# Zero Trust 기반 원격근무 보안강화

유용상, 장예나, 한원표

# 목차

# 1. 연구 배경

## 배경

- 코로나 팬데믹 이후 원격근무 보편화
  => **기업 네트워크 외부**에서의 접속 多
- 기존 경계기반 보안모델 (Perimeter-based Security) 의 단점
  네트워크 내부를 신뢰하는 구조, **내부자 공격**과 **우회 접근**에 취약

## 과제의 필요성

- 원격근무의 지속적 확산 >> 보안 환경의 변화 필요
- 기존 모델 한계: 물리적/인적/기술적 위협 요소 존재
- 네트워크 보안 이슈:
  - VPN 보안 취약성
  - 계정 도난 및 검증 미흡
  - 외부 공용 네트워크 환경에서의 위협

# 1. 연구 배경

## Zero Trust 기술

**“절대 신뢰하지 말고, 항상 검증하라”**

사용자, 장치, 위치, 요청 시점의 맥락 고려
**접근을 매번 검증**하고 **최소 권한만** 부여

# 1. 연구 배경

- **Zero Trust 보안 아키텍처 설계**
  - NIST SP 800-207 기반 7원칙 반영
  - 동적 정책 기반 접근 제어 및 지속적인 인증 구조 설계
- **MFA(다중 인증) 시스템 적용**
  - OTP, 생체인식, WebAuthn 등 다양한 인증 기술 비교 및 적용 실험
  - MFA 적용 전/후 보안성 및 사용자 경험 분석
- **이상 행위 탐지 시스템 구축**
  - 머신러닝 기반 로그인 패턴 분석
  - 비정상 접속 탐지 및 자동 차단 시스템 구현
- **보안 게이트웨이 도입**
  - Cloudflare Access, Gateway, Browser Isolation 활용
  - AWS Gateway 및 WAF, GuardDuty로 트래픽 제어 및 위협 탐지
- **테스트 및 최적화**
  - Zero Trust 환경과 기존 환경 보안성 비교
  - 침투 테스트 및 성능 측정 기반 보안 정책 개선

# 2. 연구 내용

## MFA 시스템

**Multi Factor Authentication, 다중 인증 시스템**
최소 **두 가지 이상의 인증**을 거친 사용자에게만 접근 허용
사용자가 로그인할 때 암호 외에 추가적인 인증 요소를 제공하도록 요구

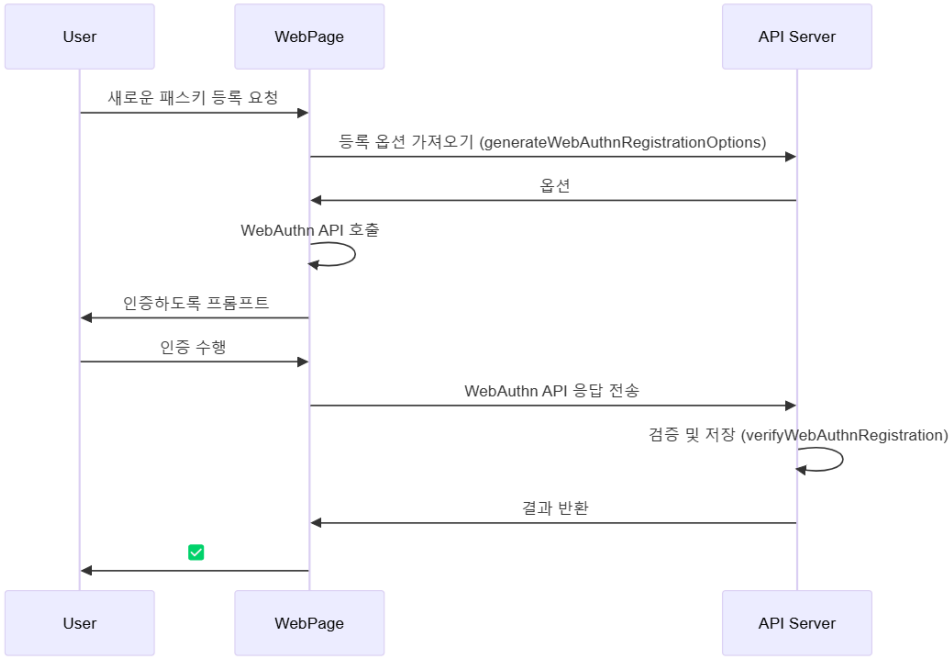
**효과**
- 비밀번호 도난 상황에 무단 접근 차단하여 보안 유지 가능
- 계정 손상 공격 차단

**진행 방식**
사용자 로그인 -> 추가 인증 요청 -> 다중 인증 확인 -> 인증 성공

# 2. 연구 내용

## 인증 요소

- **일회용 패스코드 (OTP):** 이메일, SMS로 일회용 패스코드 전달
- **푸시 알림:** 액세스 요청 확인을 요청하는 경고가 사용자의 모바일 장치로 전송
- **하드웨어 토큰:** FIDO2 키 및 사용자가 데스크탑에 연결하는 기타 물리적 장치
- **웹 인증 (WebAuthn):** 공개 키 암호화와 보안 키를 사용한 인증 환경 제공




< WebAuthn 방식 인증의 흐름

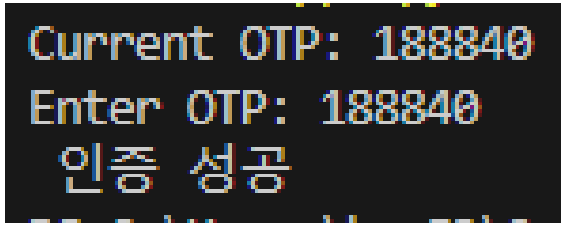
이미지 출처: "Next.js에서 WebAuthn 구현: 실습 가이드"

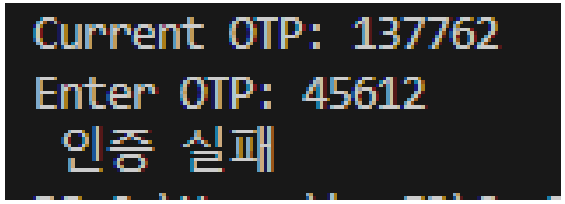
# 2. 연구 내용

## MFA:OTP적용 예시

```python
import pyotp
import time

# 사용자별로 시크릿 키 생성
secret = pyotp.random_base32()
totp = pyotp.TOTP(secret)

print("Current OTP:", totp.now())  # 사용자에게 제공할 OTP

# 사용자가 입력한 OTP 검증
user_input = input("Enter OTP: ")
if totp.verify(user_input):
    print(" 인증 성공")
else:
    print(" 인증 실패")
```

적용 예시 코드




실행 결과

# 2. 연구 내용 _ 방향성

## MFA 시스템

### 인증 요소간 비교

- 비교 기준: 보안성, 사용자 편의성, 비용, 운영 유지 보수

### 전/후 사용자 경험 분석

- 평가 항목: 로그인 속도, 로그인 실패율, 장애 및 복구 속도

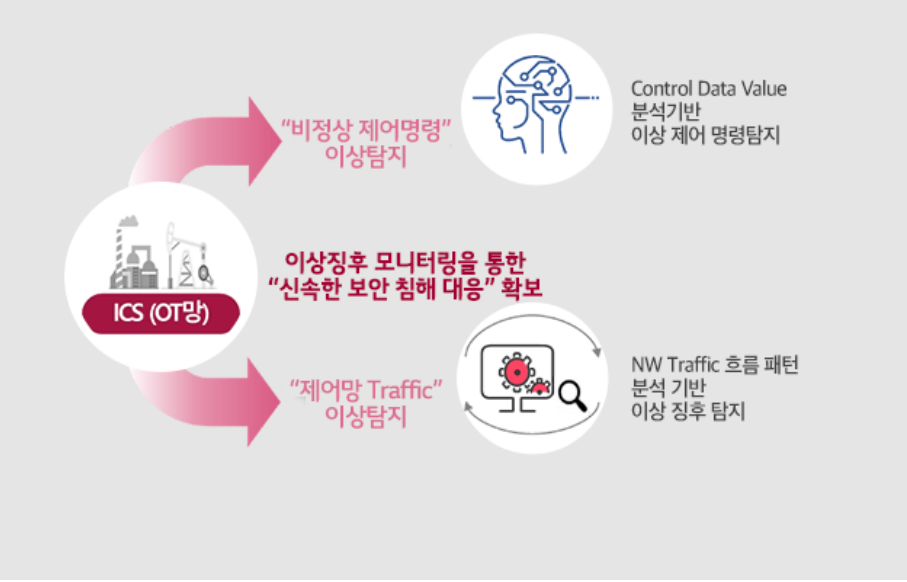
# 2. 연구 내용

## Anomaly Detection

**이상 행위 탐지 시스템**

데이터 안에서 예상하지 못한 패턴을 찾아내는 시스템

다수의 정상 데이터에서 극소수의 **비정상 데이터를 구별**하는 시스템




< NW Traffic 이상 탐지 유형 2가지

이미지 출처: "OT 보안에서 꾸준히 거론되는 '이상징후탐지솔루션' 주목!"

# 2. 연구 내용

## Anomaly Detection

### 구현 방안 방법론

- 모델 기반
  - **Isolation Forest:** Tree 구조 활용하여 데이터를 분할 및 고립시켜 이상치 구분
  - **1-class SVM:** 데이터 존재 영역을 정의, 영역 밖의 데이터들은 이상치로 간주
- 밀도, 거리 기반
  - **Gaussian Mixture Model:** 데이터가 여러 정규 분포로 구성되어 있다 가정
  - **K-최근접 이웃(kNN):** 주어진 데이터에서 특정 기준 벗어나면 이상치로 간주
  - **LOF(Local Outlier Factors):** 데이터의 밀도 또는 거리 척도로 군집을 생성하여 이상치 구분
- 재구성 기반
  - **PCA(Principal Component Analysis)**: 데이터의 주성분 추출하여 이상치 구분
  - **Auto-Encoder based Method**: 데이터를 압축/복원하여 복원 정도로 이상치 구분

# 2. 연구 내용

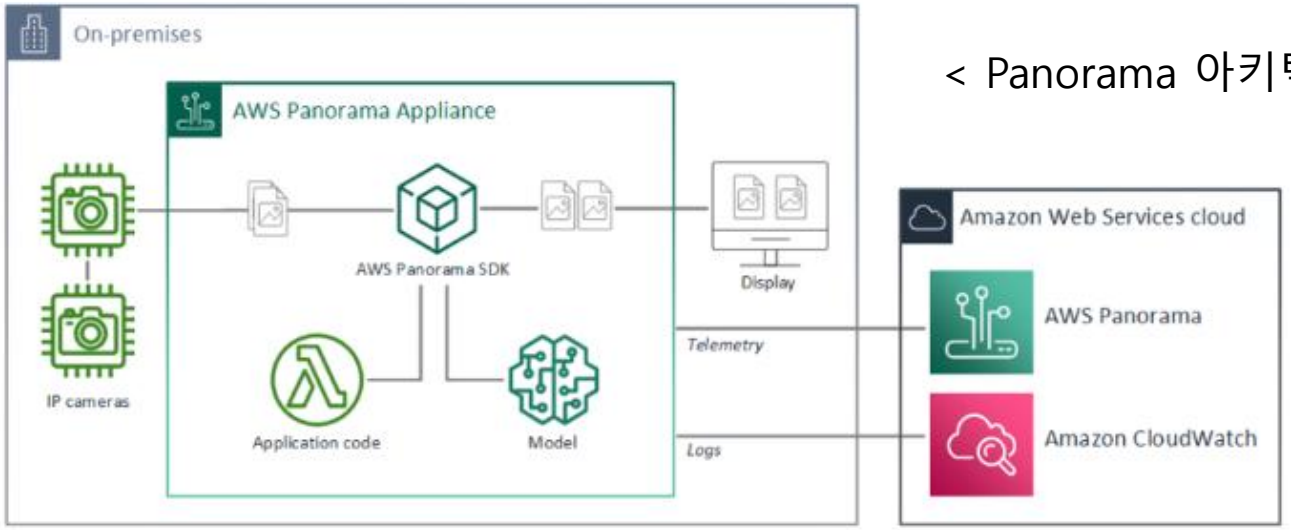
## Anomaly Detection

### 실제 적용 사례

- AWS의 이상 탐지 제품들

  AWS Panorama, Amazon DevOps, Amazon OpenSearch

  Amazon Kinesis: 데이터 수집, 탐지된 이상 현상에 점수를 첨부




< Panorama 아키텍처

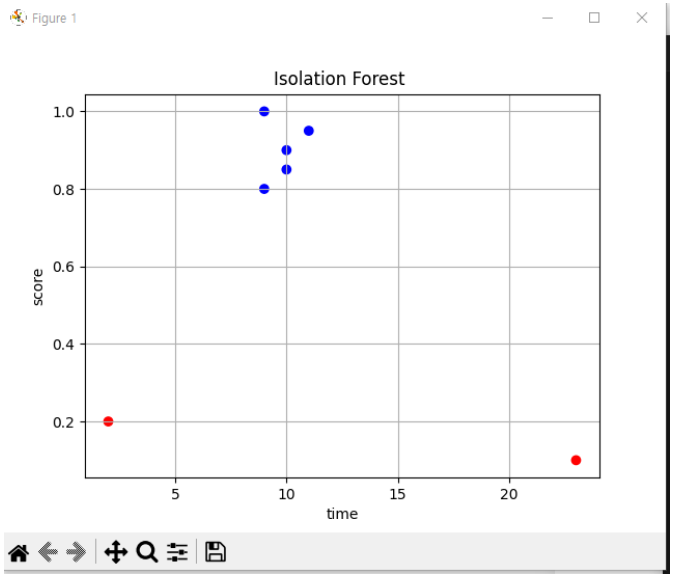
이미지 출처: "ML 이상 탐지 설명"

# 2. 연구 내용

## AI 기반 탐지 모델 구현(Isolaton Forest)

```python
from sklearn.ensemble import IsolationForest
import numpy as np
import matplotlib.pyplot as plt

# 로그인 시도 데이터 (예: [접속 시간대, 위치 점수])
X = np.array([
    [9, 0.8], [10, 0.9], [9, 1.0], [11, 0.95], [10, 0.85],  # 정상
    [23, 0.1], [2, 0.2]  # 이상값 (심야 비정상 접속)
])

# Isolation Forest 모델 구성
model = IsolationForest(contamination=0.2, random_state=42)
model.fit(X)

# 예측 수행: 1은 정상, -1은 이상값
preds = model.predict(X)
print("예측 결과:", preds)

# 이상 탐지 시각화
colors = ['red' if p == -1 else 'blue' for p in preds]
plt.scatter(X[:, 0], X[:, 1], c=colors)
plt.xlabel('time')
plt.ylabel('score')
plt.title('Isolation Forest ')
plt.grid(True)
plt.show()
```

적용 예시 코드



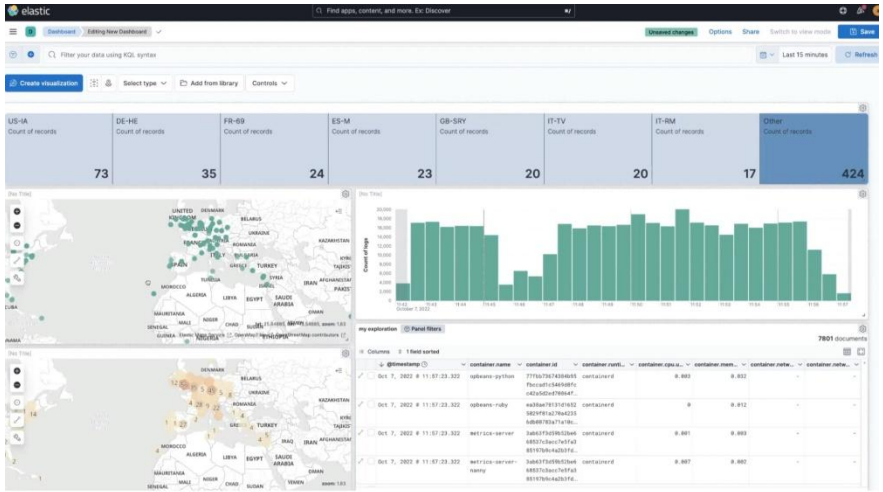

```
예측 결과: [ 1  1  1  1  1 -1 -1]
```

실행 결과

# 3. 성능 비교

- **Before & After 비교 그래프**
  - 피싱/계정 탈취/이상 로그인 이벤트의 유의미한 감소율 확인
  - 로그 분석을 통해 데이터 수집, AI 기반 탐지 모델 성능 분석
  - 보안 탐지 시간 단축을 그래프화


**Grafana**
**- 실시간 모니터링 시각화**

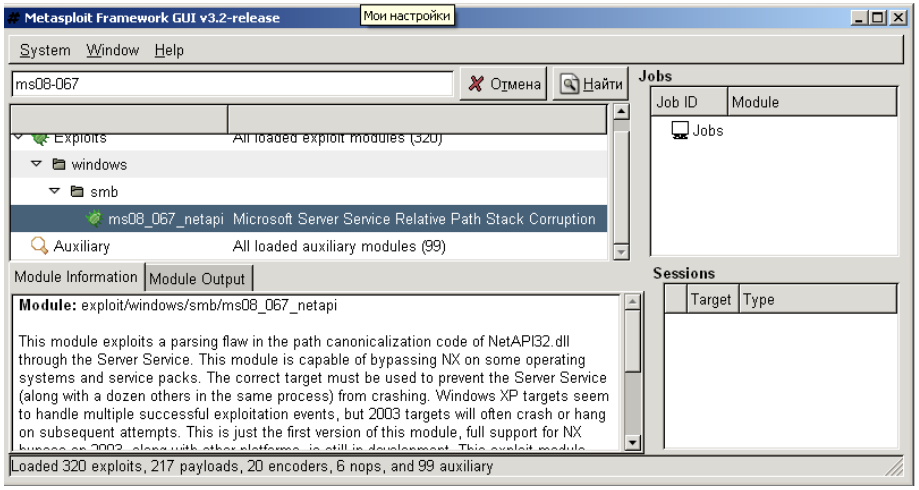


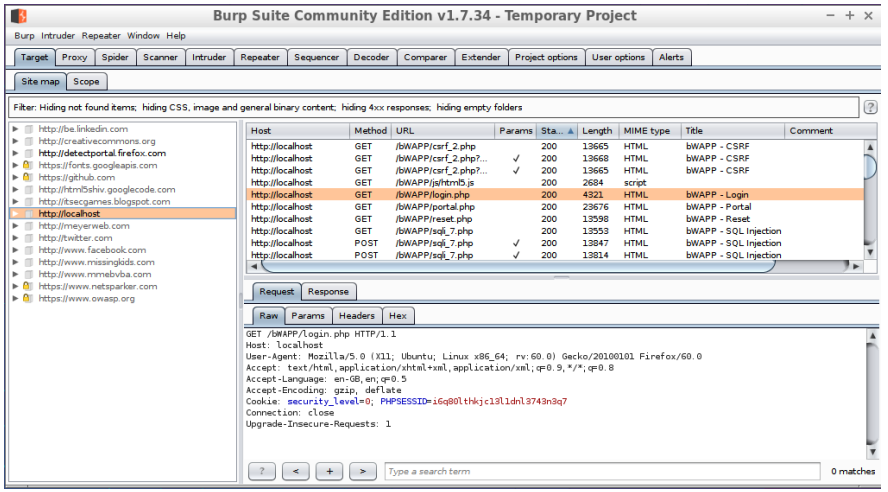

**Kibana**
**- 네트워크 보안 로그 분석**

# 3. 성능 비교

- **침투 테스트 시뮬레이션 (VPN vs Zero Trust)**
  - 주요 위협 가능성 있는 공격을 실제로 차단 가능한지 확인
  - 시나리오: 피싱, 랜섬웨어, SQL Injection, 권한 우회 공격

**Metaploit Framework**
**- 다양한 공격 기법 시뮬레이션**




**Burp Suite**
**- 웹 기반 공격 분석**

# 4. 일정 계획

**1) Zero trust 모델 연구**

- NIST SP 800-207기반 이케텍처 정리
- 기존 VPN모델과의 보안 구조 차이 분석

**2) MFA 적용 및 테스트**

- OTP, 생체인식, WebAuthn 적용 코드 작성 및 데모 영상 제작
- 인증 적용 후 데모 영상 제작 및 UX, 보안성 비교

**3) 이상 탐지 시스템 개발 및 테스트**

- 비정상 로그인 패턴 정의
- AI기반 탐지 모델 구현 및 테스트

**4) 보안 성능 비교 그래프 산출**

- Zero Trust 도입 전/후 보안 지표 시각화
- 계정탈취 방지율, 탐지속도, 차단율 등 정량 평가 설계

# 질의응답