웹으로 공부하는

취약점 마스터

Cyber Guardian

Team.답은 정해져있다

한현동 이정훈 한완섭 박유찬 손진빈

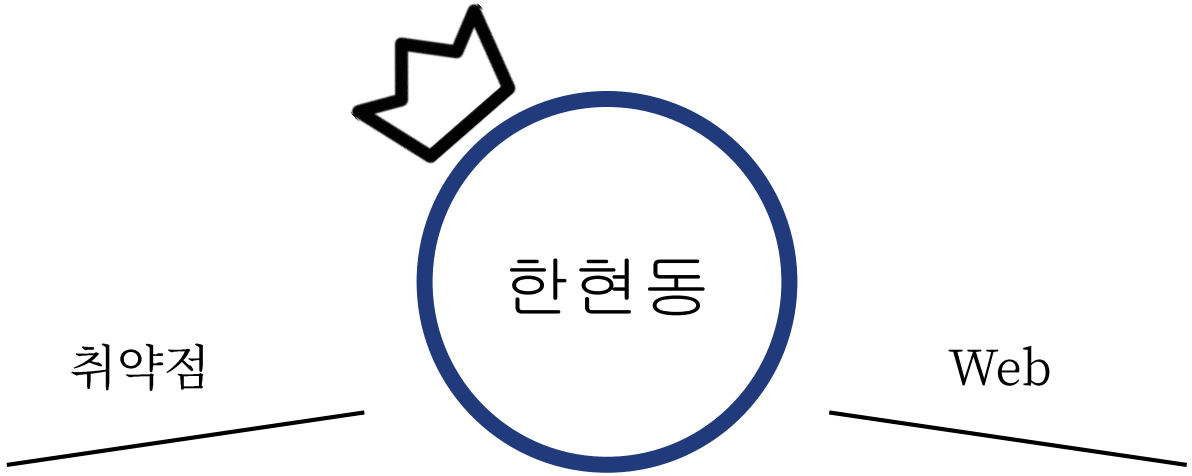
Contents

01. 웹으로 공부하는 취약점 마스터란?

'주요통신기반시설 기술적 취약점 분석 평가 방법 상세가이드'를 기반으로 학습하여

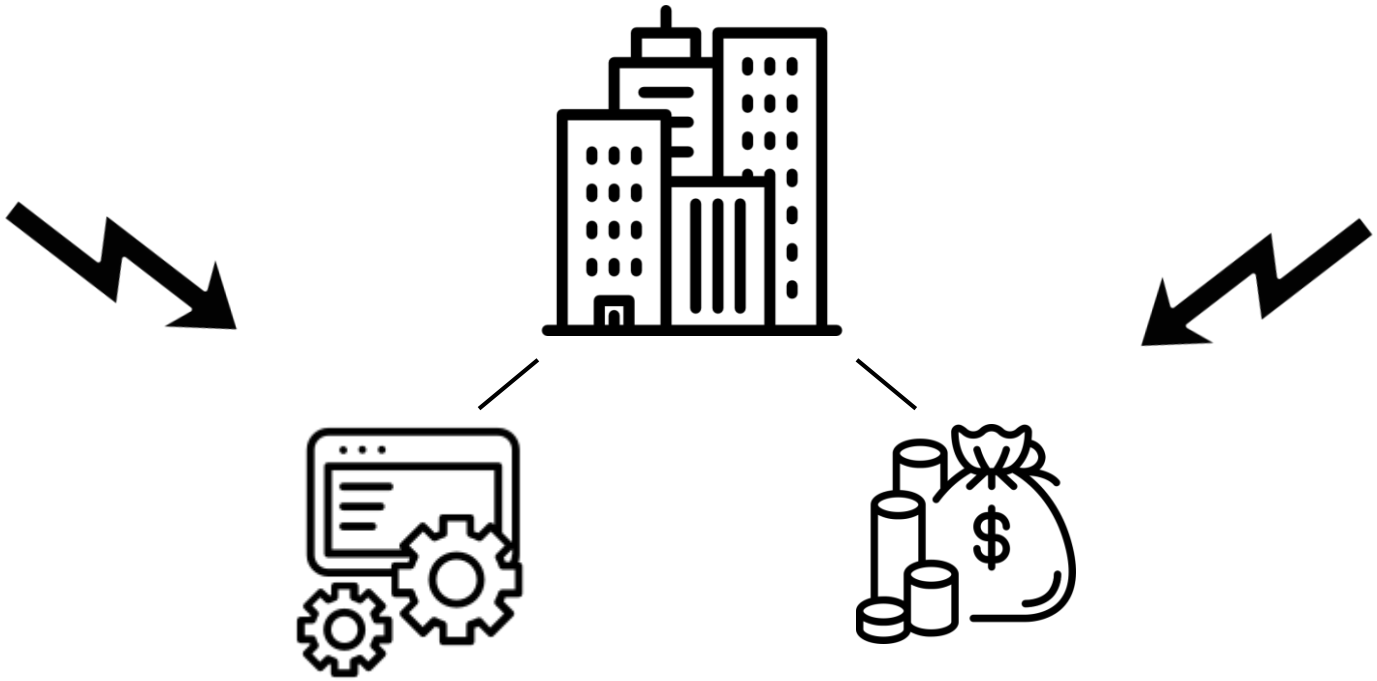
실무에서 원활한 취약점 진단을 할 수 있는 능력을 함양시키는 학습사이트입니다.

01. 역할 분담



이번 프로젝트는 팀장 한현동을 축으로 이정훈, 한완섭은 취약점 팀으로 취약점 커리큘럼 및 문제 작성,
다음은 웹팀으로 손진빈은 Frontend와 Backend 개발을,
박유찬은 Frontend 개발을 맡아 진행되었습니다.

01. 동기 및 기획



기업은 그들만의 유、무형의 자산을 가지고 있고,

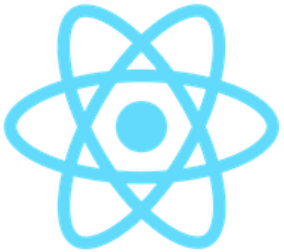
이 자산들은 해킹 및 바이러스의 위험에 노출되어 있습니다.

따라서 대부분의 기업은 '주통기 가이드' 내용을 토대로 정기적인 취약점 점검을 받아야 합니다.

01. 동기 및 기획

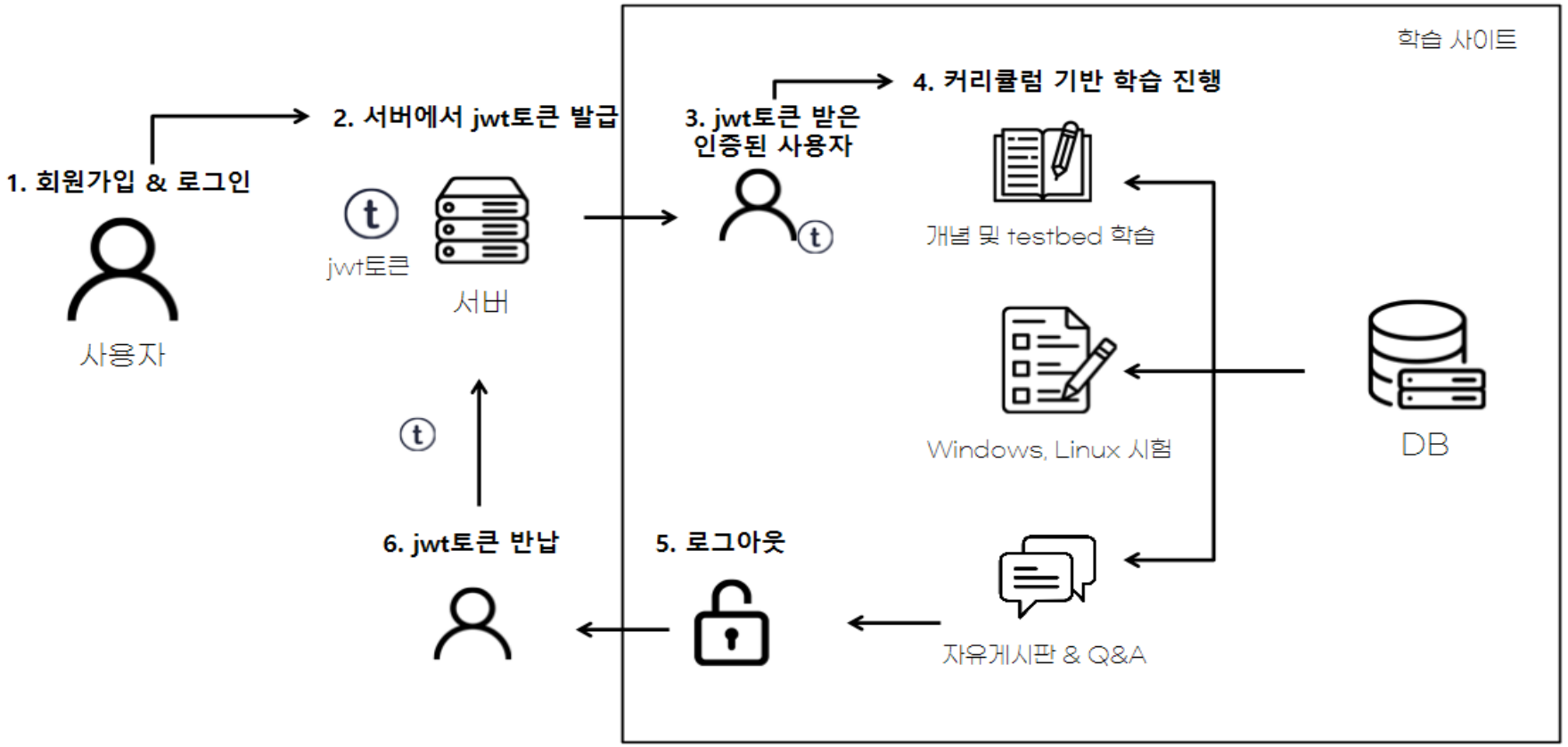
이러한 취약점들에 대해 공부하여 바이러스로 부터 지킬 수 있는 능력을 함양하고자
취약점에 관심이 있는 전공 입문자 또는 비 전공자들을 대상으로 하는
학습사이트를 제작하기로 하였습니다.

01. 기획 [개발환경]



개발 환경으로는 MERN으로 불리는 'MongoDB, Express, React, node js'과
배포한정으로는 frontend는 vercel, backend는 cloud type를
이용하여 프로젝트 개발을 진행하였습니다.

01. 커리큘럼 소개 [구상도]



이용자는 로그인 > 사이트 안내 > 학습 > testbed를 통한 실습 > 시험 > 게시판 이용하여 학습자는 취약점에 대한 이해를 높일 수 있고, 시스템에서 잠재적인 보안 위험을 파악하며 필요한 예방 조치를 취할 수 있는 효과를 기대하고 있습니다.

01. 커리큘럼 소개

Curriculum

- 기초개념

- OS (Windows, Linux)
  양호 및 취약 기준안내
  진단방법
  조치 방법 등

- 배치 파일 스크립트(.bat)

- testbed 실습

Web page

- 이론 및 가이드

- 문제풀이 Quiz

- 취약점 단어장

- 자유게시판, Q&A

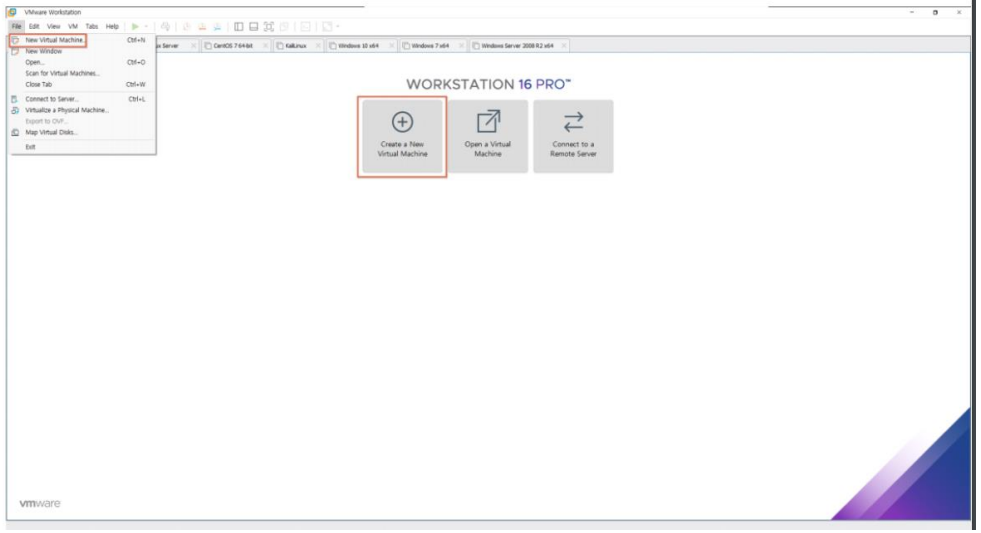
- 자료실

이번 프로젝트의 세부적인 기획 내용입니다.

저희가 직접 작성한 커리큘럼과 그 내용들을 학습할 수 있게 웹 페이지를 구성하였습니다.

02. 진행과정 [기초개념 예시1]

CentOS 7 설치



vi 편집기 사용법

1. 명령 모드 (command mode)

vi 명령어를 통해 vi를 시작할 경우 실행되는 모드. 방향키를 통해 커서를 이동할 수 있다.

2. 입력 모드 (insert mode)

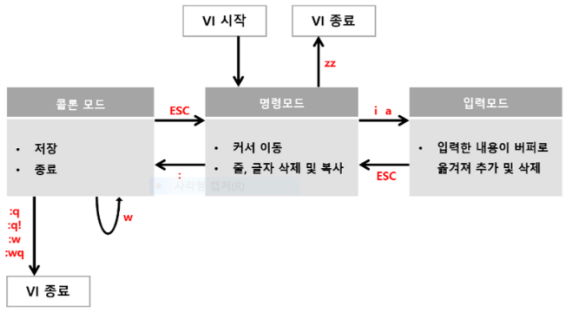
명령 모드에서 `i` 또는 `a` 키를 눌러 입력 모드로 넘어갈 수 있다. 입력 모드에서는 자유롭게 코드나 글을 작성할 수 있으며, 명령 모드로 돌아갈 때에는 `ESC` 를 누르면 된다.

- `i` : 커서가 현재 위치한 부분에서부터 시작
- `a` : 커서 바로 다음 부분부터 시작
- `shift+spacebar` : 영/한 변환

3. 콜론 모드

명령 모드에서 `:` (콜론)을 입력하면 화면 맨 아랫줄에 입력 가능한 공간이 출력된다. 여기서 vi를 종료할 수 있다.

4. vi 구성

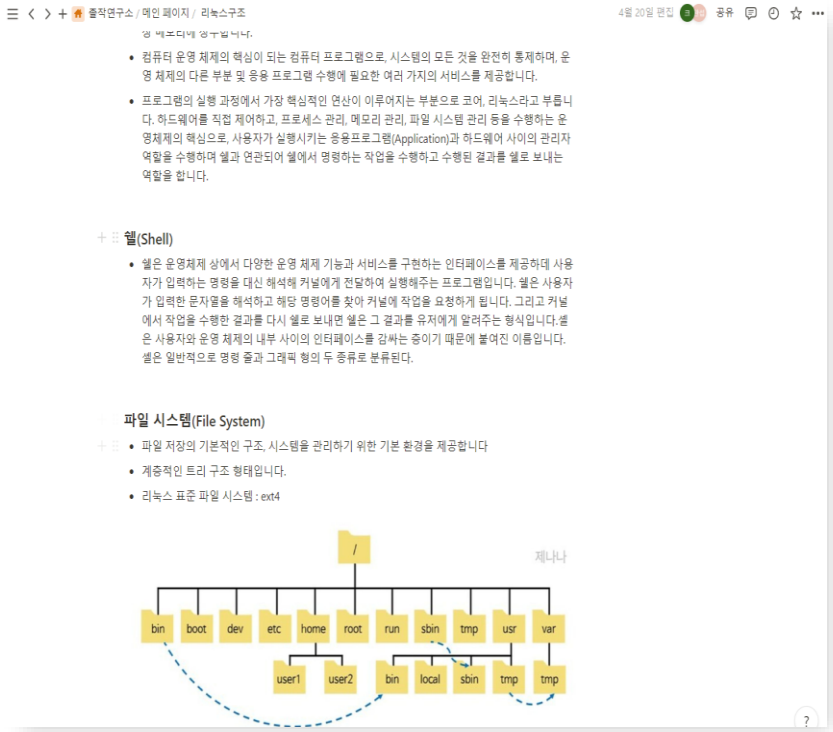


Notion에 정리된 기초개념 내용들 입니다.

좌측 내용은 testbed 실습환경 제작에 대한 이미지이고,

우측 내용은 Linux의 vi 편집기 사용법에 대한 설명입니다.

02. 진행과정 [기초개념 예시2]



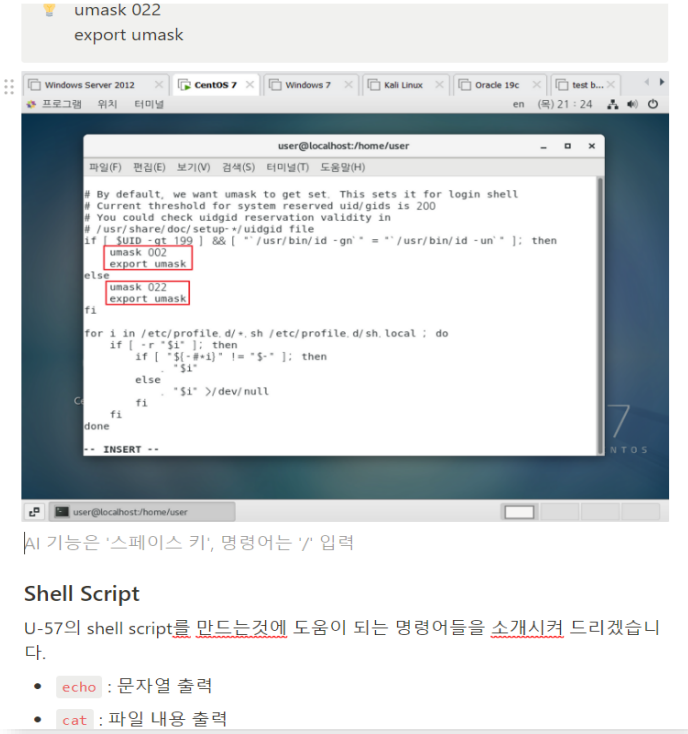
- 컴퓨터 운영 체제의 핵심이 되는 컴퓨터 프로그램으로, 시스템의 모든 것을 완전히 통제하며, 운영 체제의 다른 부분 및 응용 프로그램 수행에 필요한 여러 가지의 서비스를 제공합니다.
- 프로그램의 실행 과정에서 가장 핵심적인 연산이 이루어지는 부분으로 코어, 리눅스라고 부릅니다. 하드웨어를 직접 제어하고, 프로세스 관리, 메모리 관리, 파일 시스템 관리 등을 수행하는 운영체제의 핵심으로, 사용자가 실행시키는 응용프로그램(Application)과 하드웨어 사이의 관리자 역할을 수행하며 쉘과 연관되어 쉘에서 명령하는 작업을 수행하고 수행된 결과를 쉘로 보내는 역할을 합니다.

쉘(Shell)

- 쉘은 운영체제 상에서 다양한 운영 체제 기능과 서비스를 구현하는 인터페이스를 제공하며 사용자가 입력하는 명령을 대신 해석해 커널에게 전달하여 실행해주는 프로그램입니다. 쉘은 사용자가 입력한 문자열을 해석하고 해당 명령어를 찾아 커널에 작업을 요청하게 됩니다. 그리고 커널에서 작업을 수행한 결과를 다시 쉘로 보내면 쉘은 그 결과를 유저에게 알려주는 형식입니다.쉘은 사용자와 운영 체제의 내부 사이의 인터페이스를 감싸는 중이기 때문에 붙여진 이름입니다. 쉘은 일반적으로 명령 줄과 그래픽 형의 두 종류로 분류된다.

파일 시스템(File System)

- 파일 저장의 기본적인 구조, 시스템을 관리하기 위한 기본 환경을 제공합니다
- 계층적인 트리 구조 형태입니다.
- 리눅스 표준 파일 시스템 : ext4



```
umask 022
export umask
```

```
# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
    export umask
else
    umask 022
    export umask
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done
-- INSERT --
```

AI 기능은 '스페이스 키', 명령어는 '/' 입력

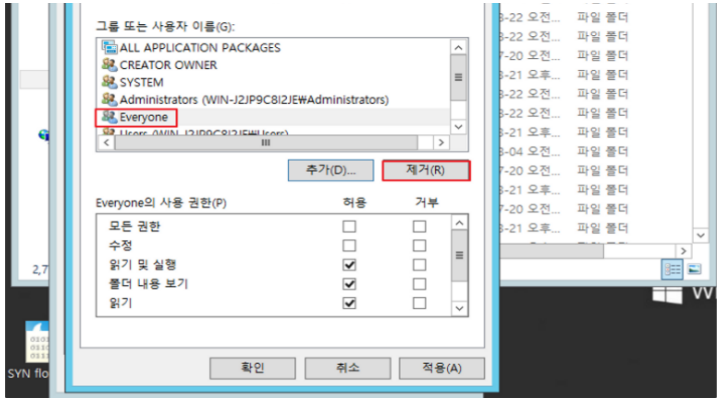
Shell Script

U-57의 shell script를 만드는것에 도움이 되는 명령어들을 소개시켜 드리겠습니다.

- `echo` : 문자열 출력
- `cat` : 파일 내용 출력

좌측 내용은 Linux의 구조(Shell과 File System)에 대한 설명이고,
우측 내용은 Linux의 Shell scripts에 대한 설명입니다.

02. 진행과정 [학습내용 정리]



- 일반적으로 시스템 로그는 C\Winnt\system32\config 파일에 저장되지만, 애플리케이션 로그 파일은 각각의 애플리케이션마다 로그 저장 위치가 다름. 웹 서버에 많이 사용하는 IIS 경우, C:\winnt\system32\LogFiles에 저장됨

Batch Script

W-71 Batch script를 만드는것에 도움이 되는 명령어들을 소개시켜 드리겠습니다.

- `echo` : 메시지를 출력합니다.
- `icacls` : 파일 또는 디렉토리의 접근 권한 정보를 가져옵니다.
- `type` : 파일의 내용을 출력합니다.

```
if [ $CHECK == '#' ] ; then
 echo "[U-31] SMTP 서비스를 사용하며 릴레이 제한이 설정되어 있지 않음 - [취약]" >> U1~73
 echo -e "[U-31] /etc/mail/sendmail.cf 파일에서 #R$* $#error $@ 5.7.1 $: 550 Relaying
 echo -e "[U-31]  vi /etc/mail/access로 생성후  예시) localhost.localdomain    RELAY
 echo -e "[U-31]  파일 제작후 또는 수정후 makemap hash /etc/mail/access.db < /etc/mail

else
 echo -e "[U-31] 스팸 메일 릴레이 제한 설정이 되어있습니다.  - [양호]" >>  U1~73/good/[
  if [ $CH -gt 0 ] ; then
   echo "특정 IP, domain, Email Address 및 네트워크에 대한 sendmail 접근 제한 파일 있음
  else
   echo "특정 IP, domain, Email Address 및 네트워크에 대한 sendmail 접근 제한 파일 없음
   echo -e "[U-31]  vi /etc/mail/access로 생성후  예시) localhost.localdomain    RELA
   echo -e "[U-31]  파일 제작후 또는 수정후 makemap hash /etc/mail/access.db < /etc/ma
  fi
 fi


else
 echo "[U-31] smtp 서비스가 실행중이지 않습니다. - [양호]" >> U1~73/good/[U-31]good.txt
fi
```

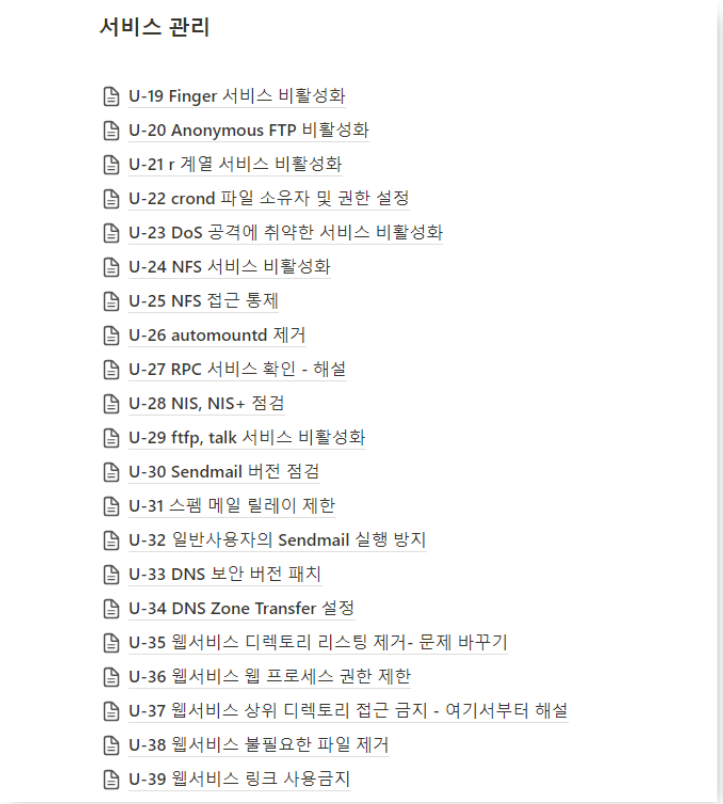
- `echo "[U-31] 스팸 메일 릴레이 제한"` : " [U-31] 스팸 메일 릴레이 제한" 이라는 문구를 출력한다.
- `cat /etc/mail/sendmail.cf > U1~73/log/[U-31]log.txt` : "/etc/mail/sendmail.cf" 파일의 내용을 "U1~73/log/[U-31]log.txt" 파일에 복사한다.
- `cat /etc/mail/access > U1~73/log/[U-31]log2.txt` : "/etc/mail/access" 파일의 내용을 "U1~73/log/[U-31]log2.txt" 파일에 복사한다.
- `CP=$(ps -ef | grep sendmail | grep -v "grep" | wc -l)` : "sendmail" 프로세스가 실행 중인지 확인한다.
- `CH=$(cat /etc/mail/access | wc -l)` : "/etc/mail/access" 파일의 라인 수를 확인한다.
- `CHECK=$(cat /etc/mail/sendmail.cf | grep "R$\\*" | grep "Relaying denied" | cut -c 1)` : "/etc/mail/sendmail.cf" 파일에서 "R$*"과 "Relaying denied"라는 문자열을 찾아 "#"이 있는지 확인한다.
- `if [ $CP -gt 0 ] ; then` : "sendmail" 프로세스가 실행 중인 경우, 아래의 문장을 실행한다.
- `if [ $CHECK == '#' ] ; then` : "#"이 있는지 확인한다.
- `echo "[U-31] SMTP 서비스를 사용하며 릴레이 제한이 설정되어 있지 않음 - [취약]" >> U1~73/bad/[U-31]bad.txt` : "/U1~73/bad/[U-31]bad.txt" 파일에 "[U-31] SMTP 서비스를 사용하며 릴레이 제한이 설정되어 있지 않음 - [취약]" 이라는 문구를 추가한다.

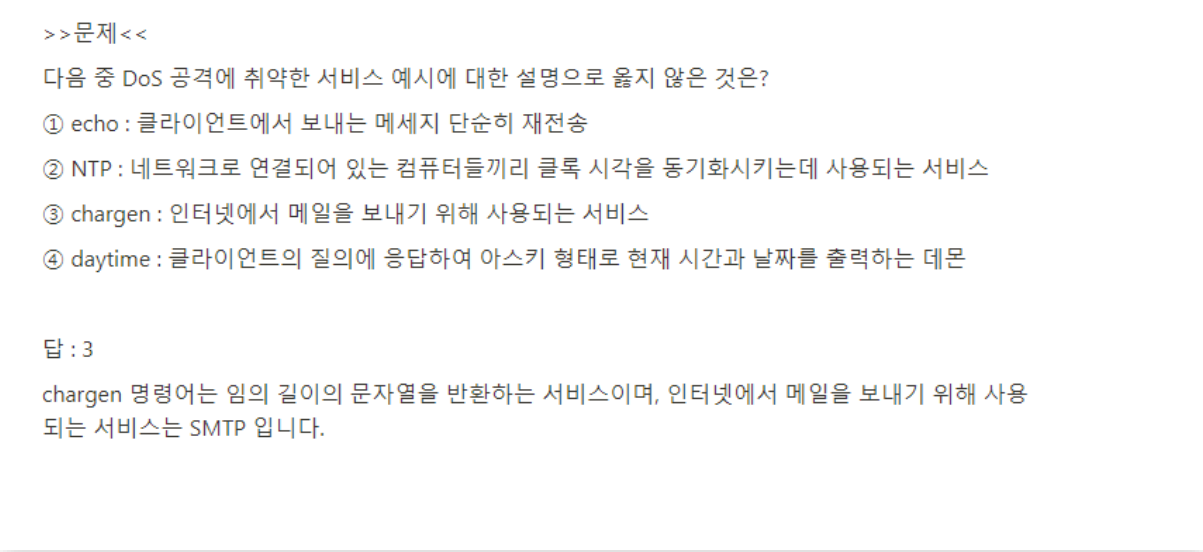
Notion에 정리된 OS 학습내용들 입니다.

좌측 내용은 Windows의 Batch file scripts에 대한 설명이고,

우측 사진은 직접 작성한 리눅스 취약점 진단 스크립트 및 해설입니다.

02. 진행과정 [학습내용 list]

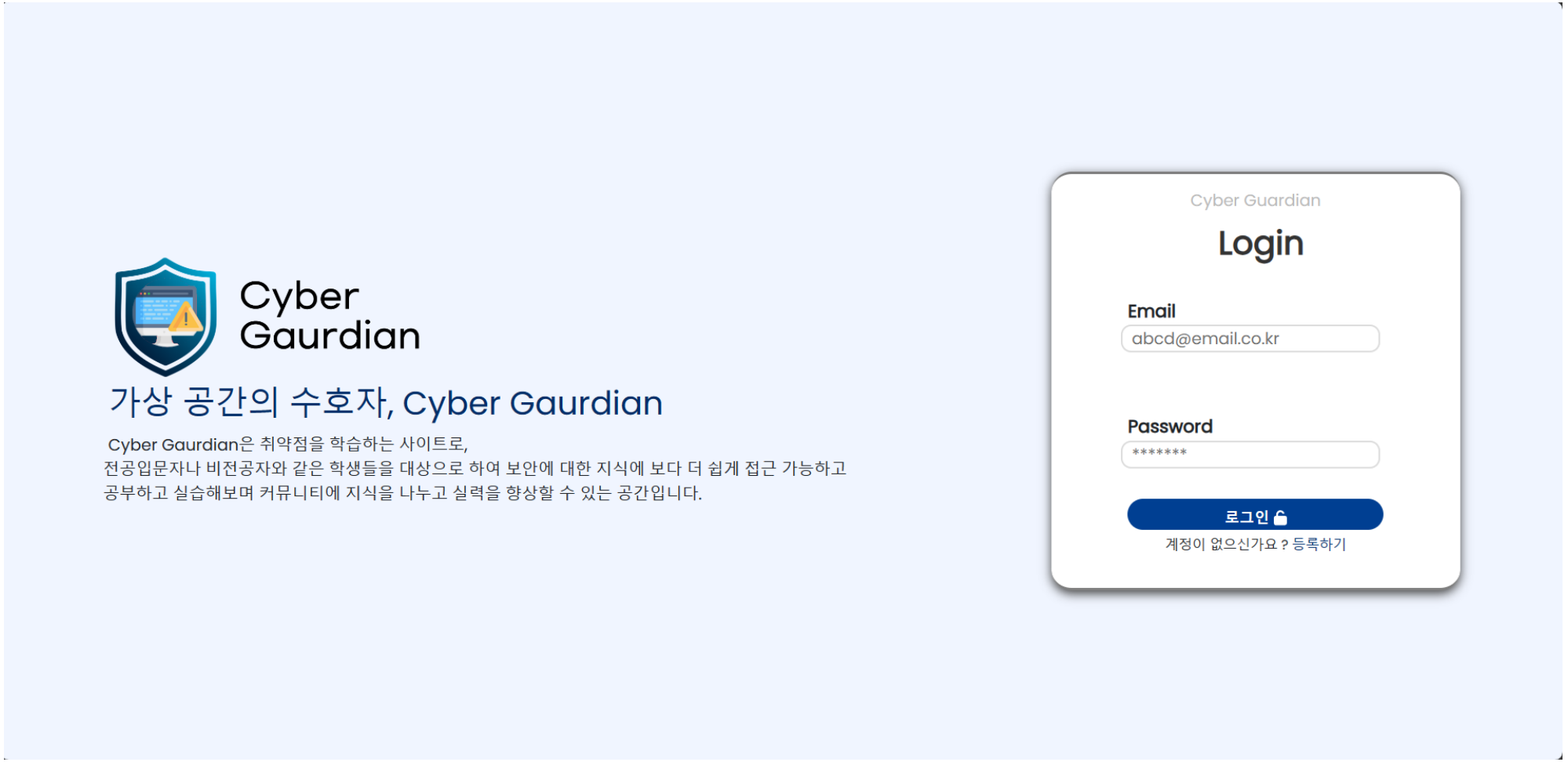




이러한 내용들은 '주통기 가이드'의 내용을 토대로 항목별로 정리 되어있습니다.

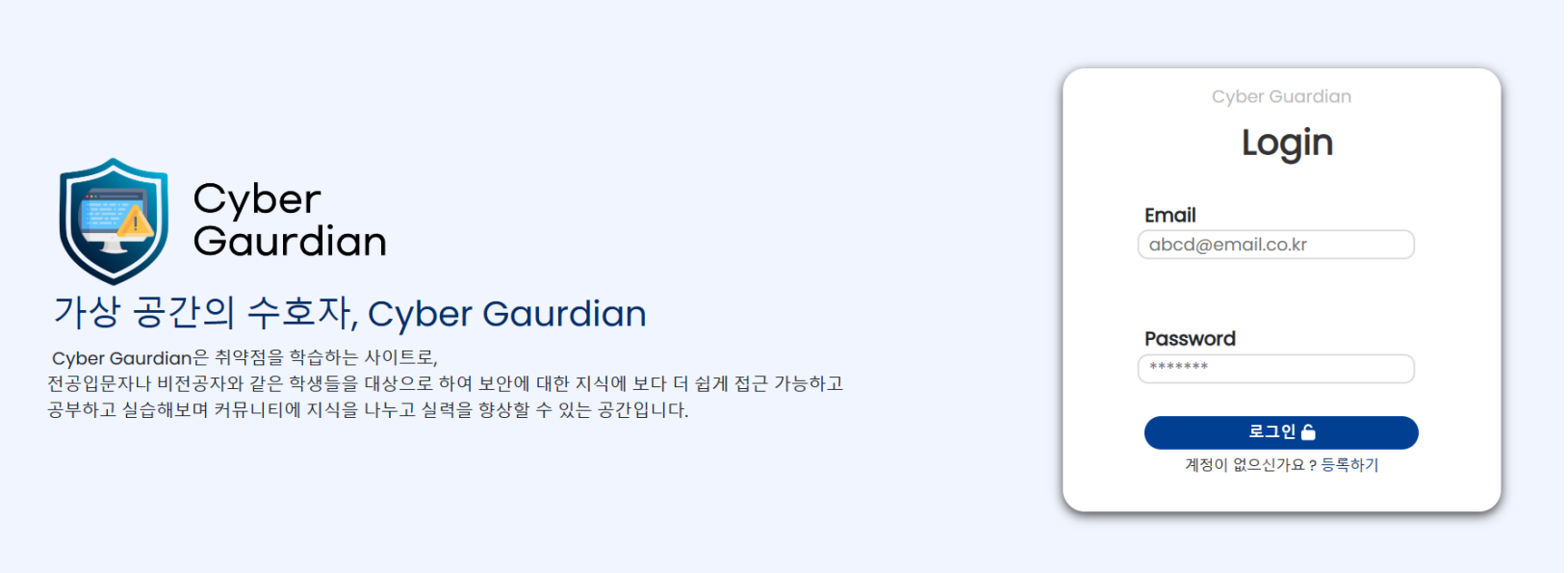
추가적으로 우측 사진과 같이 각 항목별 내용 설명 이후 문제도 포함되어 있습니다.

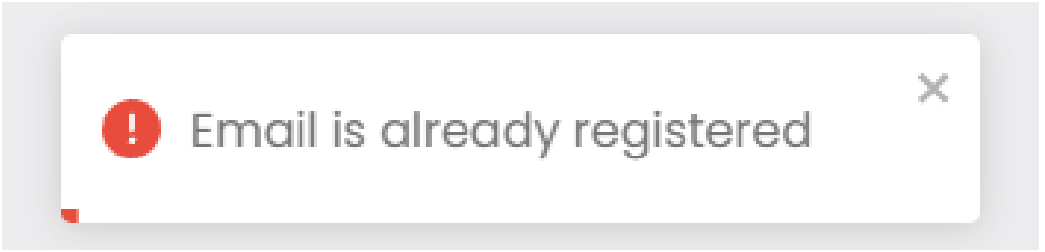
02. 진행과정 소개 [웹사이트 주소]



https://cybergaurdian.vercel.app

02. 진행과정 [Login & Register]

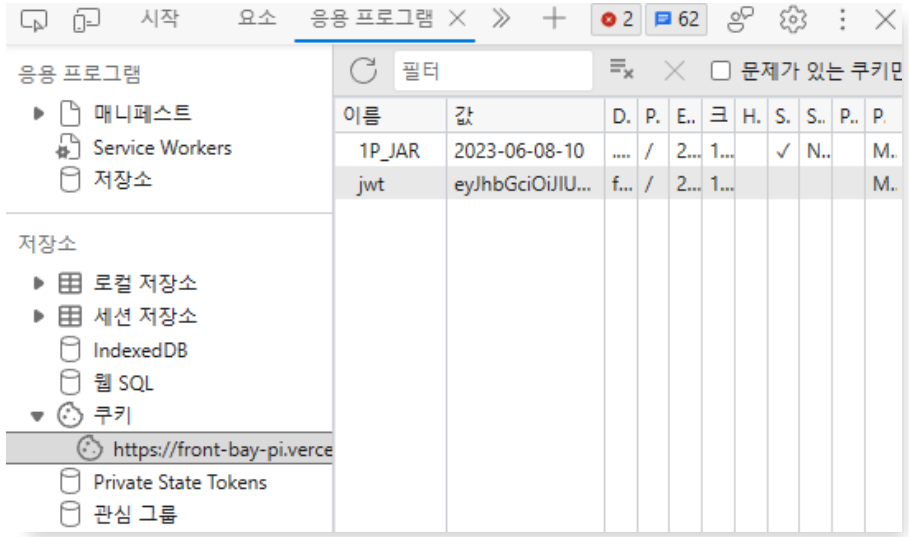


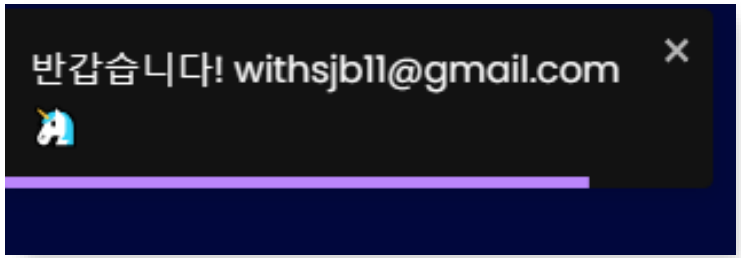


회원가입 페이지에서 Email 형식에 맞게 작성해야 가입이 가능하며

해당 ID 가 이미 있을 때에는 react-toastify 메세지가 우측 하단에 표시됩니다.

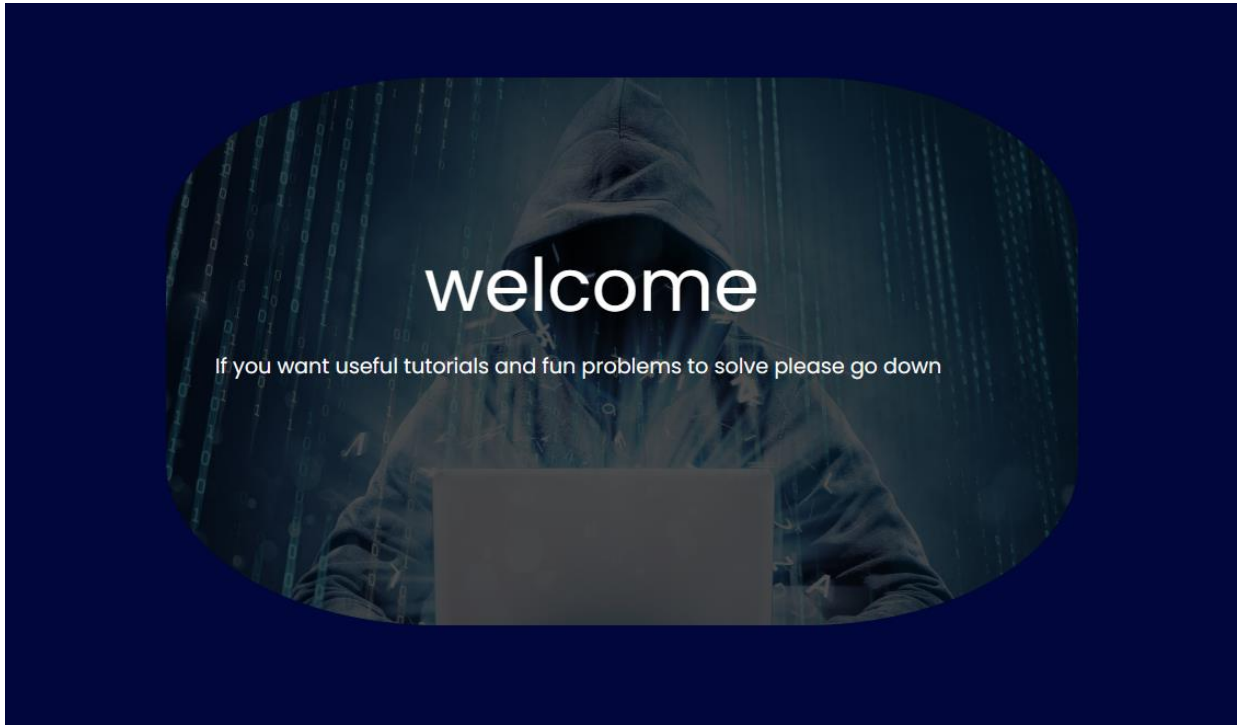
02. 진행과정 [로그인 구현]





로그인 페이지에서 로그인을 완료하게 되면 사용자는 jwt토큰을 통해 인증을 받아 사이트에 접속하게 되고 Password에 대한 정보는 암호화되어 저장됩니다.

02. 진행과정 [Roadmap]



처음 접속했을 때 보여지는 메인 페이지와

로드맵을 통해서 저희 사이트를 이용할 때의 안내를 제공하고 있습니다.

02. 진행과정 [이론 및 가이드]

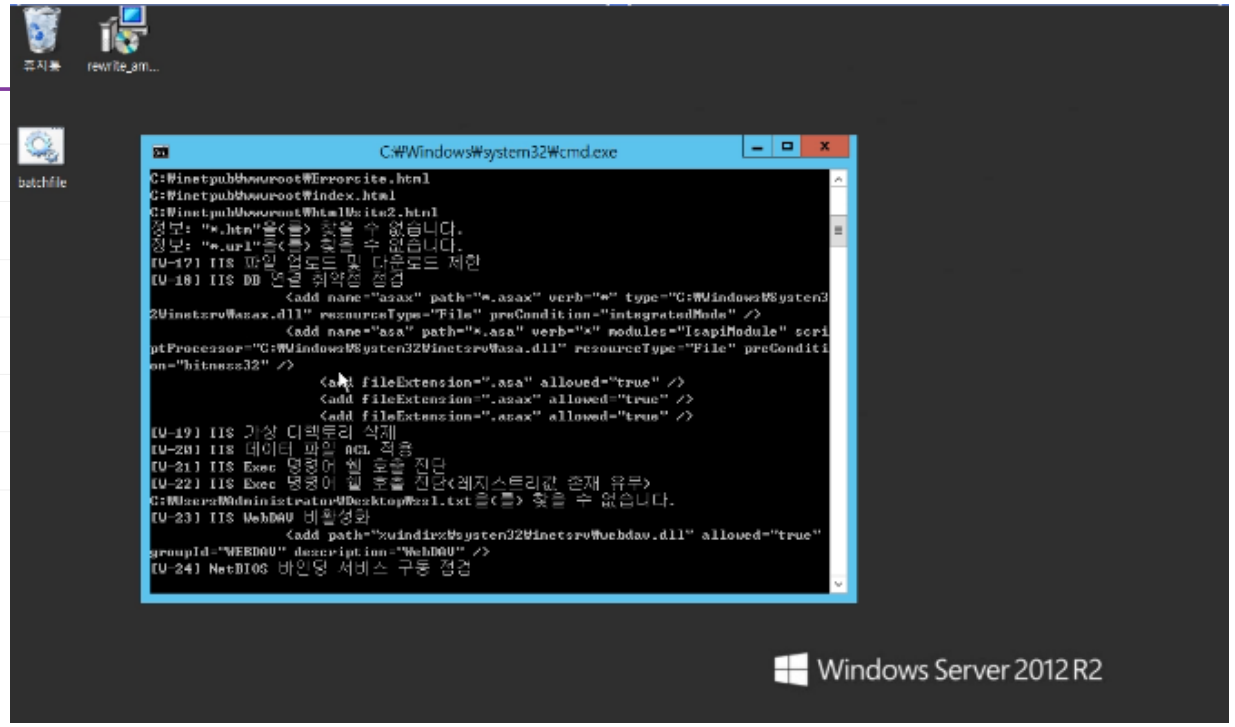
Notion에서 정리한 내용들을 바탕으로 학습할 수 있는 페이지입니다.

각 OS의 취약점들에 대해 공부할 수 있습니다.

02. 진행과정 [Testbed]

Testbed File List

- 리눅스 test bed
- Windows Test bed
- 서비스 관리 1
- 서비스 관리 2
- 서비스 관리 3
- 서비스 관리 4
- 서비스 관리 5
- 파일 및 디렉터리 관리 1



실습을 위한 자료를 다운로드받는 testbed 자료실 입니다.

해당 링크를 클릭하여 testbed를 다운로드 받고

개인 VMware환경에서 실습을 진행 할 수 있습니다.

02. 진행과정 [취약점 단어장]

취약점 단어장 Vulnerability Vocabulary Note.

SFTP

Admin$

로컬 관리자 그룹의 멤버들에게만 접근이 허용되는 폴더로 보통 로그인화면에서 사용자 이름을 입력할 때, 사용자가 선택한 계정으로 로그인할 수 있도록 하는 역할을 합니다.

IPC$
(Inter-
Process
Communication)

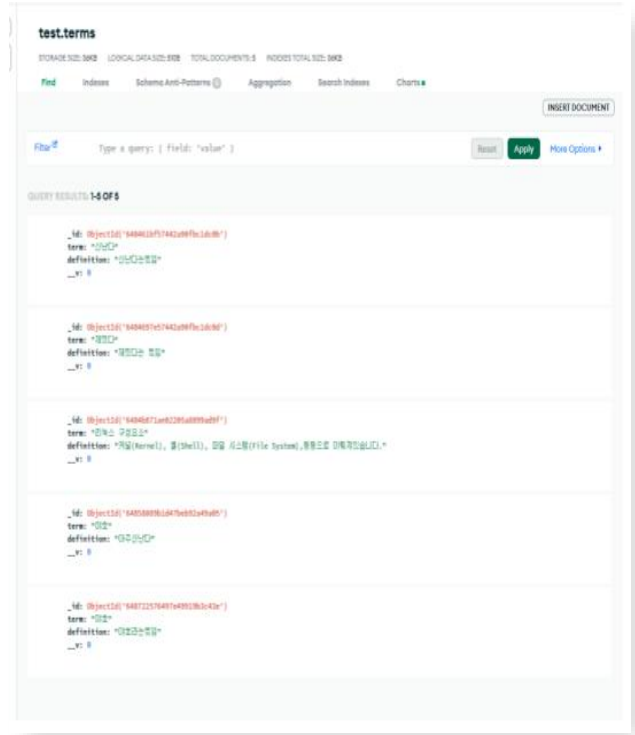
< 이전 다음 >

Add New Term

Term: Input a term

Def: Explain definition

+ Add Term

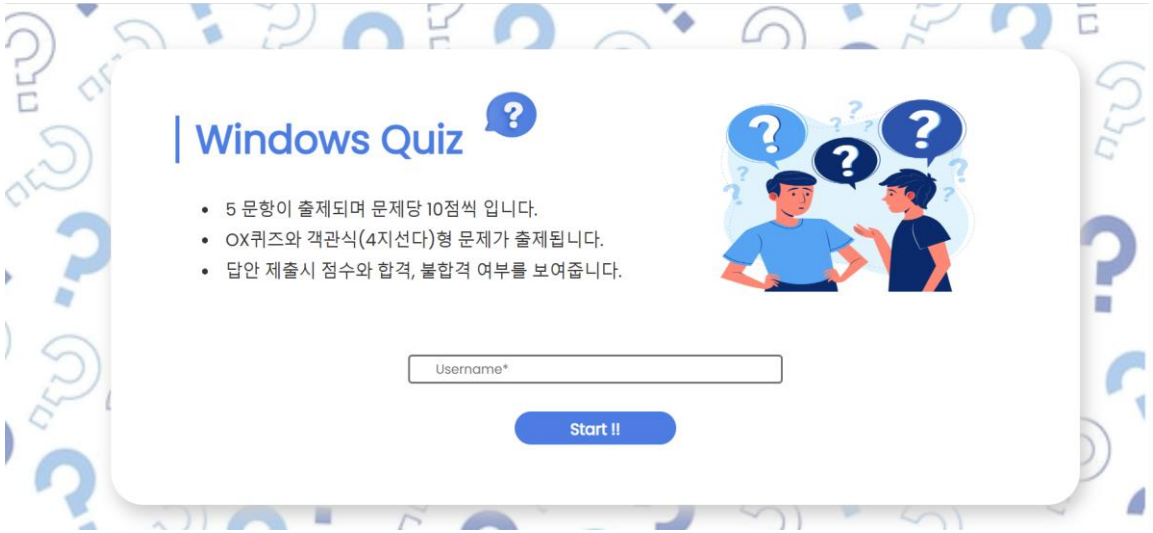


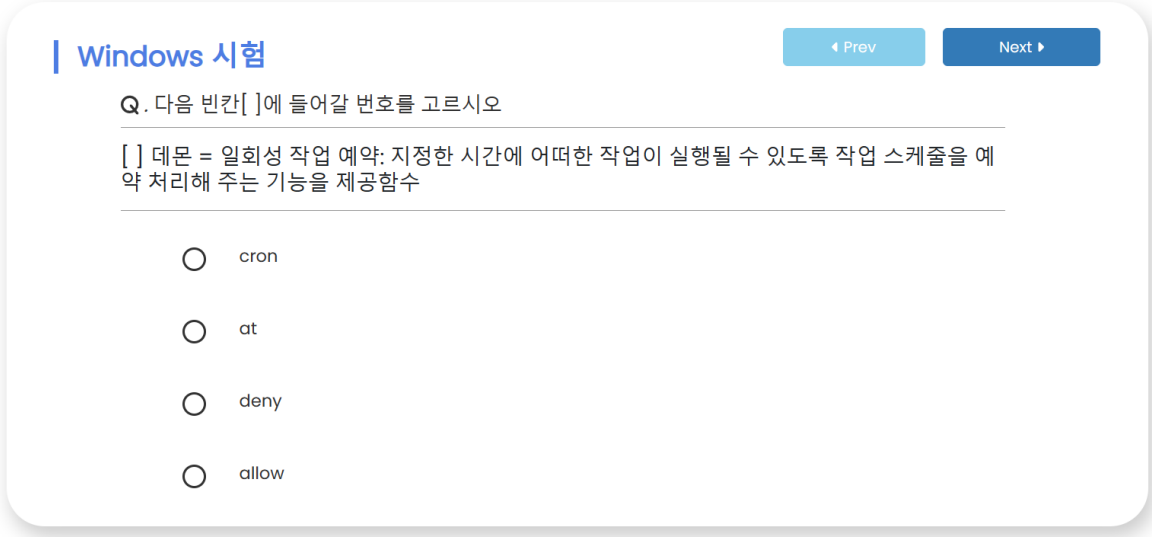
취약점에 대한 내용을 처음 공부를 시작할 때 접하게 되는 용어들이 생소할 수 있기에

'취약점 단어장'을 통하여 용어와 정의를 제공하여 공부에 도움이 되고자 하였습니다.

학습페이지에서는 마우스 호버기능을 통해 모달창이 출력되어 단어장에 등록된 설명이 나오게 됩니다.

02. 진행과정 [Quiz page] - Windows Quiz페이지

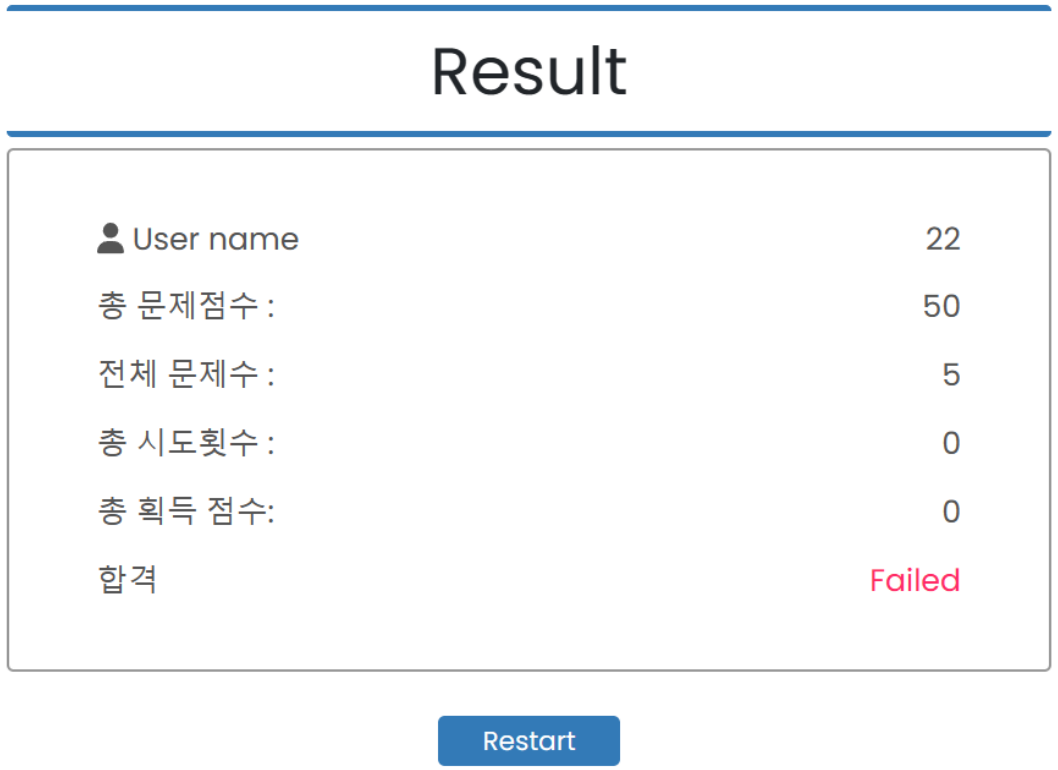




앞서 커리큘럼에서 제시한 내용들을 학습한 후에

퀴즈를 통하여 학습한 내용을 확인해보는 컨텐츠도 추가하였습니다.

02. 진행과정 [결과]



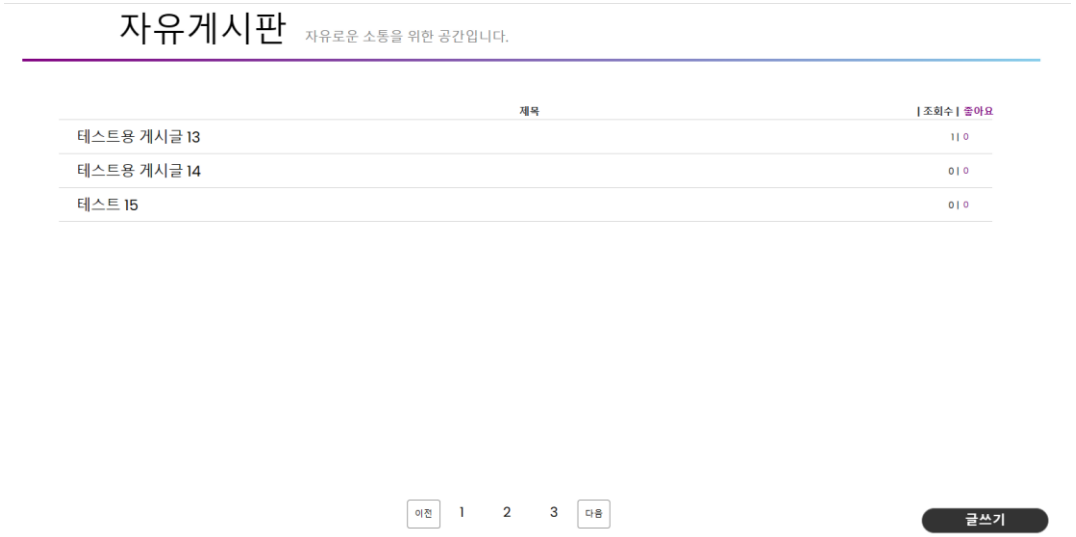
Name	Attemps	Earn Points	Result
kfkf	3	30	Passed
dd	5	20	Failed
dasgasdga	3	20	Passed

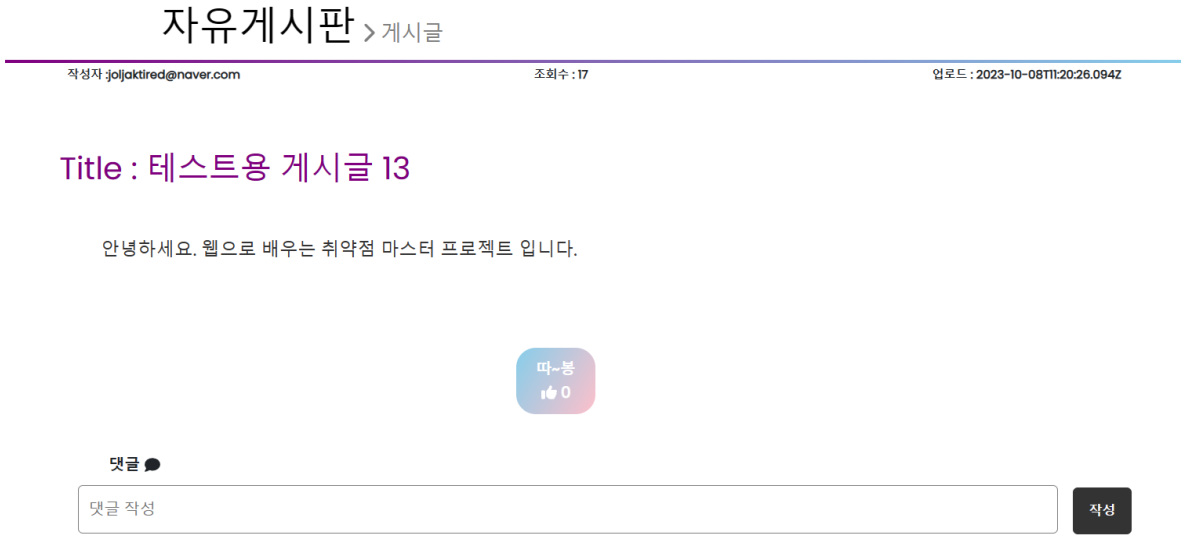
문제를 다 푼 뒤에는 총 문제점수, 전체 문제 수, 시도횟수, 취득점수와 합격 여부가 출력되며,

Restart 버튼을 통해 다시 도전하실 수도 있습니다.

하단에는 학습자들간에 점수와 합격여부를 확인하실 수 있습니다.

02. 진행과정 [자유게시판]





저희 사이트를 이용중인 학습자들 간의 자유로운 소통을 위한 자유게시판 입니다.

작성자명, 작성일, 글 제목과 내용, 추천과 댓글 기능까지 구현하였습니다.

03. 결론 및 기대효과

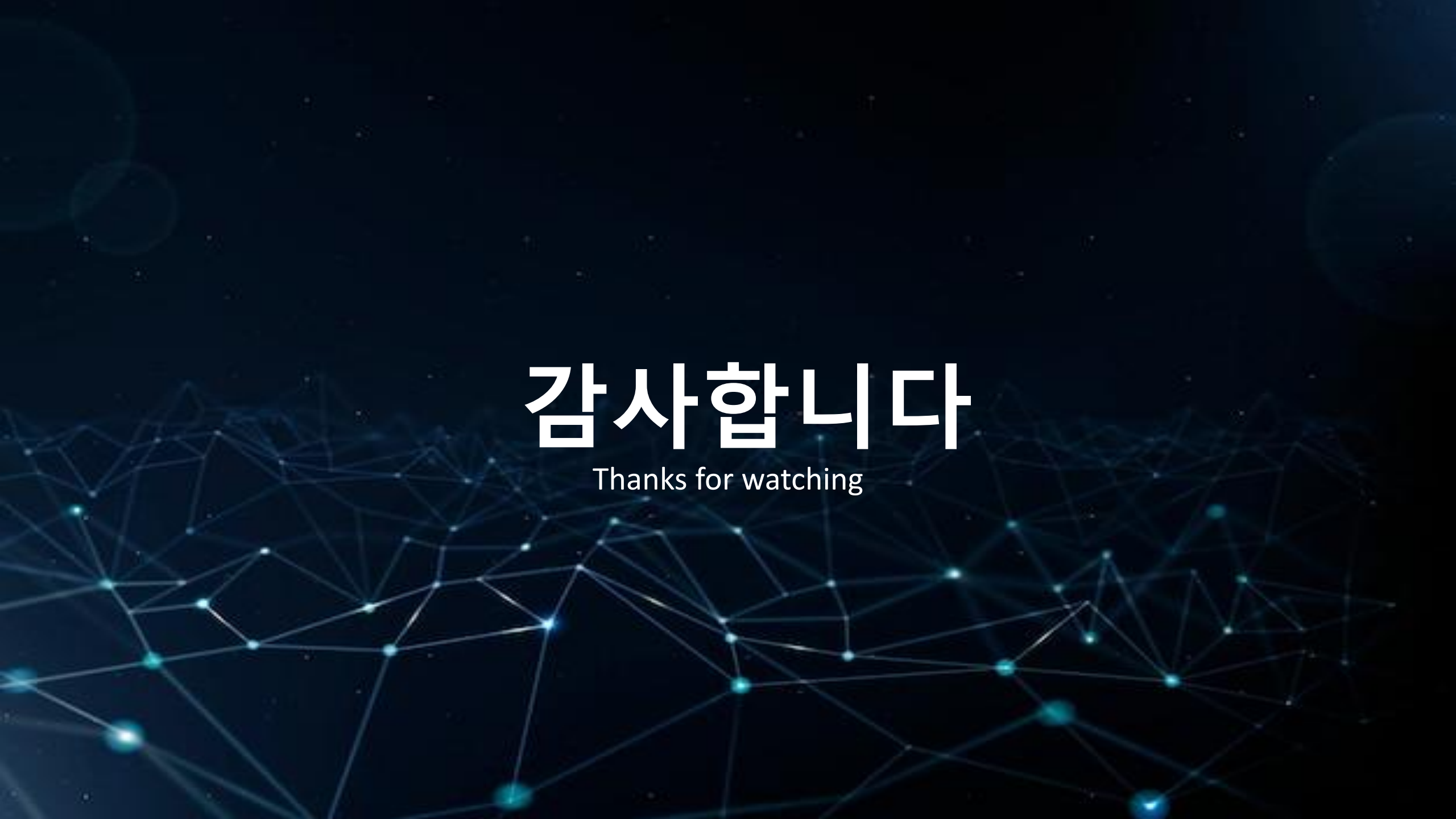
Conclusion

- 기초적인 OS라고 할 수 있는 Windows와 Linux(centOS 7)의 취약점에 대해 차근차근 배워 볼 수 있는 학습사이트를 제작하였다.

Expectation Effectiveness

- 직접 구성한 취약점 커리큘럼에 맞춰 공부를 진행한다면 실무에서 응용해서 취약점에 대해 보완할 수 있을 것이다.

- 취약점에 대해 관심이 있는 초심자들의 눈높이에 맞춰서 제작하였기에 쉽고 부담없이 접근할 수 있어 많은 이용자들이 늘어날 것으로 예상된다.


감사합니다
Thanks for watching