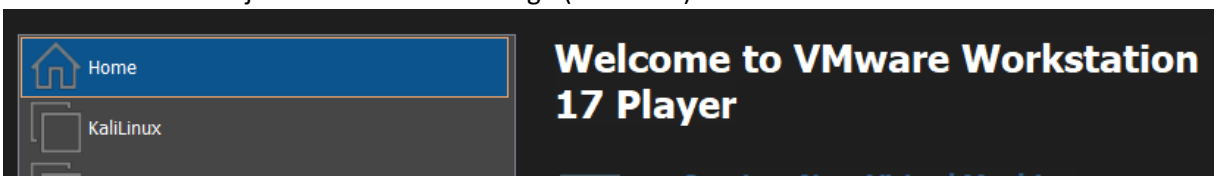
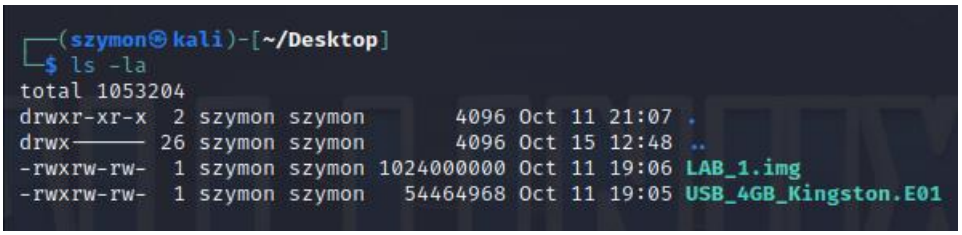
Raport z LAB1 – Szymon Szkarłat

Zadanie 1. – Instalacja środowiska wirtualnego (Kali Linux)



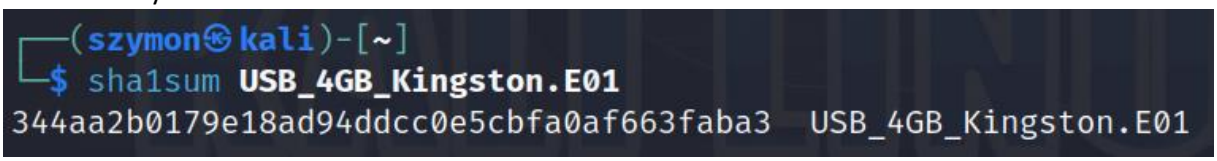
Na poniższym screenie widać pobrane pliki. Pliki te dla wygody podczas przeprowadzania ich analizy przeniosłem do katalogu domowego.



Zadanie 2. – Analiza pobranego obrazu (plik .E01)

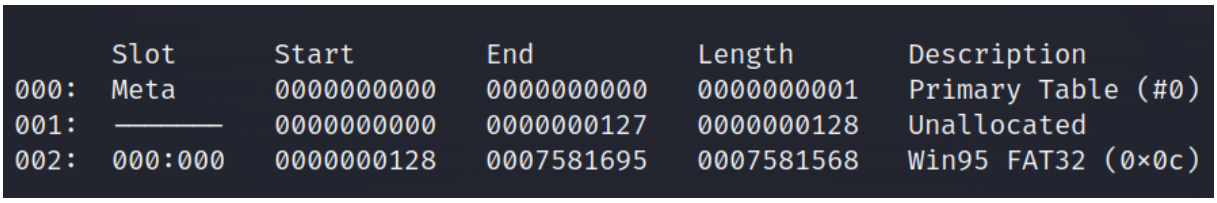
1. Wartość skrótu dla funkcji haszującej md5 dla USB_4GB_Kingstom.E01 wynosi:

Dla SHA-1 wynosi natomiast:

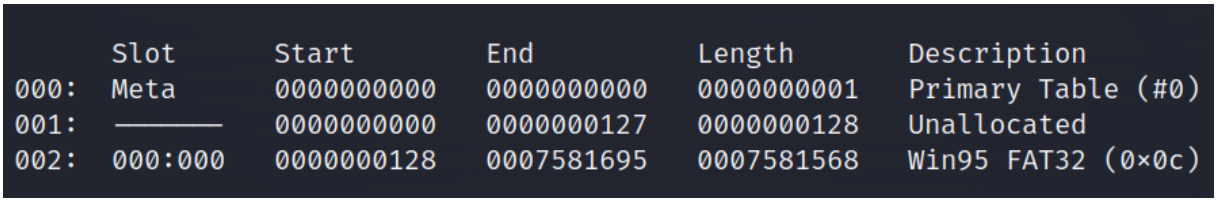


Wykorzystanie polecenia mmls

1. Niealokowana pamięć znajduje się w sektorze, który mieści się w zakresie od 0000000000 do 000000127.



2. Pliki systemowe znajdują się w partycji 002, o czym świadczy opis tej partycji, tj. Win95 FAT32



3. Początek sektora należącego do partycji Win95 to: 0000000128, a koniec: 0007581568. Co potwierdza powyższy screen.

Wykorzystanie narzędzia fsstat

1. FAT32 zaczyna się w sektorze 0000000128 analizowanego pliku.

```
┌──(szymon㉿kali)-[~]
└─$ fsstat -o 128 USB_4GB_Kingston.E01
FILE SYSTEM INFORMATION
--------------------------------------------

File System Type: FAT32
```

2. Wielkość sektora wynosi: 7581568 oraz wielkość klastra w badanym obszarze to: 7574111.

```
└─$ fsstat -o 128 USB_4GB_Kingston.E01
FILE SYSTEM INFORMATION
--------------------------------------------

File System Type: FAT32

OEM Name: MSDOS5.0
Volume ID: 0x779c953c
Volume Label (Boot Sector): USB DISK
Volume Label (Root Directory):
File System Type Label: FAT32
Next Free Sector (FS Info): 11392
Free Sector Count (FS Info): 7504624

Sectors before file system: 128

File System Layout (in sectors)
Total Range: 0 - 7581567
* Reserved: 0 - 47
** Boot Sector: 0
** FS Info Sector: 1
** Backup Boot Sector: 8
* FAT 0: 48 - 3751
* FAT 1: 3752 - 7455
* Data Area: 7456 - 7581567
** Cluster Area: 7456 - 7581567
*** Root Directory: 7456 - 7471
```

Narzędzie fls

1. Wszystkie pliki głównego katalogu USB_4GB_Kingstom.E01 to (6 plików):
IMG_5609.JPG, IMG_5627.JPG, IMG_5753.JPG, IMG_6002.JPG, IMG_8064.JPG, text2.rar

```
┌──(szymon㉿kali)-[~]
└─$ fls -i ewf -f fat32 -o 128 USB_4GB_Kingston.E01
r/r 3:  USB DISK    (Volume Label Entry)
d/d 6:  .Spotlight-V100
d/d * 8:        .fseventsd
d/d 9:  1
r/r 10: IMG_5609.JPG
r/r * 13:       ._IMG_5609.JPG
r/r 14: IMG_5627.JPG
r/r * 17:       ._IMG_5627.JPG
r/r 18: IMG_5753.JPG
r/r * 21:       ._IMG_5753.JPG
r/r 22: IMG_6002.JPG
r/r * 25:       ._IMG_6002.JPG
r/r 26: IMG_8064.JPG
r/r * 29:       ._IMG_8064.JPG
r/r 30: text2.rar
r/r * 32:       ._text2.rar
r/r * 34:       ._1
v/v 121185795:  $MBR
v/v 121185796:  $FAT1
v/v 121185797:  $FAT2
V/V 121185798:  $OrphanFiles
```

2. Wszystkie pliki znajdujące się w folderze „1”. Wykorzystałem komendę:

```
┌──(szymon㉿kali)-[~]
└─$ fls -i ewf -f fat32 -o 128 -p -r USB_4GB_Kingston.E01
r/r 3:  USB DISK    (Volume Label Entry)
r/r * 1037:     .fseventsd/000000
r/r 62725:      1/IMG_6110.JPG
r/r 62726:      1/IMG_5592.JPG
r/r 62727:      1/text.txt
```

W folderze tym znajdują się 3 pliki: IMG_6110.JPG, IMG_5592.JPG, text.txt

Użycie *EWFTools/ewfinfo*

```
┌──(szymon㉿kali)-[~]
└─$ sudo apt install libewf-dev ewf-tools
```

1. Numer sprawy to: 001
2. Nazwa osoby tworzącej obraz dysku: Kali

```
┌──(szymon㉿kali)-[~]
└─$ ewfinfo USB_4GB_Kingston.E01
ewfinfo 20140814

Acquiry information
        Case number:            001
        Examiner name:          Kali
        Evidence number:        001
        Acquisition date:       Sun Oct  3 16:31:05 2021
        System date:            Sun Oct  3 16:31:05 2021
```

3. Plik został utworzony: 3 października 2021 roku w niedzielę o 16:31:05

4. Numer seryjny fizycznego dysku oraz nazwa modelu na poniższym screenie, tj. USB DISK 2.0 oraz nr seryjny: 0D7117891080

```
Model:                  USB DISK 2.0
Serial number:          0D7117891080
```

5. Format pliku: EnCase6

```
EWF information
        File format:            EnCase 6
```

6. Metoda kompresji: deflate

```
Compression method:     deflate
```

7. Pełna wielkość badanego nośnika w bajtach to: 3881828352 bajtów.

```
    Media size:             3.6 GiB (3881828352 bytes)
```

8. Poziom kompresji: good (fast)

```
Compression level:      good (fast) compression
```

Zadanie 3. Analiza pobranego obrazu (LAB_1.img)

1. Liczba sektorów to: 1

```
gpt_load_table: Sector: 1
gpt_load: 0  Starting Sector: 2048  End: 104447 Flag: 0
gpt_load: 1  Starting Sector: 104448  End: 309247 Flag: 0
gpt_load: 2  Starting Sector: 309248  End: 718847 Flag: 0
gpt_load: 3  Starting Sector: 718848  End: 1058815 Flag: 0
gpt_load: 4  Starting Sector: 1058816  End: 1091583 Flag: 0
gpt_load: 5  Starting Sector: 1091584  End: 1173503 Flag: 0
```

2. Sektor startowy gpt_load: 0 to 2048

```
gpt_load: 0  Starting Sector: 2048  End: 104447 Flag: 0
```

3. Liczba niealokowanych sektorów to: 2.

Pierwszy: 0000 – 2047

Drugi: 1173504 - 1999999

```
┌──(szymon㉿kali)-[~]
└─$ mmls -A LAB_1.img
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
001:  ------    0000000000   0000002047   0000002048   Unallocated
010:  ------    0001173504   0001999999   0000826496   Unallocated
```

4. Ujawnione woluminy to: fat16, fat32, ntfs, ext4, swap, minix

```
┌──(szymon㉿kali)-[~]
└─$ mmls -a LAB_1.img
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
004:  000       0000002048   0000104447   0000102400   fat16
005:  001       0000104448   0000309247   0000204800   fat32
006:  002       0000309248   0000718847   0000409600   ntfs
007:  003       0000718848   0001058815   0000339968   ext4
008:  004       0001058816   0001091583   0000032768   swap
009:  005       0001091584   0001173503   0000081920   minix
```

5. Przy pomocy polecenia *mmstat* wyświetlam informacje tablicy partycji. Odczytujemy z poniższego screena, w którym sektorze znajduje się *gpt_load_table* oraz w którym sektorze rozpoczyna się *gpt_load*.

```
┌──(szymon㉿kali)-[~]
└─$ mmstat -v LAB_1.img
tsk_img_open: Type: 0   NumImg: 1  Img1: LAB_1.img
aff_open: Error determining type of file: LAB_1.img
aff_open: No such file or directory
Error opening vmdk file
Error checking file signature for vhd file
tsk_img_findFiles: LAB_1.img found
tsk_img_findFiles: 1 total segments found
raw_open: segment: 0  size: 1024000000  max offset: 1024000000  path: LAB_1.i
mg
dos_load_prim: Table Sector: 0
raw_read: byte offset: 0 len: 65536
raw_read: found in image 0 relative offset: 0 len: 65536
raw_read_segment: opening file into slot 0: LAB_1.img
dos_load_prim_table: Testing FAT/NTFS conditions
load_pri:0:0    Start: 1   Size: 1999999  Type: 238
load_pri:0:1    Start: 0   Size: 0  Type: 0
load_pri:0:2    Start: 0   Size: 0  Type: 0
load_pri:0:3    Start: 0   Size: 0  Type: 0
bsd_load_table: Table Sector: 1
gpt_load_table: Sector: 1
gpt_load: 0  Starting Sector: 2048  End: 104447 Flag: 0
```

6. Przy pomocy narzędzia fsstat wyświetliłem informacje o woluminie „ntfs” oraz dowiedziałem się, że „Volume Serial Number” to: 451AF24C771A6637. Natomiast wersja to: Windows XP

```
Volume Serial Number: 451AF24C771A6637
OEM Name: NTFS
Volume Name: NTFS
Version: Windows XP
```

Zadanie 4 – tworzenie kopii binarnej pendrive'a

1. Wyświetlenie podłączonych urządzeń

```
┌──(szymon㉿kali)-[~]
└─$ sudo fdisk -l
[sudo] password for szymon:
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x85e77e42

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1  *        2048 60913663 60911616   29G 83 Linux
/dev/sda2       60915710 62912511  1996802  975M  5 Extended
/dev/sda5       60915712 62912511  1996800  975M 82 Linux swap / Solaris


Disk /dev/sdb: 7.5 GiB, 8053063680 bytes, 15728640 sectors
Disk model: UDisk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x0005cc9e

Device     Boot Start      End  Sectors  Size Id Type
/dev/sdb1  *     2048 15728639 15726592  7.5G  c W95 FAT32 (LBA)
```

2. Za pomocą poniżej przedstawionego polecenia tworzymy kopię binarną pendrive'a

```
┌──(szymon㉿kali)-[~]
└─$ sudo ewfacquire /dev/sdb1
```

Opcje jakie wybrałem podczas tworzenia kopii

```
The following acquiry parameters were provided:
Image path and filename:                /home/szymon/2023_USB_1_Szymon.E01
Case number:                            001
Description:                            image
Evidence number:                        001
Examiner name:                          Szymon
Notes:
Media type:                             removable disk
Is physical:                            no
EWF file format:                        EnCase 6 (.E01)
Compression method:                     deflate
Compression level:                      none
Acquiry start offset:                   0
Number of bytes to acquire:             7.5 GiB (8052015104 bytes)
Evidence segment file size:             1.4 GiB (1493172224 bytes)
Bytes per sector:                       512
Block size:                             64 sectors
Error granularity:                      64 sectors
Retries on read error:                  2
Zero sectors on read error:             no

Continue acquiry with these values (yes, no) [yes]: yes
```

Potwierdzenie wprowadzonych przeze mnie opcji

```
┌──(szymon㉿kali)-[~]
└─$ sudo ewfacquire /dev/sdb1
ewfacquire 20140814

Device information:
Bus type:                               USB
Vendor:                                 General
Model:                                  UDisk
Serial:

Storage media information:
Type:                                   Device
Media type:                             Removable
Media size:                             8.0 GB (8052015104 bytes)
Bytes per sector:                       512

Acquiry parameters required, please provide the necessary input
Image path and filename without extension: /home/szymon/2023_USB_1_Szymon
Case number: 001
Description: image
Evidence number: 001
Examiner name: Szymon
Notes:
Media type (fixed, removable, optical, memory) [removable]: removable
Media characteristics (logical, physical) [logical]: logical
Use EWF file format (ewf, smart, ftk, encase1, encase2, encase3, encase4, encase5, encase6, linen5, linen6, ewfx) [encase6]: encase6
Compression method (deflate) [deflate]: deflate
Compression level (none, empty-block, fast, best) [none]: none
Start to acquire at offset (0 ≤ value ≤ 8052015104) [0]: 0
The number of bytes to acquire (0 ≤ value ≤ 8052015104) [8052015104]: 8052015104
Evidence segment file size in bytes (1.0 MiB ≤ value ≤ 7.9 EiB) [1.4 GiB]: 1.4 GiB
The number of bytes per sector (1 ≤ value ≤ 4294967295) [512]: 512
The number of sectors to read at once (16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768) [64]: 64
The number of sectors to be used as error granularity (1 ≤ value ≤ 64) [64]: 64
The number of retries when a read error occurs (0 ≤ value ≤ 255) [2]: 2
Wipe sectors on read error (mimic EnCase like behavior) (yes, no) [no]: no
```

Tworzenie kopii zakończyło się pomyślnie, o czym informuje nas komunikat u dołu poniższego screena.

```
Status: at 99%.
        acquired 7.5 GiB (8041201664 bytes) of total 7.5 GiB (8052015104 bytes).
        completion in 7 second(s) with 10 MiB/s (10910589 bytes/second).

Acquiry completed at: Oct 15, 2023 09:19:24

Written: 7.5 GiB (8052015292 bytes) in 12 minute(s) and 11 second(s) with 10 MiB/s (11015068 bytes/second).
MD5 hash calculated over data:          7c9eaa64fbcde880c211f9f0275ba796
ewfacquire: SUCCESS
```

Za pomocą polecenia ls -la dowiadujemy się, że kopia binarna pendrive'a została utworzona, znajduje się w lokalizacji, którą podałem podczas wprowadzania opcji oraz posiada dokładnie taką samą nazwę jaką podałem podczas tego procesu.

```
┌──(szymon㉿kali)-[~]
└─$ ls -la
total 7868908
drwx------ 26 szymon szymon       4096 Oct 15 09:18 .
drwxr-xr-x  3 root   root         4096 May 10 16:21 ..
-rw-r--r--  1 root   root   1493163983 Oct 15 09:09 2023_USB_1_Szymon.E01
-rw-r--r--  1 root   root   1493163281 Oct 15 09:11 2023_USB_1_Szymon.E02
-rw-r--r--  1 root   root   1493163281 Oct 15 09:14 2023_USB_1_Szymon.E03
-rw-r--r--  1 root   root   1493163281 Oct 15 09:16 2023_USB_1_Szymon.E04
-rw-r--r--  1 root   root   1493163281 Oct 15 09:18 2023_USB_1_Szymon.E05
-rw-r--r--  1 root   root    589156553 Oct 15 09:19 2023_USB_1_Szymon.E06
-rw-r--r--  1 szymon szymon       220 May 10 16:21 .bash_logout
```

Zweryfikowanie hashy MD5 za pomocą komendy przedstawione na poniższym screenie.