서버 취약점 자동진단 프로그램 개발

팀　　명 :　공포의 외인구단

지 도 교 수 :　양환석 교수님

팀　　장 :　91712781 조재연

팀　　원 :　91806427 김세온

92251150 조영준

92015491 최송이

91914377 최유찬

2023. 11.

중부대학교 정보보호학과

목 차

1. 서 론

1.1 연구배경

3차 산업혁명 이후 대한민국의 수많은 기업 및 국가시설 또는 개인이 서버를 구축하고 사용하고 있다. 이에 따라 서버에 대한 정보보안의 중요성이 대두되고 있지만 정작 서버를 운영하는 기업의 관심 부재로 인해 기업 서버에서 보안 취약점이 무더기로 발견되고 있는 실정이다. <그림 1-1>과 같이 2022년 1월 과학기술정통부의 발표에 따르면 KISA(한국인터넷진흥원)와 합동으로 기업 285개사를 대상으로 진행한 사이버위기 대응 모의훈련 간 참여 기업 서버를 대상으로 모의침투 훈련을 진행한 결과 32개 기업의 서버에 침투에 성공했고, 이 중 27개 기업 서버에서 중요 정보를 탈취, 17개 기업의 서버에 내부망 침투, 13개 기업의 서버에서 시스템 제어권을 획득하는 등 기업 서버 보안 위협이 심각한 것으로 나타났다. 서버 보안을 위협하는 공격자(해커)는 운영하고 있는 서버의 보안 취약점을 찾아 서버로 침투하여 정보시스템의 위·변조, 파괴 등을 통해 서버에 저장된 각종 정보를 탈취하거나 서버에 접속하는 사용자의 개인정보를 탈취하는 등의 피해를 발생시켜 서버를 운영하는 기업 또는 국가기관의 신뢰도를 하락시키고 사용자들로 하여금 불안에 떨게 만든다. 이러한 문제점을 보안하고 해결하기 위해 KISA(한국인터넷진흥원)에서 제공하는 주요 정보통신기반시설 기술적 취약점 분석·평가방법 상세가이드를 바탕으로 서버 취약점 진단 자동화 프로그램을 제작하게 되었다.

<그림 1-1>

1.2 연구 필요성

주요정보통신기반시설 관리기관은 “정보통신기반 보호법” 제 9조에 따라 주요정보통신기반시설로 신규 지정된 이후 6개월 이내, 그리고 매년 취약점 분석·평가를 실시하여야 한다. 취약점 분석·평가는 453개의 관리적 / 물리적 / 기술적 점검항목에 대한 주요정보통신기반시설의 취약여부를 점검하여, 악성코드 유포, 해킹 등 사이버 위협 대응을 위한 종합적 개선과정이다. 하지만 기업 서버의 정보보안을 담당하는 담당자들이 취약점 분석·평가를 수행함에 있어 기술적 점검은 시스템에 대한 실제 보안 값 설정에 관한 것으로, 각 시스템에 대한 명령어 코드, 메뉴 구성과 같은 기술적인 사항을 충분히 숙지하여야 가능하기에 담당자들의 어려움이 따르게 된다. 이에 따라 기업에서는 기업에서 운영하는 서버에 대한 취약점 분석·평가를 전문 업체에 위탁하고 있으나 전문 업체선정, 진단 소요기간, 진단 후 보완에 이르기까지 많은 인력과 시간 소요가 발생하고 있는 실정이다. 이러한 구조적인 문제점을 해결하기 위해 원격으로 기업의 서버에 접속하여 자동으로 취약점을 분석 및 평가하고, 그 결과를 리포트(PDF) 형태로 실시간으로 확인하고 보완할 수 있는 자동화 진단 프로그램 개발이 필요하다.

1.3 연구 목적 및 주제선정

본 연구는 기업 및 기관, 개인이 운영하는 서버에 대한 취약점을 분석·평가함에 있어 보다 신뢰성 있고, 간편하고, 시간 소요가 적은 방법으로 서버 취약점을 분석·평가하고, 진단 결과를 바탕으로 도출된 취약점을 업무 담당자들이 보다 편리하고 간편하게 보완 가능하게 구현하는 것을 목적으로 서버 취약점 자동화 진단 프로그램 개발을 주제로 선정하였다.

2. 관련연구

2.1 Windows

윈도우는 마이크로소프트사가 개발한 운영체제로 MS-DOS에서 멀티태스킹과 GUI 환경을 제공하기 위한 응용 프로그램으로 처음 출시되었다. 현재 전 세계 90%의 개인용 컴퓨터에서 쓰고 있으며, 서버용 운영 체제로도 점차 영역을 넓혀 나가고 있다. 윈도우 운영 체제의 경우 일반 사용자들에게 매우 익숙하고 호환되는 유명한 응용 프로그램이 많다는 장점을 지니고 있지만, 그만큼 보안 문제에서는 취약한 부분이 많은 운영 체제로 인식되기도 한다. 현재 PC에서 가장 많이 사용하는 운영체제로 자리매김하고 있다.

2.2 Liunx

리눅스는 1991년 핀란드 소프트웨어 개발자인 리누스 토발즈(Linus Torvalds)가 개발한 오픈소스 운영 체제로 GPL(GNU General Public License)를 사용하여 출시되었다. 따라서 누구나 소프트웨어를 실행, 연구, 공유, 수정할 수 있는 프리웨어로 누구든 리눅스를 자신의 사용 용도에 맞게 마음대로 바꿀 수 있다. 또한, 모든 소스가 공개되어 있기 때문에 취약점이 노출되어도 비교적 빠른 보안 업데이트를 진행한다. 현재 가장 많은 사용자 기반을 보유하고, 공개적으로 이용 가능한 인터넷 서버에서 가장 많이 사용되며, 가장 빠른 상위 500대 슈퍼컴퓨터에서 유일하게 사용되는 OS가 되었다.

2.3 Unix

유닉스는 1970년대 초반 벨 연구소 직원인 켄 톰슨, 데니스 리치 등이 소형 컴퓨터용으로 처음 개발한 운영체제로 교육 및 연구 기관에서 즐겨 사용되는 범용 다중 사용자 방식의 대화식, 시분할처리 시스템용 운영 체제이다. 유닉스는 처음부터 다양한 시스템 사이에서 서로 이식할 수 있고, 멀티태스킹과 다중 사용자를 지원하도록 설계되었다. 유닉스는 안정적이고 신뢰할 수 있으며 다양한 유형의 작업을 처리할 수 있는 것으로 알려져 있다.

2.4 Batch File

배치 파일은 마이크로소프트사의 DOS 운영체제인 MS-DOS에서 명령어를 사용하는 것에 익숙하지 않은 사용자를 위해 명령 인터프리터에 의해 실행되게끔 고안된 명령어들이 나열되어 있는 텍스트 파일이다. 배치 파일이 실행될 때, COMMAND.COM 또는 cmd.exe와 같은 셸 프로그램이 파일을 읽어 명령어를 줄 단위로 실행한다. 배치 파일은 보통 실행 파일을 자동으로, 연속적으로 실행할 때 유용하며 시스템 관리자가 반복적인 업무를 자동화하는데 많이 사용된다.

2.5 Shell script

Shell(쉘)은 운영체제상에서 사용자가 입력하는 명령을 읽고 해석하여 대신 실행해주는 프로그램이다. 즉, 운영체제의 커널과 사용자 사이를 이어주는 역할을 하며 사용자의 명령어를 해석하고 운영체제가 알아들을 수 있도록 도와주는 명령어 해석기이다. Shell Script(쉘 스크립트)란 Shell(쉘)에서 사용할 수 있는 명령어들의 조합을 모아서 만든 배치(batch)파일이다. 즉, 운영체제의 Shell을 이용하여 한 줄씩 순차적으로 읽으면서 명령어들을 실행시켜주는 인터프리터 방식의 프로그램이다. Shell Script를 활용하여 묶어진 명령어 조합을 수행하거나 반복적인 명령어를 단일 명령으로 쉽게 사용할 수 있다.

2.6 Python

파이썬은 1991년에 발표된 인터프리터 방식의 프로그래밍 언어이다. 파이썬의 강력한 라이브러리와 풍부한 생태계를 통해, 데이터를 수집하고 분석하며 시각화하기 용이하다. 파이썬은 효율적인 자료 구조와 객체 지향 프로그래밍에 대한 간단하고도 효과적인 접근법을 제공한다. 파이썬은 우아한 문법과 동적 타이핑(typing)을 지원하는 인터프리터 언어로서, 대부분 플랫폼과 다양한 문제 영역에서 스크립트 작성과 빠른 응용 프로그램 개발에 이상적인 환경을 제공하며, 파이썬 소프트웨어는 무료로 다운로드할 수 있고, 모든 유형의 시스템과 원활하게 통합되며, 개발 속도를 증가시킨다.

2.7 GUI(Graphical user interface)

그래픽 사용자 인터페이스(GUI)는 사용자가 컴퓨터와 정보를 쉽게 교환하고 상호 작용을 하기 위해 아이콘 등과 같은 그래픽을 이용한 사용자 인터페이스를 뜻한다. 그래픽 사용자 인터페이스(GUI)는 사용자가 커맨드 라인(명령행)을 키보드를 통하여 컴퓨터에 입력하여 작업을 수행시키고 컴퓨터는 작업 결과를 문자로 화면에 표시하는 문자 중심의 조작 대신에, 사용자가 키보드 입력뿐만 아니라 마우스 등의 위치 지정 도구를 사용하여 도형의 형태로 화면에 표시되는 아이콘(icon)을 지정하거나 메뉴 항목 목록 중에서 메뉴를 선택함으로써 명령을 선택하고, 프로그램을 기동하며, 파일 목록을 열람하고 기타 선택을 하면서 작업을 진행하는 상호 작용 방식으로 GUI는 사용자가 직관적으로 조작 방법을 이해할 수 있게 고안되어 있는 것이 장점이다.

2.8 Open SSH(OpenBSD Secure Shell)

오픈SSH는 시큐어 셸(Secure Shell, SSH) 프로토콜을 이용하여 클라이언트 - 서버 아키텍처를 사용하여 두 시스템 간에 보안 통신을 제공하고 사용자가 서버 호스트 시스템에 원격으로 로그인할 수 있는 프로토콜이다. FTP 또는 Telnet과 같은 다른 원격 통신 프로토콜과 달리 SSH는 로그인 세션을 암호화하여 침입자가 연결에서 암호화되지 않은 암호를 수집하지 못하게 방지한다.

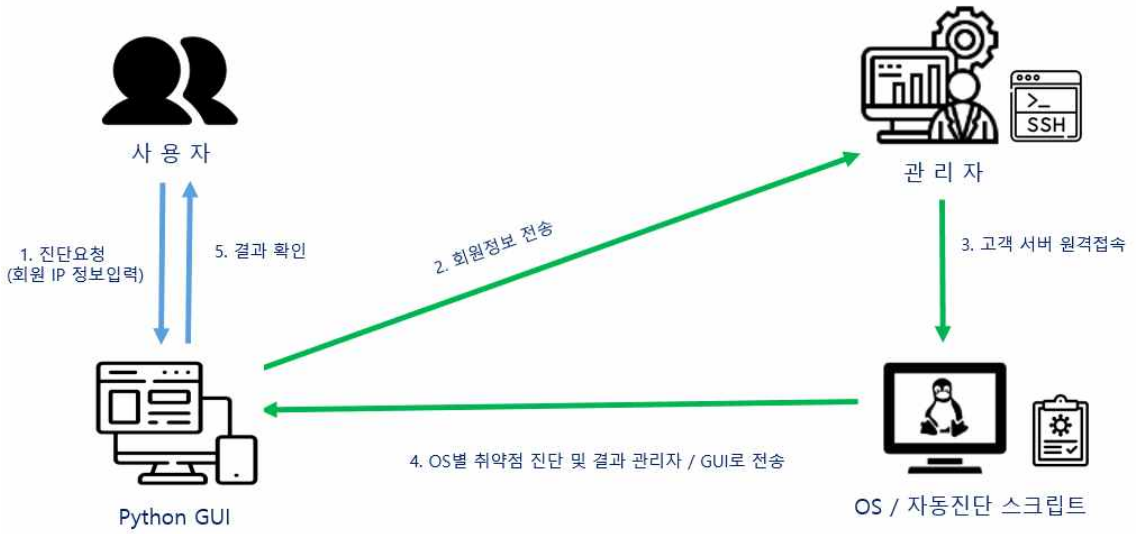
2.9 STP(Spanning tree protocol)

STP는 스패닝 트리 프로토콜(Spanning tree protocol)의 약자로 루프를 확인하고 적절히 포트를 사용하지 못하게 하는 프로토콜이다. 스패닝 트리 프로토콜은 트리 루트와 리프 사이에 단 하나의 가용 경로로 전체 브리징 네트워크의 트리 토폴로지(스패닝 트리)를 생성하여 네트워크의 루프를 지능적으로 방지한다. 스패닝 트리 프로토콜은 바람직하지 않은 루프를 방지하는 동시에 링크 이중화 제공, 브로드캐스트 스톰 방지, 에지 포트를 사용하여 다른 스위치에 연결되지 않은 PC, 서버, 라우터 또는 허브와 같이 STP를 지원하지 않는 디바이스에 연결하는 이점을 제공한다.

3. 본 론

3.1 시스템 구성

서버 취약점 분석·평가를 희망하는 사용자가 GUI를 통해 서버 IP. ID. Password를 입력하고 자신의 서버에 해당하는 OS를 선택하여 취약점 분석·평가를 요청하면 관리자가 사용자의 서버에 원격으로 접속하여 서버 취약점 분석·평가 후 진단 결과를 TXT 파일로 출력 후 엑셀 파일 및 변환하고, 엑셀 파일을 보고서 형태인 PDF 파일로 재 변환 및 출력시켜 사용자가 한눈에 결과를 확인할 수 있도록 <그림 1-2>와 같이 시스템을 구성하였다.



<그림 1-2>

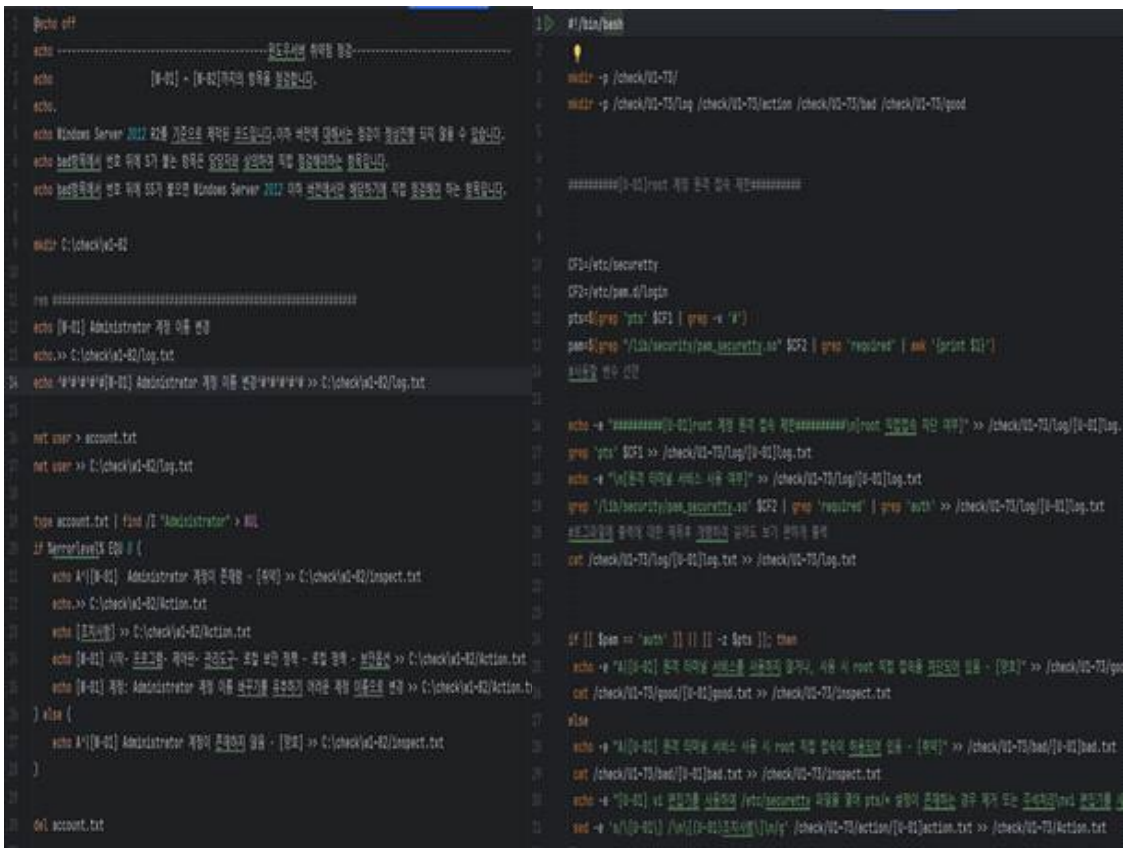
3.2 프로그램 구성

서버 취약점 진단 자동화 프로그램은 Windows, Liunx, Unix 서버를 자동으로 진단하기 위한 진단 스크립트(Batch File, Shell script), 진단 결과 보고서를 자동으로 작성해 주는 프로그램, 사용자가 정보를 입력하고 서버 취약점 진단 자동 점검 프로그램 실행 및 점검 결과를 한눈에 볼 수 있도록 구성된 Python GUI(Graphical user interface)로 구현되었다.

3.2.1 진단 스크립트

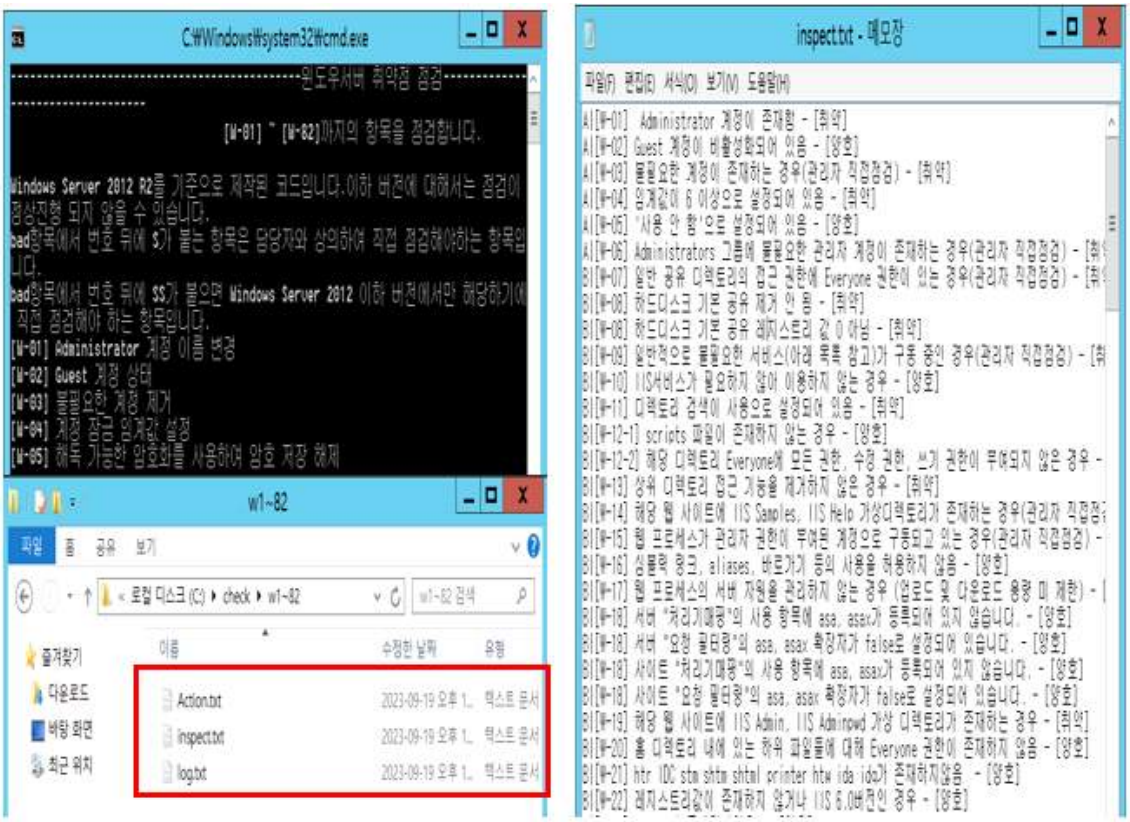
각 서버에 대한 취약점을 자동으로 진단하기 위해 VM-ware에 Windows Server 2012 R2, Cent OS 7, Solaris 10 서버를 설치하고 KISA(한국인터넷진흥원)에서 제공하는 “주요 정보통신 기반시설 기술적 취약점 분석·평가방법 상세가이드”를 바탕으로 각 서버에 대한 취약점 분석·평가를 수동으로 점검한 뒤 서버 자동 진단 스크립트 테스트를 위해 각 서버의 설정을 변경하여 테스트 서버를 제작하고 서버 자동 진단 스크립트 개발 후 테스트를 진행하여 검증함으로써 신뢰성을 확보하였다.

Windows 서버는 총 82개 항목(계정관리 18항목, 서비스 관리 22항목, 패치관리 3항목, 로그관리 4항목, 보안 관리 20항목, DB관리 1항목)을 자동으로 진단할 수 있도록 Batch File을 이용해 스크립트를 제작하였고, Linux 서버 및 Unix 서버는 총 73개 항목(계정관리 15항목, 파일 및 디렉터리관리 20항목, 서비스 관리 35항목, 패치 관리 1항목, 로그 관리 2항목)을 자동으로 진단할 수 있도록 Shell script를 이용하여 스크립트를 제작하였다.



<Windows 서버 Batch File>

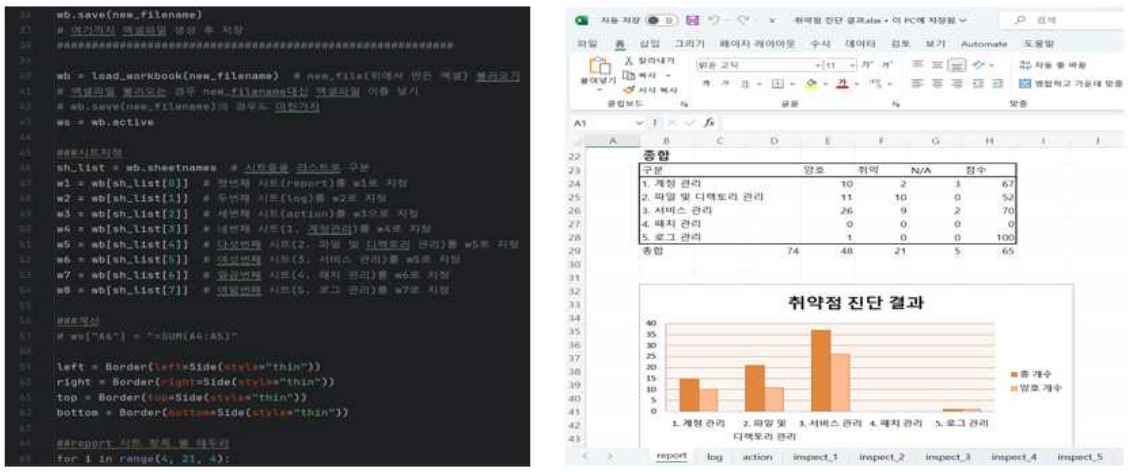
<Linux, Unix 서버 Shell Script>



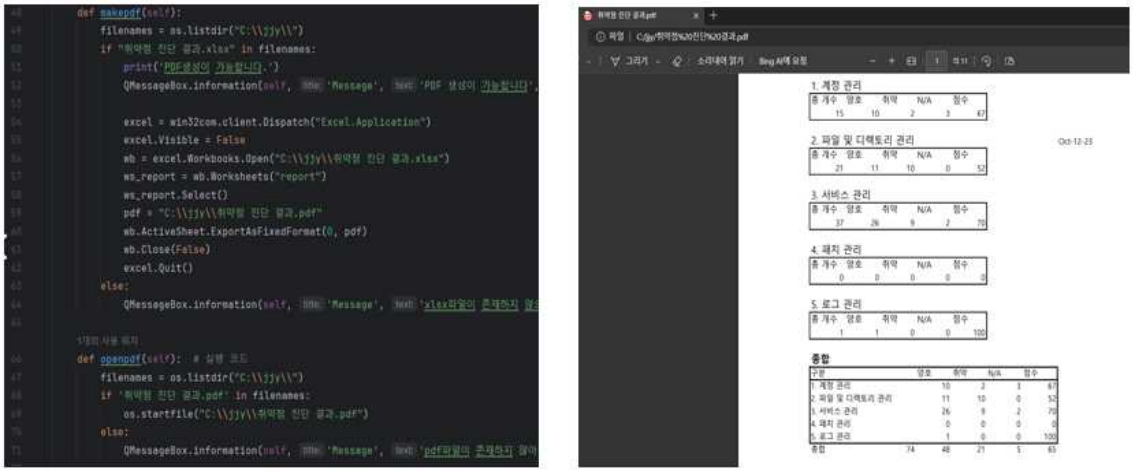
<스크립트 실행 화면 및 실행 후 생성된 진단결과 파일>

3.2.2 진단 결과 보고서 자동 작성 프로그램(Python)

사용자의 서버에 원격으로 접속하여 사용자 서버에 대한 취약점 분석·평가를 진단 스크립트를 통해 자동으로 진단하고, 진단된 결과 파일(.TXT)을 자동으로 로드하여 엑셀 파일 및 보고서 형식의 PDF 파일로 변환 및 출력하여 사용자가 진단 결과를 한눈에 확인할 수 있도록 Python 프로그래밍을 통해 구현하였다.



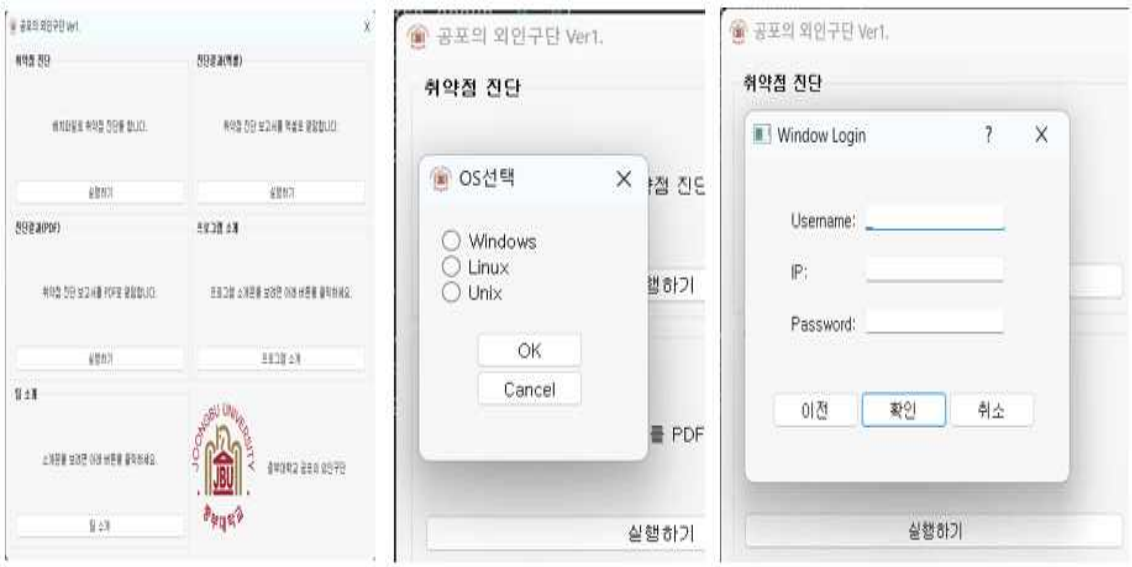
<진단 결과 보고서 엑셀 파일 자동 변환 코드소스 및 자동 생성된 엑셀 파일>



<진단 결과 보고서 PDF 파일 자동 변환 코드소스 및 자동 생성된 PDF 파일>

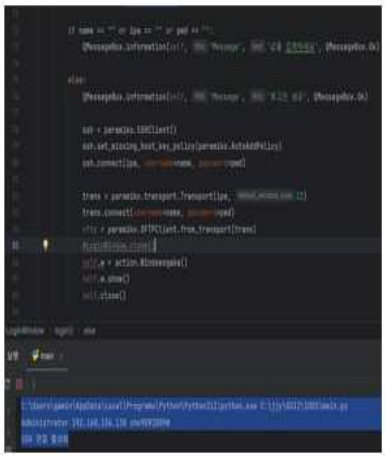
3.2.3 진단 GUI(Graphical user interface)

사용자가 개인 정보를 입력하고 서버 취약점 분석·평가 진행 사항과 진행 결과를 한눈에 볼 수 있도록 Python GUI를 이용하여 인터페이스를 구성하였다. 인터페이스는 사용자가 쉽고 편리하게 서버 취약점 분석·평가 진행 절차에 따라 순차적으로 진행될 수 있도록 ① 사용자 서버 OS 선택 → ② 사용자 ID, Password, IP 입력 → ③ 사용자서버 원격 접속 → ④ 서버 취약점 진단 스크립트 파일 전송 → ⑤ 서버 취약점 진단 프로그램 실행 → ⑥ 진단결과 가져오기 → ⑦ 진단결과 삭제 → ⑧ 진단 결과 엑셀파일 변환 → ⑨ 엑셀파일 실행(보기) → ⑩ PDF 파일 변환 → ⑪ PDF 파일 실행(보기) → ⑫ 종료 순서로 구성 및 구현하였다.



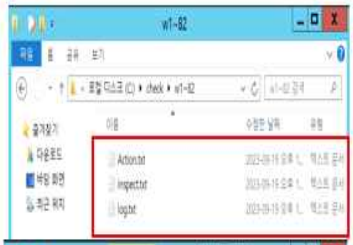
GUI 구성 모습　　서버 OS 선택　　사용자 정보 입력

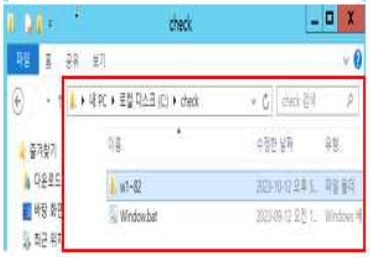
원격 접속 후 실행 화면

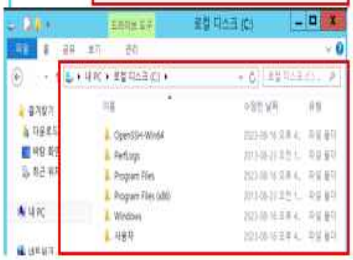


파일 전송 코드

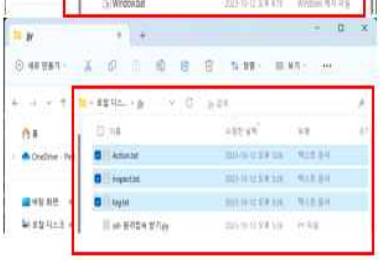
사용자 서버 파일 전송 완료







실행 후 화면(사용자 서버)



파일 추출 후 화면(관리자 서버)

엑셀 파일 생성

PDF 파일 생성

파일 생성 결과

3.2.4 원격 접속 및 파일 전송·삭제

사용자가 서버 취약점 분석·평가 의뢰 시 인력과 시간 소요를 줄이기 위하여 OpenSSH 및 STP 프로토콜을 이용하여 사용자 서버에 원격으로 접속하여 진단 스크립트 파일 전송 및 실행, 진단 완료 후 진단 스크립트를 자동으로 삭제하며 진단 결과를 관리자 서버로 가져올 수 있도록 시스템을 구축하였다.

4. 분 석

4.1 활용 결과 및 성능

서버 취약점 자동 진단 프로그램을 개발하여 VM-ware에 Windows Server 2012 R2, Cent OS 7, Solaris 10 서버를 구축하고, 각 서버에 Test - Server를 구축하여 서버 취약점 자동 진단 프로그램을 Test한 결과, “주요 정보통신 기반시설 기술적 취약점 분석·평가방법 상세가이드”를 기반으로 수동으로 직접 점검을 실시한 평가 결과와 “일치”하는 것으로 나타나 프로그램 성능에 대한 “신뢰성”을 확보할 수 있었다. 또한, 서버 취약점 분석·평가를 수동으로 진행하고 그 결과를 문서로 작성하는데 소요되는 인력(2명) 및 시간(약 2일) 대비 자동화 프로그램으로 서버를 진단하고 그 결과를 보고서 형태의 문서로 제공하는 시간은 약 5~10분으로 수동 점검 대비 많은 인력과 시간 소요를 단축할 수 있는 것으로 확인되었다.

4.2 추후 보완사항

서버 취약점 자동 진단 프로그램 Test 중 진단 스크립트를 통해 서버를 진단 시 일부 항목에서 진단 시간이 과도하게 소모(약 2~3분)되어 진단 스크립트 수행시간이 길어지는 현상을 확인할 수 있었다. 해당 항목들에 대해 면밀히 검토한 결과, 해당 항목들은 점검을 수행해야 하는 세부 내용이 다른 항목에 비해 많아서 그러한 현상이 발생하는 것을 확인할 수 있었고, 추후에는 진단 시간이 많이 소요되는 항목들을 조금 더 세부적으로 분석하여 점검 내용을 세분화하여 불필요한 점검은 삭제하는 등의 조치를 통해 진단 소요시간을 조금 더 단축시키는 방안을 연구할 계획이다.

5. 결 론

5.1 결 론

현재 가장 대표적으로 사용되고 있는 Windows, Linux, Unix 서버에 대한 취약점 분석·평가를 자동으로 실시하고, 그 결과를 엑셀 및 보고서 형식의 PDF 파일로 변환하여 사용자에게 출력해 주는 자동화 프로그램 개발에 성공하였다. 프로그램 개발 후 Test를 통해 제품의 신뢰성을 확인하였고, 수동 점검대비 얼마나 많은 효율성을 가져갈 수 있는지 성능을 확인할 수 있었다. Test 진행 간 보완 사항을 도출하고 분석함으로써 향후 성능이 보완된 자동화 프로그램들이 지속적으로 개발될 것으로 여겨진다.

5.2 기대효과

서버 취약점 진단 자동화 프로그램 개발로 서버 취약점 분석·평가에 소요되는 인력과 시간 소비를 단축하고, 서버에 잠재되어 있는 취약점을 조기 식별 및 대처할 수 있으며, 식별된 취약점은 사용자가 취약점 보완 가이드를 통해 직접 보완함으로써 사용자의 보안 역량을 강화시킬 수 있을 것으로 기대된다.

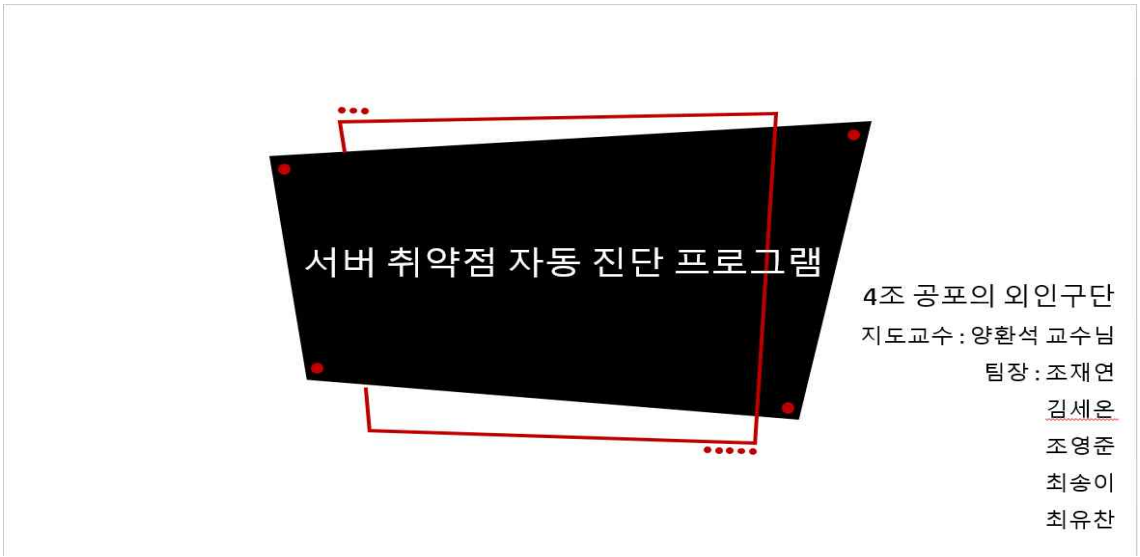
6. 별 첨

6.1 팀원 소개

6.2 소스 코드

github.com/yoochan1012/projectgaka [최유찬]

6.3 발표 자료


서버 취약점 자동 진단 프로그램
4조 공포의 외인구단
지도교수 : 양환석 교수님
팀장 : 조재연
김세온
조영준
최송이
최유찬


CONTENTS
1 프로젝트 개요
• 팀원 소개 및 역할 분담
• 프로젝트 주제 선정이유
2 프로젝트 진행
• 시스템 개발 구상도
• 개발환경
3 결론 / 기대효과
• 결론 / 기대효과


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

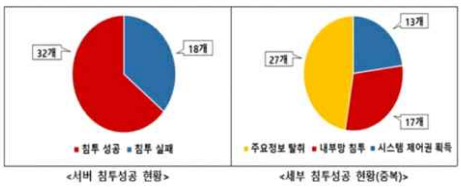
[이벤트] 이제 고민보다 실행할 때! 상품과 함께 데이터바우처 지원 받으세요 (교촌치킨, 로지텍 등)

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 위협을 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 ▲분산서버스거부(DDoS) 공격 및 복구 점검 ▲기업의 홈페이지와 서버 대상 모의 침투를 진행했다.

<모의침투 훈련 결과>



<서버 침투성공 현황> <내부 침투성공 현황(중복)>

※ 주요정보 탈취(27개社) : 기업 정보 외부 전송 가능여부, 개인정보 탈취 가능여부 등 점검
내부망 침투(17개社) : Wi-Fi 패스워드 무력화 등을 통한 내부 네트워크 접속 여부 확인
시스템 제어권 획득(13개社) : 원격실행 취약점 등을 이용한 관리자 권한 획득

※ 일부 기업에서는 3개의 시나리오에서 2개 이상 침투가 가능한 중복 침투

모의 침투 훈련 결과

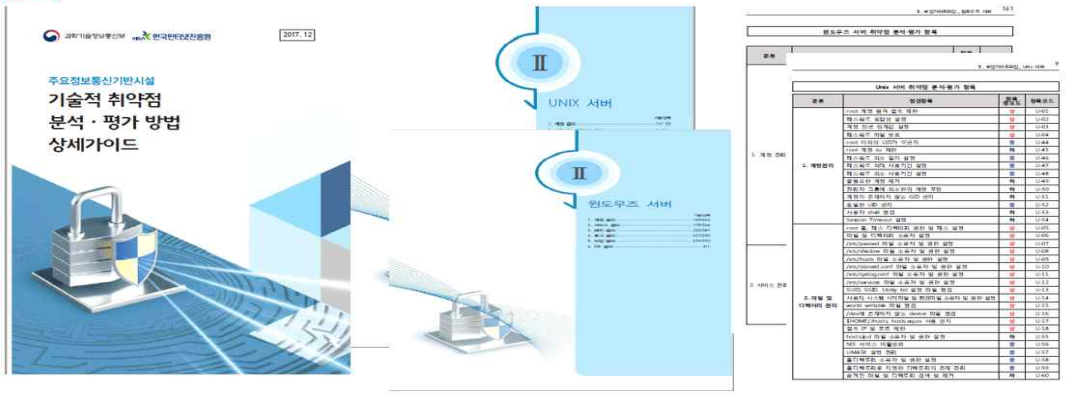
기업 서버 보안 위협의 심각성

1 주제선정이유

자동화 도구를 이용한 효율적인 서버 진단

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

1 주제선정이유



주요정보통신기반시설 취약점 점검 가이드

2 프로젝트 진행

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

2 시스템 구상도

2 개발환경

진단 스크립트

진단 환경

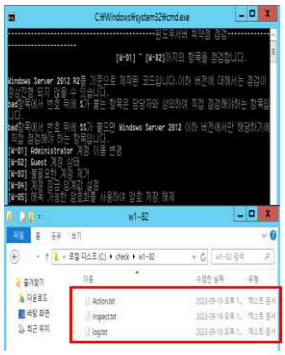
GUI

원격접속

2 프로그램 구성

batch File

Shell Script

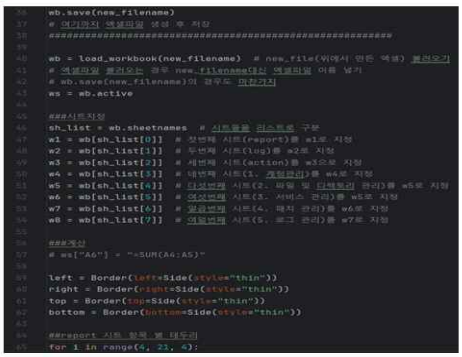


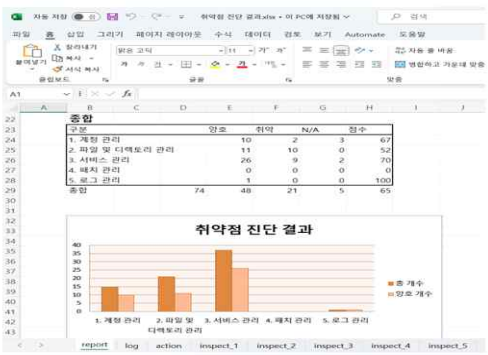
스크립트 실행

진단 결과(.TXT)

서버 취약점 자동진단 스크립트

2 프로그램 구성

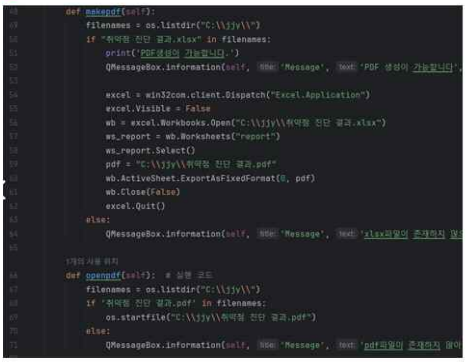


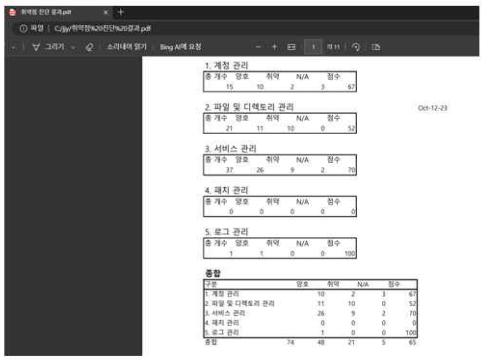


엑셀 파일 자동 생성

진단 결과 보고서 자동 작성 프로그램

2 프로그램 구성





PDF 파일 자동 생성

진단 결과 보고서 자동 작성 프로그램

2 프로그램 구성

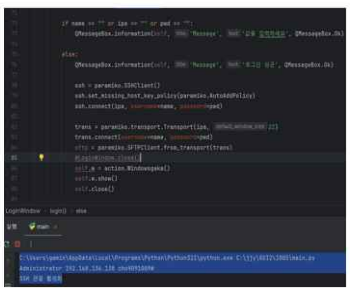
GUI 구성

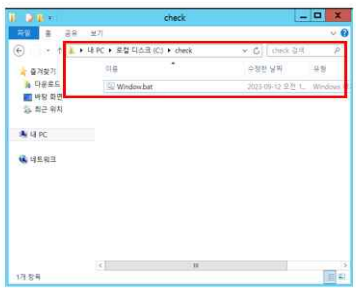
GUI 구성 모습 서버 OS 선택 사용자 정보 입력

서버 OS 선택 / 사용자 정보입력

2 프로그램 구성

GUI 구성



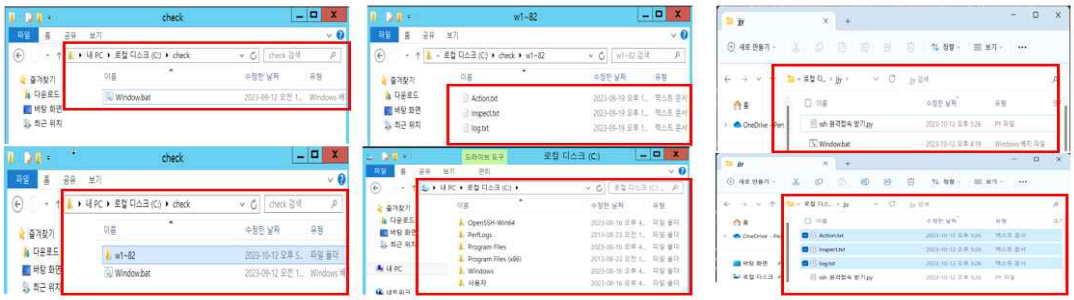


원격 접속 후 실행 화면 파일 전송 코드 사용자 서버 파일 전송 완료

원격접속 / 진단 스크립트 파일 전송

2 프로그램 구성

GUI 구성

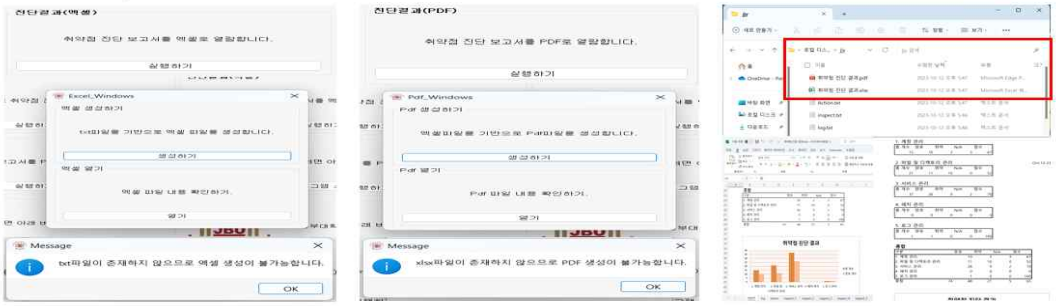


실행 후 화면(사용자 서버)　　파일 추출 후 화면(관리자 서버)

진단 스크립트 파일 실행 / 결과 가져오기 / 진단 파일 삭제

2 프로그램 구성

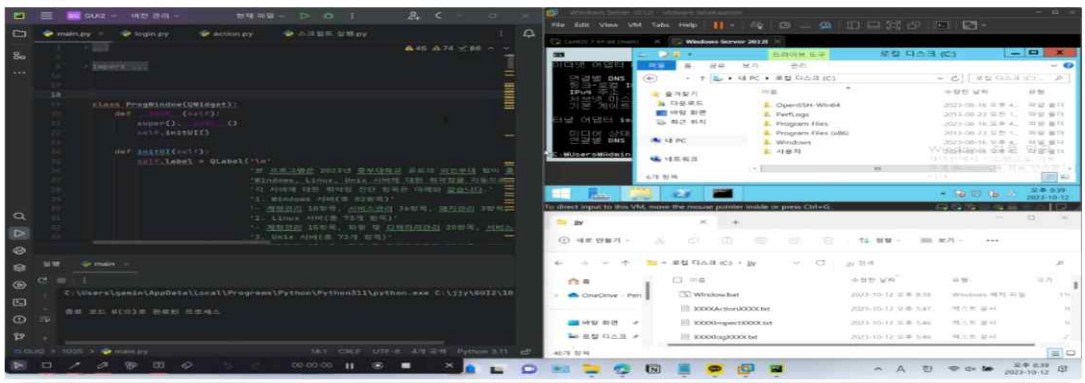
GUI 구성



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

기대효과 프로젝트 기대효과

3 기대효과

프로젝트 기대효과

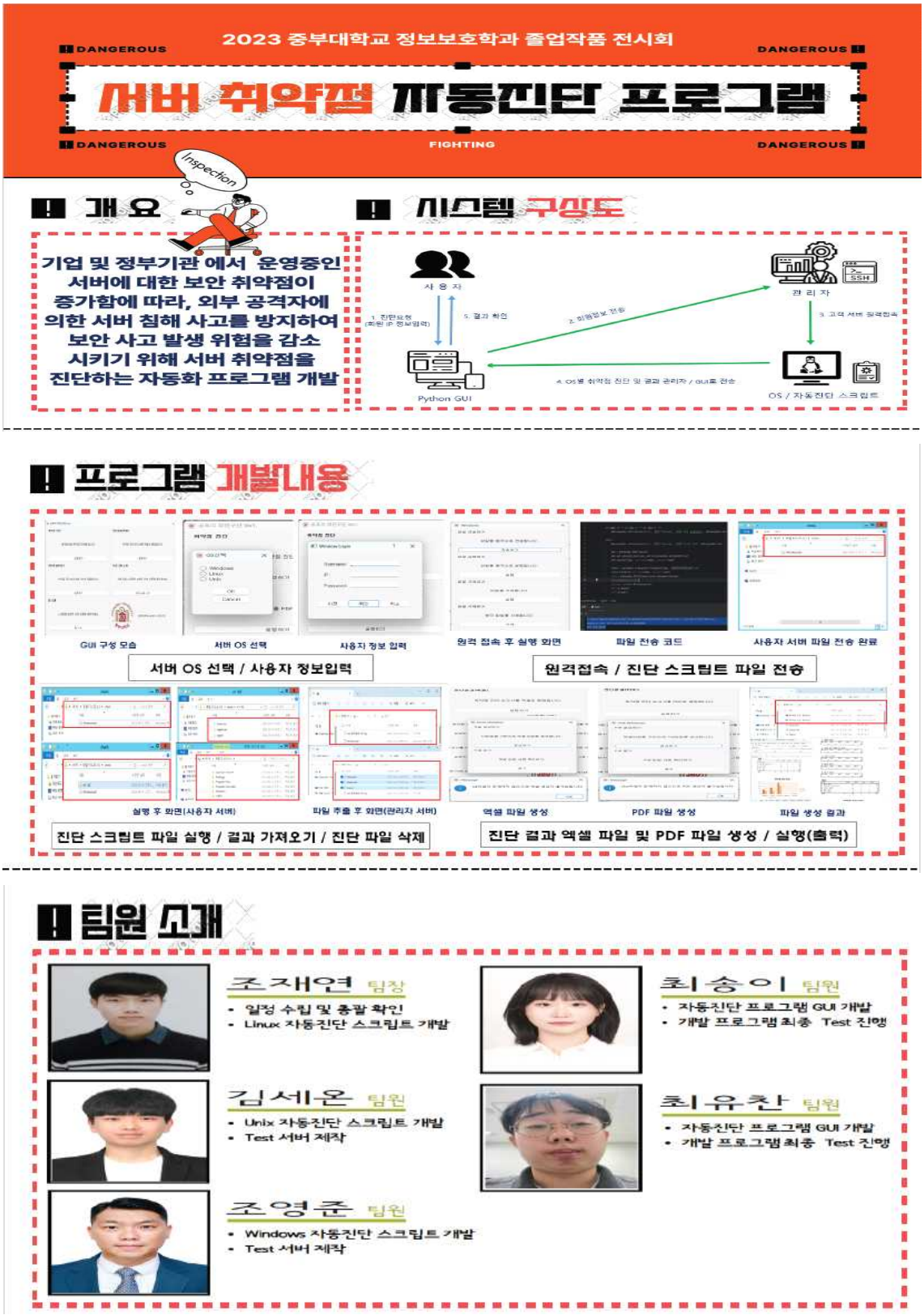
취약점 점검 효율성 증대
자동 진단도구 활용으로
취약점 진단 시간 단축

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

6.4 소개 자료


2023 중부대학교 정보보호학과 졸업작품 전시회
DANGEROUS
DANGEROUS
서버 취약점 자동진단 프로그램
DANGEROUS
FIGHTING
DANGEROUS
Inspection
개요
기업 및 정부기관 에서 운영중인 서버에 대한 보안 취약점이 증가함에 따라, 외부 공격자에 의한 서버 침해 사고를 방지하여 보안 사고 발생 위험을 감소 시키기 위해 서버 취약점을 진단하는 자동화 프로그램 개발
시스템 구성도
사 용 자
관 리 자
Python GUI
OS / 자동진단 스크립트
프로그램 개발내용
서버 OS 선택 / 사용자 정보입력
원격접속 / 진단 스크립트 파일 전송
진단 스크립트 파일 실행 / 결과 가져오기 / 진단 파일 삭제
진단 결과 엑셀 파일 및 PDF 파일 생성 / 실행(출력)
팀원 소개
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