Homework #1

Due Time: 2022/03/06 (Sun.) 21:59
Contact TAs: `vegetable@csie.ntu.edu.tw`

Instructions and Announcements

- **NO LATE SUBMISSION OR PLAGIARISM IS ALLOWED.**
- 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.
- Some problems below may not have standard solutions. We will give you the points if your answer is followed by reasonable explanations.
- 2/23 updated. Updated the sample testdata in SA/Unix Timestamp Searcher.

Submission

- Please place your answers in the same order as the problem sheet and do not repeat problem descriptions, just organize them by problem number in a tidy manner.
- Please zip all the files, including one PDF and two shell scripts, name the zip file "`{your_student_id}.zip`", and submit it through NTU COOL. The zip file should not contain any other files, and the directory layout should be the same as listed below:

```
{your_student_id}/
+-- {your_student_id}.pdf
+-- {your_student_id}_SA/
    +-- p1.sh
    +-- p2.sh
```

Grading

- NA accounts for 50 points while SA accounts for 50 points. The final score is the sum between them.
- It's possible you don't get full credits even if you have the correct answer. You should show how you get the answers step by step and list the references.
- Tidiness score: 3 bonus points, graded by TA.
- Final score = NA score + SA score + tidiness score.

Network Administration

Wireshark/tcpdump

嚴格禁止 DDoS 我們的服務！違者將會遭受到嚴重的懲罰！

Wireshark 和 tcpdump 都是功能強大的網路封包分析工具，其中 Wireshark 是圖形化介面，而 tcpdump 是終端機介面。在這份作業裡，你需要練習並熟悉這兩個工具。

看個影集也會不小心洩漏密碼 ?! (18%)

請執行以下步驟：

- 用 Wireshark 或 tcpdump 開始擷取網路封包。
- 打開瀏覽器，前往 netflix-http。
- 帳號輸入什麼都可以，密碼請輸入你的學號。(不要真的打你的密碼!)

然後，回答下列問題:

1. 找到那個含有你帳號密碼的封包並附上螢幕截圖。(8%)
2. 重複上列步驟，但這次請前往 netflix-https。請找到那個含有你帳號密碼的封包並附上螢幕截圖，如果找不到的話請說明原因。(10%)

農場危機 (16%)

你的大學好友莉莎出生於全世界最快樂的村莊-快樂村，她的父母擁有整個村最大的農場，從小生長於這樣的環境，使得莉莎很喜歡動物，因此她最喜歡的事，就是在週末時，跟哥哥一起到雞舍與小雞們玩耍、到豬圈看著小豬們在泥濘中開心地打滾。
明天是莉莎的生日，她已經期待一整個禮拜了，然而有點奇怪的是，哥哥一整天都不見蹤影，莉莎在心裡咕噥著：一定是忘記幫我準備禮物，今天才趕著去買吧！然而到了傍晚他們約定一起玩牌的時間，哥哥卻始終沒有出現，這時候莉莎便開始覺得不太對勁，於是決定到農場找找看。
她繞了整個農場三次，卻始終沒有看到一絲哥哥的蹤影，正當焦急如焚的莉莎準備跑回家跟爸媽說時，突然憋見豬群的中央有一個從未看過的小木盒，她好奇地走近，發現盒子上有一個小紙條，寫著：「哇哈哈哈，你哥哥就是太不小心才會掉進我設下的陷阱，要找回你哥哥就必須展現你的聰明與勇氣！如果能解開盒子裡的線索，得到一組密碼，你明天就能在生日派對上看到他為你唱著生日快樂歌，如果不行的話....」，莉莎趕緊打開盒子，只見裡面有：

- pigstory.pcapng
- Token page

作為莉莎最好的朋友，你決定要坐夜車火速趕往快樂村幫忙她解出線索，能不能讓莉莎的哥哥平安的回來，就端看你的聰明才智了。請回答下列問題，並詳細說明你是如何得到答案的，若你找不到密碼，仍然可以寫下你的想法，我們會視情況給予部份分數。

1. 你是怎麼找出奇怪的圖片？(Hint: 不一樣最奇怪！)(8%)
2. 密碼是什麼？NASA... (Hint: 就說沒有要為難莉莎，線索都在網站留下來了)(8%)

這麼多的網路協定要是能全部都認識的話該有多好 (16%)

請用 Wireshark 或 tcpdump 擷取下列的封包。在以下的每題當中，請附上封包的螢幕截圖、簡答該協定的用途，並說明該協定運作在 TCP/IP 五層網路架構中的哪些層。

1. ICMP request & reply (2%)
2. DNS query & response (2%)
3. ARP request & reply (4%)
4. DHCP discover & Offer & Request & Ack (8%)

System Administration

Permission (20%)

I. Basic (10%)

以下是資料夾 `myfolder` 下的檔案架構與其個別權限，命名前綴為 `dir` 者為資料夾，命名前綴為 `file` 者為一般檔案：

```
[r-x]  .
[--x]  ./dir1
[r--]  ./dir1/dir2
[rw-]  ./dir1/dir2/fileA
[rwx]  ./dir1/dir3
[r--]  ./dir1/dir3/fileB
[rw-]  ./dir1/fileC
[r-x]  ./dir4
[rw-]  ./dir4/fileD
```

回答是否可成功在資料夾 `myfolder` 下進行以下每小題的動作。除了須回答 true 或 false 外，也須簡單說明理由。若無理由或理由錯誤者該小題不給分。

1. `ls dir1`
2. `ls dir1/dir2`
3. `ls dir1/dir3`
4. `cd dir1`
5. `cd dir1/dir2`
6. 修改 `dir1/dir2/fileA`
7. 修改 `dir1/dir3/fileB`
8. 修改 `dir1/fileC`
9. 刪除 `dir1/dir3/fileB`
10. 刪除 `dir4/fileD`

II. ACL (10%)

請先進入工作站 linux7 ，在 `/tmp2/NASA-HW1/` 下建立自己學號的資料夾，英文字母部分需使用小寫，下面題述會以 `<student_ID>` 代稱。以下每個小題若須使用指令操作，除了須在 linux7 上設定外，也須在 report.pdf 寫下設定的指令，若無則對應的小題不給分。

1. 觀察資料夾 `NASA-HW1` 的設定，為何你不可以刪除別人在 `NASA-HW1` 下建立的資料夾與檔案？(1%)
2. 請將資料夾 `<student_ID>` 設定為只有自己與 group `ta` 的人可檢視其內容 (`ls`)。(1%)
3. 請在資料夾 `<student_ID>` 下新增一個資料夾 `chatroom` 來模擬聊天室，並設定成只有你與你的好朋友能進入。(3%)

(a) 請任意選擇一個存在的工作站帳號當作好朋友來設定下面的題目，建議使用非 group `ta` 的帳號，並且在 report.pdf 中說明。(0%，但沒說明就無法取得第 3 題的分數)

(b) 設定相關權限，只有自己與好朋友可以檢視資料夾 `chatroom` 的內容 (`ls`) ，以及新增與刪除檔案。(1%)
Hint: 此題可能會需要設定資料夾 `<student_ID>`。

(c) 這個聊天室的設計為一個檔案代表一個主題，因此你與你的好朋友們應該都要可以編輯資料夾 `chatroom` 下的所有檔案，才能讓雙方針對同一個主題討論。請設定資料夾 `chatroom` ，使之後新增的檔案都可以被雙方編輯，不需一個一個檔案設定。(1%)

(d) 過了一段時間，你發現你們只會在其中一個主題下聊天，因此想將新增主題的功能暫時關閉，也就是將你與好朋友對於資料夾 `chatroom` 的權限設為只能檢視內容，但目前已有的檔案仍可編輯。請對資料夾 `chatroom` 使用 ACL 的 mask 來設定。(1%)

4. 最近很流行猜字遊戲 Wordle ，身為程式高手的你也想自己寫寫看。請在資料夾 `<student_ID>` 下建立資料夾 `wordle` 。(5%)

(a) 因為遊戲還有一些 bug ，但 TA 自告奮勇說可以幫你找出 bug 。請在 `wordle` 資料夾內建立檔案 `game.sh` ，並幫 group `ta` 加上可讀寫的權限讓 TA 幫你修改，但 TA 不可在 wordle 資料夾內新增或刪除其它檔案。(1%)

(b) 雖然你的遊戲尚未完成，但遊戲用的單字表已經建立好了。請在 `wordle` 資料夾內建立單字表檔案 `wordlist.txt` ，並將權限設為只有自己可讀。(1%)

(c) 雖然上一個小題的 `wordlist.txt` 只有自己可讀，但未來若要開放讓大家玩遊戲，仍需要透過程式讀取單字表。有個叫做 `setuid` bit 的設計可以幫助你完成這個任務，不過由於一些安全性的問題，目前 `setuid` 在 Linux 中無法使用於 shell script 。請簡述 `setuid` 的功用，並舉一個例子說明若 shell script 可使用 `setuid` 設定可能會有什麼漏洞。(3%)

以上提及的檔案內容隨意，但如果你想真的自己寫出 Wordle ，TA 也不反對 XD

Shell Scripting (30%)

注意事項

1. 請在 script 前加上 *shebang* 並開啟 script 的執行權限 (`chmod +x`)。
2. 你的 script 必須能在以下幾種 shell 中之一執行，`bash`、`sh`、`zsh`、`fish`、`tcsh` 或 `ksh`。
3. 我們會在工作站測試你的 script，請確認所有使用到的指令在工作站上皆可以正常運作。
4. 此部分需全部使用 shell script 完成，不可在中途執行自己的 python 腳本或使用其他程式語言，如果不確定是否可以使用某些指令，請寄信詢問助教。
5. 請勿在 shell script 內新增檔案，也請勿夾雜惡意程式碼，並在一分鐘內成功執行。

I. Unix Timestamp Is Interesting (15 %)

1. Argument Parser (5%)

這個小題只需要檢查參數是否有誤，請按照以下的內容讀取相對應的參數，不會有重複的輸入檔的情況。

```
usage: ./p1.sh -s <Unix timestamp> -e <Unix timestamp> <input file>
ex: ./p1.sh -s 1133642864 -e 1133643505 log1.txt log3.txt
```

指令內容

1. `-s` 會讀取起始 timestamp，不能為空，並且一定要有該參數
2. `-e` 會讀取結束 timestamp，不能為空，並且一定要有該參數
3. 必須至少要有一個檔案
4. 起始時間要小於或等於結束時間 (合法的區間)
5. Timestamp 皆為 Unix timestamp 並以秒為單位
6. 若上述有一項沒有滿足，或是有不合法的參數，則必須輸出上述方框內的使用說明

範例合法參數

```
./p1.sh -s 1133642864 -e 1133643505 log1.txt log3.txt
./p1.sh -e 1133643505 -s 1133642864 log2.txt
```

範例非法參數

```
./p1.sh -s 100 -e 0 log1.txt log3.txt    # start time > end time
./p1.sh -s -e 100 log1.txt log2.txt      # no start time
./p1.sh -s 0 -e 113643505                # no input file
./p1.sh -x -s 0 -e 113643505 log1.txt    # invalid flag -x
```

2. Unix Timestamp Searcher (10%)

有多個沒有依照時間排序的檔案 log files，請延續 1. 寫的 `p1.sh` 並透過上述的使用說明讀取多個檔案（一至三個），以及兩個參數 `-s`, `-e`。搜尋包含這兩個 timestamp 區間的所有 log，並且按照時間先後回傳，若有相同時間者，則須按照字典序排序，並輸出以下格式：

```
<Unix timestamp> <[notice]|[error]> <description>
<Unix timestamp> <[notice]|[error]> <description>
...
```

範例輸出

```
./p1.sh -s 1133642864 -e 1133643505 log1.txt log2.txt log3.txt
1133642864 [error] mod_jk child workerEnv in error state 6
1133642864 [notice] workerEnv.init() ok /etc/httpd/conf/workers2.properties
1133643068 [notice] jk2_init() Found child 6725 in scoreboard slot 10
....
1133643469 [notice] jk2_init() Found child 8541 in scoreboard slot 9
1133643491 [notice] workerEnv.init() ok /etc/httpd/conf/workers2.properties
1133643498 [error] mod_jk child workerEnv in error state 6
```

II. Directory Tree Visualizer (15%)

這題我們要製作一個能遞迴顯示資料夾內容的工具，範例資料夾為以下範例中所使用的輸入。

1. Basic (5%)

```
usage: ./p2.sh [<start path>] [-l <layer>]
```

- `<start path>` 為開始遞迴的起始路徑，如果為空，則從目前的路徑開始
- `-l <layer>` 代表遞迴的層數，預設為無限大，0 則代表只印出起始路徑資料夾下的內容

將資料夾與檔案按照以下格式遞迴印出

- 每個檔案/資料夾各占一行
- 在輸出的第一行印出 `<start path>`，如果 `<start path>` 為空則印出一個點 "." 代表當前目錄
- 除了第一行，在每一行的開頭先印出一個字元 "|"、3n 個 "-" 字元與一個空白，再印出檔案/資料夾的名稱，n 為遞迴的層數
- 同一層的檔案/資料夾以字典序印出

範例

```
./p2.sh dir1 -l 2
dir1
| file
| src
|--- dirA
|------ somefile
|--- dirB
|------ dirC
```

補充事項

- 列出檔案只能使用 `ls`，禁止使用 `tree` 或類似功能的指令，若不確定指令是否可以使用，請先詢問助教
- 檔案名稱只含有 `[0-9a-zA-Z]` 、"." 、"-" 和 "_" 字元
- 不須處理不合法的指令或起始路徑
- 本小題不須印出隱藏的檔案/資料夾 (與 `ls` 印出的檔案相同)
- 本小題中檔案中不包含 symlink 或 hard link

2. More Parameters (5%)

基於前一個小題的結果，在這個小題中新增一些新的參數與功能

```
usage: ./p2.sh [<start path>] [-l <layer> -a -r -i <string>]
```

- `-a` 印出隱藏的檔案，start path 以外的部分不需印出 "." 與 ".."
- `-r` 逆序印出同一層的檔案/資料夾，也就是字典序從大到小

- `-i <string>` 只印出檔案名稱含有 `<string>` 的檔案/資料夾與遞迴經過的路徑
- 不同參數可能會調換順序

範例

```
./p2.sh dir2 -l 1 -a -r
dir2
| assets
|--- images
|--- audios
| .hidden
|--- .hidden_file
```

```
./p2.sh dir2 -l 2 -i .png
dir2
| assets
|--- images
|------ cat.png
```

3. Handle symlinks and loops (5%)

在正常的情況下，我們已經能夠順利地印出資料夾架構了，然而，如果碰到檔案中含有 symlink 的話，程式可能會發生無限迴圈，也就是遞迴進入已經遞迴過的資料夾，因此在這個小題中我們需要預防無限迴圈的發生

```
usage: ./p2.sh [<start path>] [-l <layer> -a -r -i <string> -s]
```

- `-s` 不遞迴進入所有 symlink 連結到的資料夾，並且在所有 symlink 檔案名稱後多印出 `" -> <link path>"`，`<link path>` 為 symlink 檔案中儲存的路徑
- 當 `-s` 沒有設置且發生迴圈時，在發生迴圈的檔案名稱後面多印出 `" (loop)"`，並且不要遞迴進入

範例

```
./p2.sh dir3 -s
dir3
| dirA -> ../dir4
```

```
./p2.sh dir3
dir3
| dirA
|--- dirB
|------ dirC (loop)
```

補充事項

- 檔案中不包含 hard link
- symlink 不會連結到測資外面的路徑

- 同一個絕對路徑可能有多種表示方法，在比對路徑時需注意
- `-s` 印出的 `<link path>` 為 symlink 檔案中儲存的路徑，也就是建立 symlink 時指令所輸入的路徑