Computer Networks
@CS.NCTU

Lab. 1: Packet Manipulation via Scapy

Tools Introduction

Location: EC-315, 316

Instructor: 陸勇盛 (David Lu)

Tools

- Docker
- Linux Networking
- Python and Scapy
- Wireshark
- Git and GitHub

Docker

- **Docker** separate applications from infrastructure to deliver software quickly.
- Docker provides the ability to package and run an application in a loosely isolated environment called a container.
  - Run many containers simultaneously on given host
  - Run Docker containers within host machines that are actually virtual machines
- **Docker Hub** is a cloud-based registry service.

Docker (cont.)

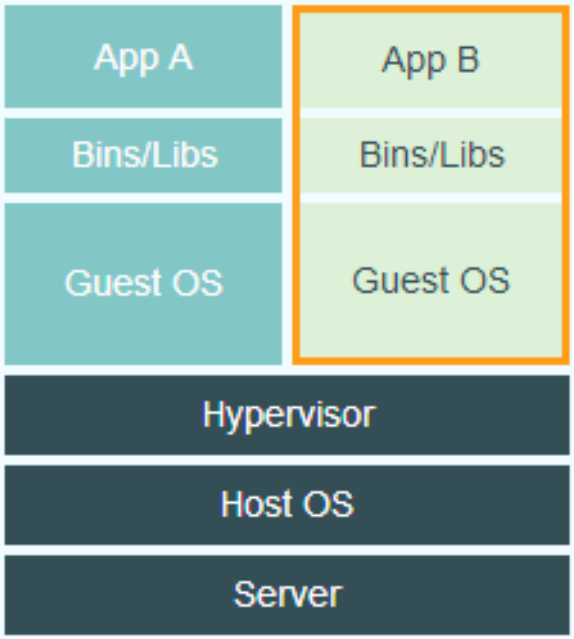
- **Docker command line**
  - To list available commands, either run **`docker`** with no parameters or execute **`docker help`**:

```
$ docker help
```

- **Dockerfile**
  - Contain all the commands a user could call on the command line to assemble an image
  - Build an image from Dockerfile

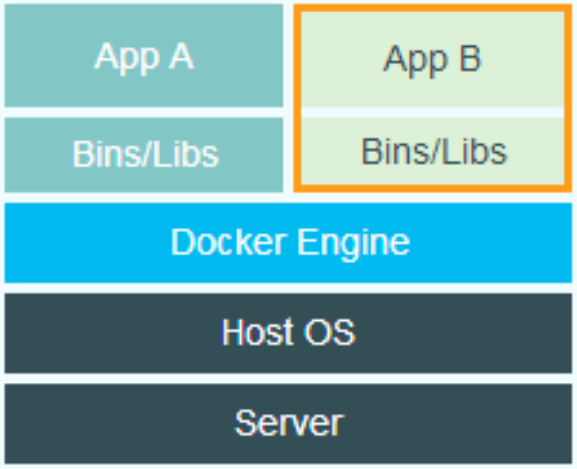
```
$ docker build [-t <TAG>] [-f <PATH_TO_DOCKERFILE>] .
```

VM vs. Docker



Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.



Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

Some Docker command line

- List images

```
$ docker images
```

- Pull an image or a repository from a registry

```
$ docker image pull <NAME>[:TAG]
```

- Push an image or a repository to a registry

```
$ docker image push <NAME>[:TAG]
```

- Build an image from a Dockerfile

```
$ docker build [-t <TAG>] [-f <PATH_TO_DOCKERFILE>] .
```

- Log in to a Docker registry

```
$ docker login
```

Some Docker command line (cont.)

- List containers

```
$ docker ps
```

- Create a new container

```
$ docker create [OPTIONS]
```

- Kill one or more containers

```
$ docker kill [OPTIONS] <CONTAINER…>
```

- Copy files/folders between a container and the local filesystem

```
$ docker cp [OPTIONS] <SRC_PATH> <CONTAINER>:<DST_PATH>
$ docker cp [OPTIONS] <CONTAINER>:<SRC_PATH> <DST_PATH>
```

- Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

```
$ docker tag <SOURCE_IMAGE>[:TAG] <TARGET_IMAGE>[:TAG]
```

Tools

- Docker
- Linux Networking
- Python and Scapy
- Wireshark
- Git and GitHub

OpenSSH

The SSH command is used from logging into the remote machine, transferring files between the two machines, and for executing commands on the remote machine.

- Log into a remote machine

```
$ ssh [-p <PORT>] <USERNAME>@<IP_ADDR>
$ ssh [-p <PORT>] <USERNAME>@<DOMAIN_NAME>
```

- File transfer client with RCP-like command

```
$ scp [-P <port>] <SOURCE_PATH> <TARGET_PATH>
# <TARGER_PATH>: <USERNAME>@<IP_ADDR>:<PATH>
# <TARGET_PATH>: <USERNAME>@<DOMAIN_NAME>:<PATH>
```

Linux Networking

- Create network namespaces

```
$ ip netns add <NAMESPACE>
```

  Display network namespaces

```
$ ip netns show
$ ip netns list
```

- Configure interfaces in network namespaces

```
$ ip exec <NAMESPACE> <COMMAND>
```

- Configure connection between interfaces

```
$ ip link add <INTERFACE> type veth peer name <INTERFACE>
```

- Configure network interfaces

```
$ ip link set dev <INTERFACE> [up/down]
$ ip link set <INTERFACE> address <MAC_ADDR>
$ ip addr add <IP_ADDR> dev <INTERFACE>
```

Linux Networking (cont.)

- Disable IPv6

```
$ sysctl net.ipv6.conf.<INTERFACE>.disable_ipv6=1
```

- Configure default route

```
$ ip route add default via <IP_ADDR>
```

- Display routing table

```
$ ip route show
$ ip route list
```

Tools

- Docker
- Linux Networking
- Python and Scapy
- Wireshark
- Git and GitHub

Python

- Beginner's Guide
  - For programmers
  - For non-programmers

- Python 2.7.12 documentation

- Python Package Index (PyPI)
  - Hosts thousands of third-party modules for Python
  - Install module from PyPI

```
$ pip install <MODULE_NAME>
```

Scapy

- **Scapy** is a interactive packet manipulation program for Python.
  - forge or decode packets of protocols,
  - send packets to wire,
  - capture packets,
  - match requests and replies, etc.

- Example of Scapy (Scapy's documentation)

```
Welcome to Scapy (2.4.0)
>>> a = IP(dst="172.16.1.40")
>>> a
<IP dst=172.16.1.40 |>
```

Tools

- Docker
- Linux Networking
- Python and Scapy
- Wireshark
- Git and GitHub

Wireshark

- **Wireshark** is a widely-used network protocol analyzer.
  - Deep inspection of hundreds of protocols
  - Live capture and offline analysis
  - Most powerful display filter
  - Read/write many different capture file formats
- Examples of DisplayFilter
  - Show only SMTP (port 25) or ICMP

```
>>> tcp.port eq 25 or icmp
```

  - Show any traffic to or from 10.0.0.1

```
>>> ip.addr == 10.0.0.1
>>> ip.src == 10.0.0.1 or ip.dst == 10.0.0.1
```

Filtering Rules

- Filter the packets that satisfy some conditions
  - For example, to find TCP packets with a port number of 80, you can use **tcp.port == 80**
- For more filter instructions, please reference to:
  - Building display filter expressions
  - DisplayFilters
- Frequently used:
  - ip.src, ip.dst, ip.addr, …(IP address)
  - tcp.port, tcp.srcport, tcp.dstport, … (port)
  - eth.src, eth.dst, eth.addr, ... (MAC address)

Tools

- Docker
- Linux Networking
- Python and Scapy
- Wireshark
- Git and GitHub

Git & GitHub

- **Git** is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
  - Pro Git Book – (ENG / CH)
  - 連猴子都能懂的 Git 入門指南 (CH)
  - Slides - Let's Git (CH)
  - Online practice - Try Git / Codecademy – Learn Git
- **GitHub** is a web-based hosting service for version control using Git

Some Git command line

- Get and set repository or global options

```
$ git config --global user.name "<NAME>"
$ git config --global user.email "<EMAIL>"
```

- Create an empty Git repository or reinitialize an existing one

```
$ git init
```

- Show the working tree status

```
$ git status
```

- Add file contents to the index

```
$ git add <FILENAME>
```

- Record changes to the repository

```
$ git commit -m "<COMMIT_MESSAGE>"
```

Some Git command line (cont.)

- Manage set of tracked repositories

```
$ git remote add origin <REPO_URL>
```

- Update remote refs along with associated objects

```
$ git push <REMOTE_REPO> <LOCAL_BRANCH>
```

Useful References

- Docker
  - Dockerfile reference
  - Docker CLI reference
- Python
  - Beginner's guide – Programmers / Non-programmers
  - Python 2.7.12 documentation
- Scapy
  - Documentation / PDF Documentation
- Wireshark
  - Wireshark DisplayFilters
  - Building display filter expressions
- Git
  - Pro Git Book – (ENG / CH)
  - 連猴子都能懂的 Git 入門指南 (CH)
  - Slides - Let's Git (CH)