Homework #1

Due Time: 2015/3/16 (Mon.) 17:00
Contact TAs: `vegetable@csie.ntu.edu.tw`

Submission

- Compress all your files into a file named " ⟨**StudentID**⟩ **.zip**", which contains two folders named **StudentID_NA** and **StudentID_SA** respectively.
- **Folder StudentID_NA** should contain a pdf file of all your answers in *Network Administration Part*, and a file named "⟨*studentID*⟩ *.pcapng*".
- **Folder StudentID_SA** should contain all files in *System Administration Part*.
- **Submit your zip file to ceiba.**

Instructions and Announcements

- Discussions with others are encouraged. However, you should write down your solutions `in your own words`. In addition, for each problem you have to specify the references (the Internet URL you consulted with or the people you discussed with) on the first page of your solution to that problem.
- Problems below would be related to the material taught in the class and might be far beyond that. Try to search for additional informaition on the Internet and give an reasonable anwser.
- Some problems below might not have standard solution. We would give you the point if your answer is followed by reasonable explainations.
- If you get stuck in problems below, feel free to contact TAs.
- **NO LATE SUBMISSION IS ALLOWED**.

Network Administration

Part 1

While configuring the settings of Internet in a PC, we often encounter a keyword “TDP/IP”, which is actually a combination of two protocol, “TCP” and “IP”. Known that Internet is often classified into 5 layers in networking industry, the ISO/OSI model, however, tries to standardize the communication model, making it a 7-layer model, from the lowest layer, “Physical layer”, to the highest layer, “Application Layer”. In either model, a wellness of upper layer relys on stable connection of lower layer during communication, just like construction of upper floor relies on stability of lower floor while constructing a building.

a) According to the regulation of ISO/OSI model, to which layers TCP and IP respectively belong? Please write down the English name of the layers they belong. (10%)

b) UDP is at the same layer as TCP, but most of the servers that require stable connections over the Internet would adopt TCP, rather than UDP, as their transport layer protocol. Is this a reasonable choice, why or why not? (20%)

c) Assume that you are using a browser. Whenever you try to fetch a web page, TCP and IP would both be working on your system to achieve your request. Besides those two protocols, what protocols are also used during the whole process of fetching a web page from remote. Please name one protocol which belongs to application layer, and describe what role it plays in the whole process. (20%)

Part 2

As an identification on the Internet, IP address is a critical information for two host to communicate. To send packet to the correct destination host, a sender needs to know the IP address of the reciever. IPv4, representing an IP address with 32 bits, is still the most general version of IP address. The total number of avaible address of Ipv4 is no larger than $2\hat{3}2$ and today we are running out of them. IPv6, with 128-bit long per address, is being promoted and claiming to let any sand on Earth to have a single IP address.

a) There are tools available online to translate URL to respective IP address. The IP address of NTU web server (www.ntu.edu.tw) is 140.112.8.116, which is apparently an IPv4 address. In fact, the web server of NTU-CSIE (www.csie.ntu.edu.tw) also uses IPv4 address. What is the IP address of the web server of NTU-CSIE? (15%)

b) Due to the limit amount of IP address, allocation of IP addresses is strictly managed by IANA(Internet Assigned Numbers Authority) and it is visible online. We can easily look up IP segments and countries or organizatinos that own them. However, you can’t find the owner of some centain IP addresses, such as 192.168.0.1. Why is that? (15%)
(Hint: Distinguish the differences between Public IP and Private IP)

Part 3

Please execute WireShark in any of the computer platform you can access. Try to connect to linux1.csie.ntu.edu.tw(140.112.30.32) and record all the packet flow until you login successfully. Then utilize the built-in filter to get the pakcets that the destination IP is 140.112.30.32 and the protocol being TCP. Write down the conditions you designed for filtering, and save the results as a file named “*⟨studentID⟩ .pcapng*”, which must also be uploaded as a part of your homework. (20%)

Part 4 (Bonus 30%)

NTUCC (Computer & Information Networking Center, NTU) is the primary network uplink of our whole campus. They would monitor the network traffic and take actions to ensure network security if needed. For example, blocking an IP address, of which the host is suspected of sending harmful packets or behaving abnormally. That host would not be able to connect to the Internet until the security issue is identified and solved. Like NTUCC, we, the network administration team of NTU-CSIE have our way to block a harmful host in our building. Rather than blocking the IP address, we block its MAC address to cause the similar effect.

a) What is MAC address? (15%)

b) Compared to IP address blocking, give an advantage of MAC address blocking. (15%)

System Administration

Shell Scripts

A big advantage of using commands is that it is easy to combine several simple commands to do a more complex task. Writing shell scripts is the most common and easy way to make use of command line tools to automate routine tasks, manipulate files or process output.

Please write a shell script for each following problem and send all scripts in a tarball. All scripts should have their execute permission set, so it can be used as a regular executable.

You are allowed to use the standard (POSIX) shell or GNU Bash.

- If you use the standard shell, the first line of your script must be `#!/bin/sh`. You can use dash on GNU/Linux or sh on *BSD to test your script. You are encouraged to write scripts using the standard shell, but you must not use any bash-specific features in a `#!/bin/sh` script. You can use `checkbashisms` command to check whether the syntax is allowed by the standard.
- If you use GNU Bash, the first line of your script must be `#!/usr/bin/env bash`. Please don't use `#!/bin/bash` because hard-coding the path of bash is known to cause problems on many platforms, including most *BSD systems.

1. Find queue IDs of junk messages

You are notified that some workstation accounts are stolen and being used to send junk mail. After you lock all stolen accounts and drop packets from attackers, you should delete all junk messages submitted by attackers in the queue of the mail server. You are asked to write a script to read the output of `mailq` command and print queue IDs of junk messages, so we can remove them from queue using `postsuper -d` command.

When the user of your script run the following command:

```
mailq | ./find-junk-messages.sh b12345678@csie.ntu.edu.tw | postsuper -d -
```

Your script should print the IDs of all messages submitted by `b12345678@csie.ntu.edu.tw` account.

- Input: the format output by `mailq` command

```
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-------
751068E490     8723 Tue Jan 27 00:53:38  owner-b12@list.csie.ntu.edu.tw
(temporary failure. Command output: local: fatal: execvp /usr/ucb/vacation: No such
file or directory)
                                         b12345678@csie.ntu.edu.tw

DB66D8E20E     3098 Wed Jan 28 02:37:58  b12345678@csie.ntu.edu.tw
(host mx2.comcast.net[68.87.20.5] refused to talk to me: 554 resimta-ch2-06v.sys.comcast.net
comcast 140.112.30.68 Comcast block for spam.  Please see http://postmaster.comcast.net/
smtp-error-codes.php#BL000000)
                                         ajolsson@comcast.net
                                         pmidtvedt@comcast.net
```

```
772018E234*    2898 Wed Jan 28 07:33:20  b12345678@csie.ntu.edu.tw
(delivery temporarily suspended: host mx2.comcast.net[68.87.20.5] refused to talk to
me: 554 resimta-ch2-07v.sys.comcast.net comcast 140.112.30.68 Comcast block for spam.
Please see http://postmaster.comcast.net/smtp-error-codes.php#BL000000)
                                         world@comcast.net

BF608910F4     9095 Thu Jan 29 12:24:32  b23456789@csie.ntu.edu.tw
        (connect to abc.csie.ntu.edu.tw[140.112.123.234]:25: Connection refused)
                                         hello@abc.csie.ntu.edu.tw

-- 24 Kbytes in 4 Requests.
```

- Output: the format accepted by `postsuper -d` command

```
DB66D8E20E
772018E234
```

2. A compiler wrapper to workaround some broken build systems

You are going to compile a large project which has hundreds of Makefiles. Unfortunately, its build system is much broken, and it hard-code several compiling and linking arguments which are not easy to change using environment variables. Therefore, you decide to write a compiler wrapper to trick its build system to run your wrapper instead of running the compiler directly, so you can add needed arguments.

Please write a wrapper which executes the real program located in `/usr/bin` with some arguments added. Your script should exit with the same status as the real program, so the build system can know whether the compilation is successful. You should add compiling argument `-isystem /usr/local/include` before all the other arguments or linking argument `-Wl,-Y/usr/local/lib` after all the other arguments depending on the following rule:

- If your script is named after `cc` or `c++`, you should add both the compiling argument and the linking argument. However, if any one of `-c`, `-S` or `-E` is used, you should add only the compiling argument.
- If your script is named after `cpp`, you should add only the compiling argument.

You can write only one script. We will make hard links or symbolic links to your script, so it can have several names. Please see the following examples.

```
# When an user run
cc world.c -c -o world.o
# Your script should run
/usr/bin/cc -isystem /usr/local/include world.c -c -o world.o

# When an user run
c++ -o project project.o hello.o -lcrypto -lm
# Your script should run
/usr/bin/c++ -isystem /usr/local/include -o project project.o hello.o -lcrypto \
  -lm -Wl,-Y/usr/local/lib
```

```
# When an user run
cpp -D_POSIX_C_SOURCE=200809L -D_XOPEN_SOURCE=700 mypkg.c
# Your script should run
/usr/bin/cpp -isystem /usr/local/include -D_POSIX_C_SOURCE=200809L \
  -D_XOPEN_SOURCE=700 mypkg.c
```

3. Print the most recent login time and desktop session of an user

Many desktop and workstation systems have AccountsService installed now. Although it provides a simple API for applications to use, there is no official command line tool to read information from the service. Please write a script to show the most recent login time in YYYY-MM-DD hh:mm:ss format and the most recently used desktop session of a user specified on the command line.

You don't need to install AccountsService or learn the API of AccountsService to complete this task. You should be able to generate similar output to test your script.

For example, when an user run

```
./show-recent-logintime-and-desktopsession ta217
```

you can use following commands to retrieve needed information.

```
# Find the passwd entry of the user
$ getent passwd ta217
ta217:x:41610:200:ta217:/home/dept/ta/ta217:/bin/bash

# As we only need the uid of the user, using id command is enough
$ id ta217
uid=41610(ta217) gid=200(ta) groups=200(ta),1000(admin)

# Get the most recent login time of the user from AccountsService
$ dbus-send --system --print-reply --dest=org.freedesktop.Accounts \
> --type=method_call /org/freedesktop/Accounts/User41610 \
> org.freedesktop.DBus.Properties.Get \
> string:org.freedesktop.Accounts.User string:LoginTime
method return sender=:1.18 -> dest=:1.165 reply_serial=2
   variant       int64 1422614246

# Get the most recently used desktop session of the user from AccountsService
$ dbus-send --system --print-reply --dest=org.freedesktop.Accounts \
> --type=method_call /org/freedesktop/Accounts/User41610 \
> org.freedesktop.DBus.Properties.Get \
> string:org.freedesktop.Accounts.User string:XSession
method return sender=:1.18 -> dest=:1.166 reply_serial=2
   variant       string "gnome-classic"
```

Your script should print

```
ta217 gnome-classic 2015-01-30 18:37:26
```

Shared Folders and Files

It is common to have a shared folder or shared git repository when working on a project with several people in a team. ACL can manage complex settings that the standard Unix permission is not be able to. You can use `getfacl` and `setfacl` to view and manage ACL settings.

Please create a shared folder with its permission set to satisfy the following requirements:

1. There are 3 team members: `student1`, `student2` and `student3`. All of them are members of `student` group. They can add, modify or delete any file and folder in this shared folder, regardless the owner of the file or folder.
2. There are 2 TAs: `ta1` and `ta2`. They are allowed to read all files and folders in this shared folder, but addtion, modification and deletion is not allowed.
3. There is 1 professor: `prof1`, having the same permission as TAs.
4. All other people, including users in `student` group that are not a team member, are not permitted to access any file in this shared folder.

The following message is a sample output of `getfacl` command running on the shared folder with several permission sets replaced by underscores. Please fill in all underscores with correct permission settings.

```
# file: project_shared_folder
# owner: student1
# group: student
user::rwx
user:student1:rwx
user:student2:___
user:student3:___
user:ta1:___
user:ta2:___
user:prof1:r-x
group::___
mask::___
other::---
```

Note

Although ACL is an abandoned POSIX standard, it is already implemented on many Unix-like systems, including Linux and FreeBSD. If you get `Operation not supported` error when using `setfacl` command, please try these possible solutions before asking.

1. Ext2, Ext3, Ext4 filesystem on Linux: `mount -o acl,remount /path/to/mount/point`
2. tmpfs on Linux should work without problems.
3. UFS on FreeBSD: `mount -o acls,update /path/to/mount/point`
4. tmpfs on FreeBSD does not support ACL. If you really want to use a ramdisk, please use `mdconfig -a -t malloc -s size` to create a memory disk device and format and mount it like a regular UFS filesystem.