[ 정보시스템 취약점 진단 ]

1주차 - 정보시스템 취약점 진단 방법론 + 리눅스 서버 취약점 진단

2주차 - 윈도우 서버 취약점 진단 + 네트워크 장비 취약점 진단(전상민 강사님)

지란지교 SNC - VADA(취약점 진단 솔루션) 리눅스 에이전트 개발

리눅스 자체 취약점 진단

리눅스에서 구동되어지는 DB, WEB, WAS

이런 자산들을 주통기나, 금융기반이나, 고객사 자사 가이드라인 별로

진단을 할 수 있는 에이전트 개발

정보시스템 종류와 특징, 진단 기준 -> 전체적인 리눅스 서버 수동 진단 방법

-> 쉘의 이해 -> 기본 명령어 사용법(쉘 스크립트 작성 시 주로 사용되는)

-> vim 에디터 사용법 -> 쉘 스크립트 작성 -> 결과 보고서 작성 가이드

-> 완전 자동화쪽 개념 설명 간단히

1. 정보시스템

1) 개인 또는 집단에게 유용한 정보를 제공하는 시스템

- 사람, 장소, 사물에 대한 정보를 가지고 있을 수 잇음

2) 데이터를 입력받아 처리하여 정보를 산출하는 시스템

2. 정보시스템 취약점 진단

1) 컴퓨터 기반 정보시스템의 위협/취약점을 도출 및 영향도 분석을 통해 대책을 수립하는 일련의 과정

2) 현업에서는 인프라 진단이라고도 불림

3. 정보통신망

- 컴퓨터의 이용기술을 활용하여 수집/가공/저장/검색/송수신하는 정보통신체계를 말함

4. 정보통신기반보호법

1) 정보통신기반시설

- 국가에서 주요정보통신기반시설로 지정된 사업자

- 정보통신망을 정보통신기반시설이라고 불리움

\* 금융권 같은 경우는 주요정보통신기반시설 가이드가 아닌 금보원에서 제공하는 금융기반시설 가이드를 따릅니다.

# 정보시스템의 종류와 특징

- 유닉스 서버 : AIX, HP-UX, SunOS(솔라리스) - 오라클에서 iso 파일 지원

- 리눅스 서버 : CentOS, Ubuntu, Debian, RHEL, Fedora, Slackware, SuSE, 붉은 별

Debian 계열 | RedHat 계열

Ubuntu | CentOS, Fedora (무료)

Kali Linux | RHEL(유료)

- 윈도우즈 서버 : windows server 2008, windows server 2008 R2, windows server 2012

- 보안장비

1. 방화벽 : 침입 차단 시스템, ip 또는 port 단위로 접근통제 가능

2. IDS : 침입 탐지 시스템, Rule 기반으로 공격 트래픽을 탐지하는 기능

3. IPS : 침입 방지 시스템, Rule 기반으로 공격 트래픽을 탐지 및 차단하는 기능

4. VPN : 가상사설망, 원격에서 회사 내부 시스템 접근 시 사용되는 보안장비

- 암호환된 접근 키 및 암호화 통신 기능이 적용되어 있음

5. WAF : 웹 방화벽, 웹 서버 보호를 전문적으로 담당하는 침입 차단 전용 장비

6. NAC : 네트워크 접근제어 시스템, PC와 같은 end-point 단말에 대한 보안 기술 적용

- PC 내에서 네트워크 접근 연결 요청 시에 NAC 장비의 정책 준수 여부에 따라 네트워크 연결을 허가 또는 차단하는 장비

7. WIPS : 비인가 무선신호 차단 보안장비

- 네트워크 장비

1. L2 스위치 : 보통 스위치라고 불리우며, MAC 주소를 기반으로 송수신 해주는 장비

2. L3 스위치 : ip 주소를 기반으로 송수신 해주는 장비, 같은 대역 뿐 아니라 다른 망 또는 외부네트워크 연결 목적

- 라우터 : L3 스위치와 마찬가지로 ip 주소로 송수신 해주는 주요 기능 담당, L3보다는 상대적으로 큰 인프라 구축 시에 적합

3. L4 스위치 : 포트 기반 트래픽 제어, 네트워크 스위칭 장비(로드밸런싱)

4. L7 스위치 : 컨텐츠 기반 트래픽 제어, 네트워크 스위칭 장비(로드밸런싱)

\* 로드밸런싱 : 시스템의 가용성 확보 목적, 트래픽 분산기술을 이용하여 목적지 시스템의 부하를 줄여줌

- 제어시스템

1. 국가가 지정한 주요정보통신기반시설 중 특히 산업분야에서 주로 사용하는 정보시스템

- 발전소, 한전, 한수원 등등

2. 제어시스템이란?

- 제어시스템 또는 컨트롤시스템이라고 불리우며, 속도제어, 조절이 필요한 그 밖에 시스템과 연동되어 오류나 버그 또는 해킹에 의해 오작동을 일으켰을 때 제어시스템을 통해 통제 또는 조절하는 역할을 담당

- 데이터베이스 (DBMS)

데이터베이스를 관리하는 매니지먼트 시스템을 대상으로 진단

Sybase, DB2, MariaDB

- 웹 서버를 구성하는 응용프로그램

1. WEB Server

- Apache, IIS, Nginx, IHS(IBM Http Server), OHS(Oracle Http Server)

- 클라이언트가 요청한 URL을 받아서 경로를 찾고 해당하는 파일을 실행시켜서 응답하는 역할

- 주로 정적 컨텐츠 실행이 주요 기능(html, javascript, vbscript)

2. WAS(Web Application Server)

- tomcat, JEUS, Weblogic, Websphere, JBoss

- 주로 동적 컨텐츠 실행이 주요 기능(jsp, php, asp 등)

3. 소스코드 취약점 진단

- 소스코드 상에 보안 취약점을 식별하여 안전한 소스코드를 적용하는 것이 목적

- 웹에서 주로 사용하는 소스코드 진단대상

= html, javascript, jsp, php, asp 등

- 진단방법

= 개발지식이 있는 수행원들의 수동 진단

= 소스코드 진단도구를 활용한 취약점 진단

= KISA 자료실-기술안내가이드-공개SW를 활용한 소프트웨어 개발보안 검증가이드.pdf

- 웹 애플리케이션 취약점 진단

1. 홈페이지 취약점 진단

2. 진단방법

= 입력값 검증 등을 직접 수동으로 확인하는 방법

= 웹 스캐너 등을 활용하여 자동화 도구를 통해 진단하는 방법

1. 진단항목

- 정보시스템의 안정적 운영을 위협하는 요소

# 정보시스템 취약점 진단 및 분석 절차

1. 계획 수립

- 수행 주체에 따른 취약점 진단에 소요되는 시간, 범위, 예산, 산출물 등이 상이

- 수행 주체의 분류

= 자체(기업 내에서 직접 하는) 정보 시스템 자산을 소유하고 있는 기관이나 기업의 내부 직원들로 구성된 자체 전담반을 구성하여 취약점 진단 수행

= 위탁, 보안전문컨설팅 업체를 선정하고, 해당 업체의 기술컨설턴트를 고용하여 TFT(임시프로젝트조직) 취약점 진단 수행

- 산출물 : 수행계획서, WBS(업무분업구조)

2. 진단 대상 식별

- 범위 내의 자산을 식별하는 것이 중요

= 물리적 범위, 서비스 범위

- 고객사의 담당장에게 자산목록 양식을 제공하고 담당자로부터 양식에 식별된 자산목록을 수신함

- 수행원은 받은 식별된 자산 목록들을 유형별로 그룹핑

= 같은 용도의 시스템일 경우 동일한 취약점이 나올 가능성이 높으므로 취약점 진단할 때 효율적

= 마찬가지로 도출된 취약점에 대해 유형별로 대응방안을 제시함으로써 취약점 조치 시에도 효율적

- 식별된 자산목록에 대한 자산의 중요도 평가 실시

= 정보시스템 종류별로 가장 잘 알고 있는 고객사의 운영자 또는 담당자들과의 인터뷰를 통해서 진행

= 자산 중요도 평가 시 기준은 보안 3요소(기밀성, 무결성, 가용성)

3. 취약점 분석

- 고객사 비즈니스 유형에 따른 점검항목표 수립

1) 고객 요구사항이 반영된 점검항목표 수립

= ISMS 인증을 위한 컨설팅 수행에 대한 요구사항

= ISMS 통제 항목이 준용된 체크리스트 선정

- 보통 주요정보통신기반시설 점검항목 + OWASP TOP 10 + 최신 최약점 반영

\* 하나 이상의 점검항목을 합칠 때 중복되는 점검항목을 하나로 통합하는 것이 중요

2) 고객사 유형에 따른 점검항목표 수립

= 고객사의 사업 및 업무 분석서를 통해 대내외 컴플라이언스를 준수할 수 있는 수준의 체크리스트를 준용

- 고객사 내부 점검항목 존재 시 해당 체크리스트를 준용

= 체크리스트에 대한 선정은 1차적으로 분야별 컴플라이언스를 준수하고 고객사 담당자와 협의 하에 선정

= 대표적 취약점 진단 체크리스트 종류

- 주요정보통신기반시설 정보시스템 취약점 점검항목

- 금융 분야 취약점 진단 체크리스트

- 행정안전부 취약점 진단 체크리스트

- 국정원 취약점 진단 체크리스트

3) 기술적 취약점 진단 실시

= 스크립트를 통한 자동 진단(완전한 자동화 X)

= 솔루션을 통한 자동 진단(VADA, SSR - 솔리드 스텝)

= 수동진단 및 운영자 인터뷰를 통한 진단

= 취약점을 도출 / 위협을 맵핑해서 등급 산정

- 대응방안을 수립하고 등급에 따라 조치 우선순위를 결정

= 조치 및 이행점검

- 대응방안에 따른 조치는 고객사에 이행

- 수행원은 고객사에 이행했던 정보시스템을 대상으로 이행점검

4. 취약점 평가

- 대응방안에 따른 조치가 어렵거나 장기적(3년)인 조치인 경우

- 위험평가 실시한 후 위험전략을 수립하고 이행

- 위협과 위험, 취약점의 차이

= 위협 : 취약점을 이용해서 공격하는 행위 그 자체

= 취약점 : 이미 공격자가 악용했던 공격기법이 적용될 수 있는 약점 또는 구조

= 위험 : 취약점과 위협이 결합되었을 때 미치는 영향(파급효과)

**취약점 분석 및 평가란?(악성코드 유포, 해킹 등 사이버 위협에 대한 주요정보통신기반시설의 취약점을 종합적으로 분석 및 평가 개서하는 일련의 과정)**

- 진단항목 :

-> 정보 시스템의 안정적 운영을 위협하는 요소

-> 요소의 항목별 세부 점검 항목을 도출하여 취약점 분석을 실시

- 발견된 취약점에 대한 취약성 등급 부여

-> 위험도 산정, 빠른 대응을 위한 우선순위로 사용

-> 개선방향 수립 등이 유기적 평가 수행

**취약점 분석 절차**

- 정보 시스템 현황 파악 : 시스템 구성, 운영 현황. 네트워크 구성 현황. 응용프로그램 현황 파악

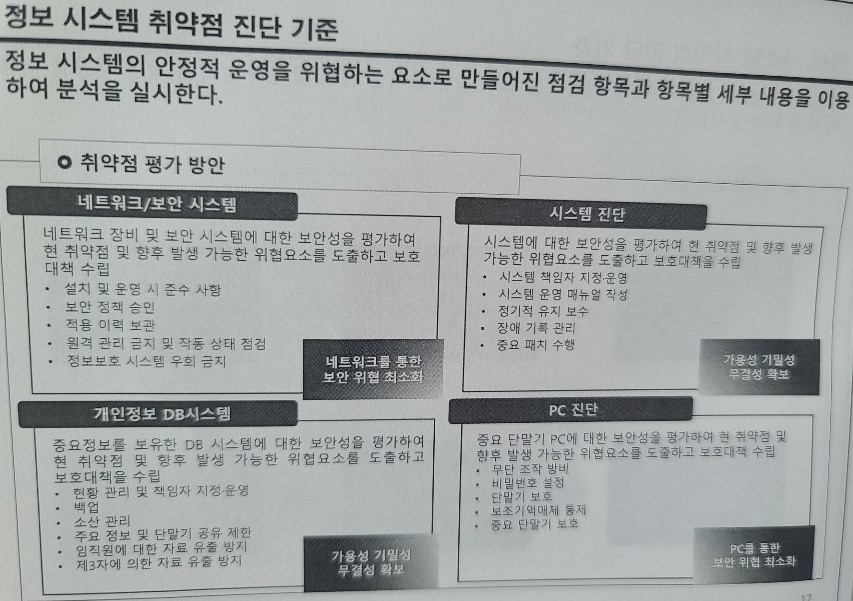
- 점검 대상 선정 : 우선순위 산정. 표본 산출 방식 검토

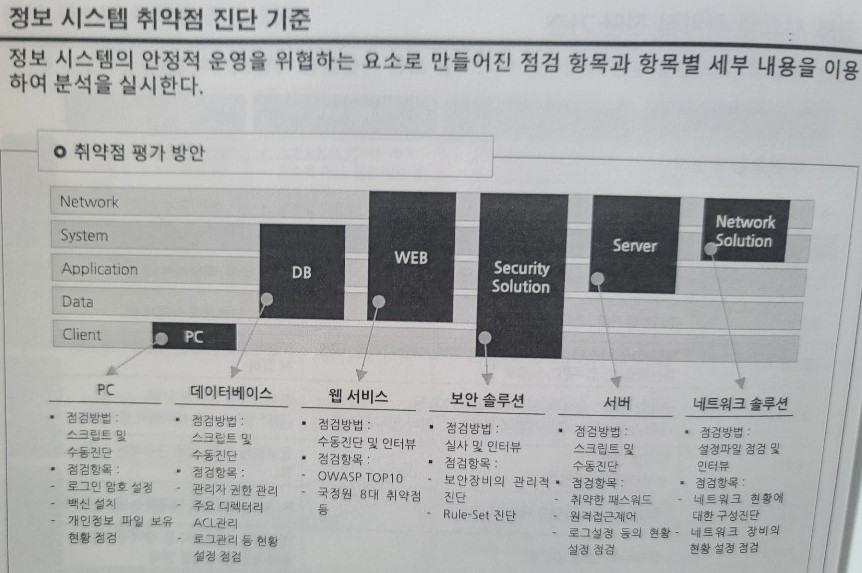
- 취약성 진단 : 도구를 이용한 진단. 보안장비 운영 진단. 점검 스크립트를 이용한 진단

- 진단 결과 분석 : 진단 도구의 결과 보고서. 수동 점검의 결과 보고서

- 보안대책 제시 : 운영 개선책 제시. 취약성 제거 방안 제시

- 이행 점검 : 운영 개선 확인. 취약성 제거 확인





점검내용, 점검목적 읽어보기

왜 하는지 등 당연히 알아야한다.

pam\_securetty.so 모듈이 설정이 되어야

securetty가 실행된다.

서비스 활성화 여부를 먼저 확인 -> 그 다음 설정을 확인하기

만약 원격 터미널 서비스를 사용하지 않으면 설정을 볼 필요가 없다.

telnet 설정 뿐만아니라 ssh설정도 같이 확인

telnet 관련 설정

= 반드시 /etc/pam.d/remote 파일 안에 pam\_securetty.so 설정이 있고

/etc/securetty 파일에 pts/ 관련 설정이 없어야 양호 또는 telnet 서비스를 사용하지 않을 시 양호

취약시 고객에게

pwconv를 사용하여 shadow 파일을 사용하도록 말해야한다.

실습파일 계정 : kisec 비밀번호 : kisec123

**# root 계정 원격 접속 제한**

- 점검내용 : 시스템 정책에 root 계정의 원격 터미널 접속 차단 설정이 되어 있는지 점검

- 점검 목적 : root계정 원격 접속 차단 설정 여부를 점검하여 외부 비 인가자의 root계정 접근 시도를 원천적으로 차단하는지 확인하기 위함

**판단기준**

- 양호 : 원격 터미널 서비스를 사용하지 않거나, 사용 시 root 직접 접속을 차단한 경우

- 취약 : 원격 터미널 서비스 사용 시 root 직접 접속을 허용한 경우

- 서비스 활성화 여부 확인 후 설정 확인

- 서비스가 비활성화 되어있으면 설정을 볼 필요가 없음

- telnet 설정 뿐만 아니라 ssh 관련 설정도 같이 확인을 하는 것이 좋음

- telnet 관련 설정

= 반드시 /etc/pam.d/remote 파일 안에 pam\_securetty.so 설정이 있고 /etc/securetty 파일에 pts/ 관련 설정이 없어야 양호 또는 telnet 서비스를 사용하지 않을 시 양호

- ssh 관련 설정

= ssh 서비스를 사용 중이지 않거나 /etc/ssh/sshd\_config 파일 안에 PermitRootLogin 이 설정이 no 로 되어있으면 또는 PermitRootLogin 설정이 주석처리 되어있어도 양호

= ssh 서비스를 사용 중이고 PermitRootLogin yes로 되어있으면 취약

- 실습 명령어

0. sudo -s

- root 계정으로 전환 명령어

1. ssh 설정 확인

1) vi /etc/ssh/sshd\_config 파일 열기

2) PermitRootLogin 설정 확인

2. telnet 설정 확인

1) yum install -y xinetd

2) yum install -y telnet-server telnet

3) service iptables status, service firewalld status

3-1) 만약 해당 방화벽이 실행 중이면 service iptables stop 이런식으로 종료

4) vi /etc/xinetd.d/telnet

4-1) disable = yes 를 disable = no 로 변경

5) vi /etc/pam.d/remote

5-1) pam\_securetty.so 모듈 설정 확인

6) vi /etc/securetty

6-1) 내부에 pts 설정 확인

7) telnet 실행을 위하여 service xinetd restart

8) telnet localhost 실행 후 root 계정으로 로그인 시도(로그인 실패)

9) /etc/securetty 파일에 pts/0, pts/1, pts/2 설정 추가

10) service xinetd restart

11) telnet localhost 실행 후 root 계정으로 로그인 시도(로그인 성공)

3. 각 서비스 실행 여부 확인

1) 프로세스 목록 확인 : ps -ef | grep telnet or ps -ef | grep ssh

2) 포트 활성화 여부 확인 : netstat -ano | grep :23, netstat -ano | grep :22

- 해당 서비스의 포트 번호 확인

= cat /etc/services | grep "서비스 이름"0

3) 서비스 실행 여부 확인

- 서비스 명령어를 통한 확인 : service ssh status

- xinetd 구동되어지는 서비스 확인

= vi /etc/xinetd.d/"서비스 이름" 해당 파일 안에 disable 설정을 확인

\* yes 면 비활성화, no 면 활성화

\* 최신버전 : CentOS 7 이상, Ubuntu 16.04 이상

initd -> systemd 최상의 프로세스가 변경되어지면서 리눅스 구조가 바뀜

명령어 변경

netstat -> ss

service -> systemctl

ifconfig -> ip addr

근데 만약에 netstat 나 ifconfig 같은 네트워크 관련 구 명령어를 사용하고 싶으면 아래처럼 설치하면 됨

yum install -y net-utils(CentOS 기준)

**# 패스워드 복잡성 설정**

- 점검 내용 : 시스템 정책에 사용자 계정(root 및 일반 계정 모두 해당)패스워드 복잡성 관련 설정이 되어 있는지 점검

- 점검 목적 : 패스워드 복잡성 관련 정책이 설정되어 있는지 점검하여 비 인가자의 공격(무작위 대입 공격, 사전 대입 공격 등)에 대비가 되어 있는지 확인하기 위함

**판단 기준**

- 양호 : 영문, 숫자, 특수문자를 조합하여 2종류 조합 시 10자리 이성, 3종류 이상 조합 시 8자리 이상의 패스워드가 설정된 경우 (공공기관 9자리 이상)

- 취약 : 영문, 숫자, 특수문자를 조합하여 2종류 조합 시 10자리 미만, 3종류 이상 조합 시 8자리 미만의 길이가 패스워드로 설정된 경우(공고기관 9자리 미만)

CentOS : /etc/pam.d/system-auth

Ubuntu : /etc/pam.d/common-password

가이드라인 상에는 pam\_cracklib.so 모듈 설정법만 있지만 실제로는 더 다양하게 설정을 할 수 있음

그 외 가능한 모듈 : pam\_pwquality.so, pam\_passwdqc.so

- pam\_pwquailty.so

= 해당 모듈을 관련된 파일쪽에 모듈만 설정을 하고 /etc/security/pwquality.conf 라는 파일을 생성해서 사용

= pwquality.conf 파일 안에는 pam\_cracklib.so 모듈처럼 설정

= pam.d 디렉터리 쪽에 설정 시에 local\_users\_only 옵션은 일반 사용자만 적용되게끔 하는 옵션, enforce\_for\_root 로 변경해야 root 계정도 적용

= /etc/security/pwquailty.conf 예시

retry = 3 // 패스워드 입력 실패 시 재시도 횟수

minlen = 8 // 최소 패스워드 길이 설정

lcredit = 1 // 소문자

ucredit = 1 // 대문자

dcredit = 1 // 숫자

ocredit = 1 // 특수문자

=참고사이트 : https://allthatlinux.com/dokuwiki/doku.php?id=%EB%B3%B4%EC%95%88%EC%B7%A8%EC%95%BD%EC%A0%90\_%EC%A0%90%EA%B2%80

- pam\_passwdqc.so

= pam\_cracklib.so 모듈은 root 사용자는 적용이 되지 않기 때문에 이 모듈을 함께 사용

= 예시 : pam\_passwdqc.so min=disabled,12,8,6,5 와 같이 설정

\* min=N1,N2,N3,N4,N5

- N1 은 한 가지 문자 조합 시 암호 구성 길이

- N2 는 두 가지 문자 조합 시 암호 구성 길이

- N3 는 단어 포함 길이 설정

- N4, N5 는 세 가지 문자 조합, 네 가지 문자 조합의 암호 구성 길이

= 참고 사이트 : http://coffeenix.net/board\_print.php?bd\_code=1561

**# 계정 잠금 임계값 설정**

- 점검내용 : 시스템 정책에 사용자 로그인 실패 임계값이 설정되어 있는지 점검

- 점검목적 : 시스템 정책에 사용자 로그인 실패 임계값이 설정되어 있는지 점검하여 비 인가자의 공격(무작위 대입 공격, 사전 대입 공격, 추측 공격 등) 시도 시 로그인 실패 임계값에 따라 로그인을 차단하고 있는지 확인하기 위함

**판단 기준**

- 양호 : 계정 잠금 임계값이 5이하의 값으로 설정되어 있는 경우

- 취약 : 계정 잠금 임계값이 설정되어 있지 않거나, 5이하의 값으로 설정되지 않은 경우

CentOS : /etc/pam.d/system-auth(로컬 접속), /etc/pam.d/password-auth(원격접속)

Ubuntu : /etc/pam.d/common-auth

설정 가능한 모듈 : pam\_tally.so pam\_tally2.so pam\_failock.so

- 실습 명령어

1. vi /etc/pam.d/system-auth

2. auth 로 시작하는 부분 맨 위에 아래의 설정 추가

auth required pam\_tally2.so deny=3

3. account 로 시작하는 부분 맨 위에 아래의 설정 추가

account required pam\_tally2.so

4. pam\_tally2 -u [USER] 명령어로 해당 계정의 틀린 횟수 조회 가능

5. root 계정으로 test 계정 생성

useradd test

passwd test

원하는 비밀번호 설정 또는 kisec123

6. kisec 계정으로 되어있는 새로운 터미널 실행

7. su test 3번 진행 후 4번째 부터는 계정 잠금 문구 출력

8. root 계정의 터미널에서 pam\_tally2 -u test 해보면 계정 실패 횟수 확인 가능

9. pam\_tally2 -u test --reset 해서 실패 횟수 초기화 가능

10. kisec 계정으로 되어있는 터미널에서 다시 su test 후 test 계정으로의 정상 전환 확인

Ex) deny=5 // 몇 회 틀렸을 경우 잠금을 할지 선언

unlock\_time=600 // 잠금시간 설정

**# 패스워드 파일 보호**

- 점검내용 : 시스템의 사용자 계정(root, 일반계정) 정보가 저장된 파일(예 /etc/passwd, /etc/shadow)에 사용자 계정 패스워드가 암호화되어 저장되어 있는지 점검

- 점검목적 : 일부 오래된 시스템의 경우 패스워드 정책이 적용되지 않아 /etc/passwd 파일에 평문으로 저장되므로 사용자 계정 패스워드가 암호화되어 저장되어 있는지 점검하여 비인가자의 패스워드 파일 접근 시에도 사용자 계정 패스워드가 안전하게 관리되고 있는지 확인하기 위함

**판단기준**

- 양호 : 쉐도우 패스워드를 사용하거나, 패스워드를 암호화하여 저장하는 경우

- 취약 : 쉐도우 패스워드를 사용하지 않고, 패스워드를 암호화하여 저장하지 않는 경우

각 파일 마다 뒤에 - 표시가 붙은 파일은 변경되기 전 상태를 저장해놓은 백업 파일

예를 들어 우분투에서 adduser로 계정 생성 시 계정이 생성되고 /etc/passwd 파일을 /etc/passwd-로 백업하고 그 뒤의 설명은 후에 작업되어 내용이 다르다

그런 식으로 중요파일 이기 때문에 백업파일을 따로 두어 무언가 작업이 시작될 경우 백업을 진행한다.

pwconv(shadow 파일 사용)/ pwunconv(shadow 파일 사용 해제)

비밀번호가 두번째 필드인데 $1$가 어떤 해시암호 알고리즘을 사용했는지 알 수 있음

$1 = md5, $5 = sha256, $6 = sha512

! 또는 !! : 계정에 비밀번호가 설정되어있지 않아 잠가놓은 상태

\* : 이 계정을 통해서 아예 로그인을 할 수 없음을 의미

빈 값 : 비밀번호 없이 로그인 할 수 있는 상태를 의미

- 리눅스 설치 시에 root 비밀번호를 설정하지 않고 설치를 하게되면

root 비밀번호가 빈 값으로 나오고 root 로 로그인할 때 비밀번호 필요 없이 로그인 가능

**# /etc/passwd 파일 소유자 및 권한 설정**

- 점검내용 : /etc/passwd 파일 권한 적절성 점검

- 점검목적 : /etc/passwd 파일을 통해 비인가자가 권한 상승하는 것을 막기 위함

**판단 기준**

- 양호 : /etc/passwd 파일의 소유자가 root이고, 권한이 644 이하인 경우

- 취약 : /etc/passwd 파일의 소유자가 root가 아니거나, 권한이 644 이하가 아닌 경우

ls -al /etc/passwd 로 확인이 가능

권한에 대해서 8진수 값을 얻고 싶다면

stat -c "%a" /etc/passwd

소유자만 얻고 싶다면

stat -c "%U" /etc/passwd

644 이하 인데 577을 양호 처리하는 경우 오탐

6 > 5 : 소유자 권한

4 > 7 : 그룹 권한

4 > 7 : 그 외 권한

**SUID, SGID, Sticky bit 설정 및 권한 설정**

- 점검내용 : 불필요하거나 악의적인 파일에 SUID, SGUID 설정 여부 점검

- 점검목적 : 불필요한 SUID, GUID 설정 제거로 악의적인 사용자의 권한상승을 방지하기 위함

**판단 기준**

- 양호 : 주요 실행파일의 권한에 SUID와 SGID에 대한 설정이 부여되어 있지 않은 경우

- 취약 : 주요 실행파일의 권한에 SUID와 SGID에 대한 설정이 부여되어 있는 경우

**$HOME/.rhosts, hosts.equiv 사용 금지**

- 점검내용 : /etc/hosts/equiv 파일 및 .rhosts 파일 사용자를 root 또는 해당 계정으로 설정한 뒤 권한을 600으로 설정하고 해당파일 설정에 ‘+’ 설정(모든 호스트 허용)이 포함되지 않도록 설정되어 있는지 점검

- 점검목적 : ‘r’ command 사용을 통한 원격 접속은 인증 없이 관리자 원격접속이 가능하므로 서비스포트를 차단해야함

**판단기준**

- 양호 : login, shell, exec 서비스를 사용하지 않거나, 사용 시 아래와 같은 설정이 적용된 경우

1. /etc/hosts.equiv 및 $HOME/.rhosts 파일 소유자가 root 또는, 해당 계정인 경우

2. /etc/hosts.equiv 및 $HOME/.rhosts 파일 권한이 600 이하인 경우

3. /etc/hosts.equiv 및 $HOME/.rhosts 파일 설정에 ‘+’ 설정이 없는 경우

- 취약 : login, shell, exec 서비스를 사용하고, 위와 같은 설정이 적용되지 않은 경우

Unix 에서는 stat 명령어 대신 perl 스크립트를 사용

- 실습 명령어

1. yum install -y rsh\*

2. vi /etc/xinetd.d/rlogin, rexec, rsh 파일 모두 활성화 여부 확인

disable = no

3. /etc/hosts.equiv, $HOME/.rhosts 파일들 확인

4. 각 파일의 권한과 소유자, '+' 설정 여부 확인

만약 서비스 실행 중이지만 해당 설정파일들이 존재하지 않을 경우 양호 처리

실행여부 확인 시 해당 포트도 같이 확인

exec(512), login(513), sh(514) 포트를 확인

**# 홈 디렉터리 소유자 및 권한 설정**

- 점검내용 : 홈 디렉터리의 소유자 외 타사용자가 해당 홈 디렉터리를 수정할 수 없도록 제한하는지 저검

- 점검목적 : 홈 디렉터리 소유자가 해당 계정이 아니고, 타 사용자 쓰기 권한이 부여된경우

**판단기준**

- 양호 : 홈 디렉터리 소유자가 해당 계정이고, 일반 사용자 쓰기 권한이 제거된 경우

- 취약 : 홈 디렉터리 소유자가 해당 계정이 아니고, 일반 사용자 쓰기 권한이 부여된 경우

이 항목은 홈 디렉터리가 존재하는 계정일 경우에만 권한을 확인하는 항목

홈 디렉터리 존재 여부 확인은 다음 항목에서 확인한다.

useradd, adduser

리눅스 기본 계정 생성 명령어는 useradd

CentOS의 경우에는 adduser -> useradd 로 심볼릭 링크가 설정되어 있어 결국 useradd로 실행되는 거나 마찬가지

Ubuntu의 경우 adduser가 펄 스크립트로 구성되어 있으며, 사용자 편의를 위해 useradd 옵션 값들을 한 줄씩 지정할 수 있도록 구성

useradd 명령어를 이용해서 생성하는 것이 기본

비밀번호는 passwd 명령어를 이용해서 생성하는 것이 기본

**#홈 디렉터리로 지정한 디렉터리의 존재 관리**

- 점검내용 : 사용자 계정과 홈 디렉터리의 일치 여부를 점검

- 점검목적 : /home 이외 사용자의 홈 디렉터리 존재 여부를 점검하여 비인가자가 시스템 명렁어의 무단 사용을 방지하기 위함

**판단기준**

- 양호 : 홈 디렉터리가 존재하지 않는 계정이 발견되지 않는 경우

- 취약 : 홈 디렉터리가 존재하지 않는 계정이 발견된 경우

**#숨겨진 파일 및 디렉터리 검색 및 제거**

- 점검내용 : 숨김 파일 및 디렉터리 내 의심스러운 파일 존재 여부 점검

- 점검목적 : 숨겨진 파일 및 디렉터리 중 의심스러운 내용은 정상 사용자가 아닌 공격자에 의해 생성되었을 가능성이 높음으로 이를 발견하여 제거함

**판단기준**

- 양호 : 디렉터리 내 숨겨진 파일을 확인하며, 불필요한 파일 삭제를 완료한 경우

- 취약 : 디렉터리 내 숨겨진 파일을 확인하지 않고, 불필요한 파일을 방치한 경우

**전체 숨김 디렉터리 및 숨김 파일 점검**

# find / -type d -name “.\*” (디렉터리 점검)

# find / -type f -name “.\*” (파일점검)

**# finger 서비스 비활성화**

- 점검내용 : Finger 서비스 비활성화 여부 점검

- 점검목적 : Finger(사용자 정보 확인 서비스)를 통해서 네트워크 외부에서 해당 시스템에 등록된 사용자 정보를 확인할 수 있어 비 인가자에게 사용자 정보가 조회되는 것을 차단하고자 함

**판단기준**

- 양호 : Finger 서비스가 비활성화 되어 있는 경우

- 취약 : Finger 서비스가 활성화 되어 있는 경우

finger 서비스 같은 경우 xinetd 로 실행되어짐

/etc/xinetd.d/finger 파일이 있는지 확인을 해서 disable 옵션 확인

yes or no 냐 확인

서비스 포트 같이 확인

cat /etc/services | grep finger

(netstat or ss) -ano | grep ':79'

**#Anonymous FTP 비활성화**

- 점검내용 : 익명 FTP 접속 허용 여부 점검

- 점검목적 : 실행중인 FTP 서비스에 익명 FTP 접속이 허용되고 있는지 확인하여 접속허용을 차단하는 것을 목적으로 함

**판단기준**

- 양호: Anonymous FTP (익명 ftp) 접속을 차단한 경우

- 취약 : Anonymous FTP (익명 ftp) 접속을 차단하지 않은 경우

**#r 계열 서비스 비활성화**

- 점검내용 : r command 서비스 비활성화 여부 점검

- 점검목적 : ‘r’ command 사용을 통한 원격 접속은 NET Backup이나 다른 용도록 사용되기도 하나, 인증 없이 관리자 원격접속이 가능하여 이에 대한 보안위협을 방지하고자 함

**판단기준**

- 양호 : 불필요한 r 계열 서비스가 비활성화 되어 있는 경우

- 취약 : 불필요한 r 계열 서비스가 활성화 되어 있는 경우

##

1. xinetd 데몬 디렉터리에서 rsh, rlogin, rexec (shell, login, exec) 서비스 확인

#ls -alL /etc/xinetd.d/\* | egrep “rsh|rlogin|rexec” | egrep -v “grep|klogin|kshell|kexec”

2. r 서비스 포트 확인

# cat /etc/services | grep login

# cat /etc/services | grep shell

# cat /etc/services | grep exec

3. 서비스 포트 활성화 여부 확인

# netstat -na | egrep “513|514|512“

**#cron 파일 소유자 및 권한 설정**

- 점검내용 : cron 관련 파일의 권한 적절성 점검

- 점검목적 : 비 인가자 allow, deny파일에 접근할 수 없도록 설정하고 있는지 점검하는 것을 목적으로 함

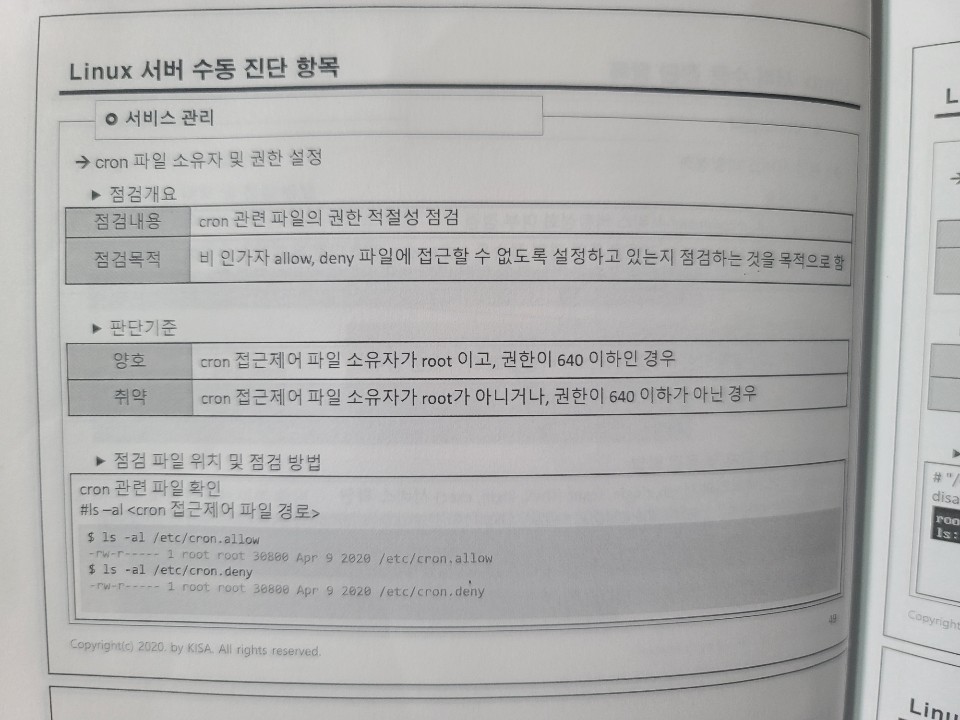
**판단기준**

- 양호 : cron 접근제어 파일 소유자가 root 이고, 권한이 640 이하인 경우

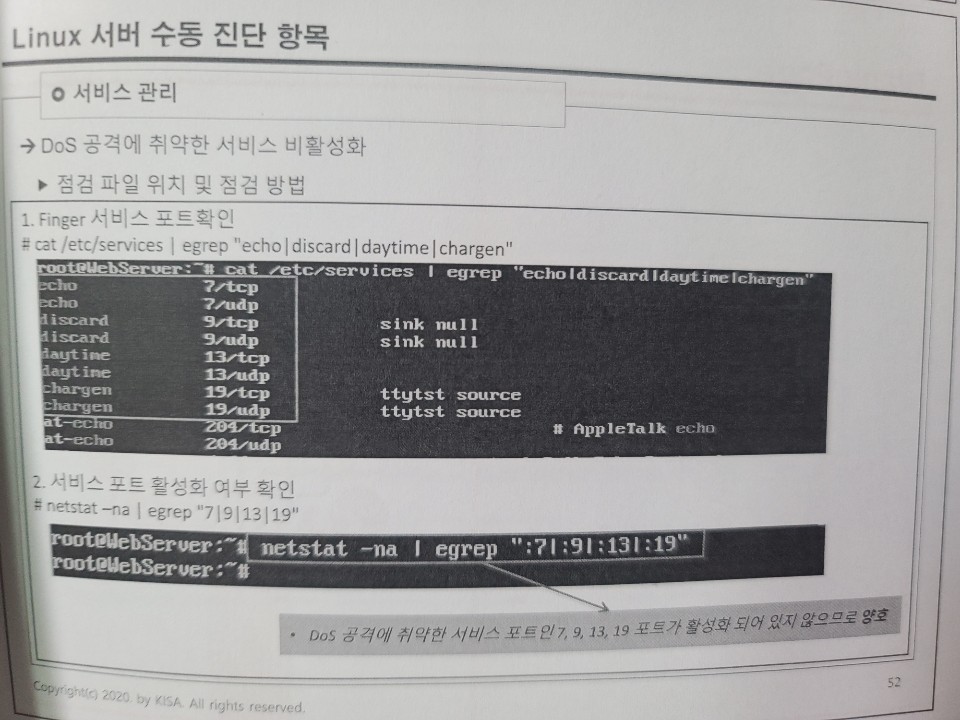
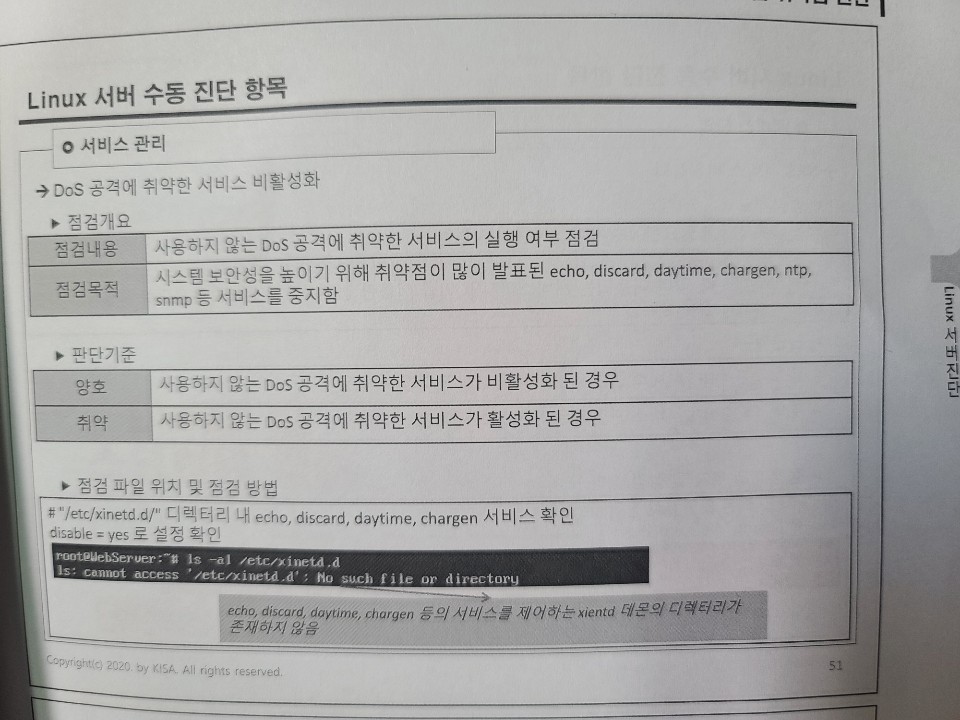
- 취약 : cron 접근제어 파일 소유자가 root가 아니거나, 권한이 640 이하가 아닌 경우

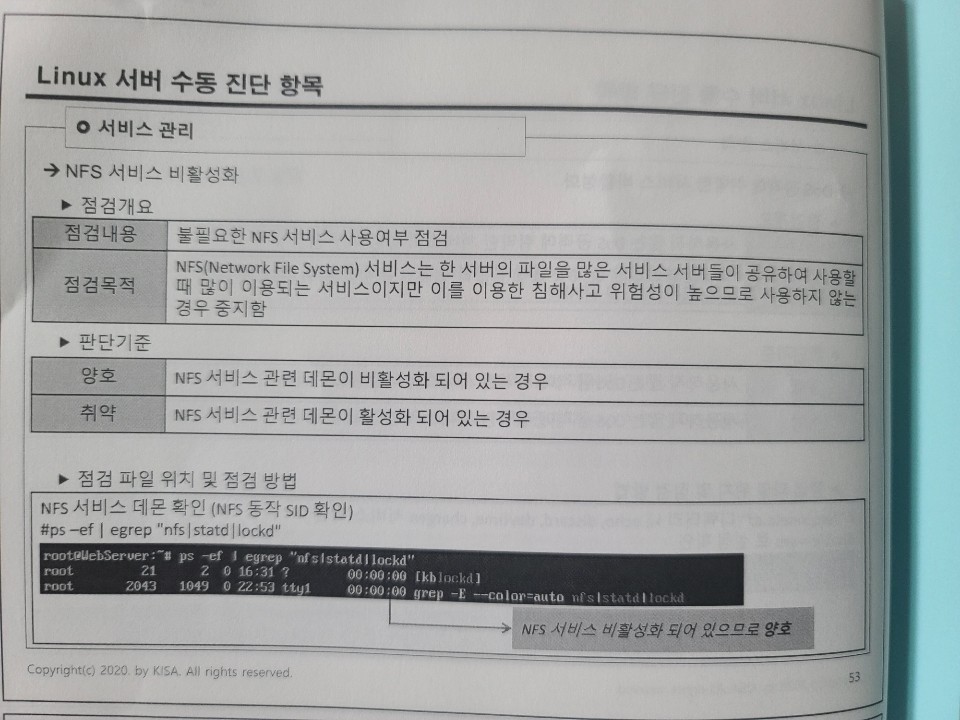
#ls -al <cron 접근제어 파일 경로>

Ex) #ls -al /etc/cron.allow









cat /etc/passwd | grep "ftp"

일반 FTP와 ProFTP는 확인할 수 있다. 위의 명령어를 이용하여

일반 FTP

ProFTP

vsFTP

3가지가 있다.

서비스 활성화 여부

1. 프로세스 확인

2. 포트 확인

3. 서비스 확인

service XXX status

xinetd.d 파일 확인 disable 설정 확인

불필요 한것은 인터뷰를 통해 사용하는지 확인 후 삭제 등 조치를 취해야 한다.

2. Finger 서비스 포트확인 => r계열 서비스 포트확인

오타이다. 48페이지

cron 관련 서비스 관리는 좀 알아두는게 좋다. 페이지 50

사용하지 않는 DoS 공격에 취약한 서비스가 비활성화 된 경우

=> 인터뷰를 통해 해결해야한다. 사용하지 않는 부분때문에

echo, discard, daytime, chargen 이 4가지 확인한다

ls -al /etc/xinetd.d

했을 때 2개씩이유는

tcp = ex) chargen-stream

udp = ex) chargen-dgram

스크립트를 작성 후 담당자에게 넘긴다.

담당자는 각 서버 운영자에게 넘긴다.

운영자는 자신이 맡은 서버에서 스크립트 실행

스크립트 결과가 나오면

운영자가 받아서 담당자에게 넘긴다.

모든 운영자에게 스크립트 결과를 받는다.

그것을 진단자 pc에게 넘긴다.

문제가 있으면 수정 후 다시 넘긴다.

다시 돌렸다가 정상적이면 가지고와서 해당 스크립트를 다시 확인한다.

/etc/passwd 맨 마지막에 보면 로그인할때 쓰이는 쉘이 무엇인지 확인가능 ex) /bin/bash

쉘에서 echo 명령어를 많이 사용한다.

리다이렉션

만약 파일이 없는 경우 새롭게 만들어진다.

페이지 74

grep이랑 egrep 바꿔야 한다.

잘못 적혀있다.

grep과 egrep의 옵션은 동일하다.

가장 많이 쓰는 것은 -i와 -v이다.

-name

-type

많이 사용.

-nogroup

-nouser

이 두개는 사용할 때가 가끔 있다.

vim

일반모드를 제외한 나머지 3모드는 일반모드를 통해서

gg뒤에 있는거 빼야한다.

y 한 줄 복사 -> yy 한 줄 복사

리눅스에서 컨트롤 z는 하면 안된다.

되돌리기아니다

리눅스에서 되돌리기는 u

fg

set nohlsearch 하이라이팅 없어진다

set nonu 숫자 없어진다.

set nu

# Anonymous FTP 비활성화

- 실습명령어

1. yum install -y vsftpd

2. service vsftpd start

3. ps -ef | grep vsftpd | grep -v grep

4. vsftpd 설정 파일 확인

/etc/vsftpd/vsftpd.conf or /etc/vsftpd.conf

5. vi /etc/vsftpd/vsftpd.conf

서비스 활성화 여부 확인

1. 프로세스 확인

2. 포트 확인(21)

3. 서비스 확인

service XXX status

xinetd.d 파일 확인 disable 설정 확인

이 항목은 일반 FTP, ProFTP, vsFTP

일반 FTP : xinetd 로만 실행

ProFTP, vsFTP : xinetd or standalone 모드로도 실행 가능

일반FTP, ProFTP : /etc/passwd 파일 내에 ftp 계정 확인

vsFTP : 해당 서비스의 설정파일 안에 anonymous\_enable 설정 확인

YES 면 취약, NO 면 양호

# r 계열 서비스 비활성화

1. xinetd 쪽에 활성화 여부를 확인 rlogin, rexec, rshell 서비스를 확인

disable 설정

2. 포트 활성화 여부를 확인

exec(512), login(513), shell(514)

netstat, ss 명령어를 통해 확인

두 가지 확인을 통해 서비스가 나왔다면 담당자와 인터뷰를 통해 해당 서비스 필요 여부 확인 후 판단

# cron 파일 소유자 및 권한 설정

/etc/cron.allow : crontab 명령어 사용 허가 파일

/etc/cron.deny : crontab 명령어 사용 제한 파일

# NFS 서비스 비활성화

- 실습명령어

1. yum install -y nfs-utils rpcbind

2. service rpcbind start

3. service nfs start

4. rpcinfo -p 명령어로 nfs 데몬들 활성화 여부 확인 가능

5. 프로세스 목록에서도 서비스 활성화 여부 확인

Shell 영어 뜻 : (달걀, 견과류 등의 딱딱한)껍데기, 껍데기 모양의 것

즉 무언가를 감싸고 있는 것을 의미

켄 톰프슨 : B언어, 유닉스 개발, Go 언어 개발, UTF-8 개발

데니스 리치 : C 언어

현재도 bourne Shell 이 존재 하지만 대부분의 배포판에서는 /bin/bash로 심볼릭 링크가 설정

Ubuntu 의 경우 sh -> dash 로 심볼릭 링크

dash 는 bash는 가볍고 sh 보다는 기능이 좀 더 있음

chsh 에서 -l 옵션은 CentOS 만 사용 가능

Ubuntu 는 chsh 에 -l 옵션이 없음

dash 일 경우 echo 할 때 -e 옵션을 주지 않아도 이스케이프 문자 사용 가능

bash 일 경우에는 -e 옵션을 주어서 이스케이프 문자를 인식을 시켜줘야 함

> : 덮어쓰기

>> : 이어쓰기

[표준 스트림]

0 : 표준 입력

1 : 표준 출력

2 : 표준 에러 출력

파이프 라인 역할 : 표준 출력 값을 다음 명령에 표준 입력으로 넣어주는 것

파이프 라인 사용시 주의 사항

- 갯수 제한은 없으나 파이프라인을 너무 많이 사용을 하면(너무 많이 연결을 하면) 중간에 값이 없어지는 경우가 생김

- 중간에 끊고 변수에 저장을 한 다음 다시 파이프 라인을 사용하는게 좋음

2>&1 : 표준 에러 출력을 표준 출력으로 표준 입력 리다이렉션

echo "Hello World" > file.txt : 표준 에러 출력을 터미널에 작성

echo "Hello World" > file.txt 2>&1 : 표준 에러 출력까지 파일에 작성

1. echo "Hello World" > file.txt 2> file.txt

2. echo "Hello World" > file.txt 2>&1

& : 추가로 리다이렉션 스트림을 지정할 때 사용

2>/dev/null : 표준 에러 출력을 /dev/null 쪽으로 표준 입력 리다이렉션

/dev/null 파일 : 아무것도 존재하지 않는 특수 파일

이 파일에 쓰여지는 모든 데이터는 버려지지만 정상적으로 쓰기 작업이 종료되었다고 인식

블랙홀 또는 비트 버킷 이라고 불리움

awk 옵션 알려주는 사이트

https://sodocumentation.net/ko/awk/topic/937/awk-%EC%8B%9C%EC%9E%91%ED%95%98%EA%B8%B0

sed 옵션 알려주는 사이트

<https://sodocumentation.net/ko/sed/topic/934/sed-%EC%8B%9C%EC%9E%91%ED%95%98%EA%B8%B0>

쉘 프로그래밍

- 첫 번째 줄에 반드시 사용할 쉘을 명시(#!/bin/bash, #!/bin/sh)

#!/bin/sh : 쉘뱅, 해시뱅

정의되어 있는 변수 해제 : unset

#!/bin/sh : 쉘뱅, 해시뱅

#!/usr/bin/env python3

C 언어의 경우 변수 생성 방법

int num

num = 0

쉘 스크립트 변수 생성 방법

num=0

만약 빈 값을 할당 해놓고 있다면

num=""

1num=0 (x) : 숫자로 시작하는 변수 사용 불가

- 쉘의 내부 변수 사용 시 문제로 인해 허용하지 않음

$1num

$0, $1, $2

num1=0 (o)

num-1=0 (x)

num\_1=0 (o)

test\_str="" (O)

test\_str = "" (X)

test\_str= "" (X)

test\_str ="" (X)

${test\_str} == $test\_str

value="Hello"

echo $value

value="World"

echo $value

출력 값

Hello

World

쉘 스크립트 작성 시 /bin/bash가 아닌 /bin/sh 로 하는 이유

- bash 가 없는 경우가 있으 수 있음(하위호환성 유지)

if(띄어쓰기)[(띄어쓰기)조건식(띄어쓰기)];(띄어쓰기)then

실행 구문

fi

if(띄어쓰기)[(띄어쓰기)조건식(띄어쓰기)];(띄어쓰기)then

실행 구문

elif(띄어쓰기)[(띄어쓰기)조건식(띄어쓰기)];(띄어쓰기)then

실행 구문

else

실행 구문

fi

문자열 비교 시 많이 사용하는 구문

value=`ps -ef | grep kisec`

if [ "$value" != "" ]; then

실행 구문

fi

for 변수 in 나눠질 값

do

실행 구문

done

for str in Hello World Script!

do

echo $str

done

출력값

Hello

World

Script!

for str in "Hello World Script!"

do

echo $str

done

출력값

Hello World Script!

full\_str="Hello World Script!"

for str in $full\_str

do

echo $str

done

출력값

Hello

World

Script!

RESULT\_FILE="result\_centos\_hostname\_(날짜와시간).txt"

#!/bin/sh

RESULT\_FILE="result.txt"

echo "[U-01] root 계정의 원격 접속 제한"

echo "[U-01] root 계정의 원격 접속 제한" > $RESULT\_FILE 2>&1

if [ -f "/etc/pam.d/remote" ]; then

pam\_chk=`cat /etc/pam.d/remote | egrep -v '^#|^$' | grep "pam\_securetty.so"`

echo "/etc/pam.d/remote 파일 현황" >> $RESULT\_FILE 2>&1

echo "`cat /etc/pam.d/remote | egrep -v '^#|^$'`" >> $RESULT\_FILE 2>&1

else

TELNET=0

echo "/etc/pam.d/remote Not Found" >> $RESULT\_FILE 2>&1

fi

if [ -f "/etc/securetty" ] && [ "$pam\_chk" != "" ]; then

pts\_chk=`cat /etc/securetty | egrep -v '^#|^$' | grep -i 'pts'`

echo "/etc/securetty 파일 현황" >> $RESULT\_FILE 2>&1

echo "`cat /etc/securetty | egrep -v '^#|^$'`" >> $RESULT\_FILE 2>&1

if [ "$pts\_chk" != "" ]; then

TELNET=0

else

TELNET=1

fi

else

TELNET=0

echo "/etc/securetty Not Found or pam\_securetty.so module Not Found" >> $RESULT\_FILE 2>&1

fi

if [ -f "/etc/ssh/sshd\_config" ]; then

ssh\_chk=`cat /etc/ssh/sshd\_config | grep -v '^#' | grep -i 'permitrootlogin'`

echo "/etc/ssh/sshd\_config 파일 현황" >> $RESULT\_FILE 2>&1

echo "`cat /etc/ssh/sshd\_config | egrep -v '^#|^$'`" >> $RESULT\_FILE 2>&1

if [ "`echo $ssh\_chk | awk '{ print $2 }'`" = "no" ]; then

SSH=1

elif [ "$ssh\_chk" = "" ]; then

SSH=1

else

SSH=0

fi

else

echo "/etc/ssh/sshd\_config Not Found" >> $RESULT\_FILE 2>&1

fi

if [ $TELNET -eq 1 -a $SSH -eq 1 ]; then

echo "결과 : 양호" >> $RESULT\_FILE 2>&1

else

echo "결과 : 취약" >> $RESULT\_FILE 2>&1

fi

unset TELNET

unset SSH

echo "[U-04] 패스워드 파일 보호"

echo "[U-04] 패스워드 파일 보호" >> $RESULT\_FILE 2>&1

SHADOW=0

PASSWD=0

if [ -f "/etc/shadow" ]; then

SHADOW=1

echo "/etc/shadow 파일 현황" >> $RESULT\_FILE 2>&1

echo "`ls -al /etc/shadow`" >> $RESULT\_FILE 2>&1

else

echo "/etc/shadow Not Found" >> $RESULT\_FILE 2>&1

fi

if [ -f "/etc/passwd" ]; then

pass\_chk=`cat /etc/passwd | awk -F":" '{ print $2 }' | grep -v 'x'`

echo "/etc/passwd 파일 현황" >> $RESULT\_FILE 2>&1

echo "`cat /etc/passwd`" >> $RESULT\_FILE 2>&1

if [ "$pass\_chk" = "" ]; then

PASSWD=1

fi

else

echo "/etc/passwd Not Found" >> $RESULT\_FILE 2>&1

fi

if [ $SHADOW -eq 1 -a $PASSWD -eq 1 ]; then

echo "결과 : 양호" >> $RESULT\_FILE 2>&1

else

echo "결과 : 취약" >> $RESULT\_FILE 2>&1

fi

unset SHADOW

unset PASSWD

echo "[U-11] /etc/syslog.conf 파일 소유자 및 권한 설정"

echo "[U-11] /etc/syslog.conf 파일 소유자 및 권한 설정" >> $RESULT\_FILE 2>&1

if [ -f "/etc/syslog.conf" ]; then

sys\_file="/etc/syslog.conf"

elif [ -f "/etc/rsyslog.conf" ]; then

sys\_file="/etc/rsyslog.conf"

fi

sys\_perm=`stat -c "%a" "$sys\_file"`

sys\_owner=`stat -c "%U" "$sys\_file"`

owner\_perm\_val=`echo "$sys\_perm" | awk '{ print substr($0, 1, 1)}'`

group\_perm\_val=`echo "$sys\_perm" | awk '{ print substr($0, 2, 1)}'`

other\_perm\_val=`echo "$sys\_perm" | awk '{ print substr($0, 3, 1)}'`

echo "/etc/(r)syslog.conf 파일 현황" >> $RESULT\_FILE 2>&1

echo "`ls -al $sys\_file`" >> $RESULT\_FILE 2>&1

if [ "$owner\_perm\_val" -le 6 ] && [ "$group\_perm\_val" -le 4 ] && [ "$other\_perm\_val" -le 4 ]; then

if [ "$sys\_owner" = "root" -o "$sys\_owner" = "bin" -o "$sys\_owner" = "sys" ]; then

echo "결과 : 양호" >> $RESULT\_FILE 2>&1

else

echo "결과 : 취약" >> $RESULT\_FILE 2>&1

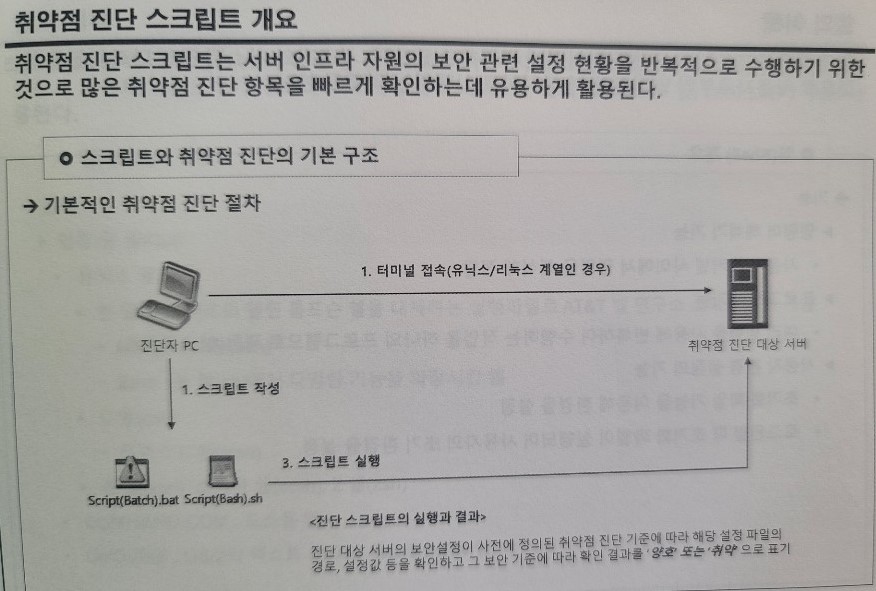
fi

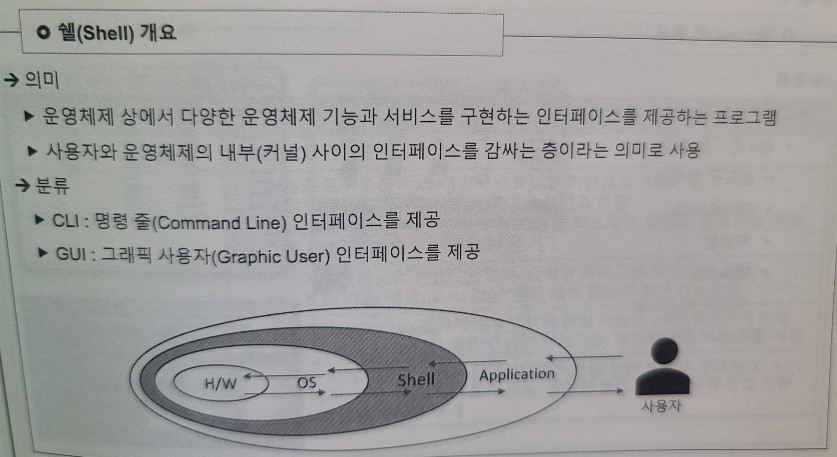
else

echo "결과 : 취약" >> $RESULT\_FILE 2>&1

fi

**쉘(Shell)(응용프로그램과 운영체제 사이에 위치하고 있으며, 사용자와 시스템 간의 다리역할을 하는 것으로 사용자로부터 명령을 받아 해석하고 실행하는 역할을 한다.)**

****

****

**기능**

- 명령어 해석기 기능 : 사용자와 커널 사이에서 명령을 해석해 전달

- 프로그래밍 기능 : 여러 명령을 사용해 반복하여 수행하는 작업을 하나의 프로그램으로 제작(쉘 스크립트)

- 사용자 환경 설정의 기능 :

-> 초기화 파일 기능을 이용해 환경을 설정

-> 로그인할 때 초기화 파일이 실행되어 사용자의 초기 환경을 설정

**종류**

- 그래픽 쉘(GUI)

-> MS 윈도우 환경

--> 윈도우 탐색기. 라이트스탭. 지오쉘. BB4Win. 이머지 데스크톱

- 매킨토시 파인더

- X 윈도우 시스템 기반 환경(주로 유닉스)

- 명령 줄 쉘(CLI)

-> 유닉스 쉘

--> 본 쉘(sh) : 최초의 쉘인 톰프슨 쉘을 대체하는 실행파일로 AT&T 벨 연구소 스티븐 본이 개발

- Almquist shell(ash)

- Bash : 본 쉘(sh)에서 다양한 기능을 확장시킨 쉘

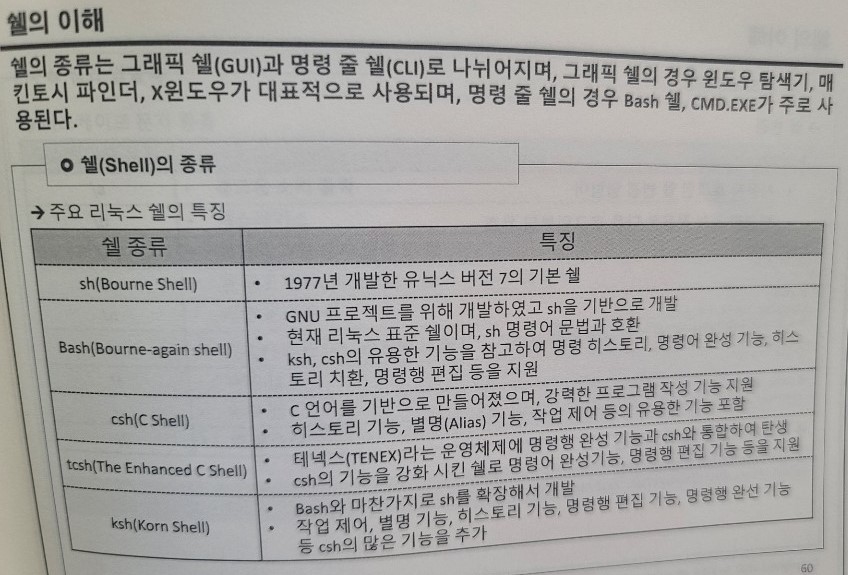
--> C 쉘(csh)

- 테넥스 C 쉘(tcsh)

--> 콘 쉘(ksh), 스키마 쉘(scsh), Z 쉘(zsh)

-> COMMAND.COM : 도스용 쉘

-> CMD.EXE : OS/2의 텍스트 모드와 윈도우 NT용 쉘



cat /etc/passwd | grep "ftp"

일반 FTP와 ProFTP는 확인할 수 있다. 위의 명령어를 이용하여

일반 FTP

ProFTP

vsFTP

3가지가 있다.

서비스 활성화 여부

1. 프로세스 확인

2. 포트 확인

3. 서비스 확인

service XXX status

xinetd.d 파일 확인 disable 설정 확인

불필요 한것은 인터뷰를 통해 사용하는지 확인 후 삭제 등 조치를 취해야 한다.

2. Finger 서비스 포트확인 => r계열 서비스 포트확인

오타이다. 48페이지

cron 관련 서비스 관리는 좀 알아두는게 좋다. 페이지 50

사용하지 않는 DoS 공격에 취약한 서비스가 비활성화 된 경우

=> 인터뷰를 통해 해결해야한다. 사용하지 않는 부분때문에

echo, discard, daytime, chargen 이 4가지 확인한다

ls -al /etc/xinetd.d

했을 때 2개씩이유는

tcp = ex) chargen-stream

udp = ex) chargen-dgram

스크립트를 작성 후 담당자에게 넘긴다.

담당자는 각 서버 운영자에게 넘긴다.

운영자는 자신이 맡은 서버에서 스크립트 실행

스크립트 결과가 나오면

운영자가 받아서 담당자에게 넘긴다.

모든 운영자에게 스크립트 결과를 받는다.

그것을 진단자 pc에게 넘긴다.

문제가 있으면 수정 후 다시 넘긴다.

다시 돌렸다가 정상적이면 가지고와서 해당 스크립트를 다시 확인한다.

/etc/passwd 맨 마지막에 보면 로그인할때 쓰이는 쉘이 무엇인지 확인가능 ex) /bin/bash

쉘에서 echo 명령어를 많이 사용한다.

리다이렉션

만약 파일이 없는 경우 새롭게 만들어진다.

페이지 74

grep이랑 egrep 바꿔야 한다.

잘못 적혀있다.

grep과 egrep의 옵션은 동일하다.

가장 많이 쓰는 것은 -i와 -v이다.

-name

-type

많이 사용.

-nogroup

-nouser

이 두개는 사용할 때가 가끔 있다.

vim

일반모드를 제외한 나머지 3모드는 일반모드를 통해서

gg뒤에 있는거 빼야한다.

y 한 줄 복사 -> yy 한 줄 복사

리눅스에서 컨트롤 z는 하면 안된다.

되돌리기아니다

리눅스에서 되돌리기는 u

fg

set nohlsearch 하이라이팅 없어진다

set nonu 숫자 없어진다.

set nu

# Anonymous FTP 비활성화

- 실습명령어

1. yum install -y vsftpd

2. service vsftpd start

3. ps -ef | grep vsftpd | grep -v grep

4. vsftpd 설정 파일 확인

/etc/vsftpd/vsftpd.conf or /etc/vsftpd.conf

5. vi /etc/vsftpd/vsftpd.conf

서비스 활성화 여부 확인

1. 프로세스 확인

2. 포트 확인(21)

3. 서비스 확인

service XXX status

xinetd.d 파일 확인 disable 설정 확인

이 항목은 일반 FTP, ProFTP, vsFTP

일반 FTP : xinetd 로만 실행

ProFTP, vsFTP : xinetd or standalone 모드로도 실행 가능

일반FTP, ProFTP : /etc/passwd 파일 내에 ftp 계정 확인

vsFTP : 해당 서비스의 설정파일 안에 anonymous\_enable 설정 확인

YES 면 취약, NO 면 양호

# r 계열 서비스 비활성화

1. xinetd 쪽에 활성화 여부를 확인 rlogin, rexec, rshell 서비스를 확인

disable 설정

2. 포트 활성화 여부를 확인

exec(512), login(513), shell(514)

netstat, ss 명령어를 통해 확인

두 가지 확인을 통해 서비스가 나왔다면 담당자와 인터뷰를 통해 해당 서비스 필요 여부 확인 후 판단

# cron 파일 소유자 및 권한 설정

/etc/cron.allow : crontab 명령어 사용 허가 파일

/etc/cron.deny : crontab 명령어 사용 제한 파일

# NFS 서비스 비활성화

- 실습명령어

1. yum install -y nfs-utils rpcbind

2. service rpcbind start

3. service nfs start

4. rpcinfo -p 명령어로 nfs 데몬들 활성화 여부 확인 가능

5. 프로세스 목록에서도 서비스 활성화 여부 확인

Shell 영어 뜻 : (달걀, 견과류 등의 딱딱한)껍데기, 껍데기 모양의 것

즉 무언가를 감싸고 있는 것을 의미

켄 톰프슨 : B언어, 유닉스 개발, Go 언어 개발, UTF-8 개발

데니스 리치 : C 언어

현재도 bourne Shell 이 존재 하지만 대부분의 배포판에서는 /bin/bash로 심볼릭 링크가 설정

Ubuntu 의 경우 sh -> dash 로 심볼릭 링크

dash 는 bash는 가볍고 sh 보다는 기능이 좀 더 있음

chsh 에서 -l 옵션은 CentOS 만 사용 가능

Ubuntu 는 chsh 에 -l 옵션이 없음

dash 일 경우 echo 할 때 -e 옵션을 주지 않아도 이스케이프 문자 사용 가능

bash 일 경우에는 -e 옵션을 주어서 이스케이프 문자를 인식을 시켜줘야 함

> : 덮어쓰기

>> : 이어쓰기

[표준 스트림]

0 : 표준 입력

1 : 표준 출력

2 : 표준 에러 출력

파이프 라인 역할 : 표준 출력 값을 다음 명령에 표준 입력으로 넣어주는 것

파이프 라인 사용시 주의 사항

- 갯수 제한은 없으나 파이프라인을 너무 많이 사용을 하면(너무 많이 연결을 하면) 중간에 값이 없어지는 경우가 생김

- 중간에 끊고 변수에 저장을 한 다음 다시 파이프 라인을 사용하는게 좋음

2>&1 : 표준 에러 출력을 표준 출력으로 표준 입력 리다이렉션

echo "Hello World" > file.txt : 표준 에러 출력을 터미널에 작성

echo "Hello World" > file.txt 2>&1 : 표준 에러 출력까지 파일에 작성

1. echo "Hello World" > file.txt 2> file.txt

2. echo "Hello World" > file.txt 2>&1

& : 추가로 리다이렉션 스트림을 지정할 때 사용

2>/dev/null : 표준 에러 출력을 /dev/null 쪽으로 표준 입력 리다이렉션

/dev/null 파일 : 아무것도 존재하지 않는 특수 파일

이 파일에 쓰여지는 모든 데이터는 버려지지만 정상적으로 쓰기 작업이 종료되었다고 인식

블랙홀 또는 비트 버킷 이라고 불리움

awk 옵션 알려주는 사이트

https://sodocumentation.net/ko/awk/topic/937/awk-%EC%8B%9C%EC%9E%91%ED%95%98%EA%B8%B0

sed 옵션 알려주는 사이트

https://sodocumentation.net/ko/sed/topic/934/sed-%EC%8B%9C%EC%9E%91%ED%95%98%EA%B8%B0

쉘 프로그래밍

- 첫 번째 줄에 반드시 사용할 쉘을 명시(#!/bin/bash, #!/bin/sh)

#!/bin/sh : 쉘뱅, 해시뱅

정의되어 있는 변수 해제 : unset

#!/bin/sh : 쉘뱅, 해시뱅

#!/usr/bin/env python3

C 언어의 경우 변수 생성 방법

int num

num = 0

쉘 스크립트 변수 생성 방법

num=0

만약 빈 값을 할당 해놓고 있다면

num=""

1num=0 (x) : 숫자로 시작하는 변수 사용 불가

- 쉘의 내부 변수 사용 시 문제로 인해 허용하지 않음

$1num

$0, $1, $2

num1=0 (o)

num-1=0 (x)

num\_1=0 (o)

test\_str="" (O)

test\_str = "" (X)

test\_str= "" (X)

test\_str ="" (X)

${test\_str} == $test\_str

value="Hello"

echo $value

value="World"

echo $value

출력 값

Hello

World

쉘 스크립트 작성 시 /bin/bash가 아닌 /bin/sh 로 하는 이유

- bash 가 없는 경우가 있으 수 있음(하위호환성 유지)

if(띄어쓰기)[(띄어쓰기)조건식(띄어쓰기)];(띄어쓰기)then

실행 구문

fi

if(띄어쓰기)[(띄어쓰기)조건식(띄어쓰기)];(띄어쓰기)then

실행 구문

elif(띄어쓰기)[(띄어쓰기)조건식(띄어쓰기)];(띄어쓰기)then

실행 구문

else

실행 구문

fi

문자열 비교 시 많이 사용하는 구문

value=`ps -ef | grep kisec`

if [ "$value" != "" ]; then

실행 구문

fi

for 변수 in 나눠질 값

do

실행 구문

done

for str in Hello World Script!

do

echo $str

done

출력값

Hello

World

Script!

for str in "Hello World Script!"

do

echo $str

done

출력값

Hello World Script!

full\_str="Hello World Script!"

for str in $full\_str

do

echo $str

done

출력값

Hello

World

Script!

RESULT\_FILE="result\_centos\_hostname\_(날짜와시간).txt"

#!/bin/sh

RESULT\_FILE="result.txt"

echo "[U-01] root 계정의 원격 접속 제한"

echo "[U-01] root 계정의 원격 접속 제한" > $RESULT\_FILE 2>&1

if [ -f "/etc/pam.d/remote" ]; then

pam\_chk=`cat /etc/pam.d/remote | egrep -v '^#|^$' | grep "pam\_securetty.so"`

echo "/etc/pam.d/remote 파일 현황" >> $RESULT\_FILE 2>&1

echo "`cat /etc/pam.d/remote | egrep -v '^#|^$'`" >> $RESULT\_FILE 2>&1

else

TELNET=0

echo "/etc/pam.d/remote Not Found" >> $RESULT\_FILE 2>&1

fi

if [ -f "/etc/securetty" ] && [ "$pam\_chk" != "" ]; then

pts\_chk=`cat /etc/securetty | egrep -v '^#|^$' | grep -i 'pts'`

echo "/etc/securetty 파일 현황" >> $RESULT\_FILE 2>&1

echo "`cat /etc/securetty | egrep -v '^#|^$'`" >> $RESULT\_FILE 2>&1

if [ "$pts\_chk" != "" ]; then

TELNET=0

else

TELNET=1

fi

else

TELNET=0

echo "/etc/securetty Not Found or pam\_securetty.so module Not Found" >> $RESULT\_FILE 2>&1

fi

if [ -f "/etc/ssh/sshd\_config" ]; then

ssh\_chk=`cat /etc/ssh/sshd\_config | grep -v '^#' | grep -i 'permitrootlogin'`

echo "/etc/ssh/sshd\_config 파일 현황" >> $RESULT\_FILE 2>&1

echo "`cat /etc/ssh/sshd\_config | egrep -v '^#|^$'`" >> $RESULT\_FILE 2>&1

if [ "`echo $ssh\_chk | awk '{ print $2 }'`" = "no" ]; then

SSH=1

elif [ "$ssh\_chk" = "" ]; then

SSH=1

else

SSH=0

fi

else

echo "/etc/ssh/sshd\_config Not Found" >> $RESULT\_FILE 2>&1

fi

if [ $TELNET -eq 1 -a $SSH -eq 1 ]; then

echo "결과 : 양호" >> $RESULT\_FILE 2>&1

else

echo "결과 : 취약" >> $RESULT\_FILE 2>&1

fi

unset TELNET

unset SSH

echo "[U-04] 패스워드 파일 보호"

echo "[U-04] 패스워드 파일 보호" >> $RESULT\_FILE 2>&1

SHADOW=0

PASSWD=0

if [ -f "/etc/shadow" ]; then

SHADOW=1

echo "/etc/shadow 파일 현황" >> $RESULT\_FILE 2>&1

echo "`ls -al /etc/shadow`" >> $RESULT\_FILE 2>&1

else

echo "/etc/shadow Not Found" >> $RESULT\_FILE 2>&1

fi

if [ -f "/etc/passwd" ]; then

pass\_chk=`cat /etc/passwd | awk -F":" '{ print $2 }' | grep -v 'x'`

echo "/etc/passwd 파일 현황" >> $RESULT\_FILE 2>&1

echo "`cat /etc/passwd`" >> $RESULT\_FILE 2>&1

if [ "$pass\_chk" = "" ]; then

PASSWD=1

fi

else

echo "/etc/passwd Not Found" >> $RESULT\_FILE 2>&1

fi

if [ $SHADOW -eq 1 -a $PASSWD -eq 1 ]; then

echo "결과 : 양호" >> $RESULT\_FILE 2>&1

else

echo "결과 : 취약" >> $RESULT\_FILE 2>&1

fi

unset SHADOW

unset PASSWD

echo "[U-11] /etc/syslog.conf 파일 소유자 및 권한 설정"

echo "[U-11] /etc/syslog.conf 파일 소유자 및 권한 설정" >> $RESULT\_FILE 2>&1

if [ -f "/etc/syslog.conf" ]; then

sys\_file="/etc/syslog.conf"

elif [ -f "/etc/rsyslog.conf" ]; then

sys\_file="/etc/rsyslog.conf"

fi

sys\_perm=`stat -c "%a" "$sys\_file"`

sys\_owner=`stat -c "%U" "$sys\_file"`

owner\_perm\_val=`echo "$sys\_perm" | awk '{ print substr($0, 1, 1)}'`

group\_perm\_val=`echo "$sys\_perm" | awk '{ print substr($0, 2, 1)}'`

other\_perm\_val=`echo "$sys\_perm" | awk '{ print substr($0, 3, 1)}'`

echo "/etc/(r)syslog.conf 파일 현황" >> $RESULT\_FILE 2>&1

echo "`ls -al $sys\_file`" >> $RESULT\_FILE 2>&1

if [ "$owner\_perm\_val" -le 6 ] && [ "$group\_perm\_val" -le 4 ] && [ "$other\_perm\_val" -le 4 ]; then

if [ "$sys\_owner" = "root" -o "$sys\_owner" = "bin" -o "$sys\_owner" = "sys" ]; then

echo "결과 : 양호" >> $RESULT\_FILE 2>&1

else

echo "결과 : 취약" >> $RESULT\_FILE 2>&1

fi

else

echo "결과 : 취약" >> $RESULT\_FILE 2>&1

fi

service nfs start

service nfs stop

# DNS Zone Transfer 설정

- 실습명령어

1. yum install -y bind

2. vi /etc/named.conf

3. allow-query 아래에

allow-transfer { localhost; }; 구문 삽입

4. named-checkconf 명령어를 사용하여 설정 파일 구문 오류 확인

아무런 결과 값이 없어야 정상

5. service named start

6. dig @localhost axfr localhost

[ 첫 번째 dig 결과 ]

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.68.rc1.el6\_10.3 <<>> @localhost axfr localhost

; (2 servers found)

;; global options: +cmd

localhost. 86400 IN SOA localhost. rname.invalid. 0 86400 3600 604800 10800

localhost. 86400 IN NS localhost.

localhost. 86400 IN A 127.0.0.1

localhost. 86400 IN AAAA ::1

localhost. 86400 IN SOA localhost. rname.invalid. 0 86400 3600 604800 10800

;; Query time: 0 msec

;; SERVER: ::1#53(::1)

;; WHEN: Tue Mar 31 03:19:35 2020

;; XFR size: 5 records (messages 1, bytes 170)

7. vi /etc/named.conf

allow-transfer { none; }; 으로 변경

8. named-checkconf

9. service named restart

10. dig @localhost axfr localhost

[ 두 번째 dig 결과 ]

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.68.rc1.el6\_10.7 <<>> @localhost axfr localhost

; (2 servers found)

;; global options: +cmd

; Transfer failed.

취약한 설정 : allow-transfer { any; };

named.conf 주석 가능 문자

1. #

2. //

3. /\* ~~~~ \*/

/\*

slfjsdlkf

sadflksadjfl;ksdaf

safjlsdfjal;k

aslasdfjsdfk

\*/

allow-transfer 설정 예시)

allow-transfer { 192.168.100.0/24; };

allow-transfer {

none;

};

allow-transfer

{

localhost;

};

allow-transfer{

localhost;};

allow-transfer {none;};

allow-transfer

{

localhost;

};

allow-transfer{

localhost;};

if($i ~ /\/\\*/)

\* : 모든, 0회 이상

**쉘**

- 쉘 확인

-> 로그인 쉘 관련 환경 변수 : SHELL

-> 현재 설정된 쉘 확인 방법 : echo $SHELL

-> 변경 가능한 쉘 확인 : chsh -l cat /etc/shells

**쉘 변경**

- chsh

-> 사용자 로그인 쉘 변경 명령어

-> 변경한 쉘의 적용은 다음 로그인부터 유효

$ echo $SHELL

$ chsh

$ echo $SHELL

-> 사용자 로그인 쉘 정보 확인

$ cat /etc/passwd | grep kisec

**Echo**

- 주어진 문자열을 공백과 개행 문자 등을 포함하여 표준 출력으로 출력하는 명령어

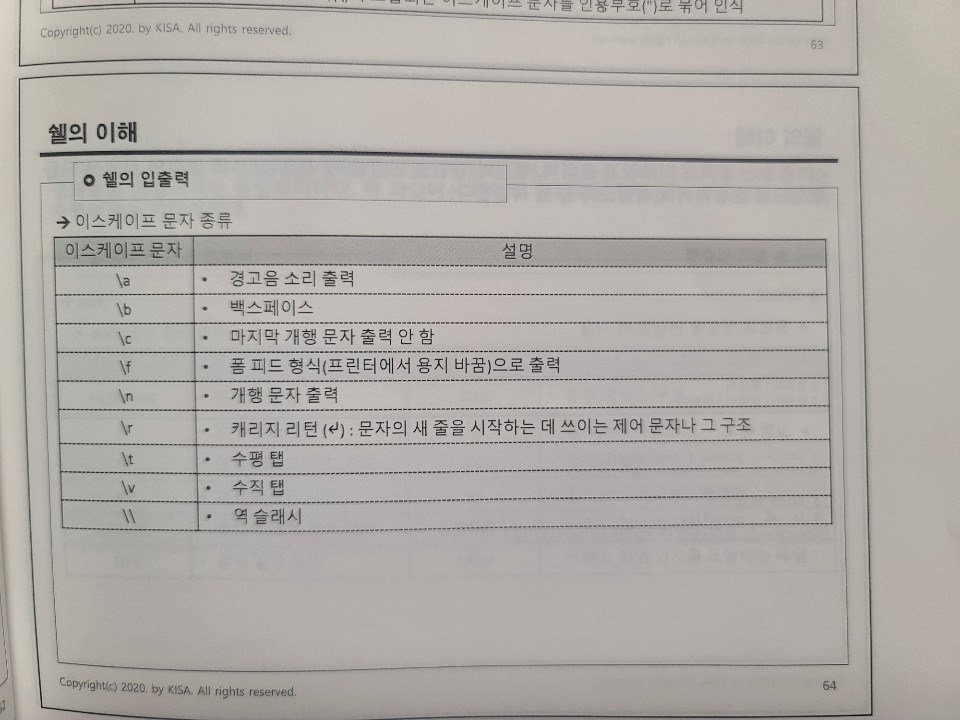
- 사용법

Echo [옵션][문자열]

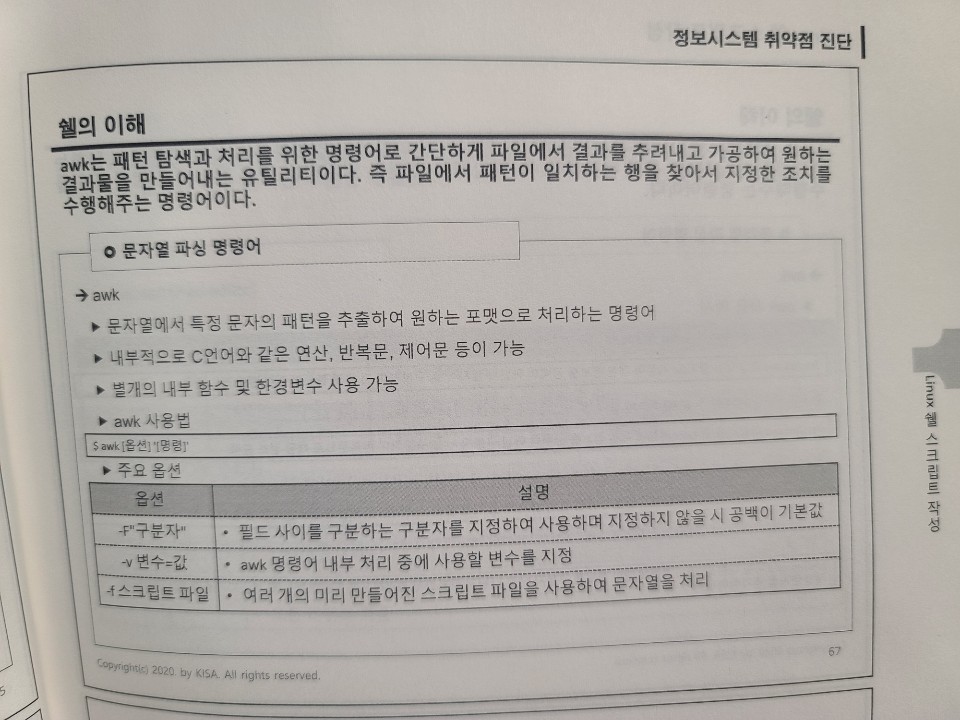
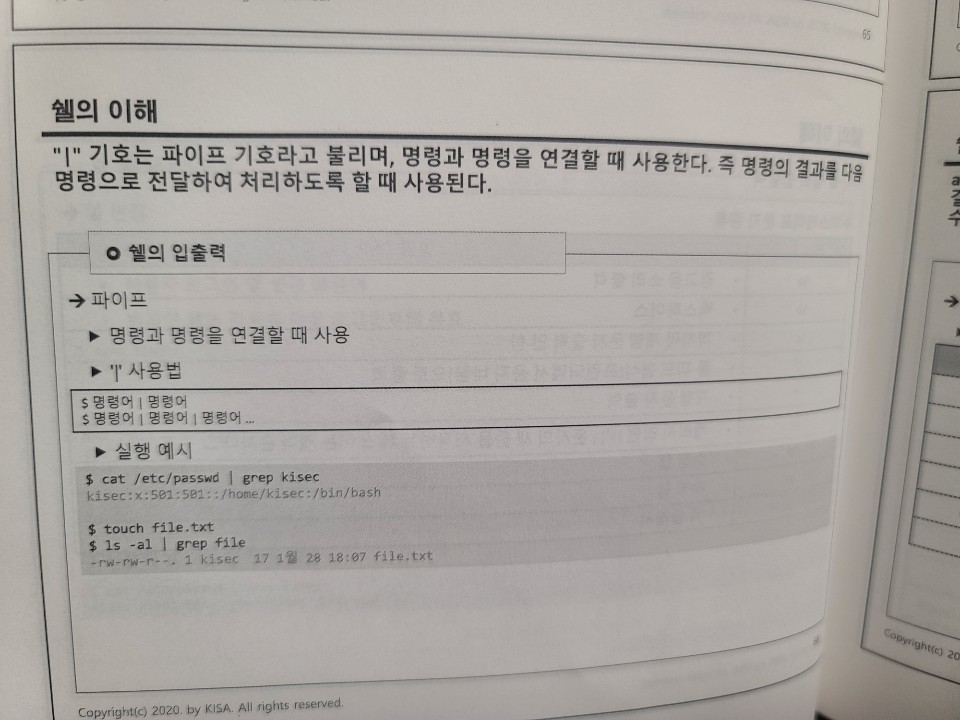
- 옵션

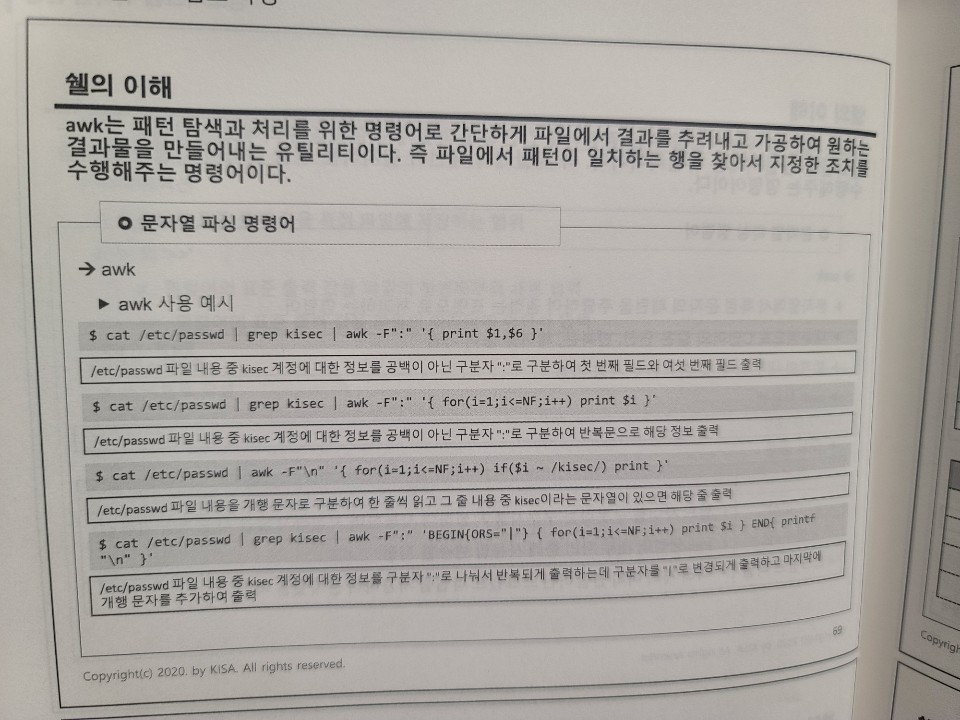
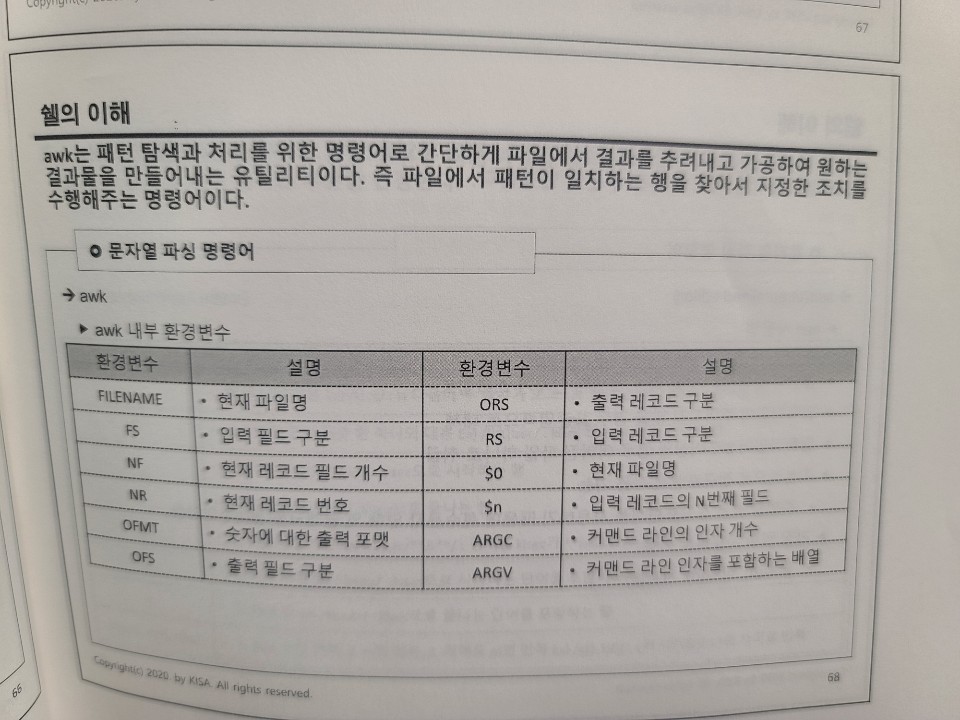
-n : 마지막에 따라오는 개행 문자(newline) 문자를 출력하지 않음

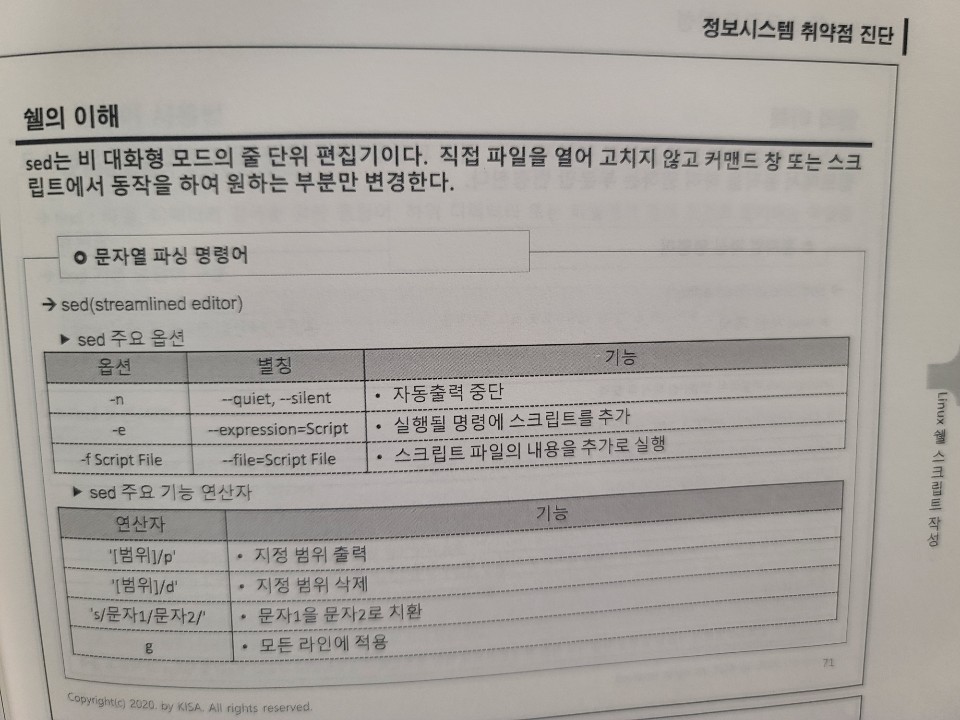
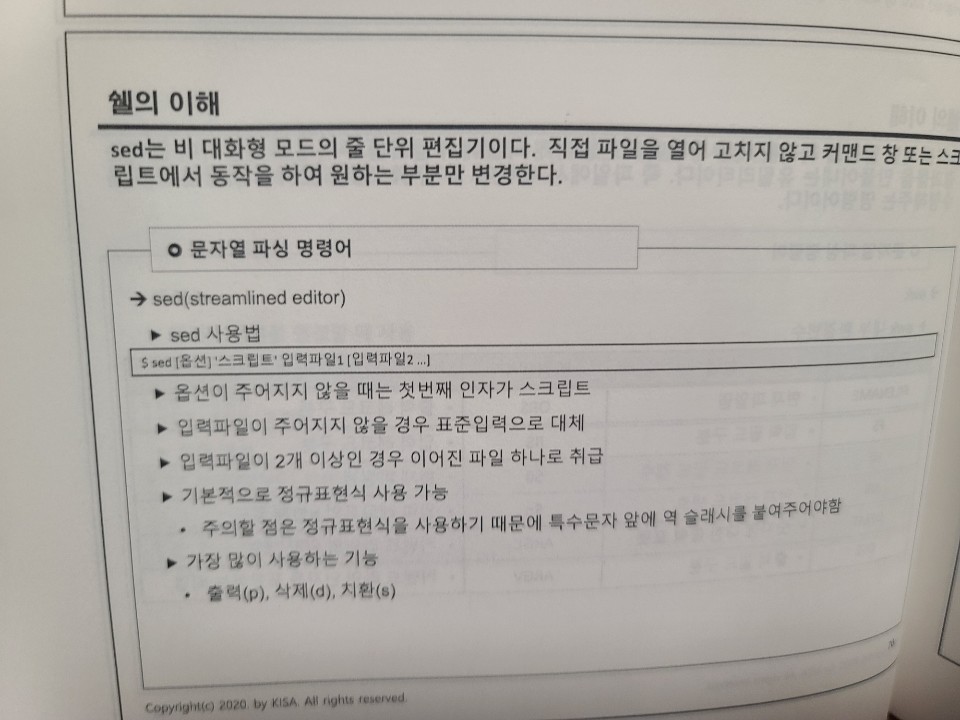
-e : 문자열에서 역 슬래시(\) 와 조합되는 이스케이프 문자를 인용부호(“)로 묶어 인식

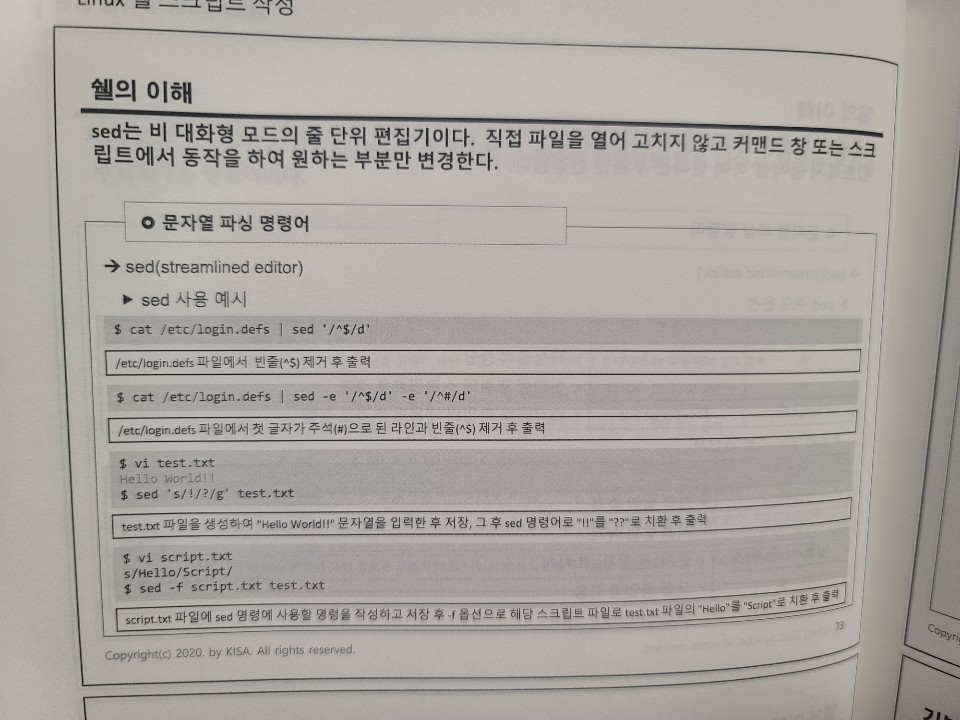
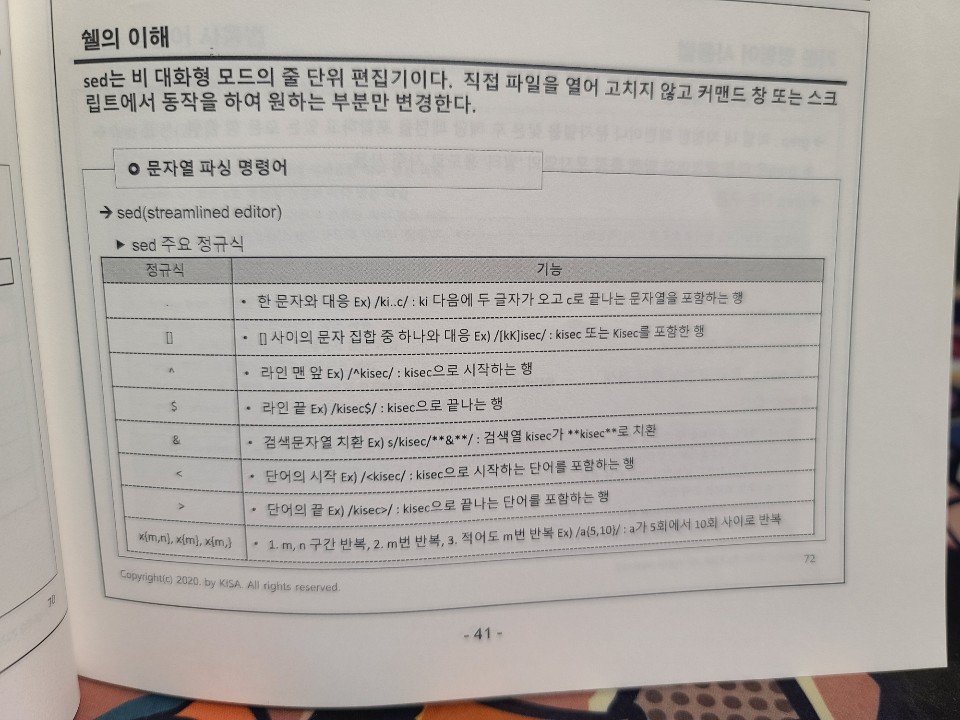


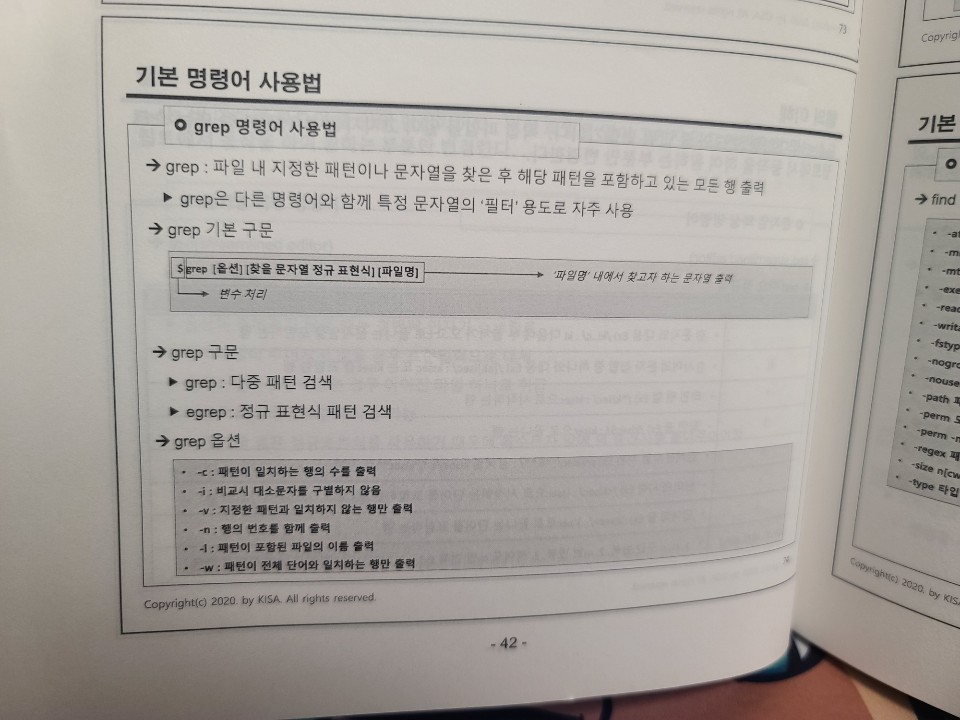


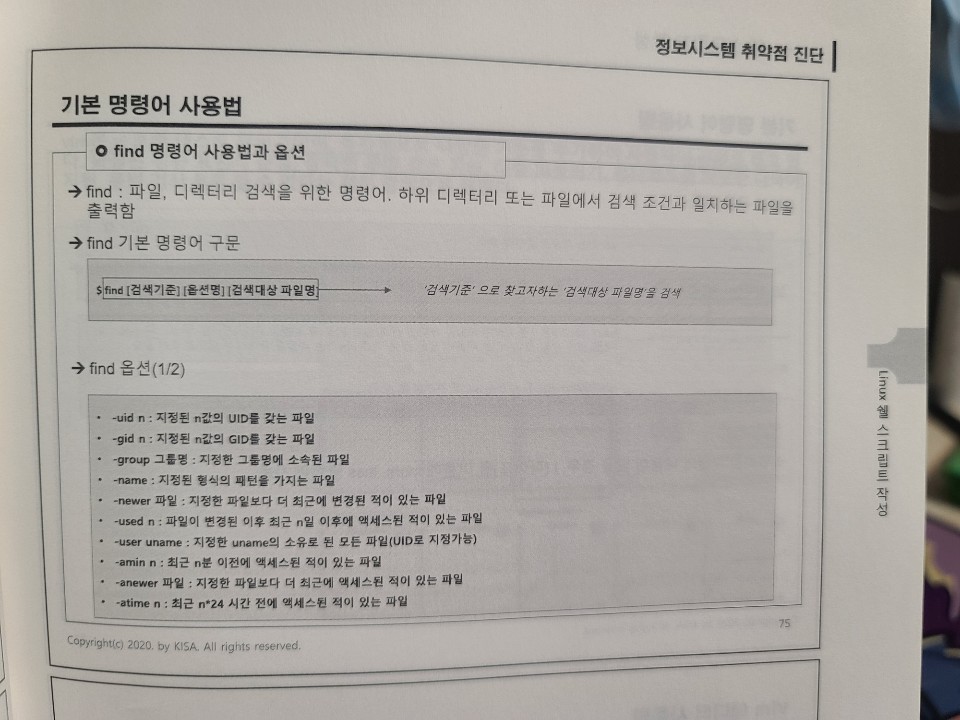


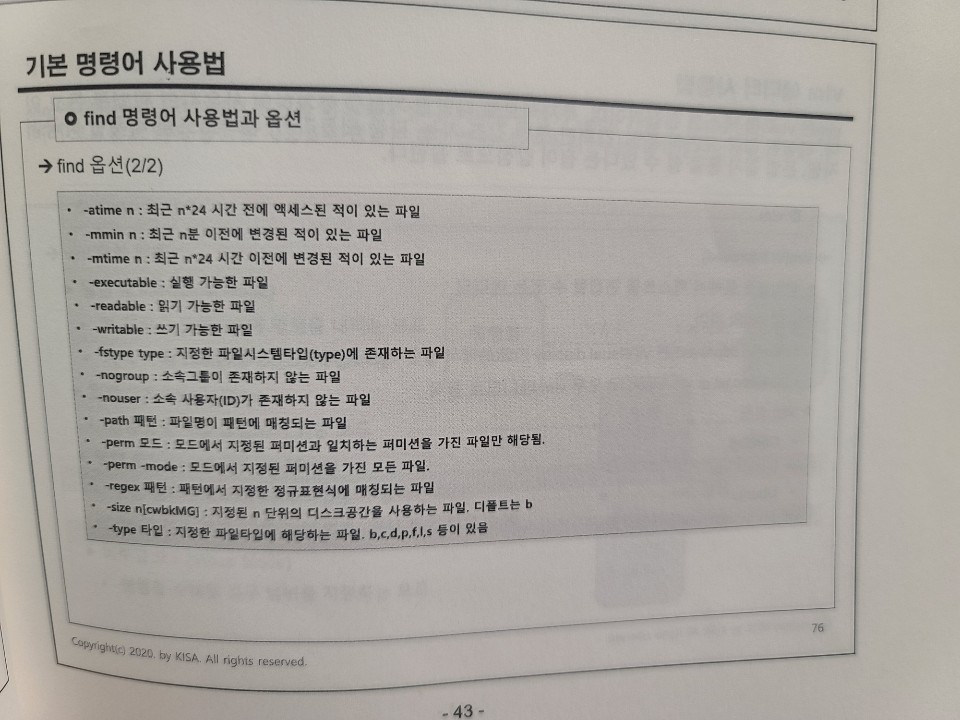


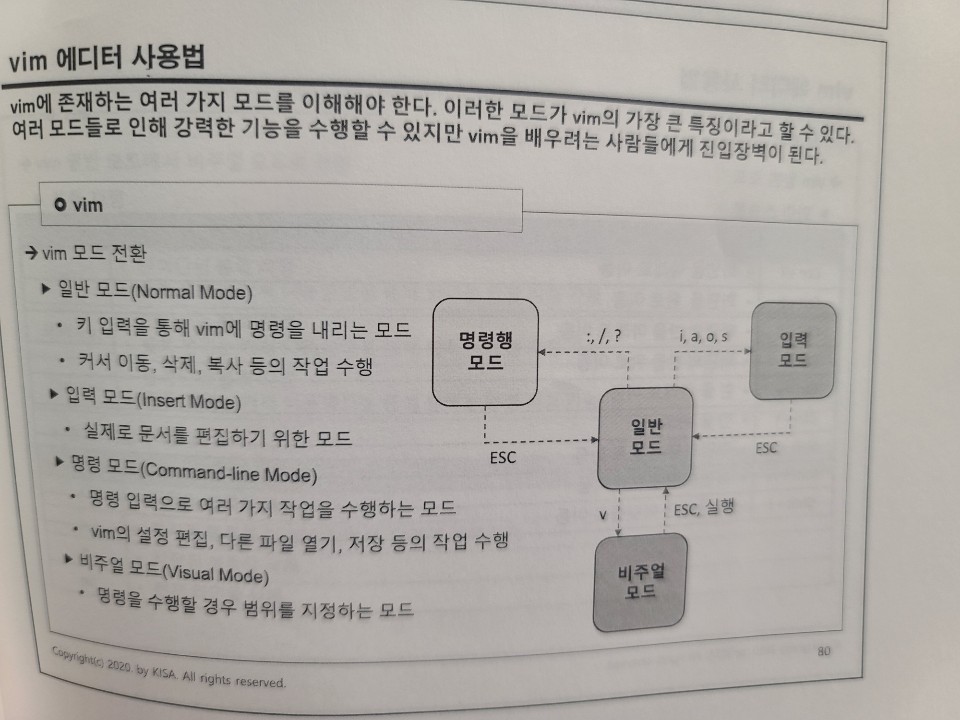


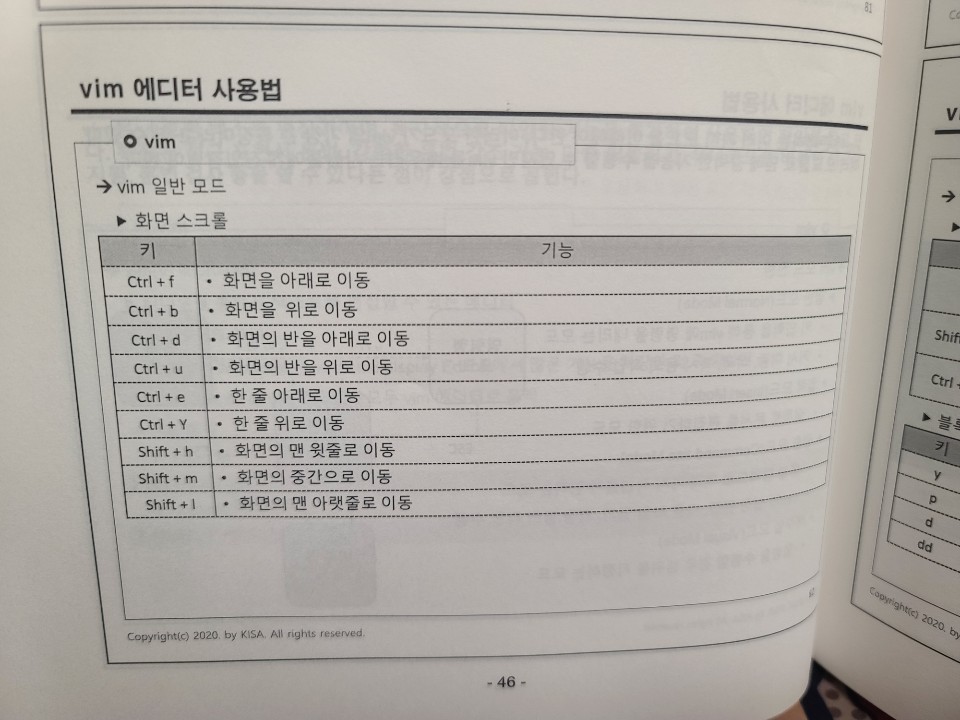
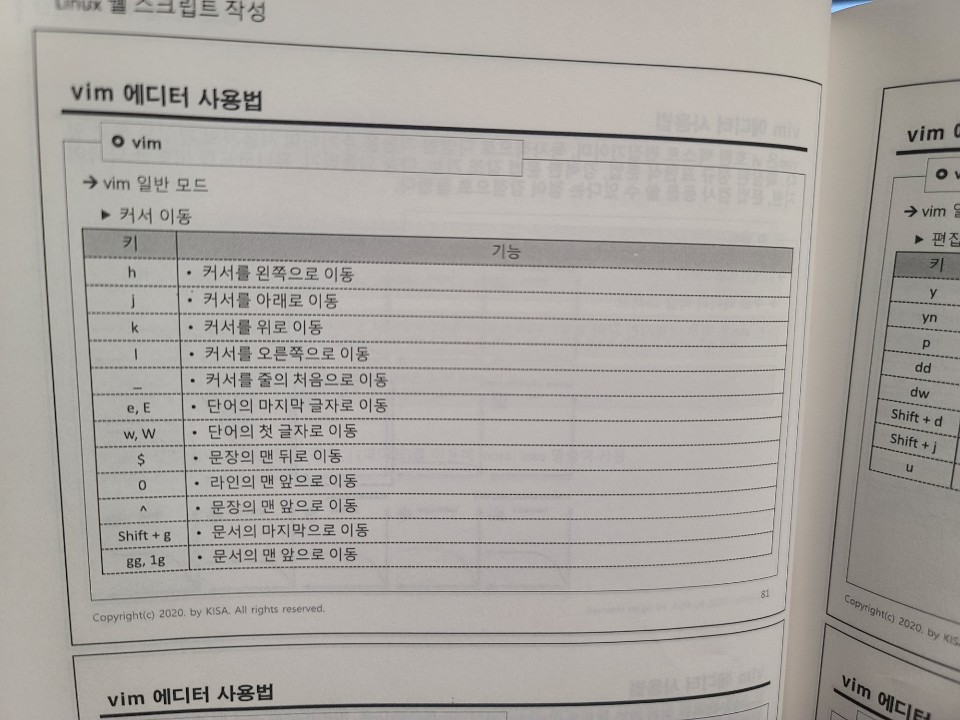


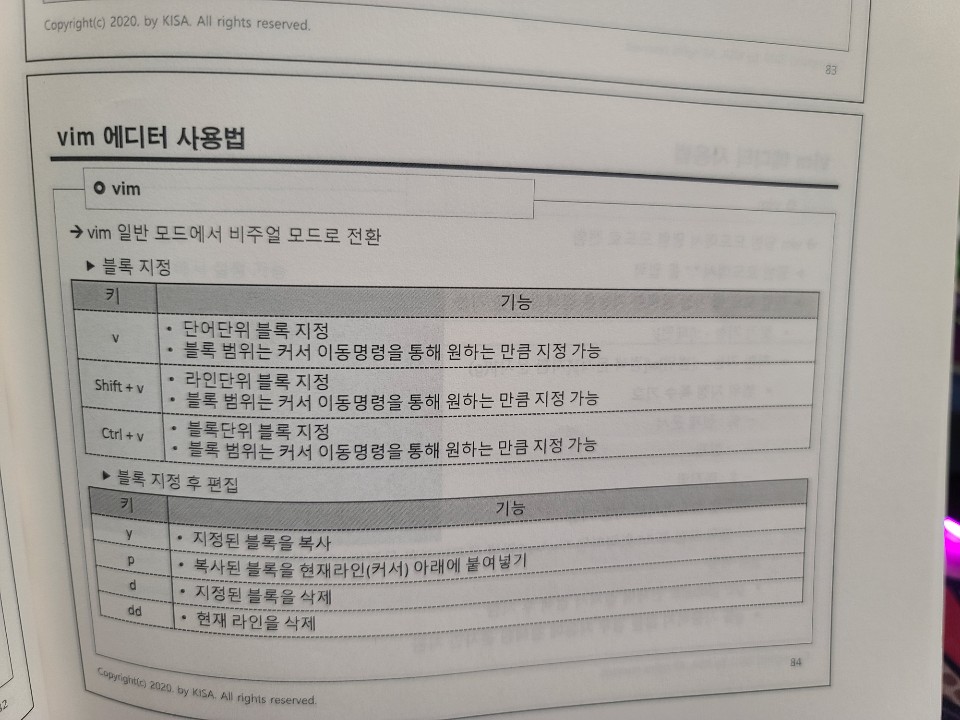
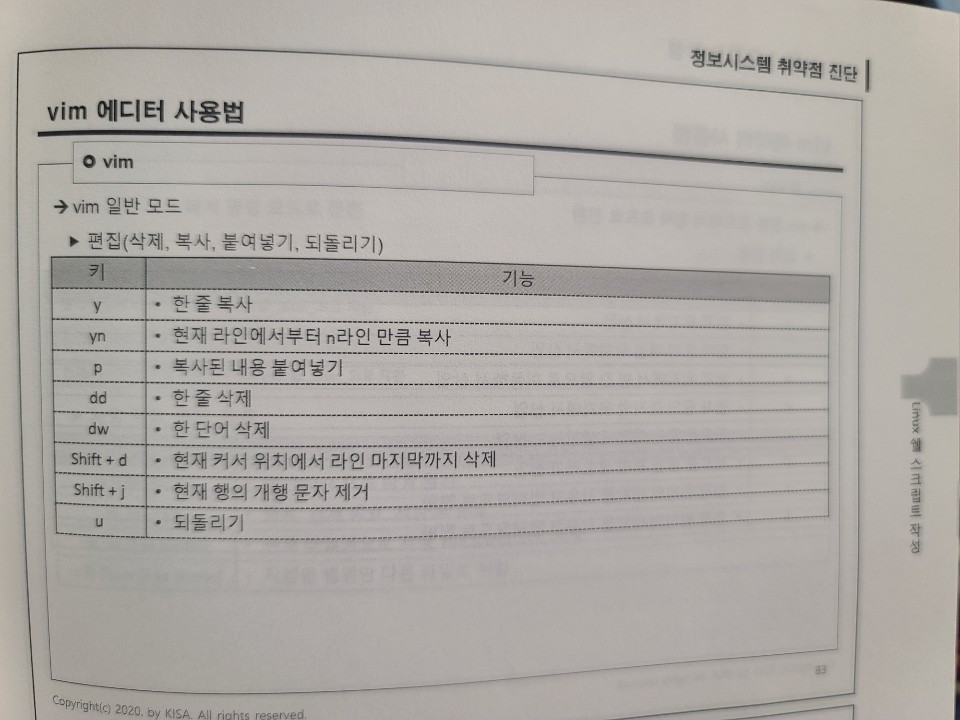


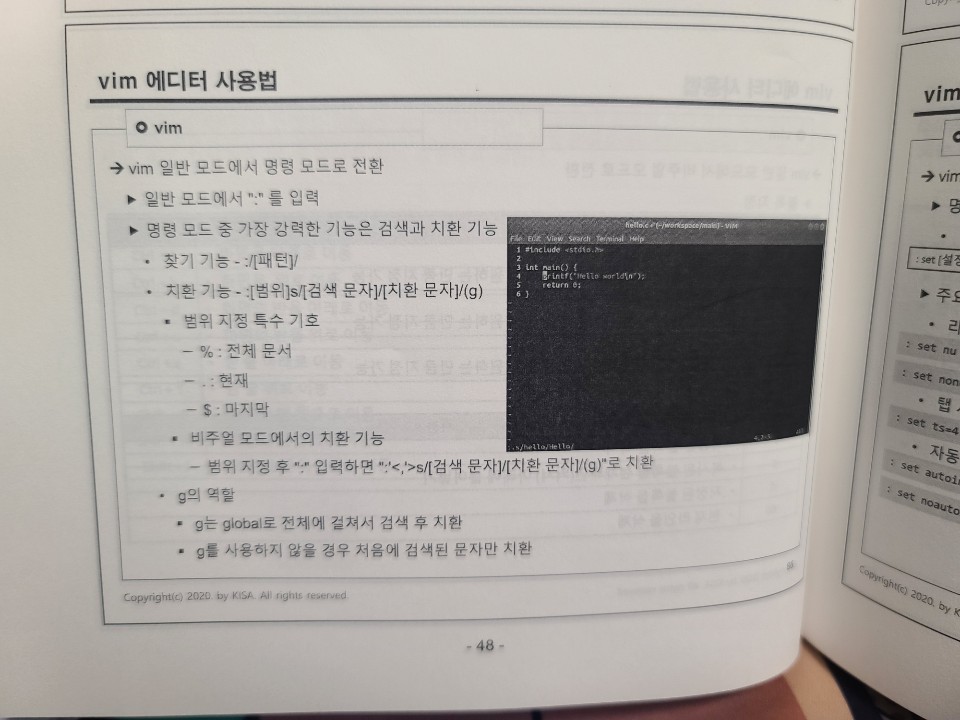
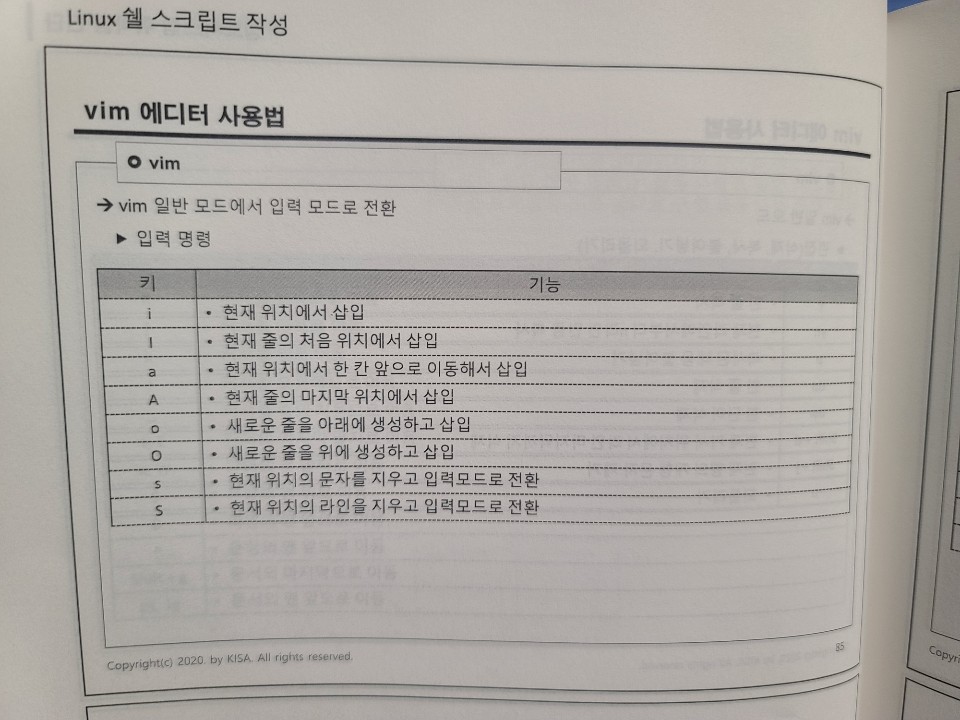


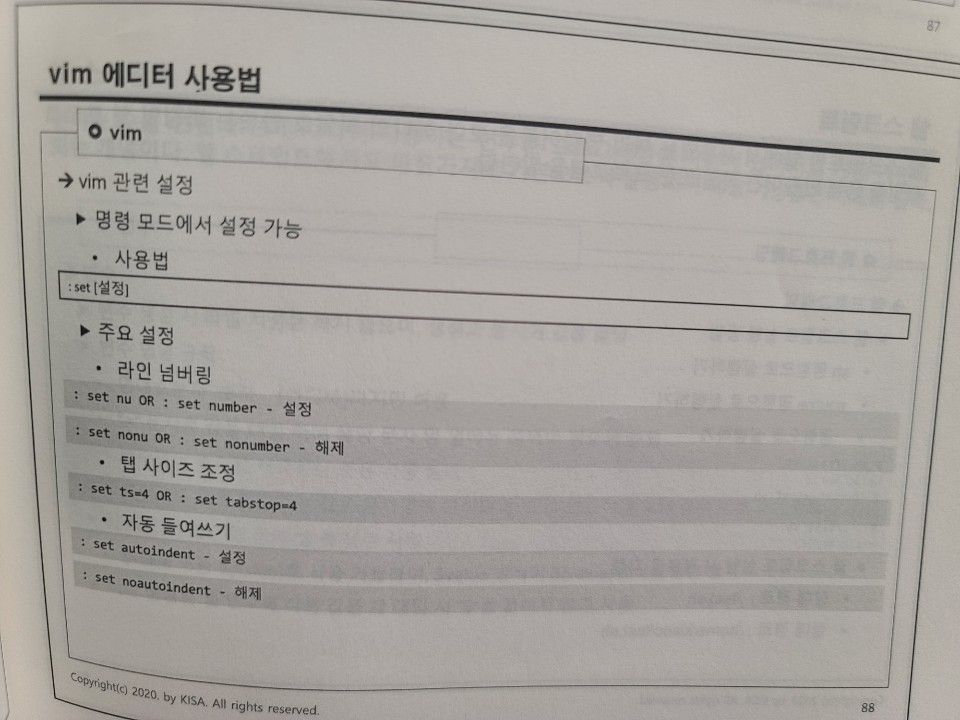
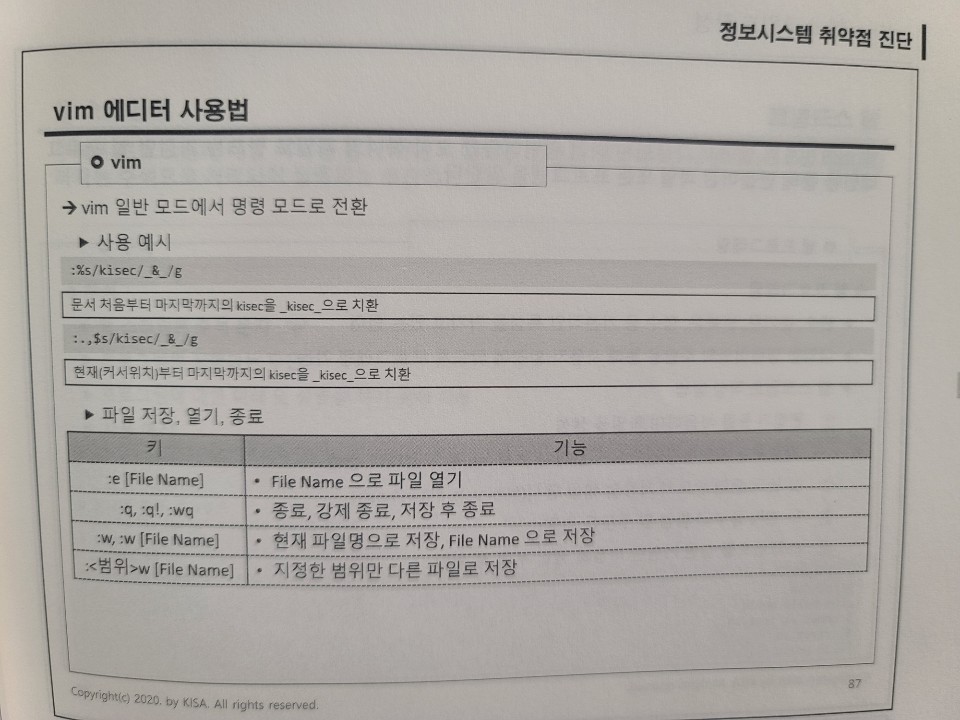


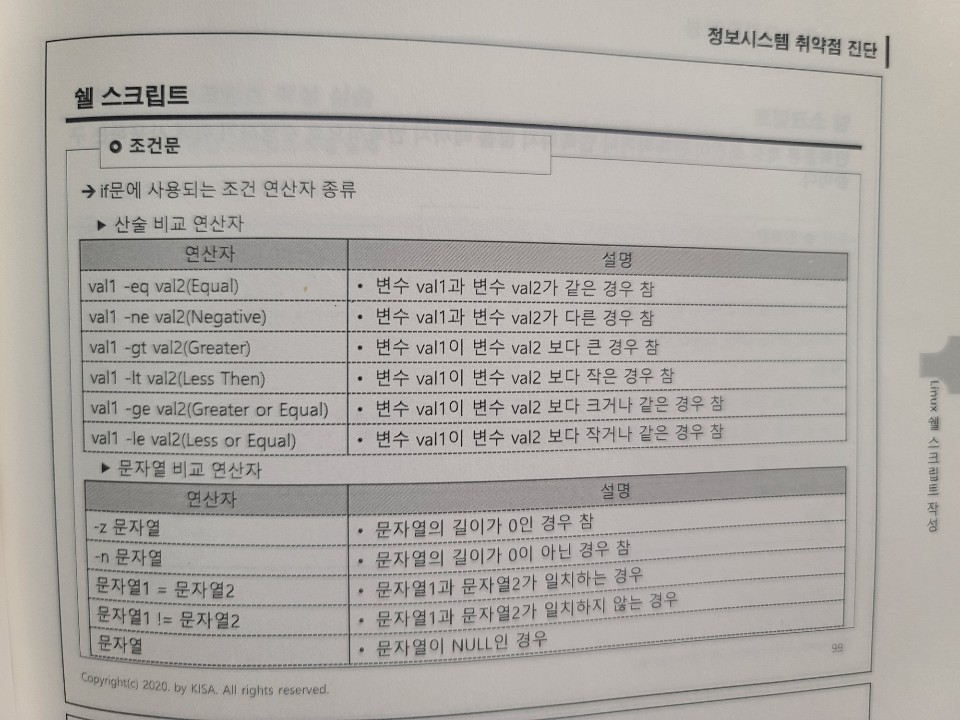
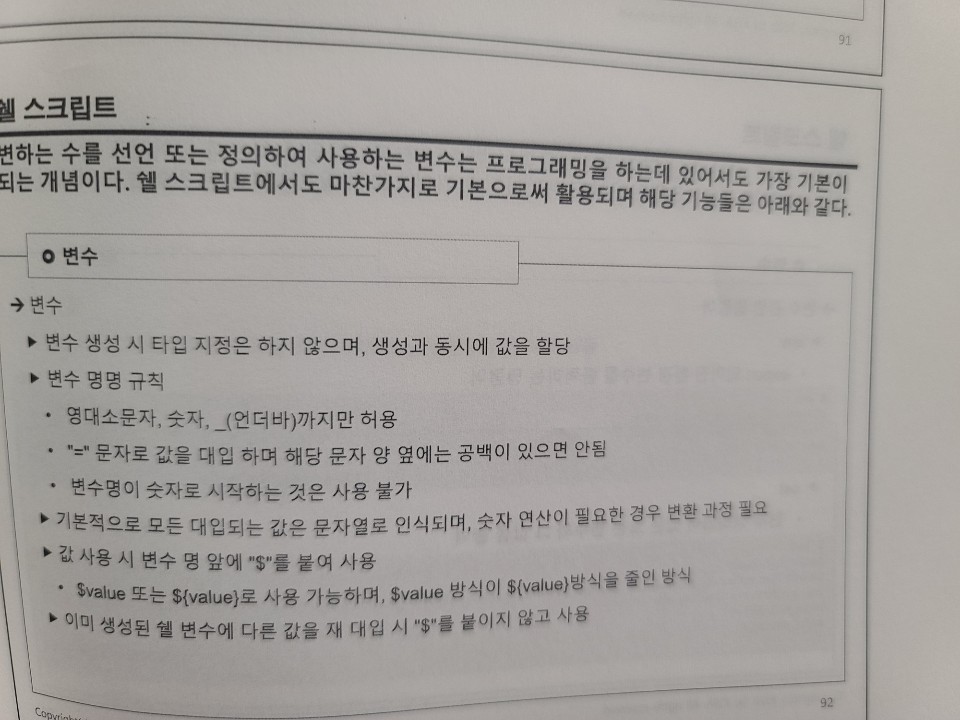


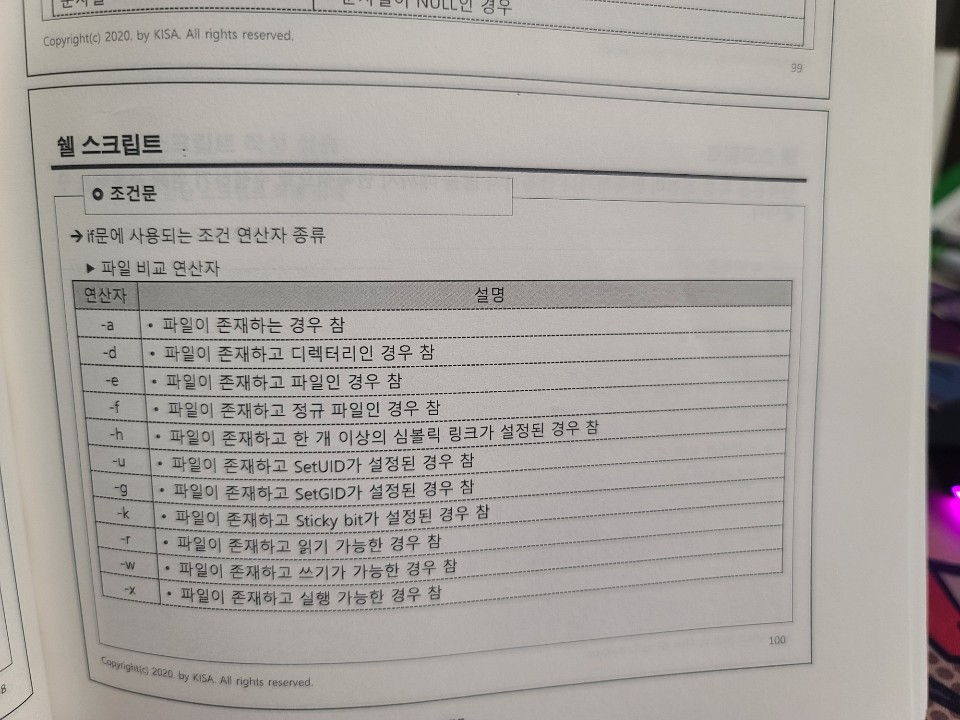












[주요정보통신기반시설]

#윈도우 서버 진단항목

1. 계정 관리

1) 기본계정은 이름을 변경해야 한다.

2) 패스워드는 강력한 패스워드를 사용해야 한다.

2. 서비스 관리

1) 불필요한 서비스는 중지 및 삭제해야 한다.

2) 필요한 서비스는 접근권한 및 접근제어가 설정되어야 한다.

3. 패치 관리

1) 최신 패치가 적용되어야 한다.

4. 로그 관리

1) 로그를 남기도록 설정되어야 한다.

2) 로그가 효율적으로 저장될 수 있도록 용량 관리되어야 한다.

3) 로그는 주기적으로 분석되어져야 한다.

5. 보안 관리

1) 윈도우 서버 운영체제에서 제공하는 보안설정이 유지되어야 한다.

6. DB 관리

1) 윈도우 서버이기 때문에 MS-SQL의 인증방식은 윈도우 서버 인증방식으로 운영되어야 한다.

[패스워드 최대 사용 기간]

1. net accounts

최대 암호 사용 기간 (일): 90

2. 윈도우 서버에 존재하는 계정들의 상세정보를 출력

Q) 윈도우 서버에 존재하는 모든 계정에 대해 NET USER [계정명]을 출력하는 스크립트를 만들어보세요!

주요 스크립트 명령어: for, token, skip, %%j(변수), net user, in 등을 이용할 수 있음

FOR token=1 skip=4 %%j ~~~~~~~~~~~ net user %%j >>txt

FOR token=2 skip=4 %%j ~~~~~~~~~~~ net user %%j >>txt

FOR token=3 skip=4 %%j ~~~~~~~~~~~ net user %%j >>txt

[윈도우 패스워드 크랙]

1. 윈도우 계정 및 패스워드 시스템 파일 덤프를 통해 Hash 값 확인

1) Windows\System32\Config\SAM (원본)

2) Windows\System32\Config\RegBack\SAM (백업)

3) Tool: openwall.com -> pwdump

4) CMD: PwDump7.exe >>C:\hash.txt

2. 윈도우 Hash 크랙을 이용하여 패스워드 평문 정보 획득

**Windows 서버 수동 진단 항목**

1. 계정 관리

**- Administrator 계정 이름 바꾸기**

-> 점검 내용 : 윈도우 최상위 관리자 계정인 Administrator의 계정명 변경 여부 점검

-> 점검 목적 : 윈도우 기본 관리자 계정인 Administrator의 이름을 변경하여, 잘 알려진 계정을 통한 악의적인 패스워드 추측 공격을 차단하고자 함

**판단 기준**

-> 양호 : Administrator Default 계정 이름을 변경한 경우

-> 취약 : Administrator Default 계정 이름을 변경하지 않은 경우

**점검 파일 위치 및 점검 방법**

1. 시작 – 실행 – cmd(명령 프롬프트)에서 “net user” 명령어 입력 후 계정 확인

**- 계정 잠금 임계값 설정**

-> 점검내용 : 계정 잠금 임계값의 설정 여부 점검

-> 점검목적 : 계정 잠금 임계값을 설정하여 공격자의 자유로운 자동화 암호 유추 공격을 차단하기 위함

**판단기준**

- 양호 : 계정 잠금 임계값이 5이하의 값으로 설정되어 있는 경우

- 취약 : 계정 잠금 임계값이 6이상의 값으로 설정되어 있는 겨우

**점검 파일 위치 및 점검 방법**

1. 시작 – 실행 – cmd(명령 프롬프트)에서 “net accounts” 명령어 입력 후 “잠금 임계값” 확인

**- 계정 잠금 기간 설정**

-> 점검내용 : 사용자 계정 감금 기간 정책 설정 여부 점검

-> 점검목적 : 로그인 실패 임계값 초과 시 일정 시간 동안 계정 잠금을 실시하여 공격자의 자유로운 암호 유추 공격을 차단하기 위함

**판단기준**

- 양호 : “계정 잠금 기간” 및 “계정 잠금 기간 원래대로 설정 기간”이 설정되어 있는 경우(60분 이상의 값으로 설정하기를 권고함)

- 취약 : “계정 잠금 기간” 및 “계정 잠금 기간 원래대로 설정 기간”이 설정되지 않은 경우

**점검 파일 위치 및 점검 방법**

1. 시작 – 실행 – cmd(명령 프롬프트)에서 “net accounts” 명령어 입력 후 “잠금 기간 ” 확인

**- 패스워드 최소 암호 길이**

-> 점검내용 : 패스워드 최소 암호 길이 정책 설정 여부 점검

-> 점검목적 : 암호에 필요한 최소 문자 수를 지정하여 강화된 패스워드를 사용하기 위함

**판단기준**

- 양호 : 최소 암호 길이가 8문자 이상으로 설정되어 있는 경우

- 취약 : 최소 암호 길이가 설정되지 않았거나 8문자 미만으로 설정되어 있는 경우

**- 패스워드 최대 사용 기간**

­-> 점검내용 : 패스워드 최대 사용 기간 정책의 설정 여부 점검

-> 점검목적 : 암호가 유효한 최대 날짜를 설정하여 이 날짜가 경과된 사용자는 암호를 변경하도록 하여 암호 크래킹의 가능성을 낮추고, 불법으로 획득한 암호의 무단 사용을 방지하고자 함

**판단 기준**

- 양호 : 최대 암호 사용 기간이 90일 이하로 설정되어 있는 경우

- 취약 : 최대 암호 사용 기간이 설정되지 않았거나 90일을 초과하는 값으로 설정된 경우

**점검 파일 위치 및 점검 방법**

1. 시작 – 실행 – cmd(명령 프롬프트)에서 “net accounts” 명령어 입력 후 “최대 암호 사용 기간” 확인

**- 최근 암호 기억**

-> 점검내용 : 최근 암호 기억 정책 설정 여부 점검

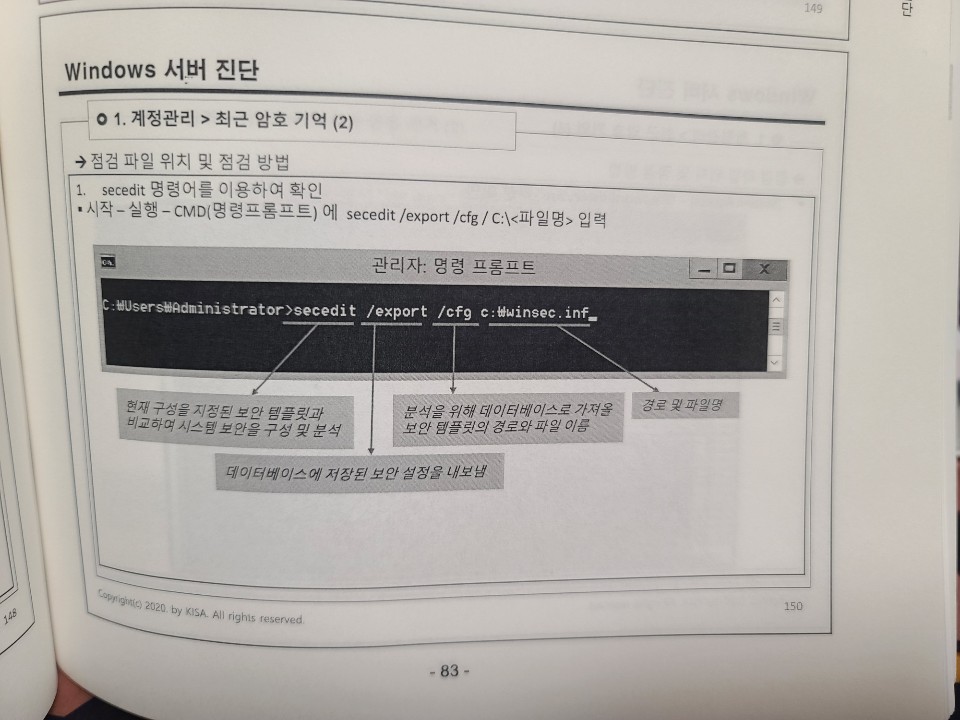
-> 점검목적 : 사용자가 현재 암호 또는 최근에 사용했던 암호와 동일한 새 암호를 만드는 것을 방지하기 위함

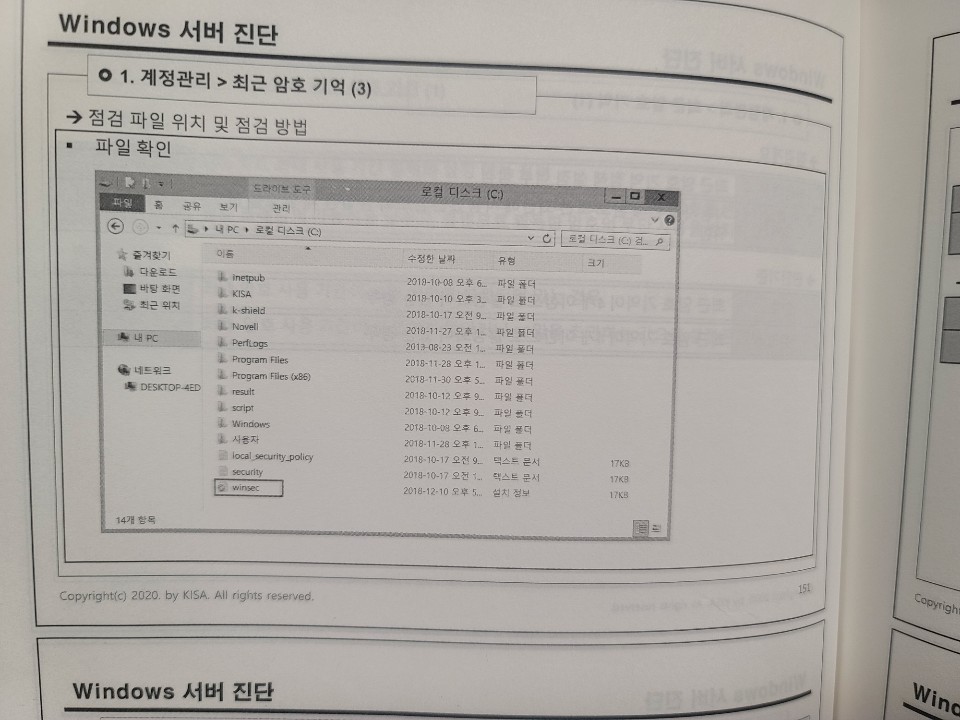
**판단기준**

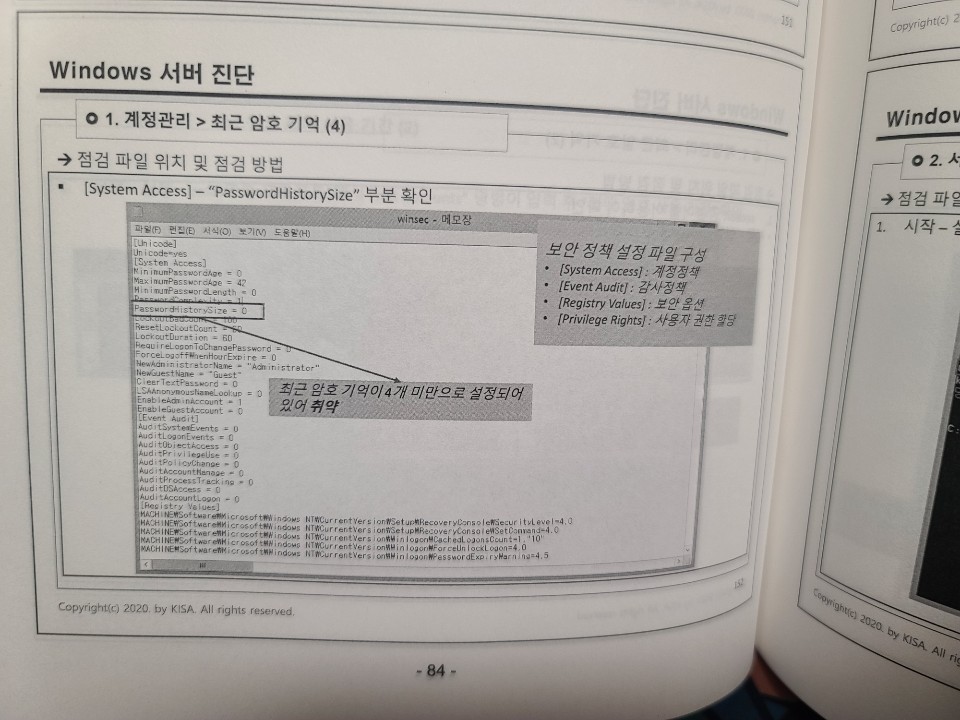
- 양호 : 최근 암호 기억이 4개 이상으로 설정되어 있는 경우

- 취약 : 최근 암호 기억이 4개 미만으로 설정되어 있는 경우

**점검 파일 위치 및 점검 방법**







2. 서비스 관리

- 하드디스크 기본 공유 제거

-> 점검내용 : 하드디스크 기본 공유 제거 여부 점검

-> 점검목적 : 하드디스크 기본 공유를 제거하여 시스템 정보 노출을 차단하고자 함

**판단기준**

- 양호 : 레지스트리의 AutoShareServer가 0이며 기본 공유가 존재하지 않는 경우

- 취약 : 레지스트리의 AutoShareServer가 1이거나 기본 공유가 존재하는 경우

**점검 파일 위치 및 점검 방법**

1. 시작 – 실행 – cmd(명령 프롬프트)에서 “net share” 명령어 입력 후 “기본 공유” 확인

- FTP 서비스 구동 점검

-> 점검내용 : 시스템 내 FTP 서비스 구동 여부 점검

-> 점검목적 : 인증 정보가 기본적으로 평문전송 되는 취약한 프로토콜인 FTP의 사용을 제한하여 네트워크 보안성을 높이고자 함

**판단기준**

- 양호 : FTP 서비스를 사용하지 않는 경우 또는 secure FTP 서비스를 사용하는 경우

- 취약 : FTP 서비스를 사용하는 경우

**점검 파일 위치 및 점검 방법**

1. 시작 – 실행 – cmd(명령프롬프트)에서 “net start” 명령어 입력 후 서비스 확인

- FTP 서비스 실행 확인 -> 가이드에서는 FTP 서비스를 사용 중이면 취약이라 판단 하지만 1shop File Server 에서는 필요에 의해 사용 중 이므로 양호

- SNMP 서비스 구동 점검

-> 점검내용 : SNMP 서비스 구동 여부 점검

-> 점검목적 : 취약한 SNMP 서비스를 비활성화하여 시스템의 주요정부 유출 및 불법 수정을 방지하기 위함

**판단기준**

- 양호: SNMP 서비스를 사용하지 않는 경우

- 취약 : SNMP 서비스를 사용하는 경우

**점검 파일 위치 및 점검 방법**

1. 시작 – 실행 – CMD에서 “net start” 명령어 입력 후 서비스 확인

Find /I : 지정한 문자열을 대/소문자 구별하지 않고 찾음

- SNMP 서비스 커뮤니티스트링의 복잡성 설정

-> 점검내용 : SNMP 서비스 커뮤니티 스트링(Community String 적절성 점검)

-> 점검목적 : SNMP에서 일종의 패스워드로 사용하는 Community String을 유추할 수 없는 복잡한 값으로 변경하여 불필요한 시스템 정보 노출을 차단하기 위함

**판단기준**

- 양호 : SNMP 서비스를 사용하지 않거나 Community String이 public, private이 아닌 경우

- 취약 : SNMP 서비스를 사용하며, Community String이 public, private인 경우

**점검 파일 위치 및 점검 방법**

1. 시작 – 실행 - cmd에서

reg query “HKLM\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ValidCommunities”

3. 패치관리

4. 로그 관리

- 정책에 따른 시스템 로깅 설정

-> 점검내용 : 시스템 로깅 설정 여부 및 적절성 점검

-> 점검목적 : 적절한 로깅 설정으로 유사 시 책임 추적을 위한 로그가 확보될 수 있게 하기 위함

**판단기준**

- 양호 : 감사 정책 권고 기준에 따라 감사 설정이 되어 있는 경우

- 취약 : 감사 정책 권고 기준에 따라 감사 설정이 되어 있지 않는 경우

**점검 파일 위치 및 점검 방법**

­1. secedit 명령어를 이용하여 확인

시작 – 실행 – cmd에 secedit /export /cfg C:\<파일명> 입력

**점검 파일 위치 및 점검 방법**

- 파일 확인

- [Event Audit] – “AuditLoginEvents, AuditPrivilegeUse, AuditPolicyChange, AuditDSAccess, AuditAccountLogon, AuditAccountManage” 부분 확인

5. 보안 관리

- 화면보호기 설정

-> 점검내용 : 시스템 화면보호기 설정 여부 점검

-> 점검목적 : 사용자가 일정 시간 동안 아무런 작업을 수행하지 않을 경우 자동으로 로그 오프 되거나 워크 스테이션이 잠기도록 설정하여, 유휴 시간 내 불법적인 시스템 접근을 차단하기 위함

**판단기준**

- 양호 : 화면 보호기를 설정하고 대기 시간이 10분 이하의 값으로 설정되어 있으며, 화면 보호기 해제를 위한 암호를 사용하는 경우

- 취약 : 화면 보호기가 설정되지 않았거나 암호를 사용하지 않은 경우 또는, 화면 보호기 대기 시작이 10분을 초과한 값으로 설정되어 있는 경우

**점검 파일 취이 및 점검 방법**

1. reg 명령어로 해당 레지스트리 값 찾기

Reg query “HKCU\Control Panel\Desktop” /v “ScreenSaveActive” -> 화면보호기 설정 여부 확인

Reg query “HKCU\Control Panel\Desktop” /v “ScreenSaverIsSecure” -> 화면보호기 해제 암호 사용 여부

Reg query “HKCU\Control Panel\Desktop” /v “ScreenSaveTimeOut” -> 대기시간 확인

6. DB 관리

**Windows Batch File**

**기본 개념**

- 윈도우 환경에서 쓰이며 명령 인터프리터에 의해 실행되게 명령어들로 작성한 텍스트 파일

- 유닉스 계열 운영체제에서 쉘 스크립트의 역할

**적용 환경**

- COMMAND.COM(도스용 쉘) 또는 cmd.exe(텍스트 모드와 윈도우 NT용)와 같은 쉘 프로그램이 파일을 읽어 줄 단위로 실행

**장단점**

­- 취약점 진단처럼 많은 점검항목을 자동/연속으로 실행할 때 유용

- 시스템 관리자가 반복된 업무를 최소화시켜 주는데 유용함

- 다만, 구문과 명령어가 제한되어 일반 목적의 프로그래밍에 부적합

**배치 파일(Batch File) 작성 시 고려사항**

- 확장자 및 편집기

-> 확장자는 \*.bat로 설정

-> 다양한 문서 편집기를 사용해도 무방하며, 일반적으로 Notepad++ 등을 사용할 수 있음

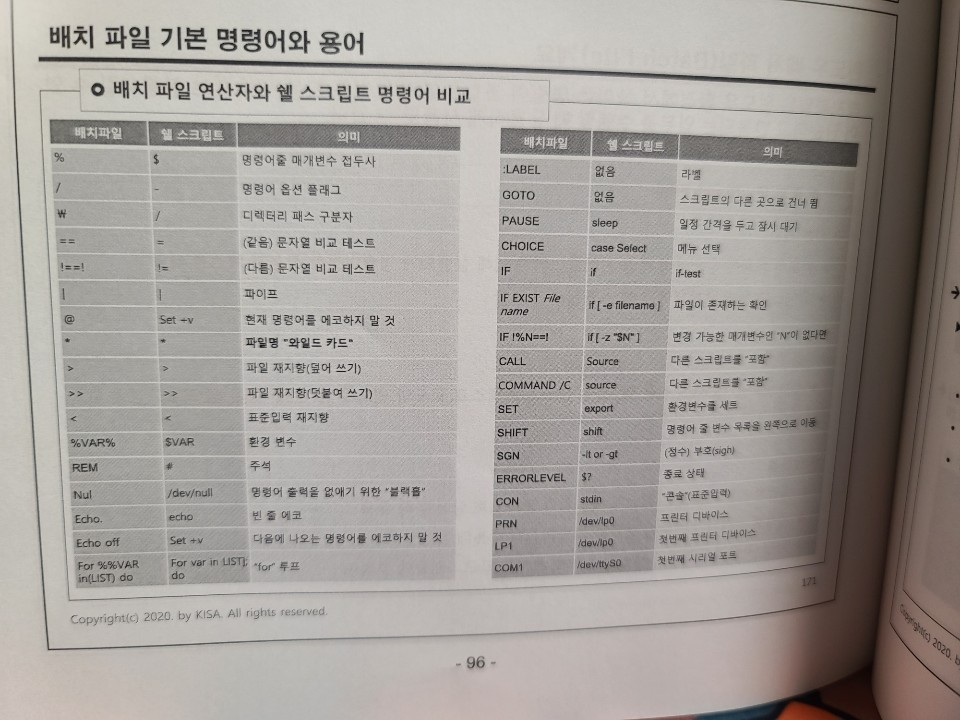
**파일 형태**

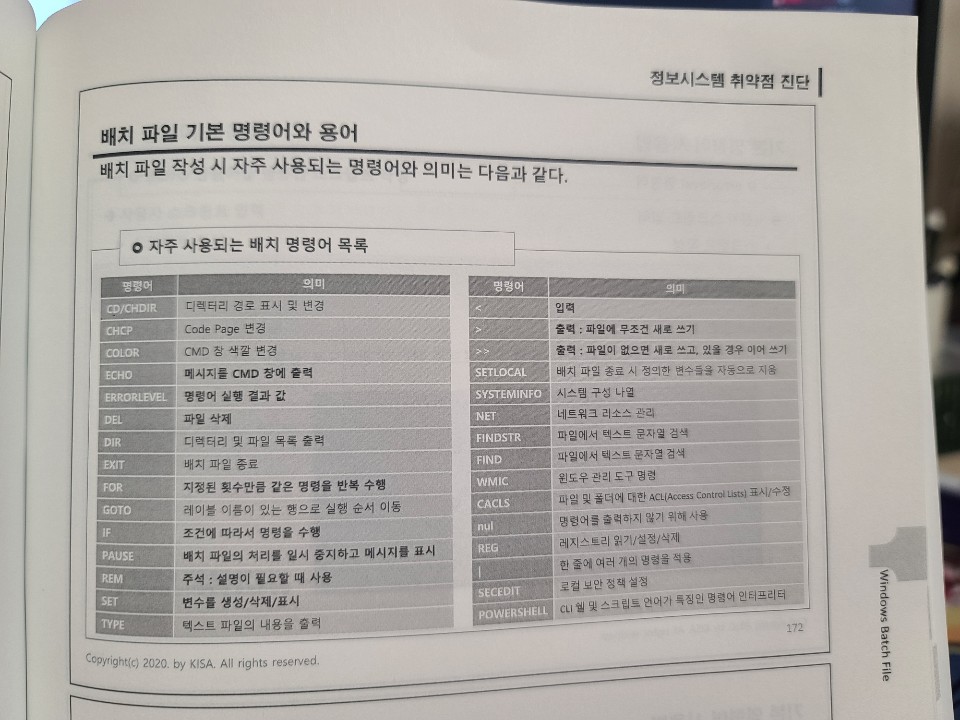
-> ASCII 문장(아스키) 파일

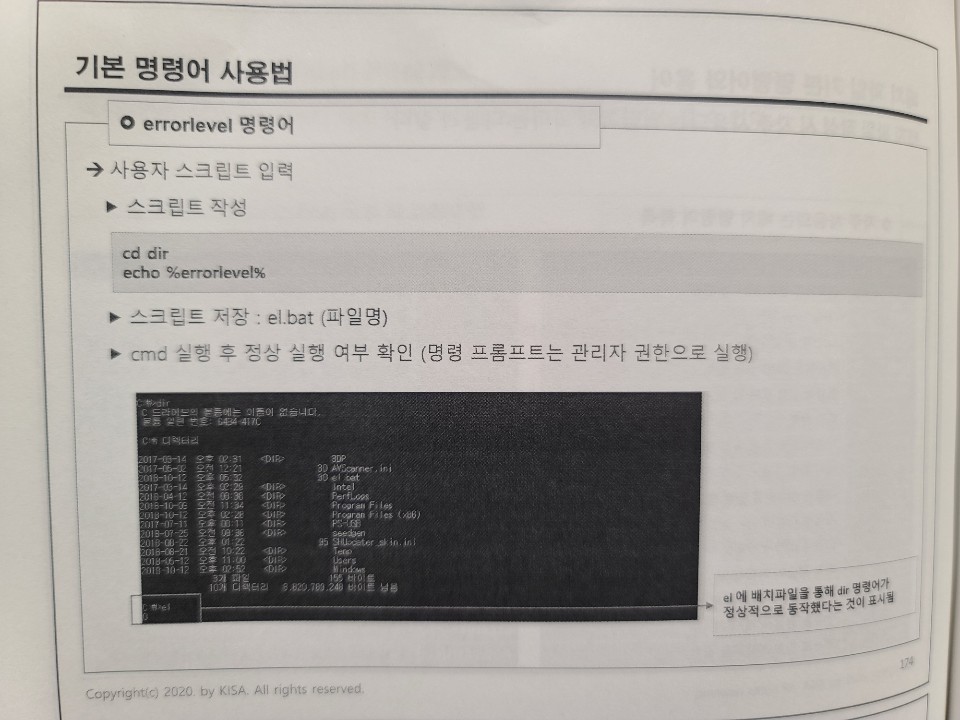
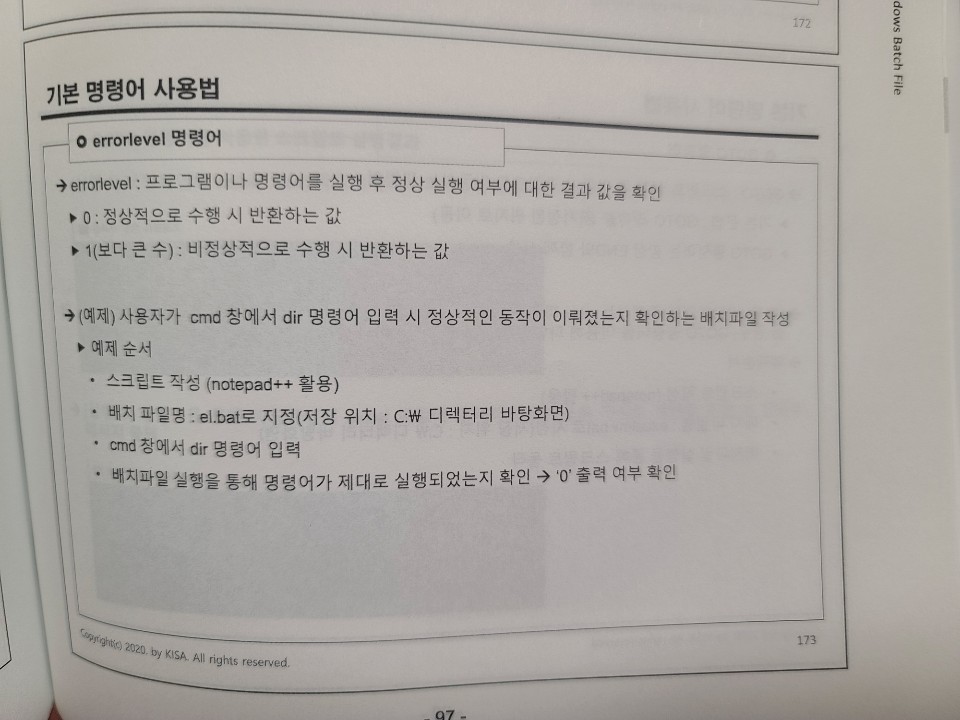
**고려 사항**

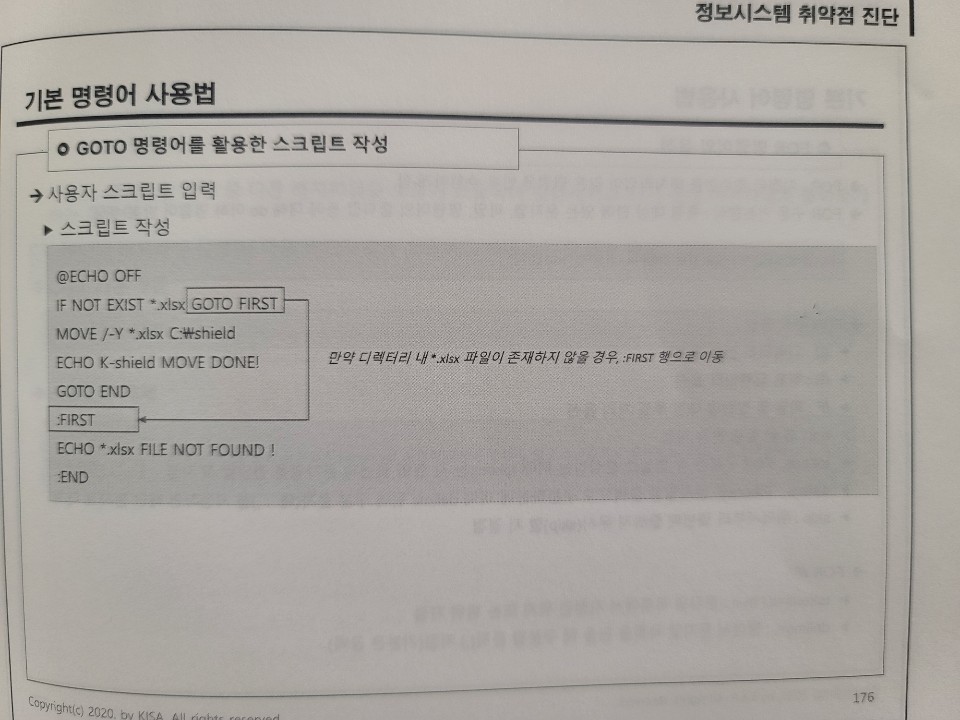
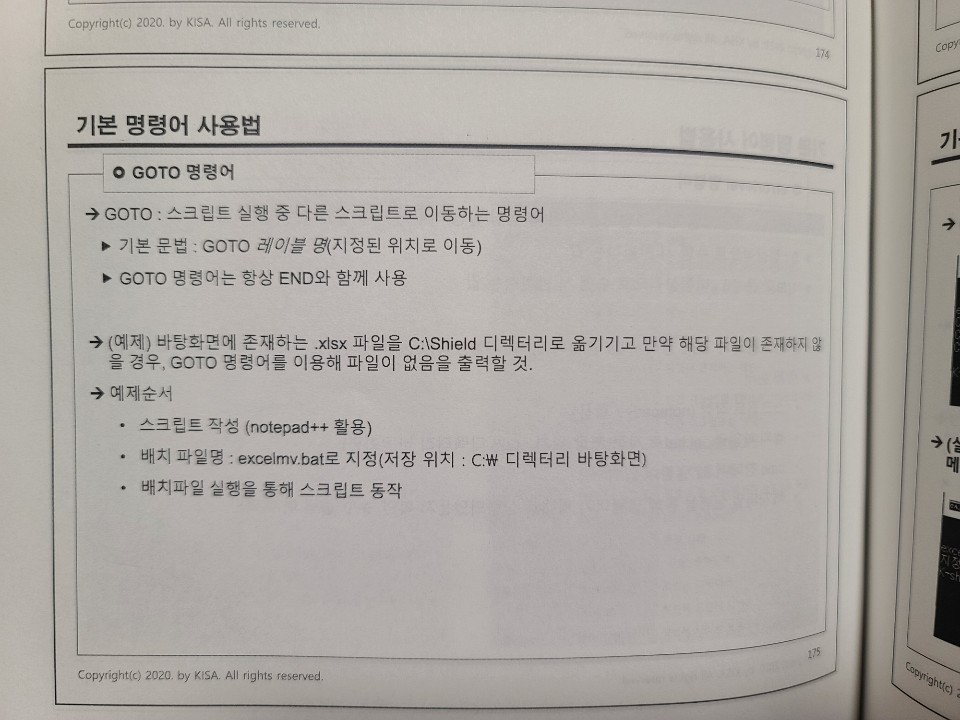
-> \*.bat 확장자를 이용하는 이유는 다른 \*.exe, \*.com과 같은 확장자와의 충돌을 피하기 위한 것임

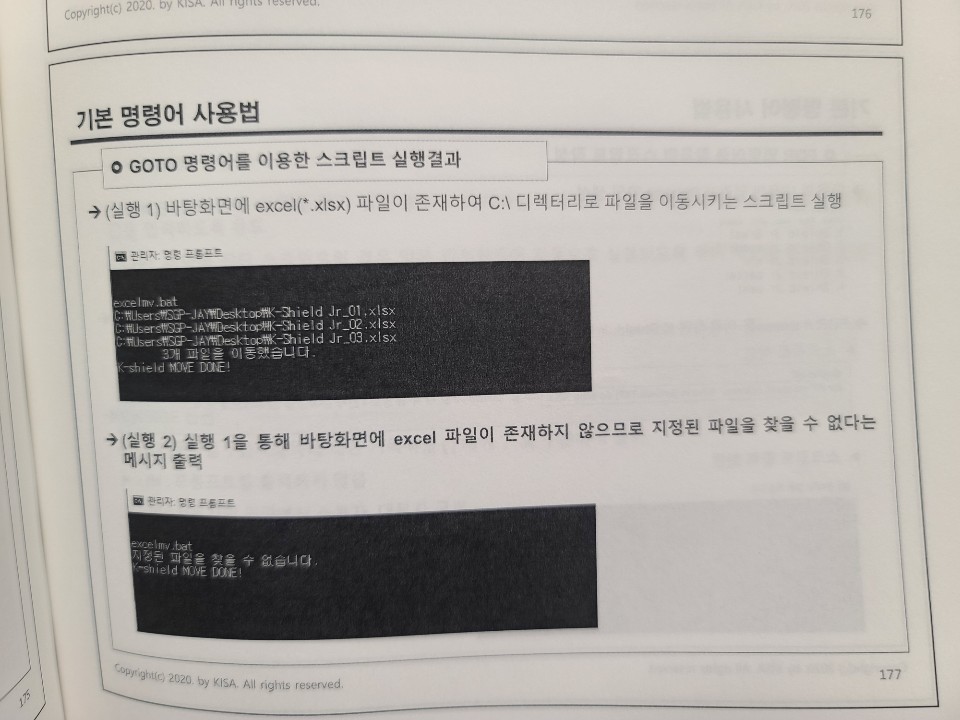
-> 파일명은 도스의 내부 명령어 이름을 피할 것(예 : dir.bat, date.bat, time.bat 등)

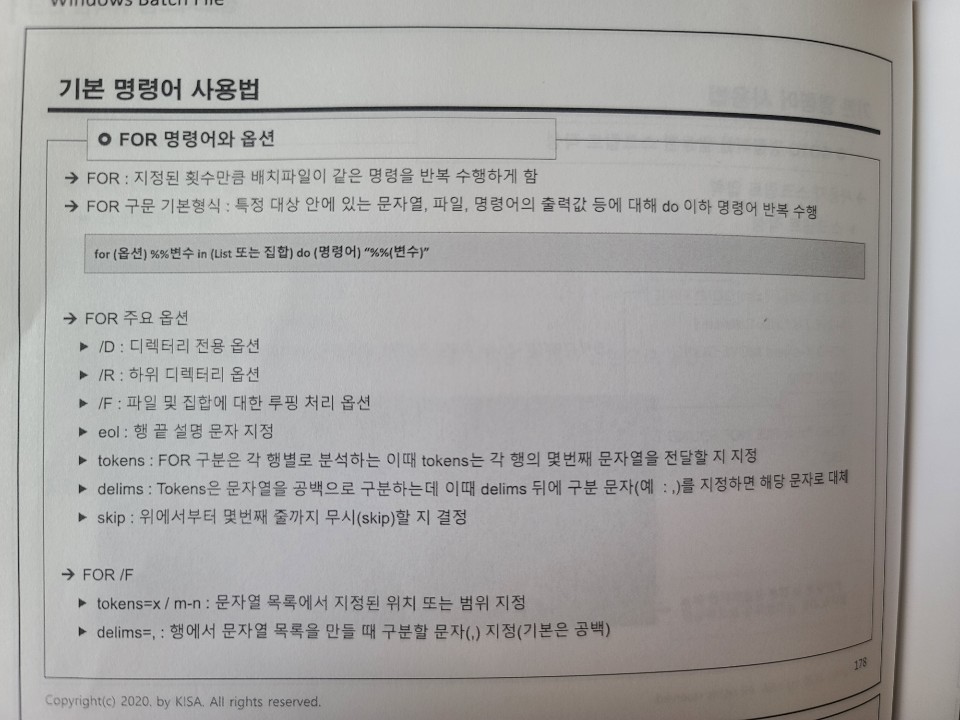


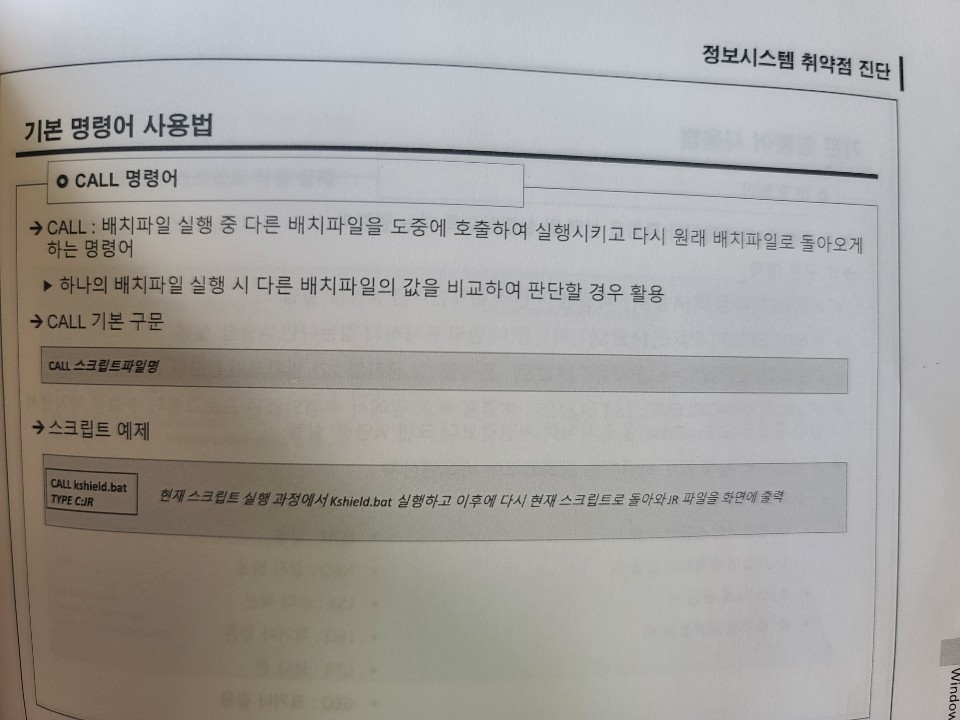
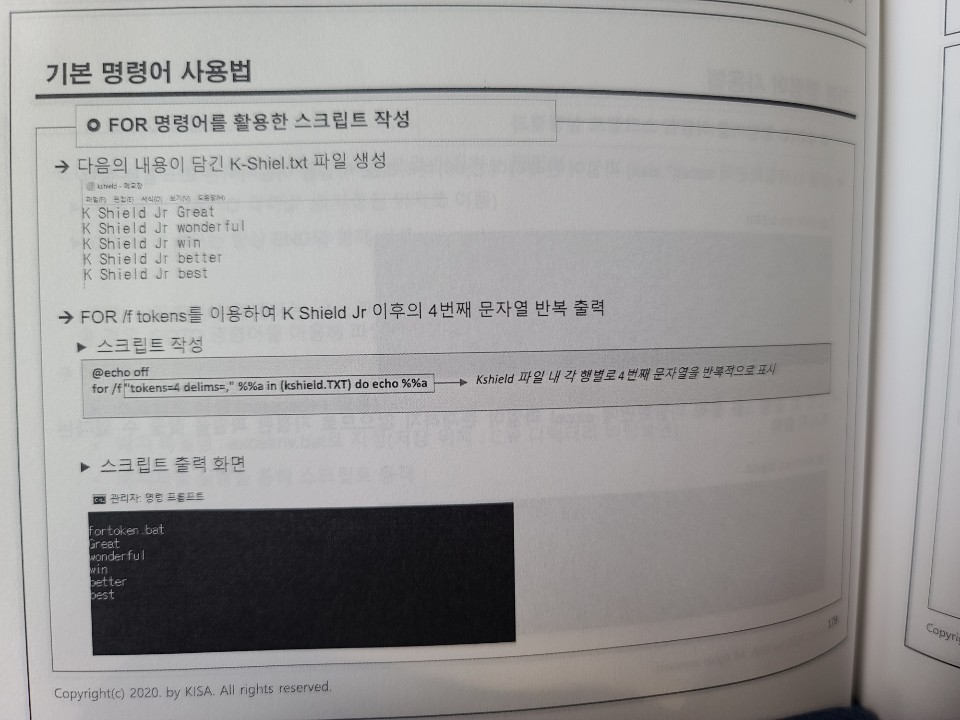


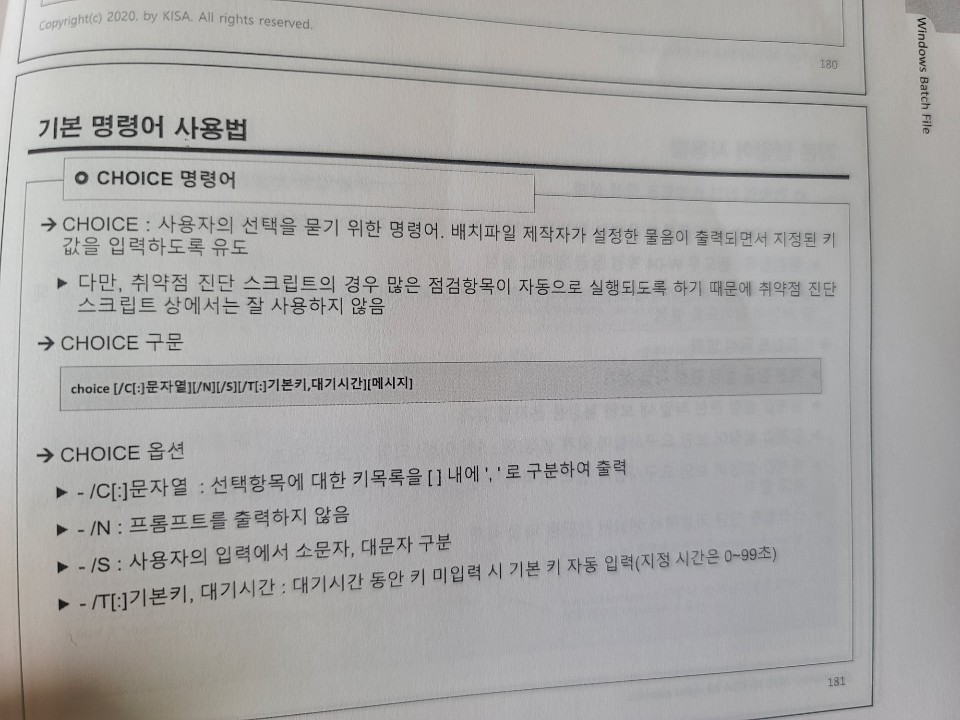


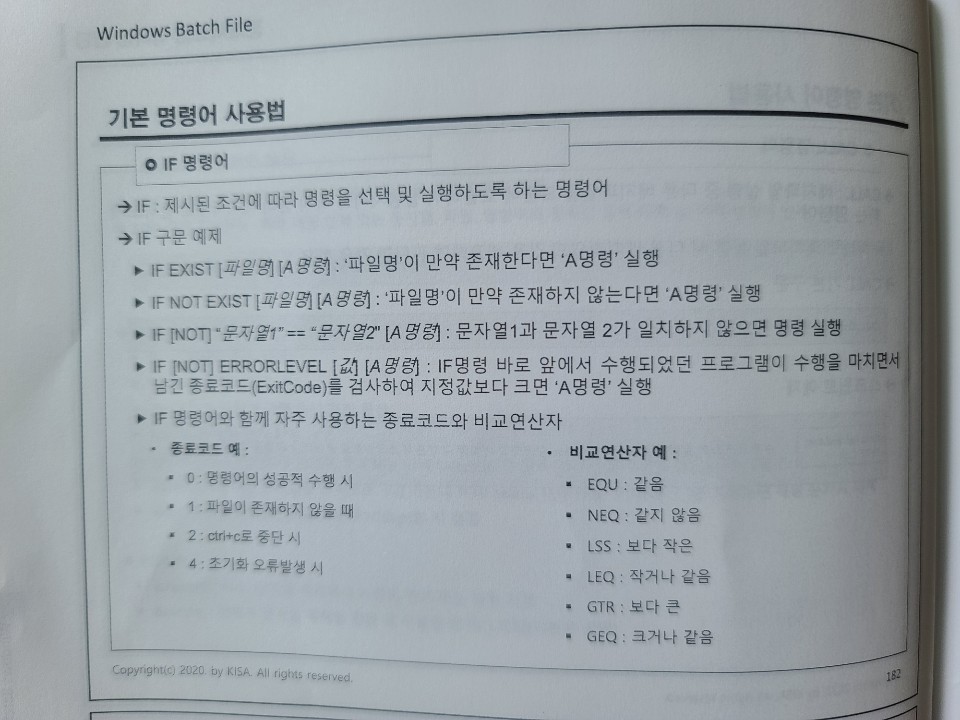


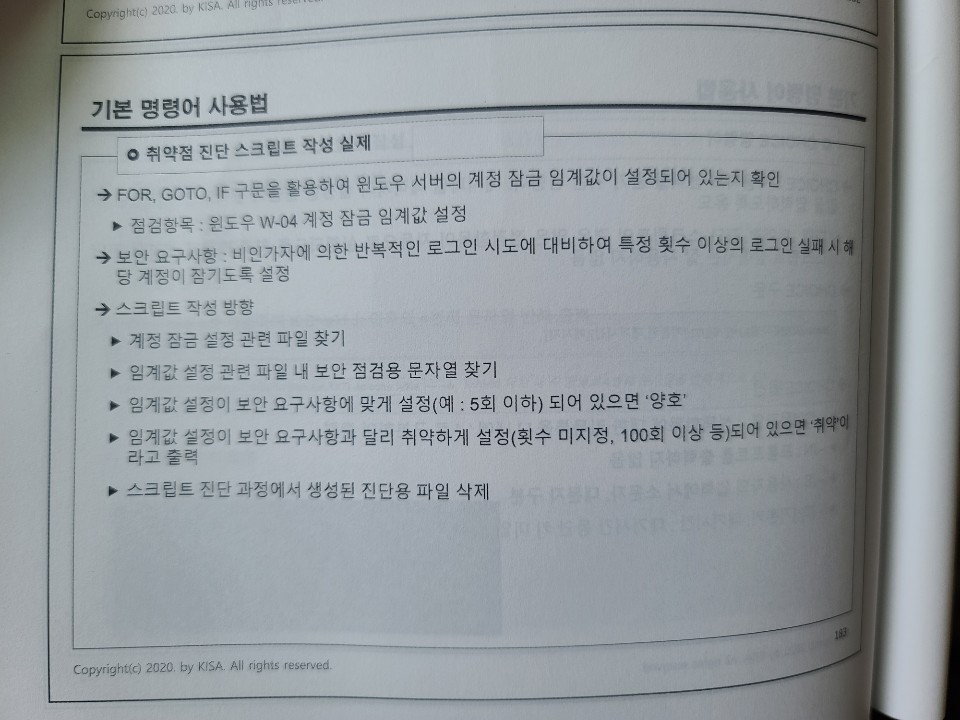












**결과분석 보고서 작성 방법**

- 작성 단계

취약점 진단 결과 파일 확인 -> 취약 여부 확인 -> 진단 보고서 작성

**네트워크 장비 진단**

**라우터 진입 모드**

- User Mode

-> 프롬프트에 ‘>’ 표시가 있는 경우

-> ping 등의 테스트 목적으로 사용되며, 현재 상태 확인만 가능

- Privileged mode

-> 프롬프트에 ‘#’ 표시가 있는 경우

-> User mode에서 enable 명령어를 통해 전환 가능

-> 관리자 모드로서 설정파일 정보를 확인할 수 있음

-> 설정파일을 수정할 경우에는 config terminal 명령어를 입력해 Configuration mode로 진입

**패스워드 설정**

- 점검내용 : 관리 터미널(콘솔, SSH, https 등) 을 통해 네트워크 장비 접근 시 기본 패스워드(기본 관리자 계정도 함께 변경하도록 권고)를 사용하는지 점검

- 점검목적 : 기본 패스워드를 변경 후 사용하는지 점검하여 기본 패스워드를 변경하지 않고 사용함으로써 발생할 수 있는 비인가자의 네트워크 장비 접근에 대한 통제가 이루어지는지 확인하기 위함

**판단 기준**

­- 양호 : 기본 패스워드를 변경하여 사용하는 경우

- 취약 : 기본 패스워드를 변경(패스워드 미설정 사용 포함)하지 않고 사용하는 경우

**점검 파일 위치 및 점검 방법**

Router# show running-config

Enable 패스워드 설정 확인

Line Access에서 각 라인별(VTY, CON, AUX) 패스워드 설정 확인

**패스워드 복잡성 설정**

- 점검내용 : 네트워크 장비에 기관 정책에 맞는 계정 패스워드 복잡성 정책이 적용되어 있는지 점검. 패스워드 복잡성 정책 설정 기능이 장비에 존재하지 않을 경우 기관 정책에 맞게 계정 패스워드를 설정하여 사용하는지 점검

- 점검목적 : 패스워드 복잡성 정책이 장비 정책에 적용되어 있는지 점검하여 비인가자의 네트워크 장비 터미널(콘솔, SSH, https 등) 접근 시도 공격(무작위 대입 공격, 사전 대입 공격 등)에 대한 대비 여부를 확인하기 위함

**판단기준**

- 양호 : 기관 정책에 맞는 패스워드 복잡성 정책이 네트워크 장비에 적용되어 있거나, 패스워드 복잡성 기능이 장비에 존재하지 않을 경우 기관 정책에 맞게 패스워드를 설정하여 사용하는 경우

- 취약 : 기관 정책에 맞지 않는 패스워드를 설정하여 사용하는 경우

**점검 파일 위치 및 점검 방법**

1. 장비에 패스워드 복잡성 설정이 적용되어 있는지 점검

2. 패스워드 복잡성 설정이 존재하지 않을 경우 기관 내 정책에 따라 패스워드를 설정하여 사용하는지 확인

Router#show running-config

영문 소문자, 특수문자, 숫자로 이루어진 8자 이상의 패스워드 사용 중이므로 **양호**

**암호화된 패스워드 사용**

- 점검내용 : 계정 패스워드 암호화 설정이 적용되어 있는지 점검

- 점검목적 : 계정 패스워드 암호화 설정 유무를 점검하여 비인가자의 네트워크 장비 터미널 접근으로 인해 발생할 수 있는 장비 내 계정 패스워드 유출에 대비가 되어 있는지 확인하기 위함

**판단기준**

- 양호 : 패스워드 암호화 설정이 적용된 경우

- 취약 : 패스워드 암호화 설정이 적용되어 있지 않은 경우

**점검 파일 위치 및 점검 방법**

­Router#show running-config

1. Enable Secret 사용 확인

2. Password-Encryption서비스 동작 확인

**VTY접근(ACL) 설정**

- 점검내용 : 원격 터미널(VTY)를 통해 네트워크 장비 접근 시 지정된 IP에서만 접근이 가능하도록 설정되어 있는지 점검

- 점검목적 : 지정된 IP만 네트워크 장비에 접근하도록 설정되어 있는지 점검하여 비인가자의 터미널 접근을 원천적으로 차단하는지 확인하기 위함

**판단기준**

­- 양호 : 원격 터미널(VTY) 접근 시 지정된 IP만 접근 하도록 설정이 되어있는 경우

- 취약 : 원격 터미널(VTY) 접근 시 지정된 IP만 접근 하도록 설정이 되어있지 않은 경우

**점검 파일 위치 및 점검 방법**

Router# show running-config

1. Access-List 설정 확인

2. 각 Line VTY에 access-class 설정 확인

**Session Timeout 설정**

- 점검내용 : 기관 정책에 맞게 Session Timeout 설정이 적용되어 있는지 점검

- 점검목적 : Session Timeout 설정 유무를 점검하여 터미널 접속 후 일정 시간(Session Timeout 지정 시간)이 지난 뒤 터미널 세션이 자동으로 종료되어 관리자의 부재(터미널 작업 중 자리 비움, 작업 완료 후 터미널 접속을 종료하지 않음) 시 발생 가능한 비인가자의 터미널 접근 통제가 되는지 확인하기 바람

**판단기준**

- 양호 : Session Timeout 시간을 기관 정책에 맞게 설정한 경우

- 취약 : Session Timeout 시간을 기관 정책에 맞게 설정하지 않는 경우

**점검 파일 위치 및 점검 방법**

Router#show rinning-config

각 Line Acess의 exec-timeout 설정 확인

Ex) exec-timeout 0 0

Exec-timeout 0 0 Console, VTY 모두 Timeout 설정 되어 있지 않으므로 **취약**

**패치 관리 > 최신 보안 패치 및 벤더 권고사항 적용**

- 점검내용 : 패치 적용 정책에 따라 주기적인 패치를 하고 있는지 점검

- 점검목적 : 네트워크 장비의 보안 수준을 높이고 성능 및 기능 향상을 위해서 버전 업그레이드 및 보안 패치 작업을 수행해야 함

**판단기준**

- 양호 : 패치 적용 정책을 수립하여 주기적으로 패치를 관리하고 있을 경우

- 취약 : 패치 적용 정책을 수립하여 주기적으로 패치를 관리하고 있지 않을 경우

**점검 파일 위치 및 점검 방법**

Router#show version

버전정보 확인

Version 12.2(20)

시스템 타이머에 대한 원격 공격 취약점이 발견된 버전을 사용 중이므로 **취약**

**SNMP 서비스 확인**

- 점검내용 : 네트워크 장비에 사용하지 않는 SNMP 서비스가 구동되고 있거나 SNMP 서비스 사용 시 암호화가 지원되는 버전을 사용하고 있는지 점검

- 점검목적 : 불필요한 SNMP 서비스 차단 및 보안에 취약한 SNMP 서비스의 취약점(조작된 MIB 정보를 통한 네트워크 설정 변경, 전송 데이터 평문 전송 등)을 이용한 공격을 차단하기 위함

**판단기준**

- 양호 : SNMP v3 이상 버전을 사용하거나 서비스를 사용하지 않는 경우

- 취약 : SNMP v3 이상 버전을 사용하거나 불필요하게 서비스를 활성화한 경우

**점검 파일 위치 및 점검 방법**

1. SNMP 설정 확인

Router#show running-config

EX) snmp-server community public RW => Community String을 통해 인증하는 SNMP 서비스 동작 확인 -> 불필요한 서비스이므로 **취약**

2. SNMP 서비스 동작 확인

Router#show snmp

/// SNMP 서비스 비활성화 시 아래 문구 출력

%SNMP agent not enabled

**SNMP community string 복잡성 설정**

- 점검내용 : SNMP 서비스 사용 시 Community String을 기본 설정(public, private)으로 사용하고 있는지 점검

- 점검목적 : SNMP Community String을 공격자가 쉽게 유추하지 못하도록 설정하여 Community String 탈취에 대한 위험을 줄이기 위함

**판단기준**

- 양호 : SNMP 서비스를 사용하지 않거나, 유추하기 어려운(영문자, 숫자 포함 10자) Community String을 설정한 경우

- 취약 : 디폴트 Community String을 변경하지 않거나, 유추하기 쉬운 Community String으로 설정한 경우

**점검 파일 위치 및 점검 방법**

SNMP 설정 확인

Snmp-server community public RW

Public 이부분

디폴트 Community String을 변경하지 않았으므로 **취약**

**SNMP 커뮤니티 권한 설정**

- 점검내용 : SNMP 서비스 사용 시 Community String 권한이 불필요하게 RW로 설정되어 있는지를 점검

- 점검목적 : 불필요한 SNMP Community String의 RW 권한을 제거함으로서 공격자의 SNMP를 통한 라우터 정보 수정을 막기 위함

**판단기준**

- 양호 : SNMP Community String 권한이 RO인 경우

- 취약 : SNMP Community String 권한이 불필요하게 RW로 설정된 경우

**점검 파일 위치 및 점검 방법**

Router#show running-config

SNMP 설정 확인

snmp-server community public RW -> 불필요한 권한 설정(RW)되어있어 **취약**

**사용하지 않는 인터페이스의 shutdown 설정**

- 점검내용 : 사용하지 않는 불필요한 인터페이스가 shutdown되지 않고 활성화되고 있는지를 점검

- 점검목적 : 필요한 인터페이스만 활성화하여 비인가자가 사용하지 않는 인터페이스를 통하여 네트워크에 접근하는 것을 차단하기 위함

**판단기준**

- 양호 : 사용하지 않는 인터페이스가 차단된 경우

- 취약 : 사용하지 않는 인터페이스가 차단되지 않은 경우

**점검 파일 위치 및 점검 방법**

1. 각 인터페이스 설정 확인

Rounter#show running-config

Interface FastEthernet0/0 -> 담당자 인터뷰 확인 : 사용 중인 인터페이스

2. 인터페이스 상태 확인 (해당 포트가 down/up이 되어 있는지 확인할 수 있음)

Router#show ip interface brief

FastEthernet0/0 10.10.10.1 YES manual up -> up부분…. 인터페이스상태 : 활성화 -> 사용 중인 인터페이스만 활성화 되어 있으므로 양호