Exercise 2

Abd El Rahaman Shehata

Exercise 2

2.1 Explain the meaning and syntactic representation of the third and fifth fields of the /etc/shadow file. With what shell program can these fields be altered (without manually editing the shadow file)? Give an example command which changes the fifth field.

Source: https://www.cyberciti.biz/faq/understanding-etcshadow-file/

- Third Field

  The date of the last password chage

- Fifth Field

  The maximum number of days before the password has to be changed.

- With what shell program can these fields be altered?

  `chage`

- Give an example command which changes the fifth field.

  `chage <username> -M 60`

2.2 Name functional differences between the following two command-lines:

- `sh     cat < /etc/passwd | grep $USER | cut -d':' -f1 > /tmp/username`

  Gets the current user and writes the result in a file in the tmp directory.

- `sh     grep $(whoami) /etc/passwd | awk 'BEGIN { FS = ":" }; { print $1 }'` Gets the current user and prints the name out.

This doesn't work on macos because users are stored in a seperate db. (Not POSIX compliant)

2.3 You need to mount an USB thumbdrive whose device file is /dev/sdd1 to ~/usb with the following constraints:

- The file-access timestamp of all file-system objects located on the drive should never be altered.
- Due to security reasons, executing prog
- Disallow setting of SUID or SGID bits.

```
mount /dev/ssd1 -o noatime, nosuid, noexec ~/usb
```

2.4 Find a way to establish a disk usage quota on per-user basis (i.e., as administrator, you define that a particular user bob must not use more than e.g. 3 gigabytes of storage).

---

quota-tools need to be installed

1. Edit /etc/fstab `/dev/sda3 /home ext4 defaults,usrquota 0 2`

2.
   ```
   sudo mount -vo remount /home
   ```

3.
   ```
   sudo quotacheck -cum /home
   ```

4.
   ```
   sudo quotaon -v /home
   ```

5.
   ```
   sudo edquota abdo 5GB 6GB
   ```

2.5 What exactly does the following command do?

```
time find / -type d >/dev/null 2>&1
```

- `time`: shows the amount of CPU seconds a command spends in user and kernel space and how much cpu was used and the total found items
- `find / -type d`: Finds all directories starting from the root dir
- `>/dev/null`: changes the output stream of `time find / -type d` from stdout to `>/dev/null`
- `2>&1`: Joins the stderr to the same streams as stdout

2.6 Find an alternative but similar command-line for `touch filename`

```
time echo $null >> filename
```

2.7 Install ffmpeg. Create a command that downloads this video file, and if successful, automatically transcode it to the HEVC/h265 codec using the following command:

```
ffmpeg \ -i
 ↪  'https://filesamples.com/samples/video/m2v/sample_1920x1080.m2v'
 ↪  -c:v libx265 -crf 26 \ -preset fast -c:a aac -b:a 128k
 ↪  sample_1920x1080.mov
```

2.7 What are pushd and popd commands useful for?

For switching directories using a navigation stack like Web-Browsers or Gui-Applications but for the terminal.

2.8 Use the find command to search for...

(a) all directories in /usr/ up to a maximum depth of 3, and store the output in a textfile in your home directory.

```
find /usr -type d -maxdepth 3 > ~/usr_dirs.txt
```

(b) all files on your system without actual content (i.e. zero bytes in size).

```
find / -type f -empty
```

(c) all executable files on the entire system for which either SUID or GUID is set, while suppressing any error messages on the console.

```
sudo find / -type f -perm /6111 2>/dev/null
```

2.9 Copy /usr/bin/ls to your home directory and change its permissions using octal noctation to reflect the following settings and describe what each permission settings means in detail, and give meaningful examples of files (or classes of file) for which the respective permissions do make sense.

- (a) `-rw-r--r--` : 644

  ```
  cp /usr/bin/ls ~ && chmod 644 ~/ls
  ```

  Owner can read, write everyone else can only read.

  Example usage: documentation
- (b) `-rwxr-xr-x` : 755

  ```
  cp /usr/bin/ls ~ && chmod 755 ~/ls
  ```

  Owner can read, write and exec everyone else can only read and exec.

  Example usage: apps
- (c) `-r-xr-s---` : 2550

  ```
  cp /usr/bin/ls ~ && chmod 2550 ~/ls
  ```

  Owner can read, exec. The group can read and always exec as a member of it's owners group.

  Example usage: shared workspace, every subdir and files inherit the setguid so a switch is not needed
- (d) `-r-sr-x---` :

  ```
  cp /usr/bin/ls ~ && chmod 4550 ~/ls
  ```

  Owner can read, exec. The group can read and exec as if they were the owner.

  Example usage: admin tools, like passwd

2.10 Explain the meaning of the Unix signals `SIGHUP`, `SIGCONT`, `SIGALRM`, `SIGSEGV`, `SIGUSR2`.

- `SIGHUP`: A signal sent to the process when its controlling terminal is closed
- `SIGCONT`: A signal sent to the process, which causes the operating system to resume its execution after being halted by `SIGSTOP`
- `SIGALRM`: A signal to the process after a specified amount of time elapsed by calling `alarm(time)`
- `SIGSEGV`: A signal to the process that indicates memory access violation (segmentation fault)
- `SIGUSR2`: A signal that is set aside for you to use any way you want. They're useful for simple interprocess communication, if you write a signal handler for them in the program that receives the signal. The default action is to terminate the process.

2.11 Use the find and sha512sum commands in order to create a command-line which calculates the hash sum of each executable file on your system.

```
find / -type f -perm +111 | xargs -n1 sha512sum
```

2.12 You need to generate a random secret-key for symmetric encryption.

- a) To do this, read x bytes from the system entropy device file and create a SHA-512 hash from it.

  ```
  $ dd if=/dev/urandom bs=1 count=10 | sha512sum

  82fa229239cbc4860bcc2f7b1f5b38b642318c7a8dd3a49b5da24a4d82c
  72221f688a6c1d630202065b33468b4613daccd1597d425b176c927b307e316dc53f8
  ```

- b) How many bytes x must at least be read for the available entropy to fully exploit the value domain of the specified hash function?

  64 bytes

- c) Why is the hash output significantly longer than x?

  Because SHA-512 always creates a hash that's 512 bytes long.

2.13 - Get to know regular expressions and understand how to effectively use them. In order to do so, visit https://regexone.com. Make sure to successfully accomplish all exercises (consisting of 15 tutorial lessions and 8 problems).

---

1. `.*`
   1. `.*\d.*`
2. `.\*\.`
3. `[fmc].*`
4. `[^b]`
5. `[A-Z]\w\w`
6. `.*z{2,}.*`
7. `aa+.*`
8. `\d+.*\?`
9. `\s.*`
10. `^Mission.*`

11. `^(file.*)\.pdf$`
12. `^(\w+.*(\d{4}))$`
13. `^(\d+).(\d+)$`
14. `^.*(cat|dog)s$`
15. `^.*$`

2.14 - Download https://delta-xi.net/sms/sample_access_log.txt, which resembles a sample log-file from the Apache web-server, and construct the correct command lines to answer the following questions (one-liners only):

- How many distinct IP source addresses or hostnames are contained in the logfile?

```
$ cat sample_access_log.txt | cut -d '-' -f1 | sort -u | wc -l
```

```
155
```

- How many HTTP requests other than GET requests have reached the web-server on 08.03.2004 between 20:00 and 23:59?

```
$ cat sample_access_log.txt | egrep -v 'GET' | egrep -E
 ↪  '08/Mar/2004:(2[(0|1|2|3)]:[0-59])'
```

```
64.246.94.152 - - [08/Mar/2004:20:09:57 -0800] "HEAD
 ↪  /twiki/bin/view/Main/SpamAssassinDeleting HTTP/1.1" 200 0
```

- What size was the largest web-server response answer of all non-GET requests?

```
$ cat sample_access_log.txt | egrep -v 'GET' | awk '{print $NF}'
 ↪  | sort -rn | head -1
```

```
24577
```

- How many clients requested the robots.txt file using HTTP version 1.0, whose source host does not originate from a .com domain?

```
$ cat sample_access_log.txt | egrep 'robots.txt HTTP/1.0' | egrep
 ↪  -v '.com' | wc -l
```

```
0
```

- How many HTTP Not Modified responses have been issued?

```
$ cat sample_access_log.txt | awk '{print $((NF - 1))}' | grep
 ↪  '304' | wc -l
```

```
137
```

- How many different distinct HTTP status codes except for 200 (OK) and 404 (Not Found) have been issued?

```
$ cat sample_access_log.txt | awk '{print $((NF - 1))}' | egrep
 ↪  -v  '(200|404)' | wc -l
```

```
267
```

- Explain the following command in detail:

  ```
  sed -E 's/^([^ ]+?).*([0-9]{3}) .*$/\1 \2/'
   ↪  sample_access_log.txt
  ```

Explanation:

The Command graps all the domains/ip-addresses and the reponse code of the request.

`^([^ ]+?).*([0-9]{3}).*$`: old value

`\1 \2`: new value

- `^`: matches start of line
- `([^ ]+?)`: create a group and match everthing up to the first space (lazyly)
- `.*`: match everthing that's not a newline
- `([0-9]{3})`: create a group and match a number between 1 and 9 three times
- `.*`: match everthing that's not a newline
- `$`: matches start of line

2.15 Create a single command-line which pretty-prints the current week's lunch menu of Campina on your shell from https://www.mittag.at/w/campina/. The result should look like this:

```
Montag, 30.05.
    Kasekrainer mit Pommes €7,50
    Cremespinat mit Kartoffelschmarrn und Bio Spiegelei €6,90
Dienstag, 31.05.
    Asia Wok mit Huhnerfleisch dazu Reis €7,50
    Nudel-Gemuseauflauf auf Petersiliensauce €6,90
Mittwoch, 01.06.
    Putengeschnetzeltes in Spargelrahm mit Kroketten €7,50
    Ofenkartoffel mit Raucherlachs und Gemuse oder Vegetarisch €6,90
Donnerstag, 02.06.
    Rinderbraten mit Semmelknodel und Speckbohnen €7,50
    Gebackenen Camembert mit Petersilkartoffel €6,90
```

```
Freitag, 03.06.
    Gebackenes Schollenfilet mit Reis und Kartoffel €7,50
    Chili con Carne oder Chili sin Carne Vegan mit Geback€ 6,90
```

```
curl https://www.mittag.at/w/campina | html2text | perl -0777 -ne
 ↪  'print "$_\n" for $_ =~
 ↪  /^.*((?:(?:Mon|Diens|Donners|Frei)tag|Mittwoch).*\.[\S\s]*?)(?=Dessert)/gm;'
 ↪  | egrep -v '([Mm]enü|Tagessuppe|Öffnungszeiten)' | sed
 ↪  '/^$/d' | grep -v '^ *$' | sed 'N;s/\n\s*€/ €/'
```