

Activity 2 - Basic Linux Networking

Daniel Schoonwinkel – E&E Postgraduate Networking Course 2019

May 21, 2018

1 Introduction

In this part of the networking course you will learn to install Linux programs, write, compile and execute very basic Python and C++ programs and use low-level networking sockets on Linux.

2 Practical examples

In this section we will present a walkthrough of commands and programs that will allow you to use Linux network sockets. The information contained in this section will enable you to complete the assignment at the end of this activity.

2.1 Installing programs in Linux

These are a couple of ways that you can install programs in Linux, and this section will discuss 3 ways: the XUbuntu package manager *APT*, compiling from source and installing using the Python packet manager *Pip*.

2.1.1 *Advanced Packaging Tool (APT)*

Most Linux distributions have a package manager. Package managers are tools that provide pre-compiled binary files (executable programs and software libraries) as packages for your specific version of the operating system (OS). The managers of the your flavour of Linux - in XUbuntu's case that is Canonical - may impose some constraint or quality control requirement on binary files that are included as packages in the official package repository. Developers can also maintain their own repository, but Canonical then provides little guarantee of the usability or security of the external repository.

In XUbuntu, the package manager can be invoked from the commandline using `apt-` commands. For example, if you want to search for Wireshark packages, you would use the following command:

```
apt-cache search wireshark
```

This will show all the packages that are related to Wireshark in some way. More details on Wireshark will be given later in this walkthrough. Note that there are both programs and libraries included in the list. Libraries are indicated with the 'lib' prefix.

To install a program using *APT*, you would use the following command:

```
sudo apt-get install wireshark
```

Note that the command `sudo` is added before the install command: this is necessary because only the root user or users with root privileges are allowed to install packages. On Linux, the root user is similar to an Administrator on Windows.

You should now be able to run Wireshark from the commandline as follows:

```
sudo wireshark
```

We will use Wireshark later, for close the program and continue with the walkthrough.

As you can see, using the package manager is very easy and no extra configuration is needed. The package manager is generally the preferred method of installing programs on Linux.

However, at times the program that you require is not available in the official XUbuntu repository. For example, *Sublime Text 3* is not available in the official repository, but in a separate developer repository.

The instructions for using this repository is at [https://www.sublimetext.com/docs/3/linux_repositories.ht](https://www.sublimetext.com/docs/3/linux_repositories.html)

Go ahead and install *Sublime* now according to the `apt` instructions.

We suggest using *Sublime* for text editing and programming as it is lightweight and performs syntax highlighting in various computer programming languages. Note: it is not a complete IDE, but compiling and running will be done from the commandline.

Finally, to remove a package, use:

```
sudo apt-get remove wireshark
```

2.1.2 *Pip* - Python's package manager

Pip is the Python equivalent of `apt`, but only for Python modules. *Pip* is cross-platform (as is Python), and can therefore be used on Windows and Mac OSX as well. It works with the following commands (*Pip* might need to be installed with `apt` before it is available for use):

```
pip3 list #lists the installed python packages
pip3 search pypacker #shows all packages related to pypacker
pip3 install pypacker #installs package
pip3 uninstall pypacker #removes package
```

See [1] for more details on

2.1.3 Compiling and Installing from source

3 Assignment 2: Write your own ping-request and ping-responder.

4 References

List of sources:

[1] https://pip.pypa.io/en/stable/user_guide/