IT 1100 : Introduction to Operating Systems

Chapter 16

Remote Connectivity

So far in this class we have learned the following ways to remote connect to another computer

- `wget` allows us to download content from a website
  - `wget http://cit.dixie.edu/it/1100/syllabus.pdf`
- `ssh` allows us to create a terminal connection to another computer.
  - `ssh smorgan@ssh.cs.dixie.edu`

Remote Connectivity

- `vncviewer` allows us to create a visual connection to another computer.
  - `GTK VncViewer / Vinagre` - Linux
  - `Chicken` - Mac
  - `TightVNC` - Windows, Mac, Linux

Remote Connectivity

- SCP and FTP allow us to transfer files to a remote destination.
- `scp` is a secure copy protocol which means securely transferring computer files between a local machine **localhost** and a remote machine **remote host** or between two remote hosts. It is a non-interactive program. `s` is for secure or SSH.
- `ftp` or the improved version `sftp` is an interactive file transfer protocol. It allows us to not only copy, but also give commands such as creating and deleting directories and files. Again, `s` is for secure or SSH.

Remote Connectivity

There are GUI programs that allow us to have a GUI view of the directories and files of a remote computer and transfer those files to and from our localhost computer.

The most common are these:

- `Cyberduck` - Mac, Windows
- `WinSCP` - Windows
- `Filezilla` - Linux, Mac, and Windows

Remote Connectivity

These programs can used to access your CIT account (ssh.cs.dixie.edu) and Scratch (scratch.cs.dixie.edu) while at home or outside the CIT Network and transfer files back and forth.

Network Review

- What is a Computer Network? It is a group of computers with permissions to talk to each other. Here on campus we have at least two networks: The DSU network and the CIT network.
- The CIT network consists of all the computers in the SCC building. These are on a unique network which is why they have a unique login name and password. This network includes scratch and our virtual machines.
- The DSU network consists of all other computers on campus including any laptops, phones or devices connected through wifi.

- To this let's add one more network - The rest of the world!

Network Review

- The rest of the world will mean if you are on any other computer or connected through any other wifi or internet connection. Such as when you are at home.
- All networks are set up differently, but we won't discuss that in this class. What you do need to know is that many networks have a firewall. A firewall protects a network from outsiders. A firewall is a server(aka router) set up to check the credentials of everyone trying to gain access to the network.
- Up until now, we have learned that to ssh to `scratch`, we have to ssh to `ssh.cs.dixie.edu` first. `ssh.cs.dixie.edu` is our firewall. To connect to `scratch` using ssh, first we ssh to `ssh.cs.dixie.edu` and then we ssh to `scratch.cs.dixie.edu` or any other machine in the network. Thus connecting to `scratch` through the firewall.
- This protects the CIT network. If we are on The rest of the world network, at home or somewhere else in the world, we must prove our identity through the ssh firewall before we can access the CIT computers. Similarly, if we are on the DSU Network, such as when we use a laptop, we must prove our identity before we can access the CIT network.

Network Review

- We saw evidence of this while completing the GUI install. In order to access the VNCViewer we needed to create a "tunnel" either by checking a box or by using a connection manager such as putty. To access our Virtual Machine, first we had to prove our identity to the firewall.
- The same restrictions apply when making scp or sftp connections.

Network related commands

- `ping` allows us to examine and monitor a network.
  - we can see if our network connection is up
  - we can test if DNS is working
- `traceroute` allows us to see what `hops` our packets will take through the network

---

Network related commands

- `ip a` shows our ip address
- So does `ifconfig` (on older systems, or if you install net-tools)
- `netstat` shows our open port numbers (and other networking info) (net-tools)

Ports

- Computers communicate with each other through ports. When we want to VNC connect to our virtual machine - we access that through a port. Anytime we create an ssh connection, our computer sends out the message through port 22.
- When we create a tunnel it uses port 22 to create the tunnel because it is an SSH connection, but we need to specify
  - which machine and port we want to receive the information on
  - and our destination machine and port number - the machine we are tunneling to.

When selecting Ports to receive information on we can choose most any number between 1024 and 65535. These are the ports not reserved for system use and are available for user use.

Creating a Tunnel

Use the `VNC Connect From Home` Pages in Canvas to see how to create a command line tunnel on your particular machine.

A tunnel connection has the following format:

- ssh <username>@ssh.cs.dixie.edu -L <receivingport>:<destination machine:port>

Creating a Tunnel (Examples)

To create an SSH tunnel to our VM on desdemona with port number 8467 through the ssh.cs.dixie.edu firewall for a VNCViewer connection, it looks like this:

- ssh <cit-username>@ssh.cs.dixie.edu -L 8467:desdemona.cs.dixie.edu:8467

To create an SSH tunnel to Scratch through the ssh.cs.dixie.edu firewall for an SFTP connection, it looks like this:

- ssh <cit-username>@ssh.cs.dixie.edu -L 8467:scratch.cs.dixie.edu:22

Creating a Tunnel (Examples)

Do you see the similarities and the differences? The main differences are the destination machine and the destination machine port number. You must keep that terminal open to keep your tunnel connection. Next you open your VNCViewer or your SFTP/SCP program and use the <receiving machine:port> provided in your tunnel. In the host or server box you type localhost and in the port box you type the port number.

SCP Command Line Connections

SCP - Secure Copy - allows you to securely copy files from one machine to another.

scp works just like cp. The only difference is that for scp we declare the source and destination computer and file

- cp <source> <destination>
- scp <source> <destination>

By default when we use scp to access a remote computer our relative path begins in our $HOME directory. Absolute paths always work and relative paths start in $HOME.

SCP Command Line Connections (Examples)

To copy a file from my `localhost to a remote host` (watch the spacing)

- scp filename destination-machine:./new_filename
- **scp myfile.txt smorgan@scratch.cs.dixie.edu:./myfile.txt**

SCP Command Line Connections (Examples)

To copy a file from a `remote host to my localhost` (watch the spacing)

- scp source-machine:./filename ./new_filename
- **scp smorgan@scratch.cs.dixie.edu:./myfile.txt ./myfile.txt**

Interpret These Commands - Which file(s) are being transferred and where will the file(s) end up?

- scp stats.tar shauna@oxygen.cs.dixie.edu:/tmp/
- scp shauna@oxygen.cs.dixie.edu:/tmp/stats.txt ./
- scp shauna@oxygen.cs.dixie.edu:/tmp/*.txt ./

SSH Basic Connection Shortcuts

- ssh <username>@<machine name or ip>
- ssh joe@scratch.cs.dixie.edu <— inside the network we can go directly to scratch without connecting through the firewall.

- ssh scratch.cs.dixie.edu <— username can be omitted if it is the same as the current user
- ssh joe@scratch <— suffix can be omitted from inside the network
- ssh scratch <— username and suffix can both be omitted if the username is the same and you are inside the network
- ssh ssh.cs.dixie.edu <— from off campus or outside the network you must continue to ssh through the firewall
- ssh ssh <— accessing through the CIT firewall from inside the network

---

Accessing Remote Machines

Keys Watch this

Every time we access a computer for the first time we are asked in computer-ese - Do you trust this computer? Because we are safely inside of our network we always say yes. The computer then remembers our answer and never asks us again.

Inside of our $HOME directory - we have a directory called .ssh. It is a hidden directory because it is not necessary for everyday use.

- ssh into ssh.cs.dixie.edu
- ls .ssh

Accessing Remote Machines

In this directory you should see

- authorized_keys id_rsa id_rsa.pub known_hosts

known_hosts

known_hosts is a list of hosts(computers) that you have connected to previously. When you say ‘yes’ and authorize a connection to a new computer it stores the information in here.

- less .ssh/known_hosts

Accessing Remote Machines

id_rsa and id_rsa.pub

These are the files used to prove your identity to other computers without entering a password.

- id_rsa is your private key
  - less .ssh/id_rsa
- id_rsa.pub is your public key
  - less .ssh/id_rsa.pub

Accessing Remote Machines

authorized_keys

authorized_keys is a list of the public keys of other computers authorized to login without entering a password.

- less .ssh/authorized_keys

Accessing Remote Machines

ssh-keygen

ssh-keygen (no spaces) can generate a public and private key if you don’t already have them. This command does not work on a Mac.

- The command is just `ssh-keygen`. Do NOT generate a new public and private key if you already have one.
  - Check for an existing pair
  - `ls .ssh` to see if they already exist.
  - if you see id_rsa and id_rsa.pub then you already have a generated key pair
  - if they are not there - it is safe to generate new ones
    - if you don't already have a .ssh file you should create one now. `mkdir .ssh`
    - `ssh-keygen`
    - When prompted for the name of the file enter `.ssh/id_rsa`
    - Enter NO passphrase - A passphrase is for extra security and defeats our purpose of not entering a password.
  - Check your work `ls .ssh`

Accessing Remote Machines

ssh-copy-id

Once you have a public/private key pair you can share the public key with another machine to allow yourself to log in without the use of a password.

ssh-copy-id (no spaces) enables you to log into remote machines without entering a password by copying your public key to the other computer. The ssh-copy-id command does all the work of copying the public key to the remote machine and correctly putting the information everywhere it needs to be. It stores this information in the authorized_keys file. This command does not work on a Mac.

Accessing Remote Machines

Then when you ssh to that computer, the authorized_keys file is checked, if there is an entry for your computer, then the private key on your computer talks to the public key on the other computer - and if they are a correct pair then you can connect to the machine without a password.

- ssh-copy-id <username>@<remote machine name or ip address >
- ssh-copy-id smorgan@scratch.cs.dixie.edu
- ssh-copy-id scratch

This will allow you to log into remote machine without entering a password because by sharing your public key it's like saying I'm the same person on both machines - trust me.

Beware

Be very careful to only generate a public/private key pair if it doesn't already exist. You can break your CIT profile and/or your virtual machine access if you generate a new one within your profile. Of course it can be fixed.

Within the CIT network - our access to our Virtual Machines depends on the known_hosts and authorized_keys file. Don't delete these files.

On most computers you can delete both the known_hosts and authorized_keys file without doing any real damage. You simply have to restore known_hosts and authorized_keys that you want to keep.

If these commands are done incorrectly or if you replace an existing rsa_key - it could break your vm.cs.dixie.edu account. If some of your virtual machines start disappearing or if you are not able to turn on and off your virtual machines. Then run the following commands:

- ssh vm.cs.dixie.edu
- /qemu/bin/citv init

Textbook Time

- WES-16 - bottom of page 205-212, wget, ssh, scp and sftp
