# Software Installation

CS4239 - Software Security

School of Computing
National University of Singapore

Aug 2022

# Students with M1/M2 Macs / other VMs

These slides are not geared for students using non-X86 hardware

However you may want to look at the slides for other points

# Software Installation

- Download Ubuntu disk image file
  - SG mirror: https://mirror.0x.sg/ubuntu-cd/20.04.4/
    Slower: http://releases.ubuntu.com/focal/
  - Download `ubuntu-20.04.4-desktop-amd64.iso` for 64-bit machines
  - we are using a slightly older version of Ubuntu and Linux kernel, however, this version  has end of life beyond what we need (its also to be compatible with any students using the M1 UTM pre-built VM)
    - this is OK as your install is meant to be only for our usage rather than general usage

# X86 Virtualization

- these slides use Virtual Box however other Virtualization solutions  for X86 is also feasible
  - Note that you should be using some X86 virtualization
  - Alternatively you can install native but that has disadvantages because we want to keep a fixed Linux setup
  - Other X86 virtualization solutions are: Parallels (MacOS), Vmware Player, etc

# VirtualBox Installation

- **In case not already installed / want to install on your own machine**
- VirtualBox download & install:
  https://www.virtualbox.org/wiki/Downloads
- After VirtualBox installation is done, install VirtualBox extension. This is needed for accessing USB device and host OS file system.
- Go to: https://www.virtualbox.org/wiki/Downloads
- Select VirtualBox Oracle VM VirtualBox Extension Pack for all supported platforms, save it
- In VirtualBox, click File -> Preferences -> Extensions -> (small plus) -> select  and install extension pack

# Software Installation

- Start VirtualBox
  - Setup location for VDI images
  - Click File -> Preferences -> General
  - Choose/create a Default Machine Folder
- Click “New”
  - Name the virtual machine, e.g. “Ubuntu 20.04” (if you are using a Lab PC, then you will want to name the VM with your student ID)
  - Select 4096 MB RAM (other sizes are possible, depending on your RAM)
  - Select “Create virtual hard disk now” -> “VDI (VirtualBox Disk Image)” -> “Dynamically allocated”
  - Select the disk size, e.g. 10 GB (the virtual disk will expand as needed)

# Software Installation

- Back to VirtualBox main window, select e.g. “Ubuntu 20.04” (whatever you named) and click “Start”
- When asked to select start-up-disk, **add** the downloaded iso
- Click “Start”
- Ubuntu installation started
- In “Welcome”
  - Click “Install Ubuntu”

# Software Installation

- In “Preparing to install Ubuntu” – Normal Install (Minimal should be usable also, depending on your disk space)
  - Unselect “Download updates while installing”
  - Unselect “Install third-party software”
- In “Installation type”
  - Only select “Erase disk and install Ubuntu” (erases virtual drive)
- In “Where are you?”
  - Enter “Singapore”
- In “Keyboard layout”
  - Select “English (US)” and “English (US)”

# Software Installation

- In “Who are you?” (this is just sample – you may use different choices)
  - Your name: cs4239 (if using lab PC, add your name/NUSid)
  - Your computer’s name: cs4239**-VirtualBox**
  - Pick a username: cs4239
  - Choose a password:*** (**if using lab PCs, choose a better password**)
  - Confirm your password: ***
  - Select “Log in automatically” – ok as we are using a VM, not recommended for real usage. For lab PCs, choose login with password.
- In “Restart”
  - Confirm restart
  - Press Enter to remove installation media

# Generic Setup

- You have installed a generic Ubuntu Linux
  - user: cs4239 (or whatever you used) passwd:***
  - Note that you can use whatever username you wish (but note if you are using lab PCs)

# Using Linux

- Run the terminal program
  - Click on the Applications button (bottom-left of screen)
  - Type “terminal”, and click on the Terminal app icon
- In the terminal window, you can execute shell command
- Assumption is that you can use basic Linux
  - Help for Ubuntu is easy to find on the web – just search

# Administrative Mode

- Usually in Linux – never run as root (privileged user)
- To run commands needing privileges
  - `sudo` command
  - e.g. `mount` command needs root privileges
    - `$ sudo mount ...`
    - ... are the other options to the command
    - root shell – only if you know what you are doing!

      `$ sudo bash`

# A Note on Apt

- Apt is the main Ubuntu utility to install software
- Sometimes, when typing a shell command, you find some software is missing, for example:

```
$ gcc
gcc: command not found
```

- In this case we use apt to install the missing software:

```
$ sudo apt install gcc
```

  - now **gcc** C compiler should be installed
  - also install **make** (using **sudo apt** as above)

# Working Files

- Files are stored inside the virtual disk
- where to store working files for use outside VM?
  - various options: can mount USB drive / Windows directory / SSHFS
  - you may still want to investigate file sharing with your own machine – **Option 3** is the most likely version to use on your own machine
    - this is likely the mode for this semester

# Storing Working Files: Option 1

- **Mounting from USB drive**
- VirtualBox can see existing USB drives in Windows
- Mount from the VirtualBox menu: Devices -> USB -> Select the USB device you want to access from Ubuntu Linux
  - In Ubuntu try: see what devices are mounted in the file system (note they are also separate logical filesystems)

    $ mount
  - location by default in /media/...
  - note: USB drive is normally FAT filesystem, not native Unix so some things dont work, e.g. Unix file permissions
- unmount after use
  - `$ sudo umount` *dir*
  - *dir* from above mount

# Storing Working Files: Option 2

- **Sharing using sshfs** (this may not be suitable for your laptop, **it is meant more for lab PCs** to share with the sunfire Solaris server, probably you don't need this)
- Install sshfs
  - `$ sudo apt install sshfs`
- Mount your sunfire home directory (or any other UNIX home directory): this is where you will save your files
  - Create an empty work directory, e.g., /home/student/workdir
  - Mount your remote home directory, and point your current directory to it, for example, the following will mount `user`'s home folder on sunfire server to `workdir` in Ubuntu (`unmount` after use)

    ```
    $ sshfs user@sunfire.comp.nus.edu.sg:workdir
    ```

# Storing Working Files: Option 3

- **Mounting Windows folder (or host OS directory)**
- VirtualBox can share folders from/to the host
  - If you are using a lab PC then best to use an external drive from the host OS
- Install VirtualBox Guest Additions
  - You will need to have “make” and “perl” installed (use “apt install” as before). You also need gcc from previous slides.
  - In the VM Window: Devices -> Insert Guest Additions CD Image
  - In Ubuntu Linux: Click “Run” -> Enter Password -> Restart the OS (click on triangle icon at the top-right corner -> Power Off … -> Restart)

# Storing Working Files: Option 3 (Continued)

- Create a folder in Windows (or your host OS) you want to share with Ubuntu
- Create shared folder from VirtualBox menu:
  - create a directory
    - **`$ sudo mkdir /mnt/Shared`**
  - VB menu: Devices -> Shared Folders -> Shared Folder Settings …
    - Click on “+” folder icon
    - Enter shared folder path
    - You can select permanent
    - **unmount** when finished: **`$ sudo umount /mnt/workdir`**

# Storing Working Files: Option 3 (Continued)

- Manual mounting
  - mount folder: *let S be the shared folder*
    - **`$ sudo mount -t vboxsf S /mnt/workdir`**
    - **unmount** when finished: **`$ sudo umount /mnt/workdir`**
      - `not needed if you just save the VM state`

# File Sharing Note

- files in Windows (host OS) can appear in Linux as normal files
- but
  - file semantics can be slightly different, i.e. Windows not same as Linux
  - Unix file permissions not applicable
  - some Linux programs may not work the same mainly because no execute permissions with Windows files
- This link gives more details on the shared folder instructions, you can also mount USB drives
  - https://linuxhint.com/virtualbox_shared_folders

# Start Working

- Again, run the terminal program
  - Click on the Applications button (top-left of screen)
  - Type “terminal”, and click on the Terminal app icon
- Edit `hello.c` (your own hello world C program)
  - many editors to use: e.g.

    http://www.informit.com/articles/article.aspx?p=1670957&seqNum=3

    programmers usually use: vim/gvim, emacs (they can be installed as before with “apt”)
- Save hello.c in your working files folder
- compile & run

```
$ gcc -o hello hello.c
$ ./hello
```

# Saving State

VirtualBox can save the VM state

- Just close the VM window and it will ask if you want to save state
- After saving, you can use the VirtualBox manager to continue a saved state