# Introduction to Linux Workshop 2

## The George Washington University SEAS Computing Facility

Created by Jason Hurlburt (hurlburj@gwu.edu), Hadi Mohammadi (hadimoh@gwu.edu), Marco Suarez (msuarez@gwu.edu)

# Logging In

The lab computers will authenticate with your NetID and password. Please log into the computer you are sitting at.

If you are watching online please go to:

https://www.tutorialspoint.com/unix_terminal_online.php

# Course Goals

- Permissions
- SSH and communicating with other machines
- SFTP, SCP, and file transfer
- Porting X11 session to your local machine

# Permissions

- Format
- View Permissions
- Set Permissions

# Permissions - Format

**drwxrwxrwx**

10 characters of information

- First character: tells you whether the listing is a file (-) or a directory (d)

Subsequent characters: either r (read), w (write), x (execute), or - (no permission)

- Characters 2-4: tells you the read (r), write (w), and execute (x) permissions for the owner
- Characters 5-7: tells you the read (r), write (w), and execute (x) permissions for the group
- Characters 8-10: tells you the read (r), write (w), and execute (x) permissions for everyone

# Permissions - Format
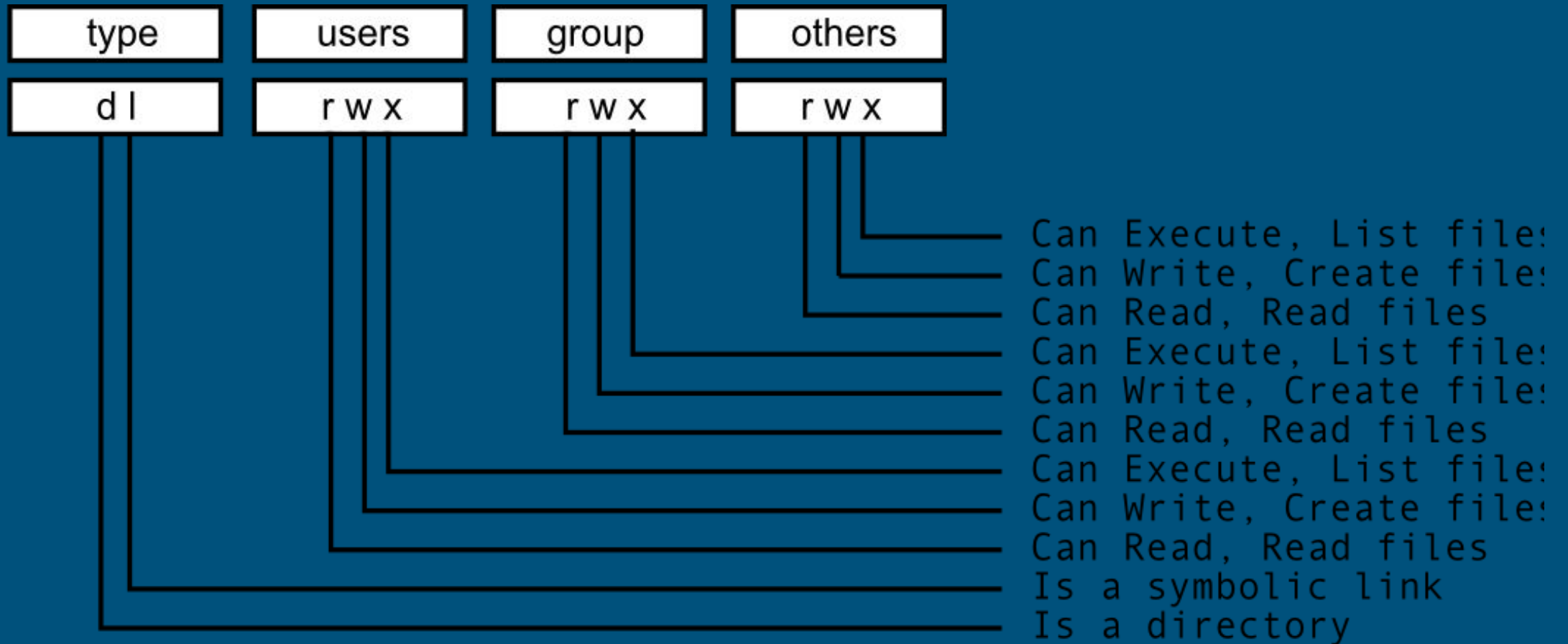
# Permissions - View Permissions

| d | rwx | rwx | rwx |
|---|---|---|---|
| Directory or File | Owner | Group | Other (Everyone) |

- Example ls -la:

  drwxr-xr-t   2 hurlburj user   2048 Sep 30 10:54 thinclient_drives

  drwxr-xr-x   2 hurlburj user   2048 Jan 26 10:57 training_dir

  -rwxr-x---   1 root     bin     175 Sep 22 21:56 usernames.txt

  -rw-------   1 hurlburj user      44 Sep 22 19:22 .vas_disauthcc_478615

# Permissions - Set Permissions

Modify file permissions using the "chmod" command

Syntax: chmod [u | g | o | a] [+ | -] [r | w| x] arg

- u = user, g = group, o = other, a = all
- + will add the permission, - will remove it
- r = read, w = write, x = execute
- arg is a filename, a directory name, or a list of either or both

For more details, type "man chmod"

Example: *chmod u+x file*

- Adds execute permission for the current user on the file named "file"

# Exercise - Create a file and set permissions

- Make a directory in your home directory
- Set permissions so everyone can read the directory
- Create a file inside the directory
- Set permissions so everyone can read the file
- Go to the directory and enter "pwd"

# SSH - What is it?

- SSH, also known as Secure Socket Shell, is a network protocol that provides administrators with a secure way to access a remote computer.
- Allows users to securely log into another computer over an <u>insecure network</u>, executes commands and transfers files
- Created as a replacement for telnet, ftp, rlogin, rsh, and rcp
- SSH traffic is always encrypted, which makes it safer
- Usually communicates on port 22, but it can use any port you specify

# SSH – SSH Components

## SSH Always has a Client and a Host

- Client - the machine you are sitting in front of
  - A client needs a program, called an SSH client, to connect to another machine via SSH.

- Host - the server you want to connect to
  - A host is any machine listening for SSH connections.
  - It runs a program called SSHD that waits for SSH connections.
  - When your client sends a request to connect, SSHD handles the negotiation that lets you log in.

# SSH - Authentication

SSH has two common methods of authentication:

- Password authentication
  - Not as secure as key authentication
  - Easier to set up
- Key authentication (Advanced)
  - Vastly more secure than password authentication
  - Difficult to set up
  - Easier to use once set up

# SSH - Features

SSH is available on most platforms

- Clients are available for many platforms: Windows, Mac, Unix, Linux, iPhone, Android, etc.

Free for noncommercial use

SSH can multiplex services over the same connection

- That means you can open an SSH connection and send other kinds of traffic through it
- One of the most powerful functions of multiplexing is port forwarding or tunneling
- SSH can securely tunnel insecure applications like POP3, SMTP, IMAP, and CVS.

# SSH - Connect to a remote host

We are going to use shell.seas.gwu.edu for our remote host in this class

To connect:

    *ssh [username@shell.seas.gwu.edu](mailto:username@shell.seas.gwu.edu)*

    *enter your account password*

To disconnect:

    *type "exit" at the prompt*

You can daisy chain connections from one host to another

    *ssh [username@shell01.seas.gwu.edu](mailto:username@shell01.seas.gwu.edu) (from shell.seas.gwu.edu)*

# SSH - Useful commands when connected

You can sometimes forget which machine you are connected to.

Tips to find out where you are:

- look at the prompt
- type *hostname*
- type *uname -a*

# SSH - Remote Terminal Exercise

A big advantage of SSH is the ability to remotely control a machine without having physical access to it.

Exercise:

Make sure you are at your lab machine prompt.

- From your terminal, type */bin/uname -a*
  - What server name pops up?
- Type *ssh [username@shell.seas.gwu.edu](username@shell.seas.gwu.edu) '/bin/uname -a'*
  - Enter your password and answer yes to any questions
  - What server name pops up?

# SSH - SFTP

The SSH File Transfer Protocol (SFTP) enables secure file transfer capabilities between networked hosts.

- All SFTP traffic is encrypted
- FTP (File Transfer Protocol) is insecure.
- SFTP, like SSH, requires a client and a host (server).
- You authenticate with the same credentials you use for SSH

# SSH - SFTP command line client

SFTP can be used at the command line on linux machines and some Apple machines. You can launch it by typing:

       *sftp [username@shell.seas.gwu.edu](mailto:username@shell.seas.gwu.edu)*

- Use ? for help
- Regular commands occur on the remote server. **ls, pwd, cd**, etc.
- To work on your local machine, prepend  the letter l to the commands: **lpwd, lls, lcd**, etc.
- Use "**put**" to transfer files TO the server
- Use "**get**" to transfer files FROM the server

# SSH - SFTP GUI Clients

Using the command line SFTP client is confusing, frustrating, and inconvenient. There are many free SFTP utilities for Windows, Apple, and Linux machines.

GUI clients are easy to use and necessary if you will transfer a large amount of folders or files between machines.

- Windows
  - Filezilla - https://filezilla-project.org
- Apple
  - Cyberduck - https://cyberduck.io
- Linux
  - Filezilla - https://filezilla-project.org

# SSH - Secure Copy (SCP)

The scp command securely copies files and directories between remote hosts without starting an FTP session or logging into the remote systems explicitly. The scp command uses SSH to transfer data, so it requires a password or passphrase for authentication.

Usage:

- *scp **Source Destination***
- *scp user@host1:file1 user@host2:file2*
- host1 or host2 can be your local workstation or the server
- You do not need to include the user@host* section when referring to your local workstation. Just the path is OK

# SSH - SCP Examples

SCP is good for moving one or two files quickly.

Move a file FROM a server TO your workstation:
- *scp username@shell:/tmp/class_example.txt /homedirectory/*

Move a file TO a server FROM your workstation:

- *scp /homedirectory/class_example.txt username@shell.seas.gwu.edu:/tmp/**username.txt***

In the second example, we are renaming the file so the whole class doesn't try to write the same file name at the same time.

# SSH - SCP Examples

SCP is good for moving one or two files quickly.

Move a file FROM a server TO your workstation:
- *scp username@shell:/tmp/class_example.txt ~*

Move a file TO a server FROM your workstation:

- *scp ~/class_example.txt* [*username@shell.seas.gwu.edu*](username@shell.seas.gwu.edu)*:/tmp/**username.txt***

In the second example, we are renaming the file so the whole class doesn't try to write the same file name at the same time.

# SSH - X Windows

X or XWindows was designed from the beginning to be network-centric, and adopts a "client-server" model. It provides a Graphical User Interface for Linux applications.

- An X "client" is usually an application written with X libraries, like Matlab or your Window Manager that makes up your desktop
- The X Server draws what you see on the screen and manages the keyboard and mouse.
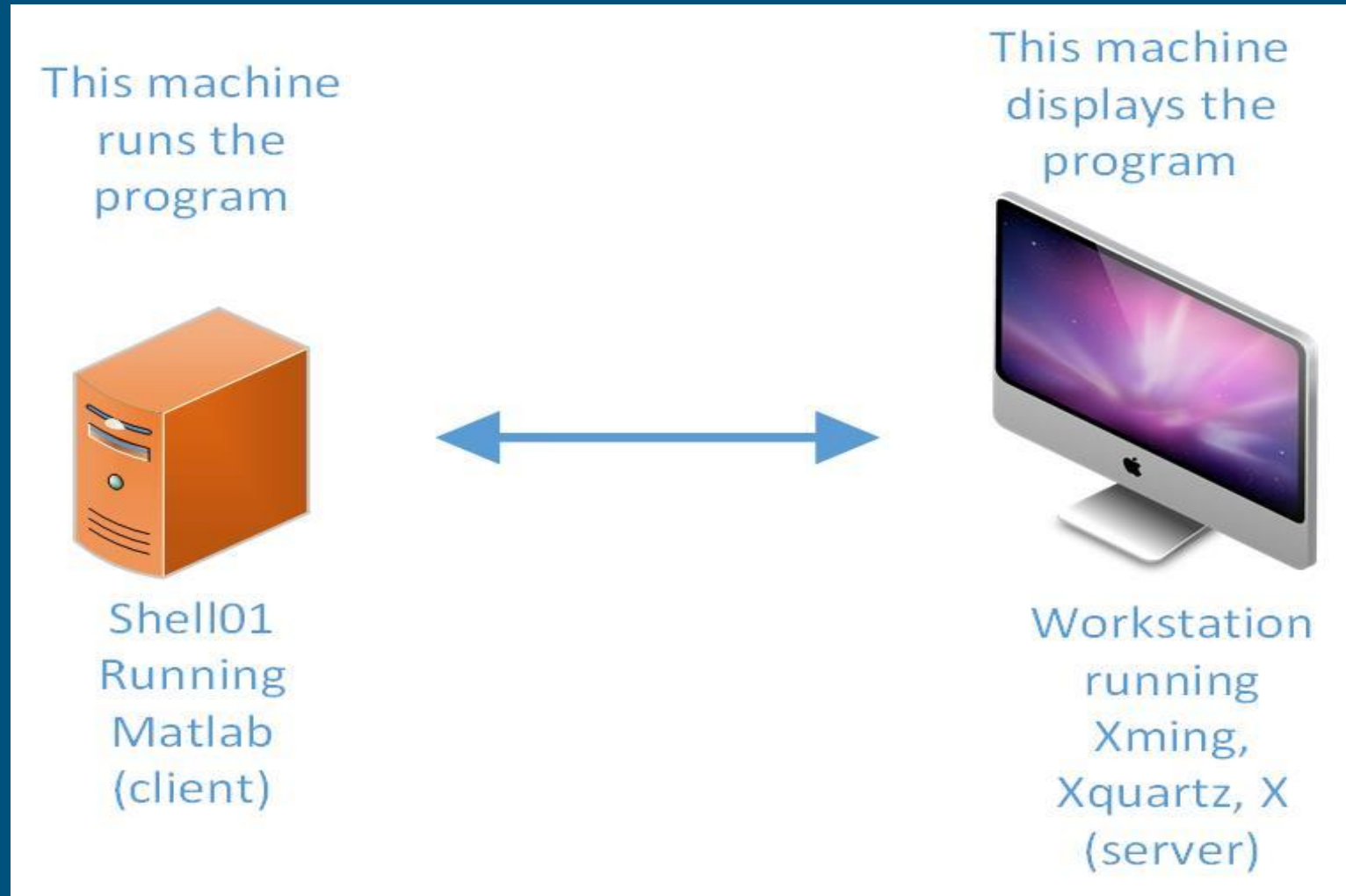- When you run an application remotely, your workstation is running the X Server.

# SSH - X Windows

X or XWindows was designed from the beginning to be network-centric, and adopts a "client-server" model. It provides a Graphical User Interface for Linux applications.

- An X "client" is usually an application written with X libraries, like Matlab or your Window Manager that makes up your desktop
- The X Server draws what you see on the screen and manages the keyboard and mouse.
- When you run an application remotely, your workstation is running the X Server.

# SSH - X Windows Architecture

This machine runs the program

This machine displays the program

Shell01 Running Matlab (client)

Workstation running Xming, Xquartz, X (server)

# SSH - X Windows Forwarding Requirements

- X Window Forwarding refers to sending the display of a program from one machine to another.
- Forwarding requires a small program to run on machines without a native X Server if you want to port the display to your workstation.
- Windows and OS X do not have a native X Server built in
- Most linux builds have an X Server built in

# SSH - X Windows Server programs

Windows
- MobaxTerm (Easier) - http://mobaxterm.mobatek.net/

Apple OS X
- XQuartz - https://www.xquartz.org/
- XQaurtz may require you to install the Apple OS X SDK from the Apple Store (free download)

Linux
- An X Server is usually installed with the OS

# SSH - X Window Forwarding Connection

You need to tell your SSH session that you want to forward X traffic from the machine you are connecting to.

Windows
- Your SSH client will have a checkbox somewhere that says "Forward X traffic"
- In Putty it is under the Connection/SSH/X11 menu
- If you use Putty, you have to launch XMing manually

Apple OS X
- *ssh -X username@server* (this will automatically launch XQuartz)

Linux
- *ssh -X username@server*

# SSH - X Window Forwarding Exercise

Use your linux workstation to start an X connection to shell.seas.gwu.edu

From the terminal:
1. ssh -X username@shell01.seas.gwu.edu
2. type *xeyes*
3. Close *xeyes*
4. type *xclock*
5. close *xclock*
6. type *xterm*
7. In the xterm window, type *hostname*
8. close *xterm*

# Useful Links

Online Linux Lab:
https://shell.seas.gwu.edu:3443

Request VMWare Workstation License:
https://seascf.seas.gwu.edu/vmware-subscription

https://vpn.gwu.edu

# Questions?