ACADEMIA TEHNICĂ MILITARĂ
„FERDINAND I”

Proiect practica

SnapshotsDiff

Amzuloiu Teodor (C112B)

Joița Ștefan-Eugen (C112B)

BUCUREȘTI
2021

Cuprins

Capitol 1 - *Introducere*

1.1.Scopul Proiectului

Acest proiect se bazează pe o mașină virtuală (sistemul de operare – Linux) căreia i s-au creat două snapshot-uri, în diferite stări, urmând ca apoi, prin intermediul unui utilitar dezvoltat prin Shell scripting, să se determine diferențele dintre imaginile celor două snapshot-uri ale mașinii virtuale.

1.2. Descrierea proiectului

Analiza și colectarea probelor digitale constituie procese importante în domeniul securității cibernetice. În prezent asistăm la o cerere mare de spații de stocare și de resurse de calcul avansate, motiv pentru care tehnologiile virtuale reprezintă un punct mare de interes din punct de vedere comercial, în multe situații fiind preferate mediile virtuale în defavoarea celor statice. Virtualizarea este pretutindeni întâlnită, întrucât se intenționează reducerea costurilor și migrarea către sistemele în cloud.

Într-un centru de date tradițional compromiterea unui singur calculator implică un risc ridicat la nivelul unui singur sistem sau unui singur proces. În schimb, în mediul virtual, lucrurile stau complet diferit. Dacă este compromisă o mașină virtuală, într-o organizație se pot produce efecte devastatoare.

Obiectivul principal al proiectului este realizarea unor module de bază pentru analiza unei mașini virtuale pe baza determinării diferențelor dintre două snapshot-uri ale acesteia, realizate la momente diferite de timp.

Atunci când maşinile virtuale sunt implicate într-un incident, acestea sunt de regulă suspendate sau investigatorul realizează un snapshot al maşinii virtuale, păstrând astfel procesele şi caracteristicile conexiunilor din reţea ale maşinii virtuale. Snapshot-ul unei maşini virtuale captează starea la un anumit moment a discului virtual, conţinutul memoriei maşinii virtuale, precum şi setările acesteia. Având captat discul virtual prin intermediul unui snapshot, datele regăsite pe acesta pot fi inspectate şi analizate, în funcţie de scopul investigaţiei. Un aspect important ce trebuie menţionat este că, spre deosebire de analiza mediilor fizice, în care datele şterse anterior puteau fi recuperate şi analizate, în cazul analizei mediilor virtuale prin intermediul snapshot-urilor datele şterse anterior nu pot fi recuperate decât în cadrul sistemului gazdă, întrucât stocarea se realizează în cadrul acestuia.

Capitol 2 – *Structură și organizare*

2.1. Organizarea proiectului

Pentru organizarea și implementarea acestui proiect s-a trecut prin urmatoarele etape:

- Pregatirea unei mașini virtuale cu distribuție de Linux pentru analiza imaginilor virtuale
    a) Crearea unui cont de utilizator pentru efectuarea analizei
    b) Setarea în modul permisiv a implementării Linux pentru controlul accesului (*Selinux - permissive mode*)
    c) Instalarea librăriilor specifice *The SleuthKit*

- Instalarea FTK Imager pentru realizarea in format dd/raw a discului virtual

- Pregatirea unei mașini virtuale cu distribuție de Linux pentru realizarea snapshot-urilor
    a) În starea 1, se va crea un director nou denumit *My Directory,* în care se creează fișierele *deleteme1.txt, deleteme2.txt* și *in.txt*, având un conținut la alegere.
    b) În starea 2, se va crea un fișier *added.txt,* se vor șterge fișierele *deleteme1.txt* și *deleteme2.txt* din directorul *My Directory* și se modifică conținutul fișierului *in.txt*.

2.2. Structura proiectului

Proiectul va urmari implementarea cu *shell scripting* a unui utilitar de comparare a celor două stări diferite ale mașinii virtuale analizate. Acesta va cuprinde:

a. Un meniu principal care să permită utilizatorului să aleagă opțiunile oferite de utilitar

b. Posibilitatea selectării fișierelor imagine de tip *dd* în care sunt regăsite cele două stări diferite ale mașinii virtuale analizate

c. Vizualizarea fișierelor șterse de pe mașina virtuală, pe baza comparării celor două snapshot-uri

d. Vizualizarea fișierelor adăugate pe mașina virtuală (fișierele care apar în cel de-al doilea Snapshot, dar nu apăreau în primul Snapshot)

e. Vizualizarea fișierelor editate din cadrul mașinii virtuale

f. Calcularea valorilor hash ale imaginilor discului virtual, folosind algoritmii MD5 și SHA1

g. Posibilitatea ieșirii din utilitar

Capitol 3 – *Implementare și testare*

3.1. Obținerea imaginilor discului virtual

Deschidem aplicația FTK Imager.

Accesam *File > Create disk Image.*

Selectam tipul fișierului sursă (în cazul de față, *Image File*).

Adăugam path-ul către fișierul sursă (adică path-ul catre discul virtual al mașinii virtuale create pentru analiză).

Selectam tipul de date al imaginii pe care dorim sa o opținem (în cazul de față, *Raw(dd)*) și alegem în ce folder să se creeze imaginea.

3.2. Inițializare utilitar

Aplicația noastră este formată din 3 fișiere de tip shell script: *meniu.sh* (care este core-ul aplicației), *hash_sum.sh* și *diff_files.sh.*

De asemenea, după rularea aplicației (și prin parcurgerea tuturor funcționalitațiilor), se vor creea în folderul imaginilor urmatoarele fisiere text: *sum.txt, temp_file1, temp_file2, temp_out_file1.txt, temp_out_file2.txt.*

Fiecare fișier va conține urmatoarele inițializari de variabile și de funcții:

```
#!/bin/bash

#### Color  Variables
green='\e[32m'
blue='\e[34m'
red='\e[31m'
yellow='\e[33m'
clear='\e[0m'
curr_scripts_path=$(pwd)

#### Color Functions

ColorGreen(){
        echo -ne $green$1$clear
}
ColorBlue(){
        echo -ne $blue$1$clear
}
ColorRed(){
        echo -ne $red$1$clear
}
ColorYellow(){
        echo -ne $yellow$1$clear
}
```

3.3. Creare meniu principal

Pentru a intra in meniul principal (meniul ce ofera utilizatorului funcționalitățiile utilitarului) trebuie configurati mai întâi câtiva parametri: folderul din mașina virtuală unde sunt salvate imaginile de analizat și selectarea celor două imagini de comparat.

```
declare -i menu_error=0
menu(){

        echo -ne  "$(ColorYellow 'Meniu')

$(ColorGreen '1)') Add Path
$(ColorGreen '0)') Exit
$(ColorBlue 'Choose an option:') "
        read a
        case $a in
                1)read_path ;;
                0) exit 0 ;;
                *)menu_error=$menu_error+1;clear; echo -e "$(ColorRed 'Wrong option(+'$menu_error'). Choose a valid option!')"; menu ;;

        esac
}
```

```
Meniu

1) Add Path
0) Exit
Choose an option: 1
Introdu path-ul catre folderul dorit:
```

Un meniu de început ce iti oferă: posibilitatea de a adăuga path-ul către folderul cu imaginile discului virtual și posibilitatea de a ieși din utilitar.

```
declare -i conf_path_error=0
function conf_path()
{
        echo -ne "$(ColorGreen 'Ai introdus path-ul') $path_folder$(ColorGreen ' . Va rog sa confirmati daca este corect')"
        echo -ne " [y/n]: "
        read msg
        case $msg in
                y) listare_img ;;
                n) clear; menu_error=0; menu ;;
                *)conf_path_error=$conf_path_error+1;clear; echo -e "$(ColorRed 'Wrong option (+'$conf_path_error').Choose a valid option!')";conf_path;;

       esac

}

##
function read_path()
{
echo -ne "$(ColorGreen 'Introdu path-ul catre folderul dorit: ')" ;
                        read path_folder;
                        clear;
                        if [ -d $path_folder ]
                          then
                                funct2_error=0; conf_path;
                          else
                                echo -e "$(ColorRed 'Enter a path to an existing directory!')"; read_path;

                        fi

}
```

(citirea și confirmarea path-ului folderului)

În urma introducerii unui folder existent în mașina virtuală utilizată analizei, se vor lista imaginiile discurilor virtuale existente în folder și se va oferi

posibilitatea de a selecta două dintre ele (secțiunea de listare și alegere a imaginiilor va fi desrisă separat, ulterior).

După selectarea a două imagini valide, se va deschide meniul cu funcționalități.

```
declare -i menu_error2=0
function meniu_functionalitati()
{
        echo -ne "$(ColorYellow 'Meniu functionalitati')

$(ColorGreen '1)') Fisiere sterse
$(ColorGreen '2)') Fisiere nou adaugate
$(ColorGreen '3)') Fisiere modificate
$(ColorGreen '4)') Valorile hash
$(ColorGreen '0)') Exit
$(ColorBlue 'Choose an option:') "
        read b
        case $b in
                0)exit 0;;
                1)clear;
                        echo "$(ColorGreen 'Loading....(< 1 min )')"
                        $curr_scripts_path/diff_files.sh $first_snap $second_snap $path_folder;
                        clear;
                        fisiere_sterse_error=0;
                        fisiere_sterse  ;;
                2) clear;
                        echo "$(ColorGreen 'Loading....(< 1 min )')"
                        $curr_scripts_path/diff_files.sh $first_snap $second_snap $path_folder
                        clear;
                        fisiere_adaugate_error=0;
                        fisiere_adaugate        ;;
                3) clear;
                        echo "$(ColorGreen 'Loading....(< 1 min )')"
                        $curr_scripts_path/diff_file.sh $first_snap $second_snap $path_folder
                        clear;
                        fisiere_modificate=0;
                        fisiere_modificate;;
                4) clear;
                        echo "$(ColorGreen 'Loading....(> 2 min )')"
                        $curr_scripts_path/hash_sum.sh $first_snap $second_snap $path_folder
                        clear;
                        hash_error=0
                        back_to_menu_functionalitati ;;

                *)menu_error2=$menu_error2+1;clear; echo -e "$(ColorRed 'Wrong option(+'$menu_error2'). Choose a valid option!')"; meniu-functionalitati ;;

        esac

}
```

```
declare -i hash_error=0
function back_to_menu_functionalitati()
{
        cat sum.txt
        echo " "
        IFS=" "
        echo -ne "$(ColorGreen 'Inapoi la meniu') [y/n]$(ColorGreen '? ')"
        read c
        case $c in
                y)clear;
                        meniu_functionalitati;;
                n)clear;
                        exit 0 ;;
                *)hash_error=$hash_error+1;clear; echo -e "$(ColorRed 'Wrong option (+'$hash_error').Choose a valid option!')";cat sum.txt; back_to_menu_functionalitati;;
        esac

}
```

```
Meniu functionalitati

1) Fisiere sterse
2) Fisiere nou adaugate
3) Fisiere modificate
4) Valorile hash
0) Exit
Choose an option:
```

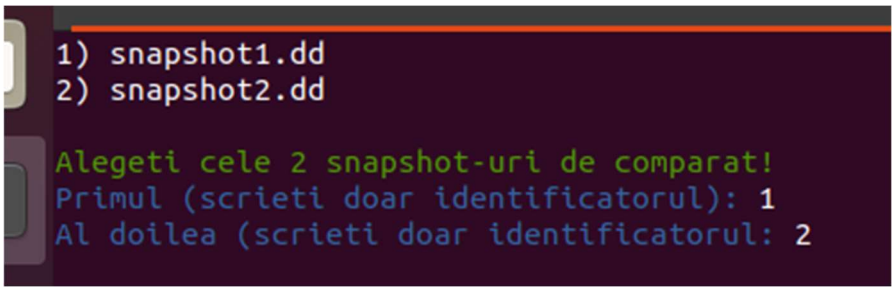
Meniul de funcționalități ce cuprinde urmatoarele opțiuni:

- Vizualizarea fișierelor șterse;
- Vizualizarea fișierelor adaugate în a doua stare;
- Vizualizarea fișierelor al caror conținut a fost modificat în starea a doua;
- Calcularea valorilor hash ale imaginilor discurilor virtuale, folosind algoritmii MD5 și SHA1;
- Posibilitatea ieșirii din utilitar.

3.4. Listare și selecție imagini

În urma introducerii unui folder existent în mașina virtuală utilizată analizei, se vor lista imaginiile discurilor virtuale existente în folder și se va oferi posibilitatea de a selecta două dintre ele.

```
##
function listare_img()
{

cd $path_folder
        if [ -f temp.txt ]
        then
                rm temp.txt
        fi
declare -i nr_list_img=1
for i in $(ls $path_folder)
do
        if  file $i|grep -q "boot sector";
        then
                echo "$nr_list_img) $i" >> temp.txt
                nr_list_img=$nr_list_img+1
        fi
done
clear;
alegere_snap_error=1
alegere_snap
}
```

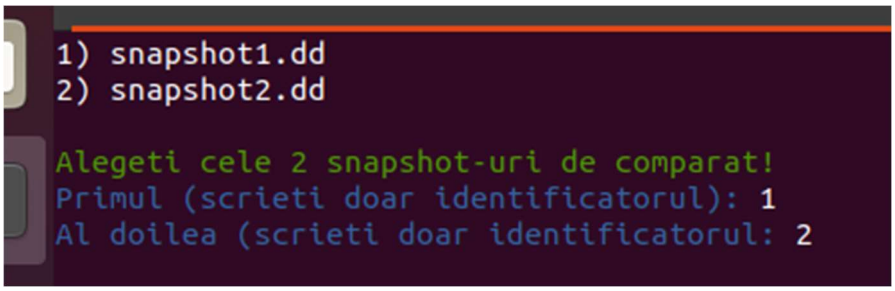


(listarea imaginilor din folderul selectat)

În cazul în care folderul introdus nu are fisiere de tip raw(dd), se va oferi posibilitatea de a introduce alt folder.

```
declare -i alegere_snap_error=1
function alegere_snap()
{
        if [ -f temp.txt ]

        cat temp.txt
        echo ""
        else
        clear;
        echo -e "$(ColorRed 'Folderul '$path_folder' nu are fisiere de tip dd! ')";
        menu
        fi
        echo -e "$(ColorGreen 'Alegeti cele 2 snapshot-uri de comparat!')"

        echo -ne "$(ColorBlue 'Primul (scrieti doar identificatorul): ')";
        read id1;
        if cat temp.txt|egrep -q $id1;
        then
        echo -ne "$(ColorBlue 'Al doilea (scrieti doar identificatorul: ')";
        read id2;
        if cat temp.txt|egrep -q $id2;
        then


        if [ "$id1" = "$id2" ]
        then
                clear;
                echo -e "$(ColorRed 'Alege snapshot-uri diferite! ')";
                alegere_snap;
        else
                first_snap=$(cat temp.txt| egrep "$id1)" | cut -f2 -d" ");
                second_snap=$(cat temp.txt | egrep "$id2)" | cut -f2 -d" ");
                rm temp.txt
        fi
        clear;
        meniu_functionalitati
        else
        clear;
        echo -e "$(ColorRed 'Nu exista cel putin unul dintre identificatorii alesi(+'$alegere_snap_error')!')";
        alegere_snap_error=$alegere_snap_error+1
        alegere_snap

        fi
```

```
        fi
        else
        clear;
        echo -e "$(ColorRed 'Nu exista cel putin unul dintre identificatorii alesi(+'$alegere_snap_error')!')";
       alegere_snap_error=$alegere_snap_error+1
        alegere_snap
        fi
}
```



(Se oferă posibilitatea selectării imaginilor de comparat)

3.5. Vizualizare fișiere șterse

Pentru a obține output-ul acestei funcționalității se selecteaza optiunea 1 din meniul de funcționalități.

```
1)clear;
        echo "$(ColorGreen 'Loading....(< 1 min )')"
        $curr_scripts_path/diff_files.sh $first_snap $second_snap $path_folder;
        clear;
        fisiere_sterse_error=0;
        fisiere_sterse  ;;
```

(codul din interiorul meniului de functionalități)

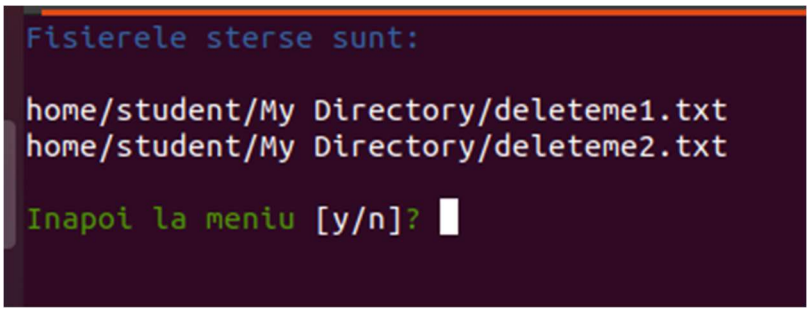
Fișierul de tip shell script *diff_files* creează în interiorul folderului cu imagini, fișierele text *temp_file1* și *temp_file2*. Aceste fișiere conțin desfășurarea întregului sistem de fișiere, însă supus anumitor filtre, al imaginilor selectate.

```
cd $3

if [ -f temp_file1 ]
then
      rm temp_file1
fi

if [ -f temp_file2 ]
then
     rm temp_file2
fi

for temp_counter in $(mmls -a $1 | tr -s " " |cut -f3 -d" "|tail -n +6)
do
       fls -p -r -o $temp_counter $1 | egrep "home/.*/" | egrep -v "(\/\.)|(realloc)" >>temp_file1
done

for temp_counter2 in $(mmls -a $2| tr -s " " |cut -f3 -d" "| tail -n +6)
do
       fls -p -r -o $temp_counter2 $2 | egrep "home/.*/" |egrep -v "(\/\.)|(realloc)"  >>temp_file2
done

#END
```

(diff_files.sh)

```
declare -i fisiere_sterse_error=0
function fisiere_sterse()
{
        IFS=$'\n'
        echo "$(ColorBlue 'Fisierele sterse sunt: ')"
        echo " "
        for i in $(cat temp_file1 |cut -f2 -d":"|tr -d '\t')
        do
                if cat temp_file2 | egrep -q "$i$" ;
                then
                echo -ne ""     #nothing
                else
                echo "$i"
                fi
        done
        echo " "
        IFS=" "
        echo -ne "$(ColorGreen 'Inapoi la meniu') [y/n]$(ColorGreen '? ')";
        read c
        case $c in
                y)clear;
                        meniu_functionalitati;;
                n)clear;
                        exit 0 ;;
                *)fisiere_sterse_error=$fisiere_sterse_error+1;clear; echo -e "$(ColorRed 'Wrong option (+'$fisiere_sterse_error').Choose a valid option!')";fisiere_sterse;;
        esac

}
```

(codul funcției *fisiere_sterse*)



(output-ul funcției)

3.6. Vizualizare fișiere adăugate

Pentru a obține output-ul acestei funcționalități se selecteaza optiunea 2 din meniul de funcționalități.

```
  2) clear;
          echo "$(ColorGreen 'Loading....(< 1 min )')"
          $curr_scripts_path/diff_files.sh $first_snap $second_snap $path_folder
          clear;
          fisiere_adaugate_error=0;
          fisiere_adaugate        ;;
```

(codul din interiorul meniului de functionalități)

Fișierul de tip shell script *diff_files* creează în interiorul folderului cu imagini, fișierele text *temp_file1* și *temp_file2*. Aceste fișiere conțin desfășurarea întregului sistem de fișiere, însă supus anumitor filtre, al imaginilor selectate.

```
cd $3

if [ -f temp_file1 ]
then
       rm temp_file1
fi

if [ -f temp_file2 ]
then
      rm temp_file2
fi

for temp_counter in $(mmls -a $1 | tr -s " " |cut -f3 -d" "|tail -n +6)
do
        fls -p -r -o $temp_counter $1 | egrep "home/.*/" | egrep -v "(\/\.)|(realloc)" >>temp_file1
done

 for temp_counter2 in $(mmls -a $2| tr -s " " |cut -f3 -d" "| tail -n +6)
 do
        fls -p -r -o $temp_counter2 $2 | egrep "home/.*/" |egrep -v "(\/\.)|(realloc)"  >>temp_file2
 done

 #END
```

(diff_files.sh)

```
##
declare -i fisiere_adaugate_error=0
function fisiere_adaugate()
{
        IFS=$'\n'
        echo "$(ColorBlue 'Fisierele adaugate sunt: ')"
        echo " "
        for j in $(cat temp_file2| cut -f2 -d":"|tr -d '\t')
        do
                if cat temp_file1 | egrep -q "$j$" ;
                then
                echo -ne ""     #nothing
                else
                        echo "$j"
                fi
        done
        echo " "
        IFS=" "
        echo -ne "$(ColorGreen 'Inapoi la meniu') [y/n]$(ColorGreen '? ')"
        read d
        case $d in
                y)clear;
                        meniu_functionalitati;;
                n)clear;
                        exit 0 ;;
                *)fisiere_adaugate_error=$fisiere_adaugate_error+1;clear; echo -e "$(ColorRed 'Wrong option (+'$fisiere_adaugate_error').Choose a valid option!')";fisiere_adaugate;;
        esac
}
```

(codul funcției *fisiere_adaugate*)

```
Fisierele adaugate sunt:

home/student/My Directory/added.txt

Inapoi la meniu [y/n]?
```

(output-ul funcției)

3.7. Vizualizare fișiere modificate

Pentru a obține output-ul acestei funcționalității se selecteaza optiunea 3 din meniul de funcționalități.

```
3) clear;
        echo "$(ColorGreen 'Loading....(< 1 min )')"
        $curr_scripts_path/diff_file.sh $first_snap $second_snap $path_folder
        clear;
        fisiere_modificate=0;
        fisiere_modificate;;
```

(codul din interiorul meniului de functionalități)

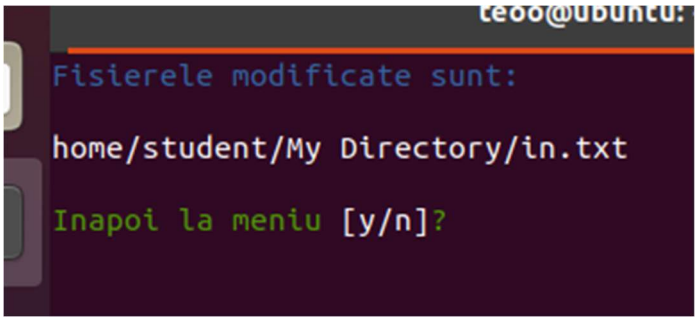
Fișierul de tip shell script *diff_files* creează în interiorul folderului cu imagini, fișierele text *temp_file1* și *temp_file2*. Aceste fișiere conțin desfășurarea întregului sistem de fișiere, însă supus anumitor filtre, al imaginilor selectate.

```
cd $3

if [ -f temp_file1 ]
then
      rm temp_file1
fi

if [ -f temp_file2 ]
then
     rm temp_file2
fi

for temp_counter in $(mmls -a $1 | tr -s " " |cut -f3 -d" "|tail -n +6)
do
       fls -p -r -o $temp_counter $1 | egrep "home/.*/" | egrep -v "(\/\.)|(realloc)" >>temp_file1
done

for temp_counter2 in $(mmls -a $2| tr -s " " |cut -f3 -d" "| tail -n +6)
do
       fls -p -r -o $temp_counter2 $2 | egrep "home/.*/" |egrep -v "(\/\.)|(realloc)"  >>temp_file2
done

#END
```

(diff_files.sh)

```
##
declare -i fisiere_modificate_error=0
function fisiere_modificate()
{
        if [ -f temp_out_file1.txt ]
        then
                rm temp_out_file1.txt
        fi
        if [ -f temp_out_file2.txt ]
        then
                rm temp_out_file2.txt
        fi
        IFS=$'\n'
        echo "$(ColorBlue 'Fisierele modificate sunt: ')"
        echo " "
        for i in $(cat temp_file1|cut -f2 -d":"|tr -d '\t')
        do
                if cat temp_file2|grep "r/r" | egrep -q "$i$";
                then
                        innode1=$(cat temp_file1| egrep "$i"| tr '\t' " " |cut -f2 -d" " | tr -d ":")
                        innode2=$(cat temp_file2| egrep "$i"| tr '\t' " " |cut -f2 -d" " | tr -d ":")
                        for temp_counter in $(mmls -a $first_snap | tr -s " " |cut -f3 -d" "|tail -n +6)
                        do
                        icat -o $temp_counter $first_snap $innode1 1>>temp_out_file1.txt 2>/dev/null
                        done
                                md5_file1=$(md5sum temp_out_file1.txt|cut -f1 -d" ")
                        for temp_counter2 in $(mmls -a $second_snap | tr -s " " |cut -f3 -d" "|tail -n +6)
                        do
                        icat -o $temp_counter2 $second_snap $innode2 1>>temp_out_file2.txt 2>/dev/null
                        done
                                md5_file2=$(md5sum temp_out_file2.txt|cut -f1 -d" ")
                        if [ "$md5_file1" = "$md5_file2" ];
                        then
                            echo -ne ""
                    else
                                echo "$i"
                        fi
                fi
        done
        echo " "
        IFS=" "
        echo -ne "$(ColorGreen 'Inapoi la meniu') [y/n]$(ColorGreen '? ')"
        read c
        case $c in
                y)clear;
                        meniu_functionalitati;;
                n)clear;
                        exit 0 ;;
                *)fisiere_modificate_error=$fisiere_modificate_error+1;clear; echo -e "$(ColorRed 'Wrong option (+'$fisiere_modificate_error').Choose a valid option!')";fisiere_modificate;;
        esac
}
```

(codul funcției *fisiere_adaugate*)



(output-ul funcției)

3.8. Calculare hash al imaginilor selectate

Pentru a obține output-ul acestei funcționalități se selecteaza optiunea 4 din meniul de funcționalități.

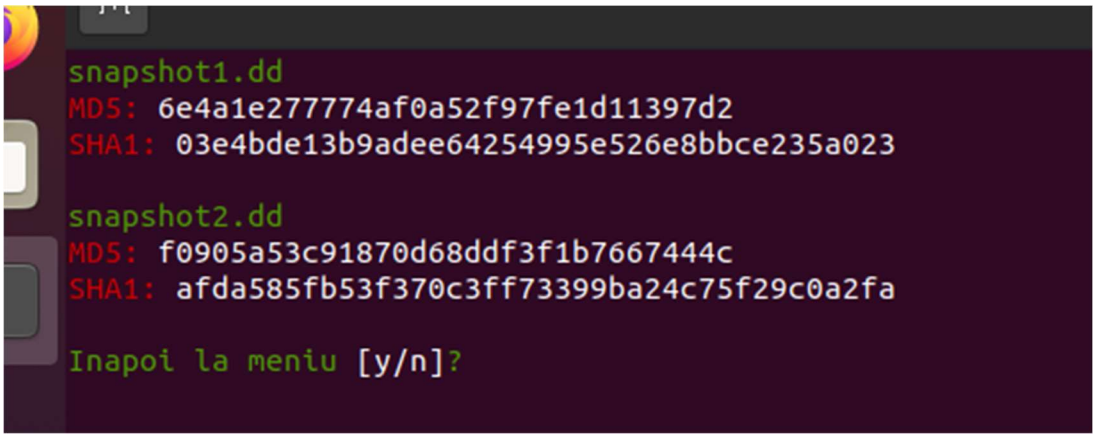
```
4) clear;
        echo "$(ColorGreen 'Loading....(> 2 min )')"
        $curr_scripts_path/hash_sum.sh $first_snap $second_snap $path_folder
        clear;
        hash_error=0
        back_to_menu_functionalitati ;;
```

(codul din interiorul meniului de functionalități)

Fișierul de tip shell script *hash_sum* creează în interiorul folderului cu imagini, fișierul text *sum.txt.* Aceste fișier va conține output-ul algoritmilor MD5 și SHA1 aplicați imaginilor selectate.

```
### code
## MD5 si SHA1 pentru img disk-urilor date ca parametri

cd $3
if [ -f sum.txt ]
then
        rm sum.txt
else
md5_1=$(md5sum $1|cut -f1 -d" ");
md5_2=$(md5sum $2|cut -f1 -d" ");
sha1_1=$(sha1sum $1|cut -f1 -d" ");
sha1_2=$(sha1sum $2|cut -f1 -d" ");

echo "$(ColorGreen ''$1'')" >>sum.txt
echo "$(ColorRed 'MD5:   ')$md5_1" >> sum.txt
echo "$(ColorRed 'SHA1:  ')$sha1_1" >>sum.txt
echo " " >>sum.txt
echo "$(ColorGreen ''$2'')" >>sum.txt
echo "$(ColorRed 'MD5:   ')$md5_2" >> sum.txt
echo "$(ColorRed 'SHA1:  ')$sha1_2" >>sum.txt

fi

#END
```

(hash_sum.sh)



(output-ul funcției)

Capitol 4 – *Webografie*

https://hackernoon.com/getting-started-with-digital-forensics-using-the-sleuth-kit-c34a3wkg

https://cyberoperations.wordpress.com/class-archives/2013-class/10-vmware-forensics-with-autopsy/

https://gurramvinayiiit.files.wordpress.com/2016/11/introduciton-tothe-sleuthkit.pdf

https://zaptyt.ru/ro/sluzhba-selinux-v-komandnoi-stroke-nastroika-selinux-vklyuchenie-otklyuchenie-i/

http://acmelabs-galleries.s3.amazonaws.com/48/0000/2352/forensic_cheatsheet.pdf

https://www.hackingarticles.in/comprehensive-guide-on-ftk-imager/