
리눅스 기초 강의자료

(3 ~ 4 일차)

목차 - 3 ~ 4일차

3일차 :

1. 명령어 학습
2. Ubuntu 호스트이름 변경
3. 계정 생성/삭제와 권한 및 파일 접근 권한
4. SFTP 를 통한 파일 업로드/다운로드
5. Linux에서 C파일 컴파일하고 실행해보기

4일차 :

1. Shell Script
2. Web 서버(HTTP/HTTPS)와 DNS 서버 구축
3. FTP 서버 구축
4. 프로세스 관리

3일차

명령어 학습

리다이렉트(>) 명령어

- `echo "Hello World2" > testfile`
- `echo "Hello World3" >> testfile`
- `>` 만 하면 덮어쓰움
- `>>` 하면 행 추가됨

파일읽는 명령어

- `cat` : 파일의 내용을 전부 출력
 - `cat /var/log/syslog`
- `tail` : 최근의 내용을 출력 (10행)
 - `tail /var/log/syslog`
 - `tail -F /var/log/syslog` (실시간으로 최근내용 확인할 때 사용)
- `more` : 파일 내용을 일부만 출력, 스페이스바 누르면서 내용으로 넘어감
 - `more /var/log/syslog`
- `grep`
 - 특정 문자의 관련 행만 출력
 - `ps -ef | grep "volume"`
 - `cat /var/log/syslog | grep "xinet"`

명령어 학습

편집명령어 (vi)

- `%s/[현재이름]/[바꿀이름]`
- `:`에서 숫자 입력 하면 => 관련 번호 행으로 이동
- `/`에서 이름 입력하면 => 관련 이름있는 행으로 이동
 - `n` 버튼 누르면 해당이름의 다음 행으로 이동

시스템 명령어

- `free` 또는 `free -m` : 메모리 보여줌
- `top -d 1` : CPU 1초 주기로 갱신하면서 보여줌
 - 1번 클릭시, 멀티코어를 보여줌
 - 실시간으로 높은 CPU 사용률은 상단에 프로세스가 확인된다.
- `vmstat 1` : 1초마다 cpu 및 메모리 등의 정보 확인
 - `-t` 옵션 : timestamp 추가하여 cpu 및 메모리 등 확인 (`vmstat -t`)

특정 파일 찾는 명령어

- `find`
 - `find / -name "**관련 이름**"`
 - `/` (슬래시)는 경로

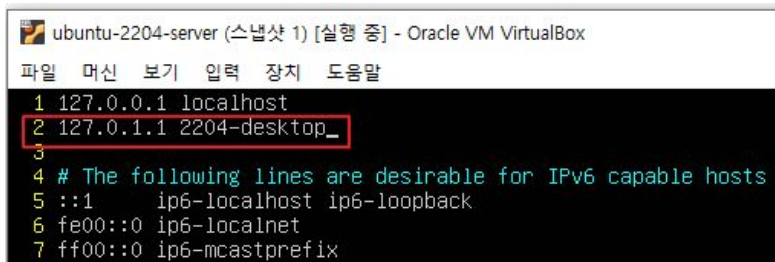
Ubuntu 호스트이름 변경

hostnamectl :

시스템의 호스트네임을 관리할 수 있는 명령어로 옵션없이 사용하거나 `status` 옵션을 주면 현재 정보를 출력

변경 방법 :

- 1) `hostnamectl set-hostname [host 이름]` (ex. 1-20-04-desktop)
- 2) `vi /etc/hosts`
 - a) 2번 행에서 이름 변경



```
ubuntu-2204-server (스냅샷 1) [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말
1 127.0.0.1 localhost
2 127.0.1.1 2204-desktop_
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1      ip6-localhost ip6-loopback
6 fe00::0  ip6-localnet
7 ff00::0  ip6-mcastprefix
```

- 3) ubuntu 재로그인
- 4) 호스트 이름 변경 확인 (로그인 화면 또는 `hostnamectl / hostnamectl status` 명령어로 확인)

계정 생성/삭제와 권한 및 파일 접근 권한

계정생성 방법 1 :

useradd (간략한 user 생성)

- 기본적인 쉘만 지원하고, home 디렉토리를 생성하지 않음
- passwd 명령어로 계정 패스워드 설정해야함 - passwd [계정명]
- 세부설정
 - a. useradd test3
 - b. passwd test
 - c. mkdir /home/test3
 - d. chown -R [계정명]:[그룹명] [경로명]
 - i. chown -R test3:test3 /home/test3
 - e. groupadd test3 (그룹없는 경우 명령어 실행)
 - i. group 확인 방법 : cat /etc/group
 - f. usermod -G [그룹이름] [사용자이름]
 - i. usermod -G test3 test3
 - g. groups test3
 - h. id test3
 - i. usermod -s [셸파일경로] [계정이름]
 - i. usermod -s /bin/bash test3
 - j. useradd [계정이름] -m -s [셸파일경로]
 - i. useradd test3 -m -s /bin/bash (이미 추가되어 있으면 안해도됨)

계정 생성/삭제와 권한 및 파일 접근 권한

계정생성 방법 2 :

adduser (계정 생성에 대한 가이드를 해줌)

- **adduser** [계정명]
- 대부분의 우분투 사용자들이 자주 사용
- 계정 권한, **UID**, **GID** 등 자동 할당
- **home** 디렉토리가 자동 생성

생성계정 로그인 방법 :

- **login** [계정명]

계정삭제 방법 :

- **deluser** [계정명]
 - **deluser test3**
 - **--remove-home** : 홈 디렉토리 삭제
 - **deluser --remove-home test3**
 - **--remove-all-files** : 홈 디렉토리, 계정명으로된 모든 파일 삭제
 - **deluser --remove-all-files testdd**

생성 계정의 sudo 권한 설정하기

생성한 계정의 **sudo** 설정 방법 1 :

- vi/etc/sudoers

```
19 # User privilege specification
20 root    ALL=(ALL:ALL) ALL
21 test2   ALL=(ALL:ALL) ALL
22 test3   ALL=(ALL:ALL) ALL
```

생성한 계정의 **sudo** 설정 방법 2 :

1. usermod -G sudo [계정명]
2. groups [계정명] 으로 관련 그룹 추가 확인

계정 생성/삭제와 권한 및 파일 접근 권한

파일 권한

- `ls -al` 로 확인 가능
- `rwX(1) r-x(2) r-x(3) => 755`
- `rwX` :
 - `r` : 읽기 (read 권한) - 4
 - `w` : 쓰기 (write 권한) - 2
 - `x` : 실행 (execute 권한) - 1

일반적으로 권한을 주는 방법 :

`chmod` [권한값] [파일명]

ex) `chmod 777 file1.txt`

리눅스와 유닉스 기반 시스템에서 파일이나 디렉토리의 권한을 변경 (Change Mode)

mathematica Copy code

Permissions	Links	Owner	Group	Size	Last Modified	Name
-rw-r--r--	1	user1	group1	1024 bytes	Feb 20 12:30	file1.txt
drwxr-xr-x	2	user2	group2	4096 bytes	Feb 19 08:45	directory1

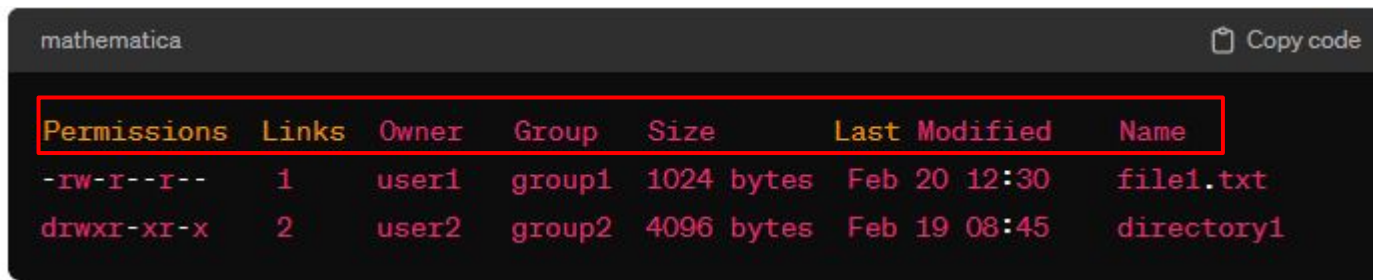
(0) (1) (2) (3)

- (0) : 'd' = 디렉토리, '-' = 파일
- (1) : 소유자 (계정명)
- (2) : 그룹
- (3) : 해당 소유자와 그룹이 아닌 다른 계정 및 그룹 (other)

계정 생성/삭제와 권한 및 파일 접근 권한

ls -al 시 칼럼의 내용들

- Permissions: 파일 또는 디렉토리의 허가권
- Links: 파일 또는 디렉토리에 대한 하드 링크의 수
- Owner: 파일 또는 디렉토리의 소유자
- Group: 파일 또는 디렉토리가 속한 그룹
- Size: 파일의 크기를 바이트 단위로 표시
- Last Modified: 파일이 마지막으로 수정된 날짜와 시간
- Name: 파일 또는 디렉토리의 이름을 표시



Permissions	Links	Owner	Group	Size	Last Modified	Name
-rw-r--r--	1	user1	group1	1024 bytes	Feb 20 12:30	file1.txt
drwxr-xr-x	2	user2	group2	4096 bytes	Feb 19 08:45	directory1

참고 : 하드링크와 심볼릭 링크 1

하드 링크 (Hard Link):

- 동일한 파일에 대한 다른 이름을 제공 (파일만 가능)
- 파일 시스템의 i-node(메타데이터를 저장하는 데이터 구조) 번호가 동일하여 서로 독립적인 파일 엔트리로 취급됨
- 원본 파일이나 다른 하드 링크 중 하나를 통해 파일에 접근 가능
- 파일을 삭제할 때, 모든 하드 링크가 삭제되어야만 실제 데이터가 삭제됨
- 동일한 파일 시스템 내에서만 생성 가능

확인 방법 :

- touch abc1
- ln abc1 abc2
- ls -al -i abc*

```
root@test:~# ls -al -i abc*
461589 -rw-r--r-- 2 root root 0 Feb 21 20:32 abc1
461589 -rw-r--r-- 2 root root 0 Feb 21 20:32 abc2
461590 lrwxrwxrwx 1 root root 3 Feb 21 20:33 abc3 -> abc
root@test:~#
root@test:~#
```

참고 : 하드링크와 심볼릭 링크 2

심볼릭 링크 (Symbolic Link 또는 Soft Link):

- 심볼릭 링크는 파일에 대한 바로 가기 생성
- 하드링크와 달리 파일만 아닌, 디렉토리도 생성가능
- 원본 파일을 가리키는 새로운 파일 엔트리를 생성하며, 이 파일 엔트리는 원본 파일의 경로를 가지게 됨
- 원본 파일이나 심볼릭 링크 중 하나를 통해 파일에 접근 가능
- 원본 파일이 삭제되면 심볼릭 링크는 더 이상 유효하지 않게 됨 ('끊긴 링크')
- 심볼릭 링크는 다른 파일 시스템이나 네트워크 드라이브 등에 대해서도 생성 가능

확인 방법 :

- touch abc1
- ln -s abc1 abc3
- ls -al -i abc*

```
root@test:~# ls -al -i abc*
461589 -rw-r--r-- 2 root root 0 Feb 21 20:32 abc1
461589 -rw-r--r-- 2 root root 0 Feb 21 20:32 abc2
461590 lrwxrwxrwx 1 root root 3 Feb 21 20:33 abc3 -> abc
root@test:~#
root@test:~#
```

SFTP를 통한 파일 업로드 / 다운로드

FileZilla 설치하기 (Windows)

- <https://filezilla-project.org/>

Home

FileZilla

Features

Screenshots

Download

Documentation

FileZilla Pro

FileZilla Server

Download

Community

Forum

Wiki

General

FAQ

Support

Contact

License

Privacy Policy

Trademark Policy

Development

Source code

Nightly builds

Overview

Welcome to the homepage of FileZilla®, the free FTP solution. The *FileZilla Client* not only supports FTP, but also FTP over TLS (FTPS) and SFTP. It is open source software distributed free of charge under the terms of the GNU General Public License.

We are also offering *FileZilla Pro*, with additional protocol support for WebDAV, Amazon S3, Backblaze B2, Dropbox, Microsoft OneDrive, Google Drive, Microsoft Azure Blob and File Storage, and Google Cloud Storage.

Last but not least, *FileZilla Server* is a free open source FTP and FTPS Server.

Support is available through our [forums](#), the [wiki](#) and the [bug and feature request trackers](#).

In addition, you will find documentation on how to compile FileZilla and nightly builds for multiple platforms in the development section.

Quick download links

Download FileZilla Client

All platforms

Download FileZilla Server

All platforms

Pick the client if you want to transfer files. Get the server if you want to make files available for others.



FileZilla

The free FTP solution

Home

FileZilla

Features

Screenshots

Download

Documentation

FileZilla Pro

FileZilla Server

Download

Community

Forum

Wiki

General

FAQ

Support

Contact

License

Privacy Policy

Trademark Policy

Development

Source code

Nightly builds

Translations

Version history

Changelog

Issue tracker

Other projects

libFileZilla

Octochess

Sponsors

NCC

Promotion

FileZilla® Pro

GET IN NOW

Download FileZilla Client

The latest stable version of FileZilla Client is 3.46.0

Please select the file appropriate for your platform

Windows (64bit x86)

Download FileZilla Client

This installer may include bundled offers. Click here to learn more.

More download options

Other platforms:

Not what you are looking for?

Show additional download options

Please select your edition of FileZilla Client

	FileZilla	FileZilla with manual	FileZilla Pro	FileZilla Pro + CLI
Standard FTP	Yes	Yes	Yes	Yes
FTP over TLS	Yes	Yes	Yes	Yes
SFTP	Yes	Yes	Yes	Yes
Comprehensive PDF manual	-	Yes	Yes	Yes
Amazon S3	-	-	Yes	Yes
Backblaze B2	-	-	Yes	Yes
Dropbox	-	-	Yes	Yes
Microsoft OneDrive	-	-	Yes	Yes
Google Drive	-	-	Yes	Yes
Google Cloud Storage	-	-	Yes	Yes
Microsoft Azure Blob + File Storage	-	-	Yes	Yes
WebDAV	-	-	Yes	Yes
OpenStack Swift	-	-	Yes	Yes
Box	-	-	Yes	Yes
Site Manager synchronization	-	-	Yes	Yes
Command-line interface	-	-	-	Yes
Batch transfers	-	-	-	Yes

Download

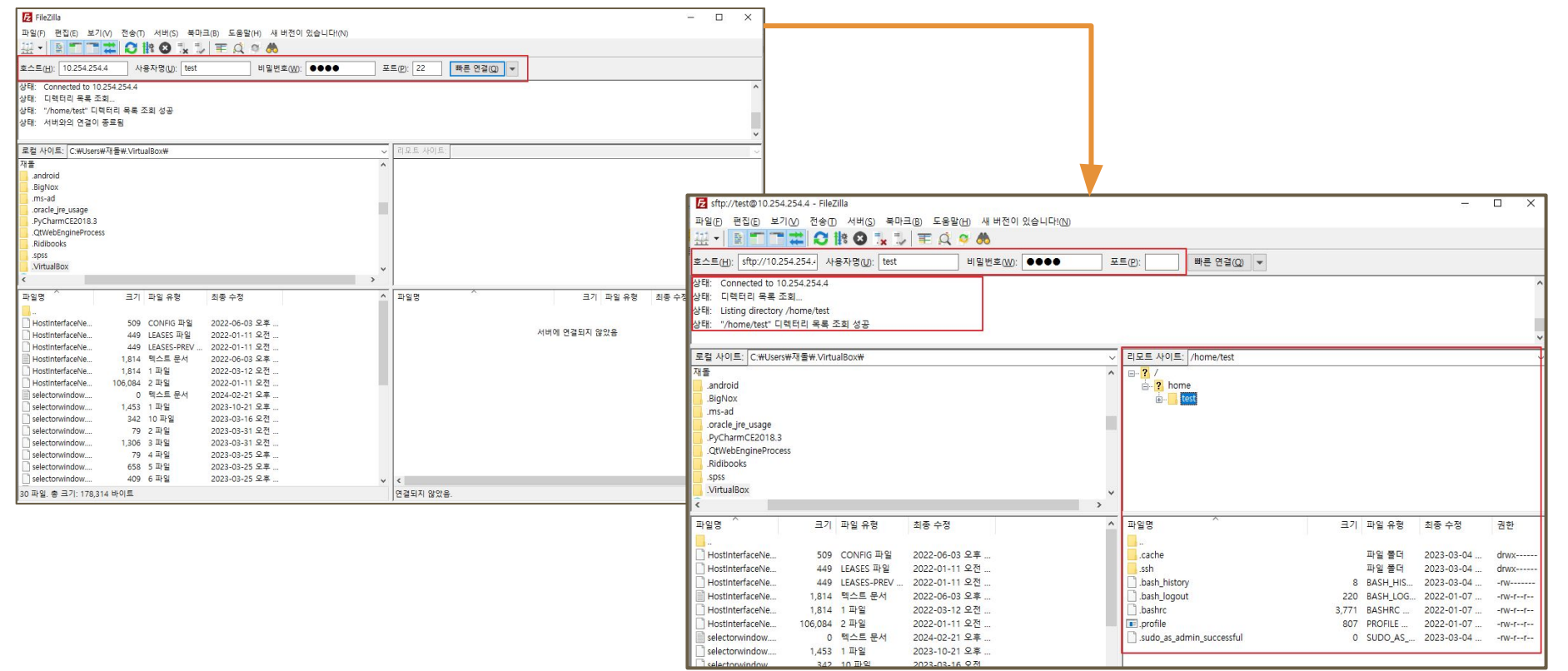
Select

Select

Select

SFTP를 통한 파일 업로드 / 다운로드

FileZilla 실행 후 Ubuntu로 SFTP 접속 시도



Linux에서 C파일 컴파일하고 실행해보기

gcc (GNU Compiler Collection)

- GNU 프로젝트의 일환으로 개발되어 여러 언어를 위한 컴파일러 및 관련 도구의 모음

사용 방법의 예 :

- `gcc -Wall [소스코드 파일] -o [실행 파일명]`
- `Wall` : 모든 모호한 코딩에 대해서 경고를 보내는 옵션
 - `-W` : 컴파일러가 합법적이지만 모호한 코딩에 대해서 알려줌 (Warning)
 - `-all` : 컴파일러가 아주 사소한 모호성에 대해서도 알려줌

(가능한 모든 경고를 알려줌, 프로그램의 잠재적인 문제를 식별하기 위해 가장 포괄적인 경고 수준을 설정)

vi 편집기로 fork.c
파일을 작성



```
#include <unistd.h>
#include <stdio.h>

int main(int argc, char **argv)
{
    pid_t pid;
    printf("Calling fork()\n");
    pid=fork();

    if(pid>0) // 부모
        printf("Parent PIDs : %d, %d, %d\n",pid,getpid(),getppid());

    else if(pid==0) // 자식
        printf("Child PIDs : %d, %d, %d\n",pid,getpid(),getppid());

    else
        printf("fork() error.\n");

    return 0;
}
```



```
root@test1:/home/test1# gcc -Wall fork.c -o fork
root@test1:/home/test1# ./fork
```

```
Calling fork()
Parent PIDs : 8187, 8186, 2657
Child PIDs : 0, 8187, 8186
```


4일차

Shell Script

Shell Script 란?

주로 Unix 및 Linux 계열 운영 체제에서 사용되는 스크립트 언어 (파일 확장자 : **.sh**)

- 셸 환경에서 실행되는 명령어들을 스크립트(대본) 형태로 작성한 것
- 일련의 명령어들을 순차적으로 실행하거나 특정 조건에 따라 분기하여 실행하는 등의 작업을 수행 가능
- **Shell Script**는 일반적으로 셸 (예: **Bash, Zsh, Ksh**)을 사용하여 작성됨
- 시스템 관리, 자동화, 배치 처리 등 다양한 용도로 사용
 - 일상적인 작업을 자동화하고 반복적인 과정을 효율적으로 처리하는 데 유용

Shell Script를 작성하려면?

- 기본적인 프로그래밍 개념과 셸의 명령어 구문에 대한 이해가 필요.
- 일반적으로 제어 구조 (조건문, 반복문), 변수, 함수 등의 개념

Shell Script의 예시

파일 및 디렉토리 관리: 파일 복사, 이동, 삭제, 디렉토리 생성 등

- 시스템 관리: 사용자 계정 관리, 패키지 설치 및 업데이트, 서비스 관리 등
- 자동화 작업: 백업 스크립트, 로그 파일 모니터링 및 정리, 스케줄된 작업 실행 등

Shell Script

3일차에 학습한 계정 생성 내용을 아래 스크립트로 실행해보기

- 1) `cd /root`
- 2) `vi useradd.sh`
- 3) 아래 스크립트 작성 후 저장
- 4) `chmod 777 useradd.sh`
- 5) `./useradd.sh` (또는 `sh /root/useradd.sh`)

```
#!/bin/bash

for i in $(seq 10 15);
do

    useradd test$i
    echo "test$i:.com" | chpasswd
    mkdir /home/test$i
    chown -R test$i:test$i /home/test$i
    usermod -G test$i test$i
    groups test$i
    id test$i
    usermod -s /bin/bash test$i
    useradd test$i -m -s /bin/bash

done
```

Shell Script

by 22년 사이버보안 지방대회 3과제 문제

- 1. 리눅스 시스템의 보안 상태를 지속적으로 점검할 수 있도록 `bash shell script`를 작성.
- 2. 관리자가 `root` 계정으로 `/root/check.sh` 스크립트를 실행할 경우, 아래와 같은 3가지 항목을 검사하여 양호/취약 여부를 출력할 수 있도록 작성.
 - 1). 패스워드 파일 권한이 적절하게 부여되어 있는지
=> other 권한에 쓰기 또는 실행 권한이 부여되어 있는 경우 취약 것으로 판단
 - 2). 해당 시스템에 라우팅 기능이 활성화되어 있는지?
 - 3). SetUID, SetGID, Sticky bit가 설정된 파일들의 목록

□ 출력 결과

```
*****
                        시스템 취약점 점검을 실시합니다.
*****
시작 시간 : Tue Apr 27 10:23:54 KST 2021

01. /etc/passwd 파일 권한 점검
    ==> [취약] 현재 권한 : -rw-r--rwx

안전한 경우
    ==> [양호] 현재 권한 : -rw-r-r--

02. 라우팅 기능 활성화 여부 점검
    ==> [취약] 라우팅 기능이 활성화 되어 있습니다.

안전한 경우
    ==> [양호] 라우팅 기능이 비활성화 되어 있습니다.

03. SetUID, SetGID, Sticky bit 권한이 설정된 파일을 검색하여 /root/file_list.txt에 저장합니다.
```

Shell Script

by 22년 사이버보안 지방대회 3과제 문제

/root/ 디렉토리에 [check.sh](#) 파일 생성

1) vi /root/check.sh

2) 옆에 스크립트 작성 후 저장

3) chmod 777 check.sh

4) ./check.sh

(또는 sh /root/check.sh)

```
#!/bin/sh

echo "
*****

시스템 취약점 점검을 실시합니다.

*****
"
d=`date`
# P= Privilege, F = Full, R= Routing
PVAL=`ls -al /etc/passwd | cut -c 8-10` # PVAL은 other 권한만 변수 넣기
FVAL=`ls -al /etc/passwd | cut -c 1-10` # FVAL은 소유자, 그룹, other 권한 변수 넣기
RVAL=`cat /proc/sys/net/ipv4/ip_forward`

echo "시작 시간 : $d"
echo

# 권한 확인
echo "01. /etc/passwd 파일 권한 점검"
# other 권한만 조건으로 검색한다.
if [ $PVAL = "r--" ]
# 권한을 전부 출력한다
then
    echo " ==> [양호] 현재 권한 : $FVAL"

else
    echo " ==> [취약] 현재 권한 : $FVAL"

fi

echo

# 라우팅 확인#
echo "02. 라우팅 기능 활성화 여부 점검"

if [ $RVAL -eq 0 ]
then
    echo " ==> [양호] 라우팅 기능이 비활성화 되어 있습니다."

else
    echo " ==> [취약] 라우팅 기능이 활성화 되어 있습니다."
fi

echo

# 특수권한 파일 찾기
echo "03. SetUID, SetGID, Sticky bit 권한이 설정된 파일을 검색하여 /root/files.txt에 저장합니다"
rm /root/files.txt
find / -perm -4000 >> /root/files.txt 2> /dev/null
find / -perm -2000 >> /root/files.txt 2> /dev/null
find / -perm -1000 >> /root/files.txt 2> /dev/null
```

DNS 서버 구축하기 (BIND9)

Bind9 (Berkeley Internet Name Domain) 이란?

- DNS(Domain Name System) 서버 소프트웨어 패키지
- 인터넷에서 호스트 이름을 IP 주소로 해석하고 IP 주소를 호스트 이름으로 변환하는 DNS 서비스를 제공
- DNS 서버는 네트워크에 연결된 장치들 간에 도메인 이름을 해석하는데 사용

KISA에서 소개하는 도메인이란?

<https://xn--3e0bx5euxnjj69i70af08bea817g.xn--3e0b707e/jsp/resources/domainInfo/domainInfo.jsp>

DNS와 작동 원리

<https://hi-guten-tag.tistory.com/380>



DNS 서버 구축하기 (BIND9)

DNS query (쿼리) 비교 (Client -> Server, 로컬 DNS 기준)

- <https://hi-guten-tag.tistory.com/380>

특징	Authoritative query	Iterative(반복적인) query	Recursive(재귀적인) query
역할 & 동작 방식	클라이언트 또는 리쿠르시브 DNS가 도메인의 데이터를 가진 (권한을 가진) DNS 서버로 바로 질의 (해당 서버의 Zone 파일에 해당 레코드 값이 있는 경우)	로컬 DNS서버는 다른 DNS서버에게 각각 질의하며 최종적으로는 클라이언트에게 응답	쿼리를 받은 DNS 서버는 관련 도메인 데이터가 없다면 다른 DNS 서버로 질의 즉, 자신을 참조하는 것으로 다른 DNS 서버에게 쿼리를 전달하고, 쿼리를 받은 DNS 서버는 동일한 사유면 다른 DNS 서버로 쿼리를 전달 가능 이러한 과정을 통해 Authoritative한 DNS 서버로부터 응답을 받은 DNS 서버는 클라이언트에게 최종적으로 응답을 제공
요청 프로세스	클라이언트(PC 또는 DNS 서버)가 특정 도메인에 대한 정보를 가진 DNS 서버에게 질의 => 해당 DNS 서버가 도메인의 데이터에 대한 권한을 가지고 있으면 클라이언트에게 바로 응답	클라이언트가 요청한 정보를 찾을 때까지 관련 DNS 서버들이 서로를 여러번 순회해서 질의함	클라이언트가 요청한 정보를 찾을 때까지 쿼리를 받은 DNS 서버가 다른 DNS로 다시 쿼리를 진행
목적	도메인에 대한 정확한 정보를 제공하는 것????	클라이언트가 요청한 정보를 찾기 위한 질의의 반복????	클라이언트의 요청에 따라 완전한 응답을 제공????

Web 서버 구축하기 (apache2) – DNS+Web 동시 확인

apache2 (Apache HTTP Server) 란?

- apache2는 HTTP(HyperText Transfer Protocol) 웹 서버 소프트웨어 패키지
- 세계에서 가장 널리 사용되는 웹 서버 중 하나로, 동적이고 정적인 웹 페이지를 제공
- Apache는 다양한 모듈을 통해 기능을 확장하고, SSL/TLS를 통한 보안 연결을 지원하며, 가상 호스트 설정 등을 통해 다중 웹 사이트 호스팅을 가능하게 함



FTP 서버 구축하기 (vsftp)

vsftpd (Very Secure FTP Daemon) 란?

- vsftpd는 FTP(File Transfer Protocol) 서버 소프트웨어 패키지
- FTP는 파일 전송 프로토콜로, 클라이언트와 서버 간에 파일을 전송하는 데 사용
- vsftpd는 강력한 보안 기능을 갖추고 있어, 보안적으로 신뢰할 수 있는 FTP 서버로 널리 사용. 이는 권한 설정, IP 접근 제어, 암호화 전송 등을 통해 보안성을 유지

