서버 취약점 자동 진단 프로그램

4조 공포의 외인구단

지도교수 : 양환석 교수님

팀장 : 조재연

김세온

조영준

최송이

최유찬

CONTENTS

1

프로젝트 개요

팀원 소개	팀 구성원 소개
역할분담	팀원별 역할분담
주제선정이유	프로젝트 배경 및 필요성, 프로젝트 주제 및 목적

1 팀원 소개 및 역할 분담

조재연 팀장

- 일정 수립 및 총괄 확인
- Linux 자동진단 스크립트 개발

최송이 팀원

- 자동진단 프로그램 GUI 개발
- 개발 프로그램 최종 Test 진행

김세온 팀원

- Unix 자동진단 스크립트 개발
- Test 서버 제작

최유찬 팀원

- 자동진단 프로그램 GUI 개발
- 개발 프로그램 최종 Test 진행

조영준 팀원

- Windows 자동진단 스크립트 개발
- Test 서버 제작

1 주제선정이유

'모의 해킹' 해보니…기업 서버에서 보안 취약점 무더기 발견

| 과기정통부 "올해 훈련에 IoT·NFT·메타버스 관련 위협 시나리오 반영"

컴퓨팅 | 입력 :2022/01/17 12:00

김윤희 기자 | 기자 페이지 구독 기자의 다른기사 보기

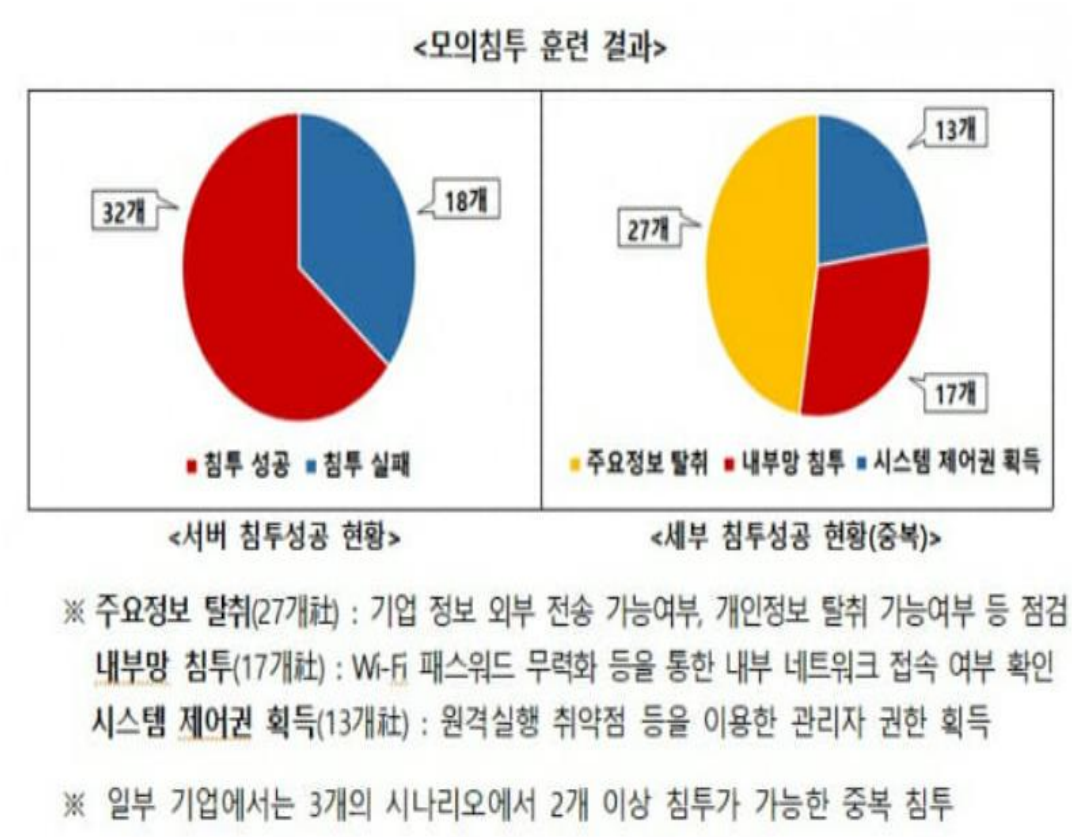
가+ 가-

[이벤트] 이제 고민보다 실행할 때! 상품과 함께 데이터바우처 지원 받으세요 (교촌치킨, 로지텍 등)

해킹 모의 훈련을 실시한 결과 기업 웹사이트와 웹서버, 업무용 서버 등에 보안 취약점이 상당수 탐지된 것으로 나타났다. 웹사이트는 45개사 중 40개사, 웹서버·업무용서버는 50개사 중 32개사가 이런 위험을 내포했던 것으로 확인됐다.

과학기술정보통신부는 한국인터넷진흥원(KISA)과 함께 실시한 지난해 하반기 사이버위기 대응 모의 훈련 결과를 17일 공개하면서 이같이 밝혔다.

지난해 하반기 모의훈련은 지난 11월1일부터 약 3주 동안 참여 기업 285개사, 임직원 9만3천257명을 대상으로 ▲해킹메일 전송 후 대응 절차 점검 ▲분산서비스거부(DDoS) 공격 및 복구 점검 ▲기업의 홈페이지와 서버 대상 모의 침투를 진행했다.



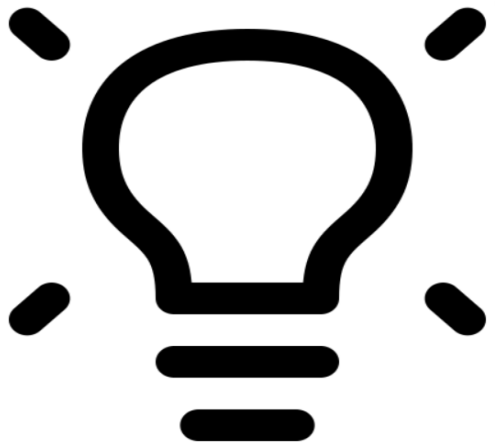
모의 침투 훈련 결과

기업 서버 보안 위협의 심각성

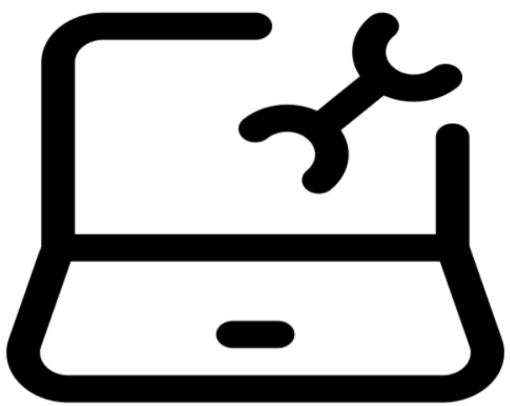
1 주제선정이유

자동화 도구를 이용한 효율적인 서버 진단

“ 서버 취약점 자동 진단 도구를 이용한 취약점 진단 및 보고서 제작 "



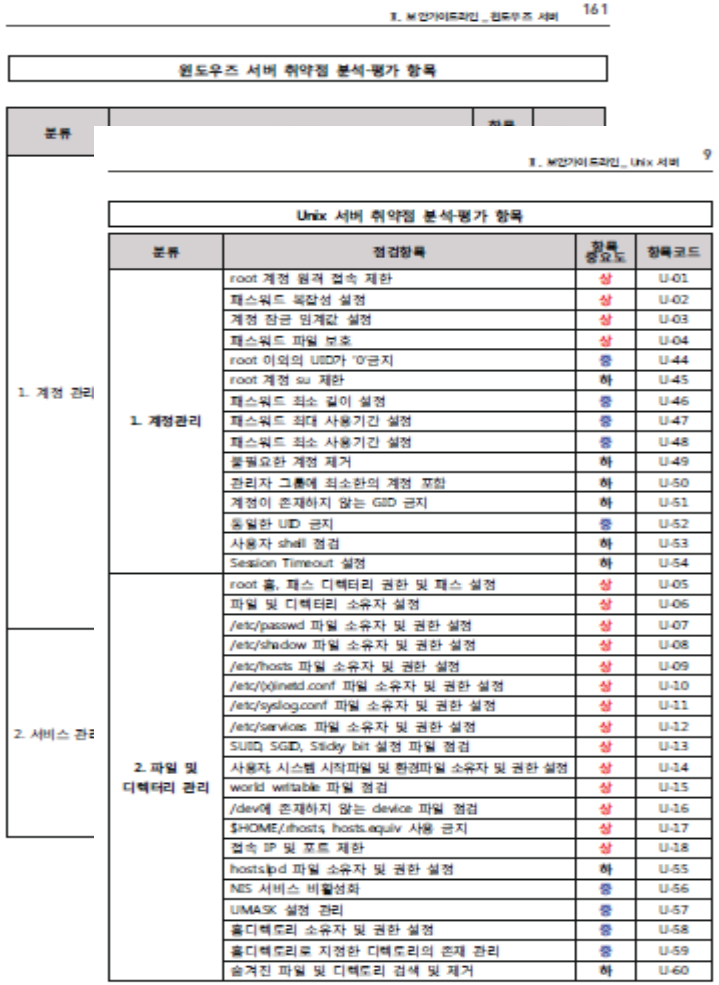
빠른 취약점 위치 식별



취약점 진단 자동화 도구

진단 보고서 제작 / 제공

1 주제선정이유



Ⅱ. 보안가이드라인 _ 윈도우즈 서버 161

윈도우즈 서버 취약점 분석·평가 항목

분류		점검 중요도	
1. 계정 관리			
2. 서비스 관리			

Ⅱ. 보안가이드라인 _ Unix 서버 9

Unix 서버 취약점 분석·평가 항목

분류	점검항목	점검 중요도	항목코드
1. 계정관리	root 계정 원격 접속 제한	상	U-01
	패스워드 복잡성 설정	상	U-02
	계정 잠금 임계값 설정	상	U-03
	패스워드 파일 보호	상	U-04
	root 이외의 UID가 '0'금지	중	U-44
	root 계정 su 제한	하	U-45
	패스워드 최소 길이 설정	중	U-46
	패스워드 최대 사용기간 설정	중	U-47
	패스워드 최소 사용기간 설정	중	U-48
	불필요한 계정 제거	하	U-49
	관리자 그룹에 최소한의 계정 포함	하	U-50
	계정이 존재하지 않는 GID 금지	하	U-51
	동일한 UID 금지	중	U-52
	사용자 shell 점검	하	U-53
	Session Timeout 설정	하	U-54
2. 파일 및 디렉터리 관리	root 홈, 패스 디렉터리 권한 및 패스 설정	상	U-05
	파일 및 디렉터리 소유자 설정	상	U-06
	/etc/passwd 파일 소유자 및 권한 설정	상	U-07
	/etc/shadow 파일 소유자 및 권한 설정	상	U-08
	/etc/hosts 파일 소유자 및 권한 설정	상	U-09
	/etc/(x)inetd.conf 파일 소유자 및 권한 설정	상	U-10
	/etc/syslog.conf 파일 소유자 및 권한 설정	상	U-11
	/etc/services 파일 소유자 및 권한 설정	상	U-12
	SUID, SGID, Sticky bit 설정 파일 점검	상	U-13
	사용자 시스템 시작파일 및 환경파일 소유자 및 권한 설정	상	U-14
	world writable 파일 점검	상	U-15
	/dev에 존재하지 않는 device 파일 점검	상	U-16
	$HOME/.rhosts, hosts.equiv 사용 금지	상	U-17
	접속 IP 및 포트 제한	상	U-18
	hosts.lpd 파일 소유자 및 권한 설정	하	U-55
	NIS 서비스 비활성화	중	U-56
	UMASK 설정 관리	중	U-57
	홈디렉토리 소유자 및 권한 설정	중	U-58
	홈디렉토리로 지정한 디렉토리의 존재 관리	중	U-59
	숨겨진 파일 및 디렉토리 검색 및 제거	하	U-60

주요정보통신기반시설 취약점 점검 가이드

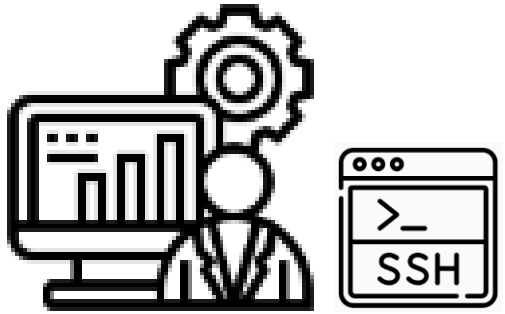
2

프로젝트 진행

시스템 구상도	시스템 개발 구상도
개발 환경	프로젝트 시스템 구축환경
개발 내용	프로그램 구성

2 시스템 구상도

사 용 자



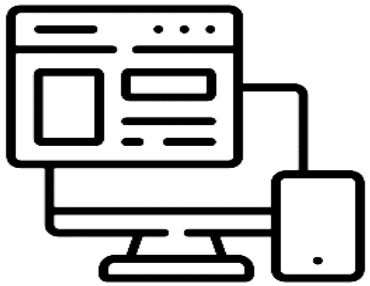
관 리 자

1. 진단요청
(회원 IP 정보입력)

5. 결과 확인

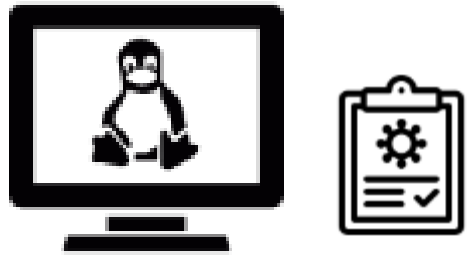
2. 회원정보 전송

3. 고객 서버 원격접속



Python GUI

4. OS별 취약점 진단 및 결과 관리자 / GUI로 전송

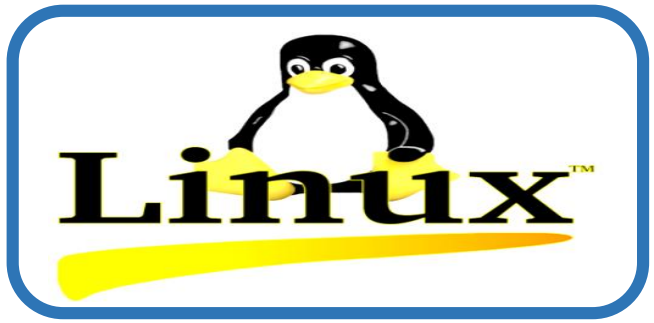


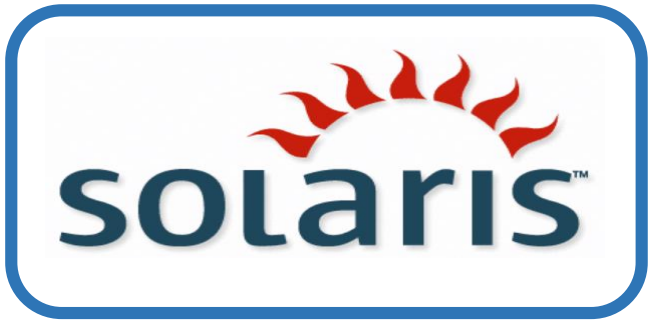
OS / 자동진단 스크립트

2 개발환경

진단 스크립트

진단 환경





GUI

원격접속

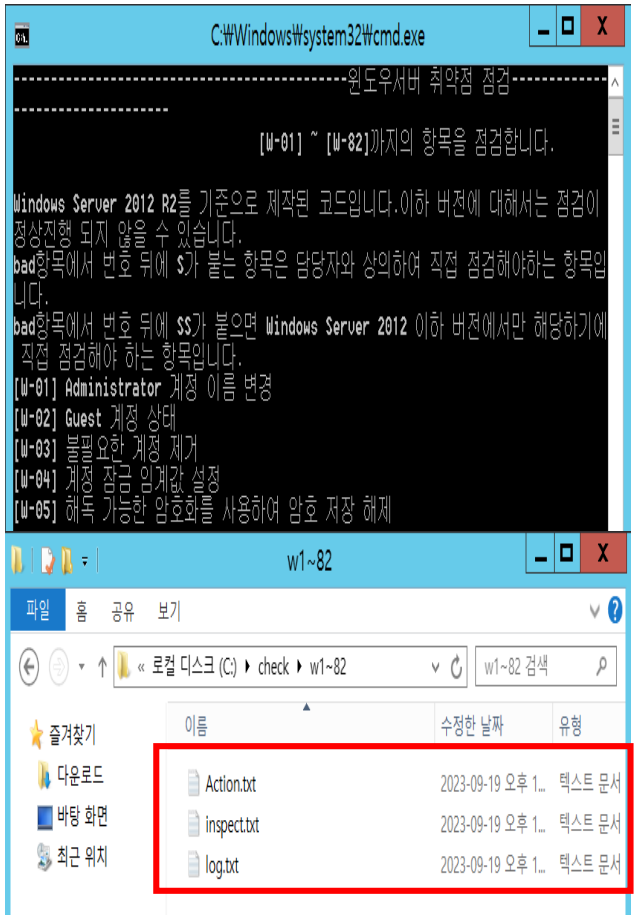
2 프로그램 구성

```
@echo off
echo ----------------------------------------윈도우서버 취약점 점검----------------------------------------
echo                    [W-01] ~ [W-82]까지의 항목을 점검합니다.
echo.
echo Windows Server 2012 R2를 기준으로 제작된 코드입니다.이하 버전에 대해서는 점검이 정상진행 되지 않을 수 있습니다.
echo bad항목에서 변호 뒤에 $가 붙는 항목은 담당자와 상의하여 직접 점검해야하는 항목입니다.
echo bad항목에서 변호 뒤에 $$가 붙으면 Windows Server 2012 이하 버전에서만 해당하기에 직접 점검해야 하는 항목입니다.

mkdir C:\check\w1-82

rem ############################################################
echo [W-01] Administrator 계정 이름 변경
echo.>> C:\check\w1-82\log.txt
echo ##########[W-01] Administrator 계정 이름 변경##########>> C:\check\w1-82\log.txt

net user > account.txt
net user >> C:\check\w1-82\log.txt

type account.txt | find /I "Administrator" > NUL
if %errorlevel% EQU 0 (
    echo A^|[W-01]  Administrator 계정이 존재함 - [취약] >> C:\check\w1-82\inspect.txt
    echo.>> C:\check\w1-82\Action.txt
    echo [조치사항] >> C:\check\w1-82\Action.txt
    echo [W-01] 시작- 프로그램- 제어판- 관리도구- 로컬 보안 정책 - 로컬 정책 - 보안옵션 >> C:\check\w1-82\Action.txt
    echo [W-01] 계정: Administrator 계정 이름 바꾸기를 유추하기 어려운 계정 이름으로 변경 >> C:\check\w1-82\Action.tx
) else (
    echo A^|[W-01] Administrator 계정이 존재하지 않음 - [양호] >> C:\check\w1-82\inspect.txt
)

del account.txt
```

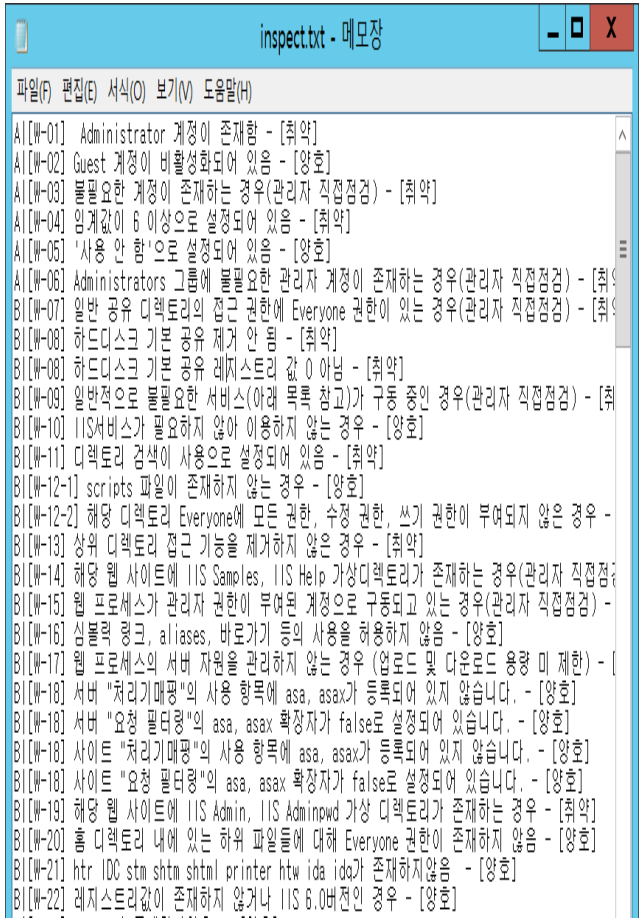
batch File

```
#!/bin/bash

mkdir -p /check/U1-73/
mkdir -p /check/U1-73/log /check/U1-73/action /check/U1-73/bad /check/U1-73/good



#########[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

echo -e "##########[U-01]root 계정 원격 접속 제한##########\n[root 직접접속 차단 여부]" >> /check/U1-73/log/[U-01]log.
grep 'pts' $CF1 >> /check/U1-73/log/[U-01]log.txt
echo -e "\n[원격 터미널 서비스 사용 여부]" >> /check/U1-73/log/[U-01]log.txt
grep '/lib/security/pam_securetty.so' $CF2 | grep 'required' | grep 'auth' >> /check/U1-73/log/[U-01]log.txt
#로그파일에 출력에 대한 제목후 개행하여 길어도 보기 편하게 출력
cat /check/U1-73/log/[U-01]log.txt >> /check/U1-73/log.txt



if [[ $pam == 'auth' ]] || [[ -z $pts ]]; then
 echo -e "A|[U-01] 원격 터미널 서비스를 사용하지 않거나, 사용 시 root 직접 접속을 차단되어 있음 - [양호]" >> /check/U1-73/go
 cat /check/U1-73/good/[U-01]good.txt >> /check/U1-73/inspect.txt
else
 echo -e "A|[U-01] 원격 터미널 서비스 사용 시 root 직접 접속이 허용되어 있음 - [취약]" >> /check/U1-73/bad/[U-01]bad.txt
 cat /check/U1-73/bad/[U-01]bad.txt >> /check/U1-73/inspect.txt
 echo -e "[U-01] vi 편집기를 사용하여 /etc/securetty 파일을 열어 pts/* 설정이 존재하는 경우 제거 또는 주석처리\nv1 편집기를 사
 sed -e 's/\[U-01\] /\n[(U-01)조치사항]\n/g' /check/U1-73/action/[U-01]action.txt >> /check/U1-73/Action.txt
```

Shell Script



스크립트 실행

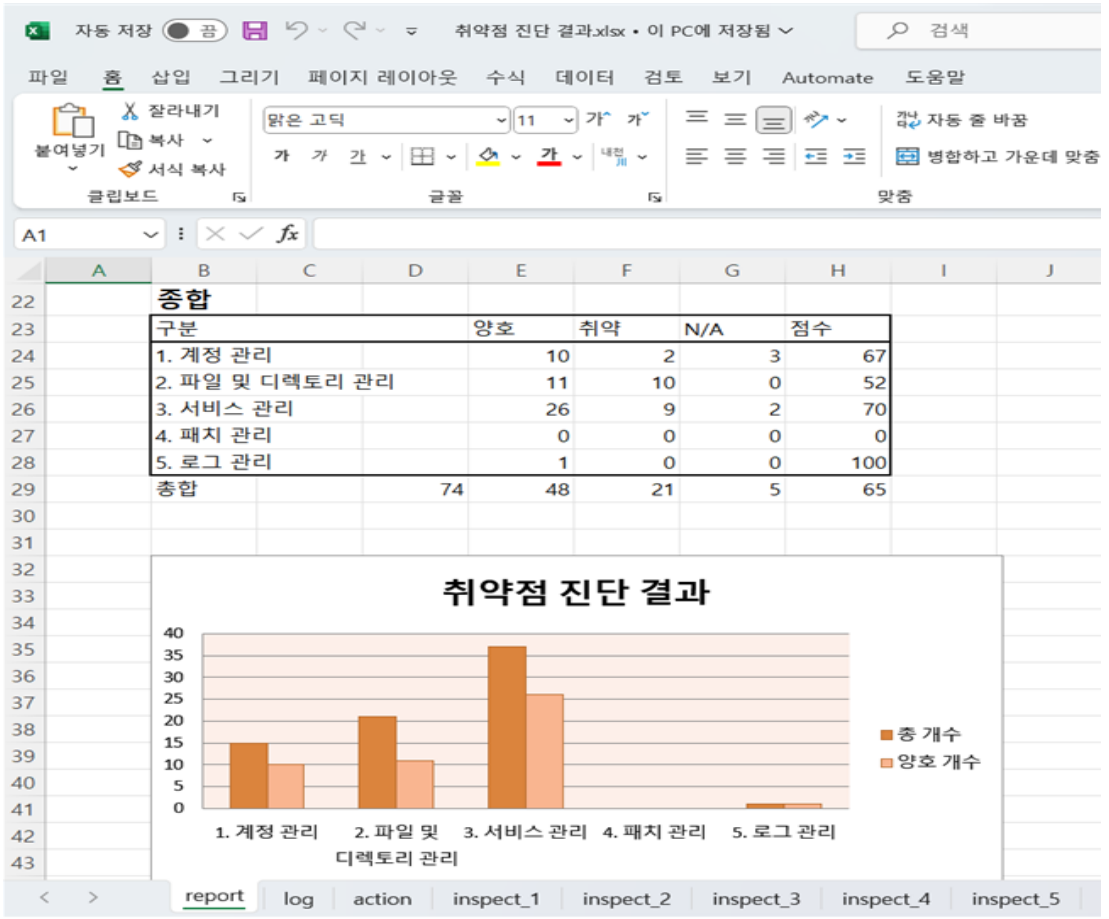


진단 결과(.TXT)

서버 취약점 자동진단 스크립트

2 프로그램 구성

```
wb.save(new_filename)
# 여기까지 엑셀파일 생성 후 저장
############################################################

wb = load_workbook(new_filename)  # new_file(위에서 만든 엑셀) 불러오기
# 엑셀파일 불러오는 경우 new_filename대신 엑셀파일 이름 넣기
# wb.save(new_filename)의 경우도 마찬가지
ws = wb.active

###시트지정
sh_list = wb.sheetnames  # 시트들을 리스트로 구분
w1 = wb[sh_list[0]]  # 첫번째 시트(report)를 w1로 지정
w2 = wb[sh_list[1]]  # 두번째 시트(log)를 w2로 지정
w3 = wb[sh_list[2]]  # 세번째 시트(action)를 w3으로 지정
w4 = wb[sh_list[3]]  # 네번째 시트(1. 계정관리)를 w4로 지정
w5 = wb[sh_list[4]]  # 다섯번째 시트(2. 파일 및 디렉토리 관리)를 w5로 지정
w6 = wb[sh_list[5]]  # 여섯번째 시트(3. 서비스 관리)를 w5로 지정
w7 = wb[sh_list[6]]  # 일곱번째 시트(4. 패치 관리)를 w6로 지정
w8 = wb[sh_list[7]]  # 여덟번째 시트(5. 로그 관리)를 w7로 지정

###계산
# ws["A6"] = "=SUM(A4:A5)"

left = Border(left=Side(style="thin"))
right = Border(right=Side(style="thin"))
top = Border(top=Side(style="thin"))
bottom = Border(bottom=Side(style="thin"))

##report 시트 항목 별 테두리
for i in range(4, 21, 4):
```

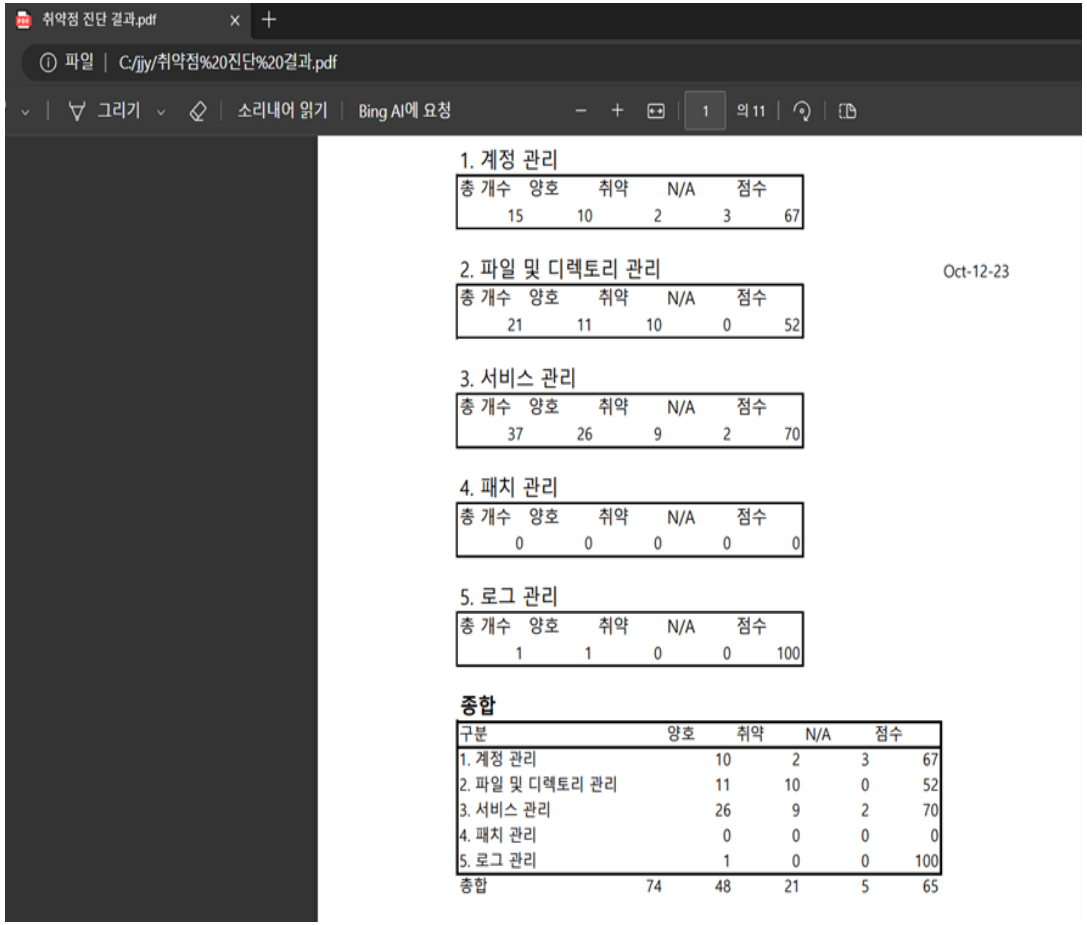


엑셀 파일 자동 생성

진단 결과 보고서 자동 작성 프로그램

2 프로그램 구성

```python
def makepdf(self):
    filenames = os.listdir("C:\\jjy\\")
    if "취약점 진단 결과.xlsx" in filenames:
        print('PDF생성이 가능합니다.')
        QMessageBox.information(self, title: 'Message', text: 'PDF 생성이 가능합니다',

        excel = win32com.client.Dispatch("Excel.Application")
        excel.Visible = False
        wb = excel.Workbooks.Open("C:\\jjy\\취약점 진단 결과.xlsx")
        ws_report = wb.Worksheets("report")
        ws_report.Select()
        pdf = "C:\\jjy\\취약점 진단 결과.pdf"
        wb.ActiveSheet.ExportAsFixedFormat(0, pdf)
        wb.Close(False)
        excel.Quit()
    else:
        QMessageBox.information(self, title: 'Message', text: 'xlsx파일이 존재하지 않

1개의 사용 위치
def openpdf(self):  # 실행 코드
    filenames = os.listdir("C:\\jjy\\")
    if '취약점 진단 결과.pdf' in filenames:
        os.startfile("C:\\jjy\\취약점 진단 결과.pdf")
    else:
        QMessageBox.information(self, title: 'Message', text: 'pdf파일이 존재하지 않아
```



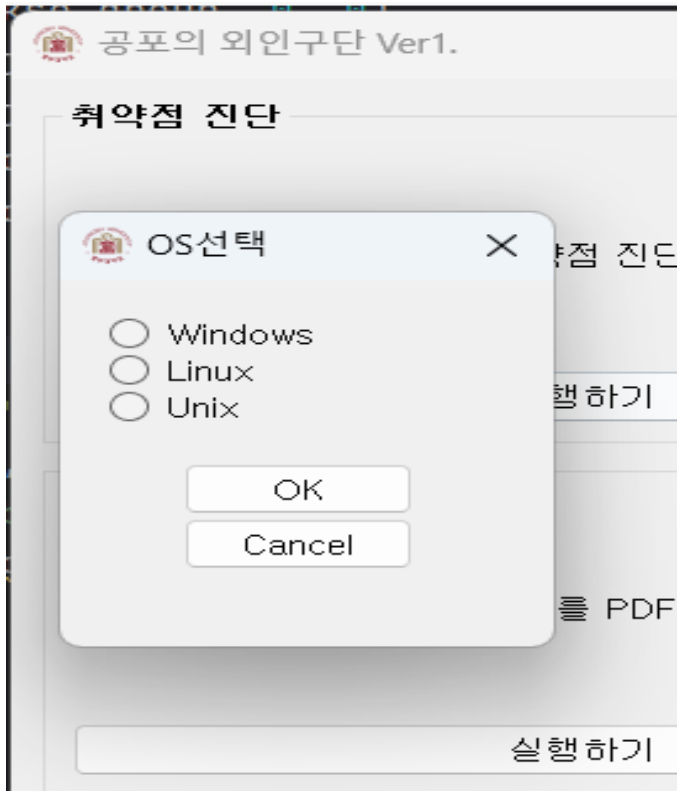
PDF 파일 자동 생성

진단 결과 보고서 자동 작성 프로그램

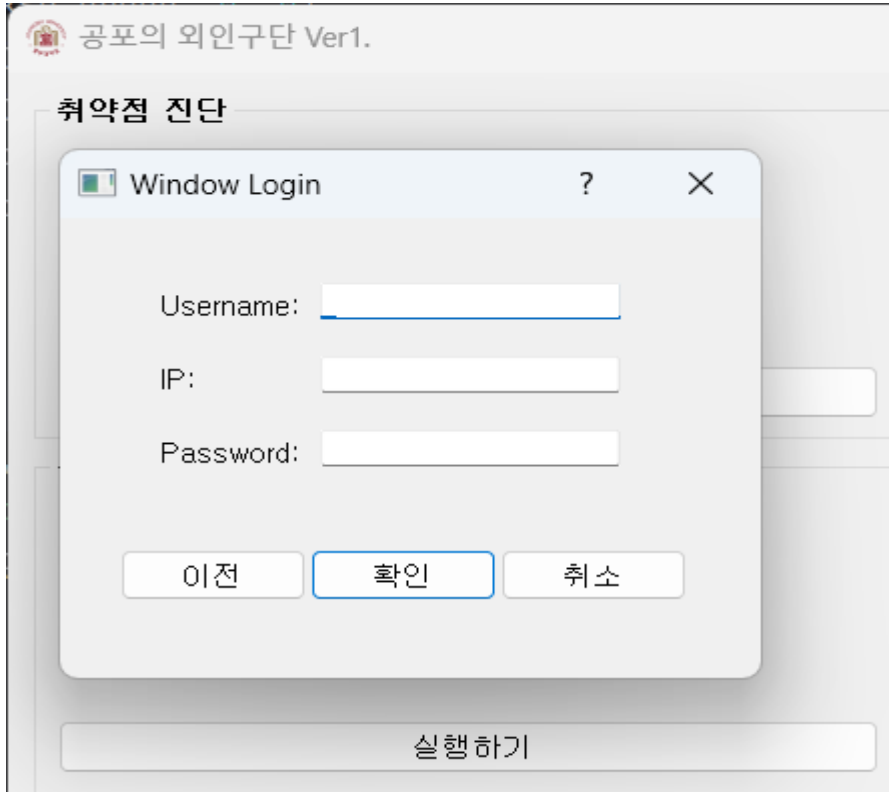
2 프로그램 구성

GUI 구성

GUI 구성 모습



서버 OS 선택

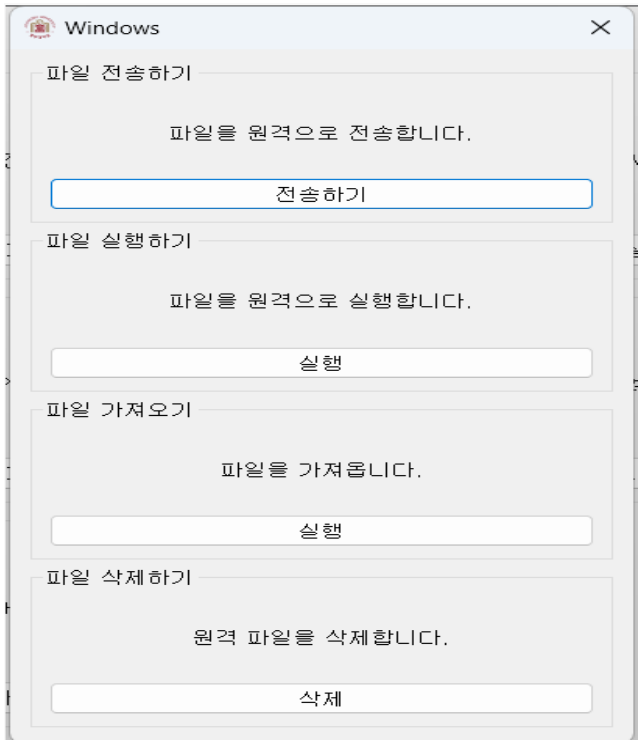


사용자 정보 입력

서버 OS 선택 / 사용자 정보입력

2 프로그램 구성

GUI 구성

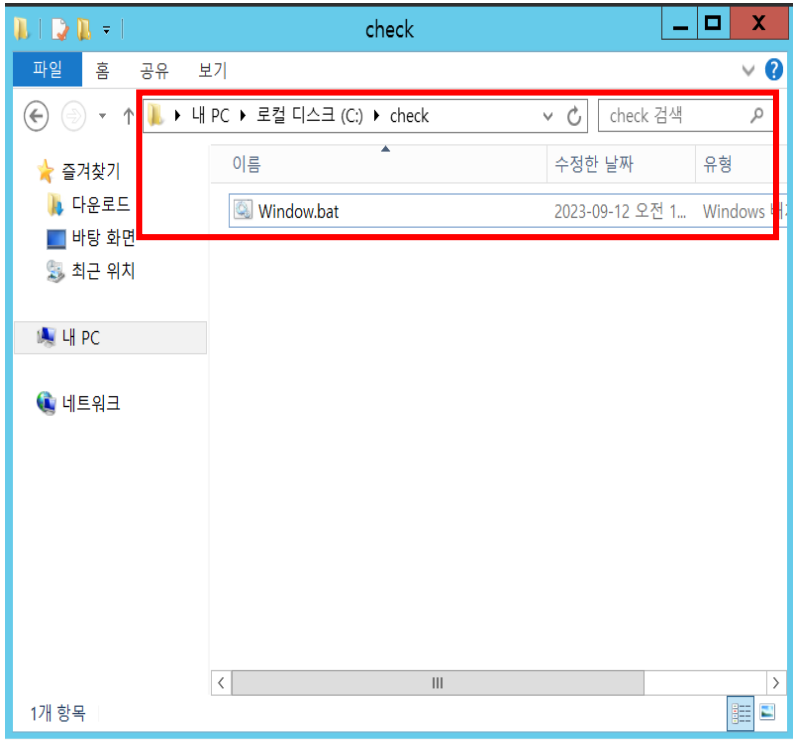


원격 접속 후 실행 화면

```
if name == "" or ipa == "" or pwd == "":
    QMessageBox.information(self, title: 'Message', text: '값을 입력하세요', QMessageBox.Ok)

else:
    QMessageBox.information(self, title: 'Message', text: '로그인 성공', QMessageBox.Ok)

    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy)
    ssh.connect(ipa, username=name, password=pwd)

    trans = paramiko.transport.Transport(ipa, default_window_size: 22)
    trans.connect(username=name, password=pwd)
    sftp = paramiko.SFTPClient.from_transport(trans)
    #LoginWindow.close()
    self.w = action.Windowsgaka()
    self.w.show()
    self.close()
```

```
LoginWindow › login() › else
실행  main
C:\Users\gamin\AppData\Local\Programs\Python\Python311\python.exe C:\jjy\GUI2\1005\main.py
Administrator 192.168.136.138 cho9091009#
SSH 연결 활성화
```

파일 전송 코드

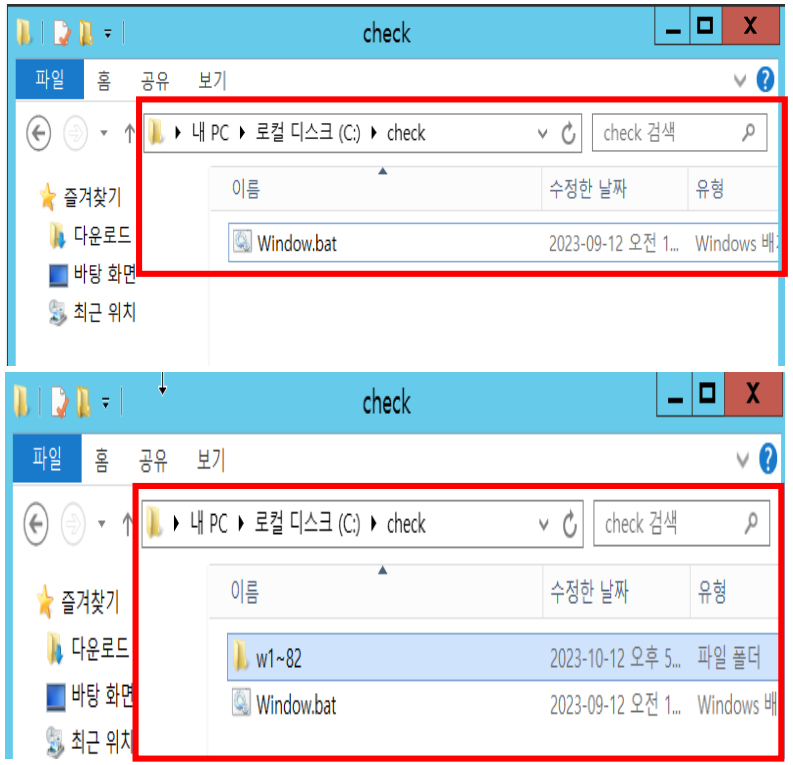


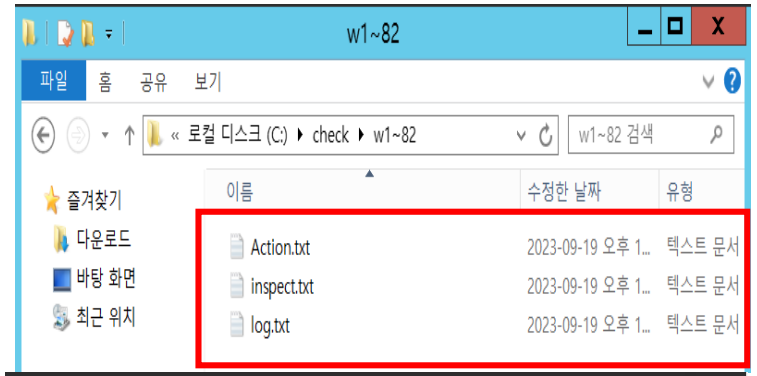
사용자 서버 파일 전송 완료

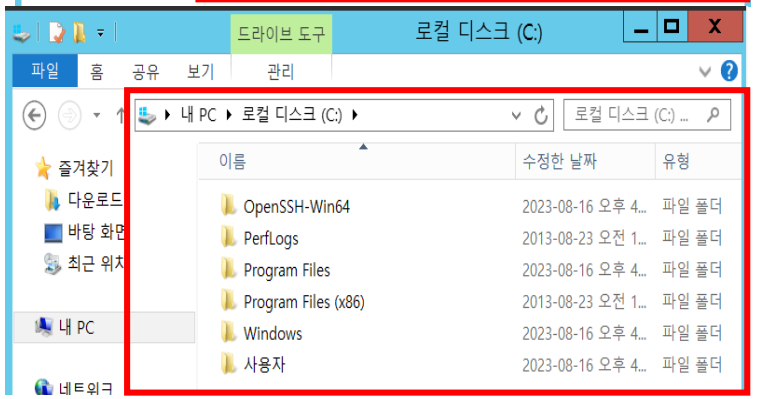
원격접속 / 진단 스크립트 파일 전송

2 프로그램 구성

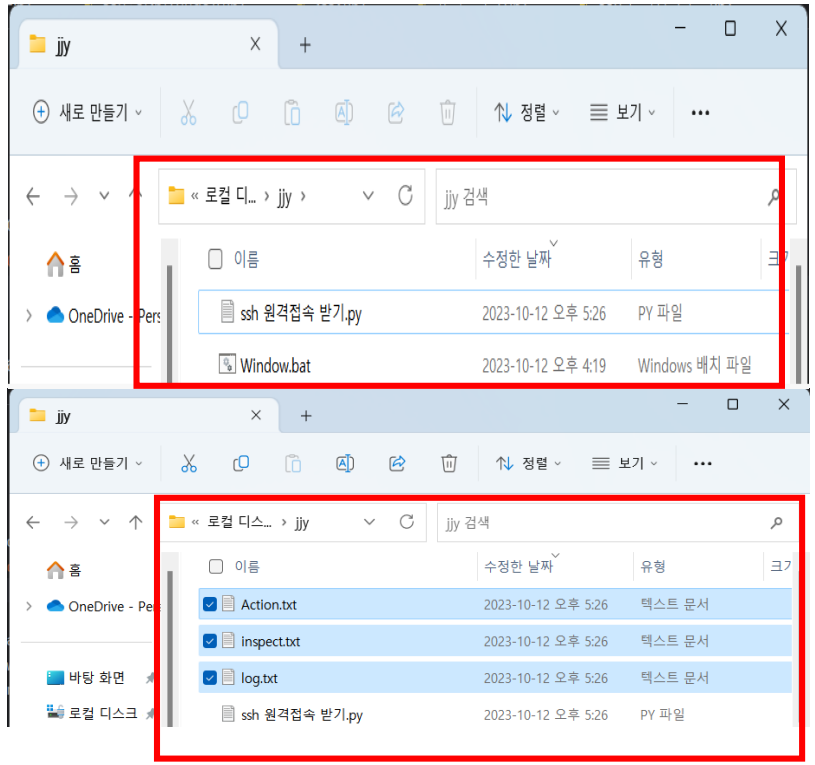
GUI 구성







실행 후 화면(사용자 서버)

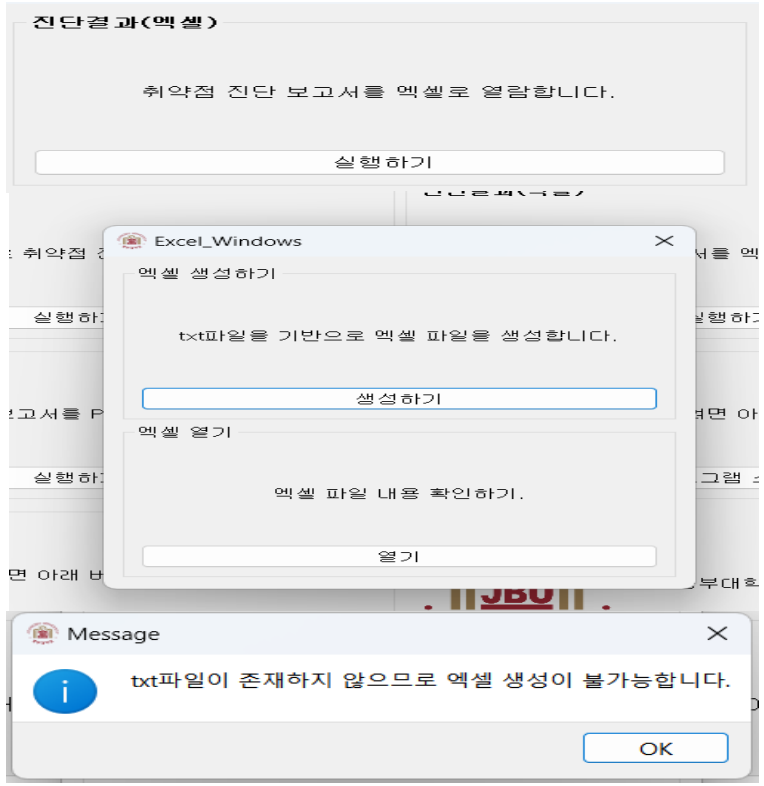


파일 추출 후 화면(관리자 서버)

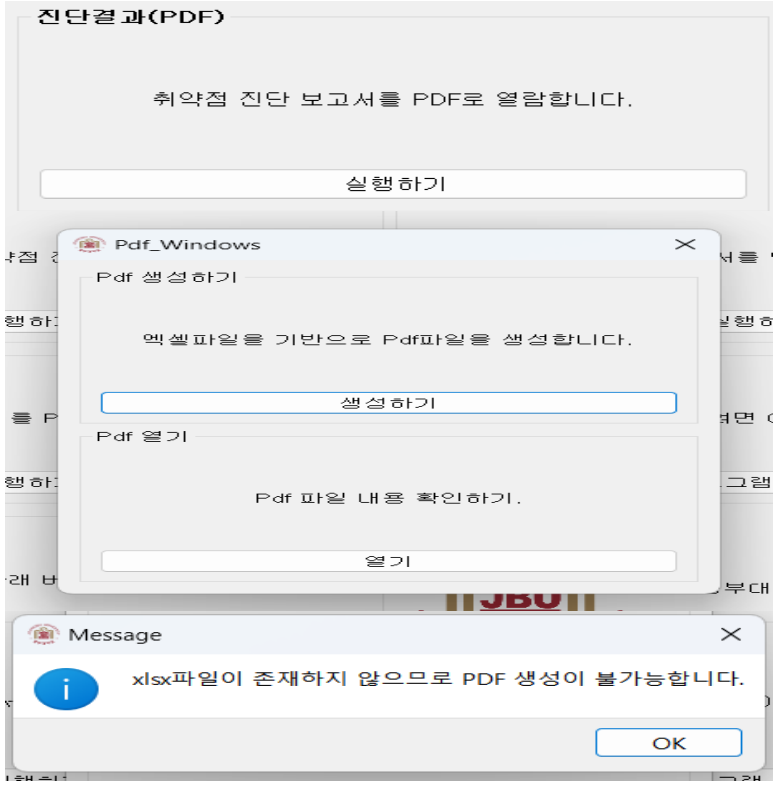
진단 스크립트 파일 실행 / 결과 가져오기 / 진단 파일 삭제

2 프로그램 구성

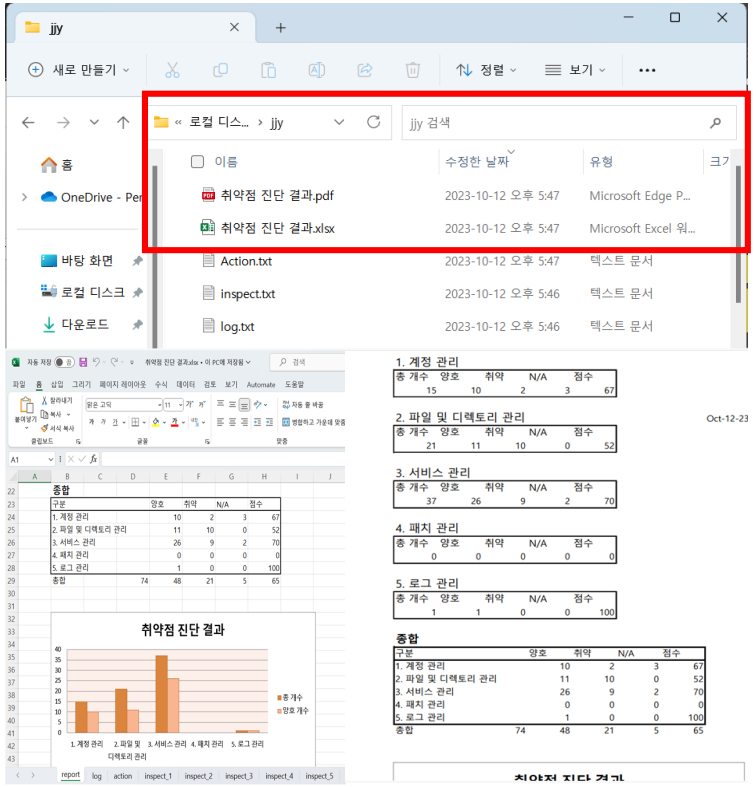
GUI 구성



엑셀 파일 생성



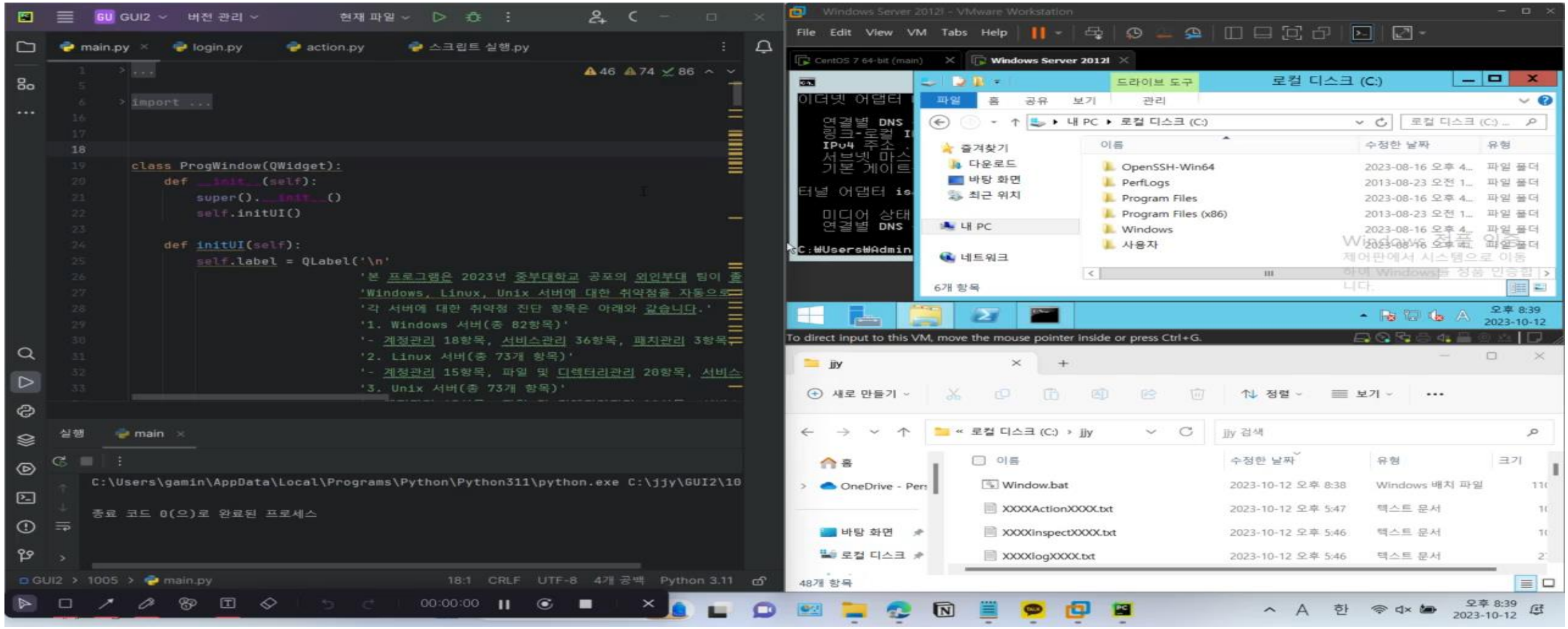
PDF 파일 생성



파일 생성 결과

진단 결과 엑셀 파일 및 PDF 파일 생성 / 실행(출력)

2 프로그램 구성



자동진단 프로그램 시연 영상

3

결 론

기대효과

프로젝트 기대효과

3 기대효과

프로젝트 기대효과

취약점 점검 효율성 증대

자동 진단도구 활용으로
취약점 진단 시간 단축

최신 취약점 유형 연구

식별된 취약점을 통해
주요정보통신기반
취약 유형 연구

빠른 취약점 위치 식별

잠재적 취약점 발생 요소
파악 및 대처

보안 역량 강화

진단 결과 보고서 및 조치
가이드 기반 취약점 보완으로
담당자 보안 역량 강화

Thank You