16:c  
ftp:x:21:21:/  
sshd:x:22:22:s  
games:x:35:35:  
ntp:x:123:123:  
guest:x:405:10  
nobody:x:65534  
root@DESKTOP-F
```



soohyun submitted a report to [Weaviate VDP](#).

October 16, 2025, 6:41am UTC

Vulnerability Overview

- This vulnerability is a Zip Slip during backup restore: files are created by blindly concatenating each gzip+tar archive entry name (`tar.Header.Name`) with the restore root path. An attacker can set an entry name to an absolute path (e.g., `/etc/...`) or use parent directory traversal (`../../../../`) to escape the restore root and create/overwrite files at arbitrary locations within the application's privilege scope.
- The core issue is that the restore logic fails to do the following: reject absolute paths and `..` segments; normalize the path and then verify that the resolved real path stays within the base directory; and block link entries (tar symlinks/hardlinks). Note that a simple `filepath.Join` is not a security filter—if the input is an absolute path, the base path is ignored.

Vulnerable Code

- There's no validation on `header.Name` for absolute paths, parent-directory (`..`) segments, or link entries, and because the code only uses `filepath.Join` to combine paths, an attacker can escape the restore root.

<https://github.com/weaviate/weaviate/blob/2a5ba1ec68fe2d0bb69b85ccf30d8bd07ef7cfa5/usecases/backup/zip.go#L241-L262>

<> Code · 621 Bytes

```
1 // target file  
2 target := filepath.Join(u.destPath, header.Name)  
3 switch header.Typeflag {  
4 case tar.TypeDir:  
5     if err := os.MkdirAll(target, 0o755); err != nil {  
6         return written, fmt.Errorf("crateDir %s: %w", target, err)  
7     }
```

HackerOne BugBounty

2건의 취약점 제보 및 Accept

03 프로젝트 성과

- k8s Misconfiguration
- 오픈소스 취약점 제보
- 향후 계획

k8s Misconfiguration

쿠버네티스 설정 점검 솔루션 개발 및 논문작성, 3건의 설정 오류를 제보를 하였음.

프로젝트개요

프로젝트진행결과

프로젝트성과

쿠버네티스 설정 점검 솔루션 및 학회 논문

The screenshot shows a web application interface for Kubernetes security analysis. It features a dashboard with four colored boxes (blue, red, orange, green) representing different security metrics. Below the dashboard is a line graph showing trends over time. The main content area displays the title of a research paper, "Kubernetes RBAC 관계 분석을 통한 보안 위협 식별 방법론", along with the authors' names and affiliations. The abstract discusses the importance of RBAC in Kubernetes security and the methodology used for analysis. A table of keywords is also provided.

보안 통계 분석

한국융합보안학회 2025년 하계학술대회 논문집

Kubernetes RBAC 관계 분석을 통한 보안 위협 식별 방법론

신재형^{1*}, 김건희¹, 김수현¹, 남지우¹, 어영민¹, 전유경¹, 김민수²
충부대학교^{1,2}

Methodology for Identifying Security Threats with Kubernetes RBAC Relationship Analysis

JaeHyeong Shin^{1*}, GeonHee Kim¹, SooHyun Kim¹, JiWoo Nam¹, YoungMin Eo¹, YuKyung Jeon¹, Minsu Kim²

요약 : 클라우드 네이티브 환경에서 쿠버네티스 매니페스트 파일의 설정 오류는 클러스터 전체에 영향을 미치는 심각한 보안 위협으로 이어질 수 있다. 특히 RBAC과 워크로드의 보안 설정이 적절히 구성되지 않을 경우, 권한 상승이나 내부 리소스 탈취 등의 공격으로 확산될 수 있다. 본 연구에서는 RBAC 권한 구성과 워크로드 보안 설정 간의 종속 관계를 정적으로 분석할 수 있는 절차를 정의하고, 이를 통해 클러스터 내 잠재적인 보안 취약점을 사전에 식별하고 시각화할 수 있는 정적 분석 기반의 방법론을 제안한다.

Key Words : 쿠버네티스, RBAC, Misconfiguration, 쿠버네티스 보안, 쿠버네티스 보안 설정 오류

1. 서론

클라우드 네이티브 환경의 확산에 따라 컨테이너 오케스트레이션 플랫폼인 쿠버네티스가 활발히 도입되고 있다. 쿠버네티스 환경에서는 워크로드 매니페스트 파일을 통해 보안 구성을 정의하며, 리소스 접근 제어에는 RBAC(Role-Based Access Control) 메커니즘이 주로 사용된다. 이처럼 쿠버네티스는 편리하고 유연한 기능을 제공하지만, 매니페스트 파일의 구성 오류는 심각한 보안 취약점으로 이어질 수 있다[1].

(그림 2) CuihaES Code Patch

3건의 Misconfiguration 제보 및 패치

The screenshot shows a GitHub repository for 'volcano' with a pull request titled 'Removing litmus sa #797'. The pull request is merged and shows a diff of the 'deploy/helm/templates/daemonset.yaml' file. The diff highlights changes to the 'securityContext' and 'resources' sections of the 'litmus' daemonset. The changes include setting 'privileged: true', 'capabilities: [ALL]', and 'resources: {}'.

volcano Public

Removing litmus sa #797

krkn

Kubescoop

```
@@ -22,7 +22,7 @@ spec:
  spec:
    hostNetwork: true
    hostPID: true
    hostIPC: true
    automountServiceAccountToken: false
    dnsPolicy: ClusterFirstWithHostNet
    {{- if .btHack.enabled }}
    initContainers:
      - server
      - d
    securityContext:
      privileged: true
      capabilities:
        add:
          - NET_ADMIN
          - SYS_PTRACE
          - SYS_ADMIN
          - SYS_RESOURCE
          - DAC_OVERRIDE
      drop:
        - ALL
    resources:
      {{- toYaml .resources | nindent 12 }}
    {{- with .nodeSelector }}
```

오픈소스 취약점 제보

| 오픈소스 취약점 분석 결과를 나열함.

프로젝트개요

프로젝트진행결과

프로젝트성과

CVE

CVE 9건 발급

CVE-2025-59528
CVE-2025-59934
CVE-2025-59527
CVE-2025-62505
CVE-2025-62156
CVE-2025-61687
CVE-2025-58044
CVE-2025-62379
CVE-2025-59426

Reserved

CVE 6건 패치 및 발급대기

Argo CD
(Open Redirect)

Gateway
(SSRF)

Activepieces
(Blind SSRF)

Directus
(Open Redirect)

Rocket.Chat
(Open Redirect),
(Blind SSRF)

BugBounty

버그바운티 2건 CVE 발급 대기

Weaviate
(ZipSlip)

Weaviate
(Path Injection)

Pending

취약점 10건 응답대기

Onlook(3건)

Langflow(3건)

Payload (2건)

Dify (1건)

Node-RED (1건)

대한민국 보안 발전 기여

각 기업 담당자의 보안 역량 성장이 한국 기업의 안전성을 높이고,
나아가 대한민국의 보안을 강화하는 기반이 됨.

클라우드 등 다양한 분야로 공부확장

클라우드, 개인정보보호법, ISMS, LLM 등 최신 IT 및 보안
분야로 학습 확장



멘토

보안 분야의 다양한 실무 경험을 바탕으로, 후배들을
양성하고 이끄는 멘토로 성장

기업보안 담당자가 되기 위한 추가 공부

기업 보안 실무를 위한 SAST/DAST 관점의 취약점 분석 역량 강화

2025. 11. 04.

졸업 프로젝트 최종 발표

하이브리드 환경에서 DevSecOps 구축 및 공개형 Kubernetes 설정 점검 솔루션 개발

Thank You

지도교수 : 유승재 교수님

김건희, 김수현, 남지우, 신재형, 어영민, 전유경

