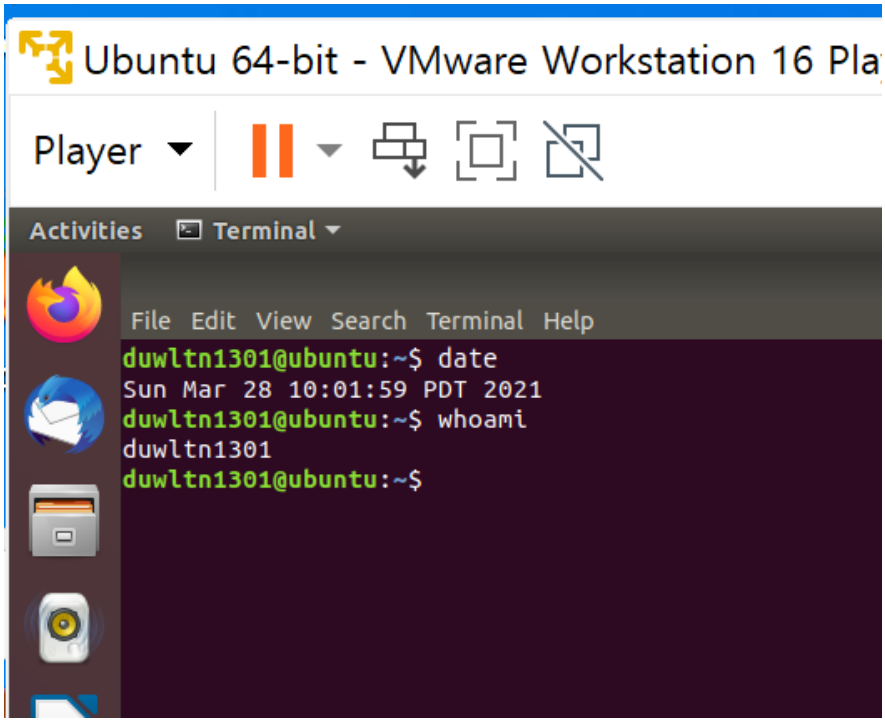
HW2. 리눅스 설치 및 명령어 실습

060분반 부산대학교 정보컴퓨터공학부
202055565
여지수

제출일: 2020-03-29

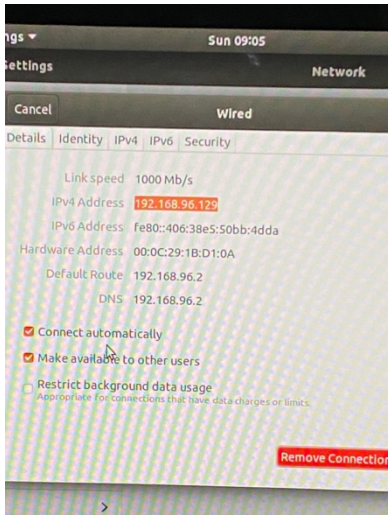
.

1. 리눅스 설치 완료 캡쳐 (30점)

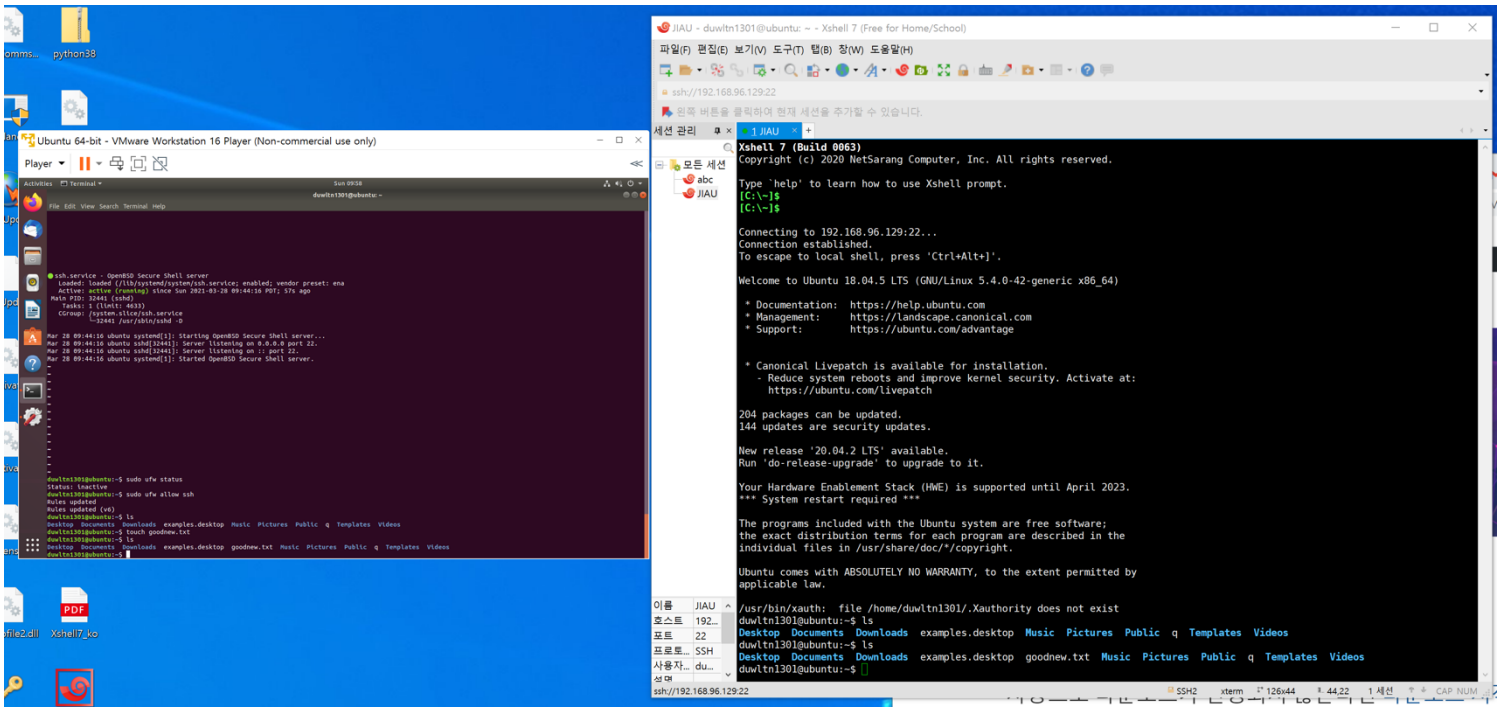


- 설치 완료 후 터미널 실행 후 date와 whoami 명령 실행 화면 첨부

2. ssh 클라이언트를 이용한 접속 (30점)



- Vmware에 설치한 리눅스의 터미널 창에서 Openssh 서버를 설치한다.
- Ssh 클라이언트 Xshell을 설치하고 xshell에서 새 새션을 등록한다. 새션에 등록할 호스트 이름은 설치한 vmware에 설치한 리눅스에서의 ip address (192.168.96.129로 등록했음)로 입력한



다. 호스트 키를 수락 및 저장하면 접속 가능하게 된다.

3. 관련 명령 조사(30점)

- Ssh 서버의 기능과 목적

우선 **SSH**(**S**ecure **Sh**ell)는 원격지 호스트 컴퓨터에 접속하기 위해 사용되는 인터넷 프로토콜이다. 뜻 그대로 보안 셸이다. 기존의 유닉스 시스템 셸에 원격 접속하기 위해 사용하던 텔넷은 암호화가 이루어지지 않아 계정 정보가 탈취될 위험이 높으므로, 여기에 암호화 기능을 추가하여 1995년에 나온 프로토콜이다.(SSH는 암호화 기법을 사용하기 때문에, 통신이 노출된다고 하더라도 이해할 수 없는 암호화된 문자로 보인다.) 셸로 원격 접속을 하는 것이므로 기본적으로 CLI 상에서 작업을 하게 된다. 기본 포트는 22번이다.

ssh 서버는 ssh 구성 중 하나이다. SSH는 웹과 유사하게 SSH Client와 SSH Server로 구성되어 있다. SSH 클라이언트와 SSH 서버의 관계로 상호작용하면서 SSH 서버가 설치된 운영체제를 제어한다. 클라이언트와 서버 사이에는 강력한 암호화 방법을 통해서 연결되어 있기 때문에 데이터를 중간에서 가로채도 해석 할 수 없는 암호화된 문자만이 노출된다. 지금까지는 Telnet을 주로 사용했는데 이것을 대체하기 위한 통신 방법이다

SSH 프로토콜을 통해 Client가 명령을 내리고 Server가 명령을 받아 컴퓨터 제어한다. 제어의 주체가 되는 컴퓨터에는 SSH Client가 설치되어 있어야 한다. 리눅스와 Mac과 같은 Unix 계열의 운영체제는 기본적으로 SSH 클라이언트가 설치 되어 있기 때문에 SSH를 이용하기 위해서 특별한 클라이언트가 필요하지 않다. 하지만 SSH 는 윈도우 운영체제에는 SSH 클라이언트가 설치되어 있지 않다. SSH 클라이언트를 설치해야 윈도우에서 Unix 계열의 운영체제를 제어 할 수 있다. 윈도우는 SSH Client를 제공하지 않기 때문에 Xshell, PuTTY와 같은 프로그램을 이용해서 다른 컴퓨터에 접속할 수 있다. SSH는 Unix 계열의 운영체제를 원격에서 제어하기 위한 방법이다. 그렇기 때문에 원격지에 있는 윈도우 운영체제를 SSH로 제어 하는 것은 일반적이지 않다. 윈도우에는 클라이언트 운영체제로 사용할 수 있을 뿐이다. 유닉스 계열의 운영체제에서는 OpenSSH가 가장 많이 사용된다.

OpenSSH는 SSH 클라이언트와 서버를 포함한다. Mac은 SSH 클라이언트와 서버가 이미 설치 되어 있기 때문에 SSH를 이용하기 위해서 특별한 조치는 필요하지 않다.

그렇다면 FTP나 Telnet과 같은 다른 컴퓨터와 통신을 위해 사용되는 프로토콜도 있는데 SSH를 사용하는가를 생각해볼 수 있다 그 이유는 보안이다. 만일 예로 언급한 두 프로토콜을 통해 민감한 정보(예를 들어 로그인 정보)를 주고받는다면 정보를 직접 네트워크를 통해 넘기기 때문에 누구나 해당 정보를 열어볼 수 있어 보안에 상당히 취약하다. 반면 SSH는 먼저 보안적으로 훨씬 안전한 채널을 구성한 뒤 정보를 교환하기 때문에 보다 보안적인 면에서 훨씬 뛰어나다. SSH는 다른 컴퓨터와 통신을 하기 위해 접속을 할 때 우리가 일반적으로 사용하는 비밀번호의 입력을 통한 접속을 하지 않는다. 기본적으로 SSH는 한 쌍의 Key를 통해 접속하려는 컴퓨터와 인증 과정을 거치게 된다. Public Key를 통해 메시지를 전송하기 전 암호화를 하게 된다. Public Key로는 암호화는 가능하지만 복호화는 불가능하다. 그리고 이와 쌍을 이루는 Private Key는 절대로 외부에 노출이 되어서는 안되는 Key로 본인의 컴퓨터 내부에 저장하게 되어있다. 이 Private Key를 통해 암호화된 메시지를 복호화 할 수 있다. 이러한 Private Key와 Public Key를 통해 다른 컴퓨터와 통신을 하기 위해서는 먼저 Public Key를 통신하고하는 컴퓨터에 복사하여 저장한다. 그리고 요청을 보내는 클라이언트 사이드 컴퓨터에서 접속 요청을 할 때 응답을 하는 서버 사이드 컴퓨터에 복사되어 저장된 Public Key와 클라이언트 사이드에 해당 Public Key와 쌍을 이루는 Private Key와 비교를 하여 서로 한 쌍의 Key인지 아닌지를 검사한다. 이렇게 서로 관계를 맺고 있는 Key라는 것이 증명이 되면 비로소 두 컴퓨터 사이에 암호화된 채널이 형성이 되어 Key를 활용해 메시지를 암호화하고 복호화하며 데이터를 주고 받을 수 있게 된다.

- Apt 명령어의 기능과 사용법

명령어 aptAdvanced Packaging Tool의 줄임말로, Ubuntu를 포함한 Debian 계열의 리눅스에서 쓰이는 패키지 관리 명령어 도구를 말한다. apt은 패키지 관련 관리도구이기 때문에 대부분 관리자 권한으로 실행되어야 하고 그래서 sudo와 함께 사용되기 마련이다. 기본적으로 Apt 는 패키지 간 의존성 문제에 대한 해법을 제시하고 이에 요구되는 패키지를 찾아내며, 실질적으로 (응용 프로그램) 패키지 설치와 삭제를 담당하는 dpkg라는 별개의 도구와 함께 동작한다. Apt 는 **매우** 강력하며, 주로 (콘솔/가상 터미널)을 통해 명령행에서 사용된다. 하지만, 명령행 인터페이스를 거치지 않고도 GUI-그래픽적인 방법으로 손쉽게 사용할 수 있도록 도와주는 도구들이 많이 존재한다.

현 시점에서 APT 집합체들과 유연하게 사용되어 권장되는 도구로 aptitude 가 있다. APT 도구들은 aptitude 로 처리하기 힘든 특수 관리 액션을 처리할 때, 또는 특히 의존성 문제로 더욱 민감한 상황일 때 사용되는 편이다.

Apt 명령어에 대해 살펴보면 다음과 같다.

install은 패키지를 설치하는 명령어다. 패키지 명을 여러개 지정하여 여러 패키지를 한꺼번에 설치할 수 있다. 이때 패키지명은 공백으로 구분한다.

remove 명령은 저장한 패키지 만을 제거한다. 패키지 설정 파일은 삭제되지 않는다.

purge 명령은 패키지와 관련 파일을 모두 제거한다.
autoremove 명령은 현재 사용되지 않는 패키지를 제거한다.
update 명령은 시스템 패키지 목록을 업데이트한다.
upgrade 명령은 전체 패키지를 최신 버전으로 업데이트한다.
full-upgrade 명령은 전체 시스템을 새 버전으로 업그레이드한다. 시스템 업그레이트가 필요할 경우 기존 패키지를 삭제하기도 한다.
search는 패키지의 검색명령이다.
show는 패키지의 버전, 분류, 의존성 패키지, 다운로드 위치, 사이트 등 주요 정보를 출력한다.
list는 레파지토리에 등록된 패키지 목록을 조회한다. 옵션을 지정하여 상태별 패키지 목록을 조회할 수 있다. -installed 옵션에서는 설치된 패키지 목록 조회를, -upgradable 옵션에서는 설치된 패키지 중 업그레이드 대상 패키지 목록 조회를, all-versions 옵션에서는 패키지의 모든 버전 목록 조회를 한다.

- Sudo 명령어의 기능과 사용법

sudo 명령어는 유닉스 및 유닉스 계열 운영 체제에서, 다른 사용자의 보안 권한, 보통 슈퍼유저로서 프로그램을 구동할 수 있도록 하는 프로그램이다.
udo 명령어를 실행하기 전에, 사용자들은 비밀번호를 입력한다. 한번 승인되고 만약 /etc/sudoers 설정 파일이 그 유저를 승인한다면, 명령은 실행된다. kdesu, kdesudo, gksu, gksudo와 같이 GUI 환경에서 사용할 수 있는 몇몇 명령어 들이 있다.
데비안 계열에만 있는 sudo 명령어는 'substitute user do' - "다른 사용자의 권한으로 실행"이라는 약자를 가지고 있는 명령어이다. 이 명령어를 사용하는 이유는 위에 있는 글처럼 보안상 이유 때문이다. 만약, 일반 계정이 중요한 파일들과 명령어들을 마구잡이로 남발하면, OS 시스템에 치명적이고 보안에 매우 취약해질 것이다. 그래서 명령어와 파일들에 권한 및 소유권 개념이 생기고 일반 계정들은 네트워크 설정 및 시스템 설정을 root 권한을 얻어야지만 수정이 가능하게 되었다. 하지만, 일반 계정에서 루트 계정으로 로그인하고 프로그램을 실행시키고 루트 계정을 빠져나오고 이러한 방법은 효율적이지도 않고 개발자들 사이에서는 매우 귀찮은 일이다. 특히 루트 계정을 계속 이용하게 된다면 파일들의 소유권과 권한이 뒤죽박죽이 되어서 나중에 시스템 오류가 생길 확률도 높아지게 된다. 그래서 나온 것이 sudo 명령어이다.
sudo 명령어는 sudo [command] 혹은 sudo -u [other user id] [command] 형식으로 root 권한 및 다른 유저 권한으로 실행할 수 있게 만든 명령어로써 root 권한으로 실행 및 다른 계정으로 실행해야 할 경우 sudo 명령어로만 사용하여, 특정 부분만 root 권한 혹은 다른 계정으로 변경하여 명령어를 실행 할 수 있는 매우 유용한 명령어이다. Sudo 다음에 실행할 명령을 입력하면 root 권한으로 명령어를 실행하는 데, 실행 전 현재 사용자의 비밀 번호를 물어본다. Sudo-i또는 sudo-s옵션을 사용하여 root계정으로 전환이 가능한다. 이때 s옵션은 현재 디렉토리를 유지하지만, i옵션은 /root 디렉토리로 이동한다.

4. 논의 (10점)

- ssh는 다른 컴퓨터와 통신을 하기 위한 접속을 할 때 한 쌍의 key를 통해 접속하려는 컴퓨터와 인증과정을 거치게 된다는 것을 새롭게 알게 되었다. Ssh가 보안에 강해서 사용한다는 점은 알고 있었지만 보안이 어떻게 강할 수 있는 지에 대해서는 알 지 못했는데 ssh 서버에 대해 공부를 해보면서 새롭게 알게 된 것이다. 두 컴퓨터 사이에 암호화된 채널이 형성되어, key를 활용해 메세지를 암호화하고 복호화하며 데이터를 주고 받을 수 있다는 것을 알게 되었다. 그리고 숙제를 하는 도중 리눅스 터미널 창에서 install net-tools를 할 때 강의와는 다른 error가 떠서 몇시간을 끙끙댔던 어려움이 있었다. 다행히 구글링을 통해 잘 해결했다.