Connecting to Linux From Other Systems

UIC Linux Users Group

September 7, 2010

ssh: The Secure SHell

Used to access a linux computer from anywhere

Bert Example

```
hef@acm:~$ ssh ssennebo@bert.cs.uic.edu
The authenticity of host 'bert.cs.uic.edu (131.193.40.32)' can't be
established.
RSA key fingerprint is
99:6a:e7:86:1f:de:19:fd:33:05:33:e8:0b:b2:72:b8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'bert.cs.uic.edu,131.193.40.32
[ssennebo@bert] ~$
```

Objectives of Private/Public Keypair Communication

- secure
- ability to verify identify does not enable imitation of identity
- a recorded network sessions cannot be replayed or reproduced by either side

Public Key Encryption

- A keypair consists of 2 parts: a private key and a public key
- A public key is shared freely
- A private key is kept secret

The Fingerprint

- The fingerprint is a hash of the servers public key
- The client sends a message by encrypting data with the servers public key
- The server uses its private key to decrypt the message
- The server sends a response that the client uses to verify the identity of the server.

.ssh/config

.ssh/config

```
Host acm
Hostname acm.cs.uic.edu
User alice
Port 22
```

Without .ssh/config

```
ssh alice@acm.cs.uic.edu -p 22
scp -P 22 alice@acm.cs.uic.edu:/ ~/acm/
rsync -e 'ssh -p 22' alice@acm.cs.uic.edu:~/ ~/acm/
```

With .ssh/config

```
ssh acm
scp acm:/ ~/acm/
rsync acm:~/ ~/acm/
```

scp: secure copy

```
scp localfile user@remotehost:~/path/to/destination/
scp user@remotehost:~/path/to/file localfile
```

scp: secure copy

```
scp -r ~/localdir user@remotehost:~/remotedir/
scp -r user@remotehost:~/directory ~/localdir/
```

rsync

- Has prettier output, and can do updated file data only (syncing)
- rsync -auP localfile user@remotehost:/path/to/destination

SSH Keys

- SSH keys can be used as a secure alternative to password based logins
- useful for ssh base version control systems like git, mercurial, and subversion

How a Keypair Works

- similar in concept to fingerprint identification
- the server has your public key
- the server send a challenge message by encrypting data against your public key
- you use your private key to decrypt the message and prove your identity

Creating a SSH Keypair

```
ssh-keygen
save file in default location (~/.ssh/id_rsa)
enter a passphrase
confirm passphrase
```

Copy Public Key

- copy public key (.ssh/id_rsa.pub) to host computers in .ssh/authorized_keys2
- ssh-copy-id can automate this task

ssh-agent

an ssh-agent is a program that can keep your decrypted keys in memory, so that you only need to enter your passphrase once per session

ssh-agent

Most window managers come with an ssh-agent that will ask for your passphrase the first time you use a key, including gnome(ubuntu) and kde. The command line program ‘ssh-agent‘ can be used in instances where an existing ssh-agent is not available.

The problem

With ssh, or any shell application, any program you launch becomes a child of that shell. When you exit the shell, the program also exits.

Disown the Process

- press ctrl + z to suspend the process
- run ‘bg‘ to background suspended processes
- run ‘disown -h‘ to disown background processes

Disadvantages

- cannot reconnect to the process

Screen

screen is a Terminal Multiplexor. This lets us create screen sessions, and disconnect and reconnect to them freely.

Using Screen

- screen -R irc
- run irssi
- press ctrl + a,d
- connect from somewhere else (or not)
- run screen -R irc

Putty

- windows based ssh client
- support ssh-agent like abilities with paegent

WinSCP

- windows based file transfer client

Fugu

- OS X based Secure File Transfer client