Internship carried out by
Guillaume BIENFAIT
M1 Info @ UBx
2021-2022

Supervised by
Aurelien ESNARD & Abdou GUERMOUCHE

With technical support of
Christophe DELMON
Outline

I. Context
II. Architecture
III. Webterm
IV. Current Status
Context – QemuNet

A solution based on virtualization to learn network (routing, firewall, …)

⇒ a light shell script based on Qemu Virtual Machine and VDE Virtual Switch

(1) describe your own network topology

```
# Topology: LAN
#    opeth   grave
#       \/
#      [s1]
#     /   \
#    syl   immortal

SWITCH s1
HOST debian immortal s1
HOST debian opeth s1
HOST debian syl s1
HOST debian grave s1
```

(2) start your virtual network (as root), break it, and start again…

⇒ [https://github.com/orel33/qemunet](https://github.com/orel33/qemunet)
$ ssh cremi                  # connect cremi gateway
$ ssh <host>                 # need to wake up <host> remotely if needed
$ /net/ens/qemunet/qemunet.sh -b -d tmux -t /net/ens/qemunet/demo/lan.topo
$ tmux a

Context – How to use QemuNet at Home?

Former solution based on SSH + TMux used by students during COVID period!!!
Context – QemuWeb

My Internship – develop a web version of QemuNet → allow students to use QemuNet from a simple browser…

Related Softwares

Filius Network Simulator (standalone software)

PT Anywhere (web app based on Cisco solution)
Architecture – Client-Server

client-side (frontend)

- Design a virtual network topology
- Host & Switch settings
- Display and interact with virtual machine via web terminals

server-side (backend)

- Handle multiple users (a single session by user)
- Run a virtual network session (based on Qemunet)
- Ensure the persistence of a session (e.g. network failure)
- Preserve the integrity of the host machine (security issue)

Requirements: Performance (WebSocket), HTTPS, Authentication (OpenID), Security Issues, Deployment at CREMI, ..
Architecture – Overview

Frontend

Backend

Node.js Web Server

HTTPS

Web

App

WebSocket

HTML  CSS  Vue.js  Xterm.js  node

Express  JS

Firejail  tmux
WebTerm – Proof-of-Concept for QemuWeb

Preliminary Project – just runs an interactive shell (on a remote server) and plays with it in your browser

- Display and interact with multiple remote terminals (based on xterm.js)
- Performance achieved using WebSocket technology (socket.io)
- Ensure persistence of multiple sessions (tmux)
- Security Issues: session containment in a harmless jail (firejail)

Idea – deploy it on the web to help students to learn basic Linux commands at home (in a sandbox)
QemuWeb – Current Status

- **Design Mode**
  - edit a virtual network topology → interconnect hosts and switches
  - host & switch settings (system, name, interfaces, …)

- **Execution Mode (true host & network emulation)**
  - components terminal can be displayed
  - Terminals are movable and resizable
QemuWeb – To Do

● Connect frontend `xterm.js` terminals with backend pty instances

● Import/Export a topology file

● Some issues to fix
  ○ Inputs/outputs do not center when changed sides
  ○ Curved connections do not connect correctly when the input is to the right of the switch
  ○ Hosts can have the same name (identification issue)
QemuWeb – Short Demo
Thank you for listening!
Appendix
Design Mode – create & modify virtual network topology
Design Mode – Host Settings
Execution Mode (Network Emulation)
Security Issues

What if a remote client...

- can run network commands on server machine?
- can see and modify others files?
- had root access to the server machine?

➢ We have to confine users in separated *jails* and give them only a restricted environment