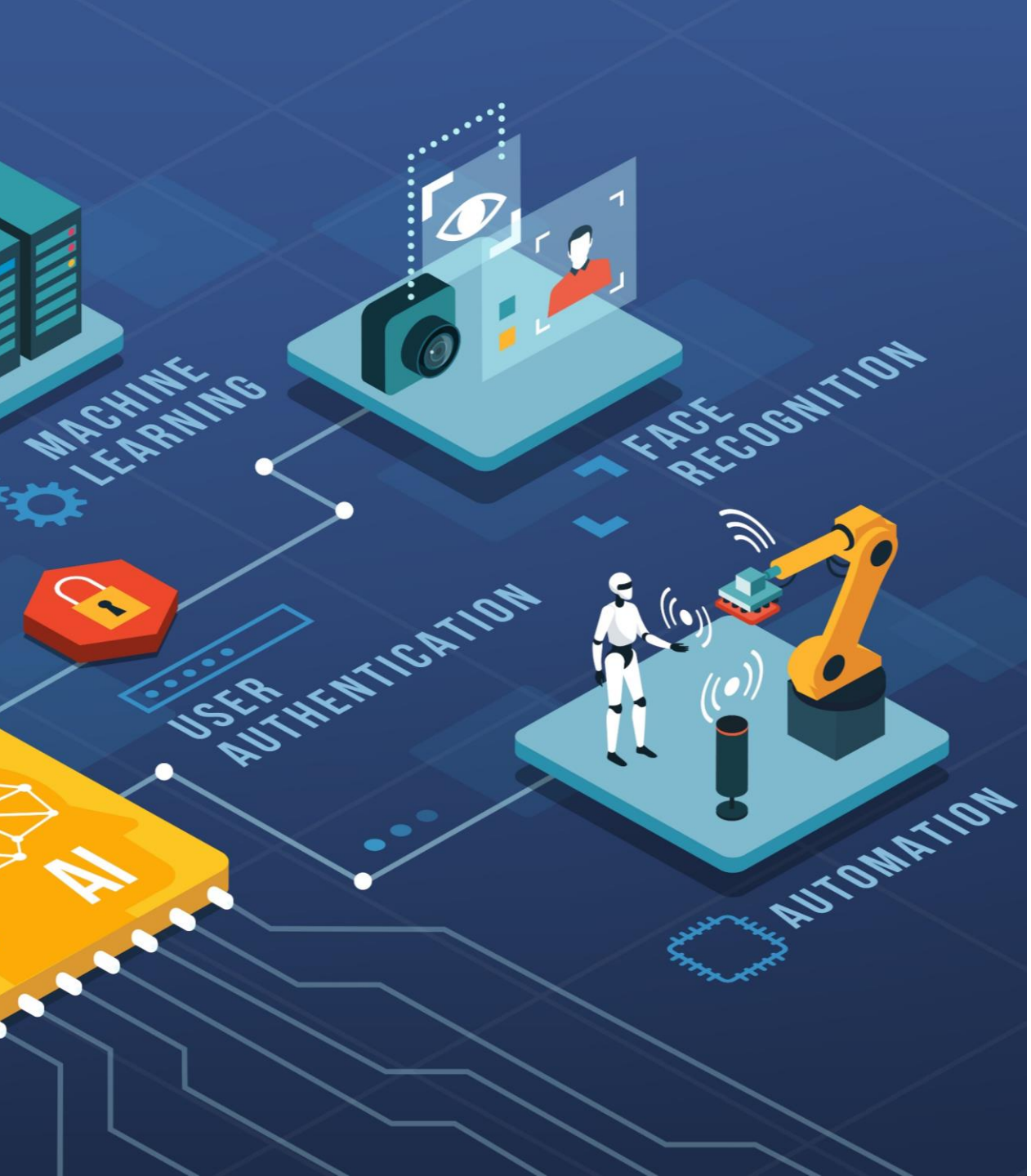
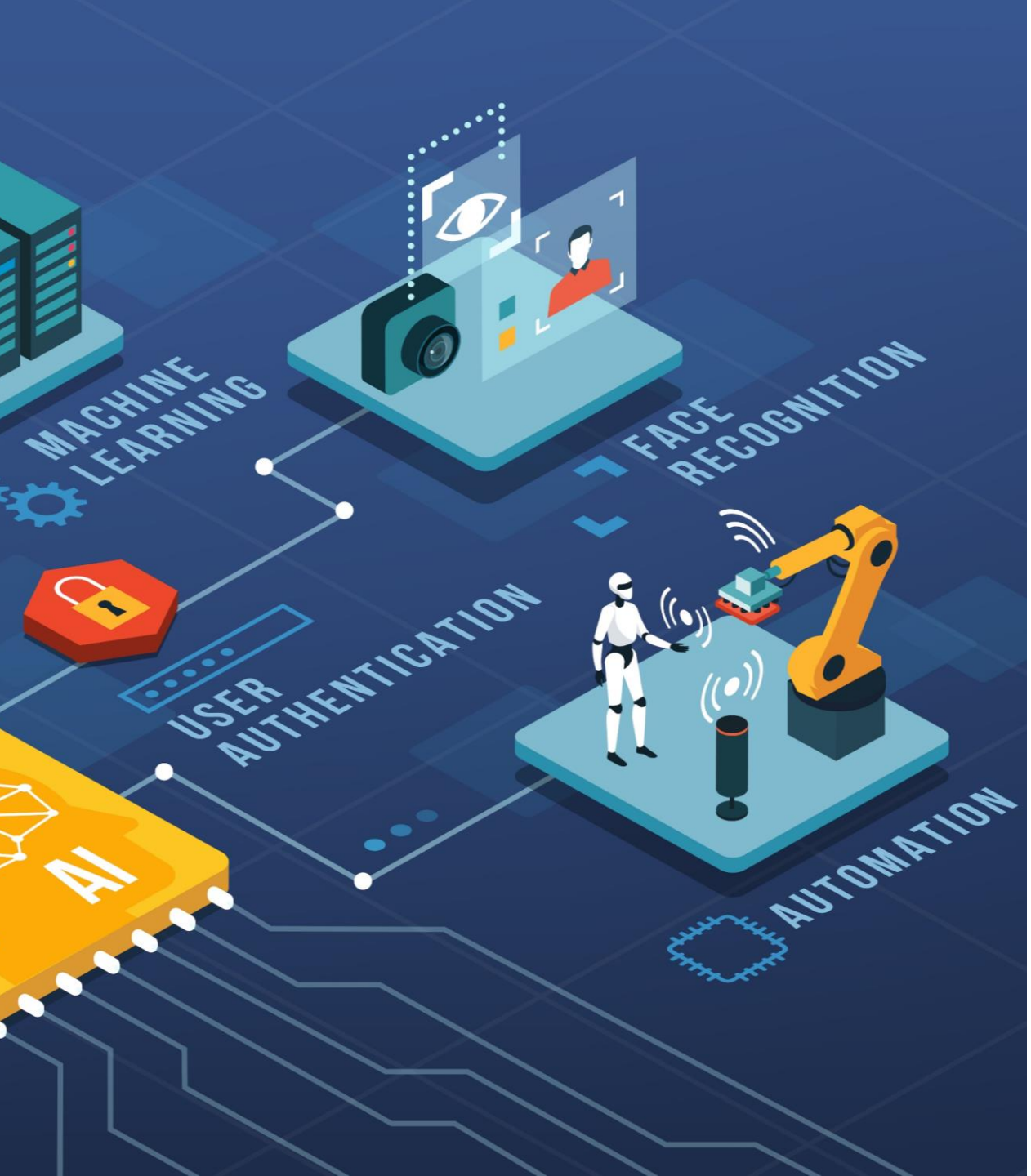


웹 취약점 자동 진단 도구 제작

6조 (저희가 할 수 있겠조?)

2022.05.24



목 차

01 프로젝트 개요

- 프로젝트 배경 및 필요성
- 프로젝트 주제 및 목적
- 팀원 소개 및 역할 분담

01 프로젝트 배경 및 필요성

취약점 및 운영자의 실수로 인한 웹 사이트에서의 개인정보 침해 신고 및 상담 건수가 나날이 증가

- 개인정보유출 피해의 75%가 외부 해킹이 원인이며, **가장 취약한 웹에서부터 시작**
- 코로나19 창궐 후 특히 사이버 상에서는 **다양한 사건·사고**가 발생
- 웹사이트는 사용자들에게 항상 **노출되어 있기 때문에** 보안 사고가 끊임없이 발생
- 기업들과 개인들은 웹 보안에 **경각심**을 가져야 함

02 프로젝트 주제 및 목적

자동화 도구를 이용한 효율적인 웹 진단

"웹 취약점 자동 진단 도구를 이용한 취약점 진단 및 보고서 제작"

잠재적 취약점
발생 위치 **파악 및 대처**

취약점 진단 시간을
단축하여
효율적인 진단 가능

고객사에게
효율적이고 친절한
진단 보고서 제공

6조 팀원 소개 및 역할 분담

이름	역할 분담
이승재(팀장)	일정 수립 및 테스트 웹 사이트 제작
범채윤	자동 진단 도구 개발
이경서	보고서 제작 및 테스트 웹 사이트 제작
이정현	자동 진단 도구 개발
이진솔	자동 진단 도구 개발
한지호	자동 진단 도구 개발 및 웹 보고서 작성
공통	주요정보통신기반시설 웹 취약점 연구

02 프로젝트 진행

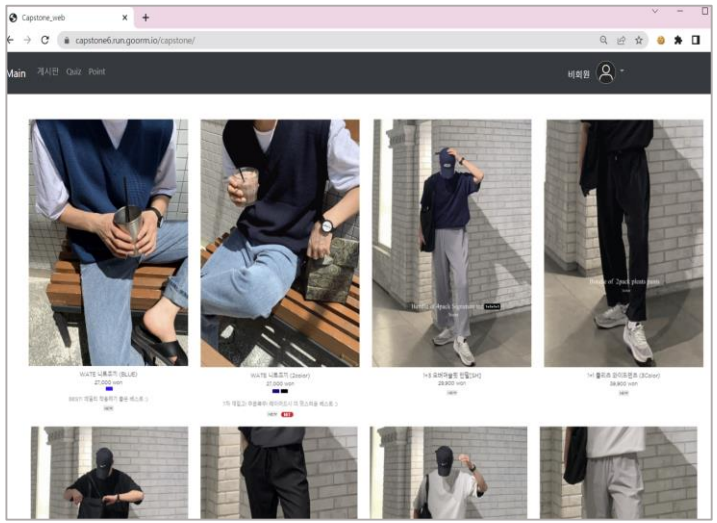
- 프로젝트 구성도
- 프로젝트 진행방법
- 자동 진단 도구 제작
- 웹 사이트(test bed) 제작
- PDF 보고서 제작

자동진단도구 프로젝트에 대한 개략적인 구성도

02 프로젝트 진행 방법

세 단계로 나누어 진행 - 테스트 베드 제작, 자동 진단 도구 제작, 보고서 제작

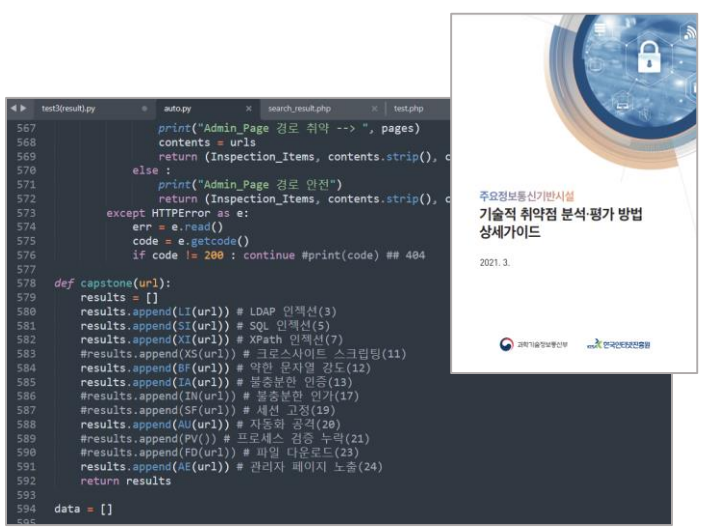
1. 웹 사이트(Test Bed) 제작



필요성

1. 자동 진단 도구 테스트 대상
2. 웹 사이트에 대한 원리 파악
3. 팀원들의 모의해킹 역량 증대

2. 자동 진단 도구 제작



필요성

주요정보통신기반시설 가이드 기준으로
대상 사이트의 효율적인 진단을 위함

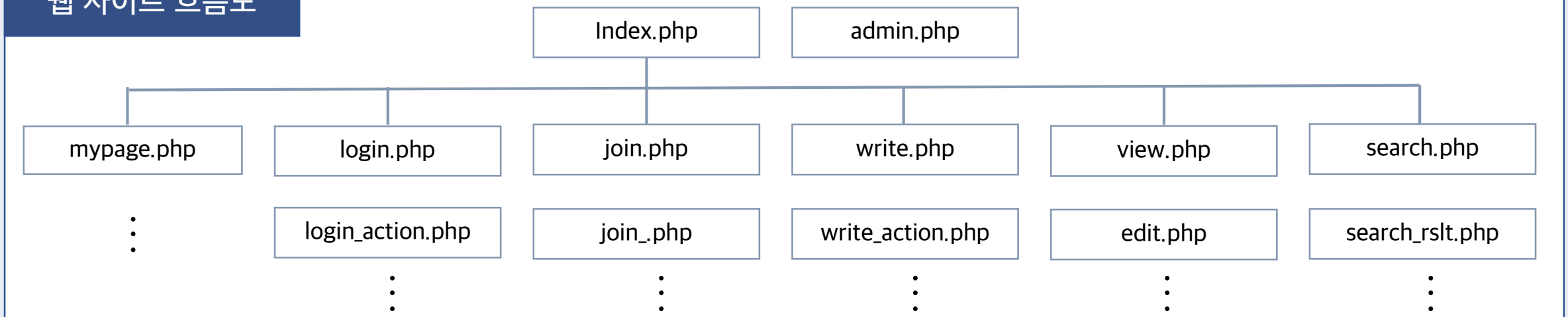
3. 보고서 제작

필요성

사용자에게 보다 친절한 보고서를
제공해주기 위함

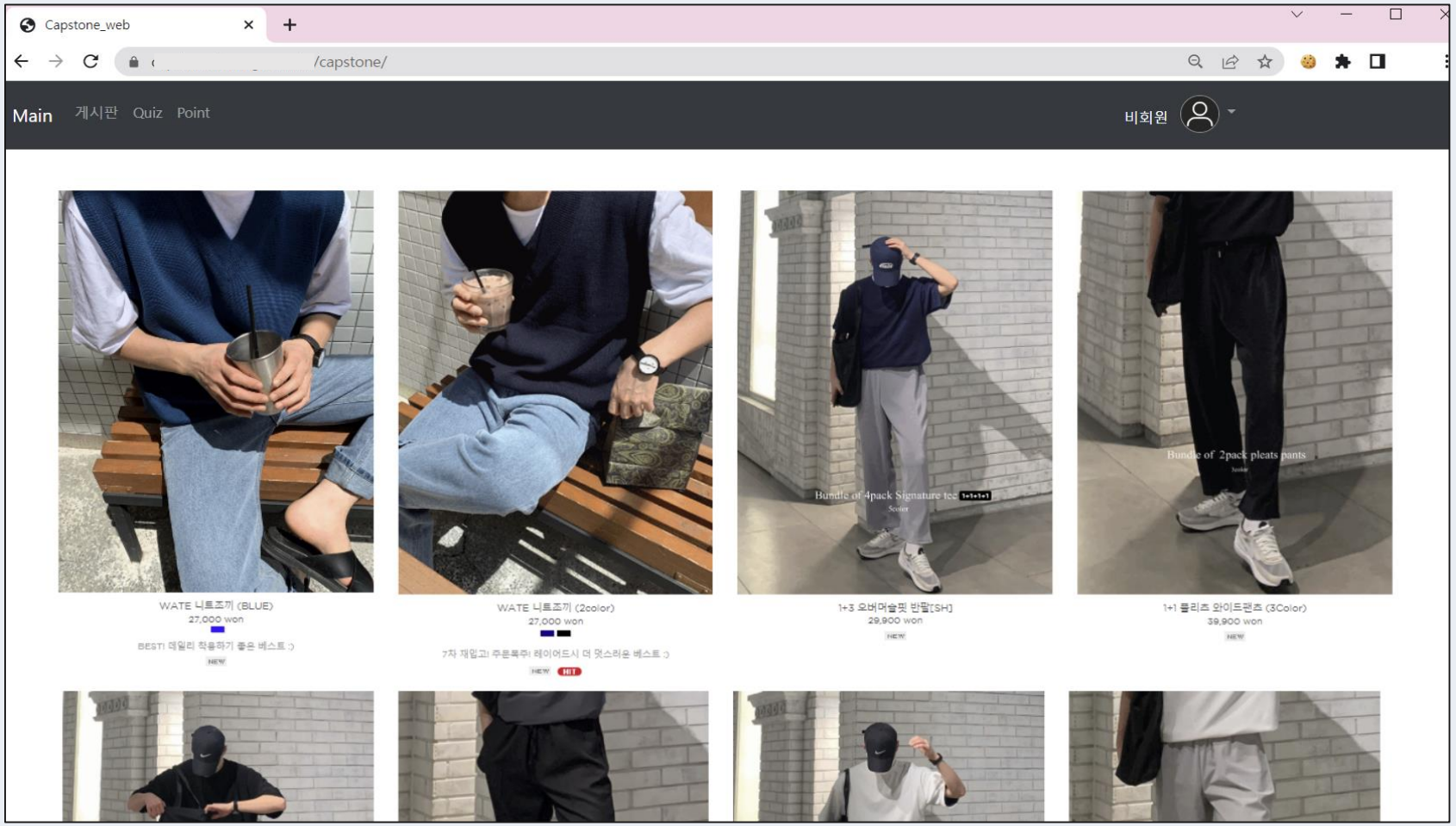
APM(Apache+php+MySQL)서버를 이용하여 자동화 도구 테스트베드 사이트 제작

웹 사이트 흐름도



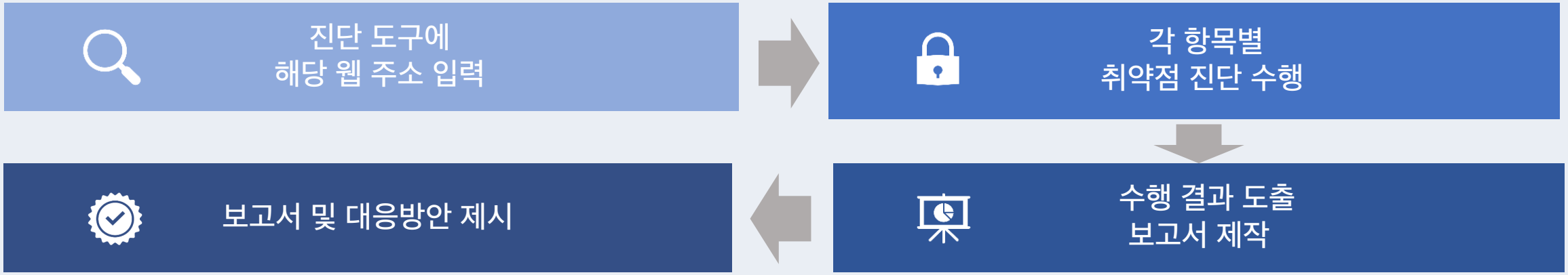
다양한 기능이 있는 쇼핑몰 컨셉의 테스트베드 사이트 제작 완료

- 주요 기능
- 회원가입/로그인
- 마이페이지
- 게시판
- 검색 기능
- 관리자페이지



04 자동 진단 도구 제작

주요정보통신기반시설 가이드 기준의 웹 공격유형을 바탕으로 Python을 이용한 자동 진단 도구 제작



항목 번호	스크립트 제작 완료된 항목
3	LDAP 인젝션
5	SQL 인젝션
7	XPath 인젝션
8	디젝터리 인덱싱
11	크로스사이트 스크립팅
12	약한 문자열 강도
13	불충분한 인증
17	불충분한 인가
19	세션 고정
20	자동화 공격
21	프로세스 검증 누락
23	파일 다운로드
24	관리자 페이지 노출

예시1 - XSS(크로스사이트스크립팅)

```
def XS(domain): # XSS(11)
    Inspection_Items = "XSS(11)"
    contents = ""
    cve = "Safety"

    urls = "http://"+domain+"/board.php"
    XPATH_click = "/html/body/div[1]/form/button"

    lines = ["<script>alert('XSS Risk')</script>",
             '"><script>alert(XSS Risk)</script>']
    count = len(lines)
    for payload in lines:
        try:
            driver.get(urls)
            input_box = driver.find_element(By.NAME, "search")
            input_box.send_keys(payload)
            driver.find_element(By.XPATH, XPATH_click).click()
        except UnexpectedAlertPresentException:
            time.sleep(2)
            count -= 1
    if count > 0:
        cve = "Risk"
        writer.addPage(report.getPage(30))
        writer.addPage(report.getPage(31))
        writer.addPage(report.getPage(32))
        writer.addPage(report.getPage(33))
        writer.addPage(report.getPage(34))
        contents = urls
        print("XSS 취약")
        return (Inspection_Items, contents.strip(), cve)
    else:
        print("XSS 안전")
        contents = "This website is \"SAFETY\" from Cross Site Scripting"
        return (Inspection_Items, contents.strip(), cve)
```

Reflected XSS 공격 기법 진단

Payload 에 공격 구문 작성 후 공격 시도 ➔ 반응 나타날 시, 취약하다고 판단

예시2 - 세션 고정

```python
def SF(domain): # 세션 고정(19)
    Inspection_Items = "SF(19)"
    contents = "This website is \"SAFETY\" from SF"
    cve = "Safety"

    urls = "http://"+domain+"/login.php"

    XPATH_id = "/html/body/div/form/p[1]/input"
    XPATH_pw = "/html/body/div/form/p[2]/input"
    XPATH_click = "/html/body/div/form/input"
    XPATH_nav = '//*[@id="navbarDropdown"]'
    XPATH_logout = '//*[@id="collapsibleNavbar"]/ul/li[5]/div/a[1]'

    driver.get(urls)
    driver.maximize_window()
    input_box = driver.find_element(By.XPATH, XPATH_id)
    input_box.send_keys('test')
    input_box2 = driver.find_element(By.XPATH, XPATH_pw)
    input_box2.send_keys('test')
    driver.find_element(By.XPATH, XPATH_click).click()
    alert = driver.switch_to.alert
    alert.accept()

    for cookie in driver.get_cookies():
        c = {cookie['name'] : cookie['value']}
    driver.find_element(By.XPATH, XPATH_nav).click()
    driver.find_element(By.XPATH, XPATH_logout).click()
    # alert = driver.switch_to.alert
    # alert.accept()

    driver.get(urls)
    driver.maximize_window()
    input_box = driver.find_element(By.XPATH, XPATH_id)
    input_box.send_keys('test')
    input_box2 = driver.find_element(By.XPATH, XPATH_pw)
    input_box2.send_keys('test')
    driver.find_element(By.XPATH, XPATH_click).click()
    alert = driver.switch_to.alert
    alert.accept()

    for cookie in driver.get_cookies():
        a = {cookie['name'] : cookie['value']}

    if c == a:
        cve = "Risk"
        writer.addPage(report.getPage(50))
        print("세션 고정 취약")
        contents = urls
        return (Inspection_Items, contents.strip(), cve)
    else:
```

로그인 시 발급받은 세션 ID 가 로그인 전/후 모두 동일하게 사용된 경우 취약 하다고 판단

URL에서 파라미터 값을 변조하여 공격 ➔ 성공, 실패 여부 판단

예시3 - 불충분한 인가

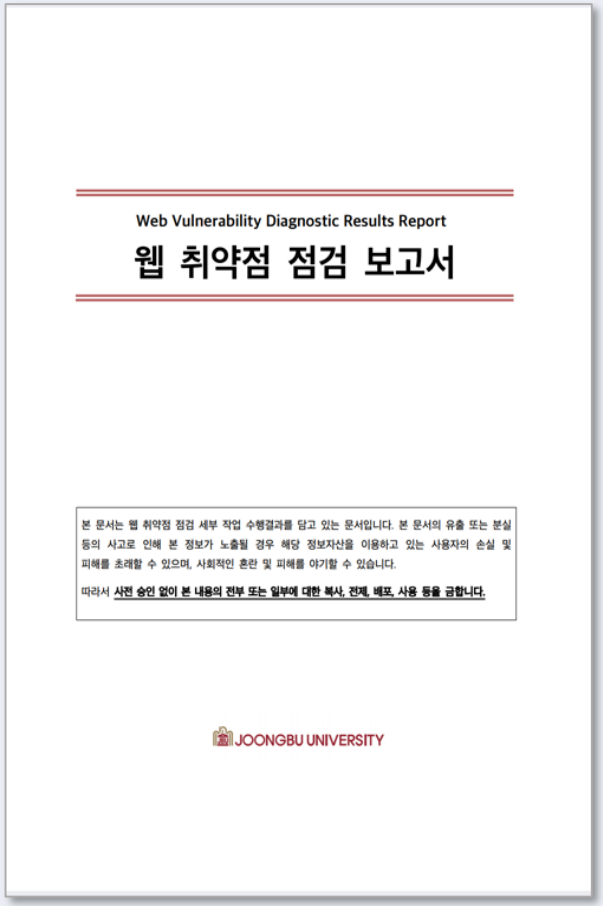
```
def IN(domain): #불충분한 인가(17)
    Inspection_Items = "IN(17)"
    contents = "This Website \"Risk\" from IN"
    cve = "Risk"
    msg = "불충분한 인가 취약"

    urls = "http://"+domain+"/board.php"
    sourcecode = urllib.request.urlopen(urls).read()
    soup = BeautifulSoup(sourcecode, "html.parser")
    li=[0 for i in range(3)]

    for href in soup.find("tr", class_="even").find_all("tbody"):
        attr = href.find("a")["href"]
        if "number" in attr:
            num = li.split('=')
            li.append(num[1])

    if li[0] == li[0]+1:
        msg = "불충분한 인가 취약"
        cve = "Risk"
        contents = "This website is \"Risk\" from IN"
        writer.addPage(report.getPage(45))
        writer.addPage(report.getPage(46))
        print(msg)
        return (Inspection_Items, contents.strip(), cve)
    try:
        driver.get(urls)
        writer.addPage(report.getPage(45))
        writer.addPage(report.getPage(46))
        print(msg)
        return (Inspection_Items, contents.strip(), cve)
    except UnexpectedAlertPresentException:
        time.sleep(1)
        msg = "불충분한 인가 안전"
        cve = "Safety"
```

다른 사용자의 비밀글에 인증을 하지 않고 접근할 수 있는지에 대한 여부

URL에서 파라미터 값을 변조하여 공격 ➔ 성공, 실패 여부 판단

05 보고서 제작 현황

Python 모듈을 이용한 보고서 제작



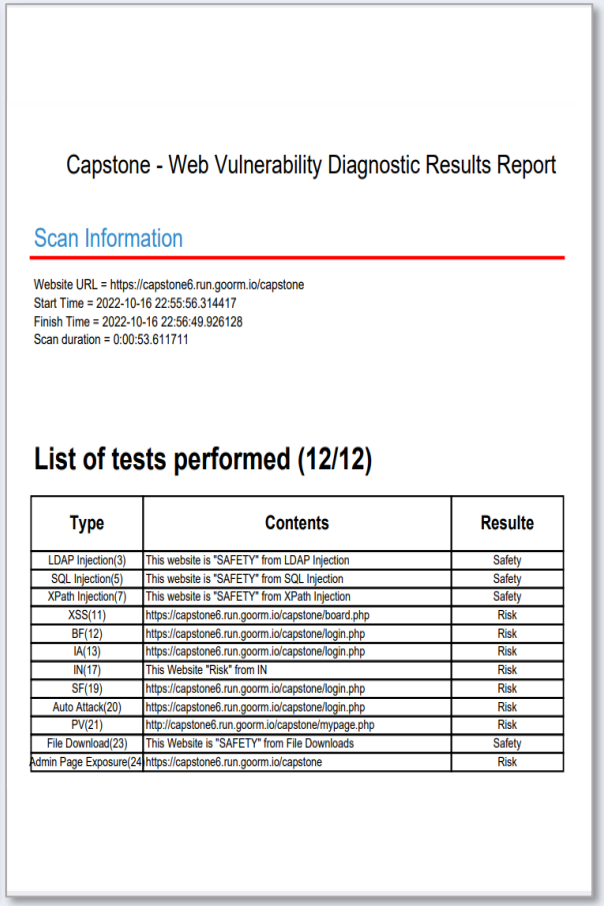
Web Vulnerability Diagnostic Results Report

웹 취약점 점검 보고서

본 문서는 웹 취약점 점검 세부 작업 수행결과를 담고 있는 문서입니다. 본 문서의 유출 또는 분실 등의 사고로 인해 본 정보가 노출될 경우 해당 정보자산을 이용하고 있는 사용자의 손실 및 피해를 초래할 수 있으며, 사회적인 혼란 및 피해를 야기할 수 있습니다.

따라서 **사전 승인 없이 본 내용의 전부 또는 일부에 대한 복사, 전재, 배포, 사용 등을 금합니다.**

JOONGBU UNIVERSITY



Capstone - Web Vulnerability Diagnostic Results Report

Scan Information

Website URL = https://capstone6.run.goorm.io/capstone
Start Time = 2022-10-16 22:55:56.314417
Finish Time = 2022-10-16 22:56:49.926128
Scan duration = 0:00:53.611711

List of tests performed (12/12)

Type	Contents	Resulte
LDAP Injection(3)	This website is "SAFETY" from LDAP Injection	Safety
SQL Injection(5)	This website is "SAFETY" from SQL Injection	Safety
XPath Injection(7)	This website is "SAFETY" from XPath Injection	Safety
XSS(11)	https://capstone6.run.goorm.io/capstone/board.php	Risk
BF(12)	https://capstone6.run.goorm.io/capstone/login.php	Risk
IA(13)	https://capstone6.run.goorm.io/capstone/login.php	Risk
IN(17)	This Website "Risk" from IN	Risk
SF(19)	https://capstone6.run.goorm.io/capstone/login.php	Risk
Auto Attack(20)	https://capstone6.run.goorm.io/capstone/login.php	Risk
PV(21)	http://capstone6.run.goorm.io/capstone/mypage.php	Risk
File Download(23)	This Website is "SAFETY" from File Downloads	Safety
Admin Page Exposure(24	https://capstone6.run.goorm.io/capstone	Risk

목적

보다 효율적인 진단 보고서 제작하고, 사용자에게 친절하게 제공하기 위함

사용 언어 및 모듈

Python - FPDF, PyPDF2

생성 방법

- 진단이 끝난 후 공격에 대한 보고서를 PDF로 생성
- 모듈을 이용하여 취약한 부분에 대해 PDF를 추출 및 병합하여 사용자에게 제공

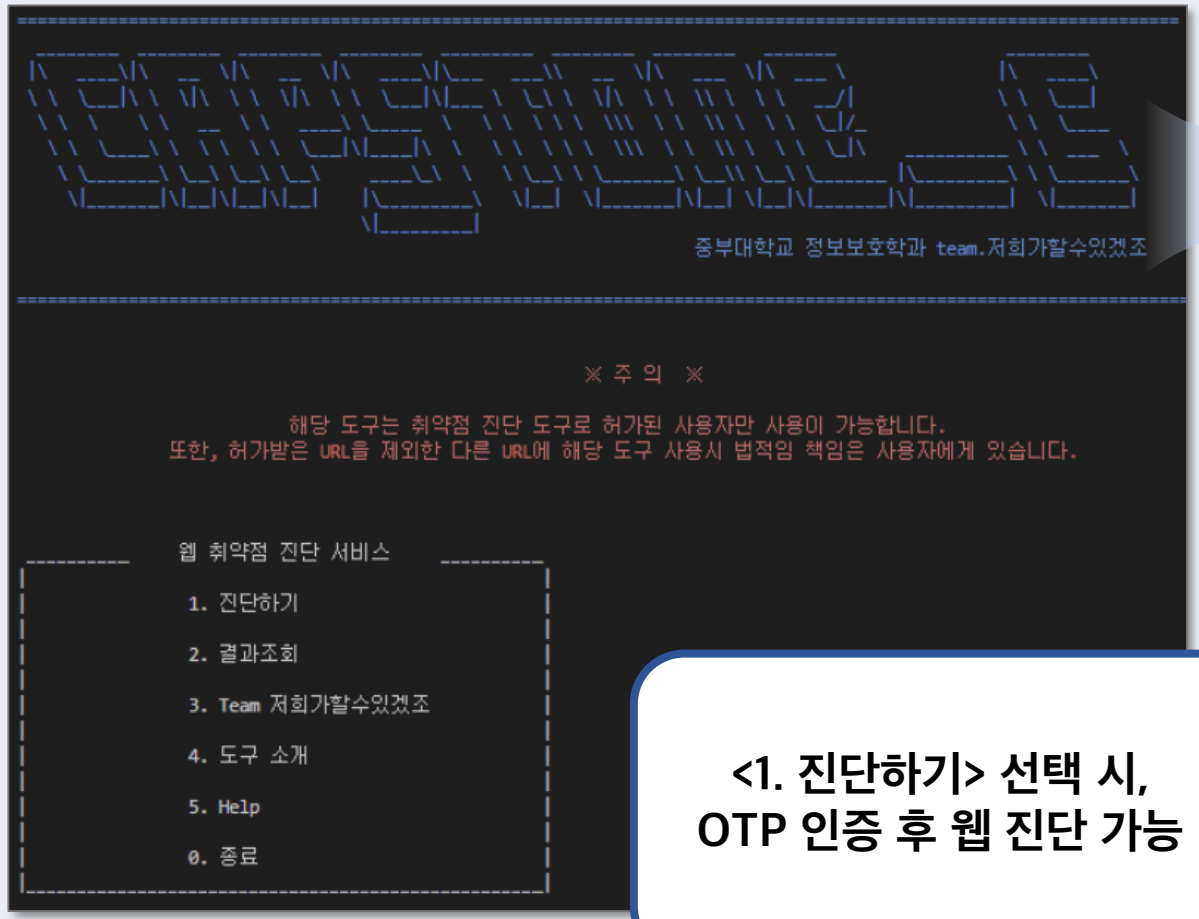
제공 보고서 내용

① 취약점 발견 위치
② 조치방안

03 프로젝트 결과

- 결과 화면
- 시연 영상
- 기대 효과

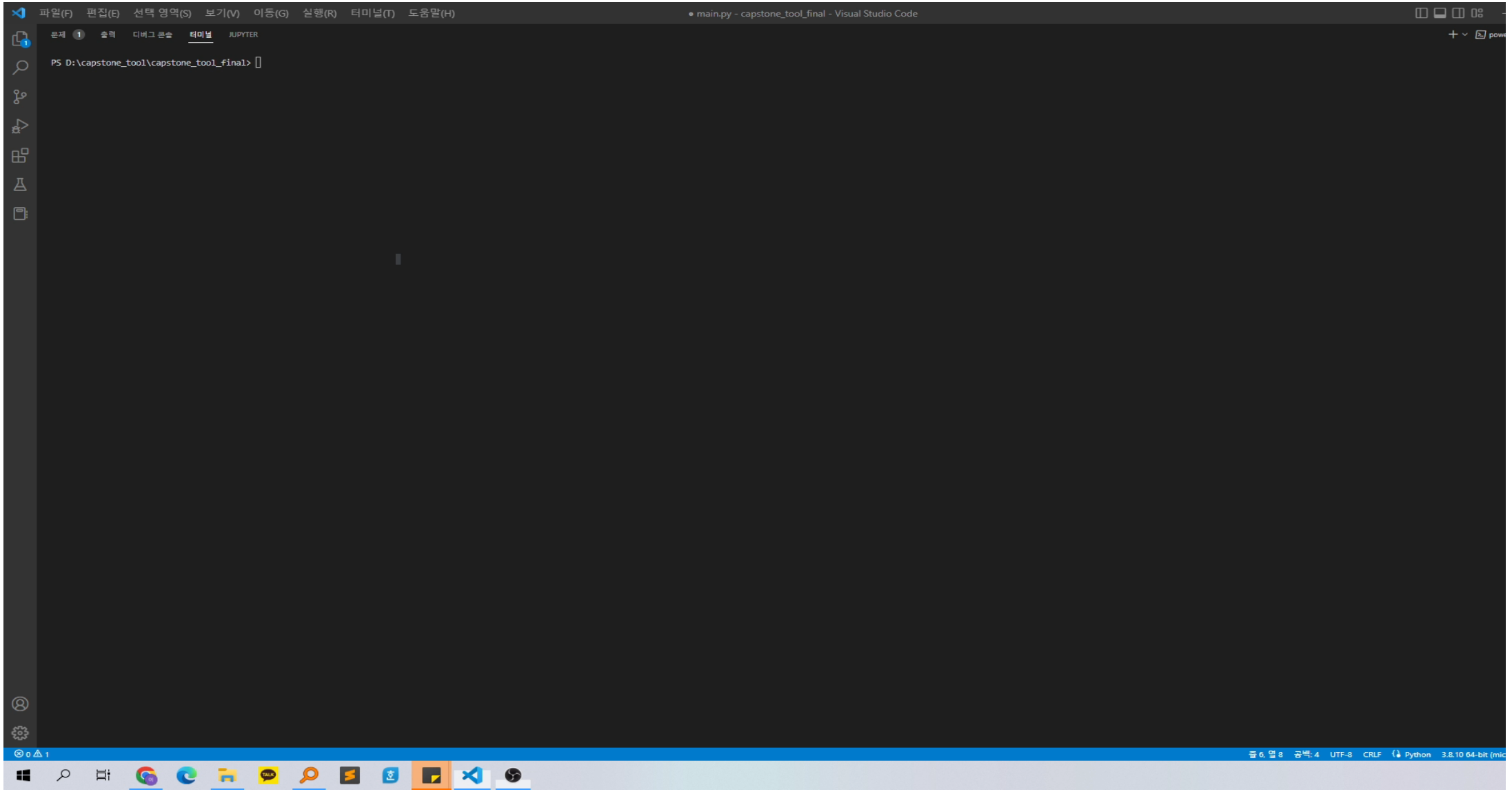
프로젝트 결과 화면



**<1. 진단하기> 선택 시,
OTP 인증 후 웹 진단 가능**

진단 결과 확인 (+조지방안 보고서)

Type	Contents	Resulte
LDAP Injection(3)	This website is "SAFETY" from LDAP Injection	Safety
SQL Injection(5)	This website is "SAFETY" from SQL Injection	Safety
XPath Injection(7)	This website is "SAFETY" from XPath Injection	Safety
XSS(11)	https://capstone6.run.goorm.io/capstone/board.php	Risk
BF(12)	https://capstone6.run.goorm.io/capstone/login.php	Risk
IA(13)	https://capstone6.run.goorm.io/capstone/login.php	Risk
IN(17)	This Website "Risk" from IN	Risk
SF(19)	https://capstone6.run.goorm.io/capstone/login.php	Risk
Auto Attack(20)	https://capstone6.run.goorm.io/capstone/login.php	Risk
PV(21)	http://capstone6.run.goorm.io/capstone/mypage.php	Risk
File Download(23)	This Website is "SAFETY" from File Downloads	Safety
Admin Page Exposure(24)	https://capstone6.run.goorm.io/capstone	Risk


파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)
• main.py - capstone_tool_final - Visual Studio Code
문제 1 출력 디버그 콘솔 터미널 JUPYTER
PS D:\capstone_tool\capstone_tool_final>

프로젝트 기대 효과

취약점 점검 효율성 증대

자동 진단 도구 활용으로 인한
취약점 진단 시간 단축

최신 취약점 유형 연구

식별된 취약점을 통한
주요정보통신기반
공격 유형 연구

빠른 취약점 위치 식별

잠재적 취약점 발생 위치
파악 및 대처

학생들의 보안 역량 강화

모의해킹 웹 사이트를
제작하고 연구함으로써
실습 경험 제공

감사합니다.