ECE 404 Homework #11

Due: Thursday 4/23/2020 at 4:29PM

Spam Filter

Design spam filter recipes that will trap all 75 messages that you'll find in the gzipped tar archive **junkMail.tar.gz** found in the homework section of the ECE 404 website. When you gunzip and untar the archive with, say,

```
tar -zxvf junkMail.tar.gz
```

you'll see 75 individual spam messages with names **junkMail_1** through **junkMail_75**. About these messages:

1. **junkMail 1 through junkMail 50** : The headers of all these messages have one thing in common: they contain multiple entries in the "From:" header. All these messages were trapped by a single recipe in your instructor's spam filter. The regex in your instructor's recipe has only 40 characters in it. (If the regex engine used by procmail allowed for Perl's '{}' metacharacters, this regex could have been made as short as just 10 characters.)

2. **junkMail 51 through junkMail 63** : These messages can be trapped just on the basis of the "Subject:" line in the email headers.

3. **junkMail 64 through junkMail 66** : In your instructor's spam filter, these messages were trapped on basis of the content (email body) of the messages.

4. **junkMail 67 through junkMail 75** : You should trap these with a single recipe that contains compound rules. Below is an example of a recipe with compound rules. It is NOT the compound recipe for trapping the messages junkMail 67 through junkMail 75:

```
:0 HB:
* ^Content-Type: text/plain
* !^Content-Type: text/html
* !^content-type: application/pdf
* !^content-type: application/zip
* !^content-type: application/msword
* !^content-type: application/.*signature
* Content-Transfer-Encoding: base64
junkMailCompound6
```

   This recipe says that if the "Content-Type" MIME header is text/plain and none of the MIME objects are of type PDF, ZIP, etc., and yet the "Content- Transfer-Encoding" MIME header calls for Base64 encoding, then there is a great chance it is a spam message.

After you have incorporated the new recipes in your **.procmailrc** file, you can test your filter on an individual message by invoking the command:

```
procmail .procmailrc < junkMail_XX
```

where "XX" is the integer suffix for the message file. Obviously, you would need to write either a shell script, or a Python script, or a Perl script to execute the above command in a loop for all 75 spam messages.

It is advised that you do this via command-line SSH (on shay.ecn.purdue.edu) instead of via ThinLinc. In the past students have had trouble using procmail when logging in via ThinLinc, particularly when using the procmail command.

If your recipes work on all 75 messages that have been sent to you, you will not see any messages being subject to the default action of your procmail filter, which is usually to put the surviving messages in your mailbox **/var/mail/account_name** (this can be viewed with the mailx command). Of course, you should test your recipes with your own messages that shouldn't be marked as spam. This way, you can ensure that your recipes allow desired messages through.

Since the spam message in the tar archive are in their raw form, it is sometimes hard to see what is in them − especially if the MIME objects in the message are Base64 encoded. To help you decipher those spam messages that are fully or partially encoded, you can use your instructor's Perl script **EmailParser2.pl** that is on the homework section of the website . Execute this script (you may need to modify the shebang line based on where perl is installed for you) and give it a command-line argument that is the name of the junk mail file you want to decipher. It will deposit the different MIME objects in the email in a subdirectory called **mimemail** in the directory in which you execute the script.

Submission Requirements

- Your submission should include the .procmailrc file you used to filter the junk mail. The file should meet the specifications described below.
- Include comments in your .procmailrc file explaining your recipes.
- Before your recipes in your .procmailrc file, put comments with your homework header (the homework header format is described in the homework section of the ECE 404 website).
- When submitting your homework electronically, you can use the `cp` command to copy your .procmailrc file to `yourPurdueUsername_dot_procmailrc` (where `yourPurdueUsername` is your normal ECN account username), and submit it using the `turnin` command below.

Electronic Turn-in

```
turnin -c ece404 -p hw11 yourPurdueUsername_dot_procmailrc
```