ProBACS: Profile-Based Automated Compilation System Installation Guide

Myra Iltefat, Shipei Zhou, Qiyang He, Hanfu Zhang

Introduction

- Our system supports a full web user interface and a command line interface. The command line interface can only be triggered from a Linux OS environment.
- The directory “Probacs” will be our valid project folder.
- To execute regular compile, set up host server, register compilers to the host server as well as adding related profiles to support compile. After setting up this platform server, compilation is ready to go.
- We have also provided some configuration examples.
    - Go to **`Probacs -> experiment-data`**, we have example source file (hello.c) as well as other sample task file and configuration files.

Configuring the system

A] Setting up the host server

1) Installing python3, pip and django

- **`sudo apt-get update`**
- **`sudo apt-get install python3-pip`**
- **`sudo -H pip3 install --upgrade pip`**
- **`sudo -H pip3 install django==1.11`**
- **`sudo reboot`**

2) Installing the git-core and requests libraries

- **`sudo apt-get install git-core`**
- **`sudo pip3 install requests`**

3) Preparing the host-server environment

- Copy the code from Github.

  `git clone` ***`https://github.com/SeveroeHe/secu-tools.git`***

- Rebuild the database using the following script:

  `./rebuildDatabase.sh`

- Specify the server ip address using the following steps:
    - Open the file “config.ini”.

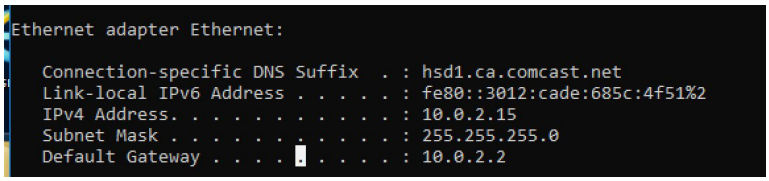
- Change “Allow_localtest” to be “false”
- Modify “Local_ip” to the ip address your host server is hosted on (for example): **`http://localhost:8000`**
- Modify “Gateway” to be the ip address of your host server that others can reach (used for interacting with platform server)
  - If you set up your platform server on a public address, you need to change this section value to be the **`<public_ip_address>:<port>`** of your host server
  - If you are using virtualbox for hosting platform server, open any of your instance, check network interface configuration

    On linux/mac, run terminal with command “`netstat -rn`”,

    - find Gateway ip corresponding to ip destination “0.0.0.0”
    - In our case, the gateway is 10.0.2.2, so we changed the “Gateway” section to be: http://10.0.2.2:8000
    - The port is the one that your server is listening on

    On windows, run command line with “`ipconfig`”

    - Get default gateway ip



    - In this case we should change “Gateway” section to be: http://10.0.2.2:8000

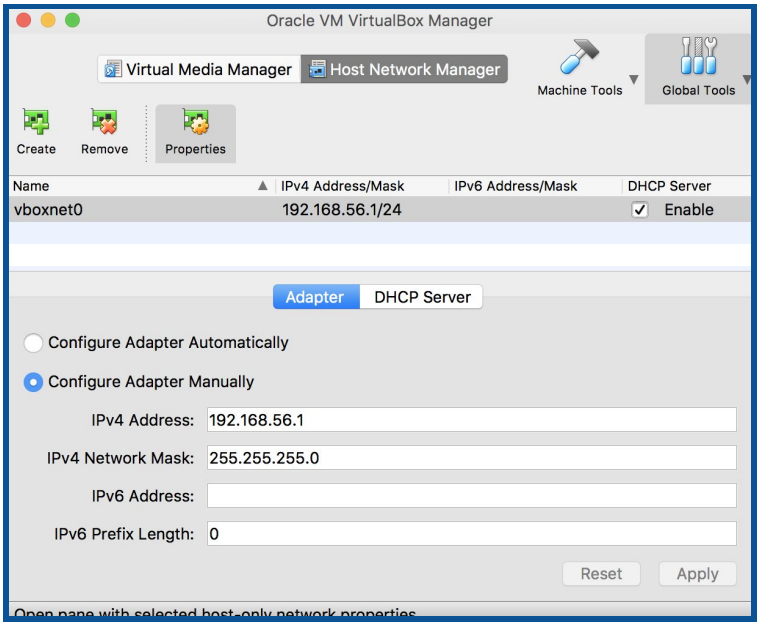
- Run the host server using the following script:

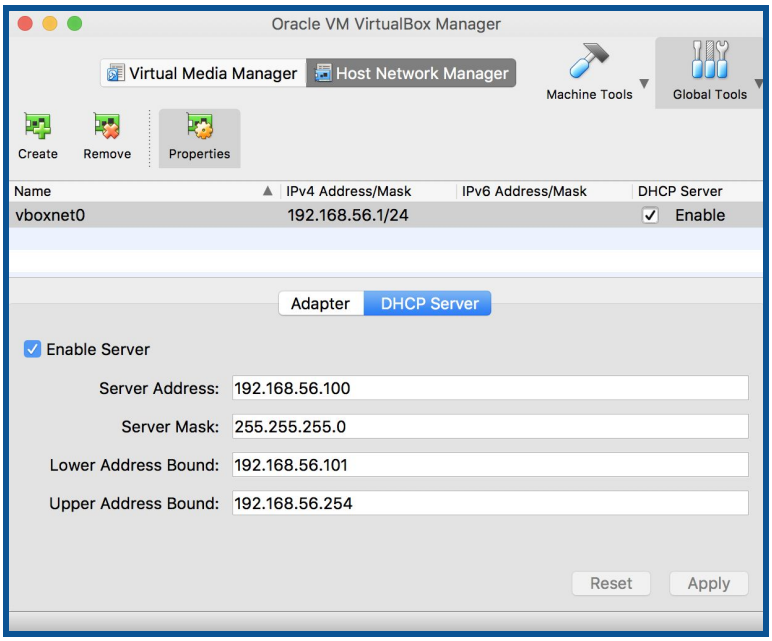
  ./run.bash

B] Setting up the platform servers

1) Configuring host-only network settings
   - On the VM virtualbox manager panel, go to **Global Tools -> Host Network Manager -> create**

- After creating a new adapter, configure the adapter as shown below





- To choose this new adapter for platform server instance, go to **Settings->Network ->Adapter 2,** choose “host-only” adapter mode and this new adapter. Then restart this VM instance

2) Configuring the linux platform server

i) Installing python3, pip and django

- `sudo apt-get update`

- `sudo apt-get install python3-pip`
- `sudo -H pip3 install --upgrade pip`
- `sudo -H pip3 install django==1.11`
- `sudo reboot`

ii) Installing the requests and git-core libraries

- `sudo apt-get install git-core`
- `sudo pip3 install requests`

iii) Preparing the platform-server environment

- `git clone` ***https://github.com/SeveroeHe/secu-tools.git***

iv) Setting up the linux platform server

- Specify the platform server ip/port by adding new compilers to our system from host server web UI
  - You could find the instance ip address by typing the command: `ifconfig`
  - The ip address should be something similar to “192.168.56.101”
- Register related profiles with this compiler.
- Set up the platform server.
  - Make database migrations: python manage.py migrate
  - Run server: `./runLinux.sh`
- Test compilation - you should be able to compile executables from compiler set up on this platform server.

3) Configuring the windows platform server

i) Installing python3, pip and django

- Download python 3.6.5 x86-64 web-based installer from official and install
- Modify to install pip and remember to check “add Python to environment variables” when installing python
- After installation, use the command **`“pip install Django==1.11”`** in Command Prompt to install Django.
- Use **`“pip install requests”`** to install requests package

ii) Installing gnuwin32

- https://sourceforge.net/projects/getgnuwin32/
- The gnu tar executable path should be

"C:\Program Files (x86)\GnuWin32\bin\tar.exe"

iii) Installing Visual Studio

- Download Visual Studio Community 2017 from official and install
- Remember to check “Desktop development with C++” to include CL

iv) Downloading the django server package from github

- `git clone` ***https://github.com/SeveroeHe/secu-tools.git***

v) Setting up the windows platform server

- Specify platform server ip/port by adding new compilers to our system from host server web UI
  - You could find instance ip address by type command: `ipconfig`
  - The ip address should be something similar to “192.168.56.102”
- Register related profiles with this compiler
- Set up the platform server
  - Make database migrations: `python manage.py migrate`
  - Run server by run script: `runWin.bat`
- Test compilation - you should be able to compile executables from compiler set up on this platform server.

C] Setting up the command line interface

1) Setting the alias:

- Go to directory host-server, use pwd to get current definite path
- Open configuration file: `Vi ~./bashrc`
- Add alias: `alias probacs=”python <path>/probacs.py”`
- Restart configuration file: `Source ~./bashrc`

2) Running the command-line tool:

- Check the user manual for details on the syntax.