

Remote Access: ssh

CIS 2230 Linux System Administration
Lecture 15a
Steve Ruegsegger

Review

- What is the pid and the ppid?
- What has pid #1?
- What does the `$ ps` command do?
What are the 2 argument formats I asked you to memorize?
- What does the `$ top` command do?
- What do these CPU states mean?
`us, sy, ni, id, wa`
- What is nice range? What is the default nice?

Remote connecting

- There are many way to remotely connect to another unix box.
- Why would I want to?
- Protocols:

protocol	purpose	secure version
<code>telnet, rlogin</code>	get a shell prompt	<code>ssh</code>
<code>rcp</code>	copy a file between 2 computers	<code>scp</code>
<code>ftp</code>	File Transfer Protocol	<code>sftp, ftps, vsftp</code>
<code>telnet + export DISPLAY</code>	run X windows remotely	<code>ssh -X</code>
<code>vnc</code>	open a remote X session and display only screen image	<code>vnc + ssh</code>

telnet

- telnet opens a port / connection to a remote system to get a 'regular' shell prompt
- telnet asks for a username and password
- neither are encrypted
 - They are 'clear text' over the internet – bad
- format: `$ telnet remote-sys`

```
—▶ steve@lucid:~$ telnet sassrv1.btv.ibm.com
AIX Secure System : ACE V6.1
(C) Copyrights by IBM and by others 1982, 2004.
login: ruegoegs
ruegoegs's Password:
—▶ ruegoegs@sassrv1:~$
```

The “r-utilities”

- The 'old' way of doing things.
- They work perfectly fine... but inherently insecure.
- Easy to have no password & no encryption
- They have been replaced with _____

Table 7-1 Common r-utilities

Utility name	Description
rwho	Lists the users that are logged in on all hosts attached to the local network that have the r-utilities networking features activated
ruptime	Lists all hosts attached to the local network (that are using r-utilities) with the uptime for each (how long that system has been running since the system was restarted)
rlogin	Logs in to a remote host; very similar to Telnet, but uses the r-utilities authentication methods described in this section
rsh	Executes a command on a remote computer without logging in (a command is executed but you never see a login prompt from the remote computer)
rcp	Copies files between different computers (the local computer and a remote computer or between two remote computers)

rlogin

- Similar to telnet, but doesn't require username and pw if setup correctly
- format:

```
$ rlogin remote_sys
$ rlogin user@remote_sys
```

- The **rhost** protocol has a list of 'known/trusted' users and hosts.
 - ~/.rhosts
- If a trusted user@host requests to connect, then the rhost allows that connection because it's assumed that all is well.
- rlogin is also non-secure, non-encrypted

```
steve@steve-lucid-laptop:~$ rlogin ruegsegs@sassrv1.btv.ibm.com
ruegsegs@sassrv1.btv.ibm.com's password:
Last unsuccessful login: Wed Oct 12 21:53:28 EDT 2011 on ssh from 9.12.229.210
Last login: Wed Oct 12 23:19:43 EDT 2011 on /dev/pts/102 from 9.12.229.210
*****
*                IBM Microelectronics - Burlington, Vermont                *
*                The highest classification allowed in this System is        *
*                IBM Confidential                                             *
*  IBM's internal systems must only be used for conducting IBM's business   *
*                or for purposes authorized by IBM management.              *
*                Use is subject to audit at any time by IBM Management       *
*                                                                           *
*
```

ssh

- ssh ("Secure SHell")
- A protocol for securely accessing one computer from another.
- Despite the name, SSH is more than just a shell.
- It allows you to run command line and graphical programs, transfer files, and even create secure virtual private networks over the Internet.
- <https://help.ubuntu.com/community/SSH>

ssh topics

ssh topics I want you to understand :

- 1) overview
- 2) ssh -X
- 3) ssh -p <non-standard port>
- 4) ssh + command
- 5) scp = ssh + cp
- 6) ssh port forwarding – ssh tunneling
- 7) ssh without password request

1. ssh overview

- Like telnet
 - You connect to another computer as another user
 - ssh does this securely. It uses strong **cryptography** to protect your connection against eavesdropping, hijacking and other attacks.
- Like rlogin:
 - SSH allows you to log in to the server **without having to type a password**.
 - SSH allows you to connect to the server and **automatically send a command**, so that the server will run that command and then disconnect. So you can use it in automated processing.
- ssh is the new standard:
 - The Internet is a hostile environment and security is everybody's responsibility.
 - If you are connecting across the open Internet, then **we recommend you use SSH**.
 - If you are behind a good firewall, it is more likely to be safe to use telnet or rlogin, but we still recommend you use SSH.

Install ssh packages

- `openssh-client` package is installed on default on Ubuntu desktop
- But the server is not
 - `$ sudo apt-get install openssh-server`
 - You could also install with Synaptic
 - The '`ssh`' package contains both `openssh-client` and `openssh-server`
- Note - one beauty of ssh is that it's OS independent. A very popular ssh-client for Windows is "PuTTY". Google it.

using ssh client command

SSH(1)

BSD General Commands Manual

SSH(1)

NAME

ssh – OpenSSH SSH client (remote login program)

SYNOPSIS

```
ssh [-1246AaCfGKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec] [-D  
[bind_address:]port] [-e escape_char] [-F configfile]  
[-i identity_file] [-L [bind_address:]port:host:hostport]  
[-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port] [-R  
[bind_address:]port:host:hostport] [-S ctl_path]  
[-w local_tun[:remote_tun]] [user@]hostname [command]
```

DESCRIPTION

ssh (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP ports can also be forwarded over the secure channel.

ssh connects and logs into the specified hostname (with optional user name). The user must prove his/her identity to the remote machine using one of several methods depending on the protocol version used (see below).

using ssh client command protocol

```
$ ssh steve@server.vtc.edu  
$ ssh ruegsegs@server.ibm.com  
$ ssh rueg@155.42.234.100  
$ ssh user@hostname
```

ssh communications

```
steve@steve-lucid-laptop:~/.ssh$ ssh ruegsegs@gridlnx020.btv.ibm.com
The authenticity of host 'gridlnx020.btv.ibm.com (9.61.38.23)' can't be established.
RSA key fingerprint is d8:51:c9:ed:cf:8f:3a:ee:b8:b2:85:49:8a:48:c5:5f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'gridlnx020.btv.ibm.com' (RSA) to the list of known hosts.
ruegsegs@gridlnx020.btv.ibm.com's password:
Last login: Sat Oct 15 16:25:49 2011 from 9.12.224.70
====> Welcome to the IBM Linux Client for e-business <====

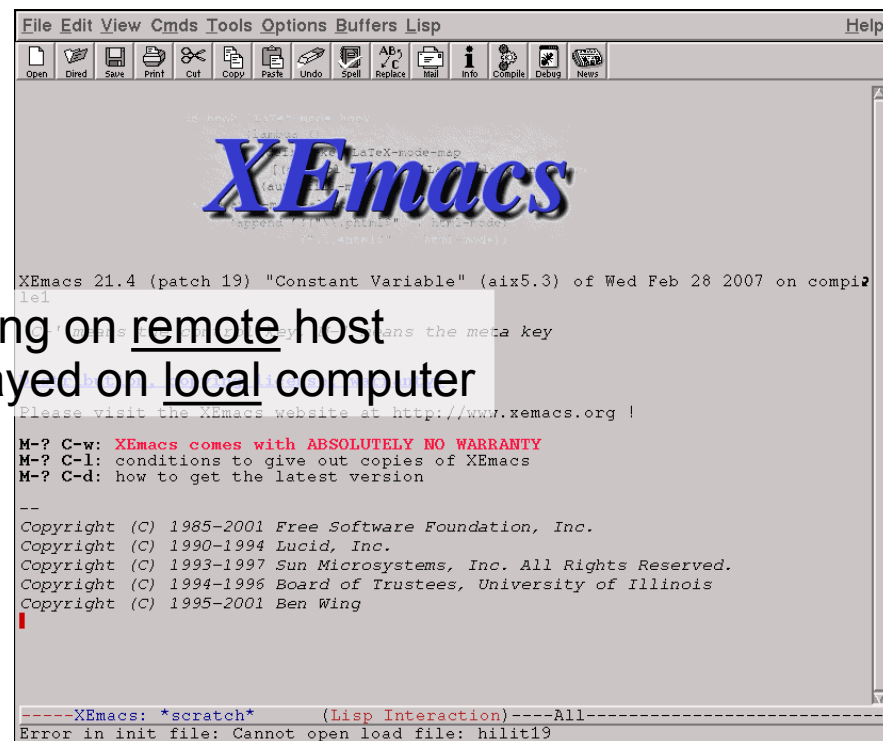
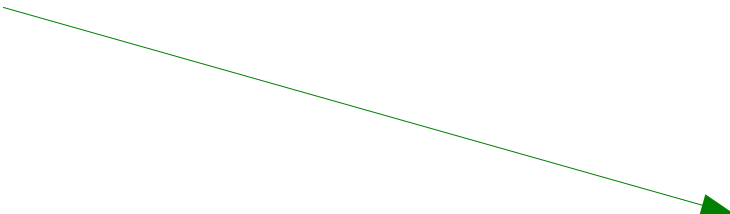
ruegsegs@gridlnx020:/afs/btv.ibm.com/u10/ruegsegs
$
```

2. Run X apps securely through ssh

- The -X option allows X commands to be displayed **thru** the ssh connection on your local host from the server

```
steve@lucid:~$ ssh -X ruegsegs@sassrv1.btv.ibm.com
ruegsegs@sassrv1.btv.ibm.com's password:
```

```
ruegsegs@sassrv1:/afs/btv.ibm.com/u10/ruegsegs
$ xemacs
```



Running on remote host
Displayed on local computer

N.B. this replaces what we "used to do":

- 1) telnet to server
- 2) lookup local IP
- 3) export DISPLAY=local_IP:0
- 4) xhost +server_IP on local box

2. Run X apps securely through ssh

Questions for you

- What is this “X”?
- Does your MS Windows laptop have “X”?
- Can you run “X apps” on your windows box?
- When is ssh -X helpful?

3. ssh default port

- You can connect using a different 'port'
- The default ssh port is **22**
- Sometimes, to be more secure, or for port-forwarding reasons, we move it.
- **-p defines a non-standard port**
- `$ ssh -p 2222 steve@server.cis.vtc.edu`

4. ssh + command

- One can also send a command with ssh.
- The one command gets executed, then the ssh session is closed.
- This can be used for automated tasks and setup in a cron.

user@host

command

```
steve@lucid:~$ ssh ruegsegs@gridlnx020.btv.ibm.com 'find . -name "lab*" '
ruegsegs@gridlnx020.btv.ibm.com's password:
./lab3.txt
./lab.02.txt
./lab.02.bak
./lab2
./lab2/lab3.txt
./lab2/lab.02.txt
./lab2/lab4.txt
./lab3
./lab3/lab03.tar
./lab5.txt
./lab4.txt
```

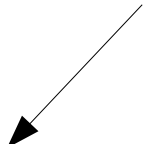
5. scp

- secure copy - uses the ssh protocol
- Ubuntu doesn't even support the 'original' remote-cp: `rcp`
 - `$ man rcp --> ?`
- The new, preferred method to copy securely to a remote system is `scp`
- Format for remote files:
`user@host:files`
- That format can a source or destination or both
 - `$ scp <source> <destination>`
 - `$ scp user@host:files local_file`
 - `$ scp local_file user@host:files`
 - `$ scp user@host:files user@host:files`
- `scp` operates just like `cp`:
 - `-r` is recursion
 - Sources and destinations can be files, dirs or globs

scp example

- Copy a local source (dir, file, glob) to a remote computer

What is '.' and how does
scp interpret this?



```
$ scp -r ~/lab03 ruego@gridlnx020.btv.ibm.com:.  
ruego@gridlnx020.btv.ibm.com's password:  
Lab03_assignment.odt          100%   32KB   32.3KB/s   00:00  
Lab03_assignment.pdf          100%  112KB  112.5KB/s   00:00  
Lab03_report.odt              100%   87KB   87.2KB/s   00:00  
Lab03_report.pdf              100%  160KB  159.6KB/s   00:00  
lab03.tar                     100%  540KB  540.0KB/s   00:00
```

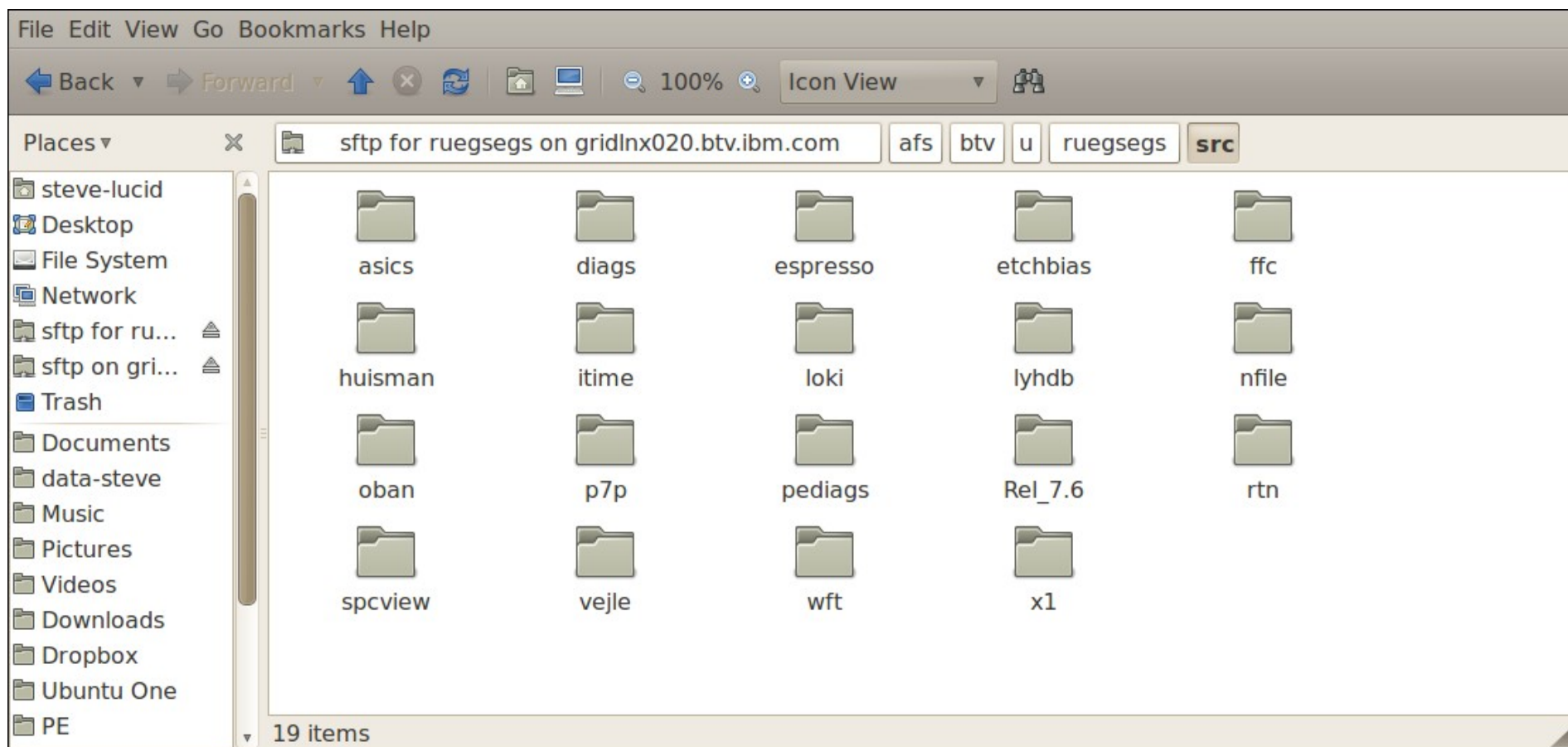
ftp / sftp

- ftp - File Transfer Protocol
- A simple protocol for coping files only
 - commands include get, put, ls, !ls
- Popular for anonymous file xfer and dropboxes
- FTP is not secure
- sftp, part of ssh, has replaced it most everywhere

```
steve@steve-lucid-laptop:~$ ftp gridlnx020.btv.ibm.com
ftp: connect: Connection refused
ftp> ^C
steve@steve-lucid-laptop:~$ sftp gridlnx020.btv.ibm.com
Connecting to gridlnx020.btv.ibm.com...
ruegsegs@gridlnx020.btv.ibm.com's password:
sftp>
```

ftp in Ubuntu Nautilus (file browser)

- `^L`
- `sftp://username@server/dir`



ftps, vsftp

- ftps is a secure ftp protocol
- vsftp stands for "very secure ftp"
- If you need secure FTP, then you can research these more.