

# Lab. Session 0

Computer Security Lab.

Name: Sangyun Kim

Email: [sangyun.kim@snu.ac.kr](mailto:sangyun.kim@snu.ac.kr)

# Contents

- **Setup: SSH to Server**
  - #1. Use **Terminal**
  - #2. Use **VS Code**
  - #3. Change Password
- **Linux Basics**
- **Setup: Docker (Optional)**
  - #1. **Install** Docker in **your computer**
  - #2. **Download** Dockerfile
  - #3. **Build** the Dockerfile image
  - #4. **Create** and **Run** the Docker Container
  - #5. **Access** the Container
  - #6. **Stop** and **Remove** the Container

# Setup: SSH to Server – Terminal

- **#1. Open Terminal**
  - e.g., PowerShell, Bash, and etc.

- **#2. Type the SSH command**

```
$ ssh -p <port> compsec@kayle.snu.ac.kr
```


- <port>: port number
    - please check your **port number** in eTL
- **#3. Press Enter and type your password**
  - please check your **initial password** in eTL

# Setup: SSH to Server – VS Code (Recommend)

- **#1. Install VS Code**

- Click [here](#) to download

- **#2. Install the Remote - SSH Extension**

- Go to the **Extensions** view by clicking on the icon () or typing Ctrl+Shift+X
- Search for **Remote -SSH** and click **install**

- **#3. Open the Command Palette**

- Open the Command Palette by typing Ctrl+Shift+P

# Setup: SSH to Server – VS Code (Recommend)

- **#4. Connect to Host**

- In the Command Palette, type **Remote-SSH: Connect to Host...** and select it
- Enter the following SSH command:

```
ssh compsec@kalye.snu.ac.kr -p [port]
```

- Replace [port] with your own port number


- **#5. Configure SSH Configurations (Optional)**

- In the Command Palette, type **Remote-SSH: Open SSH Configuration File...** and select it
  - Windows: C:\Users\<username>\.ssh\config
  - Mac/Linux: ~/.ssh/config

- **#6. Enter your Password**

# Linux Basics – Bash & Home Directory

- Bash Shell
  - Default user interface
  - Every Linux user has its own home directory
- Home directory
  - “~” : an user’s home directory
    - “/home/compsec” = “~”

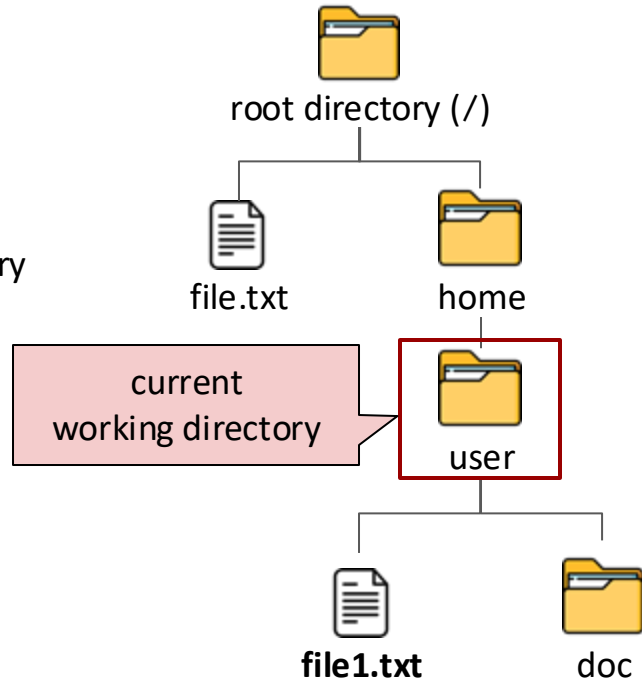


```
compsec@63a0b7745861:~$
```

# Linux Basics – Path

- Path

- Two ways of representing location
- Absolute Path
  - points to location
    - regardless of the current working directory
  - include the root directory (/)
  - example) /home/user/file1.txt
- Relative Path
  - path starts from current working directory
  - . means current working directory
  - .. means parent directory
  - example) ./file1.txt or ~/file1.txt



# Linux Basics – Command List

Command	Description
<code>pwd</code>	print current working directory
<code>ls</code>	list current location
<code>touch [new filename]</code>	make new empty file
<code>mkdir [new directory name]</code>	make new directory
<code>cd [location]</code>	move to another location
<code>cp [source] [destination]</code>	copy source file to destination
<code>cp -r [source] [destination]</code>	copy directories recursively
<code>mv [source] [destination]</code>	move source file to destination
<code>rm [filename]</code>	remove file
<code>rm -r [directory name]</code>	remove directories and their contents recursively



# Linux Commands in Bash Shell

- **mkdir** (make directory)
- **cd** (change directory)
- **ls** (list)
- **rm** (remove)

```
user@XXX:~ $ mkdir temp
```

# Linux Commands in Bash Shell

- mkdir (make directory)
- cd (change directory)
- ls (list)
- rm (remove)

```
user@XXX:~ $ cd temp
user@XXX:~/temp $ pwd
/home/user/temp
user@XXX:~/temp $ cd ~
user@XXX:~ $ pwd
/home/user
```

# Linux Commands in Bash Shell

- mkdir (make directory)
- cd (change directory)
- ls (list)
- rm (remove)

```
user@XXX:~/temp $ cd ..  
user@XXX:~ $ ls  
temp  
user@XXX:~ $ ls -l  
drwxrwxr-x 3 vagrant vagrant 4096  
Jun 1 00:00 temp
```

# Linux Commands in Bash Shell

- mkdir (make directory)
- cd (change directory)
- ls (list)
- rm (remove)

```
user@XXX:~ $ echo > file
user@XXX:~ $ ls
file temp
user@XXX:~ $ rm file
user@XXX:~ $ ls
temp
user@XXX:~ $ rm -r temp
user@XXX:~ $ ls
```

# Setup: Docker (Optional)

- **#1. Install Docker**

- Windows: [Link](#)
- Mac: [Link](#)
- Linux: [Link](#)
- Type the command to confirm if docker is installed

```
$ docker -v
```

- **#2. Download the Dockerfile**

- Create a new directory on your local machine

```
$ mkdir class-compsec  
$ cd class-compsec
```

- Download the [Dockerfile](#) into this directory

# Setup: Docker (Optional)

- #3. **Build** the Docker Image

- In the directory which contains the Dockerfile, run the following command

```
$ docker build -t class-compsec-image .
```

- Once your build is complete, please check if the image is ready

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
class-compsec-image	latest	2f30fc2c1a43	About a minute ago	802MB

# Setup: Docker (Optional)

- #4. **Create** and **Run** the Docker Container

- Once the image is built, now you can create and run a container

```
$ docker run -d -name class-compsec-container --init --cpus="4" \
  --memory="8g" -p 22222:22 --security-opt seccomp=unconfined \
  --privileged class-compsec-image
```

- To check that your container is running, run the command (*please check port number*)

```
$ docker ps -a
```

CONTAINER ID	...	PORTS	NAMES
4c93ea1deb49	...	0.0.0.0:22222->22/tcp	class-compsec-container

## Setup: Docker (Optional)

- #5. **Access** the Container

```
$ ssh -p 22222 compsec@localhost
```

- #6. **Stopping** and **Removing** the Container

- To stop the container

```
$ docker stop class-compsec-container
```

- To remove the container after stopping it

```
$ docker rm class-compsec-container
```