주식 커뮤니티 서비스

만찐두빵

유지훈 김범준 김효진 이호연 조용승

INDEX

프로젝트 개요

- 기획 의도

주식 시장이 주요한 투자 수단으로 각광받으면서 국내 투자자들은 다양한 커뮤니티를 활용해 정보를 공유하고 토론을 하고 있다.
그러나 많은 종목 토론방에서는 욕설, 비방, 리딩방 홍보 등 부정적인 요소가 눈에 띄기 때문에 이러한 부정적인 요소를 배제하여 건전한 토론을 지향하는 게시판을 만들고자 한다.

- 기대효과

- 전문성 강화: 건전한 토론을 중심으로 투자와 주식에 대한 전문적인 정보 교류를 하여 이용자들은 신뢰성 있는 정보를 얻을 수 있게 된다.
- 커뮤니티 강화: 부정적인 환경이 배제된 게시판은 건전한 커뮤니티를 형성한다.

프로젝트 개요

- 프로젝트 구현 기능사항

1. 공지사항 게시판 : 운영자가 게시판 이용자에게 공지를 전달할 수 있는 페이지
2. 종목 토론 게시판 : 이용자들이 주식에 대해 자유롭게 토론할 수 있는 페이지
3. 주가 확인 페이지 : 주가를 실시간으로 확인할 수 있는 페이지
4. 로그인 / 회원가입 페이지
   - 이용자들은 개인 계정을 생성하고 로그인하여 서비스를 이용 가능
   - ID 및 비밀번호 분실 시에 찾을 수 있는 기능 포함
   - 본인 확인을 위한 이메일 인증 기능 포함
5. 관리자 관리 페이지
   - 운영자는 불건전 사용자에 대한 제재 가능
   - 관리자는 자신의 신원을 확인하기 위한 2차 이메일 인증 기능을 사용하여 안전성을 확보

프로젝트 개요

- 프로젝트 개발 언어

WEB APPLICATION

SERVER

INFRA

프로젝트 팀원 소개

팀 구성원	역할
유지훈 (팀장)	프론트 엔드, 기능 파이썬 함수 개발, 문서작업, 개발 관리, 발표
김범준	프론트 엔드, 문서작업, 기능 구현
김효진	웹 디자인, 프론트 엔드, 문서작업, PPT
이호연	백 엔드, 데이터베이스, 인증 서비스 구현, 취약점 분석, 그 외 다수
조용승	AWS인프라 구성, WAF 기능 구현, 서비스 기능 QA 및 검증

프로젝트 수행 일정

업무 \ 날짜	28	27	28	29	30	31	1
서비스 기획 및 계획서 작성							
프론트 엔드 개발							
백엔드 개발							
DB							
이메일 인증							
AWS 구성							
AWS WAF 설정							
QA							
결과 보고서 작성							

프로젝트 결과 -일반사용자 버전

Main 페이지

주식서비스에 오신걸 환영합니다.

주가확인

시간별 주식 시세 정보
주가를 확인해요!

공지사항

공지사항입니다!

종목토론

건전한 토론문화를 정착해 갑시다!

로그인 회원가입
ID찾기 PW찾기

기타 문의사항은 운영팀에게 연락바랍니다.

프로젝트 결과 -일반사용자 버전

회원가입 페이지

주식서비스에 오신걸 환영합니다.

주가확인

시간별 주식 시세 정보
주가를 확인해요!

공지사항

공지사항입니다!

종목토론

건전한 토론문화를 정착해 갑시다!

로그인 회원가입
ID찾기 PW찾기

기타 문의사항은 운영팀에게 연락바랍니다.

프로젝트 결과 -일반사용자 버전

회원가입 페이지

개인정보 수집 및 이용 동의

가. 개인정보의 수집·이용에 관한 사항

개인정보의 수집·이용 목적

귀하의 개인정보는 주식종목토론방 가입을 위한 목적으로 수집·이용됩니다.

수집·이용할 개인정보의 항목

수집·이용되는 귀하의 개인정보는 다음과 같습니다.

☐ 동의함

나. 제공에 관한 사항

❑ 제공받는 자

❍ 종목토론게시판

❑ 제공받는 자의 목적

❍ 회원가입

☐ 동의함

다음 화면으로 넘어가기

아이디

userID 아이디 중복검사

비밀번호

password 특수문자, 숫자를 포함하여 최소 8자리 이상입니다.

특수문자는 ~!@#$%^&()? 만 가능합니다.

비밀번호 확인

checkPassword

이름

name 이름은 4자 이하입니다.

닉네임

nickName 닉네임 중복검사

닉네임에 욕설, 혐오 표현 포함은 제재 대상입니다.

이메일 주소

emailAddr 인증번호 발송

이메일 인증

input OTP 인증번호 확인

회원가입 완료하기

프로젝트 결과 -일반사용자 버전

회원가입시 이메일 인증 코드 - OTP 보내기

```python
@app.route('/send_otp', methods=['POST'])
def send_otp():
    email = request.form['email']  # 이메일 주소를 입력 폼에서 가져옵니다.
    otp = generate_otp()
    session['otp'] = otp  # 세션에 OTP 저장
    print(session)

    msg = Message('인증번호', sender=app.config['MAIL_USERNAME'], recipients=[email])
    msg.body = f'귀하의 인증번호는 {otp}입니다.'
    mail.send(msg)

    return jsonify({'message': '인증번호가 발송되었습니다.'})
```

사용자가 전송한 이메일 주소를
폼에서 가져옴

OTP를 생성하는 함수를 호출하여
반환된 OTP 값을 변수에 저장
그렇게 생성된 OTP를 세션에 저장

이메일 메시지를 생성하고,
메시지의 본문에 OTP를 포함시켜
메일을 발송

프로젝트 결과 -일반사용자 버전

회원가입시 이메일 인증 코드 - OTP 인증하기

```python
@app.route('/verify_otp', methods=['POST'])
def verify_otp():

    user_otp = request.form['InputOtp'].strip()

    if 'otp' in session:
        session_otp = str(session['otp'])  # 세션의 OTP 값을 문자열로 변환

        if session_otp == user_otp:

            session.pop('otp', None)
            session['otp_verified'] = True

            return jsonify({'message': '인증번호가 확인되었습니다.'})
        else:

            return jsonify({'message': '인증번호가 일치하지 않습니다.'}), 400
    else:
        return jsonify({'message': '세션에 인증번호가 없습니다. 다시 시도해주세요.'}), 400

```

클라이언트가 제출한 폼에서 사용자가 입력한 OTP 값을 가져온다

세션에 저장된 OTP와 사용자가 입력한 OTP를 비교하여 검증 한다.
일치하면 세션에서 OTP를 삭제하고, 'otp_verified'라는 새로운 세션 변수를 True로 설정한다.
일치하지 않거나 세션에 OTP가 없는 경우에는 에러 메시지를 반환한다

프로젝트 결과 -일반사용자 버전

로그인 페이지

주식서비스에 오신걸 환영합니다.

주가확인

시간별 주식 시세 정보
주가를 확인해요!

공지사항

공지사항입니다!

종목토론

건전한 토론문화를 정착해 갑시다!

로그인 회원가입
ID찾기 PW찾기

기타 문의사항은 운영팀에게 연락바랍니다.

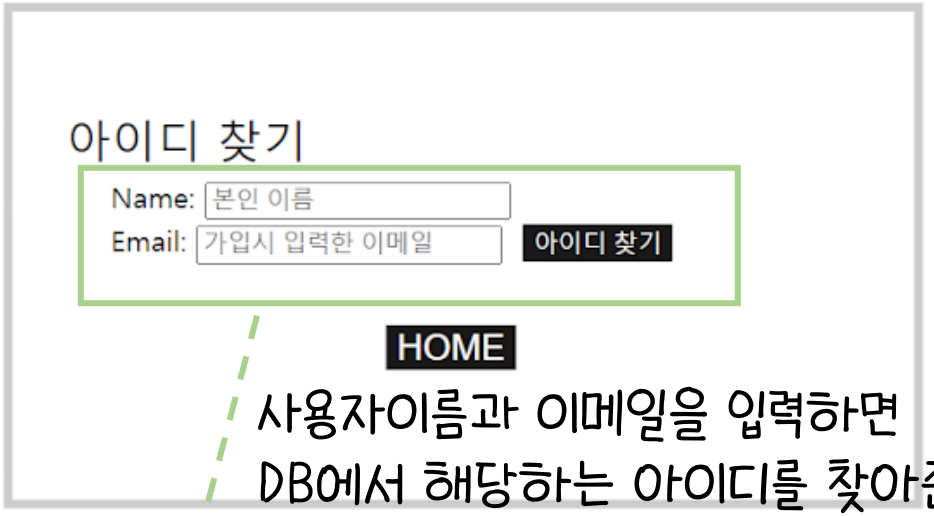
프로젝트 결과 -일반사용자 버전

로그인 페이지

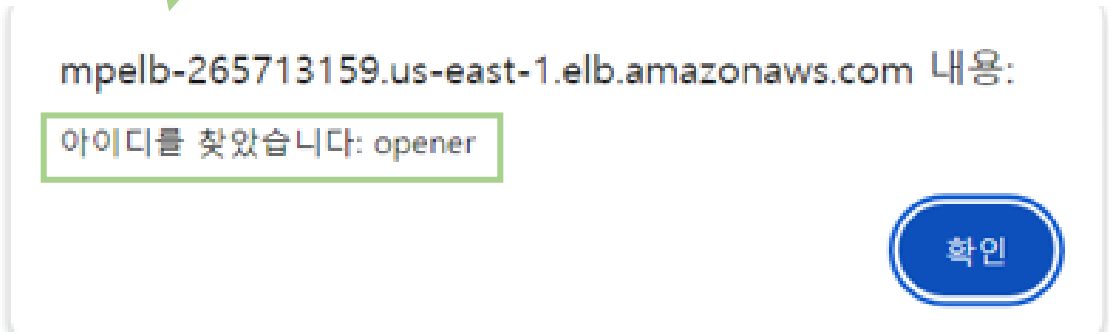
자신의 ID와 PW를 입력하고 로그인 버튼을 누르면 아래와 같이 user의 정보를 출력함

프로젝트 결과 -일반사용자 버전

ID 찾기 페이지



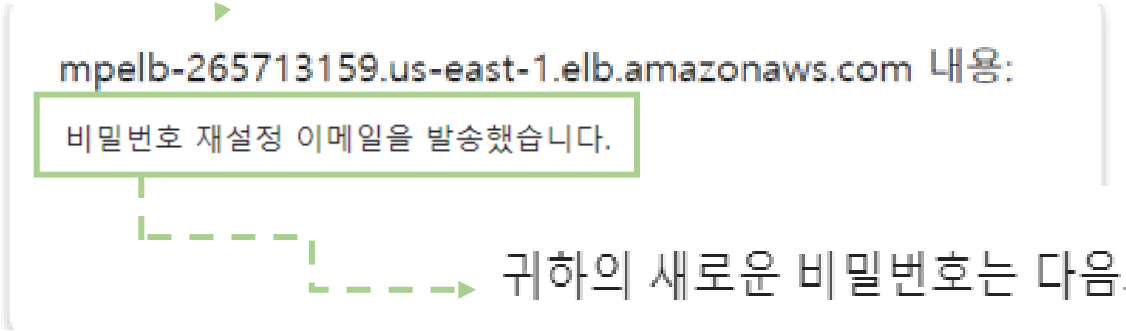
사용자이름과 이메일을 입력하면
DB에서 해당하는 아이디를 찾아준다



프로젝트 결과 -일반사용자 버전

PW 찾기 페이지

이메일을 입력하고 재설정 버튼 누르면
PW를 재설정하여 해당 메일로 전송해준다.



프로젝트 결과 -일반사용자 버전

PW 재설정 시 이메일 발송 코드

```python
def find_password():

    id_pw = request.form['id_pw']
    email_pw = request.form['email_pw']
    connection = pymysql.connect(**DATABASE_CONFIG)
    try:
        with connection.cursor() as cursor:

            sql = "SELECT email FROM users WHERE id = %s"
            cursor.execute(sql, (id_pw,))
            result = cursor.fetchone()

            if result and result['email'] == email_pw:
                # 새로운 비밀번호 생성
                new_password = generate_new_password()
                #해시화 시켜야함
                hashedPassword = hashlib.sha256(new_password.encode()).hexdigest()

                # 사용자의 비밀번호를 새로운 비밀번호로 업데이트
                sql = "UPDATE users SET passwd = %s WHERE id = %s"
                cursor.execute(sql, (hashedPassword, id_pw))
                connection.commit()
```

사용자가 제출한 폼에서 사용자 ID와
이메일 주소를 가져온다.

DB 연결하여 사용자 ID와 이메일 주소가 일치하는
사용자의 이메일 주소를 조회한다.

조회된 이메일 주소가 입력한 이메일 주소가 일치하는 경우
새로운 비밀번호를 생성하고 이를 해시화한 후에
사용자의 비밀번호를 새로운 비밀번호로 업데이트한다.

프로젝트 결과 -일반사용자 버전

PW 재설정 시 이메일 발송 코드

```
            # 이메일 발송 로직
            msg = Message('비밀번호 재설정 안내', sender=app.config['MAIL_USERNAME'], recipients=[email_pw])
            msg.body = f'귀하의 새로운 비밀번호는 다음과 같습니다: {new_password}\n'
            mail.send(msg)

            flash('비밀번호 재설정 이메일을 발송했습니다.')
            return redirect(url_for('findPW'))
        else:
            flash('입력하신 정보와 일치하는 계정이 없습니다.')
            return redirect(url_for('findPW'))
except Exception as e:
    flash('데이터베이스 오류가 발생했습니다.')
    app.logger.error(f"Database error: {e}")
    connection.rollback()

    return redirect(url_for('findPW'))
finally:
    connection.close()
```

사용자에게 새로운 비밀번호를
이메일로 발송한다.

쿼리 실행 중 오류가 발생하면
오류 메시지를 로그에 남기고,
클라이언트에게도 오류 메시지를 반환한다.

프로젝트 결과 -일반사용자 버전

주가확인 페이지

주식서비스에 오신걸 환영합니다.

주가확인

시간별 주식 시세 정보
주가를 확인해요!

공지사항

공지사항입니다!

종목토론

건전한 토론문화를 정착해 갑시다!

로그인 회원가입
ID찾기 PW찾기

기타 문의사항은 운영팀에게 연락바랍니다.

프로젝트 결과 -일반사용자 버전

주가 확인 페이지

삼성전자 (005930)

삼성전자 주가입니다.

날짜: 2023.12.28

가격: 78,500

SK하이닉스 (000660)

SK하이닉스 주가입니다.

날짜: 2023.12.28

가격: 141,500

LG에너지솔루션 (373220)

LG에너지솔루션 주가입니다.

날짜: 2023.12.28

가격: 427,500

프로젝트 결과 -일반사용자 버전

공지사항 페이지

주식서비스에 오신걸 환영합니다.

주가확인

시간별 주식 시세 정보
주가를 확인해요!

공지사항

공지사항입니다!

종목토론

건전한 토론문화를 정착해 갑시다!

로그인 회원가입
ID찾기 PW찾기

기타 문의사항은 운영팀에게 연락바랍니다.

프로젝트 결과 -일반사용자 버전

공지사항 페이지

프로젝트 결과 -일반사용자 버전

종목 토론 페이지

주식서비스에 오신걸 환영합니다.

주가확인

시간별 주식 시세 정보
주가를 확인해요!

공지사항

공지사항입니다!

종목토론

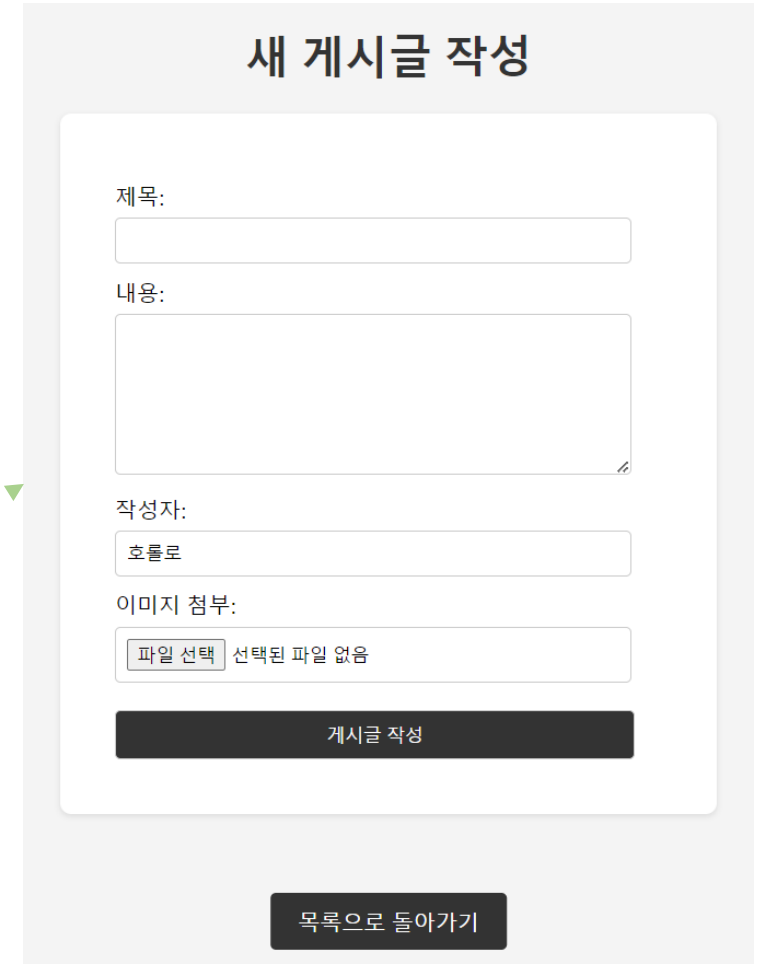
건전한 토론문화를 정착해 갑시다!

로그인 회원가입
ID찾기 PW찾기

기타 문의사항은 운영팀에게 연락바랍니다.

프로젝트 결과 -일반사용자 버전

종목 토론 페이지



프로젝트 결과 -일반사용자 버전

종목 토론 페이지

새 게시글 작성

제목:

내용:

작성자:

호롤로

이미지 첨부:

파일 선택 선택된 파일 없음

게시글 작성

목록으로 돌아가기

커뮤니티 게시판

귀여운 고양이 사진 보시고 심신의 안정을 찾으세요!

작성자: 호롤로

내용 : 고양이는 귀엽습니다!

작성일: 2023-12-31 08:30:41

hello

작성자: 1234

내용 : wow!! 잘 되는거 같은데요?!?!?!?

작성일: 2023-12-31 08:29:51

프로젝트 결과 -일반사용자 버전

게시글 & 이미지 업로드 코드

사용자가 이미지를 업로드했는지 확인

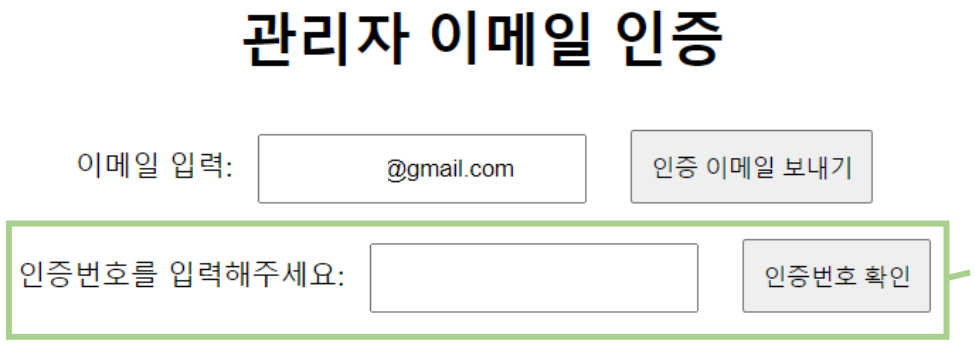
업로드한 이미지의 크기와 파일 형식을 검사한다.
이미지가 유효하다면, 이미지를 서버에 저장하고,
그 경로를 relative_image_path에 저장한다.

데이터베이스에 새 게시글을 추가하는 쿼리를 실행한다.
쿼리가 성공하면 사용자에게 성공 메시지를 표시,
실패하면 오류 메시지를 표시하고
데이터베이스의 변경 사항을 롤백한다.

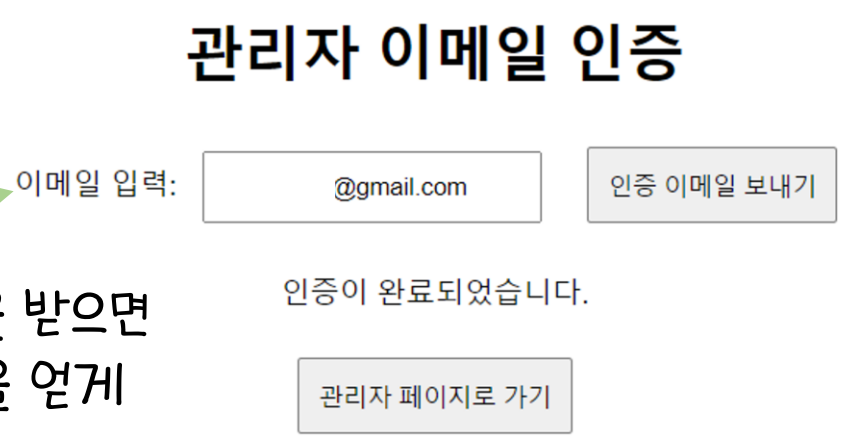
```
    relative_image_path = None

    if image and allowed_file(image.filename):
        filename = secure_filename(image.filename)
        image_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)

        img = Image.open(image.stream)  # 이미지를 열 때 image.stream을 사용
        img.thumbnail((800, 800))

        if not os.path.exists(os.path.dirname(image_path)):
            os.makedirs(os.path.dirname(image_path))

        img.save(image_path)
        relative_image_path = os.path.join('image', filename).replace("\\", "/")

    connection = pymysql.connect(**DATABASE_CONFIG)
    try:
        with connection.cursor() as cur:
            sql = "INSERT INTO community (title, content, author, image_path) VALUES (%s, %s, %s, %s)"
            cur.execute(sql, (title, content, author, relative_image_path))

        connection.commit()
        flash('게시글이 성공적으로 작성되었습니다.')
    except Exception as e:
        flash(f'게시글 작성 중 오류가 발생했습니다: {e}')
        connection.rollback()

    return redirect(url_for('community'))

return render_template('new_post.html')
```

프로젝트 결과 -관리자 버전

관리자 인증 페이지



이메일 인증을 받으면 관리자 권한을 얻게 된다.



프로젝트 결과 -관리자 버전

관리자 계정 이메일 인증 코드

```python
@app.route('/Adminauth', methods=['GET', 'POST'])
def Adminauth():
    if 'user_id' in session and session['user_id'] == 'admin':
        if request.method == 'GET':
            connection = pymysql.connect(**DATABASE_CONFIG)
            try:
                with connection.cursor() as cur:
                    cur.execute("SELECT email FROM users WHERE id = 'admin'")
                    admin_data = cur.fetchone()

                    if admin_data:
                        session['admin_email'] = admin_data['email']
                    else:
                        flash('관리자 정보를 찾을 수 없습니다.')
                        return redirect(url_for('home'))

            except Exception as e:
                flash('데이터베이스 오류가 발생했습니다.')
                app.logger.error(f"Database error: {e}")
                connection.rollback()
                return redirect(url_for('home'))
            finally:
                connection.close()
```

로그인한 사용자가 관리자인지 확인하여
관리자가 아니라면 홈 페이지로 리다이렉트한다.

요청이 GET인 경우,
DB에서 관리자의 이메일 정보를 가져온다.
이메일 정보가 없다면 홈 페이지로 리다이렉트한다.

프로젝트 결과 -관라자 버전

관리자 계정 이메일 인증 코드

```python
    # 세션에서 OTP 인증 여부 확인
    if 'otp_verified' in session and session['otp_verified']:
        # OTP 인증이 완료되었으면 관리자 페이지로 리다이렉트
        session.pop('otp_verified', None)
        return redirect(url_for('admin'))
    else:
        admin_email = session.get('admin_email', '')
        return render_template('Adminauth.html', admin_email=admin_email)
else:
    # user_id가 admin이 아니면 홈으로 리다이렉트
    flash('관리자만 접근 가능합니다.')
    return redirect(url_for('home'))
```

OTP 인증이 완료되었는지 확인한다.
인증이 완료되었으면 관리자 페이지로
리다이렉트한다.

OTP 인증이 완료되지 않았다면,
관리자 인증 페이지를 렌더링한다.
동시에 관리자의 이메일 정보를 템플릿에
전달하여 이메일을 통한 OTP 인증을 준비한다.

프로젝트 결과 -관리자 버전

관리지의 Main 페이지

주식서비스에 오신걸 환영합니다.

주가확인

시간별 주식 시세 정보
주가를 확인해요!

공지사항

공지사항입니다!

종목토론

건전한 토론문화를 정착해 갑시다!

- User : admin

logout

AdminPage

기타 문의사항은 운영팀에게 연락바랍니다.

프로젝트 결과 -관리자 버전

관리자 페이지

주식서비스에 오신걸 환영합니다.

주가확인

시간별 주식 시세 정보
주가를 확인해요!

공지사항

공지사항입니다!

종목토론

건전한 토론문화를 정착해 갑시다!

- User : admin

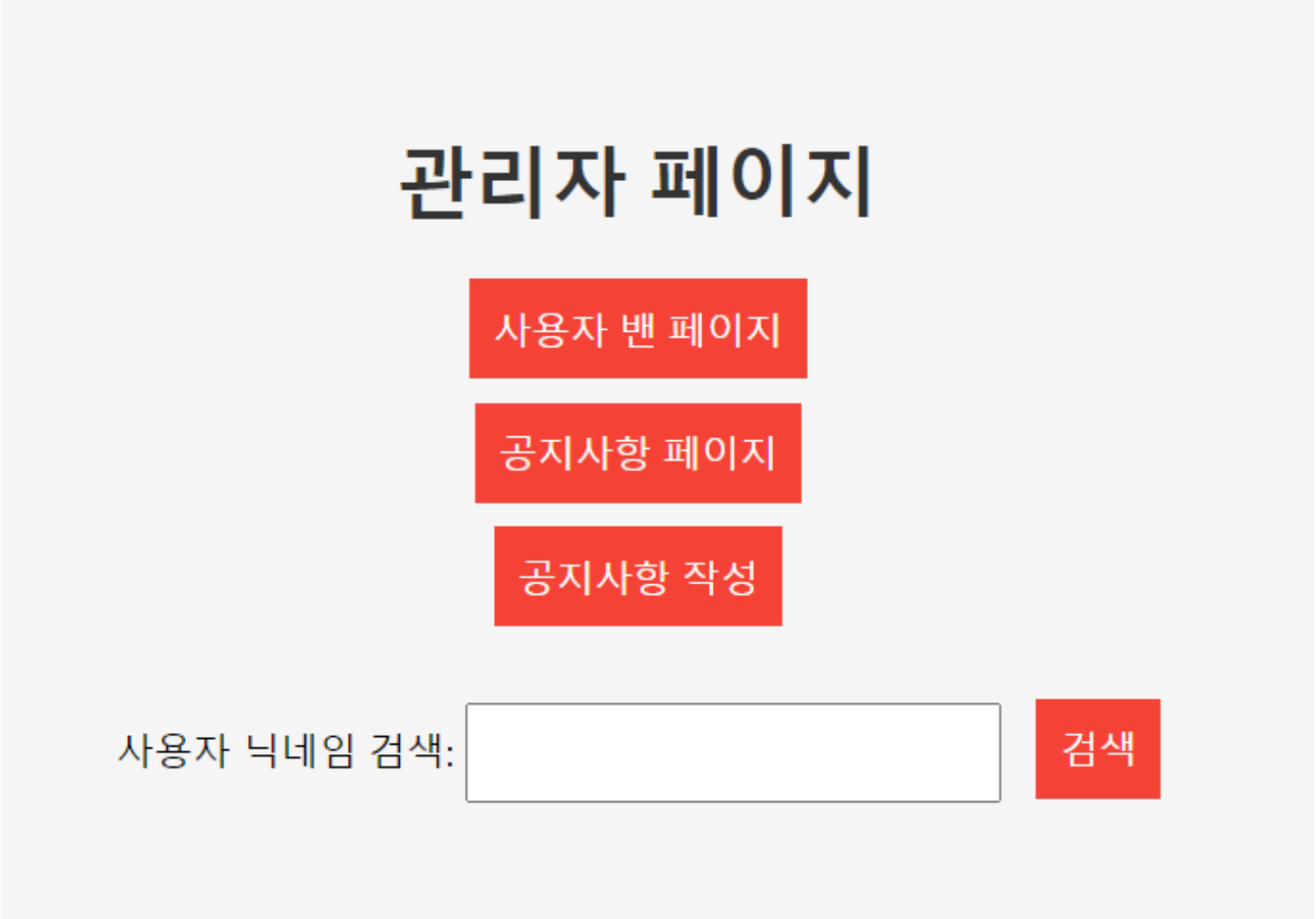
logout

AdminPage

기타 문의사항은 운영팀에게 연락바랍니다.

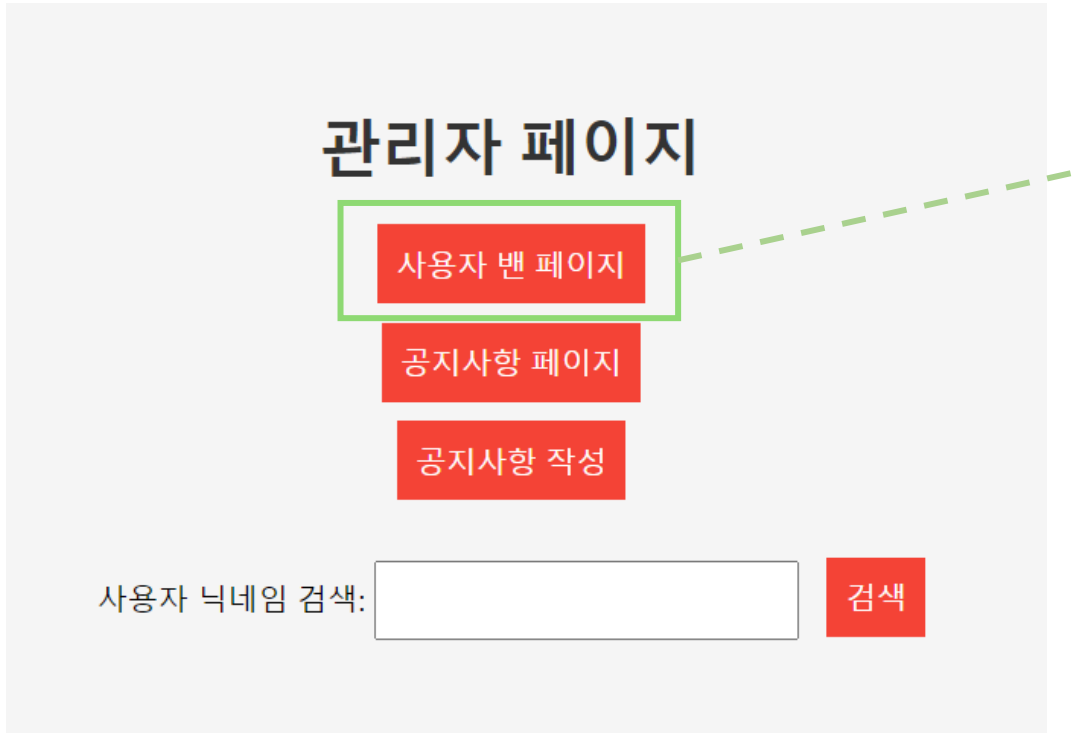
프로젝트 결과 -관리자 버전

관리자 페이지



프로젝트 결과 -관리자 버전

사용자 밴 페이지



프로젝트 결과 -관리자 버전

사용자 밴 페이지

프로젝트 결과 -관리자 버전

관리자 게시글 삭제

관리자만 특정 사용자의
게시글을 삭제 할 수 있음

프로젝트 결과 -관리자 버전

관리자 계정 사용자 밴

```python
@app.route('/ban_user', methods=['POST'])
def ban_user():
    nickname = request.form.get('nickname')
    connection = pymysql.connect(**DATABASE_CONFIG)
    try:
        with connection.cursor() as cursor:
            sql = "UPDATE users SET account_status = '1' WHERE nickname = %s"
            cursor.execute(sql, (nickname,))
            connection.commit()
        return jsonify({'banned': True})
    except Exception as e:
        app.logger.error(f"Error banning user: {e}")
        connection.rollback()
        return jsonify({'message': '사용자 밴 처리 중 오류가 발생했습니다.'}), 500
    finally:
        connection.close()
```

ajax로 넘겨 받은 nickname값을 받는다.

account_status = 1 로 계정 상태를 삭제 플래그로 수정한다.

ajax응답으로 banned를 True로 설정해서 전달하면 Ban 상태가 된다.

프로젝트 결과 -관리자 버전

관리자가 특정 사용자 게시글 삭제하는 코드

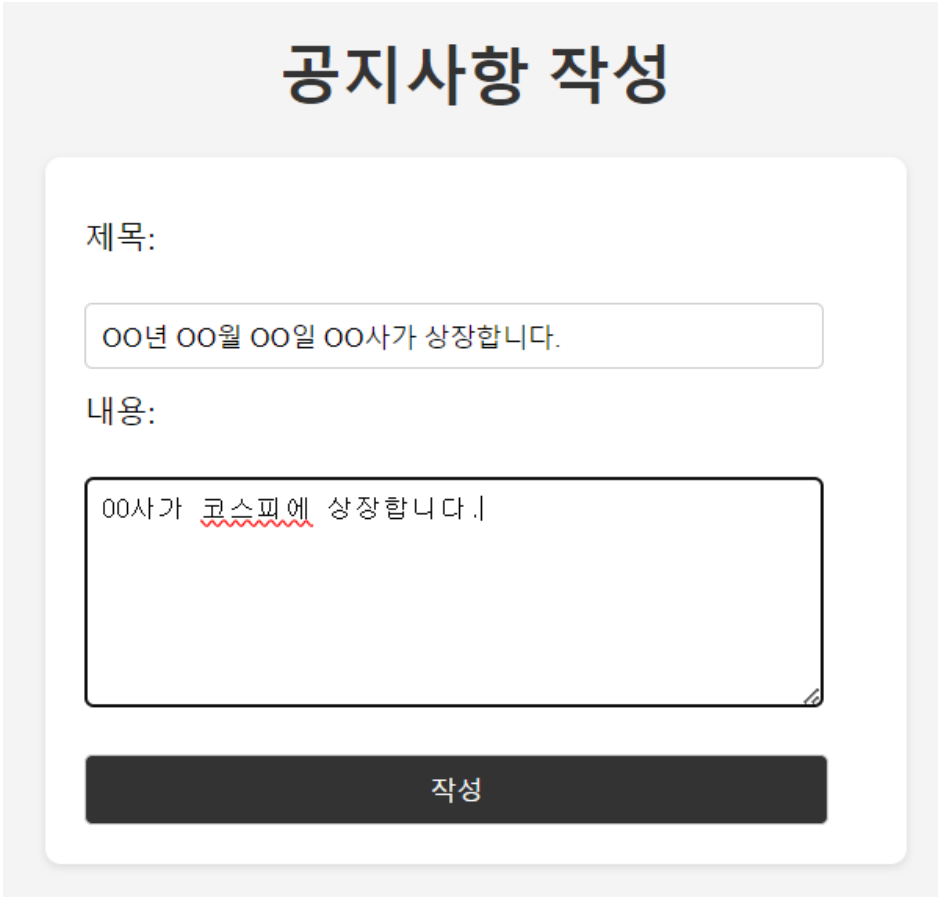
```python
@app.route('/delete_post/<int:post_id>', methods=['POST'])
def delete_post(post_id):
    connection = pymysql.connect(**DATABASE_CONFIG)
    try:
        with connection.cursor() as cursor:
            # 게시글 삭제 플래그 쿼리
            result = cursor.execute("UPDATE community SET is_deleted = 1 WHERE id = %s", (post_id,))
            connection.commit()
            if result > 0:
                return jsonify({'success': True, 'message': '게시물이 삭제 플래그 되었습니다.'})
            else:
                return jsonify({'success': False, 'message': '삭제할 게시물을 찾을 수 없습니다.'})

    except Exception as e:
        app.logger.error(f'Delete Error: {e}')
        connection.rollback()
        return jsonify({'error': '게시물 삭제 중 오류가 발생했습니다.'}), 500
```

id가 post_id인 경우,
해당 레코드의 삭제 상태를 나타내는
is_deleted를 1로 설정한다.

SQL 쿼리 실행 결과로 반환된 행의
수가 0보다 큰지 확인한다.
만약 0보다 크다면 쿼리는 성공적으로
실행되었다고 판단한다.

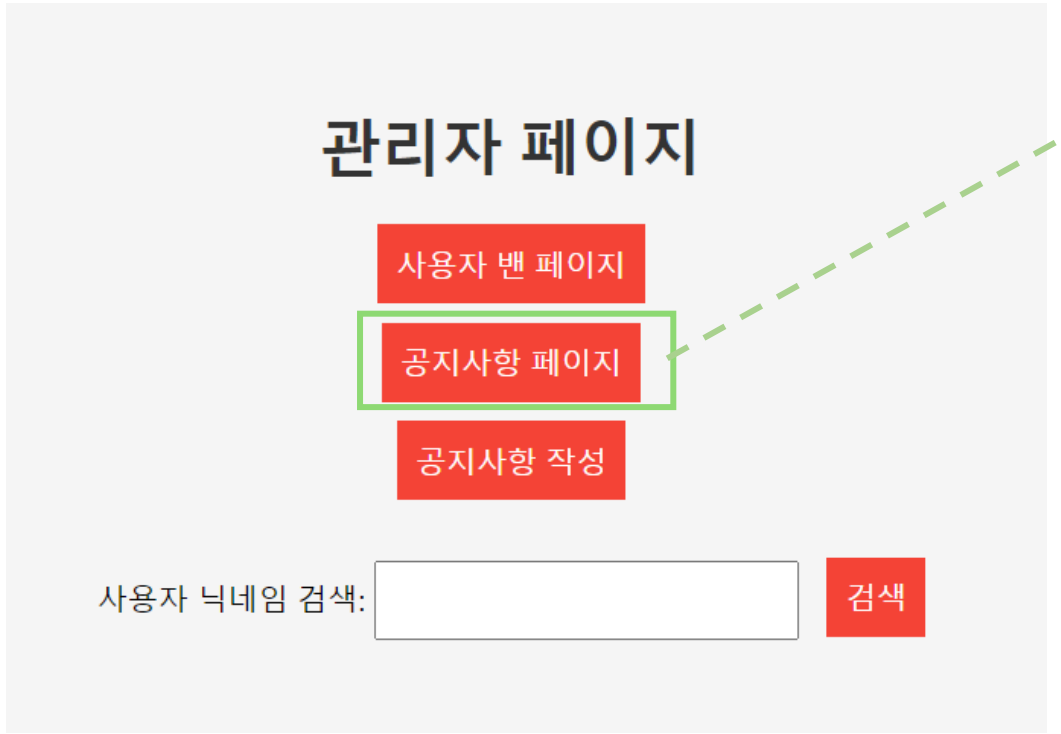
프로젝트 결과 -관리자 버전

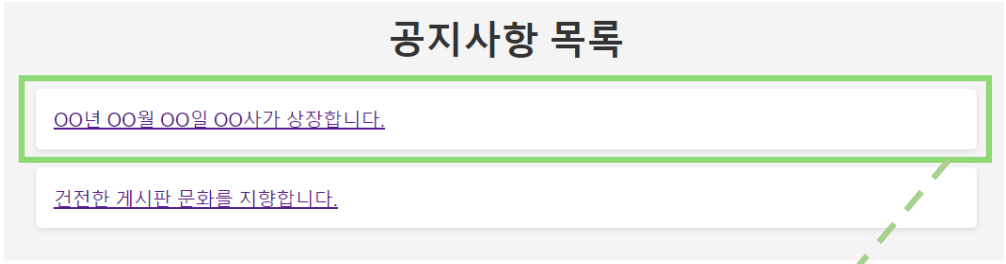
공지사항 작성 페이지

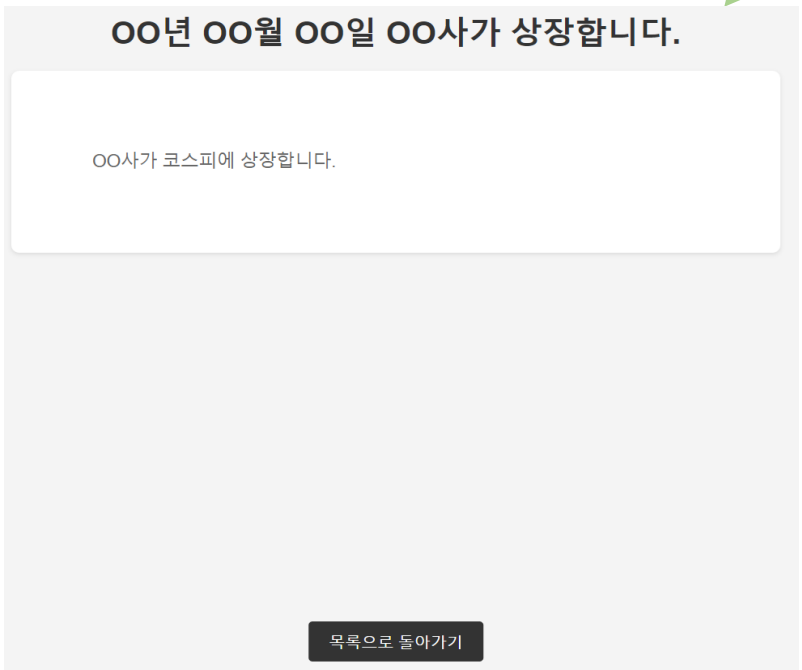


프로젝트 결과 -관리자 버전

사용자 밴 페이지

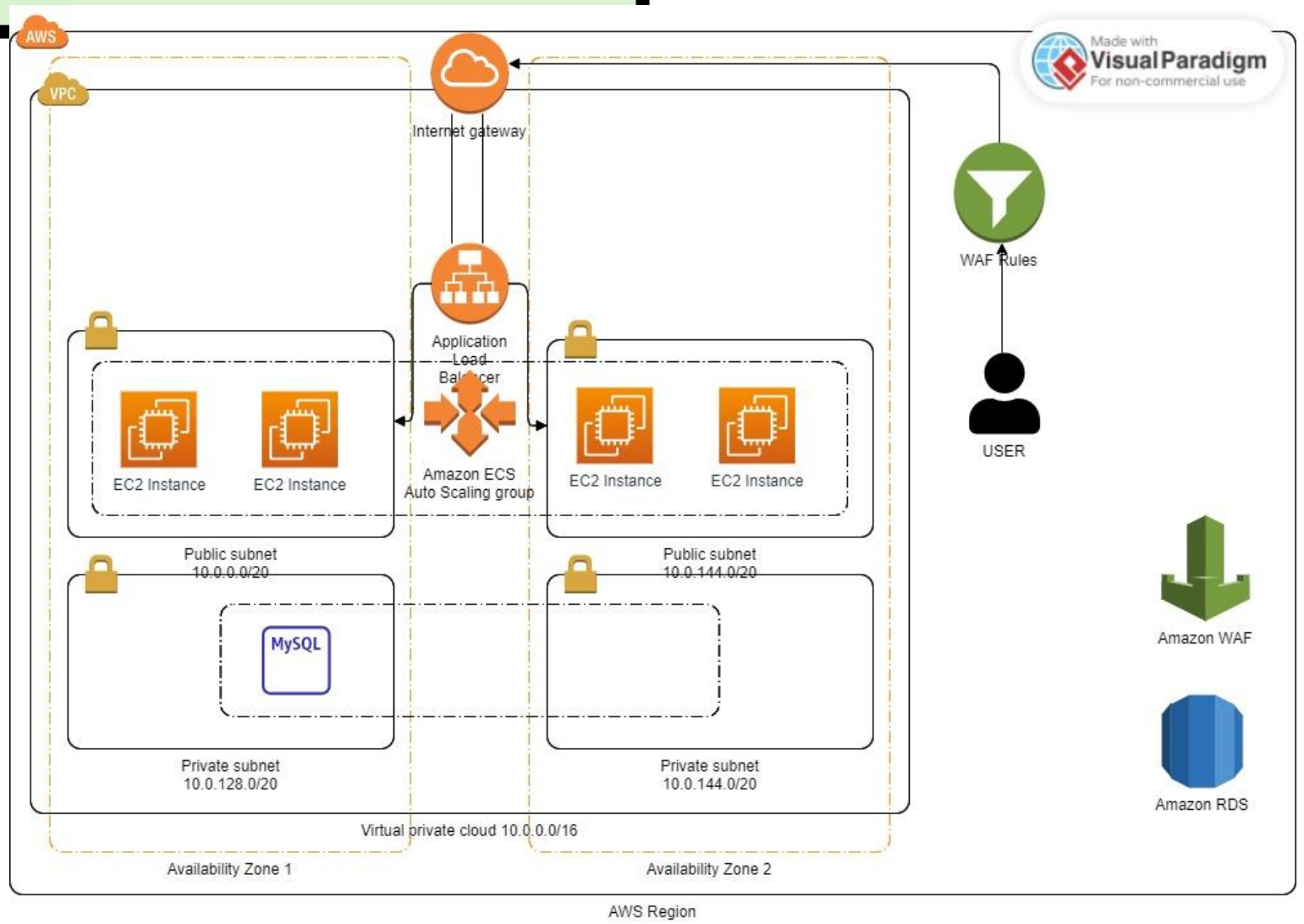






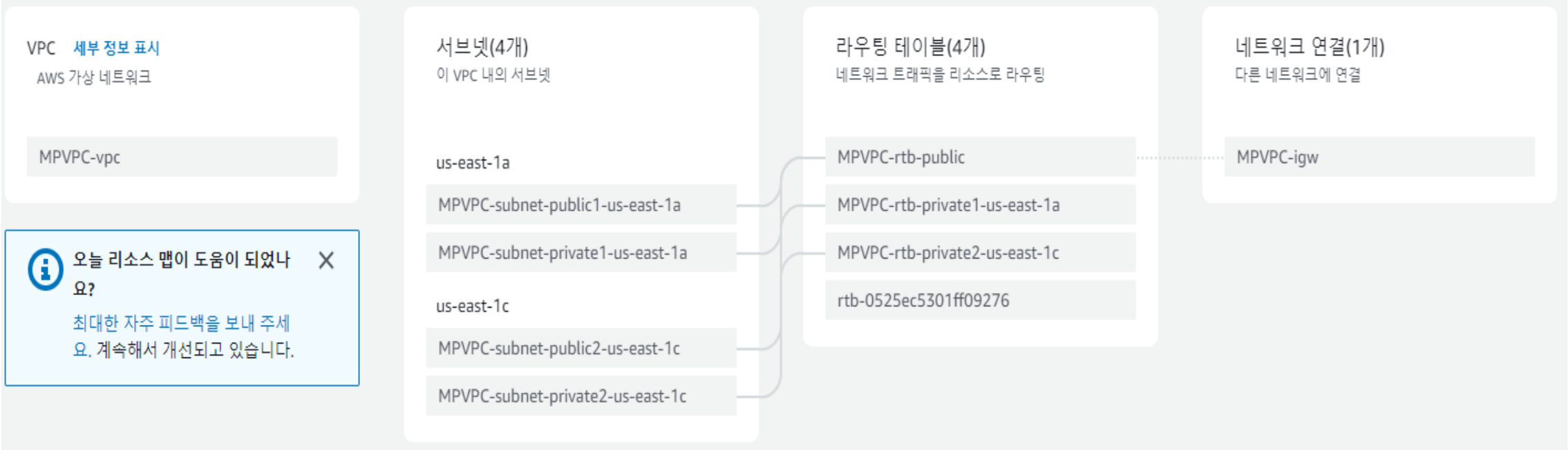
프로젝트 결과 -AWS

클라우드 아키텍쳐



프로젝트 결과 -AWS

리소스 맵



프로젝트 결과 -AWS

AWS WAF Rules

Rules (4)

Edit | Delete | Add rules ▼

Find rules

1

Name	Action	Priority	Custom response
AWS-AWSManagedRulesAmazonIpReputationList	Use rule actions	0	-
AWS-AWSManagedRulesCommonRuleSet	Use rule actions	1	-
AWS-AWSManagedRulesKnownBadInputsRuleSet	Use rule actions	2	-
AWS-AWSManagedRulesSQLiRuleSet	Use rule actions	3	-

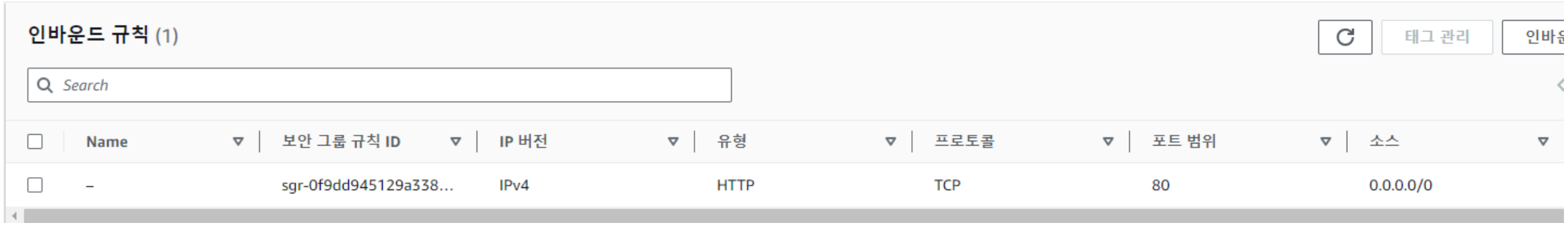
프로젝트 결과 -AWS

보안 그룹

sg-0eeae8ad179cd77cf	ec2-rds-1	vpc-0408a4fcfbc84bd5e
sg-02fda372617f526cc	MPELBSG	vpc-0408a4fcfbc84bd5e
sg-0f744cb92499c9b2e	MPSG	vpc-0408a4fcfbc84bd5e
sg-05fe7fa21870c5bc0	rds-ec2-1	vpc-0408a4fcfbc84bd5e

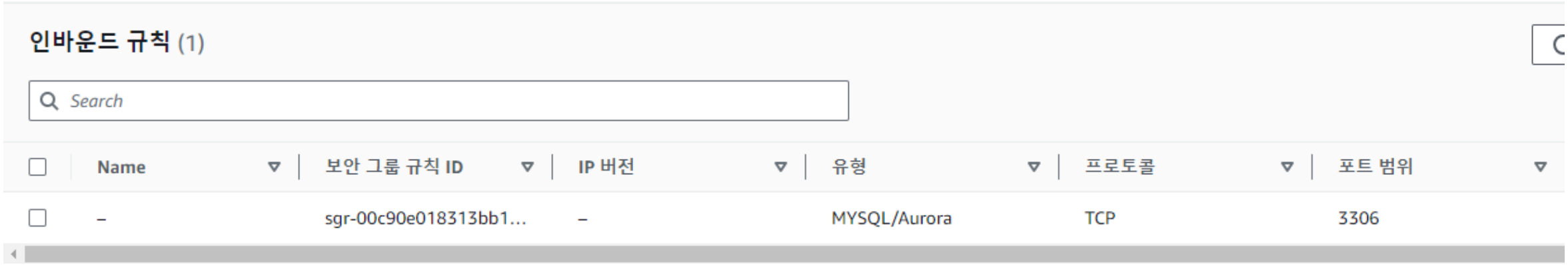
프로젝트 결과 -AWS

로드 밸런서 보안 그룹의 인바운드 규칙



프로젝트 결과 -AWS

RDS 보안 그룹의 인바운드 규칙



인바운드 규칙 (1)

Search

Name	보안 그룹 규칙 ID	IP 버전	유형	프로토콜	포트 범위
–	sgr-00c90e018313bb1...	–	MYSQL/Aurora	TCP	3306

프로젝트 결과 -AWS

인스턴스 보안 그룹의 인바운드 규칙

필터 규칙

이름	보안 그룹 규칙 ID	포트 범위	프로토콜	원본	보안 그룹
–	sgr-06e5a59c0dc0c4357	22	TCP	0.0.0.0/0	MPSG
–	sgr-0fc90b631b46ce347	80	TCP	sg-02fda372617f526cc	MPSG

▼ 아웃바운드 규칙

필터 규칙

이름	보안 그룹 규칙 ID	포트 범위	프로토콜	대상	보안 그룹
–	sgr-04f25e35e9f1d4bea	3306	TCP	sg-05fe7fa21870c5bc0	ec2-rds-1
–	sgr-03986747229784511	전체	전체	0.0.0.0/0	MPSG

프로젝트 결과 -시연 영상

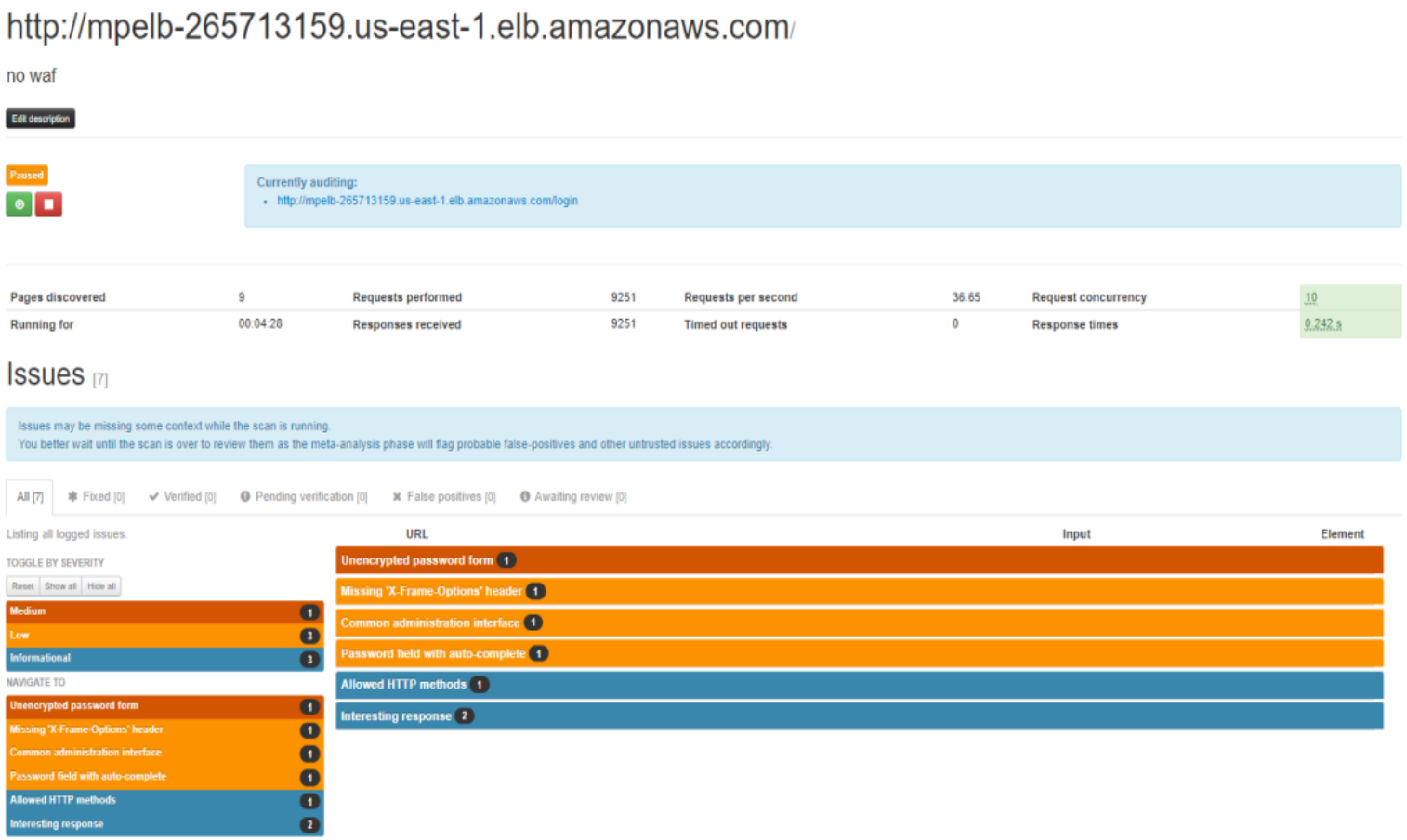
보안 검증

Checks The security checks to be run again the web application

Active These checks will actively engage the web application via its inputs (links, forms, etc.)

- Code injection (code_injection)
- Code injection (php://input wrapper) (code_injection_php_input_wrapper)
- Code injection (timing) (code_injection_timing)
- CSRF (csrf)
- File Inclusion (file_inclusion)
- LDAPInjection (ldap_injection)
- NoSQL Injection (no_sql_injection)
- Blind NoSQL Injection (differential analysis) (no_sql_injection_differential)
- OS command injection (os_cmd_injection)
- OS command injection (timing) (os_cmd_injection_timing)
- Path Traversal (path_traversal)
- Response Splitting (response_splitting)
- Remote File Inclusion (rfi)
- Session fixation (session_fixation)
- Source code disclosure (source_code_disclosure)
- SQL Injection (sql_injection)
- Blind SQL Injection (differential analysis) (sql_injection_differential)
- Blind SQL injection (timing attack) (sql_injection_timing)
- Trainer (trainer)
- Unvalidated redirect (unvalidated_redirect)
- Unvalidated DOM redirect (unvalidated_redirect_dom)
- XPath Injection (xpath_injection)
- XSS (xss)
- DOM XSS (xss_dom)
- DOM XSS in script context (xss_dom_script_context)
- XSS in HTML element event attribute (xss_event)
- XSS in path (xss_path)
- XSS in script context (xss_script_context)
- XSS in HTML tag (xss_tag)
- XML External Entity (xxe)

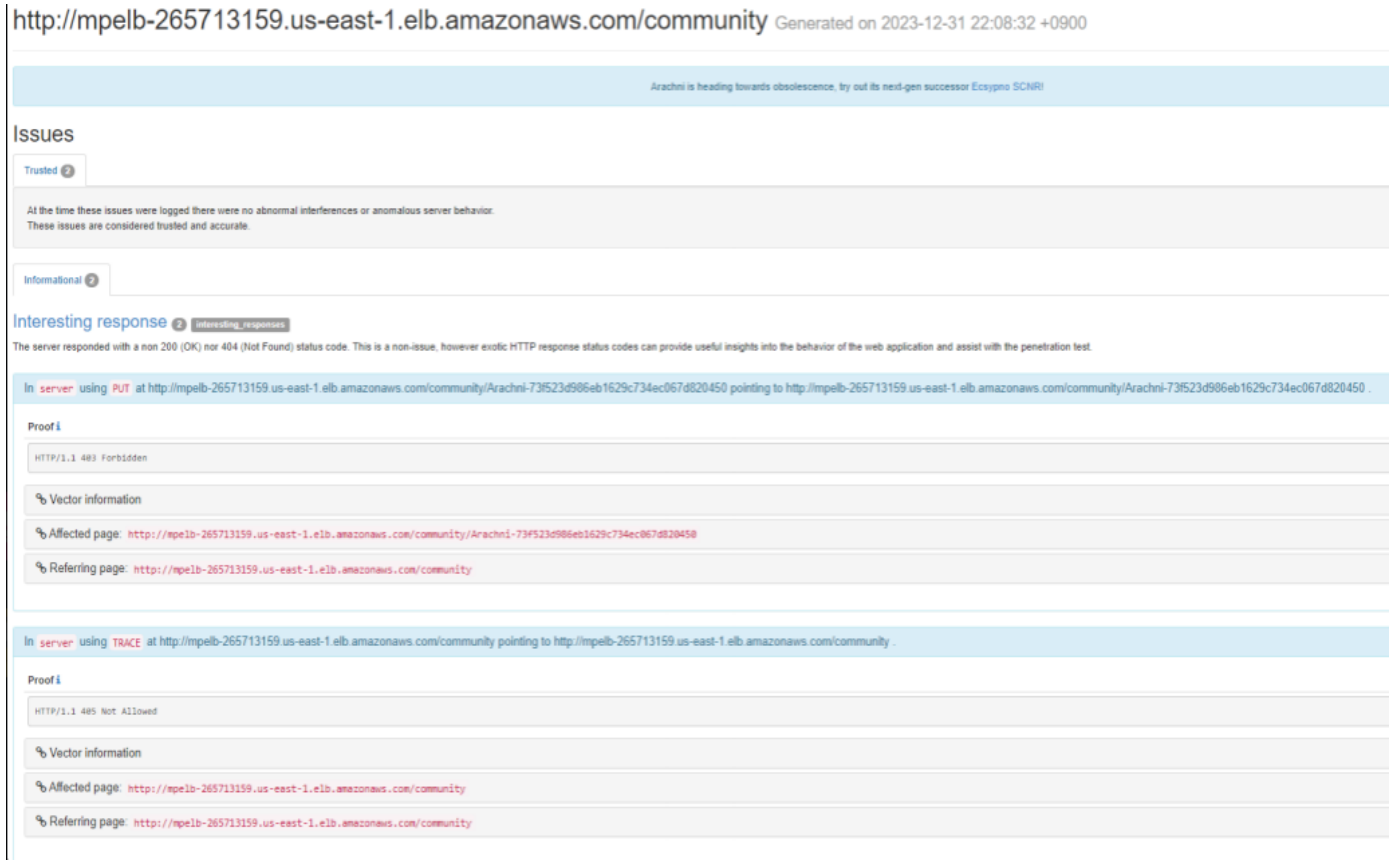
보안 검증 -WAF 규칙 적용 전



발견된 취약점

- unencrypted password form
- Missing 'X-Frame-options' header
- Common administration interface
- Password field with auto-complete
- Allowed HTTP methods
- Interesting response

보안 검증 -WAF 규칙 적용 후



http://mpelb-265713159.us-east-1.elb.amazonaws.com/community Generated on 2023-12-31 22:08:32 +0900

Arachni is heading towards obsolescence, try out its next-gen successor Ecsypno SCNR!

Issues

Trusted 2

At the time these issues were logged there were no abnormal interferences or anomalous server behavior.
These issues are considered trusted and accurate.

Informational 2

Interesting response 2 interesting_responses

The server responded with a non 200 (OK) nor 404 (Not Found) status code. This is a non-issue, however exotic HTTP response status codes can provide useful insights into the behavior of the web application and assist with the penetration test.

In server using PUT at http://mpelb-265713159.us-east-1.elb.amazonaws.com/community/Arachni-73f523d986eb1629c734ec067d820450 pointing to http://mpelb-265713159.us-east-1.elb.amazonaws.com/community/Arachni-73f523d986eb1629c734ec067d820450 .

Proof

HTTP/1.1 403 Forbidden

Vector information

Affected page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/community/Arachni-73f523d986eb1629c734ec067d820450

Referring page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/community

In server using TRACE at http://mpelb-265713159.us-east-1.elb.amazonaws.com/community pointing to http://mpelb-265713159.us-east-1.elb.amazonaws.com/community .

Proof

HTTP/1.1 405 Not Allowed

Vector information

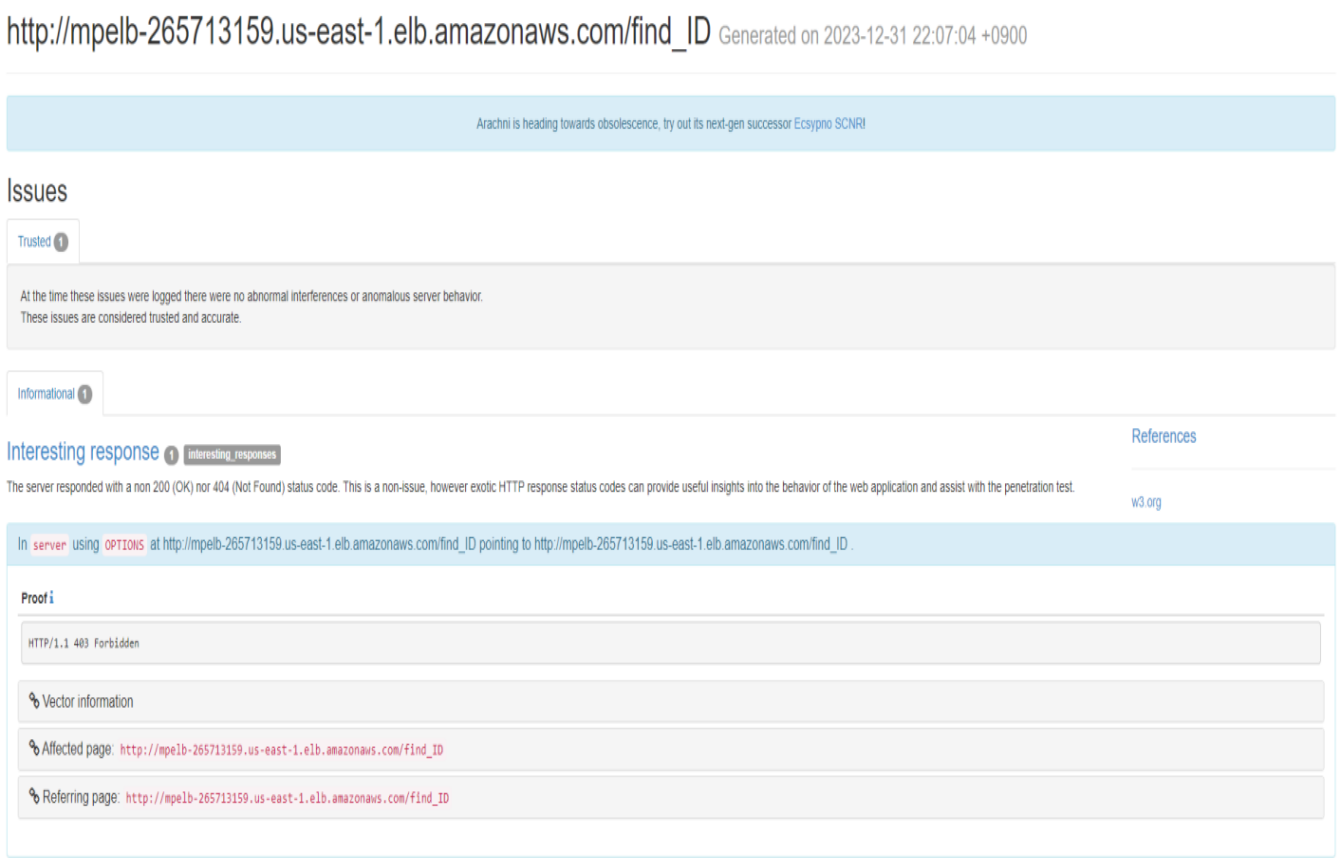
Affected page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/community

Referring page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/community

주식 종목 토론 페이지 취약점 분석

403, 405에러가 표시되며 차단당하였다.

보안 검증 -WAF 규칙 적용 후


http://mpelb-265713159.us-east-1.elb.amazonaws.com/find_ID Generated on 2023-12-31 22:07:04 +0900

Arachni is heading towards obsolescence, try out its next-gen successor Ecsypno SCNR!

Issues

Trusted 1

At the time these issues were logged there were no abnormal interferences or anomalous server behavior.
These issues are considered trusted and accurate.

Informational 1

Interesting response 1 interesting_responses

The server responded with a non 200 (OK) nor 404 (Not Found) status code. This is a non-issue, however exotic HTTP response status codes can provide useful insights into the behavior of the web application and assist with the penetration test.

References

w3.org

In server using OPTIONS at http://mpelb-265713159.us-east-1.elb.amazonaws.com/find_ID pointing to http://mpelb-265713159.us-east-1.elb.amazonaws.com/find_ID .

Proof

HTTP/1.1 403 Forbidden

Vector information

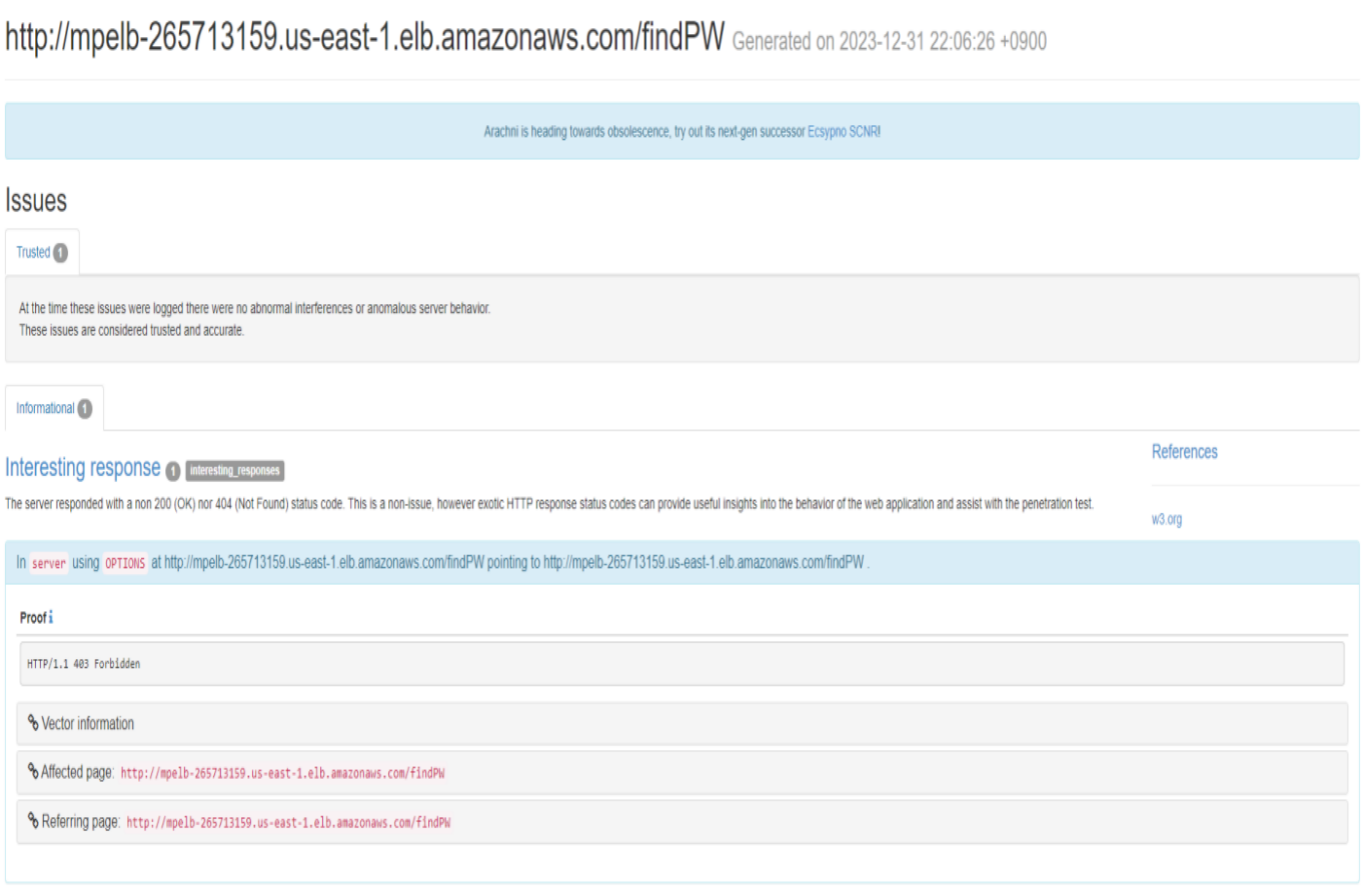
Affected page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/find_ID

Referring page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/find_ID

ID 찾기 페이지 취약점 분석

403, 405에러가 표시되며 차단당하였다.

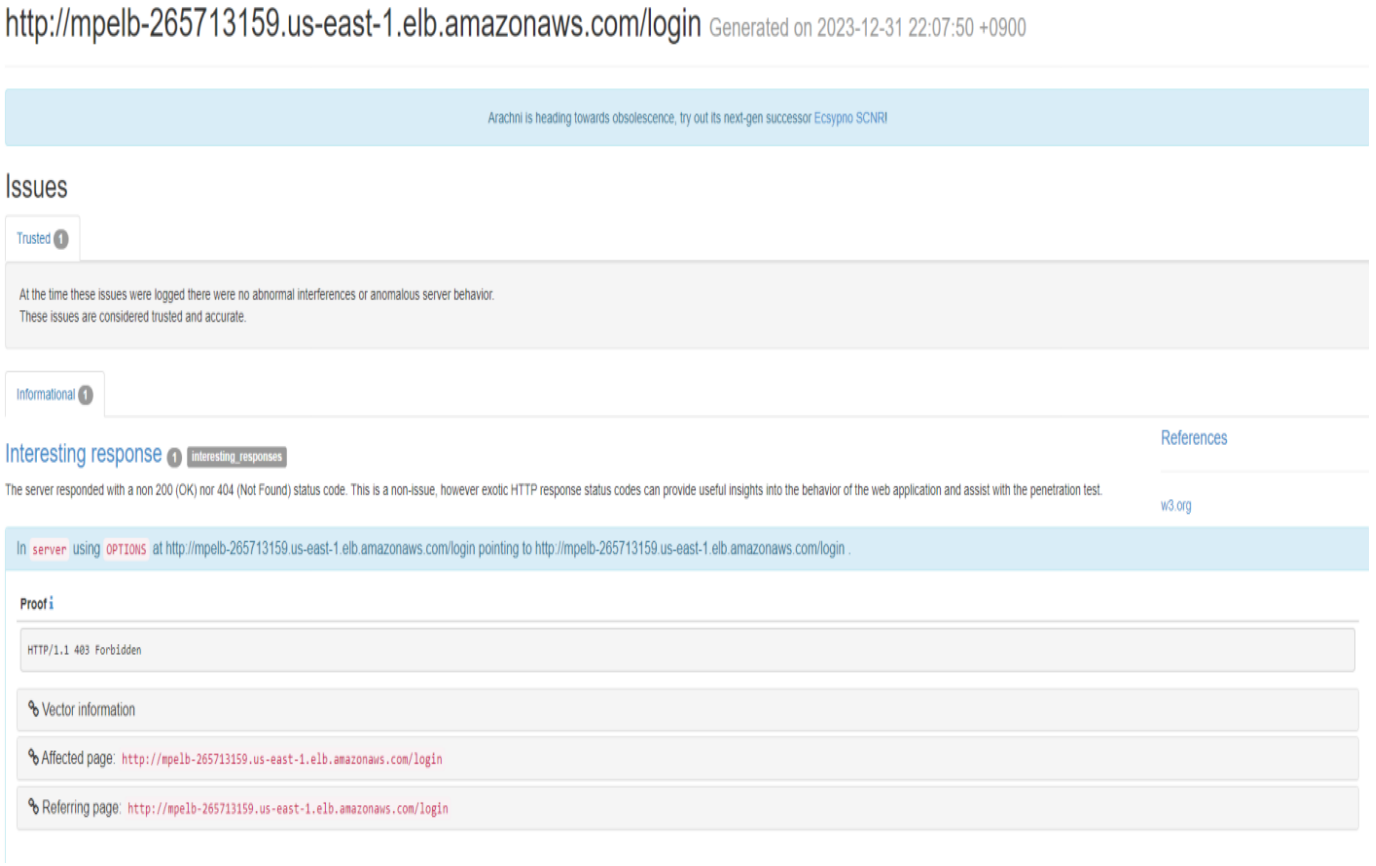
보안 검증 -WAF 규칙 적용 후



PW 찾기 페이지 취약점 분석

403, 405에러가 표시되며 차단당하였다.

보안 검증 -WAF 규칙 적용 후



http://mpelb-265713159.us-east-1.elb.amazonaws.com/login Generated on 2023-12-31 22:07:50 +0900

Arachni is heading towards obsolescence, try out its next-gen successor Ecsypno SCNR

Issues

Trusted 1

At the time these issues were logged there were no abnormal interferences or anomalous server behavior.
These issues are considered trusted and accurate.

Informational 1

Interesting response 1 interesting_responses

The server responded with a non 200 (OK) nor 404 (Not Found) status code. This is a non-issue, however exotic HTTP response status codes can provide useful insights into the behavior of the web application and assist with the penetration test.

References

w3.org

In server using OPTIONS at http://mpelb-265713159.us-east-1.elb.amazonaws.com/login pointing to http://mpelb-265713159.us-east-1.elb.amazonaws.com/login .

Proof

HTTP/1.1 403 Forbidden

Vector information

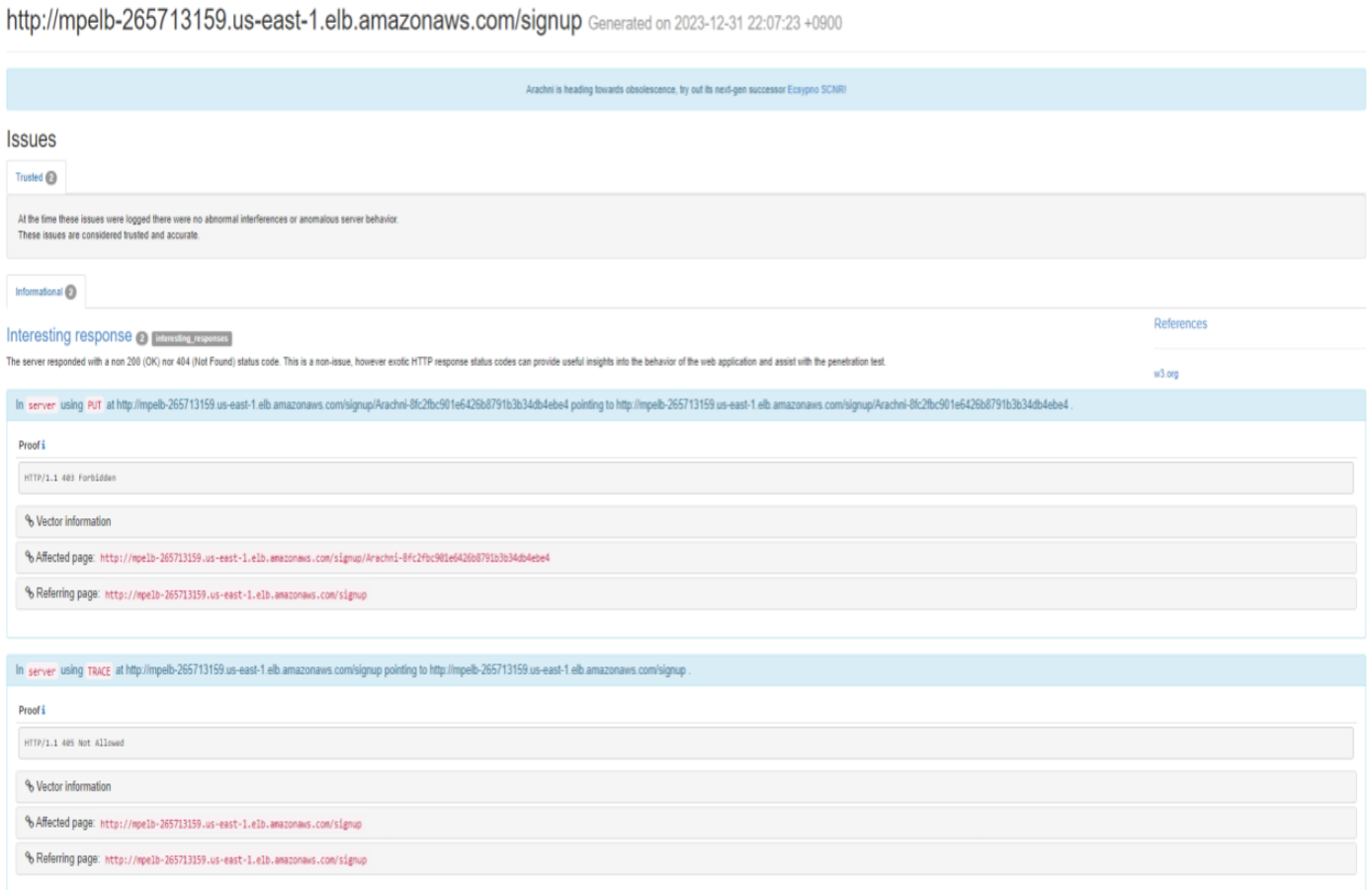
Affected page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/login

Referring page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/login

로그인 페이지 취약점 분석

403, 405에러가 표시되며 차단당하였다.

보안 검증 -WAF 규칙 적용 후



http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup Generated on 2023-12-31 22:07:23 +0900

Arachni is heading towards obsolescence, try out its next-gen successor Ecsypno SCNR!

Issues

Trusted

At the time these issues were logged there were no abnormal interferences or anomalous server behavior.
These issues are considered trusted and accurate.

Informational

Interesting response interesting_responses

The server responded with a non 200 (OK) nor 404 (Not Found) status code. This is a non-issue, however exotic HTTP response status codes can provide useful insights into the behavior of the web application and assist with the penetration test.

References

w3.org

In server using PUT at http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup/Arachni-8fc2fbc901e6426b8791b3b34db4ebe4 pointing to http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup/Arachni-8fc2fbc901e6426b8791b3b34db4ebe4 .

Proof

HTTP/1.1 403 Forbidden

Vector information

Affected page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup/Arachni-8fc2fbc901e6426b8791b3b34db4ebe4

Referring page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup

In server using TRACE at http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup pointing to http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup .

Proof

HTTP/1.1 405 Not Allowed

Vector information

Affected page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup

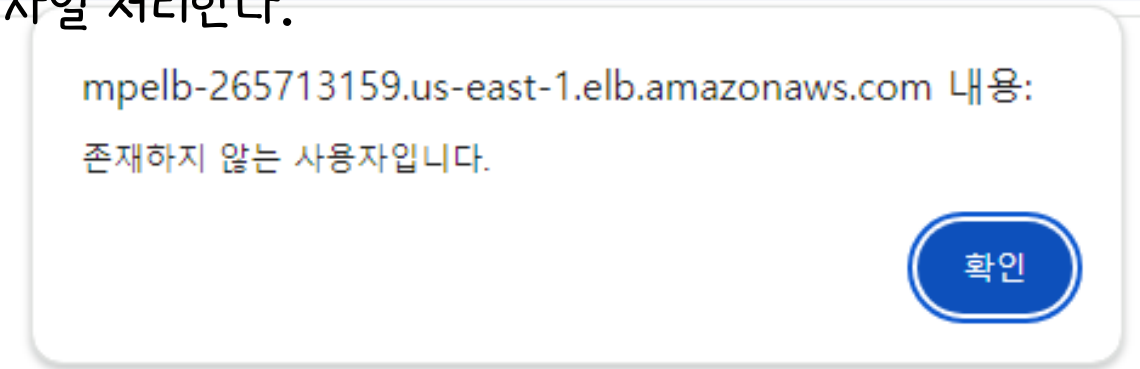
Referring page: http://mpelb-265713159.us-east-1.elb.amazonaws.com/signup

회원가입 페이지 취약점 분석

403, 405에러가 표시되며 차단당하였다.

보안 검증 -WAF 규칙 적용 전

WAF나 기타 secure code를
적용하기 전 이지만 Flask에서
XSS구문을 문자열 처리한다.



보안 검증 -WAF 규칙 적용 후

403 Forbidden

WAF에 의해서 공격 시도가
아예 차단 된다.

보안 검증 -Kali를 이용한 포트스캔

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS 3.214.116.126
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-31 03:36 EST
Nmap scan report for ec2-3-214-116-126.compute-1.amazonaws.com (3.214.116.126)
Host is up (0.017s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 48.66 seconds
```

WAF가 꺼져 있어 포트스캔으로 정보를 획득 할 수 있었다

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS 3.214.116.126
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-31 03:29 EST
```

WAF가 켜져 있어 포트 스캔을 차단했다.

자체 평가

GOOD POINT

- AWS WAF 활용: AWS WAF의 검증된 규칙을 활용하여, 익숙하지 않은 취약점에 대해서도 강력한 보안을 적용한다. 악성 공격으로부터 시스템을 효과적으로 보호할 수 있다.

-관리자 계정의 안전성 강화: 관리자 계정에 대한 2차 인증을 이메일 인증을 통해 수행하여, 보다 안전한 관리자 인증을 제공한다. 이를 통해 관리자 계정의 무단 액세스를 방지하고 시스템의 안전성을 높일 수 있다.

-회원가입 본인 인증: 회원가입 시 이메일을 활용한 본인 인증을 도입하여, 악성 이용자의 재가입 문제를 예방할 수 있다. 이는 시스템의 무단 접근을 줄이고 보안을 강화할 수 있다.

-강력한 WAF 보안 강화벽 활용: 강력한 WAF 보안 강화벽을 통해 시스템을 더욱 안전하게 보호한다.

자체 평가

BAD POINT

- 서버 이원화 부재로 인한 안전성 저하
- 이미지 업로드 파일 용량 제한: WAF 규칙으로 인해 게시판에 7KB 이상의 이미지 업로드가 불가능하다.
  → Request 바디부분 검증을 위한 의도적인 부분
- 이미지 업로드 시 파일명 중복으로 인한 문제: 업로드한 이미지의 파일명이 같으면 기존 이미지가 바뀌는 현상이 발생한다.
- 회원가입 시 다양한 본인 인증 수단 부재: 회원가입 시 본인 인증 수단이 이메일 뿐이어서, 다른 이메일로 가입 시도를 차단할 수 없다.

감사합니다!