리눅스 서버 관리자를 위한 보안

KISTI 슈퍼컴퓨팅센터

- 목 차 -

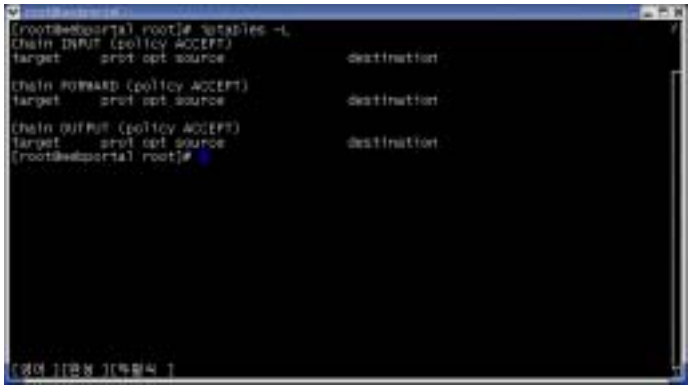
- 그림 목차 -

제 1 장 리눅스 서버 보안 지침서 (iptables 사용법을 중심으로)

기존의 매뉴얼이나 웹에 대부분의 내용들이 상세히 나와 있으나 현재 많이 쓰이는 리눅스의 사용법에 잘 맞지 않는 경우가 있고 불필요한 부분들이 너무 강조되어 쓰는데 혼동을 줌으로 다음 지침에 따라 설정하면 쉽고 빠르게 iptables를 자신의 서버에 적용하여 사용할 수 있을 것이다.
TCP_Wrapper 또한 서버 보안의 한 방법으로 많이 쓰이고 있으나 iptables가 더 강력하게 커널과 연계하여 보안 설정을 할 수 있으므로 TCP_Wrapper 보다는 iptables를 보다 권장한다.

1.1 iptables 기본 사용법

다음 환경은 redhat 9 이며 모든 명령어는 루트 권한으로 실행한다.
우선 현재 자신의 서버에 iptables의 설정 상태를 알아본다. iptables -L 명령어를 입력한다.



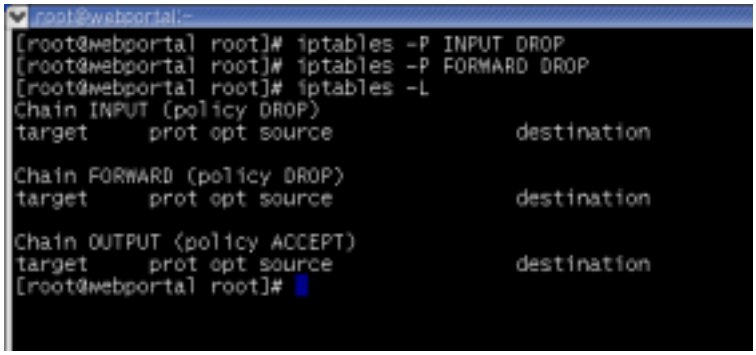
리눅스를 인스톨 한 후 따로 설정을 하지 않았다면 모든 policy가 ACCEPT로 되어 있을 것이다. 각 chain에 대한 의미는 man page나 여타 메뉴얼을 참조한다. 여기서는 자세한 설명은 자제하고 간단하고 쉽게 필요한 사항만을 적용할 수 있는 방법만 설명한다.

만일 위 그림과 같은 결과가 나오지 않는다면 다음 명령어로 iptables의 정책을 초기화한다.

iptables -F

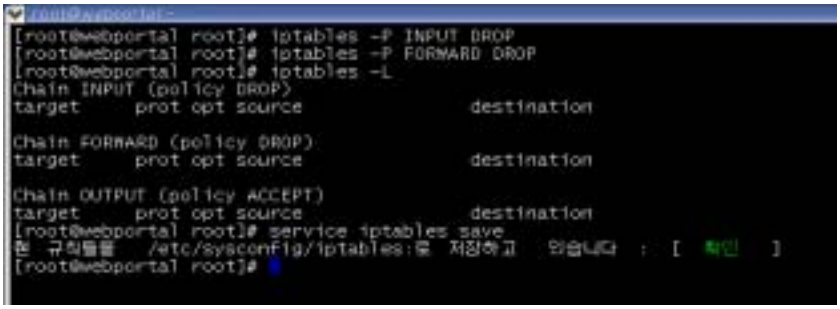
자, 이제부터 iptables의 정책을 만들어 가보자.
디폴트 정책은 모든 경우(INPUT, FORWARD, OUTPUT)에 대하여 ACCEPT이다. 이는 보안에 큰 문제가 될 수 있으므로 우선 다음의 명령어로 이를 변경한다

iptables -P INPUT DROP --> 모든 INPUT에 대하여 접근 금지
iptables -P FORWARD DROP --> 모든 FORWARD에 대하여 접근 금지
iptables -L --> iptables의 정책을 본다



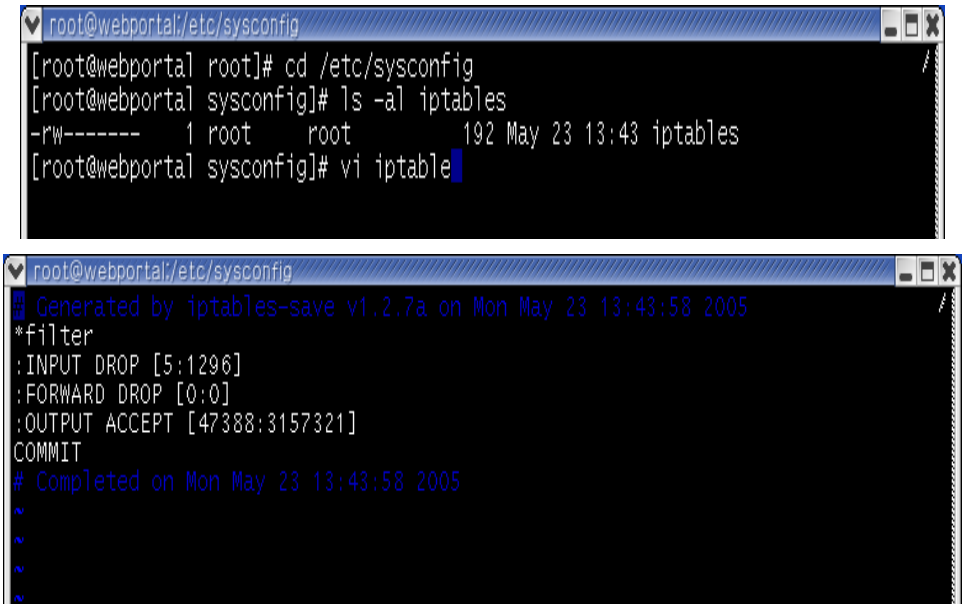
위의 명령에 의해서 변경된 정책은 서버가 재부팅되면 사라져 버리므로 일단 변경된 정책을 저장해 보자. 웹이나 여타 매뉴얼에서는 /etc/sysconfig 디렉터리에 iptables라는 파일을 만들어서 위에서 입력했던 명령어를 쓰고 저장하라고 나와 있는 경우가 있는데 이것은 리눅스의 old 버젼에서 사용되었던 방법이며 redhat 9 나 현재의 최신버젼에서는 그렇게 저장하면 않되고 다음의 명령어를 써야한다.

service iptables save



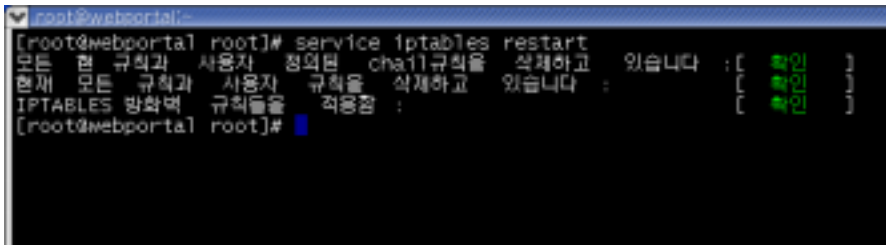
위의 명령에 의해서 변경된 정책은 저장되는데 이것이 저장되는 위치는 /etc/sysconfig 디렉터리이며 저장되는 파일은 iptables이다. 기존에 적용된 정책이 없거나 리눅스를 인스톨 한 후 아무런 설정이 없었으면 iptables 파일은 /etc/sysconfig 디렉터리에 존재하지 않는다. 즉 service iptables save 명령어에 의해서 iptables 파일이 만들어지고 정책이 저장되게 된다.
그럼 저장된 정책을 잠깐 보도록 하자.

```
# cd /etc/sysconfig
# ls -al iptables
# vi iptables
```



이 iptables 파일에 저장된 내용을 vi 편집기로 수정하지 않도록 한다.
자 그럼 정책도 바꾸었고, 바꾼 정책도 저장했으면 정책에 의해서 외부로부터의 access는 모두 거부되어야 한다. 그러나 여전히 access가능할 것이다. 마지막으로 한가지를 더해 주어야 하기 때문이다. 다음 명령어에 의해 iptables 서비스를 재가동한다.

service iptables restart



이제 제대로 정책이 적용되어 외부로부터 access는 완전 차단되었을 것이다.
이제부터 필요한 서비스를 하나하나 오픈하는 정책을 적용해 보도록 한다.
iptables의 정책적용 개념은 일단 다 막아 놓고 필요한 서비스만 오픈하는 것이다.
첫번째 정책 시나리오는 다음과 같다. A(150.183.122.68)라는 컴퓨터로 부터 리눅스 서버로 접속하느데 접속 허용은 A라는 컴퓨터만 허용하고 허용되는 서비스는 ftp, ssh이다.

이를 위해서 다음 명령어를 사용한다.

iptables -A INPUT -i lo -j ACCEPT --> localhost에서 traffic을 받아들임
iptables -A INPUT -p tcp ! --syn -j ACCEPT --> 확립된 연결에 대한 패킷을 받아들임
iptables -A INPUT -s 150.183.122.68 -p tcp --dport 20:22 -m state --state NEW,ESTABLISHED -j ACCEPT
service iptables save
service iptables restart

위에서 첫번째와 두번째 명령어는 필수적으로 입력하도록 하고 세번째 명령어중 NEW,ESTABLISHED 는 띄여쓰기 하면 않된다.
iptables -L 명령어를 이용하여 INPUT chain에 적용된 정책을 볼 수 있으며 /etc/sysconfig/iptables 파일에도 역시 같은 내용이 저장되어 있다.
특정 ip가 아니라 모든 ip로 부터 ftp, ssh접근을 허용하려면 단지 -s 옵션 부분을 빼면 된다.

```
# iptables -A INPUT -p tcp --dport 20:22 -m state --state NEW,ESTABLISHED -j ACCEPT
# service iptables save
# service iptables restart
```

--dport는 --destination-port와 같은 의미이다. --dport 뒤의 포트 넘버는 번호 대신 ftp, telnet, ssh같은 서비스 명으로 대신할 수 있다.
두번째 정책 시나리오는 다음과 같다. 모든 컴퓨터로 부터 리눅스 서버로 접속하느데 허용되는 서비스는 web(80번)이다.
이를 위해서 다음 명령어를 사용한다.

```
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
# service iptables save
# service iptables restart
```

INPUT chain에 2개의 정책이 적용된 것을 볼 수 있다.

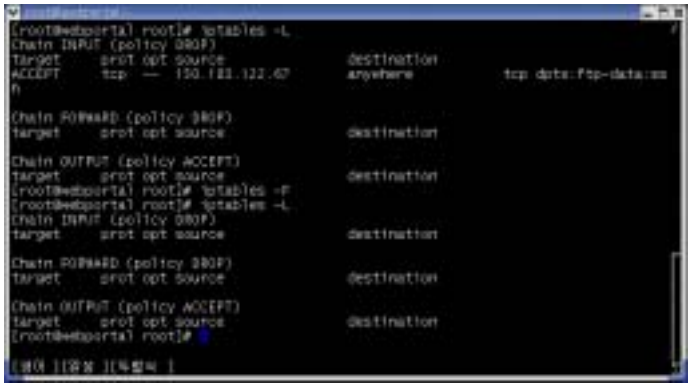
1.2 적용된 정책을 수정하거나 지우는 방법

적용된 정책을 변경하려면 설정되었던 정책을 지우고 다시 하는 적용하는 수 밖에 없다. 적용된 정책을 지우는 방법은 다음과 같다.

```
# iptables -F
```

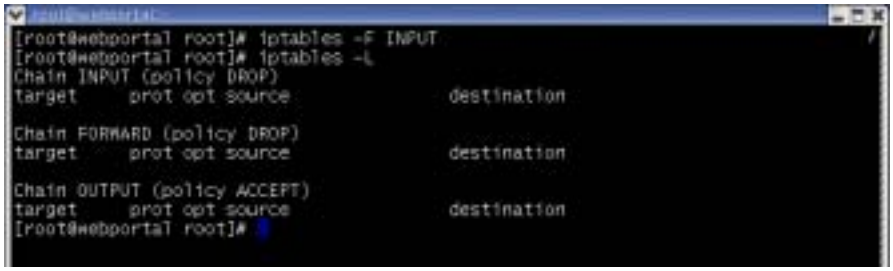
이 명령어는 설정된 정책을 모두 지우고 초기화한다. 여기서 설정된 정책이란 # iptables -A INPUT -p tcp --dport 80 -j ACCEPT 같은 명령어로 적용된 정책을 말한다. 즉 # iptables -P INPUT DROP 등으로 적용된 chain 정책은 # iptables -F 명령어로 변경되지 않으며 # iptables -P INPUT [ACCEPT|DROP] 과 같이 chain정책 변경 명령어를 사용한다.

다음을 보면 이해할 수 있을 것이다.

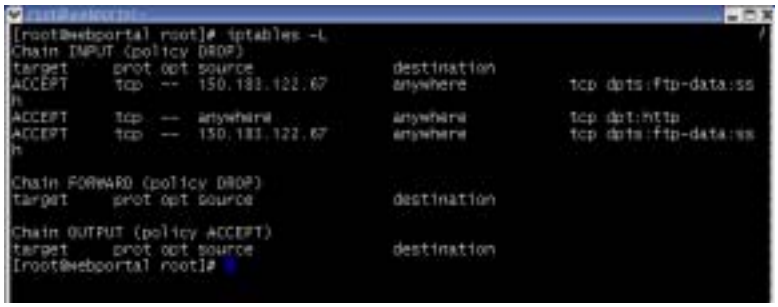


iptables -F 명령어는 INPUT, FORWARD, OUPUT에 적용된 모든 정책을 한꺼번에 초기화한다. 그러나 경우에 따라서 INPUT, FORWARD 또는 OUPUT에 각각 적용된 정책을 따로따로 초기화할 필요가 있을 것이다. 그럴 경우는 다음과 같이 chain 명을 -F 옵션 뒤에 입력 한다.

iptables -F INPUT

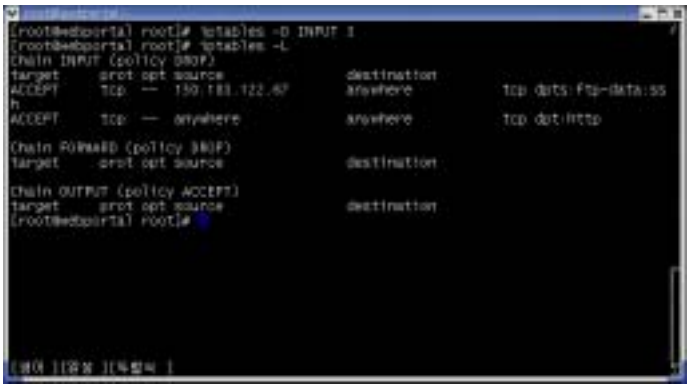


그러나 같은 chain내에서도 정책에 여러개 존재할 수 있다. 다음의 경우은 INPUT chain에 3개의 정책이 적용되어 있다.



첫번째와 세번째 정책은 동일한 것이다. 다만 사용자가 실수로 2번 정책을 입력한 것이다. 이렇게 동일한 정책을 다시 입력하더라도 iptables는 또다른 정책으로 인식하고 받아들인다. 이제 실수로 입력한 정책을 지우도록 한다. 지우고 싶은 정책은 INPUT chain의 3번재 정책이다. 이럴 경우는 다음과 같이 -D 옵션을 사용하며 chain 명을 -D 옵션 뒤에 입력 한다.

iptables -D INPUT 3



정책의 변경 또는 삭제 후에 반드시 다음 명령어를 수행해야 제대로 적용되므로 잊지말고 실행해야 한다.

service iptables save
service iptables restart

1.3 종합

종합적으로 다음과 같은 룰셋을 적용함으로서 시스템의 안전을 강화할 수 있다.

```
# iptables -F --> 기존의 iptables의 정책 정리
# iptables -P INPUT DROP --> 모든 INPUT에 대하여 접근 금지
# iptables -P OUTPUT ACCEPT--> 모든 OUTPUT에 대하여 허용
# iptables -P FORWARD DROP --> 모든 FORWARD에 대하여 접근 금지
# iptables -A INPUT -i lo -j ACCEPT --> localhost에서 traffic을 받아들임
# iptables -A INPUT -p tcp ! --syn -j ACCEPT --> 확립된 연결에 대한 패킷을 받아들임
# iptables -A INPUT -s 150.183.122.68 -p tcp --dport 20:22 -m state --state NEW,ESTABLISHED -j ACCEPT --> 소스(150.183.122.68)에서만 ssh, ftp 접근 허용
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT --> 웹서비스 허용
# iptables -A INPUT -s 150.183.135.135 -p udp --sport 53 -m state --state NEW,ESTABLISHED -j ACCEPT --> KISTI DNS 서버
# iptables -A INPUT -s 150.183.95.96 -p udp --sport 53 -m state --state NEW,ESTABLISHED -j ACCEPT --> KISTI DNS 서버
# service iptables save
# service iptables restart
```

제 2 장 SSH(Secure Shell) 사용법

2.1 SSH(Secure Shell)

2.1.1 개요

SSH(Secure Shell)은 네트워크의 다른 컴퓨터에 로그인 할 수 있으며 원격 시스템에서 명령을 실행하고 다른 시스템으로 파일을 복사할 수 있도록 해주는 프로그램이다. 강력한 인증방법과 안전하지 못한 네트워크에서 안전하게 통신을 할 수 있는 기능을 제공해 준다.

SSH는 두 호스트간의 통신 암호화와 사용자 인증을 위하여 공개 열쇠 암호 기법을 사용한다. 즉, telnet, rlogin, rcp등과 비교해 보면 이들은 스니퍼를 당하면 입력 문자 그대로 패킷이 쉽게 노출된다. 이에 반해 SSH는 이 모든 문자들을 암호화 하여 비록 노출이 된다 하더라도 이해할 수 없는 암호화된 문자로 나타나게 되는 것이다. 또 IP스푸핑, DNS 스푸핑 등으로부터 SSH를 사용하면 보호가 가능하다. 세션 하이젝킹(Session Hijacking)과 DNS 스푸핑을 방지해 주면서 원격 호스트에 로그인 하거나 호스트끼리 데이터를 복사하기 위해 사용될 수 있다. 일반 login프로그램과 달리 패킷 전송 시 암호화하기 때문에 원격 관리의 보안이 매우 안정적이다

□ SSH에서 제공하는 기능

SSH에서 지원하는 서비스는 다음과 같다.

- 버클리 r-utilities(rlogin, rsh, rcp, rdist)들에서 쓴 네트워크 데이타 통신의 암호화.
- FTP에서의 데이타 통신의 암호화.
- X 세션에서의 데이타 통신의 암호화.

□ SSH의 기본원리

보안쉘의 기본적인 개념은 데이타의 흐름에 있어서 직접적인 데이타의 송수신이 아니라 데이타를 한번 암호화 해서 송신하여 데이타의 유출이나 변조를 막아서 데이타 송수신시에 보안을 높이는 것이다.

2.1.2 SSH(Secure Shell) 프로그램 다운로드 하기

telnet으로 접속하기 위해서 netterm이나 CRT등과 같은 telnet client프로그램이 필요한 것과 마찬가지로 SSH로 서버에 접속하기 위해서는 SSH client가 필요하다. 즉 사용자 플랫폼에 맞는 SSH client를 다운로드 하여 설치해 주어야 한다.

접속이 이루어지고 난 후에는 telnet과 SSH의 사용방법이 모두 동일하다.

□ UNIX(Linux, AIX, HP-UX, Solaris, Tru64-Unix, SCO, IRIX 등)

비상업적인 용도로 사용할 수 있는 SSH 프리웨어 버전인 OpenSSH를 다음 사이트(*)(**) 에서 소스 코드 혹은 UNIX 벤더에 따른 설치 패키지를 다운로드할 수 있다.

(*) ftp://linux.sarang.net/mirror/network/daemon/security/ssh

(**) http://www.openssh.org

```
< AIX >
#ssh -V 또는 #lslpp -L | grep ssh
< Linux >
#ssh -V 또는 #rpm -qa | grep ssh
< Solaris >
#ssh -V 또는 #pkginfo | grep ssh
```

그러나 최근에 발표된 대부분의 UNIX 계열의 OS는 SSH client가 기본으로 설치되어 있어 별도의 설치 과정을 할 필요 없다. 다음 명령어에 의해 SSH client의 설치유무를 확인한 후 없을 시 별도의 인스톨 과정을 거쳐서 자신의 워크스테이션에 SSH client를 설치한다.

□ 윈도우즈 사용자

- http://www.openssh.org 에서 윈도우용 SSH(puTTY)를 다운받아서 사용
- 또는 ftp://linux.sarang.net/mirror/network/daemon/security/ssh 에서 SSHSecureShellClinet-3.2.9.exe 파일을 다운받아 사용
- 또는 http://www.zip.com.au/~roca/ttssh.html 에서 Teraterm Pro를 다운 받아서 사용

- 또는 기타 Zterm등 인터넷에서 공개용으로 사용할 수 있는 유틸리티는 아무 것이나 사용할 수 있다.

2.2 SSH 사용하기

2.2.1 사용자의 컴퓨터가 UNIX기반일 경우

□ 기본 사용법(패스워드 기반 인증) - 가장 간단하며 쉽게 사용할 수 있는 방법

SSH를 사용하여 슈퍼컴퓨터에 접속하기 위해서는 사용자 워크스테이션에 SSH client 가 설치되어 있고, 슈퍼컴퓨터에 사용자 계정이 등록되어 있어야만 한다. 슈퍼컴퓨터에 접속하기 위해서는 다음과 같은 명령을 사용하면 된다.

 # ssh -l 사용자ID 슈퍼컴퓨터호스트이름(예> #ssh -l alice nobel)
또는
#ssh 사용자ID@슈퍼컴퓨터호스트이름(예> #ssh alice@nobel)

다음은 각 호스트별로 SSH를 이용하여 접속하는 방법을 나타낸다. 계정 아이디는 alice라 가정한다.

호스트	슈퍼컴퓨터에 SSH를 이용하여 접속하는 방법
IBM	**#ssh -l alice nobel** 또는 **#ssh -l alice nobela** 또는 **#ssh -l alice nobelb** (위 3가지 경우는 사용자 워크스테이션의 /etc/hosts 파일에 {nobel, nobela, nobelb}.supercomputing.re.kr의 ip가 등록되어 있는 때 가능) 또는 **#ssh -l alice nobel.supercomputing.re.kr** 또는 **#ssh -l alice nobela.supercomputing.re.kr** 또는 **#ssh -l alice nobelb.supercomputing.re.kr** (위 6가지 중 하나를 typing 하면 된다. nobel은 nobela와 같은 호스트이다. 즉 로그인 노드는 nobela, nobelb 2개를 운영하고 있다)

호스트	슈퍼컴퓨터에 SSH를 이용하여 접속하는 방법
HP	#ssh -l alice frontsmp 또는 #ssh -l alice frontsmp.supercomputing.re.kr
NEC	#ssh -l alice necsx6a 또는 #ssh -l alice necsx6a.supercomputing.re.kr
PC cluster	#ssh -l alice 150.183.5.212 또는 #ssh -l alice 150.183.5.213 또는 #ssh -l alice 150.183.5.214 또는 #ssh -l alice 150.183.5.215

로그인 과정에서 슈퍼컴퓨터의 공개키에 대한 fingerprint를 확인하는 절차를 거치게 되는데, 이 fingerprint는 올바른 서버에 접속했는지를 판단하기 위해서 필요한 것으로 서버 관리자가 제공한 fingerprint와 일치하는 지를 확인하고 일치한다면, '예' 버튼을 눌러 다음 단계로 진행한다. 다음으로 사용자 ID에 매핑되는 패스워드를 입력하여 인증 절차를 완료하고 시스템에 로그인 한다.

```
#ssh -l alice nobela.supercomputing.re.kr
The authenticity of host 'nobela (150.183.5.101)' can't be
established.
RSA key fingerprint is
43:fc:e4:51:56:a4:c9:29:17:03:eb:dd:62:26:7e:3a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'nobela,150.183.5.101' (RSA) to the list
of known hosts.
alice@nobela's password:
```

Note 이 fingerprint를 확인하는 절차는 서버에 처음 접속할 때만 나타난다.

Note

사용자가 보기에 접속 과정이 일반 telnet과 유사하다고 느끼겠지만, 실제로는 사용자 패스워드 입력 및 앞으로의 명령 수행과정에서 주고받는 모든 데이터들은 암호화되어 안전하게 서버와 사용자 워크스테이션 사이에서 송수신 된다는 점에 유의하자.

□ 공개키 기반 인증

○ passphrase를 입력하여 인증하는 방식

- 사용자 워크스테이션에서 키 생성하기

> Note 이 작업은 UNIX 기반인 자신의 워크스테이션 및 컴퓨터에서 수행하는 작업이다.

현재 슈퍼컴퓨터가 지원하고 있는 SSH 버전 2에서는 RSA와 DSA 키를 모두 사용가능하며, 다음은 DSA 및 RSA 키를 생성하는 방법을 보여주고 있다.

```
userwork>ssh-keygen -t dsa
```

키 생성과정에서 생성된 키가 저장될 파일 이름을 지정할 것을 요청받는데, 경로를 확인하고 기본 제시된 파일 이름을 그대로 수용하여 엔터를 치면 된다.

```
Enter file in which to save the key
(/home/madhatter/.ssh/id_dsa):
```

다음으로 비밀키에 대한 안전한 보관을 위해서 passphrase[일종의 암호]를 두 번 입력한다. 이 때 passphrase는 반드시 암기하고 있어야 함에 주의한다.

```
Enter passphrase:  *****
Enter the same passphrase again:  *****
```

개인키 및 공개키는 다음과 같은 경로의 파일로 저장된다.

```
Your identification has been saved in
/home/madhatter/.ssh/id_dsa.
Your public key has been saved in
```

```
/home/madhatter/.ssh/id_dsa.pub.
```

rsa 키를 생성하는 명령은 다음과 같다.

```
userwork>ssh-keygen -t rsa
```

DSA 생성과 똑같은 과정을 거친다.

키 생성이 완료되면 *$HOME/.ssh* 디렉터리의 리스트를 확인한다.

```
userwork> ls -l .ssh
total 56
-rw-rw-r-- 1 madhatter   madhatter     79   Sep  6 09:58 config
-rw------- 1 madhatter   madhatter    736   Sep  6 07:45 id_dsa
-rw-r--r--1 madhatter   madhatter    620   Sep  6 07:45 id_dsa.pub
-rw------- 1 madhatter   madhatter    544   Aug  3 08:54 identity
-rw-rw-r-- 1 madhatter   madhatter    348   Jan  9  2001 identity.pub
-rw-------1 madhatter   madhatter   16398 Sep 10 08:57 known_hosts
-rw-r--r-- 1 madhatter   madhatter    5909 Sep 10 08:55 known_hosts2
-rw-------1 madhatter   madhatter    512 Aug  9 16:41 random_seed
```

- 사용자 홈 디렉터리의 authorized_keys 파일에 공개키 첨부하기

Note 이 작업은 KISTI 슈퍼컴퓨터에 로그인 하여 수행하는 작업이다.

사용자의 워크스테이션에서 슈퍼컴퓨터에 접속할 때 일반적인 패스워드를 사용하지 않고, 공개키를 사용하여 인증을 수행하기 위해서는 다음과 같이 앞서 생성한 사용자 워크스테이션의 키를 슈퍼컴퓨터의 사용자 홈 디렉터리의 authorized_keys 파일에 첨부하여야 한다.

```
userwork>ssh -l alice nobela.supercomputing.re.kr
alice@nobela's password :*****
nobela> mkdir .ssh
nobela> scp user@userwork.sample.edu:id_dsa.pub dsakey
Password: *****
nobela> touch .ssh/authorized_keys
nobela> cat rsakey >> .ssh/authorized_keys
nobela> exit
userwork>
```

> **Note**
>
> scp 명령은 대부분의 SSH 프로그램에 포함된 네트워크를 통해서 안전하게 파일을 전송할 수 있게 한다. 자세한 사용법은 scp man 페이지를 참조하라. 또한 기존에 .ssh/authorized_keys 파일이 존재하지 않는다면, touch 명령을 사용하여 새롭게 이 파일들을 생성해준 다음 사용자 워크스테이션에서 생성한 공개키를 이 파일에 첨부하면 된다.

위와 같이 사용자 워크스테이션에서 생성한 공개키를 슈퍼컴퓨터 사용자 홈 디렉터리의 authorized_keys 파일에 첨부하고 나서, SSH를 이용하여 슈퍼컴퓨터에 접속하면 일반 패스워드를 통한 인증이 아닌 사용자 키를 통한 인증 과정을 거치게 된다. 이 과정에서 사용자 공개키와 쌍으로 생성된 개인키에 대한 접근을 허용하도록 사용자가 생성한 passphrase를 입력할 것을 요구한다.

```
userwork> ssh -l alice nobela.supercomputing.re.kr
Please enter your passphrase for .ssh/id_dsa:  *****
nobela>
```

○ passphrase를 입력하지 않고 인증하는 방식(인증 에이전트 사용하기)

앞서 공개키 기반 인증과정에서 passphrase를 입력하는 번거로움을 해결하기 위해서 제공되는 프로그램이 인증 에이전트[ssh-agent]이다. ssh-agent는 사용자가 리모트 시스템에 로그인 시 입력해야 하는 passphrase를 기억하고 있다가 대신하여 입력해주는 역할을 수행한다. 다음은 인증 에이전트[ssh-agent]를 시작하고 ssh-add 명령을 사용하여 사용자 private 키에 대한 passphrase를 인증 에이전트에 알려주는 과정을 보여주고 있다.

```
userwork> ssh-agent $SHELL
workstation> ssh-add $HOME/.ssh/id_dsa
Enter passphrase for '.ssh/id_dsa':  *****
Identity added: id_dsa (id_dsa)
```

인증 에이전트를 사용하여 다음과 같이 로그인하면, passphrase를 입력하

지 않고 바로 슈퍼컴퓨터에 로그인 할 수 있다.

```
userwork> ssh alice@nobel.supercomputing.re.kr
--------------------------------
Welcome to nobela
--------------------------------
nobela>
```

○ X11 forwarding

X11 forwarding을 사용하면 네트워크 상에서 X 클라이언트와 서버 간에 원격 연결을 보호하는 안전한 터널을 형성할 수 있다. 다음은 X11 forwarding을 사용한 안전한 X 윈도우즈 연결을 보여주고 있다. -X 옵션은 ssh 명령이 X11 forwarding을 가능하도록 한다.

```
#ssh -l alice -X nobela
```

2.2.2 윈도우즈[PuTTY]

이 절에서는 SSH 프로그램 중에서 윈도우즈용으로 가장 많이 사용하는 PuTTY 프로그램에 대한 사용법만을 설명하고자 한다.

□ 기본 사용법

SSH2 프로토콜을 사용하여 PuTTY를 사용하기 위해서는 다음과 같이 하면 된다.

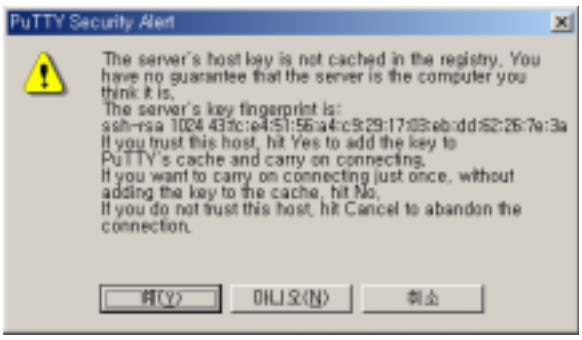
- putty.exe 파일을 탐색기로 찾아서 실행한다.
- 접속하고자 하는 슈퍼컴퓨터의 호스트이름[IP 주소]을 입력한다.
- 프로토콜로 SSH를 선택한다. 포트 번호가 22번으로 바뀌는 것을 확인할 수 있다.
- Category 부분의 Connection 메뉴을 선택하고, Auto-login username 필드에 슈퍼컴퓨터 사용자 로그인 ID를 입력한다.
- 현재 설정을 계속 사용하기 위해서는 Session 메뉴의 Saved Sessions 필드에 사용자 임의로 세션 이름을 입력하고 Save 버튼을 눌러 저장한다. Default Setting 리스트 밑에 사용자가 저장한 새로운 세션 이름이 추가되는 것을 확인할 수 있다. 사용자가 설정한 세션이 저장되면, 앞으로는 해당 세션 이름을 선택하고 Load 버튼을 눌러 설정을 불러올 수 있다.
- SSH 서버가 설치되어 있는 슈퍼컴퓨터에 접속하기 위해서는 Open 버튼을

누른다.

<그림 2-1> PuTTY 구성

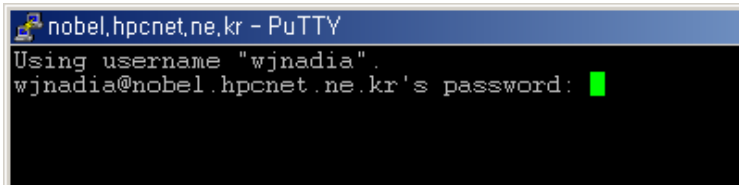
□ SSH 서버에 처음 접속되면, 경고 패널이 하나 뜬다. 이 패널은 사용자에게 서버 키 fingerprint를 표시해 준다. 이 fingerprint는 올바른 서버에 접속했는지를 판단하기 위해서 필요한 것으로 서버 관리자가 제공한 fingerprint와 일치하는 지를 확인하고 일치한다면, '예' 버튼을 눌러 다음 단계로 진행한다.

> **Note** 이 보안 경고 메시지는 서버에 처음 접속할 때만 나타난다.



<그림 2-2> PuTTY 최초 접속

□ 새로운 터미널 창에서 앞서 세션 설정 시 입력한 사용자 ID에 대한 패스워드를 입력한다. 이제 정상적인 접속이 이루어지고, 명령 입력 프롬프트가 뜨는 것을 확인할 수 있다.



<그림 2-3> PuTTY 터미널 윈도우

Note

사용자가 보기에 접속 과정이 일반 telnet과 유사하다고 느끼겠지만, 실제로는 사용자 패스워드 입력 및 앞으로의 명령 수행과정에서 주고받는 모든 데이터들은 암호화되어 안전하게 서버와 사용자 워크스테이션 사이에서 송수신된다는 점에 유의하자.

Note

서버[슈퍼컴퓨터]와 클라이언트[사용자]간에 SSH를 사용한 안전한 접속을 이루기 위해서, 다음과 같은 과정을 거치게 된다.

1. SSH 클라이언트가 서버에게 접속을 요청한다.
2. SSH 서버와 클라이언트는 서로간에 지원 가능한 SSH 프로토콜 버전을 알려준다.
3. 서버는 공개 호스트 키와 challenge를 클라이언트에게 전달한다.
4. 클라이언트는 새로운 세션 키와 challenge를 생성하고 서버에서 전달받은 공개 호스트 키를 이용하여 암호화하여 전달한다.
5. 서버는 자신의 개인키를 이용하여 클라이언트로부터 수신받은 암호화된 세션키를 복호화한다. 성공적으로 세션키가 복호화되면, 앞으로의 서버와 클라이언트 간의 통신은 이 세션키를 사용하여 암호화하여 전달하고 다시 복호화하는 방식을 통하여 안전하게 통신하게 된다.

용어 정의

-서버	: 클라이언트로부터의 SSH 접속을 기다리는 데몬을 실행하고 있는 프로그램
-클라이언트	: SSH 서버에 접속하기 위한 사용자쪽 프로그램
-사용자 키	: 사용자 신원 증명을 제공하기 위해서 클라이언트에 의해서 사용되는 비대칭키
-호스트 키	: 서버의 신원 증명을 제공하기 위해서 서버에 의해서 사용되는 비대칭키
-세션 키	: 암호화 통신을 위해서 사용되는 대칭 키. 대칭 키 암호화는 비대칭키에 비하여 암/복호 속도가 빠르다. 서버와 클라언트 간의 데이터는 대칭키 알고리즘을 사용하여 전송된다.
-비대칭 키	: 공개키와 비밀키의 쌍으로 이루어져 있으며, 세션키 보다 암/복호 속도가 느리기 때문에 주로 세션키를 전달하고 상호 인증에 사용된다.
-passphrase	: 사용자 키로 생성되는 비대칭 키의 비밀키를 안전하게 보관하기 위한 암호.
-fingerprint	: 서버로부터 전달되는 공개키가 의도하는 서버의 공개키인지를 확인하는 데 사용되는 일종의 checksum 데이터임. 서버 관리자는 사용자에게 fingerprint를 알려줌으로써 서버 접속 과정에서 이 fingerprint를 확인함으로써 올바른 서버에 접속하였음을 확인할 수 있다. 처음 서버에 접속할 때 보안 경고 메시지가 뜨면서 서버의 fingerprint를 확인하는 절차를 거치게 된다.

□ passphrase를 사용하기

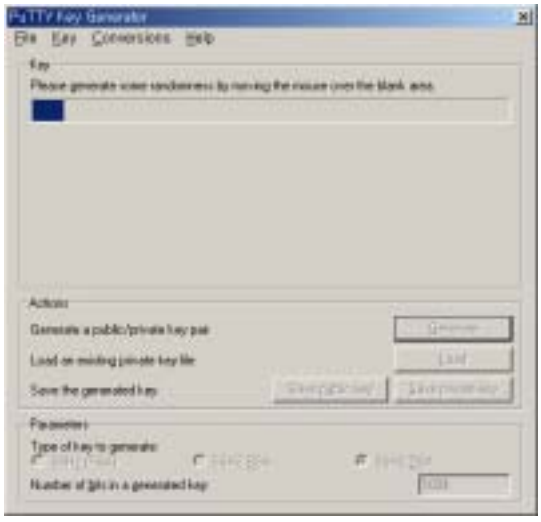
passphrase를 사용하여 클라이언트가 서버에 접속하기 위해서는 PuTTY 키 생성 유틸리티를 사용하여 공개키/비밀키 쌍을 이루는 비대칭키를 생성하여야 한다. 앞의 과정에서는 서버에서 생성한 키를 이용하여 인증을 하고 세션키를 교환하였음.

○ puttygen.exe 프로그램을 실행한다.

○ Parameters 항목에서 SSH2DSA 알고리즘을 선택하고 생성될 키의 비트수를 1024로 설정한다.

<그림 2-4> 키의 종류선택

○ Generate 키를 누르고, 키 생성 막대 그래프에서 빈공간 위에 마우스 커서를 따라 움직인다.

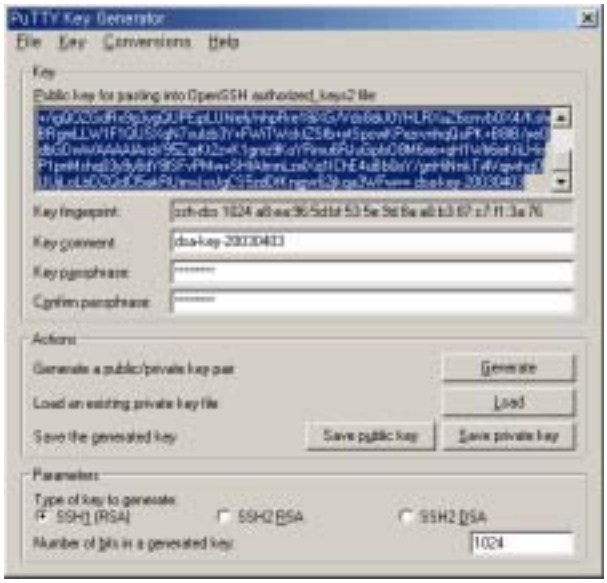


<그림 2-5> 키 생성

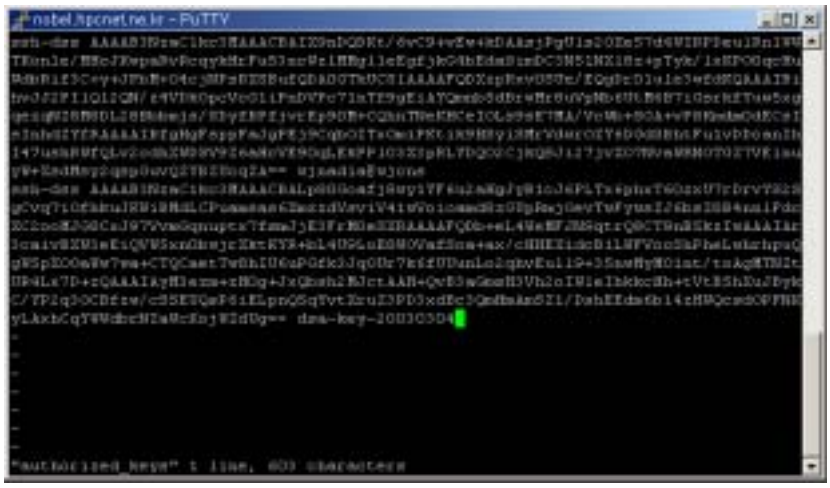
○ 키가 생성되면 key passphrase 및 Confirm passphrase에 비밀키를 안전하게 보관하기 위한 암호를 두 번 입력한다.

> **Note**
> passphrase를 잃어버리면 비밀키에 대한 접근이 불가능하므로 잃어버리지 않도록 잘 기억한다.

○ 공개키와 비밀키를 각각 저장하기 위해서 저장 버튼을 각각 누른다.
○ 생생되어 화면 상단에 표시된 Public 키 텍스트를 블록 선택하여 복사한다.
○ PuTTY.exe를 실행하여 서버에 접속하고, $HOME/.ssh/authorized_keys 파일을 vi로 열어서 조금 전에 클라이언트에서 블록 복사한 Public 키 텍스트를 터미널 윈도우에서 마우스 오른쪽 버튼을 눌러 첨부한다.



<그림 2-6> 공개키 복사



<그림 2-7> authorized_keys 파일에 사용자 공개 키 첨부하기

□ 인증 agent 사용하기

PuTTY 인증 agent를 사용하면 슈퍼컴퓨터에 접속 시에 passphrase를 입력하지 않고도 로그인이 가능해진다. 인증 agent를 사용하기 위해서는 다음과 같이 설정하면 된다.

○ pageant.exe를 실행한다.

○ 윈도우즈 하단 우측의 타스크바에서 아래 그림과 같은 아이콘을 찾아서 마우스를 대고 오른쪽 버튼을 누르고 Add Key를 선택한다.

○ 파일 선택 다이얼로그 박스에서 앞서 생성한 개인키를 선택하고 passphrase를 입력한다.

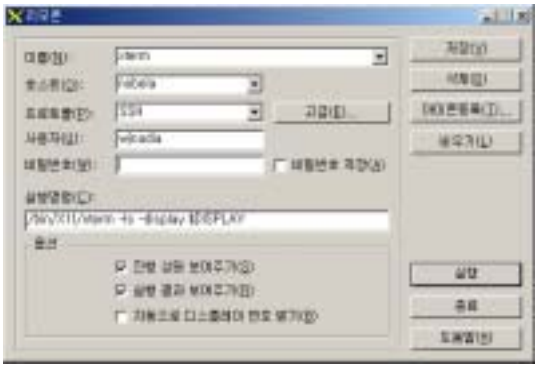
2.3 윈도우즈 환경의 X-Manager에서 SecureShell 사용하기

Xmanager 버전 1.3.9 이상에서는 리모콘에서 SSH 접속을 지원한다. 이를 사용하기 위해서는Xmanager 폴더(그룹)에 있는 리모콘을 실행하고 다음과 같은 순서에 따라 설정하면 된다.

① 이름 입력란에 "xterm-ssh"를 입력
② [프로토콜]을 SSH로 선택
③ [호스트] 입력란에 원격 UNIX 호스트의 인터넷 주소(IP Address)를 입력
④ [사용자] 입력란에 로그인 계정 이름을 입력
⑤ [비밀번호] 입력란에 로그인 계정의 암호를 입력한다.
⑥ [비밀번호 저장]을 선택, 보안이 중요할 때는 선택하지 않음
⑦ [실행명령] 입력란에 다음과 같이 xterm을 실행할 수 있는 명령을 입력
/usr/bin/X11/xterm -ls
⑧ xterm이 설치된 디렉토리는 원격 호스트에 따라 다를 수 있으므로 그에 맞게 입력

> **Note**
> SSH 프로토콜을 사용하는 경우에는 연결 시 SSH 서버가 적절한 DISPLAY 값을 부여하므로 -display 옵션이 필요 없다.

⑨ [저장] 버튼을 눌러 입력한 내용을 저장
⑩ [실행] 버튼을 클릭하면 xterm이 실행됩니다.



<그림 2-8> 리모콘 접속 설정


xterm - 원격 실행 결과
원격 호스트:
SSH 서버:
사용자:
실행명령:
실행 로그:
Trying to login as wlrada
Trying password authentication...Done.
Requesting session channel...Done.
Requesting X11 forwarding...Done.
Executing remote command...Done.
X11 channel request arrived.
Establishing X11 connection (id=2)...Done.
연결 끊기(D)

<그림 2-9> 원격 실행 결과