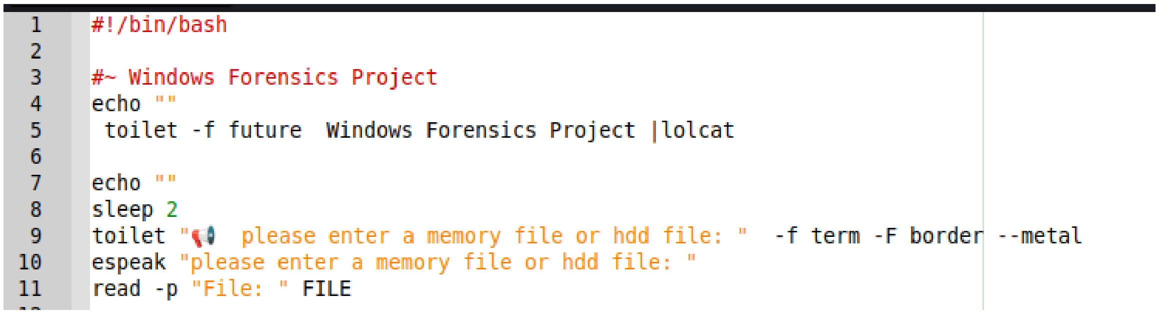
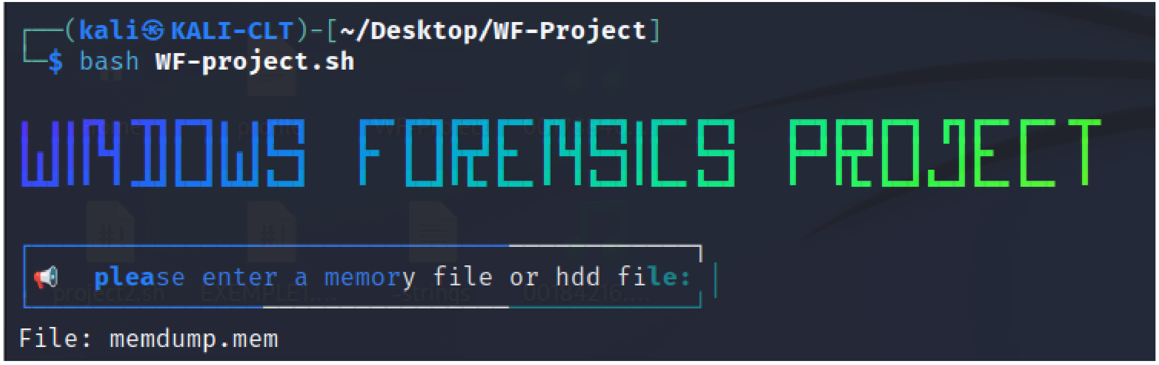
ANALYSIS PROJECT

About the project:

Analyze memory or hard disk files by extracting important files and moving them to a log folder

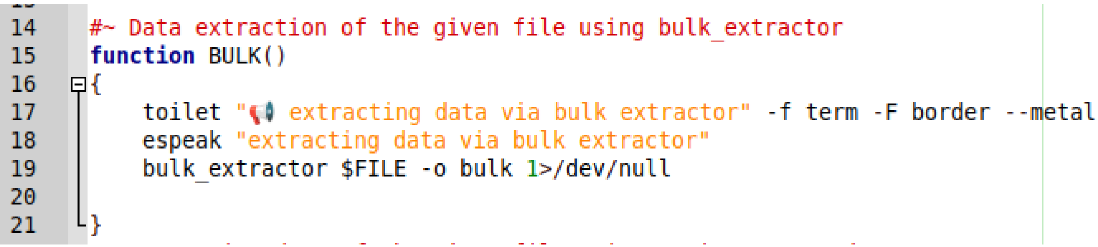


```
#!/bin/bash

#~ Windows Forensics Project
echo ""
 toilet -f future  Windows Forensics Project |lolcat

echo ""
sleep 2
toilet "📢  please enter a memory file or hdd file: "  -f term -F border --metal
espeak "please enter a memory file or hdd file: "
read -p "File: " FILE
```

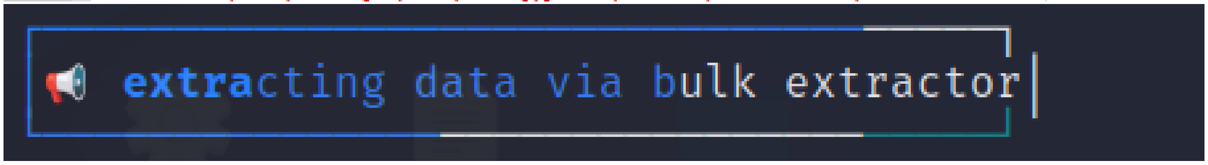


```
┌──(kali㉿KALI-CLT)-[~/Desktop/WF-Project]
└─$ bash WF-project.sh

WINDOWS FORENSICS PROJECT

┌─────────────────────────────────────────────────┐
│📢  please enter a memory file or hdd file:      │
└─────────────────────────────────────────────────┘
File: memdump.mem
```

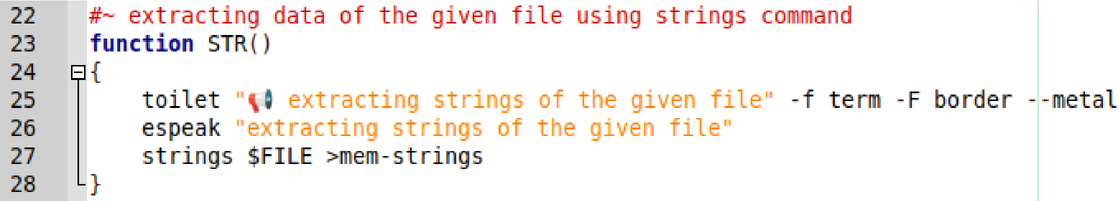
# functions for **extracting** important data from the given file



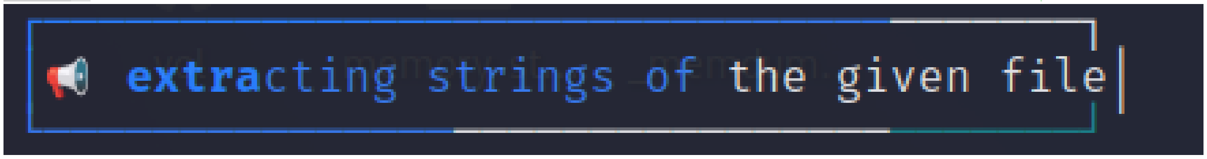
```
#~ Data extraction of the given file using bulk_extractor
function BULK()
{
    toilet "📢 extracting data via bulk extractor" -f term -F border --metal
    espeak "extracting data via bulk extractor"
    bulk_extractor $FILE -o bulk 1>/dev/null

}
```



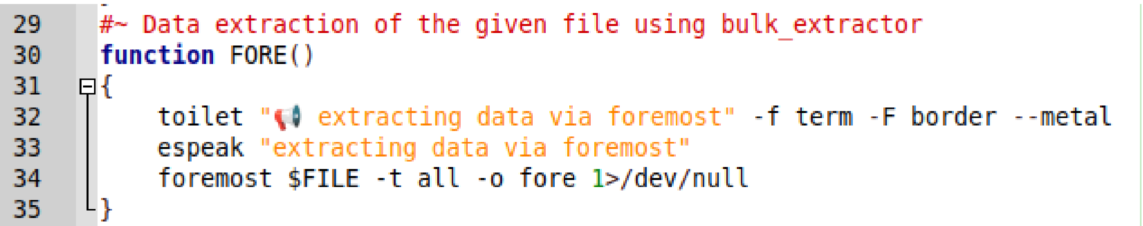
```
📢 extracting data via bulk extractor
```



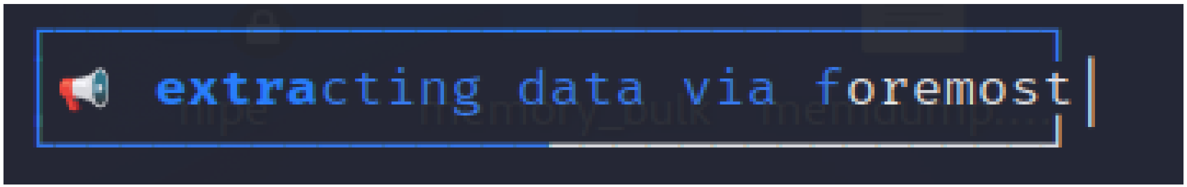
```
#~ extracting data of the given file using strings command
function STR()
{
    toilet "📢 extracting strings of the given file" -f term -F border --metal
    espeak "extracting strings of the given file"
    strings $FILE >mem-strings
}
```



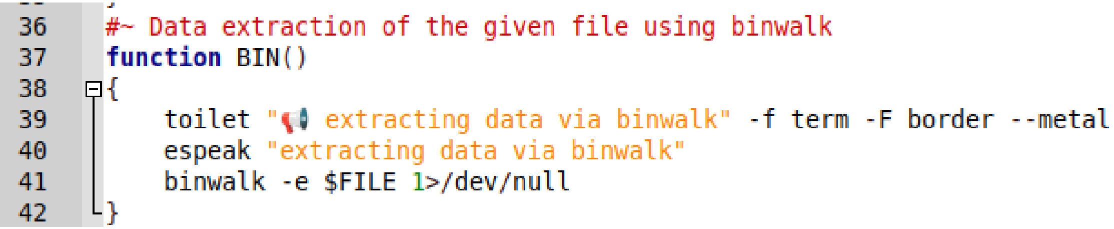
```
📢 extracting strings of the given file
```



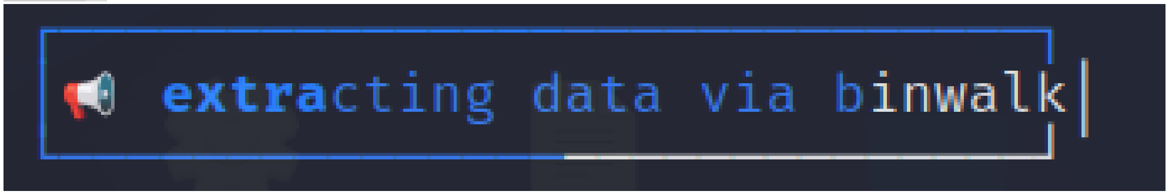
```
#~ Data extraction of the given file using bulk_extractor
function FORE()
{
    toilet "📢 extracting data via foremost" -f term -F border --metal
    espeak "extracting data via foremost"
    foremost $FILE -t all -o fore 1>/dev/null
}
```



```
📢 extracting data via foremost
```

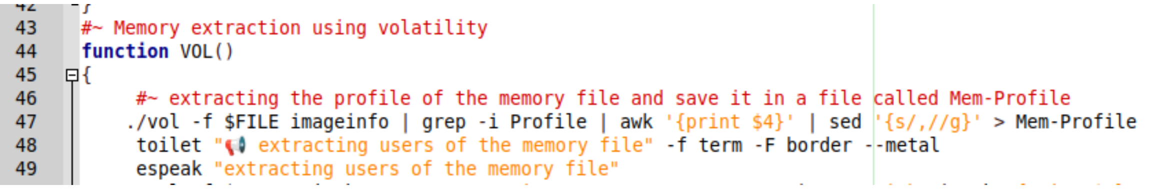


```
#~ Data extraction of the given file using binwalk
function BIN()
{
    toilet "📢 extracting data via binwalk" -f term -F border --metal
    espeak "extracting data via binwalk"
    binwalk -e $FILE 1>/dev/null
}
```

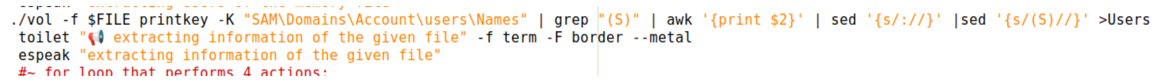


```
📢 extracting data via binwalk
```

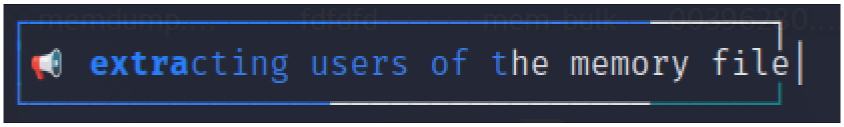
extracting the profile of the memory file via **volatility** command and save it in a file called Mem-Profile



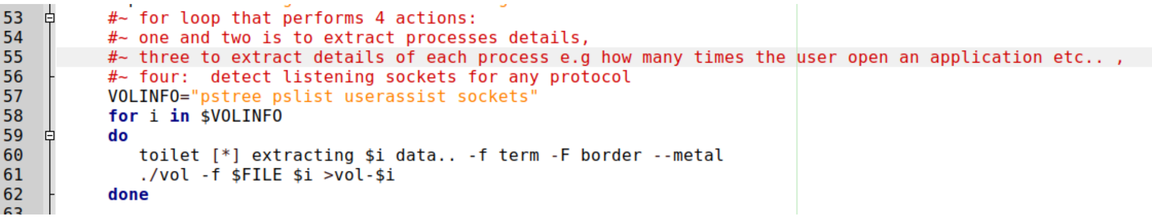
```
#~ Memory extraction using volatility
function VOL()
{
    #~ extracting the profile of the memory file and save it in a file called Mem-Profile
   ./vol -f $FILE imageinfo | grep -i Profile | awk '{print $4}' | sed '{s/,//g}' > Mem-Profile
    toilet "📢 extracting users of the memory file" -f term -F border --metal
    espeak "extracting users of the memory file"
```



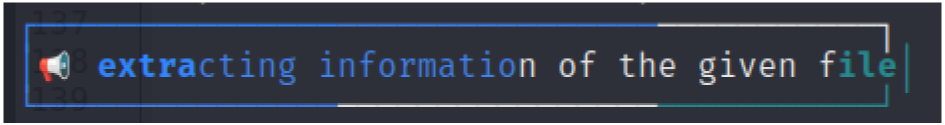
```
./vol -f $FILE printkey -K "SAM\Domains\Account\users\Names" | grep "(S)" | awk '{print $2}' | sed '{s/://}' |sed '{s/(S)//}' >Users
 toilet "📢 extracting information of the given file" -f term -F border --metal
 espeak "extracting information of the given file"
 #~ for loop that performs 4 actions:
```



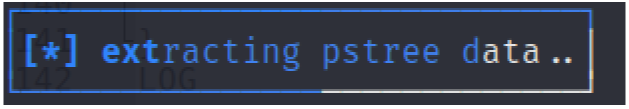
```
📢 extracting users of the memory file
```



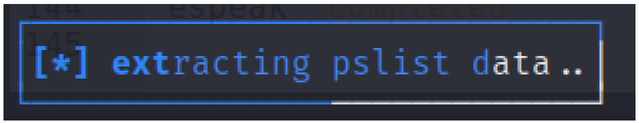
```
    #~ for loop that performs 4 actions:
    #~ one and two is to extract processes details,
    #~ three to extract details of each process e.g how many times the user open an application etc.. ,
    #~ four:  detect listening sockets for any protocol
    VOLINFO="pstree pslist userassist sockets"
    for i in $VOLINFO
    do
       toilet [*] extracting $i data.. -f term -F border --metal
       ./vol -f $FILE $i >vol-$i
    done
```



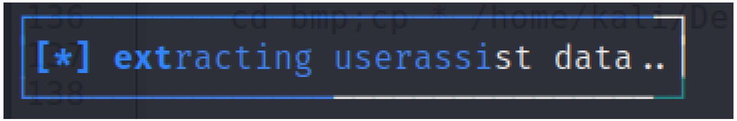
```
📢 extracting information of the given file
```



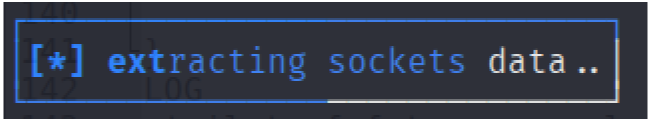
```
[*] extracting pstree data..
```



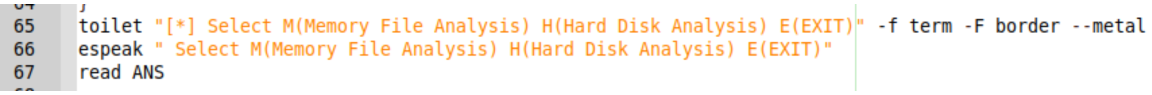
```
[*] extracting pslist data..
```



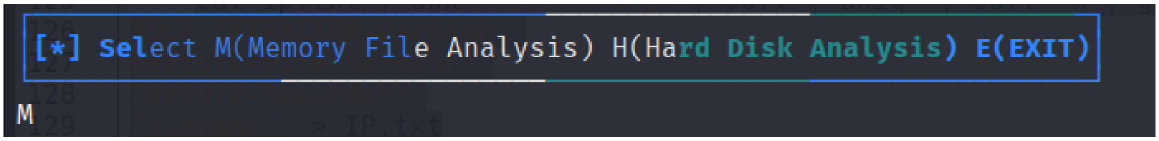
```
[*] extracting userassist data..
```



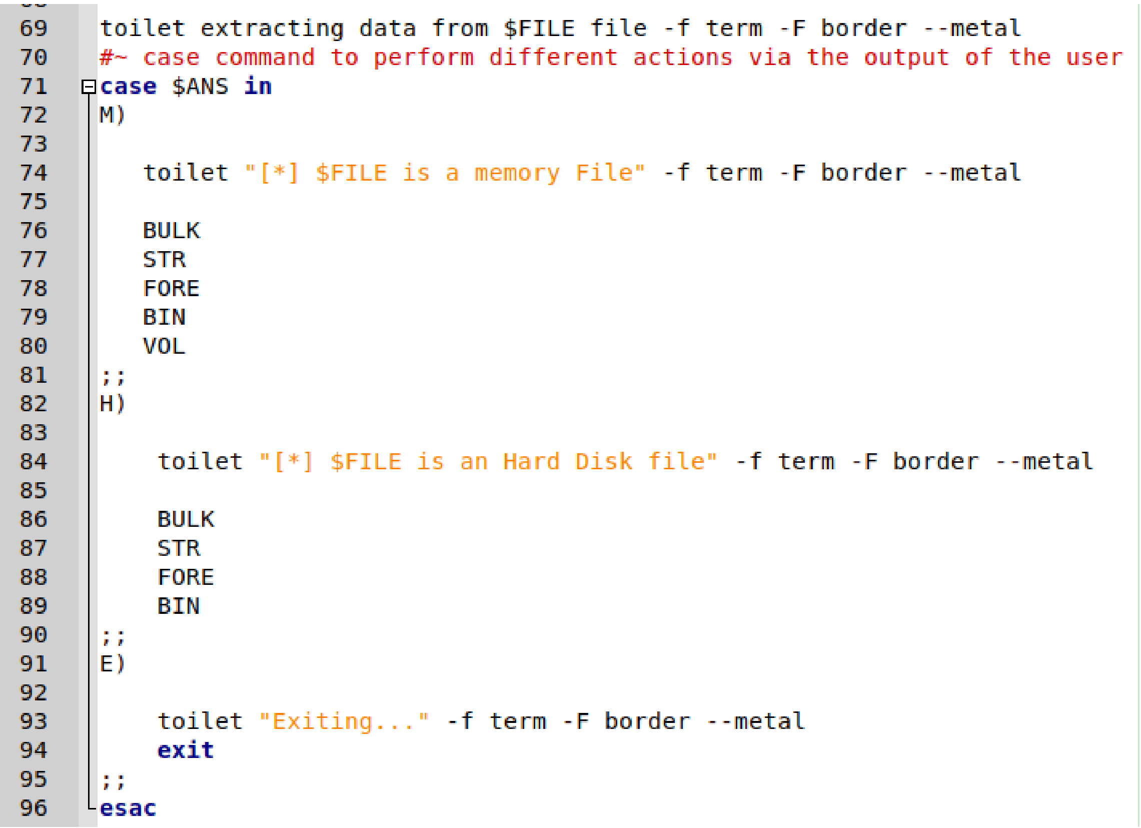
```
[*] extracting sockets data..
```



```
}
toilet "[*] Select M(Memory File Analysis) H(Hard Disk Analysis) E(EXIT)" -f term -F border --metal
espeak " Select M(Memory File Analysis) H(Hard Disk Analysis) E(EXIT)"
read ANS
```

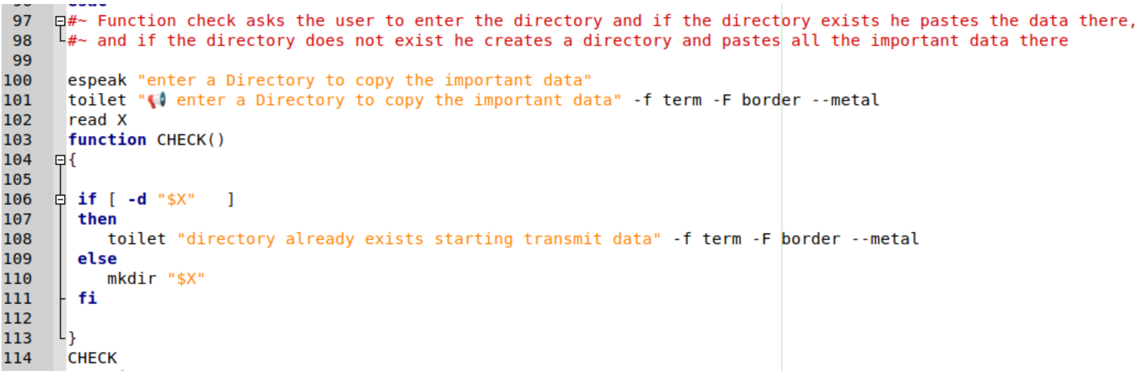


```
[*] Select M(Memory File Analysis) H(Hard Disk Analysis) E(EXIT)
M
```



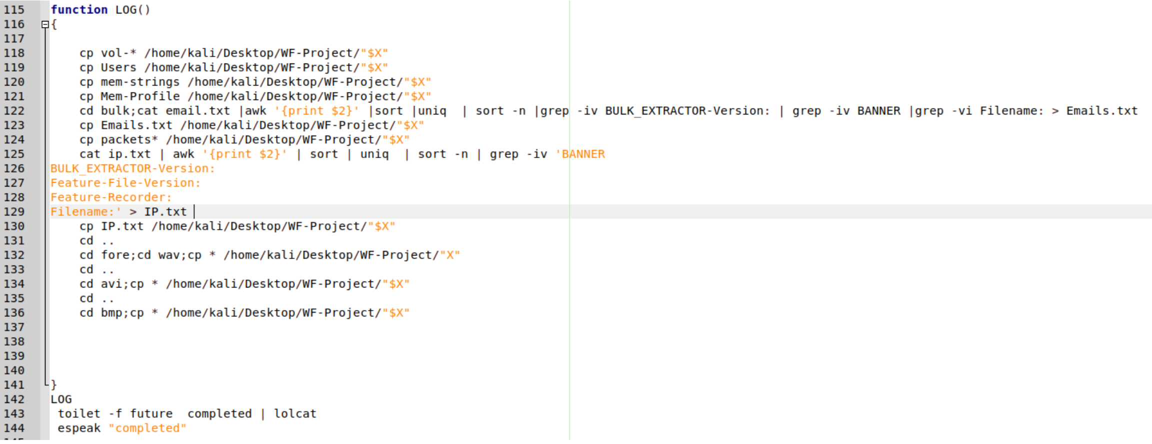
```
toilet extracting data from $FILE file -f term -F border --metal
#~ case command to perform different actions via the output of the user
case $ANS in
M)

    toilet "[*] $FILE is a memory File" -f term -F border --metal

    BULK
    STR
    FORE
    BIN
    VOL
;;
H)

     toilet "[*] $FILE is an Hard Disk file" -f term -F border --metal

     BULK
     STR
     FORE
     BIN
;;
E)

     toilet "Exiting..." -f term -F border --metal
     exit
;;
esac
```

Function check asks the user to enter the directory and if the directory exists he pastes the data there, and if the directory does not exist he creates a directory and pastes all the important data there



```
#~ Function check asks the user to enter the directory and if the directory exists he pastes the data there,
#~ and if the directory does not exist he creates a directory and pastes all the important data there

espeak "enter a Directory to copy the important data"
toilet "📢 enter a Directory to copy the important data" -f term -F border --metal
read X
function CHECK()
{

 if [ -d "$X"   ]
 then
    toilet "directory already exists starting transmit data" -f term -F border --metal
 else
    mkdir "$X"
 fi

}
CHECK
```

# function log to copy all the important details to a new directory that the user choose



```
function LOG()
{

    cp vol-* /home/kali/Desktop/WF-Project/"$X"
    cp Users /home/kali/Desktop/WF-Project/"$X"
    cp mem-strings /home/kali/Desktop/WF-Project/"$X"
    cp Mem-Profile /home/kali/Desktop/WF-Project/"$X"
    cd bulk;cat email.txt |awk '{print $2}' |sort |uniq  | sort -n |grep -iv BULK_EXTRACTOR-Version: | grep -iv BANNER |grep -vi Filename: > Emails.txt
    cp Emails.txt /home/kali/Desktop/WF-Project/"$X"
    cp packets* /home/kali/Desktop/WF-Project/"$X"
    cat ip.txt | awk '{print $2}' | sort | uniq  | sort -n | grep -iv 'BANNER
BULK_EXTRACTOR-Version:
Feature-File-Version:
Feature-Recorder:
Filename:' > IP.txt
    cp IP.txt /home/kali/Desktop/WF-Project/"$X"
    cd ..
    cd fore;cd wav;cp * /home/kali/Desktop/WF-Project/"X"
    cd ..
    cd avi;cp * /home/kali/Desktop/WF-Project/"$X"
    cd ..
    cd bmp;cp * /home/kali/Desktop/WF-Project/"$X"




}
LOG
 toilet -f future  completed | lolcat
 espeak "completed"
```

To run the script you must have the following plugins:

Toilet (sudo apt-get install toilet)

Espeak (sudo apt-get install espeak)

Lolcat (sudo apt-get install lolcat)

Yair Solomon

Analysis project

Lecturer: Natalie Erez