Project.

Team: Do And Pray
취약점 원격자동진단사이트

91813200 정보보호학과 최용준
91812103 정보보호학과 김다혁
91812165 정보보호학과 김용훈
91812969 정보보호학과 이진욱
91812713 정보보호학과 원주연
91813286 정보보호학과 허　현

목차 a table of contents

Part 1

프로젝트 설명

Part 1

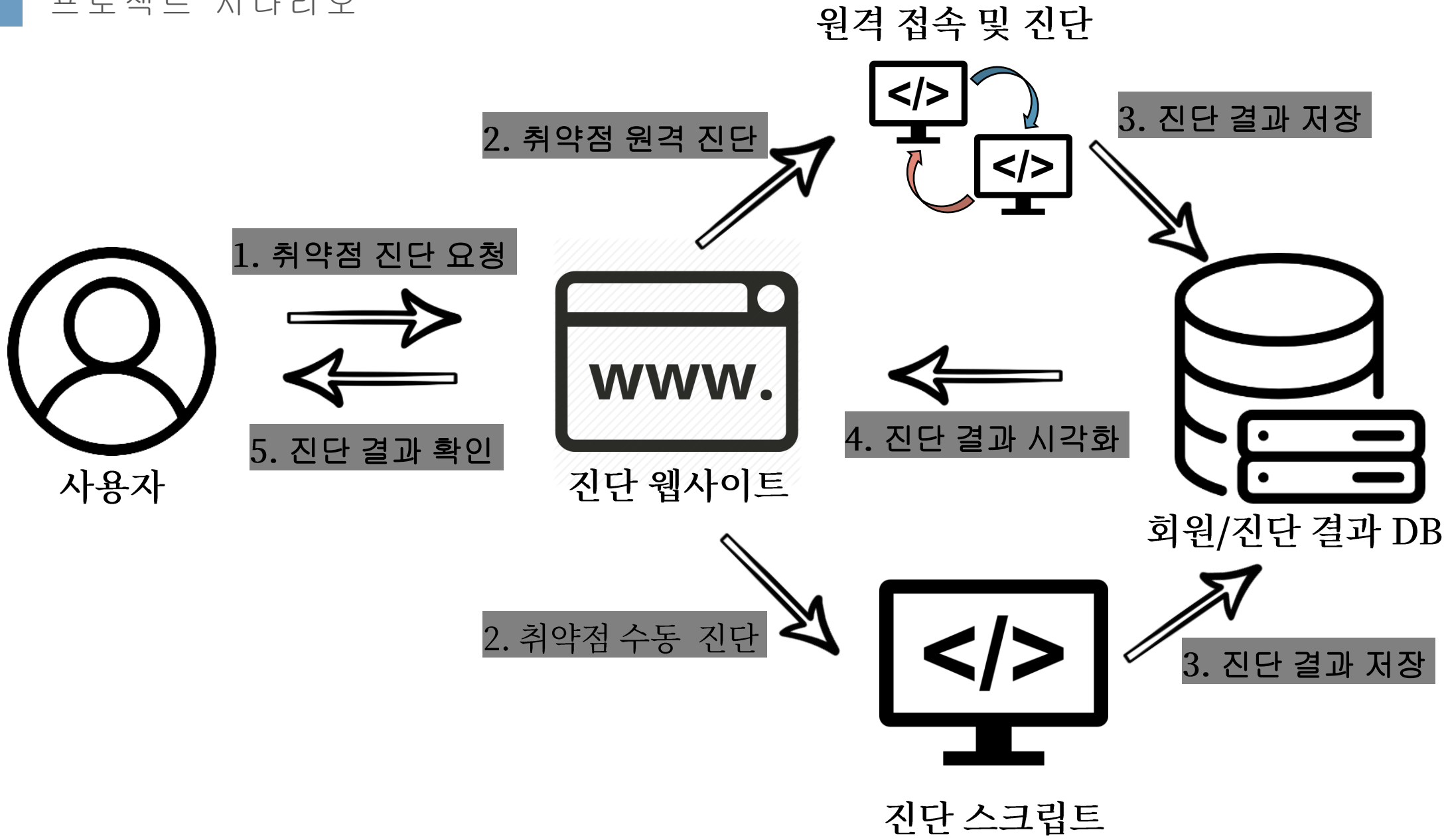
Tech Stack

프론트엔드

백엔드

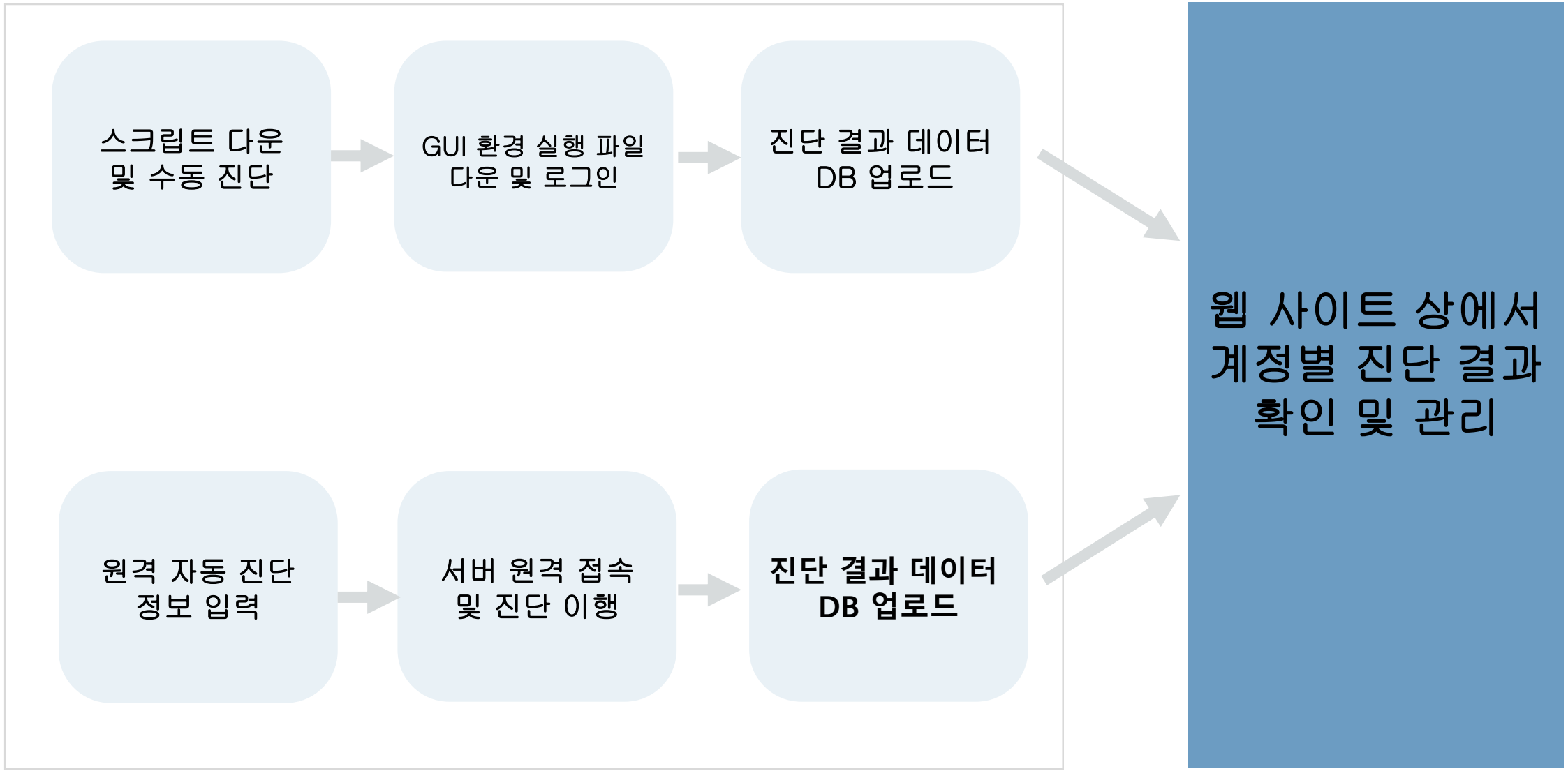
보안

프로젝트 시나리오


원격 접속 및 진단
2. 취약점 원격 진단
3. 진단 결과 저장
1. 취약점 진단 요청
www.
사용자
5. 진단 결과 확인
진단 웹사이트
4. 진단 결과 시각화
회원/진단 결과 DB
2. 취약점 수동 진단
3. 진단 결과 저장
진단 스크립트

Part 1

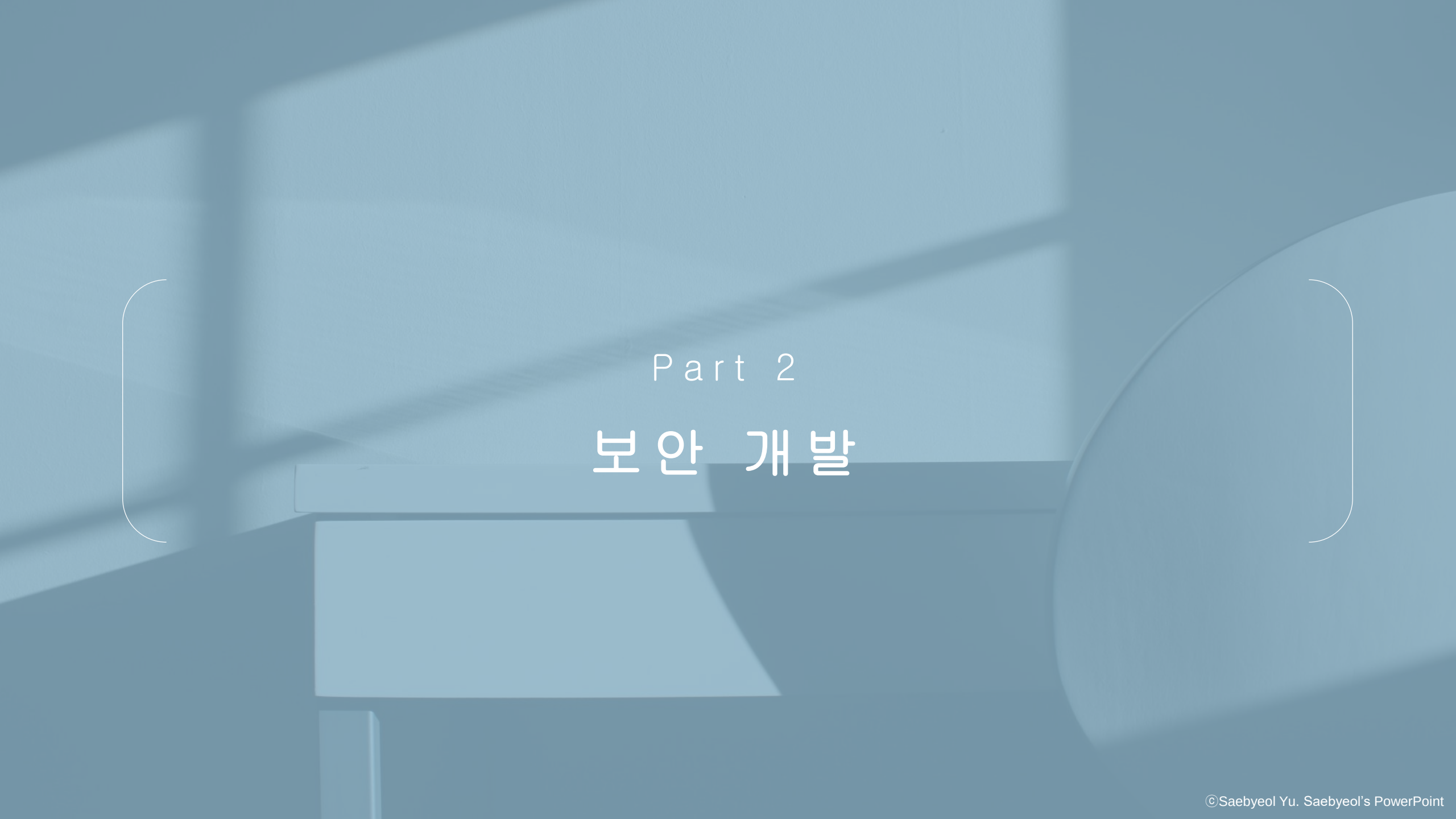
프로젝트 상세 시나리오



진단 근거 자료 및 진단 항목

주요 정보 통신 기반 시설 취약점 진단 가이드

진단 항목 리스트



Part 2

보안 개발

{ 보안 기초 기술 }

Window Server

VMware

CentOS7

{ 보안 기초 기술 }

Batch Script

Shell Script

주요 정보 통신 기반 시설
취약점 진단 가이드

원격 진단 기술

SSH Protocol

SFTP Protocol

문서 자동화

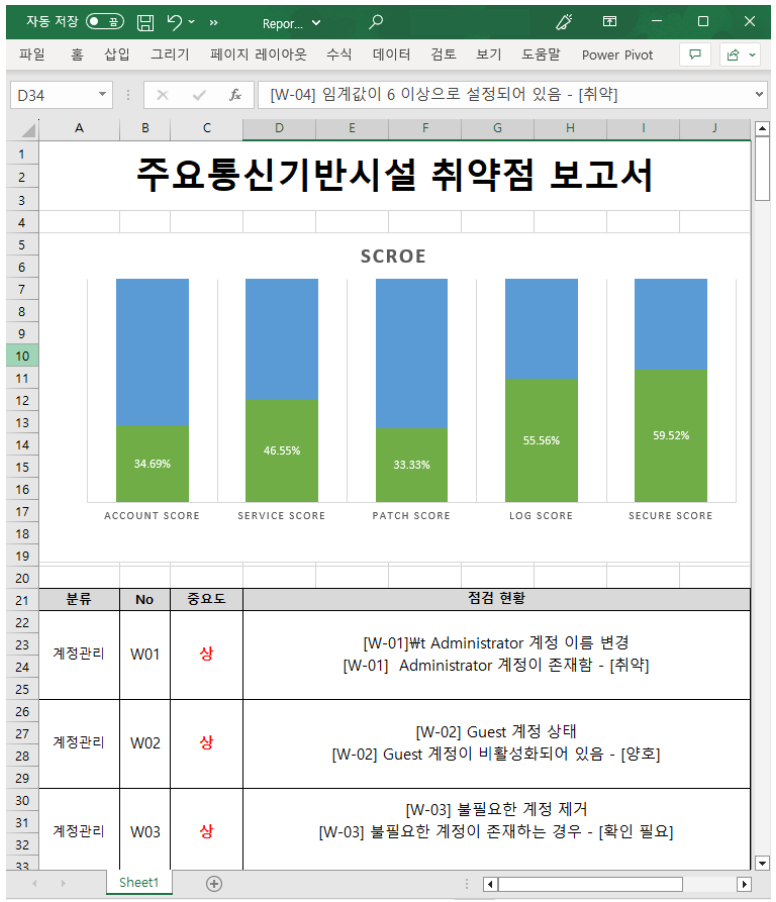
Excel

python

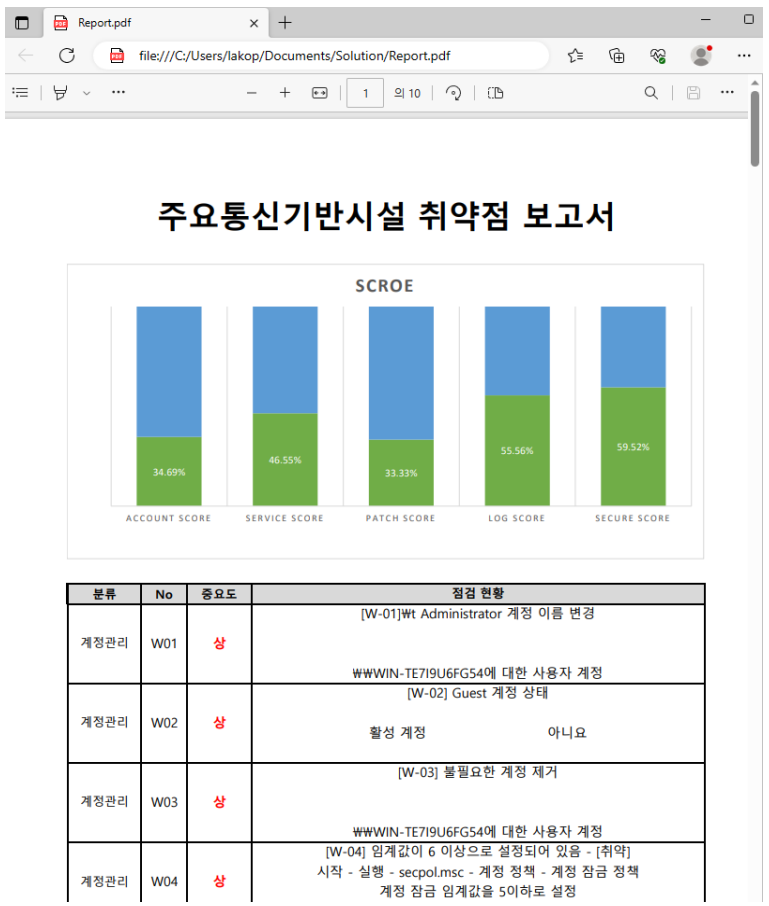
PDF

Part 2

문서 자동화 결과



EXCEL



PDF

Part 2

소스코드

```python
def ssh_execute_script(
    client_IP,
    client_name,
    client_password,
    remote_base_path,
    local_folder_path,
    local_bat_file,
):
    # SSH 세션 시작
    ssh = paramiko.SSHClient()

    # 호스트 메소드 정보 입력
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

    # SSH를 하기 위한 원격 컴퓨터 정보
    ssh.connect(client_IP, username=client_name, password=client_password)

    # SSH를 통해 점검 프로그램이 들어갈 경로를 생성
    create_folder_command = "mkdir C:₩₩informationSS3"
    stdizn, stdout, stderr = ssh.exec_command(create_folder_command)

    # SFTP 세션 시작
    sftp = ssh.open_sftp()

    # SFTP를 통해 로컬에 있던 점검프로그램을 원격 컴퓨터의 특정 경로에다 삽입
    sftp.put(local_bat_file, "C:₩₩informationSS3₩₩₩Windo.bat")

    # 만들어둔 경로를 통해, 그 경로의 점검 프로그램을 원격에서 실행
    command = "cd /d C:₩₩informationSS3 && ₩indo.bat"
    stdin, stdout, stderr = ssh.exec_command(command)
```

SSH & SFTP

```python
ws["E8"].number_format = "0.00%"
ws["E8"].value = LscorePer / 100

ws["F8"].number_format = "0.00%"
ws["F8"].value = SescorePer / 100

# 엑셀의 해당 이름의 차트를 찾아 객체 선언
# 해당 차트 이름은 Chart1
chart = None
for obj in ws._charts:
    if obj.title == "Chart1":
        chart = obj
        break

if chart is not None:
    # 데이터 갱신 부분은 2열8행 부터 6열 8행의 값들로 선언
    data = Reference(ws, min_col=2, min_row=8, max_col=6, max_row=8)
    # 카테고리는 2열7행 부터 6열 7행 부분들의 값들로 선언
    categories = Reference(ws, min_col=2, min_row=7, max_col=6, max_row=7)
    # 정의한 데이터 / 카테고리는 전에 정의한 변수를 참조하여 설정
    chart.set_categories(categories)
    chart.add_data(data)

# 해당 경로로 저장하고 엑셀 종료
wb.save(filet_path2)
wb.close()

# Excel 을 PDF 파일로 전환하기
# 기본적으로 Win32.com은 윈도우만 지원
₩ 호스팅 할 서버는 Renux 기반이므로 포기
"""
excel = win32com.client.Dispatch("Excel.Application")  # 엑셀 어플리케이션 백그라
wb = excel.Workbooks.open(Media_path + r"₩Solution₩Report.xlsx")  # 엑셀 파일을
pdf_path = Media_path + r"₩Solution₩Report.pdf"
wb.ActiveSheet.ExportAsFixedFormat(0, pdf_path)  # 지정했던 경로로 pdf 파일 생성
wb.Close(False)  # 엑셀 작업을 종료시키고 객체를 시스템에 반환
excel.Quit()  # 백그라운드로 켜져있는 엑셀을 종료. 이 문구 없으면 백그라운드로 실
"""
```

문서화

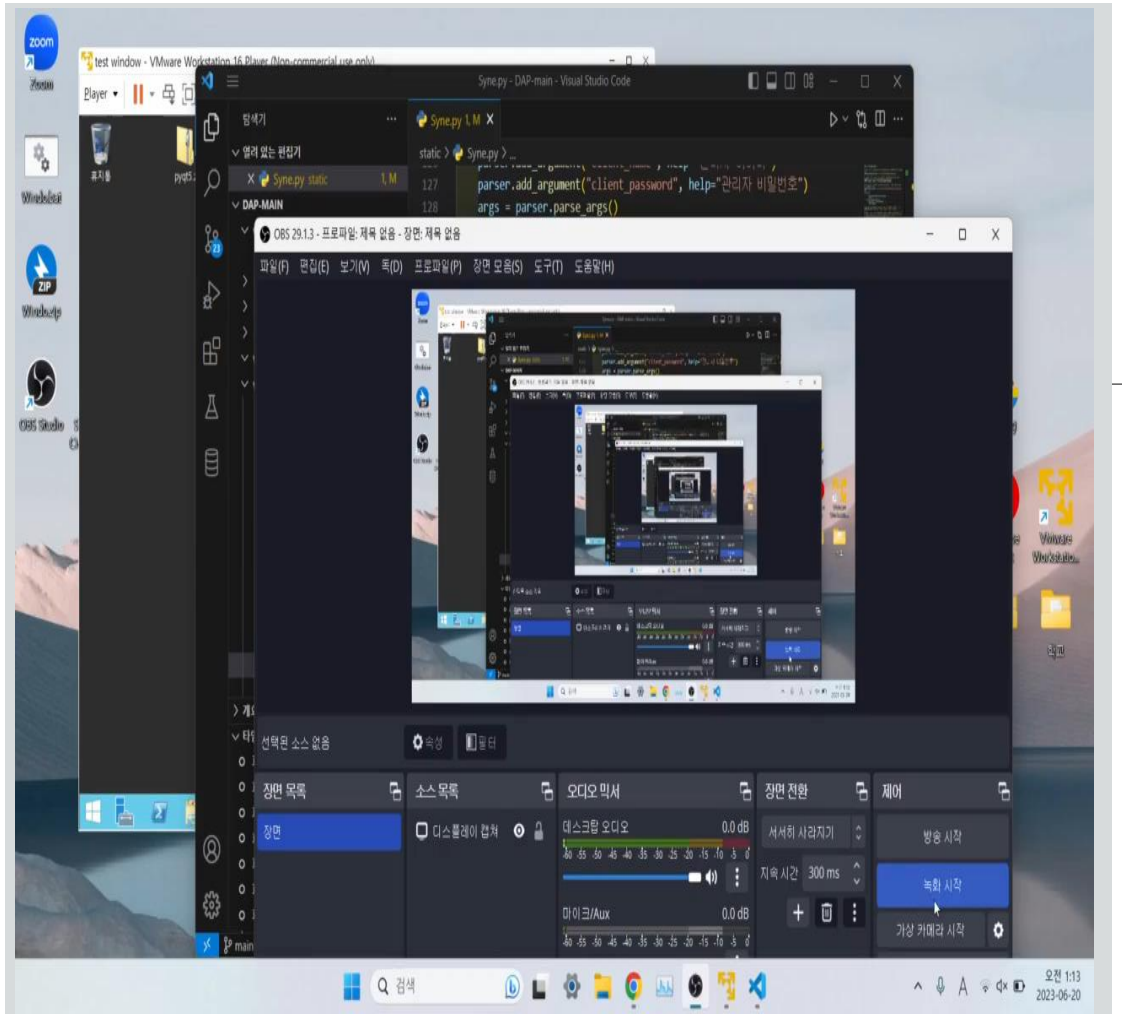
Windo.bat - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
echo [W-72] DoS 공격 방어 레지스트리 설정 >> W1~82₩report.txt
SET/a W72S=0

reg query HKEY_LOCAL_MACHINE₩SYSTEM₩CurrentControlSet₩Services₩Tcpip₩Parameters > dos.txt
reg query HKEY_LOCAL_MACHINE₩SYSTEM₩CurrentControlSet₩Services₩Tcpip₩Parameters > W1~82₩
type dos.txt | findstr /i "SynAttackProtect EnableDeadGWDetect KeepAliveTime NoNameReleaseOnDem

type inform.txt | find /i "SynAttackProtect" | findstr /i "1 2"
if %errorlevel% equ 0 (
        echo [W-72] SynAttackProtect [양호] >> W1~82₩good₩[W-72]good.txt
        echo [W-72] SynAttackProtect [양호] >> W1~82₩report.txt
        SET/a SecureScore = %SecureScore%+3
        SET/a W72S=1
) else (
        echo [W-72] SynAttackProtect [취약] >> W1~82₩bad₩[W-72]bad.txt
        echo [W-72] SynAttackProtect [취약] >> W1~82₩report.txt
        echo [W-72] 시작-실행-REGEDIT입력 >> W1~82₩action₩[W-72]action.txt
        echo HKEY_LOCAL_MACHINE₩SYSTEM₩CurrentControlSet₩Services₩Tcpip₩Parameters 검색
        echo 레지스트리 이름 : SynAttackProtect / 레지스트리 값 종류 : REG_DWORD / 유효 범위 : 0,
        echo 만약 레지스트리가 없으면 추가해주세요 >> W1~82₩action₩[W-72]action.txt

)
```

Batch & Shell Script

자동 진단 결과



자동 진단 시연 영상

SSH를 이용해서 만든 원격 진단은,
호스팅을 진행 한 웹에서 진단을 진행할 공인 IP를
할당 받은 물리적 서버 컴퓨터 부재로 인하여
자동 진단 시연이 불가능 하여, 로컬을 서버로,
VMware의 가상 서버를 클라이언트 서버로 환경을
구축하여 시연하는 영상으로 부득이하게 대체하였음.

Part 3

백엔드 개발

백 엔 드 개 발

개발 언어

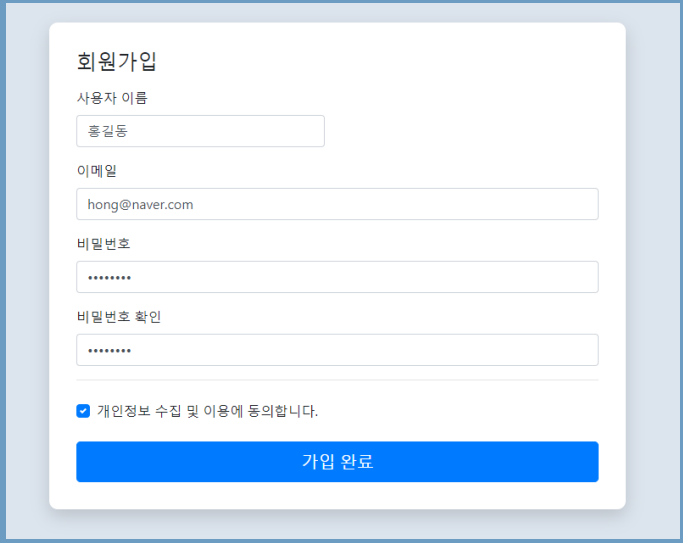
프레임 워크

데이터 베이스

로그인 & 회원가입

```
_id: ObjectId('64908dc21da4e628b0c5823e')
id: 2
password: "pbkdf2_sha256$390000$76GTiFQ29k1HZ8LVhuF9zN$NofCH82XPLX/ptXWlPoaZBztm7…"
last_login: 2023-06-19T17:17:54.438+00:00
is_superuser: false
username: "홍길동"
first_name: ""
last_name: ""
email: "hong@naver.com"
is_staff: false
is_active: true
date_joined: 2023-06-19T17:17:53.808+00:00
```

로그인&회원가입
데이터베이스 저장



게시판 기능

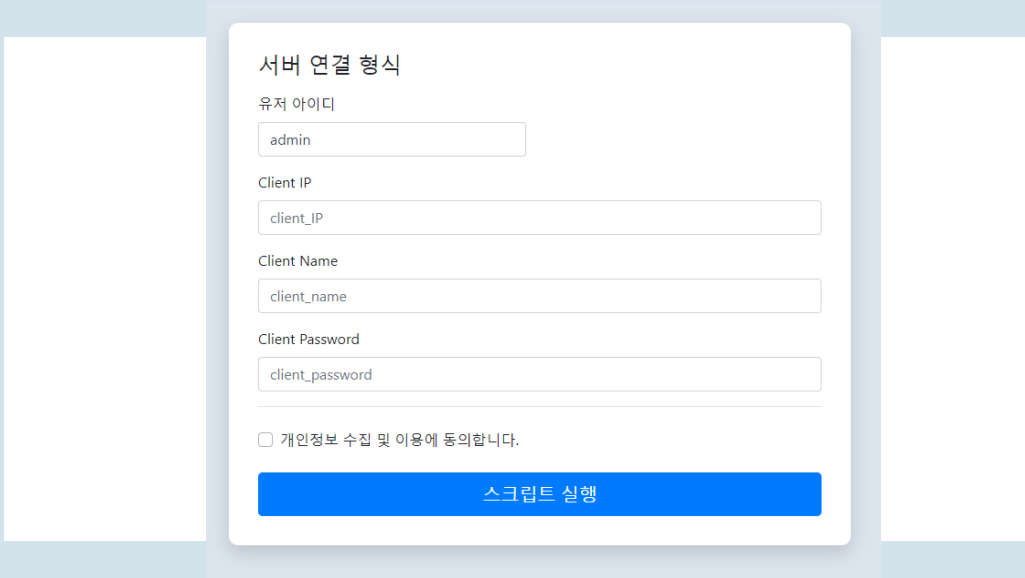

```
_id: ObjectId('64908f481da4e628b0c58244')
id: 2
author_id: 3
subject: "안녕하세요."
content: "안녕하세요. 김다혁 입니다."
modify_date: null
create_date: 2023-06-19T17:24:24.482+00:00
```

```
_id: ObjectId('649090021da4e628b0c5824e')
id: 1
author_id: 6
question_id: 2
content: "안녕하세요."
modify_date: null
create_date: 2023-06-19T17:27:30.251+00:00
```

게시판 등록 및 답글 등록
수정 및 삭제

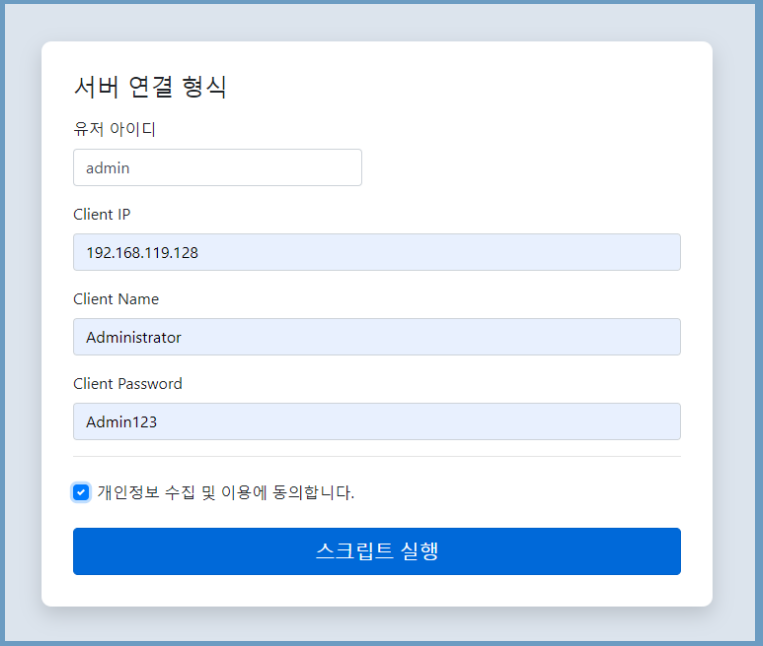
백 엔 드 개 발

수동 진단

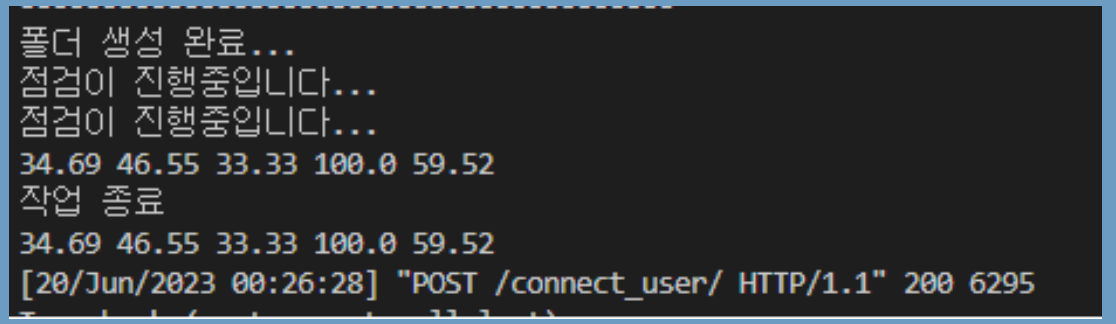


자동 진단

자동 진단 실행

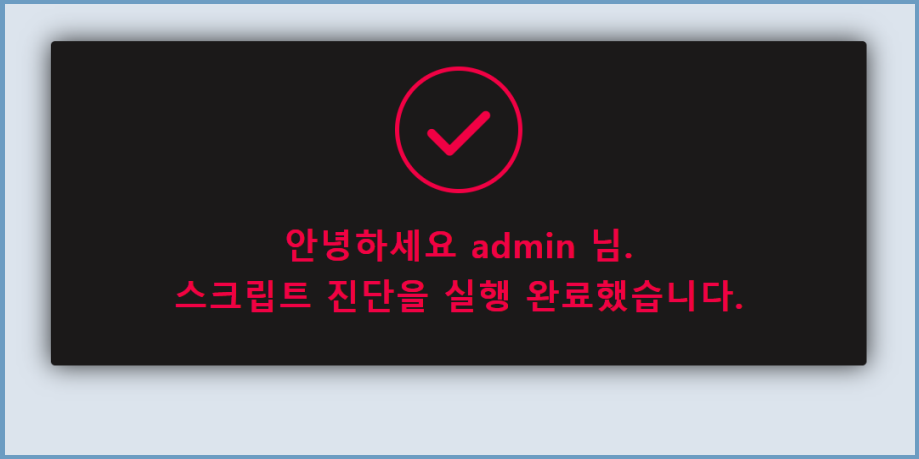


홈페이지 입력 폼



콘솔 작동 화면

자동 진단 실행

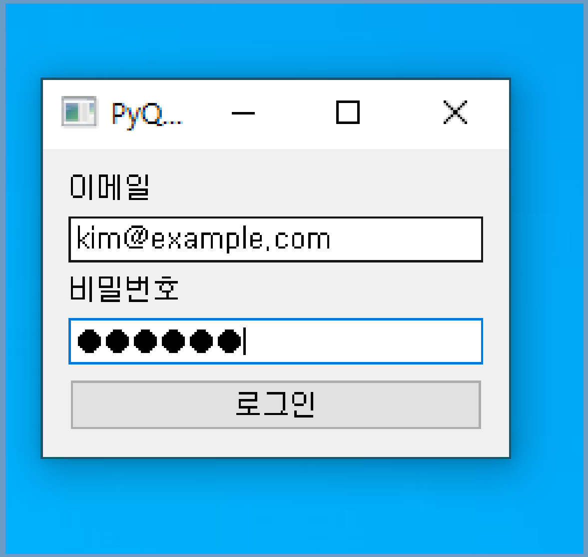



```
_id: ObjectId('649073a43c9fce5ee68638d3')
id: 14
user: "admin"
file: "uploads/admin_2023-06-20_report.txt"
create_date: 2023-06-19T15:26:28.248+00:00
AscorePer: 34.69
SscorePer: 46.55
PscorePer: 33.33
LscorePer: 100.00
SescorePer: 59.52
```

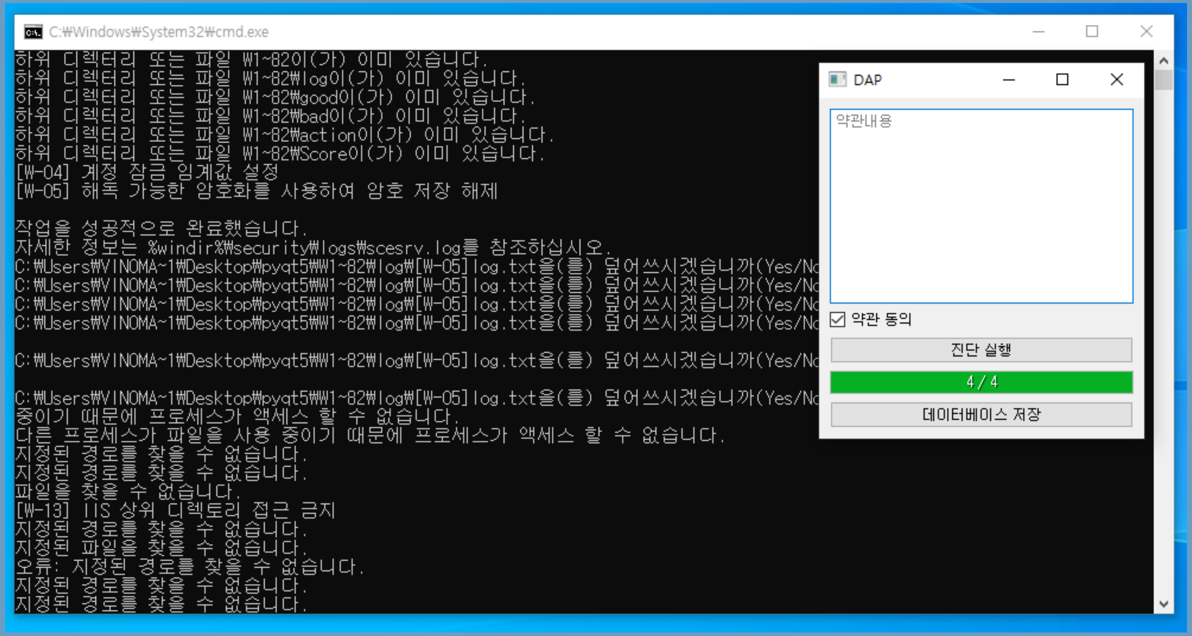

자동 진단 완료시
페이지 전환

자동진단
데이터 저장

수동 진단 결과 관리 GUI

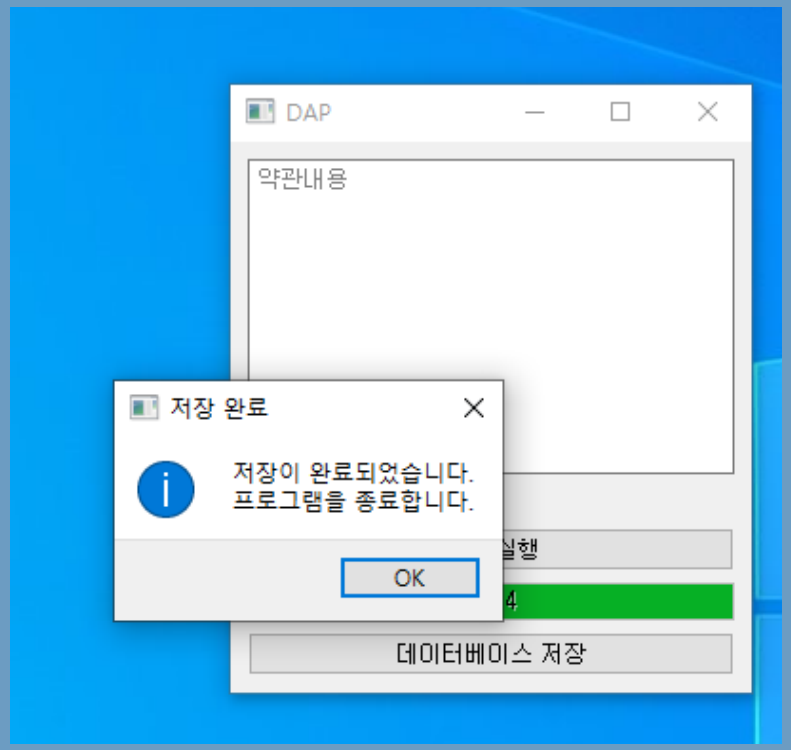


GUI 로그인 기능



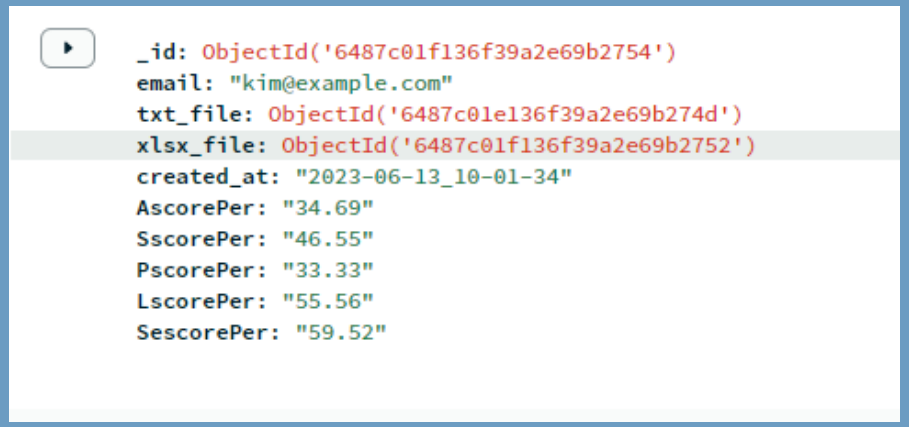
로그인 후
진단 프로그램 실행

수동 진단 결과 관리 GUI



진단 실행 후
데이터 저장 및 종료

수동 진단 결과 관리 GUI



수동 진단 결과
데이터베이스

Report.txt 파일,
Report.xlxs 파일

웹사이트 호스팅

pythonanywhere

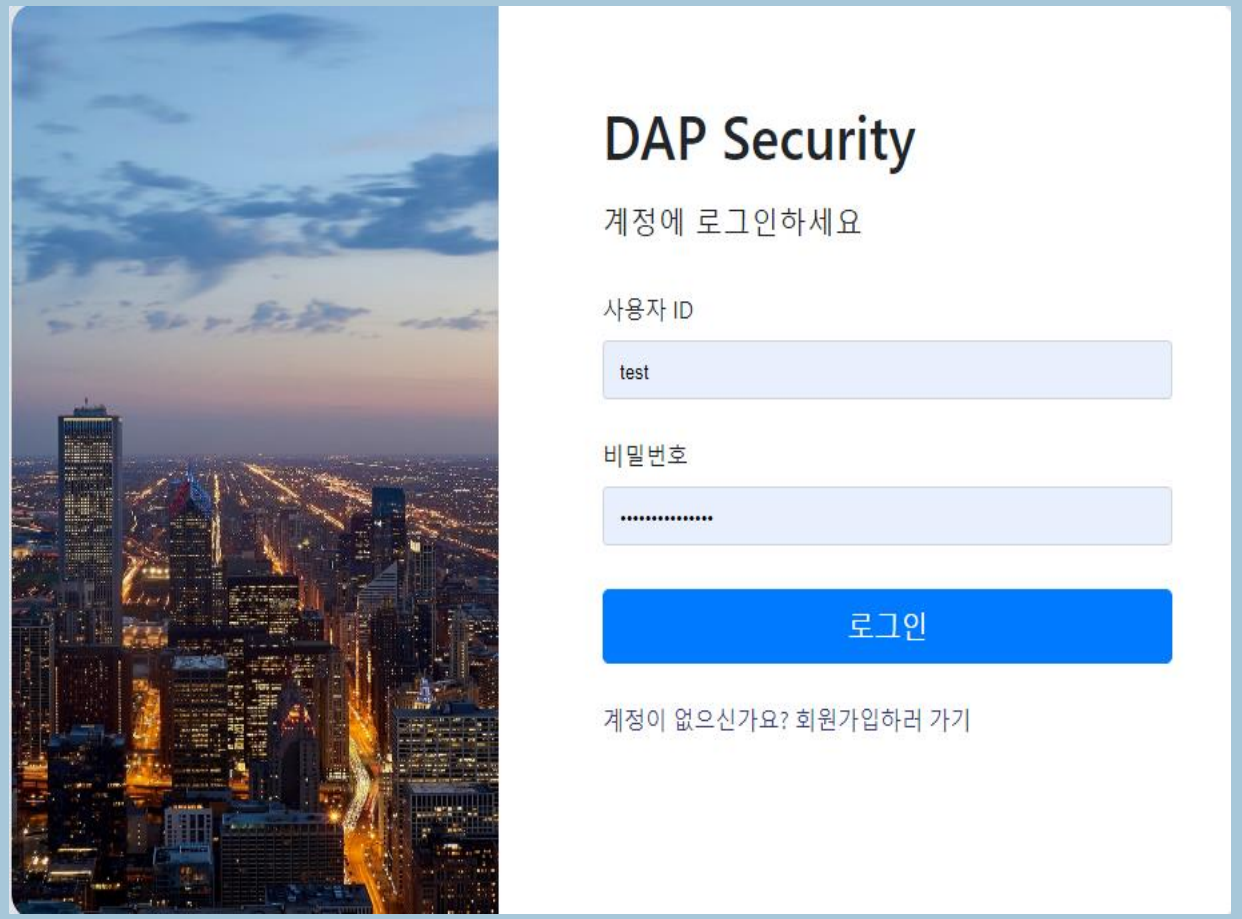
Part 4

프론트 엔드 개발

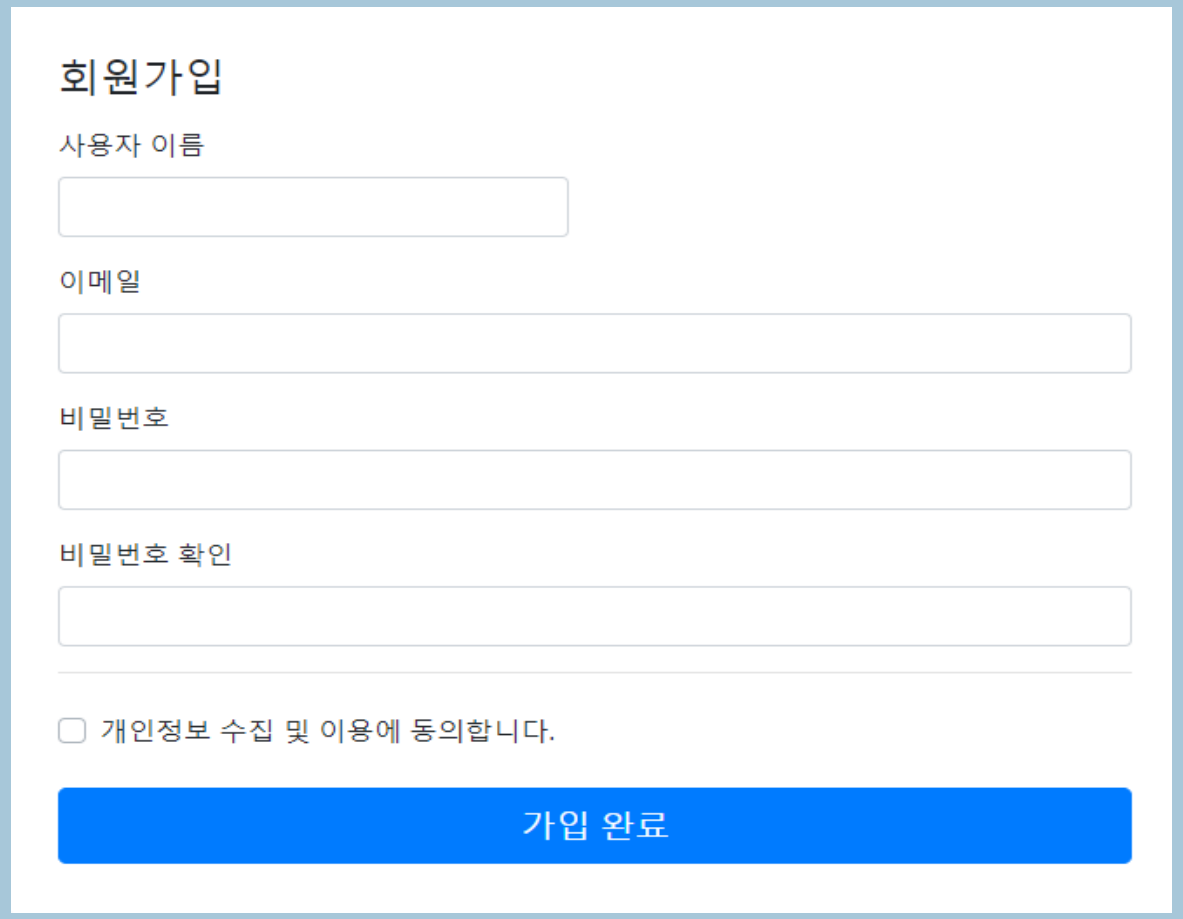
사용 기술

Part 4

로그인 화면



로그인 화면



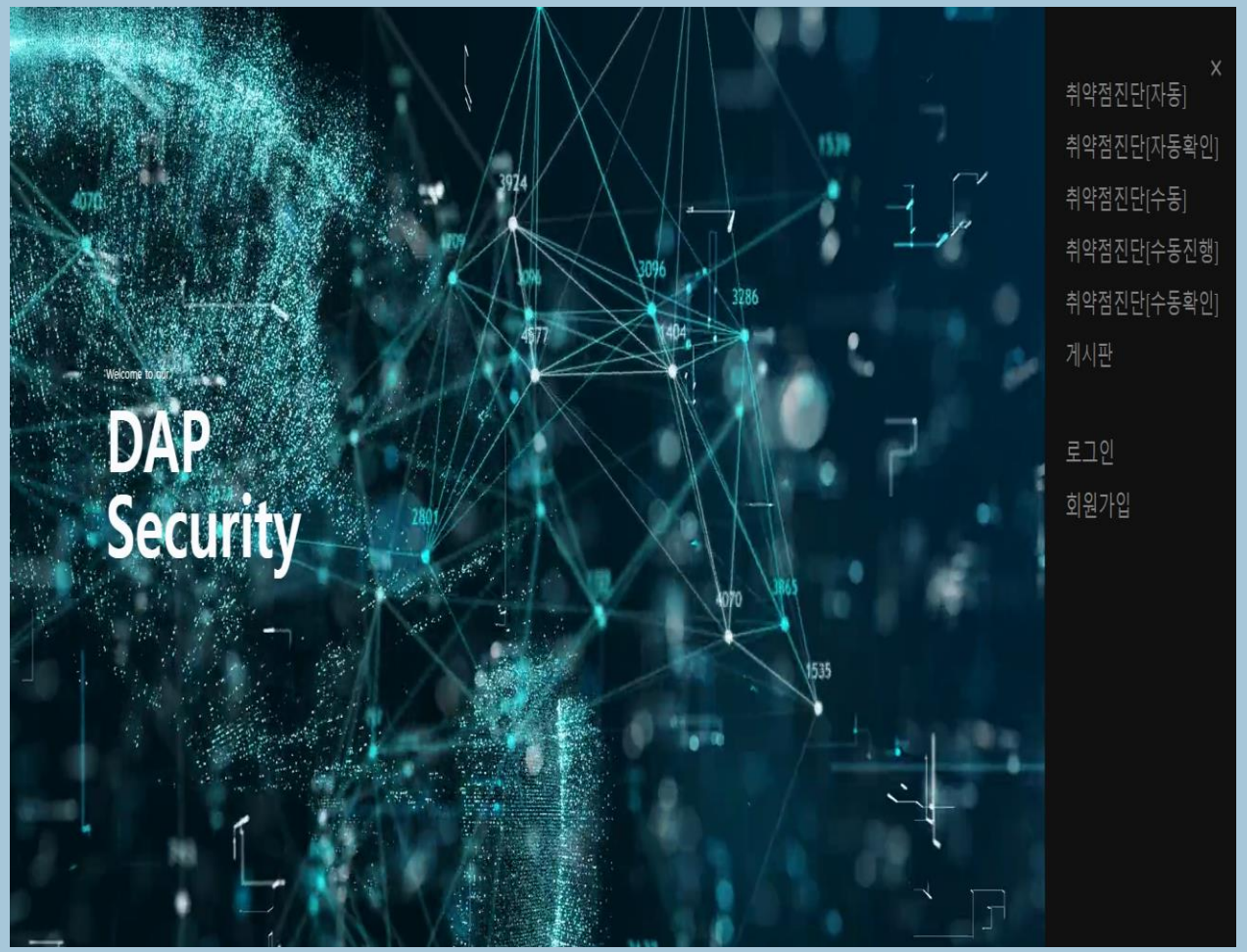
회원가입 화면

Part 4

홈 화면

Part 4

홈 화면(로그인)

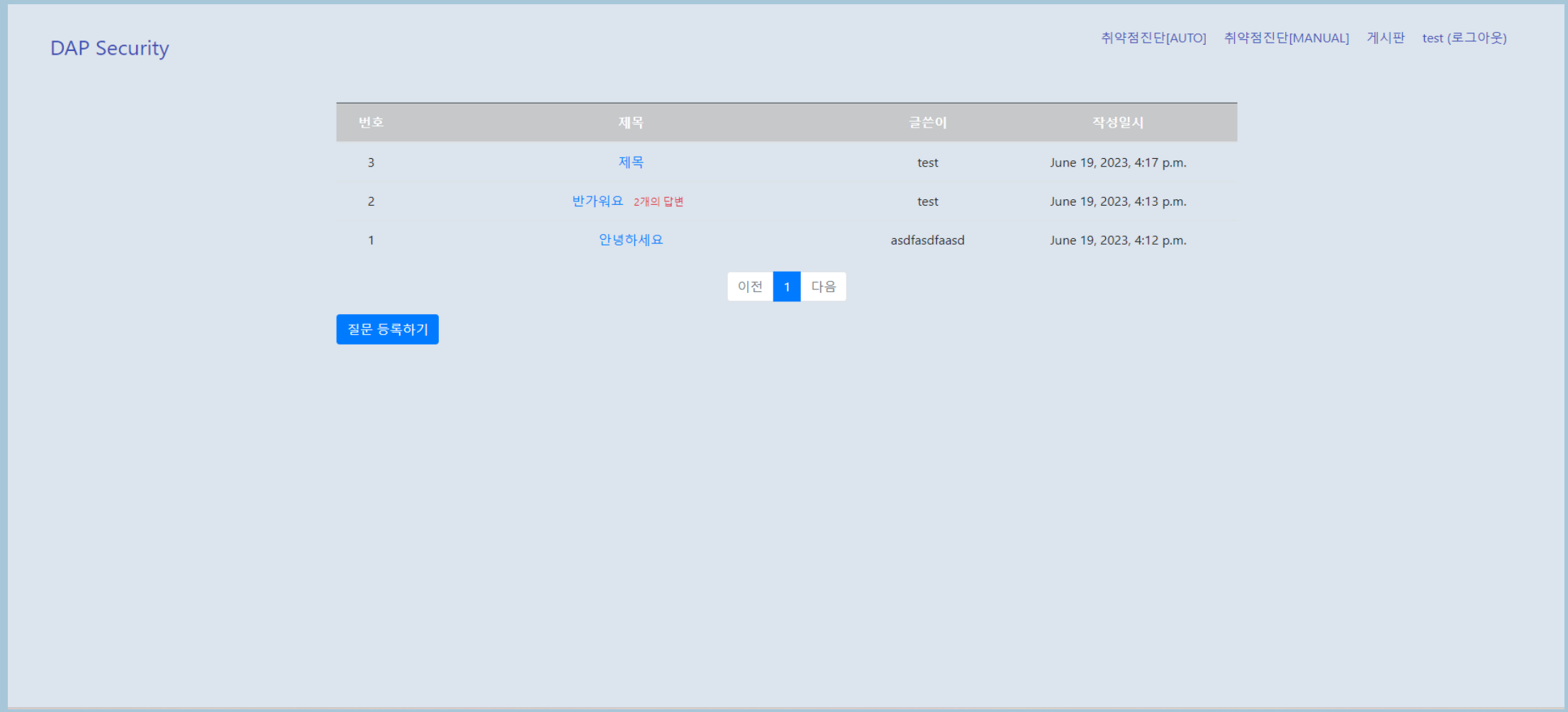


홈 화면 (로그인 전)

홈 화면 (로그인 후)

Part 4

게 시 판



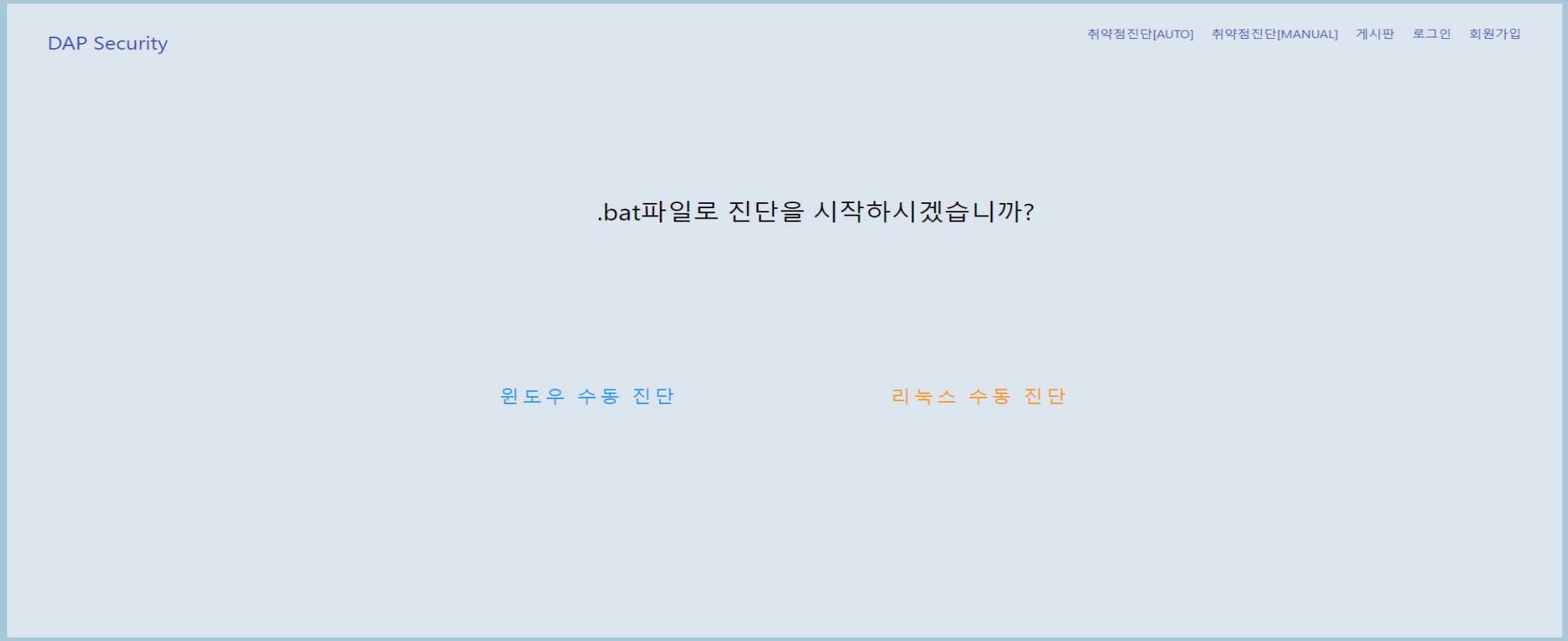
게시판 화면

게 시 판

게시판 상세 화면

Part 4

수동진단



윈도우 수동진행과 리눅스 수동진행을 선택하여 선택한 방식으로 수동진행을 진행가능

수동진단 진행/확인

DAP Security

취약점진단[AUTO] 취약점진단[MANUAL] 게시판 test (로그아웃)

test's Document

업로더	점수 1	점수 2	점수 3	파일	다운로드	업로드 날짜	삭제
test	80	80	80	documents/message_1.txt	Download	June 19, 2023, 4:14 p.m.	Delete

업로더

test

항목1 점수

항목2 점수

항목3 점수

파일 업로드

파일 선택 선택된 파일 없음

Upload

Part 5

개발/협업 과정