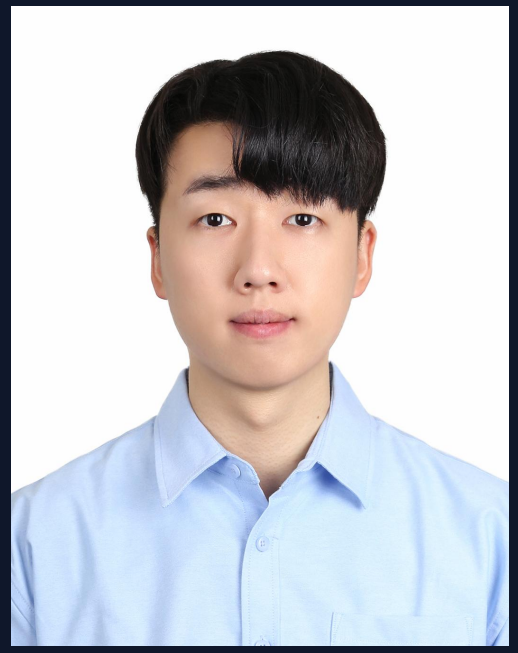
밑바닥부터 시작하는 스타트업 보안

SVN to 3S (Secret Scanning, SAST, SCA)

Jeongcheol Kang | Security Engineer

딸깍의 시대

발표자



강정철 (Jeongcheol Kang)

Enterprise Security Engineer @ Next Securities

- 전) 뱅크샐러드 현) 넥스트증권
- 물리학 / 컴퓨터공학 / 법학 전공
- PIA, CPPG, 정보보안기사, S/W 보안약점 진단원

  하는 일 : 클라우드 보안, 개인정보보호, 보안정책, 엔포도 쪼금, SIEM,Paloalto 도입하고 Confluence, Github도 관리.....

요약 : 잡부(Job富)

Enterprise Security Engineer

Network Security

방화벽, 네트워크 스위치 통제 및 L7 보안 정책 관리

CERT

SIEM 운영, 패킷 분석 및 실시간 위협 관제 대응

Cloud Security

AWS/Azure 보안 설정 및 IaC 기반 보안 자동화

Privacy Manager

개인정보보호법 준수 및 안전성 확보조치 이행

Red Team

리버싱, 프록시 분석 및 소스코드 취약점 진단

Endpoint Solution

DLP, MDM 솔루션 도입 및 보안 워크플로우 자동화

COMPANY OVERVIEW: 넥스트증권

29년+업력 증권사

1997년 현대선물로 출발해
파생상품 브로커리지부터
AI 기반 리테일 사업 확장 추진 중

금융 & 테크를 결합한 팀

넥스트증권
Google
TikTok
HYBE
McKinsey & Company
토스증권

AI 중심의 차별화된 플랫폼 전략

전략적 투자 유치

InteractiveBrokers

국내 증권사의 해외 상장 금융사
투자 유치 최초 사례

글로벌 제휴 성사

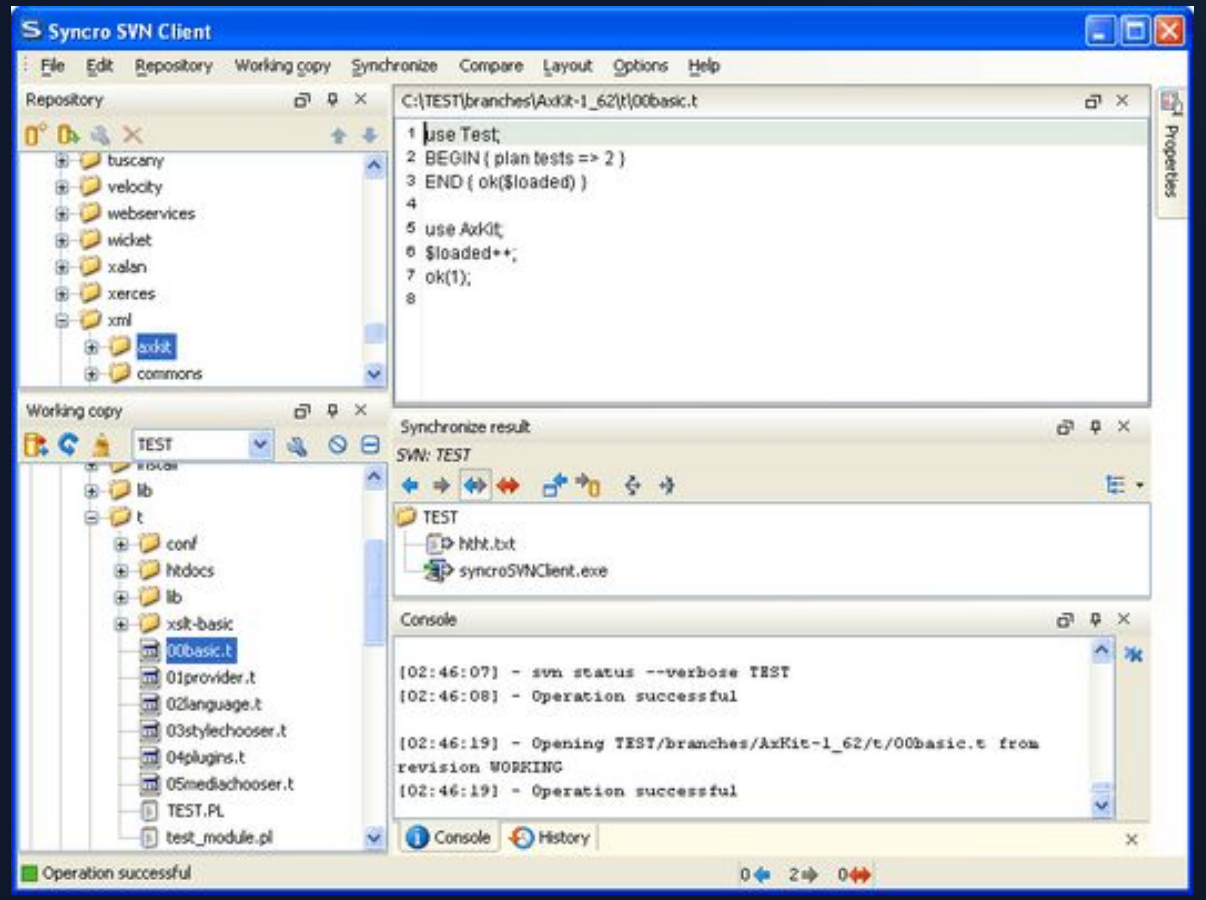
siebert.

국내 증권사 최초 미(≣) 나스닥
상장 증권사와의 전략적 제휴 사례

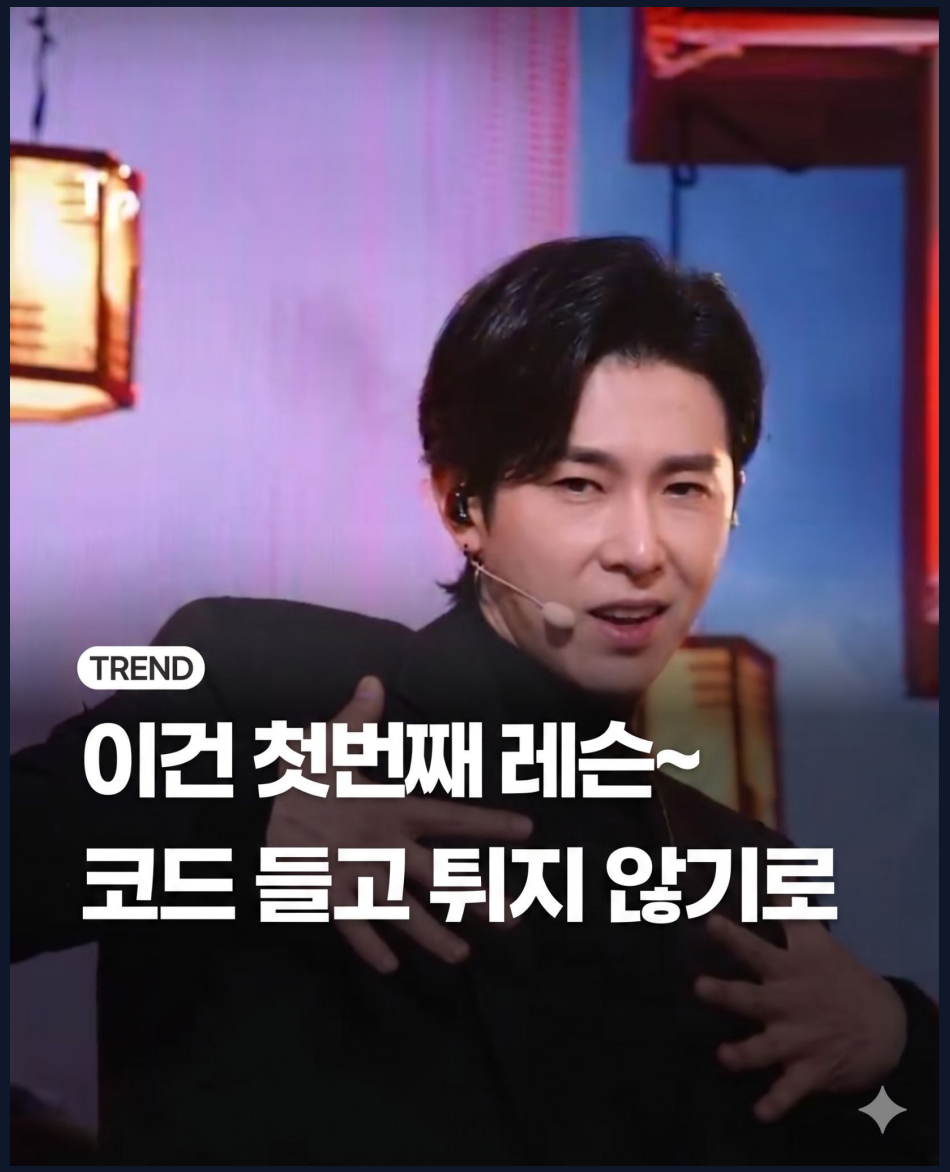
하지만 현실은....

전통적인 금융 인프라의 잔재인 **SVN(Subversion)** 중심의 개발 환경은 현대적인 보안 요구사항을 충족하기 어려웠습니다.

- 코드 변경에 대한 가시성 부족
- 수동적인 코드 리뷰 및 승인 프로세스
- 중앙 집중식 권한 관리의 한계
- CI/CD 보안 파이프라인 연동의 부재
- 개발자의 코드유출



밥(소스코드)을 지켜라



간장게장은 밥도둑이 아닙니다. [60]

유근
아바타/쪽지/글검색

2012-02-27 14:56
추천 663 반대 6 조회 62,619

제가 밥이랑 간장게장을 같이두고
쭉 지켜봤는데.
간장게장은 그냥 가만히 있을뿐.
밥을 훔치지 않았어요.
반론 하실분 말씀해보시죠.

우수 답변 가장 많은 추천을 받은 답변입니다^^

이건뭐병슨도아니고 2012-02-27 12:27 추천 663 반대 6

[redacted] 지켜보니까 못훔쳐가지

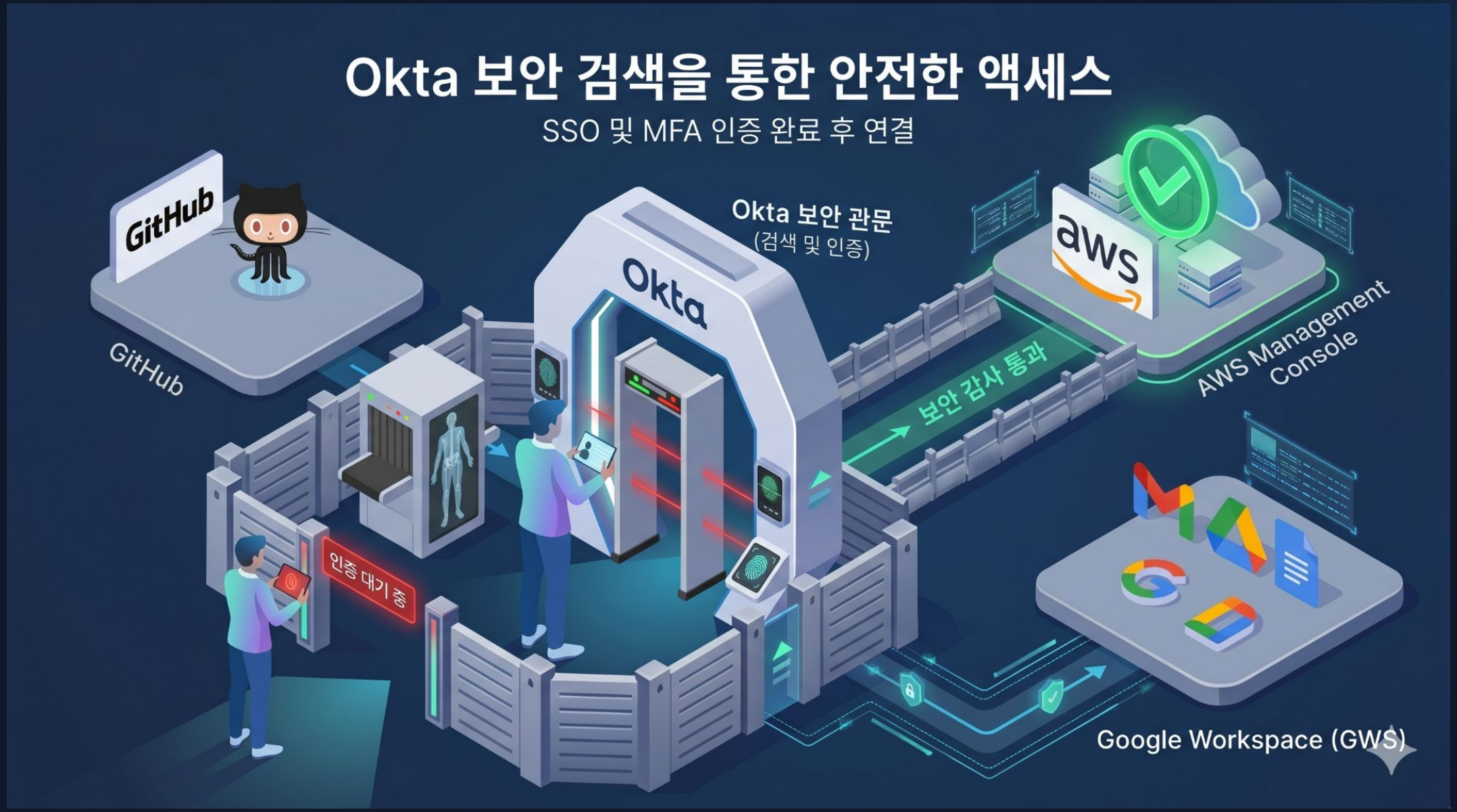
추천 반대

Quiz

가장 먼저 도입한 두 가지 솔루션은?

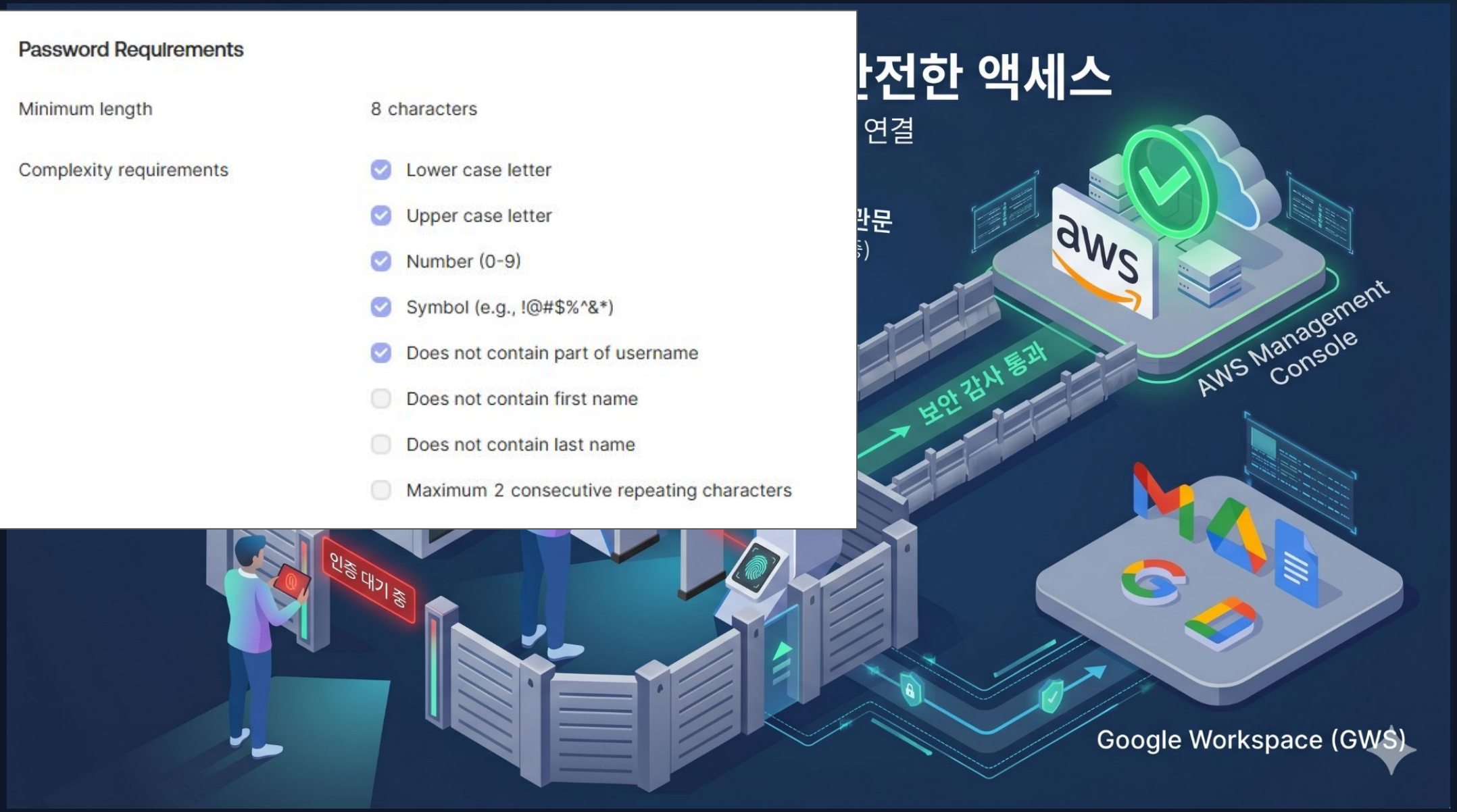
Quiz

가장 먼저 도입한 두 가지 솔루션은?


Okta 보안 검색을 통한 안전한 액세스
SSO 및 MFA 인증 완료 후 연결
GitHub
GitHub
Okta 보안 관문
(검색 및 인증)
Okta
aws
AWS Management Console
보안 감사 통과
인증 대기 중
Google Workspace (GWS)

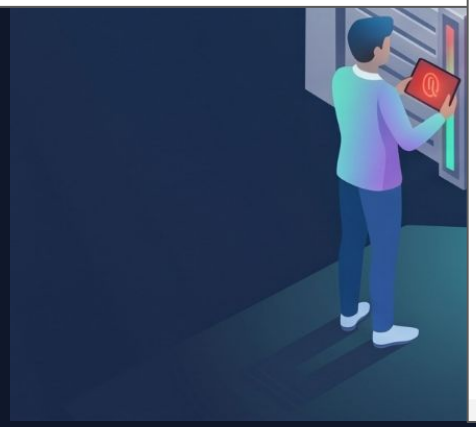
Password Requirements

Minimum length	8 characters
Complexity requirements	☑ Lower case letter
	☑ Upper case letter
	☑ Number (0-9)
	☑ Symbol (e.g., !@#$%^&*)
	☑ Does not contain part of username
	☐ Does not contain first name
	☐ Does not contain last name
	☐ Maximum 2 consecutive repeating characters



Password Requirements

Minimum length	8 characters
Complexity requirements	☑ Lower case letter
	☑ Upper case letter
	☑ Number (0-9)
	☑ Symbol (e.g., !@#$%^&*)



Priority	Rule	Status	Actions
1	Catch-all Rule	ENABLED	Actions

IF Any request THEN **Access:** Allowed with password + another factor

Your org's authenticators that satisfy this requirement:

Password

AND

Additional factor types

Okta Verify - TOTP or Okta Verify - Push or Okta Verify - FastPass

Your org allows users to verify their identity with a knowledge factor (Password) before the possession factor. To change this, protect against password-based attacks in Security > General

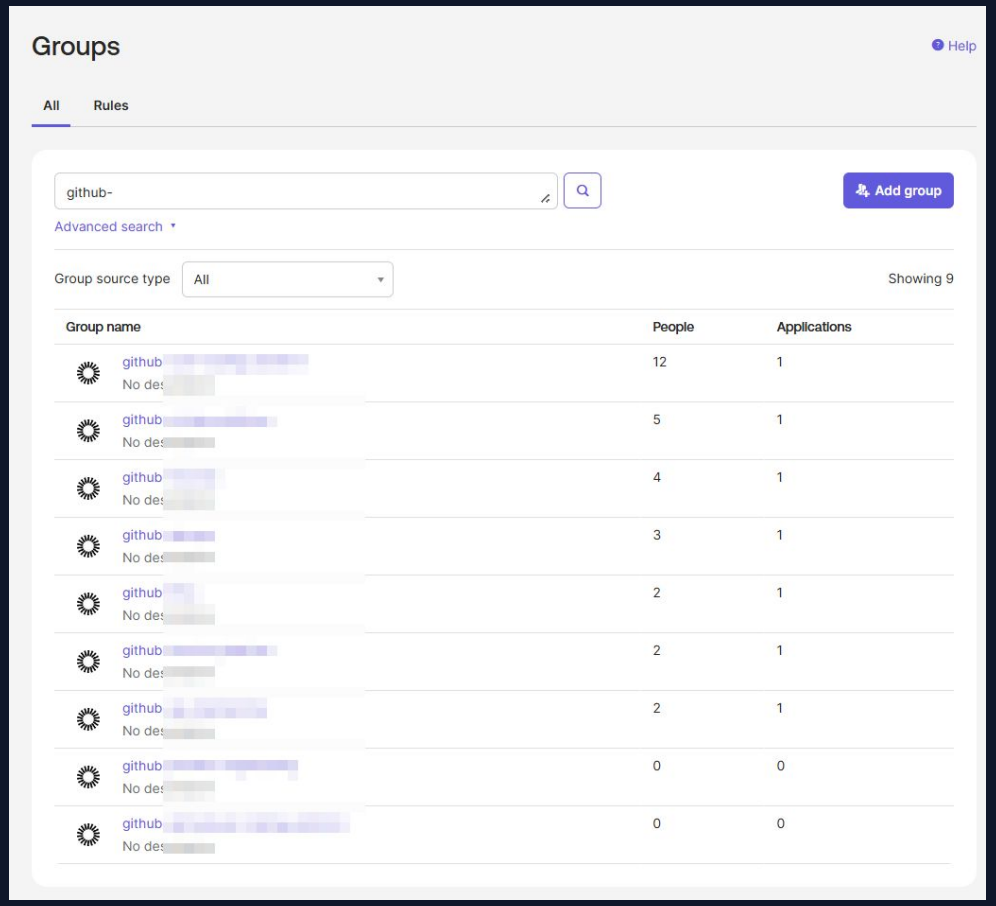
Possession factor constraints: Require user interaction

Authentication methods: Allow specific authentication methods

Password re-authentication frequency is: Every time user signs in to resource

Other authenticator re-authentication frequency: Every time user signs in to resource

Github Enterprise Managed User

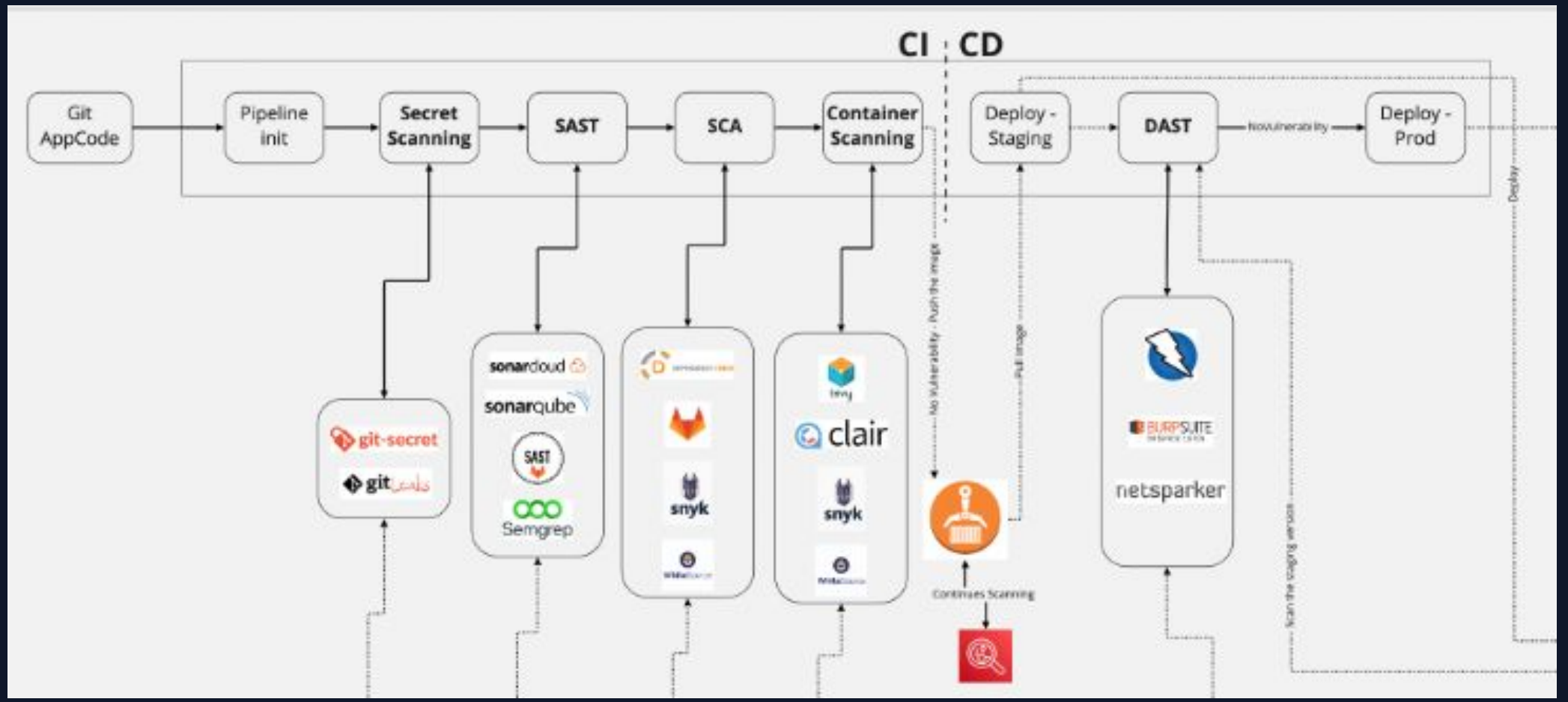


Github Enterprise Managed User의 장점

- 모든 계정을 Okta에서관리
- Okta가 Github의 단일 진입 지점
- 팀도 Okta에서 관리

STRATEGY: SHIFT LEFT SECURITY

보안은 개발의 마지막 단계가 아닌, **가장 초기 단계(Shift Left)**에서부터 통합되어야 합니다.



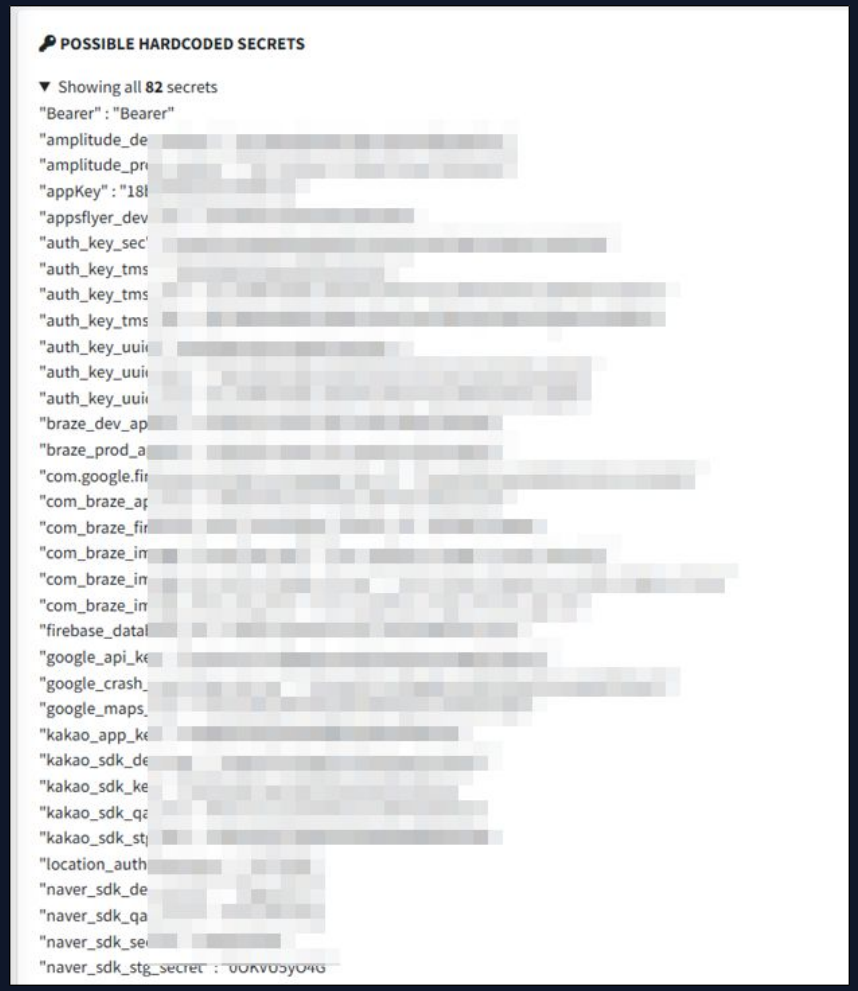
THE 3S STRATEGY: SELECTION & FOCUS

구분	Secret Scanning	SAST	SCA
대상	API Key, Passwords	Source Code	Open Source Libs
영향도	100 (매우 높음)	50 (중간)	75 (높음)
조치 난이도	0 (매우 쉬움)	10 (쉬움)	100 (어려움)
도입 순서	1순위	2순위	3순위

핵심 결론: 영향도는 크지만 조치 난이도가 낮은 **Secret Scanning**부터 도입

여담

다음을 본 개발자의 반응은?



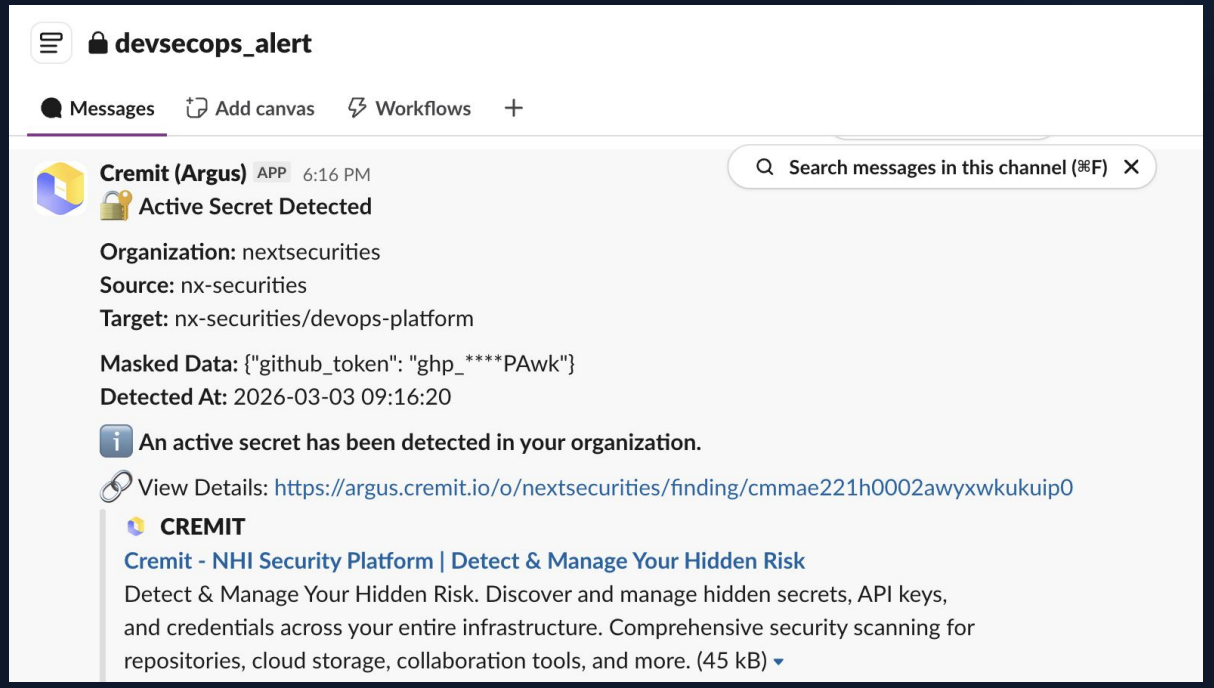
1. 키 리스트 주세요.
2. iOS인가요?
   Android인가요?

SECRET SCANNING: REAL-TIME VIGILANCE

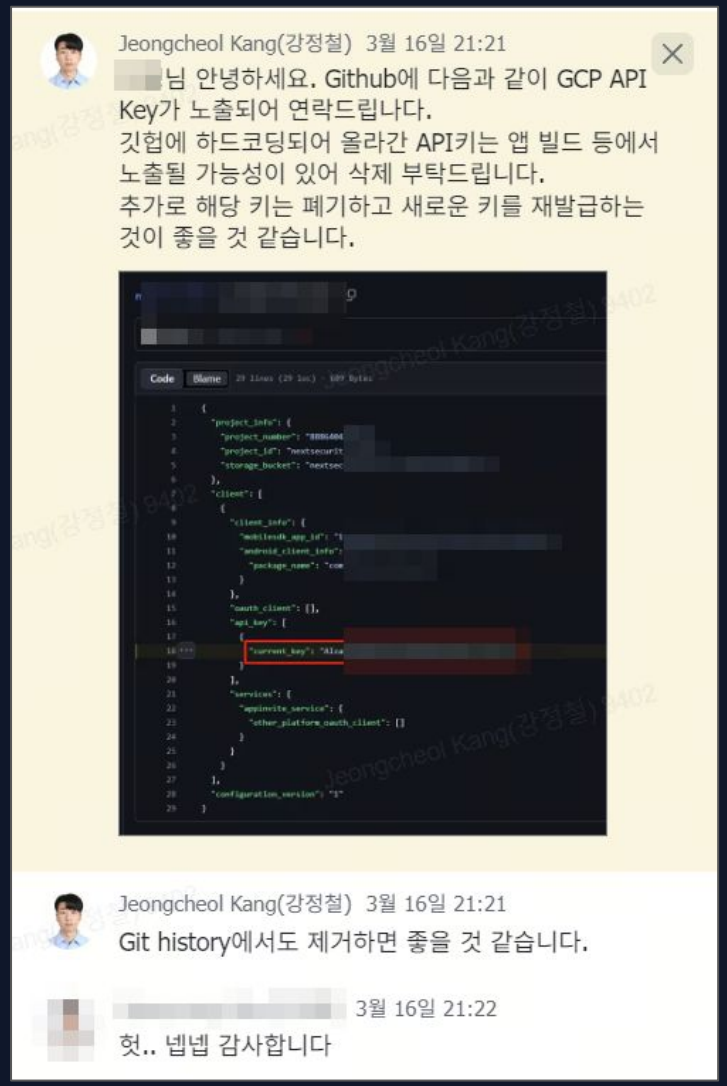
Active Secret Detected!

Cremit(Argus) 솔루션을 연동하여 실시간으로 유출된 비밀 정보를 탐지하고 Slack으로 즉시 알림을 발송합니다.

- **탐지:** GitHub PR / Push 시 실시간 스캐닝
- **알림:** 보안팀 채널에 즉각 전파
- **가시성:** 유출된 토큰 타입 및 위치 파악



FEEDBACK LOOP WITH DEVELOPERS



보안과 개발의 상생

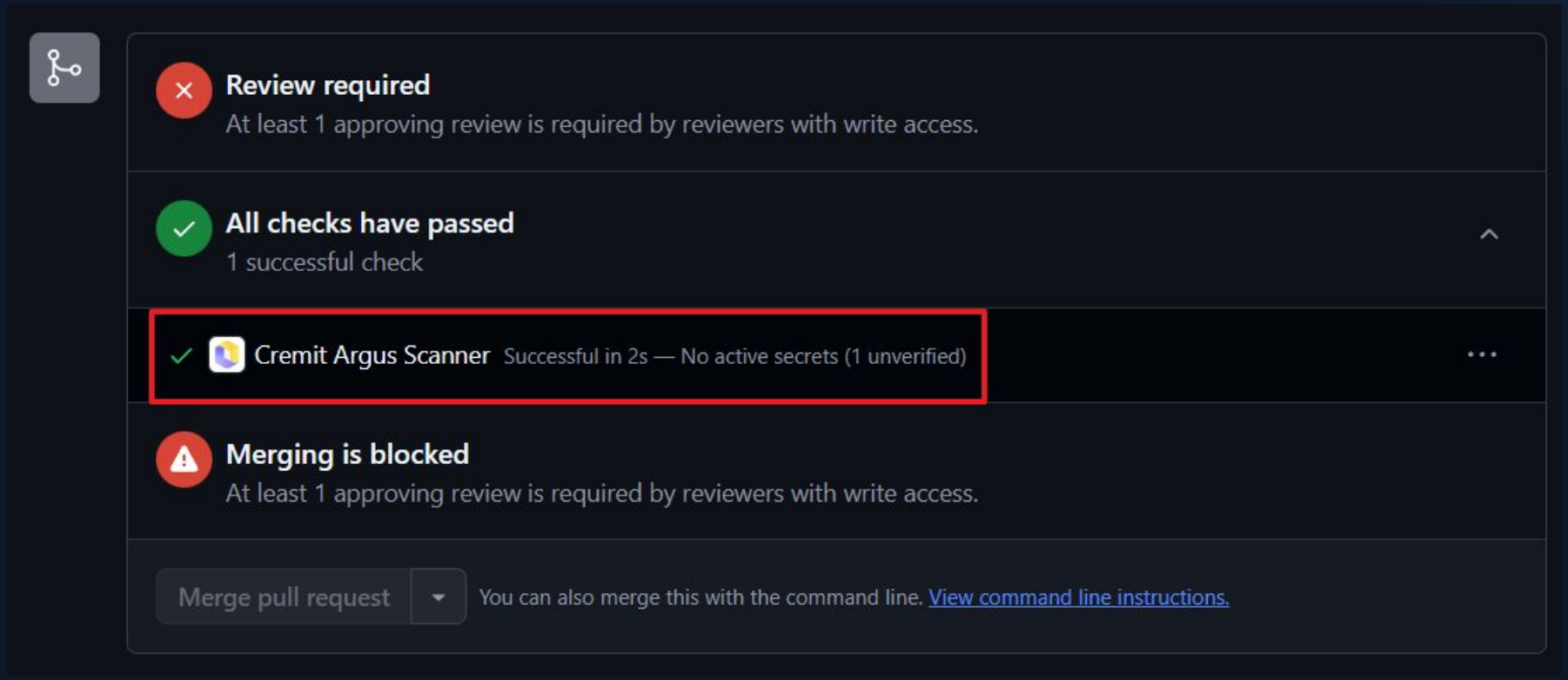
탐지만 하고 끝내는 것이 아니라, 개발자와 소통하여 **근본적인 조치**를 가이드합니다.

실제 사례: GCP API Key 노출

"안녕하세요, GCP API Key가 노출되어 연락드립니다. 해당 키는 삭제 후 재발급 부탁드리며, Git History에서도 제거해 주세요."

→ 개발자: "헉.. 넵넵 감사합니다!"

PR Scanner


Review required
At least 1 approving review is required by reviewers with write access.
All checks have passed
1 successful check
Cremit Argus Scanner Successful in 2s — No active secrets (1 unverified)
Merging is blocked
At least 1 approving review is required by reviewers with write access.
Merge pull request
You can also merge this with the command line. View command line instructions.

SAST(Static Application Security Testing)

Main Selection

Semgrep

특징: 패턴 매칭 기반 정적 분석

- 압도적으로 빠른 스캔 속도
- YAML 기반의 쉬운 커스텀 룰
- 오픈소스 커뮤니티 활성화

Sparrow

특징: 정밀 데이터 흐름 분석

- 높은 탐지 정확도 (전통 방식)
- 국내 보안 컴플라이언스 최적화
- 비교적 무겁고 느린 분석 속도

Sub Reviewer

Claude AI

특징: AI 기반 시맨틱 분석

- 코드 컨텍스트 이해도 매우 높음
- 논리적 설계 오류 탐지 가능
- 토큰 비용 및 할루시네이션 리스크

AUTOPROOF: REDUCING NOISE

90%

Alert Fatigue Reduction

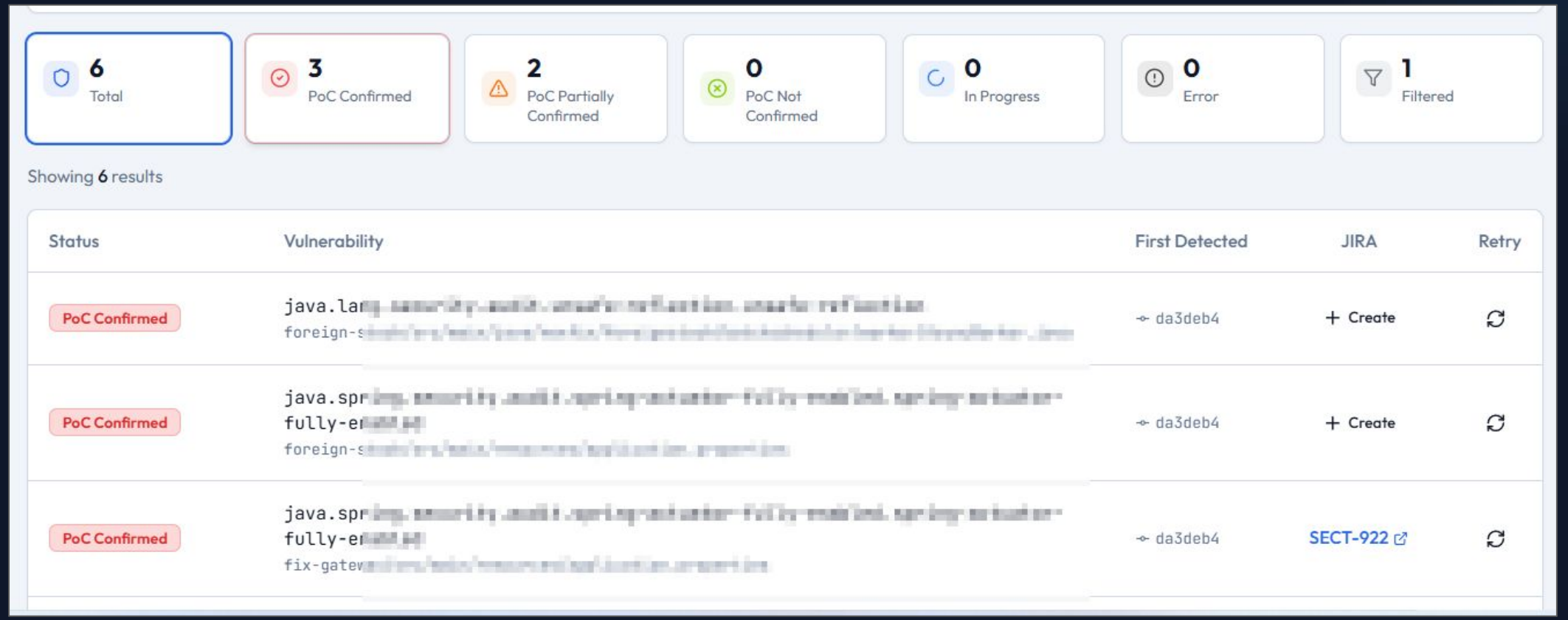
보안 솔루션의 고질적인 문제인 **오탐(False Positive)을 제거**하여 보안팀의 피로도를 획기적으로 낮춥니다.

Real Threats, Not Noise

AutoProof는 탐지된 취약점에 대해 실시간으로 **PoC(Proof of Concept)를 생성하고 검증**합니다.

- 이론적 위험을 실제 증거로 변환
- Exploit 가능성 확인 후 알림 발송
- 조치 가이드와 함께 Actionable한 결과 제공

AutoProof


6
Total
3
PoC Confirmed
2
PoC Partially Confirmed
0
PoC Not Confirmed
0
In Progress
0
Error
1
Filtered
Showing 6 results
Status
Vulnerability
First Detected
JIRA
Retry
PoC Confirmed
da3deb4
Create
PoC Confirmed
da3deb4
Create
PoC Confirmed
da3deb4
SECT-922

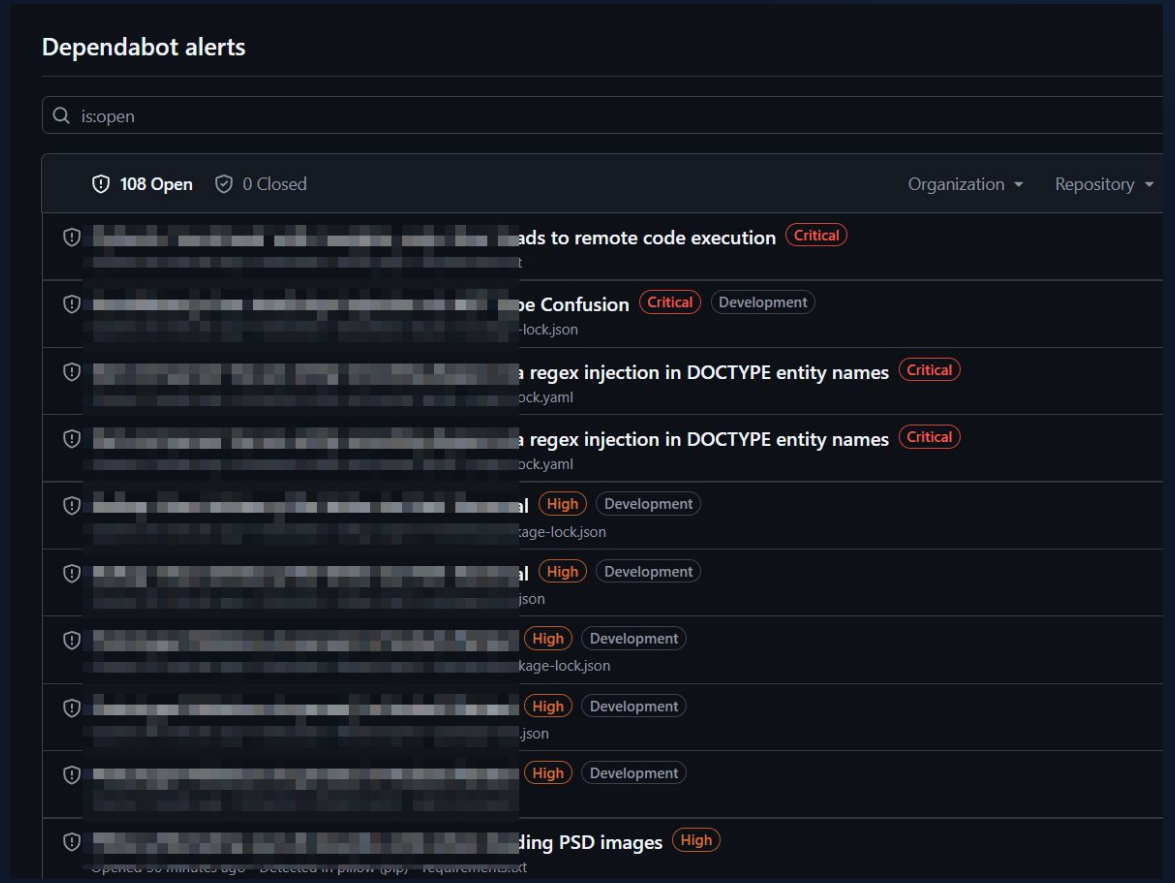
SCA(Software Composition Analysis)

Dependabot: 자동 CVE 스캔

현대 앱의 대부분은 오픈소스 라이브러리로 구성됩니다.

SCA는 우리가 직접 짜지 않은 코드의 위험을 관리합니다.

- **실시간 탐지:** CVE 취약점 DB 기반 자동 스캔
- **자동 조치:** 보안 업데이트가 포함된 PR 자동 생성
- **리스크 점수화:** 취약점 심각도별 분류



결론

실시간 코드 유출 모니터링

Github Audit log 및 접근 이력을 정밀 분석하여 코드 유출 및 비정상적인 접근 행위를 실시간으로 감시하고 즉각적인 경보 체계를 구축합니다.

CI 프로세스 보안 자동화

모든 Pull Request 및 CI 단계에 **Secret Scanning** 과정을 필수 적용하여 API 키, 패스워드 등 민감 정보가 리포지토리에 유출되는 것을 원천 차단합니다.

지속적 취약점 조치 체계

정기적인 **SAST**(정적 분석) 및 **SCA**(오픈소스 분석) 검사를 통해 코드 결함과 종속성 취약점을 상시 진단하고 체계적으로 조치합니다.

여담

데이비드 리카도 : 비교우위

	쌀	기계
갑국	5명	6명
을국	9명	8명

여담

길가는 여자에게 번호 물어보려고 하는데 이때 쓸 멘트 5가지 알려줘

1. "실례합니다. 지나가다가 너무 제 스타일이셔서요. 괜찮으시면 연락처 여쭤봐도 될까요?"
2. "죄송한데 잠깐만요. 인상이 너무 좋으셔서 그냥 지나가기 아쉬웠어요. 괜찮으시면 번호 알려주실 수 있을까요?"
3. "실례합니다. 혹시 부담 아니시면 연락처 한 번 여쭤봐도 될까요? 아니면 괜찮습니다."
4. "지나가다가 눈에 띄셔서 용기 내봤습니다. 괜찮으시면 연락처 교환 가능할까요?"
5. "갑자기 말씀드려 죄송한데, 너무 매력적이셔서요. 혹시 괜찮으시면 번호 여쭤봐도 될까요?"

여담

DevSecOps는 문화다

Q & A

linkedin.com/in/kangk7300-6a3569279/