Mini-Lab 0: Warm up! Welcome to Linux!

COMP3230, The University of Hong Kong

Sept. 2023

Total 0 point

Objective

At the end of this lab, you will be able to:

- Set up working environments on `workbench2`.
- Feel comfortable with Linux commands.

Tasks

The first mini-lab is designed to guide you to set up working environments on our grading server (`workbench2`) and familiarize you with Linux commands. We will have three Tasks to warm you up in the following OS learning.

1. **Task 1 (Setup)**: Establish a remote connection to `workbench2` server and download tutorial and lab materials using Git.
2. **Task 2 (Know some Commands)**: Practice several basic Linux commands and get yourself comfortable with the environment.
3. **Task 3 (Practice as a Linux Admin!)**: Role-play a Linux Server Administrator. Your mission is to detect potential threats by identifying the suspicious IPs from log file.

Instructions

Task 1: Setting Up Remote Connection

Unlike Windows and macOS, Linux usually comes with no GUI (Graphic User Interface)[1] but it doesn't mean they are outdated. In fact, on most high-performance servers, Linux is the most used OS for many reasons, including high scalability and stability. If you don't have a computer on hand with Linux installed, no worries, this lab will guide you on how to connect and use a Linux system remotely, *i.e.*, via security shell (SSH).

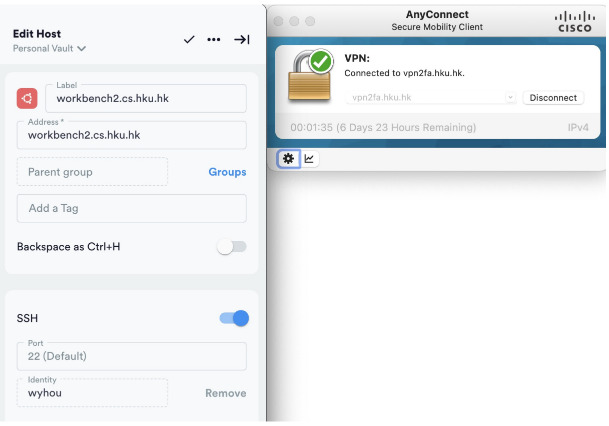
[1] There are some famous GUI for Linux, e.g. xfce and gnome, but it's not a necessity.

Login Linux server remotely

In COMP3230, we will use `workbench2` for grading, and it is a Ubuntu system, one distribution of Linux. To access that server, we have to setup remote SSH connection. There are two recommended approaches: Termius and VS Code.

Termius: Termius is a cross-platform SSH client and provides SCP (copy bulk files to remote) with GUI.

First, we should connect to HKUVPN following the instructions[2] provided by HKU ITS. Now that you can start an SSH session to `workbench2.cs.hku.hk` by adding a `Host` in Termius. See Fig. 1 for guidance.



Edit Host
Personal Vault

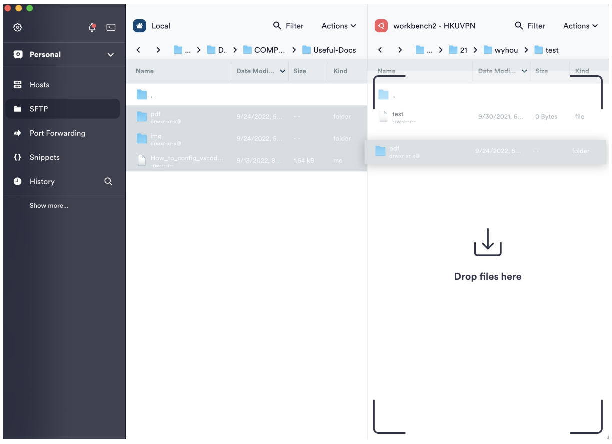
Label: workbench2.cs.hku.hk
Address *: workbench2.cs.hku.hk
Parent group
Add a Tag
Backspace as Ctrl+H
SSH
Port: 22 (Default)
Identity: wyhou

AnyConnect
Secure Mobility Client

VPN:
Connected to vpn2fa.hku.hk.
vpn2fa.hku.hk
00:01:35 (6 Days 23 Hours Remaining) IPv4

Figure 1: **(R)Step 1.** Connect to HKUVPN. **(L)Step 2.** Create a Host. Label: "workbench2". Address: "`workbench2.cs.hku.hk`". Username: <your-cs-account-name>. Password: <cs-account-password>.

Termius allows you to access remote servers from any device, including your phone! It offers both terminal access and a drag-and-drop feature for uploading or downloading files via SFTP. See Figure 2 for reference.



Local: ... > D.. > COMP... > Useful-Docs

Name	Date Modi...	Size	Kind
..			
pdf drwxr-xr-x@	9/24/2022, 5...	- -	folder
img drwxr-xr-x@	9/24/2022, 5...	- -	folder
How_to_config_vscod... -rw-r--r--	9/13/2022, 8...	1.54 kB	md

workbench2 - HKUVPN: ... > 21 > wyhou > test

Name	Date Modi...	Size	Kind
..			
test -rw-r--r--	9/30/2021, 6...	0 Bytes	file
pdf drwxr-xr-x@	9/24/2022, 5...	- -	folder

Figure 2: **Step 1.** Clip SFTP on the side bar. **Step 2.** Select Host to connect. **Step 3.** Drag-and-drop files to download or updoad.

VS Code: Visual Studio Code (VS Code) is a powerful and open-source text editor, and it's an

[2] https://its.hku.hk/services/network-connectivity/hkuvpn/

excellent choice to connect VS Code to the remote server and edit your code on it.

Here is how you set it up:

1. Install and open VS Code on your local machine
2. Install the **Remote - SSH** extension from VS Code marketplace on the left.
3. Press `F1` or `Ctrl` + `Shift` + `P` to toggle **Command Palette** and go to **Remote - SSH: Connect Current Window to Host...** > **Configure SSH Hosts** > **~/.ssh/config** (where you store your SSH configuration).
4. Add the following configuration to your SSH config file, replace <your-cs-account> with your own account.

```
Host workbench2
    HostName workbench2.cs.hku.hk
    User <your-cs-account>
    Port 22
```

5. Save and close the file.
6. Press `F1` or `Ctrl` + `Shift` + `P` to toggle **Command Palette** and go to **Remote - SSH: Connect Current Window to Host...**
7. From the dropdown list, choose `workbench2` and hit `Enter`.
8. Type your CS account password into the pop-up input box and hit `Enter`.
9. Now you are logged in.

Now you are ready to edit files, run scripts, and use the terminal (Press `F1` then type **toggle terminal**) directly from your VS Code environment! More usages of VS Code can be found on its website[3].

(Optional) Download Materials for mini-Labs and tutorials

With a connection to workbench2, use `git` command to download the material from remote git repository for labs (both subclasses) and tutorials (subclass A). For both subclasses, we share the same lab materials but a difference in tutorials. For students in subclass B, feel free to read README.md in each folder if you would like to use these tutorial materials.

Clone the Repository: Use `git clone` command in your terminal to download the code repository from GitHub:

```
    git clone https://github.com/aiot-lab/HKU-COMP3230A-Tutorialabs.git
```

[3] https://code.visualstudio.com/learn

Navigate to the Downloaded Repository: Now, change your current directory to HKU-COMP3230A-Tutorialabs using `cd` command in the terminal.

```
cd HKU-COMP3230A-Tutorialabs
```

Get Updates: To pull the latest updates from the repository to your local repository, just run `git pull` inside the **HKU-COMP3230A-Tutorialabs** directory.

```
git pull
```

For more information on basic git commands, you can follow the online tutorial[4].

Task 2: Linux Commands

In this section, you will experiment with some common Linux commands:

`cd` Change Directory Command `cd` navigates to the specified working directory.

```
cd [directory_path]
cd ~  # Navigate to home directory. Usually /home/<username>/
cd .. # Navigate up one directory level (parent directory)
```

`touch` Create File `touch` creates an empty file.

```
touch [file_name]
touch newfile.txt # Create newfile.txt
```

`mkdir` Make Directory `mkdir` creates a new directory.

```
mkdir [directory_name]
mkdir -p dir1/dir2 # Create nested directories
```

`ls` List `ls` lists directory contents.

```
ls [options]
ls -t  # Sort by time
ls -al # Detailed list
```

[4] https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html

`grep` Search The `grep` command is used for pattern searching in files.

```
grep [pattern] [file]
grep -i "pattern" [file]  # Case-insensitive search
grep -o "pattern" [file]  # Show only the matching part
grep -E "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" [file]  # Match IP addresses
```

`|` Pipe `|` passes output to another command.

```
ls | grep ".txt"
```

`>` Redirect Output `>` redirects standard output to a file.

```
ls > output.txt
```

`»` Append Output `»` appends standard output to a file.

```
ls >> output.txt
```

`wc` Word Count `wc` counts lines, words, and characters.

```
wc [file]
wc -l [file] # Count lines
```

`cat` Concatenate `cat` displays file contents.

```
cat [file]
cat [file1] [file2] > [file3] # Concatenate and redirect
```

`ps` Process Status `ps` displays active processes.

```
ps [options]
ps aux # Display processes with extended user information
```

Notes: Practicing these commands will help you become more comfortable operating in a Linux environment, allowing for more effective and efficient navigation and operation. There is a brief tutorial for these commands. If you would like to know more, you can use `man <command>`

to open up system manual book for a given command. Also, you can find an online version online[5].

Task 3: Practice: Role play

Background: On January 4, 2023, a series of irregularities were detected in the server logs of a critical infrastructure Linux server. As the appointed Linux administrator, **your responsibility is to sift through these logs to identify any suspicious failed SSH attempts** that might indicate potential security breaches or attackers trying to gain unauthorized access.

Find out who is attacking us!

As an administrator, you should find out who is attempting attacks against our server. Usually, an attacker will try to guess the username & password via SSH connections. Fortunately, this behavior will be logged to `/var/log/auth.log` on Ubuntu. Unzip the log file using `tar xvf auth.log.tar.gz` and analyze it in the command line.

Here are some hints for you:

1. The commands `cat`, `head`, and `tail` are useful when working with text files. Specifically, `head` and `tail` display the initial and final lines of a file, respectively. To specify the number of lines to display, employ the `-n` option.

   Previewing Large Files: When dealing with extensive files comprising thousands of lines, it's impractical to view the entire content directly in the command line. Typically, we employ `head` to glimpse the initial lines or `tail` to inspect the most recent entries.

2. Upon previewing the contents of `auth.log`, we notice a variety of authentication activities. Crucially, failed attempts are flagged with the "Failed" keyword, which is subsequently followed by the IP address of the origin. Consequently, our analysis strategy hinges on **filtering lines containing the "Failed" keyword and then extracting the associated IP address.** To achieve this, we can chain multiple commands using pipes, allowing the output of one command to serve as the input for the next.

   Match IP address: You can use regular expression to match IP address like `192.168.1.1` using
   `grep -oE <regular_expression>`. See `grep` example above.

3. Finally, for the purpose of future reference and potentially blacklisting malicious IPs, we aim to store the output in a file `suspicious_ips.txt`. The symbols > (for creating or overwriting a file) and » (for appending to a file) are instrumental in directing the output to a desired file.

4. Write your own command in a shell file named `lab0_<your_student_id>.sh` and run it using `bash lab0_<your_student_id>.sh` . Just like running Linux commands directly in the command line. The shell file will be executed in the same logic.

[5] https://man7.org/linux/man-pages/index.html

Block Threatening IPs

After identifying the suspicious IPs, it's time to enact protective measures. One standard cybersecurity measure is to block these potentially harmful IPs if they have multiple failures consecutively. Note that blocking an IP should always be performed carefully and responsibly. Here, it's just a conceptual practice for this lab.

Submission

(Optional, 0 pt) Submit your script file as `lab0_<your_student_id>.sh`. It's not required but submission is welcomed if you would like some feedback.

Appendix

```
// A preview of few lines in auth.log
Jan  3 21:25:22 sshd[3578590]: PAM service(sshd) ignoring max retries; 6 > 3
Jan  3 21:25:23 sshd[3578675]: Invalid user test2 from 101.42.25.236 port 37198
Jan  3 21:25:23 sshd[3578675]: pam_unix(sshd:auth): check pass; user unknown
Jan  3 21:25:23 sshd[3578675]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=101.42.25.236
Jan  3 21:25:25 sshd[3578675]: Failed password for invalid user test2 from 101.42.25.236 port 37198 ssh2
Jan  3 21:25:25 sshd[3578675]: pam_unix(sshd:auth): check pass; user unknown
Jan  3 21:25:27 sshd[3578675]: Failed password for invalid user test2 from 101.42.25.236 port 37198 ssh2
Jan  3 21:25:28 sshd[3578675]: pam_unix(sshd:auth): check pass; user unknown
Jan  3 21:25:31 sshd[3578675]: Failed password for invalid user test2 from 101.42.25.236 port 37198 ssh2
```