Homework #1

Due Time: 2019/3/10 (Sun.) 22:00
Contact TAs: `vegetable@csie.ntu.edu.tw`

Submission

- Compress all your files into a file named **HW1_[studentID].zip** (e.g. `HW1_bxx902xxx.zip`), which contains two folders named **[studentID]_NA** and **[studentID]_SA** respectively.
- **Folder [studentID]_NA** should contain a pdf file named **na.pdf** of all your answers in *Network Administration Part*.
- **Folder [studentID]_SA** should contain 3 scripts named according to the description in each task.
- Submit on NTU COOL (https://cool.ntu.edu.tw).

Instructions and Announcements

- Discussions with others are encouraged. However, you should write down your solutions **in your own words**. In addition, for **each and every** problem you have to specify the references (the URL of the web page you consulted or the people you discussed with) on the first page of your solution to that problem.
- Problems below will be related to the materials taught in the class and may be far beyond that. Try to search for additional information on the Internet and give a reasonable answer.
- Some problems below may not have standard solutions. We will give you the points if your answer is followed by reasonable explanations.
- **NO LATE SUBMISSION OR PLAGIARISM IS ALLOWED.**

Network Administration

Never DDoS our server! Or you will be severely punished!

Wireshark

> *Wireshark is a free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education.*
>
> (from Wikipedia)

In this section, we will familiarize ourselves with another common usage of Wireshark: eavesdropping. During packet transmission, an eavesdropper on any of hop of the routing path can capture your traffic using Wireshark. We will play the role of such eavesdropper, Q-mao, and capture data traffic of web browsing, transmitted using the HTTP protocol. Nowadays, most websites enforce HTTPS, the encrypted version of HTTP; but older websites still have not caught up. And this is exactly what we are going to exploit.

NANI? Is my uplink as evil as Q-mao? (15%)

Please perform the following steps:

1. Open Wireshark and start packet capturing.
2. Open your browser and go to fakebook-http[1]
3. Try to login with **nasahw1@qmao.is.so.evil** as email and your student ID as password. (**Do not enter your real F***book password**)
4. Stop packet capturing in Wireshark.

Now, answer these questions:

1. Find the packet that contains the email and password you just typed, and screenshot it. (5%)
2. Re-perform the above steps at fakebook-https[2], can you find the email and password this time? If not, does it mean the owner of the web server cannot peek your password during your login? Why or why not? (5%)
3. Give a real-world examples that your password may be eavesdropped by Q-mao when still using the HTTP protocol. (assuming no eavesdropper in client and server-side software) (5%)

In the vast sea, I see nobody but flag (10%)

Try to find the flag in flagsea[3] using Wireshark, and describe how you find it in detail. Screenshot your tools when you discover your flag.

The flag is in the form of `flag{…}`.

[1] http://nasa-hw1.csie.ntu.edu.tw/facebook-login
[2] https://nasa-hw1.csie.ntu.edu.tw/facebook-login
[3] https://nasa-hw1.csie.ntu.edu.tw/flagsea

Piepie! The world needs you! (10%)

Welcome to The Great Pie Factory, a company that specializes in inventing and producing the greatest pies of all times. Our founder, the legendary piepie, has led the company as CEO for over two hundred years, but has gone missing since last month. We believe that he was captured by the evil Arvin(a friend of Q-Mao), a pie hater who aims to erase the entire existance of pies from human civilization. Now, piepie's secret recipes were locked inside our pielygraphy system, and we're offering free pie for lifetime to anyone who can break through the authentication system. Prove you are worthy of the bounty we will award. Our Pielygraph

- The system is here: pielygraph[4].
- Some clues we have found: history.pcapng[5].
- The recipe is in the form of `flag{…}`.

You are required to screenshot the flag in browser or any tools and tell me how you find in detail. Note that any method, except for plagiarism, is allowed but we recommend using Wireshark.

Hints

1. Motto of our company : 如果手中沒有派，我就無法餵飽你；如果手中握著派，我就無法擁抱你。
2. You need to know a little bit about the HTTP protocol or have some spiorad™.
3. Replaying the token in the payload of HTTP POST request is not enough because we have timeout mechanism in the server.

Internet Protocol Stack: 5-layer Model

1. Wireshark can reveal the data fields used by protocols of hierarchical network layers. Use the packet intercepted by Wireshark to explain the purpose of each layer in the *5-layer model*. **(10%)**

   Hint:

   - Use the website metioned in **"NANI? Is my uplink as evil as Q-mao?"**. (HTTP version)
   - Roughly associate data fields to each layer and briefly explain its purpose.

2. In some scenario, the only protocol you can use in *Transport Layer* is UDP (maybe banned by admin). Now, you are asked to send an important file that cannot tolerate any data loss or re-ordering, is it possible to compensate for UDP's unreliability? How? **(5%)** (Extremely simple answer suffices)

[4] https://nasa-hw1.csie.ntu.edu.tw/pielygraph
[5] https://cool.ntu.edu.tw/files/16281/download?download_frd=1

System Administration

Welcome! In the *System Administration* section you are asked to write 3 trivial shell scripts. We intend to assist your journey of shell scripting mastery. Once you get the gist of it, you will unleash the immense automation capability and programmability of your computer. You will be liberated from tedious cursor movements and repetitive tasks; your friend will describe you as followed: *This guy really knows how to use a computer!* Your mother can boast to her freinds: *I now have more quality time with my kid because she has transfered her work to a computer simply by typing something in a dark screen!*

This will be cool!

Before you start, let's make some agreements:

- Start every script with a *shebang*, please.
- Make every script executable (`chmod +x`), please.
- Thou shalt not embed Python or programming languages other than *awk* into your script. Let's focus on shell scripting at this moment, please.
- Use the standard shell (e.g POSIX, sh) or GNU bash, please.

DateTime Format checker (10%)

The following are the supported date & time formats of our system:

```
yyyy-MM-dd HH:mm:ss  ex: 2018-02-28 14:00:05
yyyy-MM-dd HH:mm     ex: 2018-01-12 14:00
yyyy/MM/DD HH:mm:ss  ex: 2018/02/28 14:00:05
yyyy/MM/DD HH:mm     ex: 2018/01/12 14:00
yyyy.MM.DD HH:mm:ss  ex: 2018.02.28 14:00:05
yyyy.MM.DD HH:mm     ex: 2018.01.12 14:00
```

- Please write a script to check whether the given date and time are formatted correctly. If not, output **Invalid**. Otherwise, simply print it out.
- You don't have to consider the validity of the date and time themselves: the test data won't contain stupid strings such as "`0000/13/61 25:60:60`".

Script requirement

- Name: `format_check.sh`
- Usage: `./format_check.sh [DATE] [TIME]`

Sample Output

```
$ ./format_check.sh 2018-02-28 14:00
2018-02-28 14:00
$ ./format_check.sh 2018.01/01 01:01:12
Invalid
```

```
$ ./format_check.sh 2018/1/1 2:3:4
Invalid
$ ./format_check.sh 2018.01.01 14:10:12:22
Invalid
```

How close are you? (15%)

Problem Description

- Here is a list of Arch Linux mirrors (HTTP(s) servers that host software packages). We are going to find out which server has the least RTT(round-trip time) so as to download packages in the fastest way.
- The RTT can be queried with `ping -c 3 -q $url`.
- Please exclude URLs that cannot be successfully requested, i.e, those resulting in the error: `ping: cannot resolve xxx: Unknown host` or those shown to be `100.0% packet loss`.
- You should list the hostname and their RTT in milliseconds. Output shall be formatted by `sort -g -k 2`.
- **Update**: Please use the **average RTT** from `ping -c 3 -q $url` to sort the output.

Script Requirements

- Name: `rtt_test.sh`
- Usage `./rtt_test.sh [FILE]`
- Input file: https://www.csie.ntu.edu.tw/~chenpowen/mirror.txt

Sample Output

```
$ ./rtt_test.sh mirror.txt
archlinux.cs.nctu.edu.tw 22.625
ftp.tku.edu.tw 26.488
shadow.ind.ntou.edu.tw 28.677
ftp.yzu.edu.tw 36.428
mirror.rackspace.com 43.809
ftp.tsukuba.wide.ad.jp 57.946
mirrors.evowise.com 59.206
ftp.jaist.ac.jp 69.096
mirrors.163.com 90.995
jpn.mirror.pkgbuild.com 96.012
mirror.premi.st 97.494
mirror.lzu.edu.cn 214.793
ftp.kaist.ac.kr 264.158
```

CSIE Analytics (25%)

Problem Description

Website analytics plays an important role in system administration because it helps understand the behavior of website visitors. There are plenty of tools to do this, *Google Analytics* being one of the most popular. You are asked to build a simple tool to do such thing.

Using an one-day log from a CSIE website on our web server, write a shell script to analyze the data in it and find out the popular pages, i.e, those having the most queries.

Log Format

- The log is available here.
- The log is formatted as follow:

```
[client ip] [time] [client request] [status] [length of header]
10.217.44.81 [18/Feb/2019:23:59:59 +0800] "GET /news HTTP/1.1" 200 5039
.
.
```

Script Requirements

- Name: `csie_analytics.sh`
- Usage: `csie_analytics.sh [-n number] [FILE]`
- Input file: `https://www.csie.ntu.edu.tw/~chenpowen/log.txt`
- Description:
  - **csie_analytics.sh** shall extract the **path** of each record and count how many times the path appears in this log.
  - Output the following three fields for each path:
    - **Path**: the path part of client request URL. Note that you have to remove the query part after the "`?`". For example, if the request is `GET /news/news.php?Sn=416 HTTP/1.1`, you have to output `/news/news.php`.
    - **Query times**: the number of times the path is being queried.
    - **Percentage**: $\frac{\text{Query times}}{\text{Total query times}}$, "Total query times" being the total number of queries in the log.
  - `-n number` means printing the top lines of the result. If not specified, **the number is set to 10 by default**
- Erroneous arguments:
  You should also handle wrong arguments, as followed:
  - **Error: option requires an argument.** An non-negative integer shall be provided after the `-n` option.
    Example violation: `bash csie_analytics.sh -n log.txt`

- **Error: line number must be positive integer.**
  Example violation: `bash csie_analytics.sh -n -10.2 log.txt`
- **Error: log file does not exist.**
  Example violation: `bash csie_analytics.sh -n 10 xxx.xxx`
- **Usage: csie_analytics.sh [-n count] [filename]:**
  Other argument errors. Example violation: `bash csie_analytics.sh -q log.txt`
- **Update**: If the filename is not specified, e.g, `bash csie_analytics.sh` or `bash csie_analytics.sh -n 10`, please output "`Usage: csie_analytics.sh [-n count] [filename]`".
- **When we grade your script, we will only trigger one error at a time.**

Hint

- `sort`, `uniq`, `head` are your good friends.
- Use `printf "%-35s %-10s %-2.2f%%\n" $path $time $percentage` to format your output.

Sample Output

```
$ ./csie_analytics.sh log.txt
Path                                Times      Percentage
/news/news.php                      16783      34.34%
/favicon.ico                        8517       17.43%
/inc/pseudo-cron-img.php            4782       9.78%
/rpc_app/rpc_web_use_log.php        4777       9.77%
/calendar/calendar.php              3671       7.51%
/                                   1291       2.64%
/people/bio.php                     1250       2.56%
/admiss/news.php                    1156       2.37%
/app/news.php                       956        1.96%
/intro/news.php                     784        1.60%

$ ./csie_analytics.sh -n 2 log.txt
Path                                Times      Percentage
/news/news.php                      16783      34.34%
/favicon.ico                        8517       17.43%

$ ./csie_analytics.sh
Usage: csie_analytics.sh [-n count] [filename]
```