

# 서버 취약점 원격 진단 웹 사이트

팀 명 : Do And Pray  
지도교수 : 이병천 교수님  
팀장 : 최용준  
팀원 : 김다혁  
김용훈  
이진욱  
원주연  
허현

2023. 11.

중부대학교 정보보호학과

# 목 차

I. 서론 .....	3
1. 연구 배경 및 필요성 .....	3
2. 연구 주제 선정 이유 .....	3
II. 관련 연구 .....	5
1. Python .....	5
2. Windows Sever 2012 R2 .....	5
3. CentOS7 .....	5
4. Mongo DB .....	5
5. HTML .....	5
6. CSS .....	6
III. 본론 .....	7
1. 구상도 소개 .....	7
2. 취약점 진단 스크립트 제작 .....	7
3. 취약점 진단 웹사이트 제작 .....	9
A. 취약점 자동 진단 .....	11
B. 취약점 수동 진단 .....	12
C. 취약점 진단 결과 .....	13
D. 취약점 진단 실행 및 시각화 .....	14
E. 웹사이트 이용 가이드 .....	17
IV. 결론 .....	22
1. 결론 .....	22
2. 기대효과 .....	22
V. 별첨 .....	23
1. 팀원 소개 .....	23
2. 소스 코드 .....	23
3. 발표 자료 .....	23
4. 소개 자료 .....	24

# I. 서론

## 1. 연구 배경 및 필요성

시간이 지나감에 따라서, 다양한 기술이 등장하고 있는 만큼, 침해사고 신고 횟수 또한 늘어나고 있는 추세다.

[단위 : 건수]

구분 \ 연도	2021년		2022년		2023년
	상반기	하반기	상반기	하반기	상반기
건수	298	342	473	669	664
합계	640		1,142		664

표 1-1 침해사고 신고 현황

( 출처: 한국인터넷진흥원 2023년 상반기 사이버 위협 동향 보고서 )

이를 보면, 아직 상반기임에도 불구하고, 2022년 침해사고 신고 건수의 절반을 넘은 것을 확인할 수 있는데, 서버에 대한 침해사고 신고 현황은 아래와 같다.

[단위 : 건수]

구분 \ 연도	2022 (상반기)		2022 (하반기)		2023 (상반기)	
	건수	비율	건수	비율	건수	비율
침해사고 신고	DDoS 공격	48 (10.1%)	74 (11.1%)	124 (18.7%)		
	악성코드 (랜섬웨어)	125 (26.4%)	222 (30.9%)	156 (20.2%)		
	서버 해킹	275 (58.1%)	310 (46.3%)	320 (48.2%)		
	기타	25 (5.3%)	63 (9.4%)	64 (9.6%)		
	합계	473		669		664

표 1-2 유형별 침해사고 신고 현황

( 출처: 한국인터넷진흥원 2023년 상반기 사이버 위협 동향 보고서 )

이를 보면, '서버 해킹'에 대한 항목이, 침해사고 신고건수의 절반에 가깝게 누적된 것을 확인할 수 있다. 이처럼, 서버에 자산을 특정하여 공격하는 횟수가 증가하고 있는 만큼, 이에 대한 대책은 필요하며, 서버 공격에 대한 위험요소를 전부 제거하지는 못하더라도, 위협 가능성을 최대한 완화하는 작업이 필요하다.

따라서, 주요통신기반시스템 취약점 가이드2017을 기준으로 하여, Windows Server와 Linux의 서버의 취약점을 진단하는 시스템을 만들고자 한다.

## 2. 연구 주제 선정 이유

서버 취약점을 진단하는 프로그램 혹은 서비스는, 직접 고객과 대면하여 수기로 이루어지는 제한성을 극복하고자, 다양한 기술들을 시도하여 왔다.

우리도 이러한 제한성을 극복하고자, 서버의 취약점을 점검하고 이에 대한 리포팅을 제공하는 웹 사이트를 구축하여 서버에 대한 취약점을 원격으로 점검하고 대응방안을 제시하는 시스템을 개발한다.

## II. 관련 연구

### 1. Python

인터 프리터 방식으로 객체 지향적이며, 웹 어플리케이션, 기계 학습 등과 같이 다양한 분야에서 사용되는 프로그래밍 언어로, 라이브러리가 풍부하고 이식성이 좋은 것이 특징이다.

#### 1) Django:

Django는 Python 언어로 작성되었으며 웹 애플리케이션을 효율적으로 빠르게 개발하는데 사용할 수 있는 소프트웨어이다.

#### 2) PyQt :

PyQt란Qt의 레이아웃에Python의 코드를 연결하여GUI 프로그램을 만들 수 있게 도와주는프레임워크를 말한다. UI는 PyQt 프레임워크가 Qt 위젯 및 UI를 구현하게 도와주고, 내부 기능은 Python을 이용한다.

### 2. Windows Sever 2012 R2

Microsoft 가2013년에 출시한Windows Server 2012의 신 버전으로, Windows 8.1을 베이스로 제작된Server OS 이다.

### 3. CentOS7

레드햇 엔터프라이즈 버전에 대한 커스터마이징을 거친 배포판으로, GPL 라이선스를 사용하여 서버쪽으로도 자주 사용되는Linux OS 이다.

### 4. Mongo DB

테이블과 행 대신에 문서를 활용하여 데이터 형식을 처리하고 저장하는, 오픈 소스 비 관계형 데이터베이스 관리 시스템( DBMS ) 이다.

### 5. HTML

HTML(Hyper Text Markup Language)은 가장 단순한 형태의 웹 언어이다. 인터넷 서비스의 하나인 월드 와이드 웹을 통해 볼 수 있는 문서를 만들 때 사용하는 기본적인 웹 언어의 한 종류이다. 특히 하이퍼텍스트를 작성하기 위해 개발되었으며, 인터넷에서 웹을 통해 접근되는 대부분의 웹 페이지들은HTML로 작성된다. HTML은 문서의 글자크기, 글자색, 글자모양, 그래픽, 문서이동(하이퍼링크) 등을 정의하는 명령어로서 홈페이지를 작성하는데 쓰인다. HTML은 전자 문서의 서식을 정의하기 위해 만들어졌으며, 국제표준 SGML의 부분 집합으로 정의되었다. HTML은SGML에서 특히 하이퍼텍스트를 강조하여 만들어진 언어이며, 아스키코드로 된 일반적인 텍스트로 구성되었다. 이 언어는 별도 컴파일러가 필요치 않으며, 웹 브라우저에서 해석이 가능한 사용하기 쉬운 언어로 각광을 받고 있다.

## 6. CSS

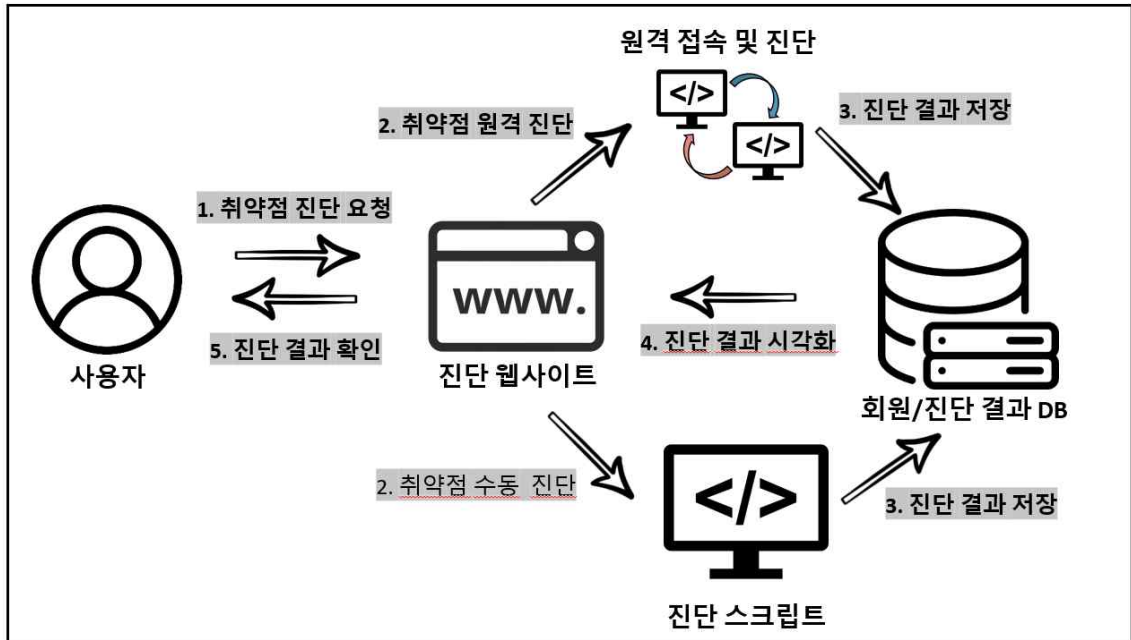
기존의HTML은 웹 문서를 다양하게 설계하고 수시로 변경하는데 많은 제약이 따르는데, 이를 보완하기 위해 만들어진 것이 스타일 시트이고 스타일 시트의 표준안이 바로 CSS이다. 웹 문서의 전반적인 스타일을 미리 저장해 둔 스타일 시트이다. 문서 전체의 일관성을 유지할 수 있고, 세세한 스타일 지정의 필요를 줄어들게 하였다. HTML과 XHTML에 주로 쓰이며, 여러 수준과 프로파일을 가지고 있다. 각 수준의 CSS는 일반적으로 새로운 기능을 담고 있으며CSS1, CSS2, CSS3, CSS4로 나뉜다. 프로파일들은 일반적으로 특정한 장치나 사용자 인터페이스를 위해 만들어진 하나 이상 수준의CSS의 하부 집합이다.

### 1) Bootstrap:

부트스트랩은 웹사이트를 쉽게 만들 수 있게 도와주는CSS, JS 프레임워크이다. 하나의 코드로 휴대폰, 태블릿, 데스크탑까지 다양한 기기에서 작동하게 만들 수 있으며, 다양한 기능을 제공하여 사용자가 쉽게 웹사이트를 제작, 유지, 보수할 수 있도록 도와준다.

### III. 본문

#### 1. 구상도 소개



< 그림3-1 : 시스템 구상도 >

#### 2. 취약점 진단 스크립트 제작

한국인터넷진흥원(KISA) 에서 발간한, 주요통신기반시스템 취약점 가이드의 항목을 토대로 취약점 진단 스크립트를 개발하였다.

분야	세부 버전
1. 유닉스	<ul style="list-style-type: none"> <li>▪ AIX 7.1</li> <li>▪ HP-UX 11i v3</li> <li>▪ SOLARIS 11.2</li> <li>▪ Cent OS 6.6 (Linux)</li> </ul>
2. 윈도우즈	<ul style="list-style-type: none"> <li>▪ Windows Server 2000, NT 5.0</li> <li>▪ Windows Server 2003 Standard SP2 x64</li> <li>▪ Windows Server 2008 Standard R2 SP1 x64</li> <li>▪ Windows Server 2012 Standard R2 x64</li> </ul>

II. 보안가이드라인 ..... 5

UNIX 서버 기본/선택

1. 계정 관리 ..... 11/ 93

2. 파일 및 디렉토리 관리 ..... 24/114

3. 서비스 관리 ..... 45/122

4. 패치 관리 ..... 88

5. 로그 관리 ..... 92/145

부록 ..... 149

윈도우즈 서버 기본/선택

1. 계정 관리 ..... 165/246

2. 서비스 관리 ..... 175/266

3. 패치 관리 ..... 225/287

4. 로그 관리 ..... 227/290

5. 보안 관리 ..... 229/293

< 그림3.2-1 : 서버OS 취약점 분석 평가 항목 >

```

echo [W-01] Administrator 계정 이름 변경 >> W1~82#report.txt

net user > account.txt
net user > W1~82#log#[W-01]log.txt

type account.txt | find /I "Administrator" > NUL
if %errorlevel% EQU 0 (
    echo [W-01] Administrator 계정이 존재함 - [취약] > W1~82#bad#[W-01]bad.txt
    echo [W-01] 시작- 프로그램- 제어판- 관리도구- 로컬 보안 정책 - 로컬 정책 - 보안옵션 >> W1~82#action#[W-01]ac
    echo [W-01] 계정: Administrator 계정 이름 바꾸기를 유추하기 어려운 계정 이름으로 변경 >> W1~82#action#[W-01]
    echo [W-01] Administrator 계정이 존재함 - [취약] >> W1~82#report.txt
) else (
    echo [W-01] Administrator 계정이 존재하지 않음 - [양호] > W1~82#good#[W-01]good.txt
    echo [W-01] Administrator 계정이 존재하지 않음 - [양호] >> W1~82#report.txt
    SET/a AccountScore = %AccountScore%+12
    SET/a AccountScore3 = %AccountScore3%+1
)

del account.txt

echo. >> W1~82#report.txt

echo [W-02] Guest 계정 상태 >> W1~82#report.txt

```

< 그림3.2-2 : Windows Server R2 2012 취약점 진단 스크립트 일부 >

```

#####[U-01]root 계정 원격 접속 제한#####

CF1=/etc/securetty
CF2=/etc/pam.d/login
pts=$(grep 'pts' $CF1 | grep -v '#')
pam=$(grep "/lib/security/pam_securetty.so" $CF2 | grep 'required' | awk '{print $1}')
#사용할 변수 선언
|
if [[ $pam == 'auth' ]] || [[ -z $pts ]]; then
echo -e "[U-01] 원격 터미널 서비스를 사용하지 않거나, 사용 시 root 직접 접속을 차단되어 있음 - [양호]" >> U1~73/good/[U-01]gc
awk '{print substr($0,index($0,$2))}' U1~73/good/[U-01]good.txt >> U1~73/inspect.txt
else
echo -e "[U-01] 원격 터미널 서비스 사용 시 root 직접 접속이 허용되어 있음 - [취약]" >> U1~73/bad/[U-01]bad.txt
awk '{print substr($0,index($0,$2))}' U1~73/bad/[U-01]bad.txt >> U1~73/inspect.txt
echo -e "[U-01] vi 편집기를 사용하여 /etc/securetty 파일을 열어 pts/* 설정이 존재하는 경우 제거 또는 주석처리#nvi 편집기를 사용:
sed -e 's/#[U-01#] /#n#[조치사항#]#n/g' U1~73/action/[U-01]action.txt >> U1~73/action.txt
fi

#####[U-02]패스워드 복잡성 설정#####

echo -e "#n=====
echo -e "=====
echo -e "=====

CF1=/etc/pam.d/system-auth
CF2=/etc/login.defs

lcredit=$(grep "pam_cracklib.so" $CF1 | tr -s ' ' '#n' | grep lcredit | awk -F= '{print $2}')
ucredit=$(grep "pam_cracklib.so" $CF1 | tr -s ' ' '#n' | grep ucredit | awk -F= '{print $2}')
dcredit=$(grep "pam_cracklib.so" $CF1 | tr -s ' ' '#n' | grep dcredit | awk -F= '{print $2}')
ocredit=$(grep "pam_cracklib.so" $CF1 | tr -s ' ' '#n' | grep ocredit | awk -F= '{print $2}')
retry=$(grep "pam_cracklib.so" $CF1 | tr -s ' ' '#n' | grep retry | awk -F= '{print $2}')
minlen=$(grep "pam_cracklib.so" $CF1 | tr -s ' ' '#n' | grep minlen | awk -F= '{print $2}')

```

< 그림3.2-3 : CentOS7 취약점 진단 스크립트 일부 >



### 3. 취약점 진단 웹사이트 제작

웹 사이트를 제작하여, 사용자가 웹 상에서 취약점 진단이 가능하도록 구성하였다.

웹 사이트 주소: <https://dydgnslra.pythonanywhere.com/>



< 그림3.3-1 : 취약점 진단 메인 홈페이지 >



< 그림3.3-2 : 취약점 진단 사이트 로그인 포맷 >

## 회원가입

사용자 이름

이메일

비밀번호

비밀번호 확인

개인정보 수집 및 이용에 동의합니다.

가입 완료

< 그림3.3-3 : 취약점 진단 사이트 회원가입 포맷 >

번호	제목	글쓴이	작성일자
5	웹사이트가 열려요.	이진욱	June 20, 2023, 2:27 a.m.
4	리눅스 환경에서는 어떻게 실행되는 건가요?	김용훈	June 20, 2023, 2:26 a.m.
3	자동진단에 대해서 궁금합니다.	최용준	June 20, 2023, 2:25 a.m.
2	안녕하세요. 1개의 답변	김다혁	June 20, 2023, 2:24 a.m.
1	질문이 있습니다.	홍길동	June 20, 2023, 2:22 a.m.

이전 1 다음

질문 등록하기

< 그림3.3-4 : 취약점 진단 사이트 게시판 화면 >

### A. 취약점 자동 진단

웹 사이트내에서 사용자의 정보를 입력받아, 자동적으로 사용자의 서버를 진단하고 시각화 및 문서화 까지 진행하도록 제작되었다.

**서버 연결 형식**

유저 아이디

Client IP

Client Name

Client Password

개인정보 수집 및 이용에 동의합니다.

**스크립트 실행**

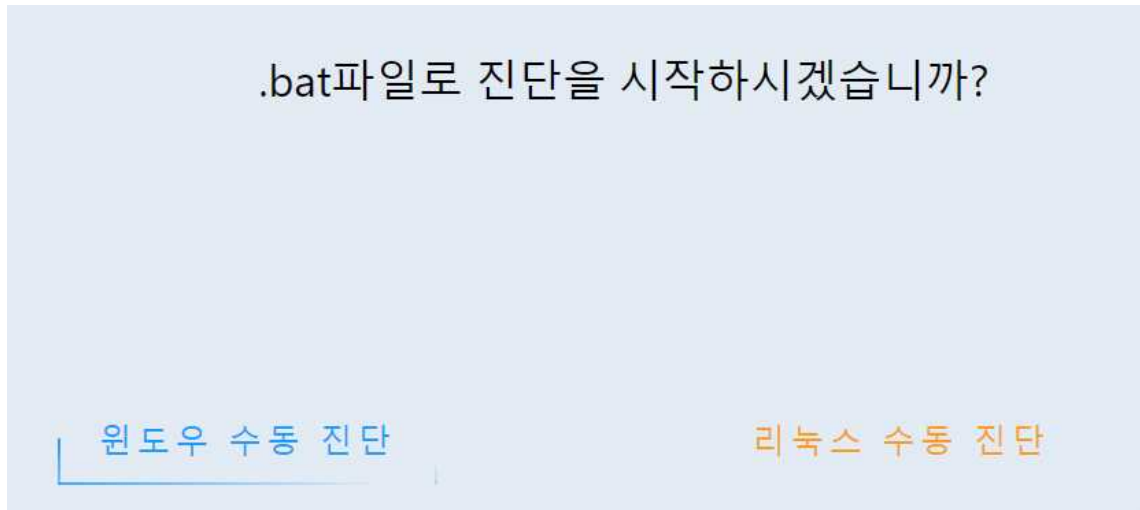
< 그림3.3.1-1 : 취약점 자동 진단 입력 포맷 >

```
_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  
SscorePer: 59.52
```

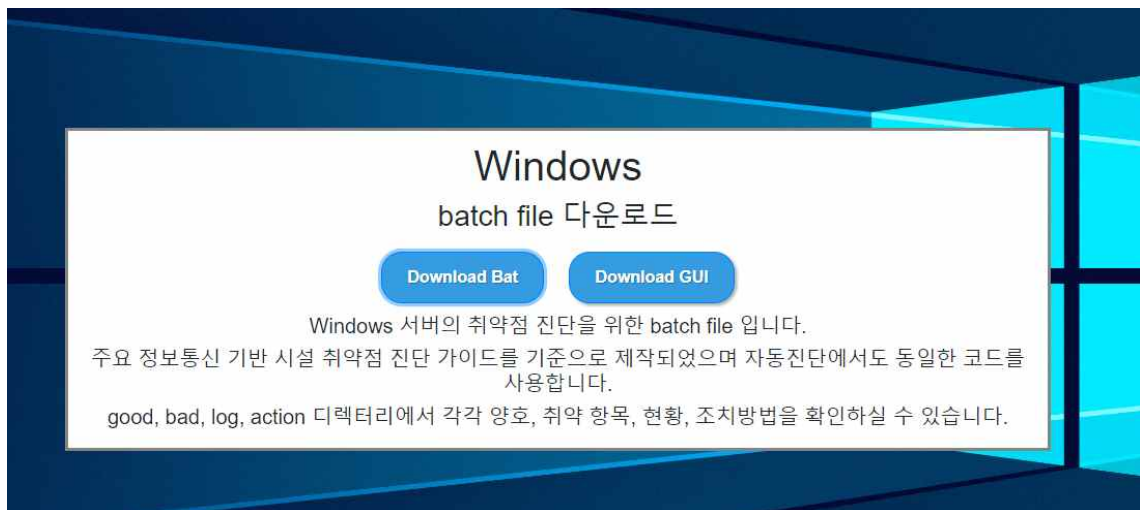
< 그림3.3.1-2 : 취약점 자동 진단 데이터 저장 >

## B. 취약점 수동 진단

사용자 진단에 필요한 정보를 제공하는 것을 거부하거나, 해당 서버의 설정이 원격 접속을 거부할 경우의 상황의 필요성에 따라 제작되었다.

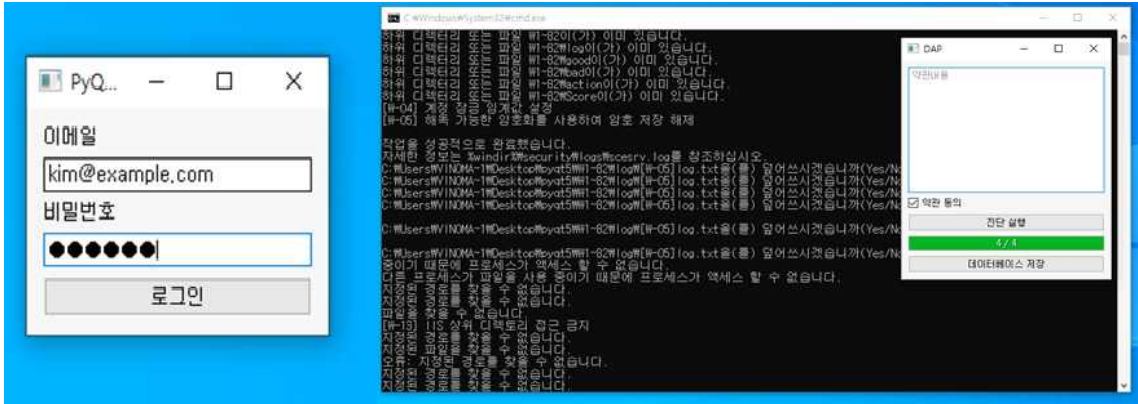


< 그림3.3.2-1 : 취약점 수동 진단 화면 >



< 그림3.3.2-2 : 취약점 수동 진단- [ 윈도우 수동 진단] 화면 >

사용자는 직접 스크립트를 다운 받아서, 해당 프로그램을 실행할 수 있고, GUI를 다운 받아, GUI를 실행하여 웹페이지와DB에 연동시킬 수 있다.



< 그림3.3.2-2 : 취약점 수동 진단- GUI 진단 화면 >



< 그림3.3.2-2 : 취약점 수동 진단- GUI 진단 데이터 저장 >

### C. 취약점 진단 결과

사용자가 자동 혹은 수동 진단으로 진행했을 시, 웹에 출력함과 동시에, 결과 파일 PDF 혹은 Excel 파일 형태로 다운로드 받을 수 있도록 제작되었다.



< 그림3.3.3 : 진단 데이터 출력 >

## D. 취약점 진단 실행 및 시각화

웹 사이트에서 사용자의 서버를 점검하는 방법과 시각화 하는 작업은 모두 Python으로 작성 되었다.

```
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"
    stdin, stdout, stderr = ssh.exec_command(create_folder_command)

    # SFTP 세션 시작
    sftp = ssh.open_sftp()

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

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

    # 원격 점검에서 생성되는 SeScore3.txt 가 있는지 확인. 없으면 생길 때 까지 5초동안 대기
    remote_file_to_check = "C:##informationSS3##1~82##Score##SeScore3.txt"
    while True:
        try:
            sftp.stat(remote_file_to_check)
            break
        except FileNotFoundError:
            print("점검이 진행중입니다...")
            time.sleep(5)
```

< 그림3.3.4-1 : SSH & SFTP 스크립트 일부 >

```
client_password = args.client_password
remote_base_path = "C:##informationSS3##1~82" # 원격 컴퓨터 경로
local_folder_path = Media_path + "##txt" # 원격 컴퓨터에서 복사할 파일을 옮길 로컬 경로
local_bat_file = "static##windo.bat" # 원격 컴퓨터에 넣을 점검 파일 경로

ssh_execute_script(
    client_IP,
    client_name,
    client_password,
    remote_base_path,
    local_folder_path,
    local_bat_file,
)

# 원격에서 가져온 파일들 중에 점수가 적혀있는 텍스트 정보
# 각각 오픈해서 읽어서 객체선언

with open(local_folder_path + r"##Score##AScore.txt", "r", encoding="ANSI") as f:
    Ascore = int(f.read())
    AscorePer = round((Ascore / 147) * 100, 2)

with open(local_folder_path + r"##Score##SScore.txt", "r", encoding="ANSI") as f:
    Sscore = int(f.read())
    SscorePer = round((Sscore / 348) * 100, 2)

with open(local_folder_path + r"##Score##PScore.txt", "r", encoding="ANSI") as f:
    Pscore = int(f.read())
    PscorePer = round((Pscore / 9) * 100, 2)

with open(local_folder_path + r"##Score##LScore.txt", "r", encoding="ANSI") as f:
    Lscore = int(f.read())
    LscorePer = round((Lscore / 27) * 100, 2)

with open(local_folder_path + r"##Score##SeScore.txt", "r", encoding="ANSI") as f:
    Sscore = int(f.read())
    SscorePer = round((Sscore / 168) * 100, 2)
```

< 그림3.3.4-2 : 점수 통계 스크립트 일부 >



```

# report 파일의 내용을 읽어드림
with open(file_path, "r") as file:
    text = file.read()

# [W-?] ?는 1부터 시작하므로 헛갈리지 않게 1부터 시작
for i in range(1, variable_count + 1):
    # 시작 인덱스는 읽어드린 내용을 기반으로 [W-i]값부터 시작
    # 여기서 i값의 형태는 01,02...08 같은 형태이므로 zfill(2)를 사용하여 문자열 왼쪽에 100이만의 숫자에 1
    # 끝의 인덱스는 i+1을 기준으로 함
    start_index = text.find(f"{start_marker}{str(i).zfill(2)}")
    end_index = text.find(f"{start_marker}{str(i+1).zfill(2)}")
    # 만약 끝의 인덱스 값이 위와 같은 규칙으로 할당되지 못한경우
    # 값이 하나라고 간주하고 시작 값으로 반환
    if end_index == -1:
        extracted_text = text[start_index:]
    else:
        # 끝의 인덱스 값이 존재한다면, 해당 변수의 값에 시작 인덱스부터 끝 문자열 전까지의 값이 할당
        extracted_text = text[start_index:end_index]
    # 발아온 변수 값을 공백을 제거한 후에,
    # 띄어쓰기 형태로 저장하는데, 해당 발류의 이름은 W{i} 형태로 저장
    # [W-01] 의 value 값 이름 : W1 이런식으로 저장
    variable_dict[f"W{i}"] = extracted_text.strip()

# 엑셀에 사용할 템플릿과 결과물로 나올 엑셀의 경로 선언
file_path1 = "static#Template.xlsx"
file_path2 = Media_path + r"##Solution##Report.xlsx"

# 시작셀 지정 및 엑셀의 오프셋 값 선언
# column의 오프셋 값은 셀의 열의 위치, row 오프셋은 행의 위치를 담당 (가로 / 세로)
# 이 오프셋값들은 해당 셀의 작업이 끝나고 다음셀로 넘어 갈 때, 이 값들을 참조하여 나아간다
# 지금 0 / 4 이므로 D22에서 시작하면 D(+0)22(+4) 해서 다음값은 D26이 된다.
# 단, 이 설정은 내가 하기 편하도록 설정한 것이므로 참고
start_cell = "D22"
column_offset = 0
row_offset = 4

# 해당 경로의 엑셀을 로드하여 활성화 한다.

```

< 그림 3.4-3 : 엑셀 전환 스크립트 일부>

```

# 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(file_path2)
wb.close()

excel = win32com.client.Dispatch("Excel.Application") # 엑셀 어플리케이션 백그라운드로 실행
wb = excel.Workbooks.open(Media_path + r"##Solution##Report.xlsx") # 엑셀 파일을 읽어드려서 객체로 지정
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 실행되기 때문에 작업관리;

print("작업 종료")

```

< 그림3.4-4 : 엑셀을PDF로 전환하는 스크립트 일부>



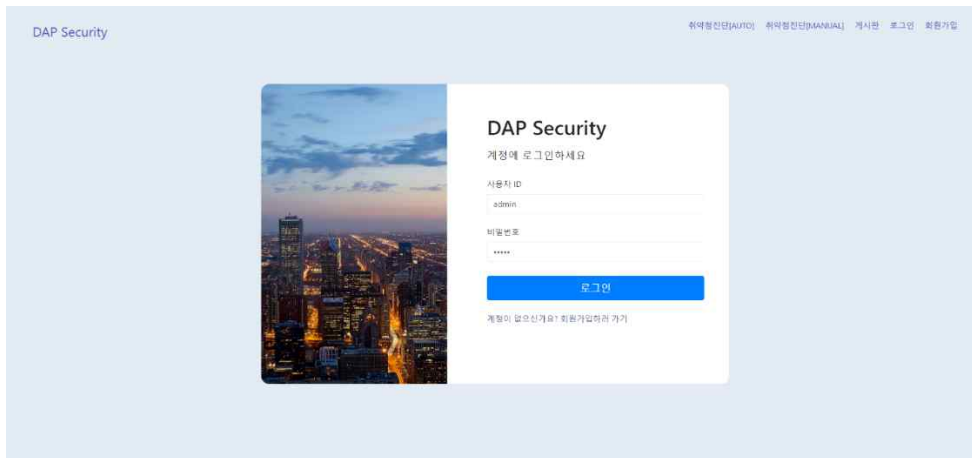


## E. 웹사이트 이용 가이드

### ① 자동 진단



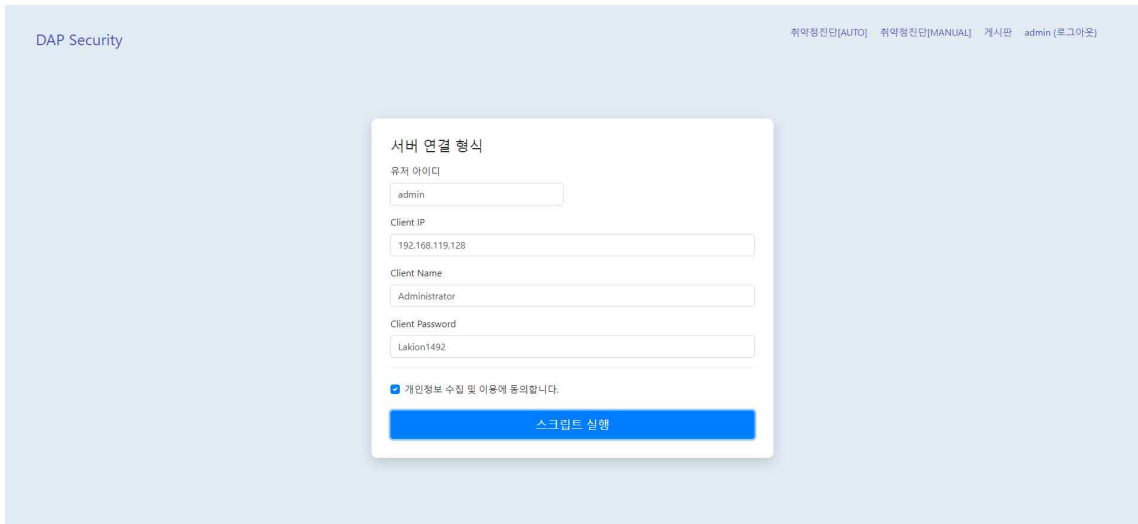
< 그림3.5.1-1 : 메인 화면 >



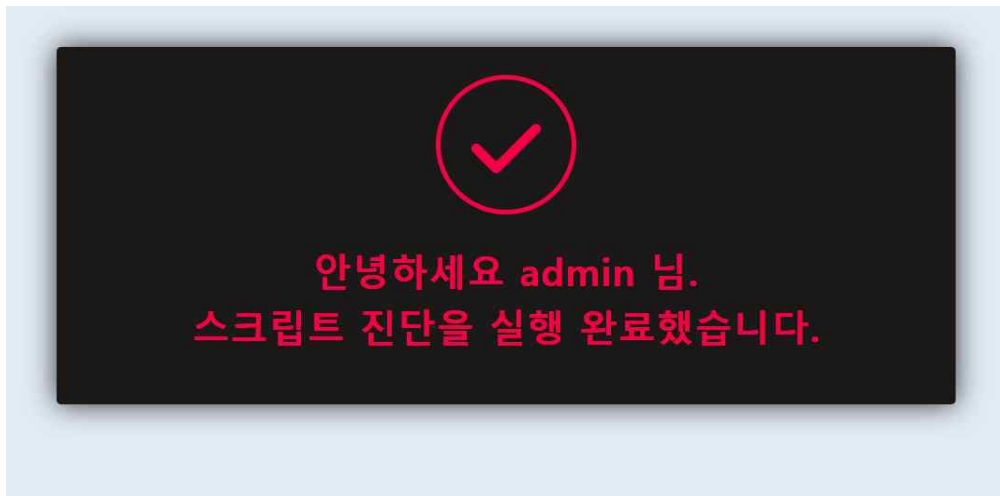
< 그림3.5.1-2 : 로그인 페이지에서 로그인 >



< 그림3.5.1-3 : 취약점 진단 사이트 로그인 후의 포맷 >



< 그림3.5.1-2 : 자동 취약점 진단 >



< 그림3.5.1-3 : 취약점 진단 성공화면 >

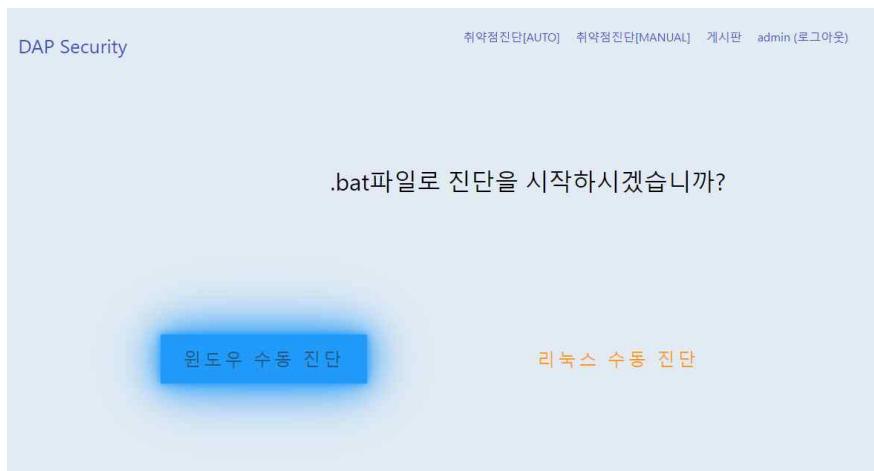


< 그림3.5.1-4 : 취약점진단 결과확인 페이지 >

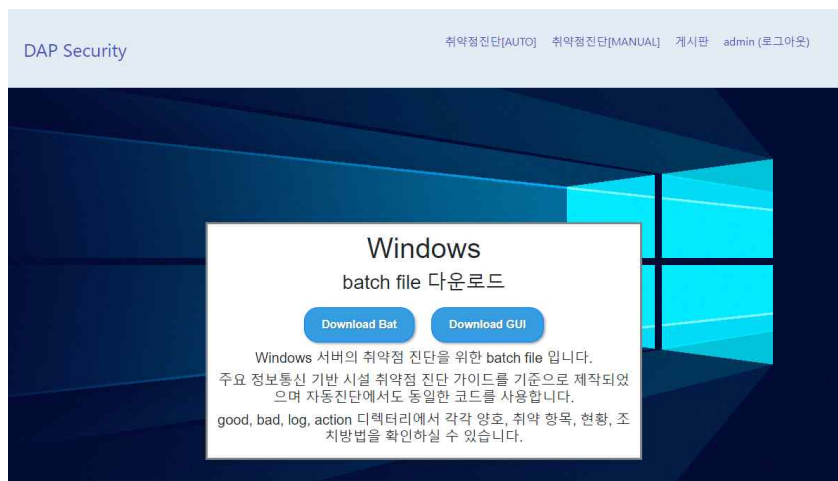
②) 수동 진단



< 그림3.5.2-1 : 메인 페이지에서 수동 진단 페이지 클릭 >



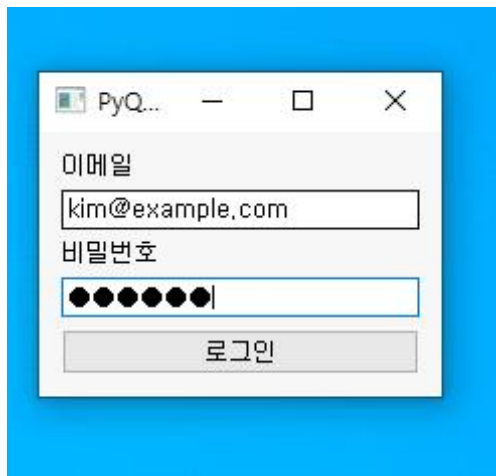
< 그림3.5.2-2 : 사용자의 OS에 따라 다운로드 >



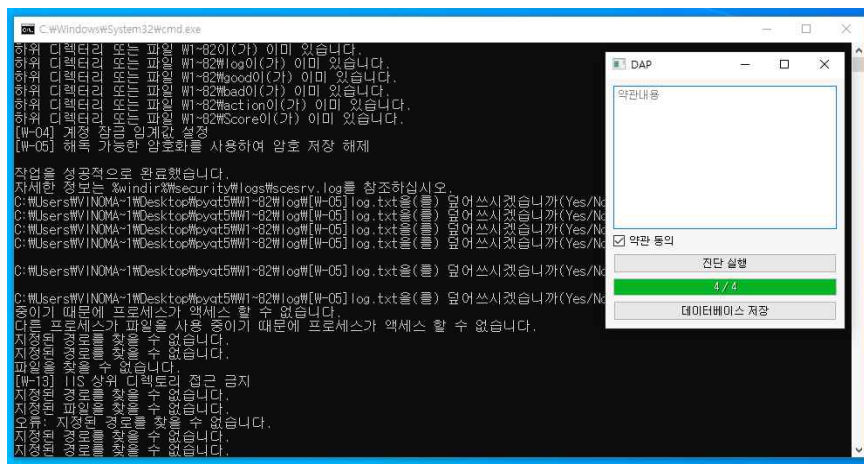
< 그림3.5.2-3 : 배치파일 또는GUI 프로그램 다운로드 >

이름	수정된 날짜	유형	크기
.vscode	2023-06-13 오후 3:41	파일 폴더	
__pycache__	2023-06-13 오후 3:41	파일 폴더	
build	2023-06-13 오후 3:50	파일 폴더	
Report_file	2023-06-16 오후 9:35	파일 폴더	
template	2023-06-13 오후 3:41	파일 폴더	
W1~82	2023-06-15 오후 9:13	파일 폴더	
d.exe	2023-06-15 오전 12:48	응용 프로그램	54,853KB
d.py	2023-06-15 오전 12:46	Python File	10KB
d.spec	2023-06-15 오전 12:47	SPEC 파일	1KB
d-BlueRain.exe	2023-06-13 오후 3:59	응용 프로그램	74,321KB
report.xlsx	2023-06-12 오후 9:50	Microsoft Excel ...	32KB
ScreenSaverIsSecure.txt	2023-06-16 오후 9:11	텍스트 문서	1KB
secpol.txt	2023-06-16 오후 9:33	텍스트 문서	17KB
syne.exe	2023-06-13 오후 3:52	응용 프로그램	90,076KB
syne.py	2023-06-13 오전 11:47	Python File	9KB
syne.spec	2023-06-13 오후 3:50	SPEC 파일	1KB
variables.csv	2023-06-16 오후 9:34	Microsoft Excel ...	315KB
variables.json	2023-06-16 오후 9:34	JSON 원본 파일	668KB
Windo_main.bat	2023-06-13 오후 12:51	Windows 배치 파일	148KB

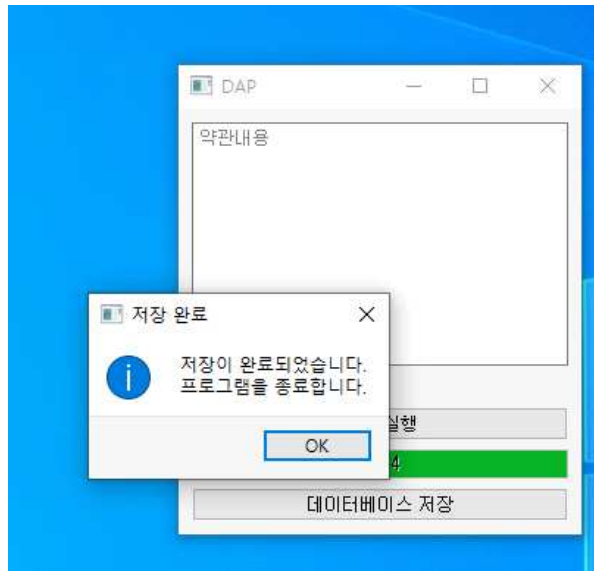
< 그림3.5.2-4 : 다운로드 후 프로그램 실행 로그인 진행>



< 그림3.5.2-5 : 다운로드 후 프로그램 실행 로그인 진행>



< 그림 3.5.2-6: 진단 실행 버튼 클릭>



< 그림3.5.2-7 : 데이터베이스 저장 클릭시 >

DAP Security 취약점진단[AUTO] 취약점진단[MANUAL] 계시판 admin (로그아웃)

### admin님의 수동 진단 결과

이름	계정 관리	서비스 관리	패치 관리	로그 관리	보안 관리	업로드 시각	다운로드	삭제
admin	34.6	46.55	33.3	100.0	59.52	2023-06-20_13-50-02	Download	Delete
admin	34.6	46.55	33.3	100.0	59.52	2023-06-20_13-56-40	Download	Delete
admin	34.6	46.55	33.3	100.0	59.52	2023-06-20_15-59-14	Download	Delete

< 그림3.5.2-8 : 수동진단 결과확인 페이지 >

## IV. 결론

### 1. 결론

서버 취약점 진단 시스템을 제작하여, 주요통신기반시스템 취약점 가이드2017에 따른 항목들을 점검하고, 사용자에게 유의미한 데이터를 제공하는 서비스를 갖추기 위한 웹 사이트를 개발하였다. 웹 서비스에 있어서 가장 중요한 신뢰도를 고려하여 진단 방법을 개발하였기에, 사용자는 원하는 진단 방법을 선택할 수 있다. 또한, 자동진단의 경우 클릭 하나로 서버 취약점에 대한 현황과 상태를 쉽게 확인할 수 있어, 서버를 운영 하는데 있어 도움이 되도록 지원한다.

### 2. 기대효과

이 웹은 호스팅이 되어있기 때문에 누구나 서버에 대한 취약점 진단이 가능하고, 사용자가 취약점 진단 결과물에 대한 데이터 형식을 다양하게 제공하기 때문에, 사용자는 결과물을 그대로 사용해도 되고, 이를 참고하여 다른 결과물을 만들어 도움이 되도록 제공한다.

## V. 별첨

### 1. 팀원 소개

팀장	최용준	보안 개발 및 스크립트 작성
팀원	김다혁	백 엔드 및 GUI 프로그램 개발
	김용훈	백 엔드 및 웹 호스팅
	이진욱	프론트 엔드
	원주연	보안 개발 및 스크립트 작성
	허 현	프론트 엔드

### 2. 소스 코드

깃허브 주소 :	<a href="https://github.com/JBUkim/Dap-host">https://github.com/JBUkim/Dap-host</a>
웹사이트 주소 :	<a href="https://dydgnslra.pythonanywhere.com/">https://dydgnslra.pythonanywhere.com/</a>
GUI 및 스크립트:	<a href="https://drive.google.com/file/d/1hCrEpPNe_rzL90FSW_-_-mu7tkklEP9XL/view">https://drive.google.com/file/d/1hCrEpPNe_rzL90FSW_-_-mu7tkklEP9XL/view</a>

### 3. 발표 자료



파일: [DoAndPray\\_발표자료](#)



#### 4. 소개 자료

# 서버 취약점 원격 진단 사이트



## ☑ 개요

최근 서버에 자산을 특정하여 공격하는 횟수가 증가 하고 있는 만큼, 이에 대한 대책은 필요하며, 서버 공격에 대한 위험요소를 전부 제거하지는 못하더라도, 위협 가능성을 최대한 완화 하는 작업이 필요하다.  
주요통신기반시스템 취약점 가이드 2017을 기준으로 하여, Windows Server와 Linux의 서버의 취약점을 진단하는 시스템을 만들고자 한다.

## ☑ 개발 내용



구상도

## ☑ 팀원 소개

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

파일: [DoAndPray\\_포스터](#)