AVIK RANJAN BHATTARAI
ALL THE CODES AND NECESSARY FILE CONTENTS

THE MAIN SCRIPT – checkgetopts

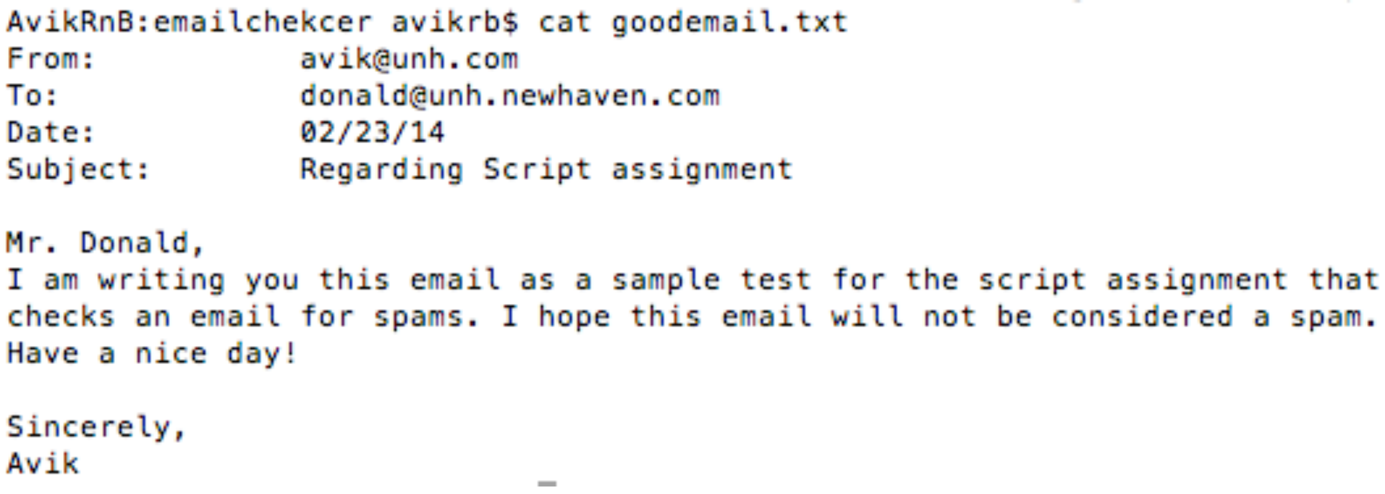
```
AvikRnB:emailchekcer avikrb$ cat checkgetopts
#!/bin/bash

## ======= DONE BY: AVIK RANJAN BHATTARAI ========
## ======= DATE OF SUBMISSION: 03-08-2014 =======
## ====== SCRIPT TO CHECK AN EMAIL FOR SPAM ======

clear
echo "*************************"
echo "WELCOME TO EMAIL SPAM CHECKER"
echo
echo

##I PREFER WORKING WITH FUNCTIONS SO I CREATED FUNCTION FOR EVERY KIND OF CHECK

##THE "FROM" FIELD CHECK OR THE SENDER CHECK
function sendercheck
{
    sed -n /From:/p $OPTARG | cut -f3 > tempfile.txt
    ##ABOVE sed STATEMENT PRINTS ONLY THE LINE WITH "From:" included AND ITS
OUTPUT
    ##IS FURTHER WORKED ON BY THE CUT STATEMENT WHERE -f3 CUTS ONLY THE EMAIL
ID
    ##PORTION AND IS STORED IN A FILE CALLED tempfile.txt

    echo "======DOING THE SENDER CHECK======"
    echo "Please wait...."
    sleep 1.0
    ##THE ABOVE ECHO AND SLEEP COMMANDS ARE ONLY USED FOR A MORE REALISTIC
OUTPUT
    ##VIEW AND HAVE NO EFFECTS ON SCANNING OF THE SPAM IN EMAIL


    if grep --quiet  -F -f tempfile.txt bademailid.txt
        then echo "!!! ======= WATCH OUT SPAM DETECTED IN THE FROM
FIELD======== !!!"
        else echo "====== CONGRATS..!!! EMAIL ID IS NOT A SPAM ======"
    fi
    ##IF GREP FINDS A MATCH THEN IT DOES NOT PRINT OUT, INSTEAD THE "THEN"
STATEMENT
    ## IS EXECUTED. IF GREP DOES NOT FIND A MATCH THEN THE ELSE STATEMENT IS
EXECUTED
    ##FILE bademailid.txt CONTAINS A LIST OF BADEMAIL IDs
    echo "____________________"
}


##THE "SUBJECT" FIELD CHECK
```

```
function subjectcheck
{

    sed -n /Subject:/p $OPTARG | cut -f2 >  tempfile.txt
    ##ABOVE sed STATEMENT PRINTS ONLY THE LINE WITH "Subject:" included AND ITS
OUTPUT
        ##IS FURTHER WORKED ON BY THE CUT STATEMENT WHERE -f2 CUTS ONLY THE
MAIN SUBJECT
        ##PORTION AND IS STORED IN A FILE CALLED tempfile.txt

    echo "======DOING THE SUBJECT CHECK======"
    echo "Please wait...."
    sleep 1.0
    ##THE ABOVE ECHO AND SLEEP COMMANDS ARE ONLY USED FOR A MORE REALISTIC
OUTPUT
        ##VIEW AND HAVE NO EFFECTS ON SCANNING OF THE SPAM IN EMAIL

    for word in $(<tempfile.txt)
    do
        if grep --quiet  -F -i "$word" badsubject.txt
        then echo "!!! ======= WATCH OUT SPAM DETECTED IN SUBJECT======== !!!"
        exit
        fi
    done
    ##I HAVE USED A FOR LOOP TO CHOOSE SINGLE WORD OF THE SUBJECT FIELD AT A
TIME
    ## THE GREP SCAN EVERY WORD IN THE SUBJECT FIELD IF  ANY
    ##WORD MATCHES THE LIST OF WORDS IN badsubject.txt FILE THEN IT IS
CONSIDERED BAD.
    ##IF NO WORDS MATCH THEN THE ECHO STATEMENT BELOW IS PRINTED OUT

    echo "CONGRATS...!!! NO SPAM FOUND IN THE SUBJECT"
}


##THE MAIN BODY OF THE EMAIL MESSAGE CHECK
function bodycheck
{

    echo "Checking the Body of the email for Spam"
    echo " Please wait ...."
    sleep 1.0
    ##THE ABOVE ECHO AND SLEEP COMMANDS ARE ONLY USED FOR A MORE REALISTIC
OUTPUT
        ##VIEW AND HAVE NO EFFECTS ON SCANNING OF THE SPAM IN EMAIL


    touch thresholdcount.txt
    ##THE TOUCH CREATES A FILE called thresholdcount.txt
    ##THE REASON BEHIND CREATING THIS FILE WILL BE MENTIONED IN STATEMENTS
```

```
BELOW

    egrep -v "From:" $OPTARG | egrep -v "To:" | egrep -v "Date:" | egrep -v
"Subject:" > tempfile.txt
    ##ANY LINE IN THE EMAIL THAT DOES NOT INCLUDE THE ABOVE SPECIFIED WORDS ARE
COPIED INTO THE tempfile.txt FILE.
    ##IN SHORT, THE MAIN BODY OF THE EMAIL MESSAGE IS COPIED INTO THE
tempfile.txt FILE

    ##AGAIN I AM MAKING USE OF FOR LOOP TO EXTRACT SINGLE WORD FROM THE
tempfile.txt FILE
    ##SO THAT I COULD USE TO SCAN IT
    for word in $(<tempfile.txt)

            do

                  if grep --quiet -woi "$word" badsubject.txt
               ##IF THE SINGLE WORD FROM FOR LOOP MATCHES THE WORDS FROM
badsubject.txt FILE
        ##THEN THE "then" STATEMENT IS EXECUTED, ELSE THE LOOP CONTINUES
        ##I USED badsubject.txt FILE FOR THE BODY CHECK AS WELL BECAUSE WE JUST
WANTED A LIST OF SPAM WORDS

             then

             grep -woi "$word" badsubject.txt >> thresholdcount.txt
        ##IF ANY KEYWORDS MATCH THEN THE WORD IS APPENDED TO THE FILE CALLED
thresholdcount.txt
           fi
             done

    uniq -c thresholdcount.txt | cut -d ' ' -f4 > tempfile.txt
    ##NOW THE thresholcount.txt FILE HAS ALL THE WORDS THAT MATCHED THE
KEYWORDS IN badsubject.txt FILE
    ##DOING A UNIQ -C COMMAND ON THE FILE GIVES US THE NUMBER OF COUNTS FOR
EACH WORD AND I FURTHER EXTRACT
    ## THE FIELD WHERE THE NUMBERS ARE AND SENT THEM TO tempfile.txt. THE
INITIAL CONTENTS ARE tempfile.txt
    ##ARE ERASED DURING THE PROCESS



    for word in $(<tempfile.txt)
    ##AGAIN, tempfile.txt CONTAINS A ROW OF NUMBERS ONLY AND I EXTRACT THEM
USING FOR LOOP TO CHECK IF ANY OF THEM
    ##ARE HIGHER THAN THE THRESHOLD WHICH IS 3 IN THIS CASE
    do
        if [ $word -ge 3 ]
        then
            echo "!!! ======= WATCH OUT SPAM DETECTED IN THE EMAIL
BODY======== !!!"
```

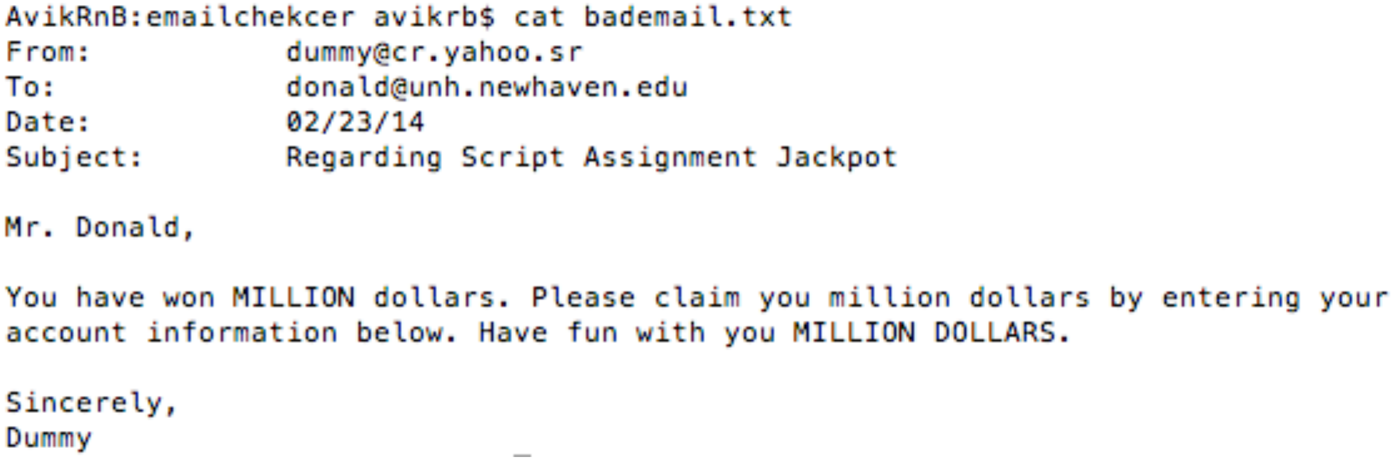
```
            rm thresholdcount.txt
            ##HERE THE SCRIPT REMOVES THRESHOLDCOUNT.TXT FILE SO IN SHORT IT
LOOKS AS IF IT DID NOT EVEN EXIST
            exit
        else
            continue
        fi
    done
    echo "==== NO SPAM DETECTED IN EMAIL BODY ==== "
    rm thresholdcount.txt
    ##IF NO SPAM IS FOUND THEN WE STILL NEED TO REMOVE THE thresholdcount.txt
file

}



echo

##GETTING COMMAND LINE ARGUMENTS WITH GETOPTS
##"OPTION" STORES THE USER SELECTED OPTION AND "OPTARG" IS THE EMAIL FILE GIVEN
TO IT
while getopts "f:s:b:"   OPTION
do
##STATEMENT BELOW CHECKS THE FILE FOR EXISTENCE OR IN SHORT VALIDATION OF THE
FILE
if [ –e $OPTARG ]
    then
    case $OPTION in
    f)
            echo "–F was triggered!"
            sendercheck
        ##IF USER SELECT "–f" OPTION THEN A SENDER CHECK IS DONE
    ;;
    s)
                echo "–S was triggered!"
            subjectcheck
        ##IF USER SELECTS "–s" THEN A SUBJECT FIELD CHECK IS DONE
    ;;
    b)
                echo "–B was triggered!"
        bodycheck
        ##IF USER SELECTS "–b" THEN AN EMAIL BODY CHECK IS DONE
    ;;

    *)
        ##ANY OTHER OPTIONS OTHER THAN "f,s or b" IS CONSIDERED INVALID
            echo "ERROR ========= You have entered an Invalid option========"
        ;;
    esac
    else
    ##IF GIVEN FILE DOES NOT EXIST THEN AN ERROR MESSAGE IS SHOWN
    echo "ERROR ==== INVALID FILENAME ===="
fi
done
shift $(( OPTIND -1));

echo "*************************"
```

THE GOOD EMAIL FILE – goodemail.txt



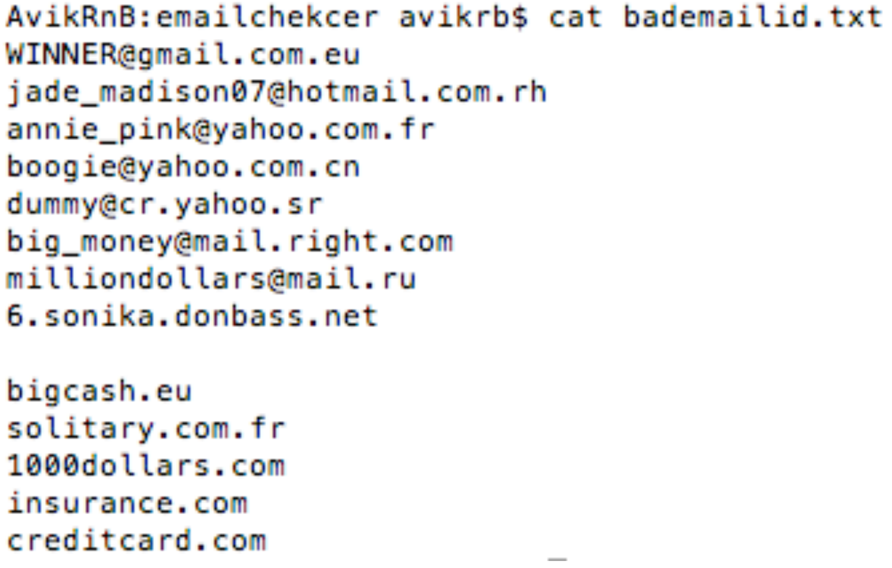
```
AvikRnB:emailchekcer avikrb$ cat goodemail.txt
From:           avik@unh.com
To:             donald@unh.newhaven.com
Date:           02/23/14
Subject:        Regarding Script assignment

Mr. Donald,
I am writing you this email as a sample test for the script assignment that
checks an email for spams. I hope this email will not be considered a spam.
Have a nice day!

Sincerely,
Avik
```

THE SPAM EMAIL FILE – bademail.txt



```
AvikRnB:emailchekcer avikrb$ cat bademail.txt
From:           dummy@cr.yahoo.sr
To:             donald@unh.newhaven.edu
Date:           02/23/14
Subject:        Regarding Script Assignment Jackpot

Mr. Donald,

You have won MILLION dollars. Please claim you million dollars by entering your
account information below. Have fun with you MILLION DOLLARS.

Sincerely,
Dummy
```

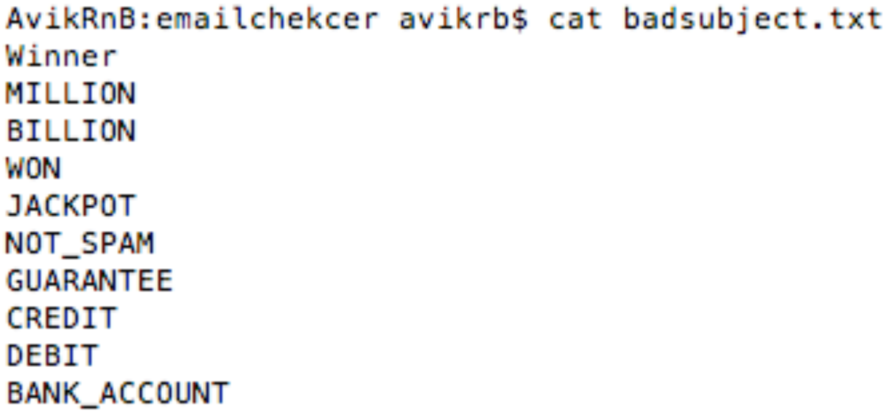
FILE THAT CONTAINS LIST OF BAD EMAIL IDs – bademailid.txt



```
AvikRnB:emailchekcer avikrb$ cat bademailid.txt
WINNER@gmail.com.eu
jade_madison07@hotmail.com.rh
annie_pink@yahoo.com.fr
boogie@yahoo.com.cn
dummy@cr.yahoo.sr
big_money@mail.right.com
milliondollars@mail.ru
6.sonika.donbass.net

bigcash.eu
solitary.com.fr
1000dollars.com
insurance.com
creditcard.com
```

FILE THAT CONTAINS LIST OF BAD KEYWORDS – badsubject.txt

NOTE: THIS FILE IS USED FOR BOTH SUBJECT CHECK AND BODY CHECK *



```
AvikRnB:emailchekcer avikrb$ cat badsubject.txt
Winner
MILLION
BILLION
WON
JACKPOT
NOT_SPAM
GUARANTEE
CREDIT
DEBIT
BANK_ACCOUNT
```