7/1

개인은 데비안 계열의 우분투를 회사에선 레드햇 계열의 rocky 사용

PDF P.22

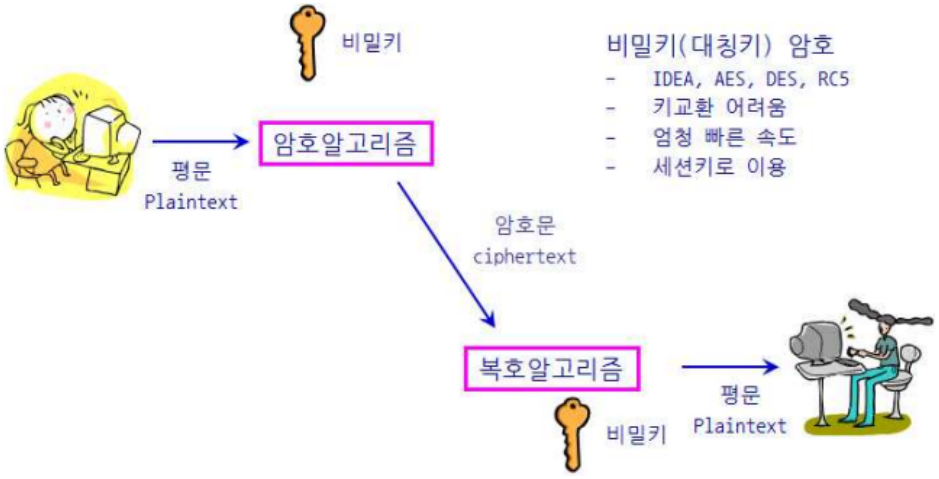
- TELNET

  암호화가 없어서 일반 트래픽이 노출이 된다 따라서 외부로 트래픽이 나가는 경우엔 사용해선 안됨 암호화 통신을 해야 한다.

- SSH

  텔넷의 문제를 해결한 제어 프로토콜 보안 사용

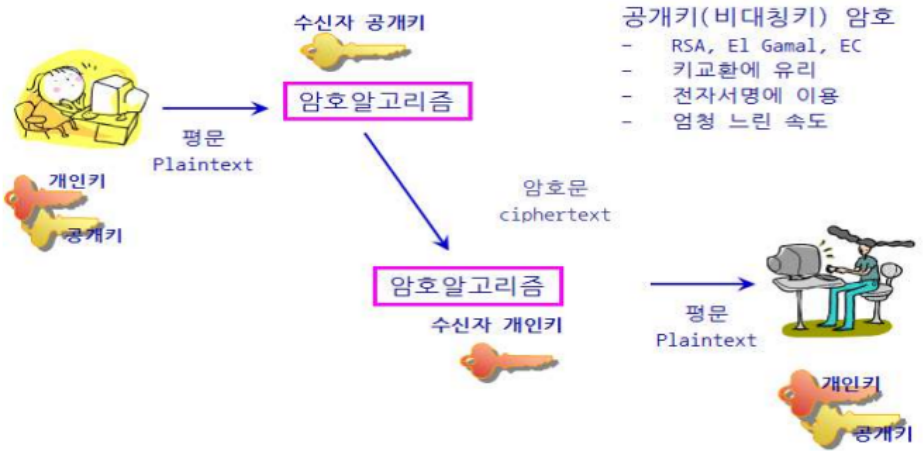
(1) 암호화 방식



< 대칭키 암호화 >

위 그림에서와 같이 대칭키(비밀키) 암호화 방식은 데이터를 암호화하고 복호화하는데 동일한 키를 사용한다. 따라서 송신자와 수신자가 동일한 키를 가지고 있어야지만 데이터를 암호화하고 복호화할 수 있다.

이러한 단점으로 공개키 방식이 등장



< 비대칭키 암호화 >

- 반면 비대칭키 암호화 방식은 서로 다른 두 키를 이용하여 암호화와 복호화에 사용을 하게 된다. 이 두 키를 공개키, 개인키라고 불러서 공개키 암호화 방식이라고도 한다.

- 세션키 교환 : RSA, DSA
- 보안 세션 : IDEA, DES

< SSH를 이용한 원격지 시스템 접속 과정 >

키 공유를 할 필요가 없지만 속도가 엄청 느려서 두 방식을 같이 사용한다.

따라서 비대칭키는 데이터가 엄청 작은 전자서명 인증으로 사용됨.

대칭키를 하기 위해 키를 공유하려고 비대칭키 방식을 사용하고 키가 공유되면 대칭키 방식으로

```
leejeuk@DESKTOP-KM62UF8:~/.ssh$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/leejeuk/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/leejeuk/.ssh/id_rsa
Your public key has been saved in /home/leejeuk/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:+0T5GZEdrdozipMRvIYUVVynKzZPGPrnkhArS78NQY8 leejeuk@DESKTOP-KM62UF8
The key's randomart image is:
+---[RSA 2048]----+
|          ..o..o.|
|         .   .o +.|
|          o. + o. |
|         ..+= +.. |
|        .S.E=Boo  |
|         +o*=.B+  |
|        ..=o=+ooo |
|         .o=o+o   |
|           oo...  |
+----[SHA256]-----+
leejeuk@DESKTOP-KM62UF8:~/.ssh$
```

rsa 방식으로 크기는 2048로 키를 만든다.

> 💡 mobaxterm 세션 생성할 때 id명이 맞지 않으면 안됨 실제 환경과 같게

1. 개인키/공개키 생성

   ssh-keygen -t rsa -b 2048

   cd /home/사용자/.ssh

   ls

   id_rsa id_rsa.pub

   cp id_rsa.pub authorized_keys ⇒ authorized_keys 파일에는 공개키가 있음

2. 윈도우에서 서버에 있는 개인키 가져오기

   scp -i 개인키 사용자아이디@서버ip:/home/사용자/.ssh/id_rsa

```
C:\Users\kdt>scp -i .\ljk-keypair.pem ubuntu@43.203.226.199:/home/ubuntu/.ssh/id_rsa2 .
The authenticity of host '43.203.226.199 (43.203.226.199)' can't be established.
ED25519 key fingerprint is SHA256:U0gMG06pZfXS+7XA5yBSqOa0eTq6voFzJJ78UFbjA2g.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '43.203.226.199' (ED25519) to the list of known hosts.
id_rsa2                                                      100% 1052   181.9KB/s   00:00
```

wsl에서 만든 키로 aws로 공개키를 올리고 windows에서 개인키로 aws 서버 접근하기

1. wsl에서 키 생성
2. aws로 공개키 올리기

   chmod 400 ljk-keypair.pem

   scp -i ljk-keypair.pem id_rsa3.pub ubuntu@43.203.226.199:/home/ubuntu/.ssh

   옮겨진 id_rsa3.pub authorized_keys에 추가해야함

   cat id_rsa3.pub >> authorized_keys
3. 개인키로 aws 서버 접근하기

   ssh -i id_rsa3 ubuntu@43.203.226.199

💡 키는 어디서 만들던지 누가 만들던지 중요하지 않고 공개키와 개인키의 쌍만 맞으면 된다.

💡 ssh 서비스에 개인키를 이용해서 접속하려면 해당되는 공개키가 서버의 authorized_keys 파일에 추가돼 있어야함.

편안하게 공개키 추가 후 접속

#ssh-copy-id -i 개인키 사용자@서버IP ⇒ 개인키의 쌍인 공개키가 서버의 authorized_keys 파일에 추가됨.

그 후에 개인키로 그 서버에 접속을 시도하면 접속이 됨을 확인할 수가 있다.

```
leejeuk@DESKTOP-KM62UF8:~/.ssh$ ssh-copy-id -i id_rsa root@192.168.150.130
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.150.130's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.150.130'"
and check to make sure that only the key(s) you wanted were added.
```

```
leejeuk@DESKTOP-KM62UF8:~/.ssh$ ssh -i id_rsa root@192.168.150.130
Last login: Mon Jul  1 06:10:23 2024 from 192.168.150.1
[root@rocky ~]#
```

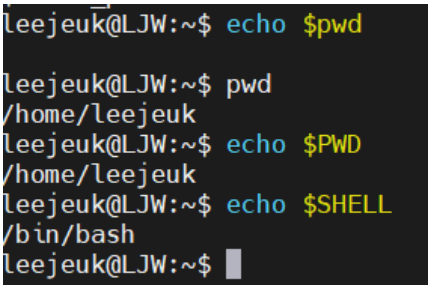
리눅스 개념

- SHELL
  1. bash

     리눅스의 기본 쉘

     root가 # 일반이 $

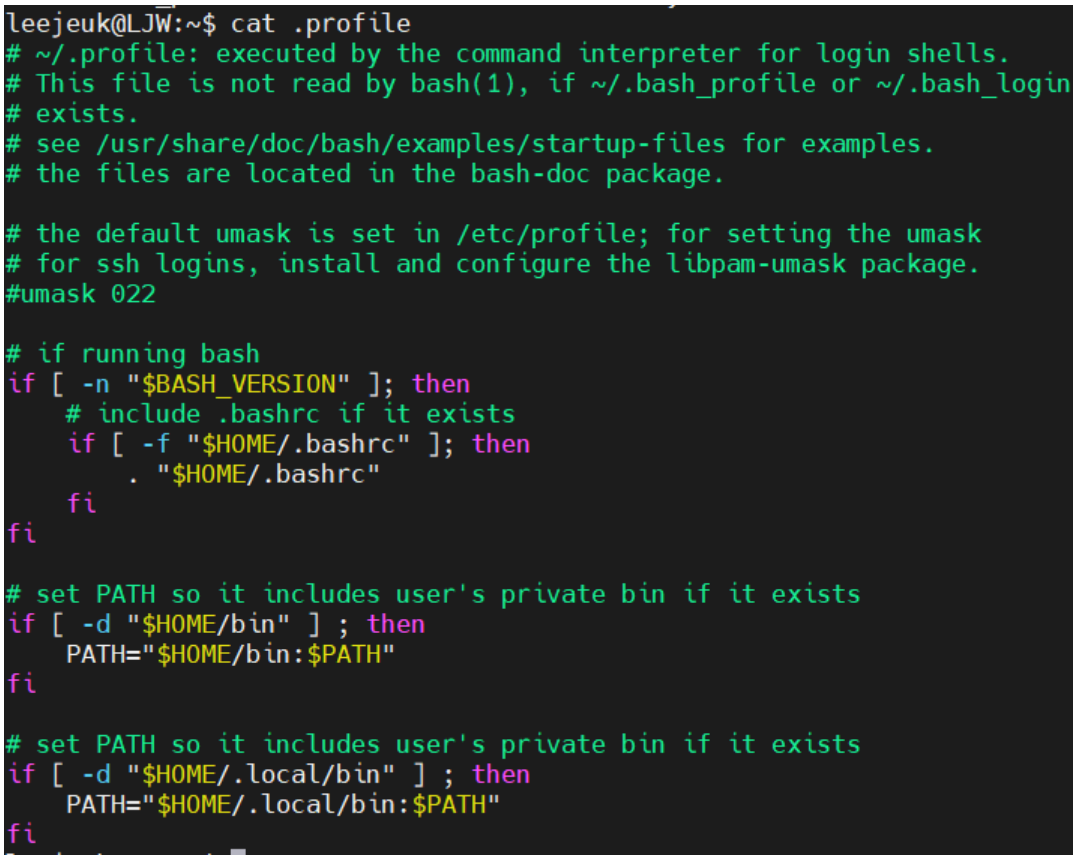
- echo



대문자 값은 환경변수이다. 프로그램에선 전역변수랑 같은 개념

반드시 대문자여야 한다.

- 환경변수



💡 - 옵션은 약자 — 은 풀네임

- 파일 및 디렉토리 관련 명령

1) ls(List) : 디렉토리의 파일 목록 출력

-a : 숨김 파일까지 출력

-l : 파일, 디렉토리의 자세한 정보를 출력

-h: 파일크기를 보기 좋게 출력 human_readble

—help : -h는 help가 아님 이렇게 입력해야 나온다

ls 디렉토리명 : 디렉토리의 파일 목록 출력

2) cd(change directory): 작업 디렉토리 변경

cd 변경할 디렉토리

cd - : 현재 디렉토리로 오기 전 디렉토리로 이동

cd : 홈 디렉토리 /home/사용자명

cd ~ : 홈디렉토리

cd . : 현재 디렉토리로 이동

cd .. : 상위 디렉토리로 이동 cd../..

-절대경로 : '/' 부터 전체 경로를 모두 표시 /var/log/messages

-상대경로 : 현재 위치를 기준으로 표시

#pwd, /home/ubuntu

#ls .ssh

#pwd /home

#ls .ssh ⇒ 오류 발생

ls ubuntu/.ssh

3) cp(copy) 파일 or 디렉토리복사

cp 원본 파일 복사할 디렉토리 or 복사 파일

-r : 하위 디렉토리까지 모두 복사

cp -r /etc/* /tmp : etc 폴더 아래에 있는 모든 파일, 폴더 tmp 폴더로 다 복사

cp -r /etc /tmp : etc 폴더 자체를 옮긴다

4) mv(move) : 파일 or 디렉토리 이동, 파일명 변경

mv 옮길 디렉토리 옮겨갈 디렉토리

mv /lab /home/사용자명 : lab 디렉토리 ⇒ home/사용자

mv /lab /home/사용자명/lab2 : 디렉토리 옮기면서 이름도 바꾸겠다.

mv test test2 : test 파일 test2 파일로 이름 변경

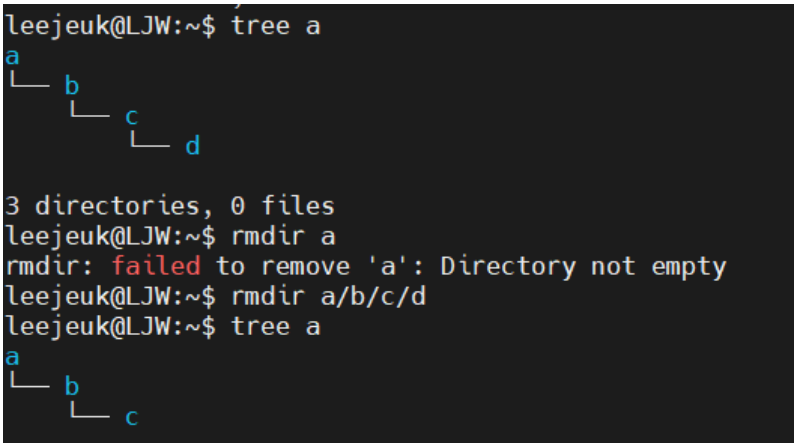
5) mkdir(make directory)

mkdir 생성할 디렉토리명

mkdir -p a/b/c/d : 없는 상위 디렉토리를 다 만들면서 만든다.

6) rmdir(remove directory): 빈 디렉토리만 삭제 가능

rmdir a/b/c/d : 다 지우는 것이 아닌 맨 뒤의 d만 지우고 a/b/c는 남아있다.



이럴 때 rm -r을 사용하게 되면 cp -r처럼 다 지운다.

7) rm(remove) : 파일 삭제

-r : 디렉토리까지 삭제

8)

Rocky

```
init 0
nmcli con del ens160
nmcli con add type ethernet con-name ens160 ipv4.method auto
ip addr
```

```
John the Ripper 설치
Rocky>
dnf install wget
wget https://www.openwall.com/john/k/john-1.9.0-jumbo-1.tar.xz
xz -d john-1.9.0-jumbo-1.tar.xz
dnf install tar
tar xf john-1.9.0-jumbo-1.tar
cd john-1.9.0-jumbo-1
dnf install gcc openssl openssl-devel
cd src
./configure
dnf install make
make -s clean && make -sj4
```

vi

- esc 누르고 : q 는 나가짐
- esc 누르고 set number 행번호

- esc 누르고 : u 는 undo
- esc 누르고 : o는 입력
- 대문자 a 제일 오른쪽 그냥 a는 한칸 오른쪽
- i는 위의 반대
- Vim은 복사, 잘라내기 및 붙여넣기를 위한 고유한 용어를 가지고 있습니다. 복사를 yank(y), 잘라내기를 delete(d), 붙여넣기를 p(p)라고 합니다.
- %s/찾을 문장/바꿀 문장
- esc gg 맨 위로
- esc G 맨 아래로
- 저장하고 나가기 wq 안하고 나가기 q!