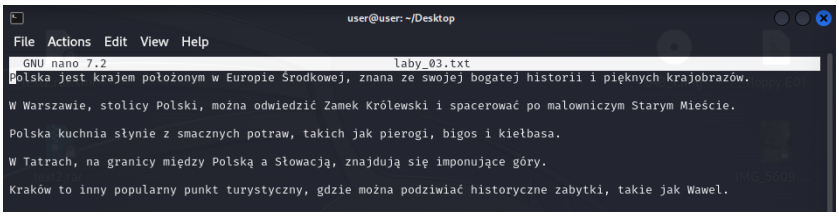
Informatyka śledcza Laboratorium nr 3

Raport – Nikodem Jakubowski

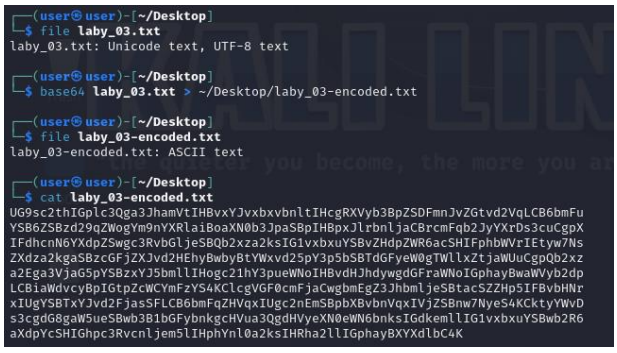
Zadanie 1 – Base64 jako narzędzie do kodowania i dekodowania.

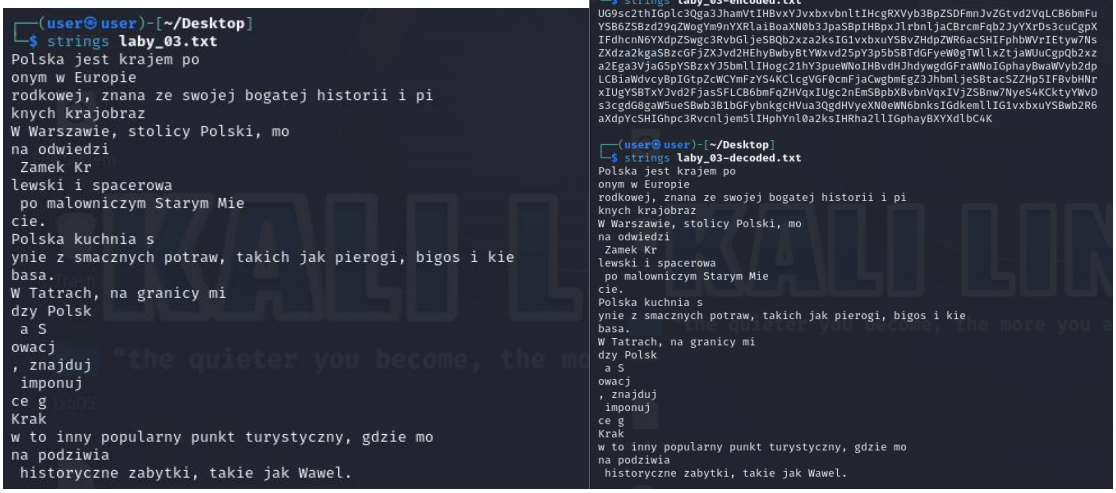
Rozwiązana łamigłówka.

Tworzę plik na pulpicie.



Korzystam z pozostałych narzędzi.





Co ciekawe $file rozpoznaje plik zakodowany w Base64 jako ASCII, a $strings nie rozpoznaje polskich znaków i wypisuje zamiast nich znak nowej linii.

Zadanie 2 – W zakładce pliki do przedmiotu Informatyka Śledcza znajduje się katalog File.zip, który zawiera przykładowe pliki o różnych rozszerzeniach.

Użycie komendy $file oraz $pdfinfo.

Następnie informacje z $pdfinfo.

D19910350Lj.pdf

Title: Akt prawny

Author: Władysław Baksza

Creator: Microsoft® Word 2013

Producer: Microsoft® Word 2013

CreationDate: Tue Oct 12 13:08:08 2021 CEST

ModDate: Tue Oct 12 13:08:08 2021 CEST

Custom Metadata: no

Metadata Stream: no

Tagged: yes

UserProperties: no

Suspects: no

Form: none

JavaScript: no

Pages: 351

Encrypted: no

Page size: 595.32 x 841.92 pts (A4)

Page rot: 0

File size: 2601654 bytes

Optimized: no

PDF version: 1.5

D2020000211201.pdf

Title: Ustawa z dnia 28 października 2020 r. o zmianie niektórych ustaw w związku z przeciwdziałaniem sytuacjom kryzysowym związanym z wystąpieniem COVID-19

Author: RCL

Creator: Microsoft® Word 2010

Producer: Microsoft® Word 2010; modified using iText 2.1.7 by 1T3XT

CreationDate: Sat Nov 28 18:39:52 2020 CET

ModDate: Sat Nov 28 18:40:01 2020 CET

Custom Metadata: no

Metadata Stream: yes

Tagged: yes

UserProperties: no

Suspects: no

Form: AcroForm

JavaScript: no

Pages: 18

Encrypted: no

Page size: 595.32 x 841.92 pts (A4)

Page rot: 0

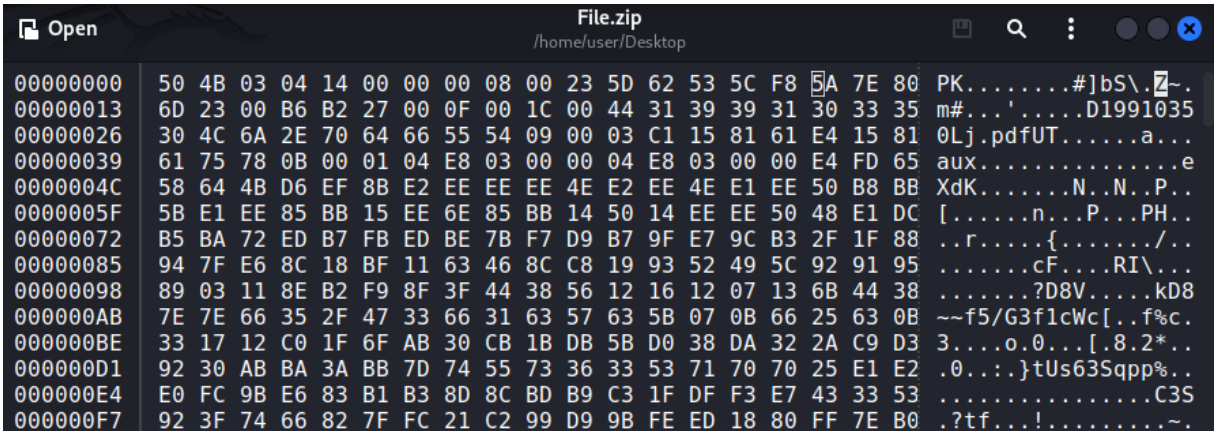
File size: 407654 bytes

Optimized: no

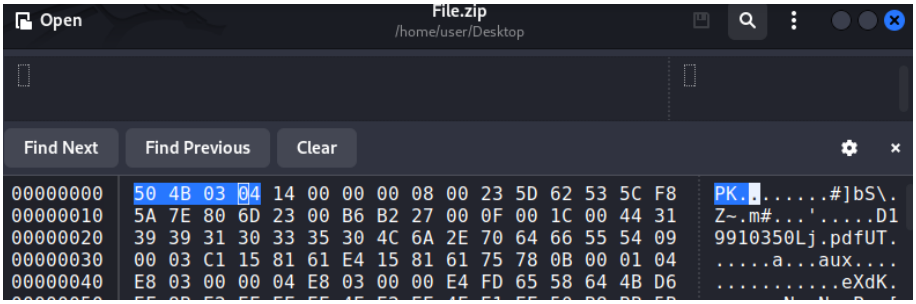
PDF version: 1.5

Zadanie 3 – Właściwości narzędzia GHex.

Informacja o rozszerzeniu archiwum znajduje się w pierwszym wierszu.

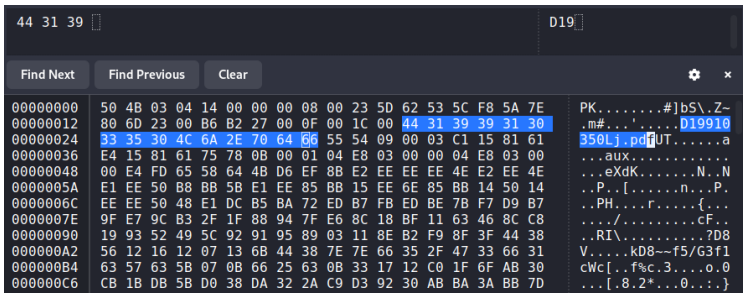


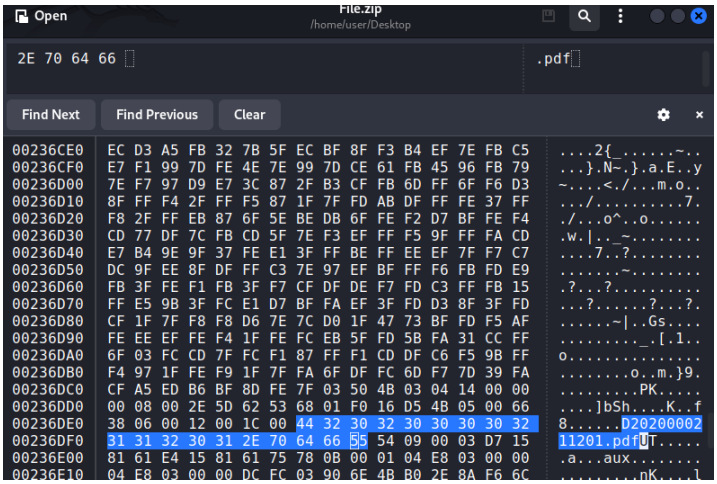
Sygnatura pliku .zip.

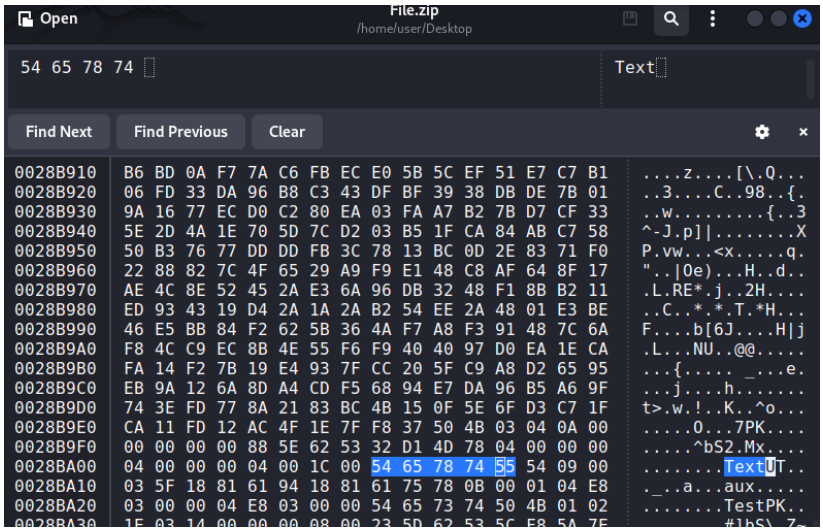


Sygnatura pliku ZIP to czterobajtowy nagłówek, który występuje na początku każdego pliku ZIP. Sygnatura ta ma postać: 50 4B 03 04. Gdzie każda liczba to reprezentacja szesnastkowa jednego bajtu. Odpowiada to ASCII dla liter "PK", to skrót od "Phil Katz" - twórcy formatu ZIP.

Zawartość archiwum (typu .zip czy .rar) jest możliwa do wykrycia. Poniżej znalezione sygnatury.





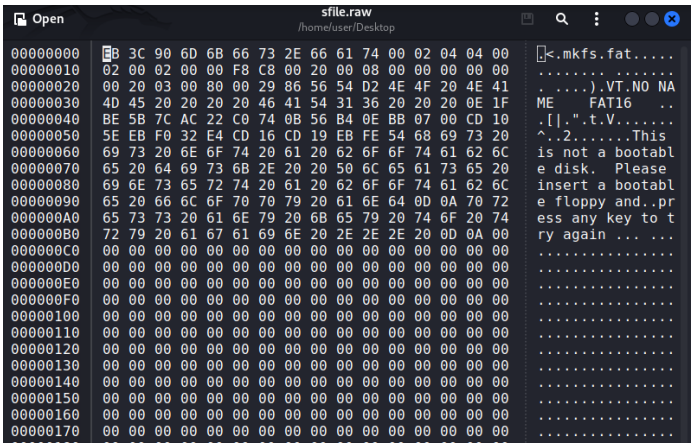


Zadanie 4 – Różnice pomiędzy systemami plików?

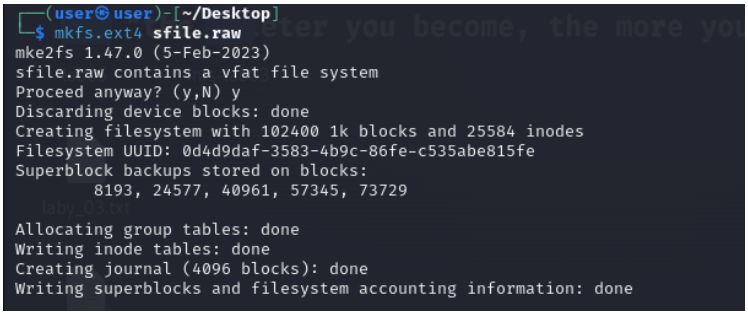
Wykonuję po kolei polecenia. Różnica – plik jest zapełniony zerami.


```
┌──(user㉿user)-[~/Desktop]
└─$ dd if=/dev/zero of=sfile.raw count=100 bs=1M
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 0.280054 s, 374 MB/s
```

Po wykonaniu $mkfs.fat, plik dostał dodatkową informacje.



Po wykonaniu $mkfs.ext4.



„Superbloki” zostały utworzone na blokach: 8193, 24577, 40961, 57345, 73729.

Użycie $dumpe2fs.

Parsowane informacje poniżej.

Magiczny numer: 0xEF53,

Numer UUID: 0d4d9daf-3583-4b9c-86fe-c535abe815fe,

Wielkość bloku: 1024,

Liczba wolnych bloków: 4095,

Podaj „checksum” typ: crc32c,

Wolne bloki w gr. 12: 98305-102399.

Dzięki wykonaniu $fsck.ext4 wiemy, że wykorzystano około 12 % bloków.

```
┌──(user㉿user)-[~/Desktop]
└─$ fsck.ext4 sfile.raw
e2fsck 1.47.0 (5-Feb-2023)
sfile.raw: clean, 11/25584 files, 12081/102400 blocks
```

Zadanie 5 – Znaczenie superbloku w odzyskiwaniu danych.

Po użyciu $losetup zostało przydzielone /dev/loop0.

```
┌──(root㉿user)-[/home/user/Desktop]
└─# sudo mount /dev/loop0 /mnt/hgfs
```

```
┌──(root㉿user)-[/home/user/Desktop]
└─# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0  100M  0 loop /mnt/hgfs
sda      8:0    0   25G  0 disk
├─sda1   8:1    0   24G  0 part /
├─sda2   8:2    0     1K  0 part
└─sda5   8:5    0  975M  0 part [SWAP]
sr0     11:0    1 1024M  0 rom
```

```
┌──(root㉿user)-[/home/user/Desktop]
└─# ls -li /mnt/hgfs
total 12
11 drwx------ 2 root root 12288 Nov 16 19:21 lost+found
```

Katalog "lost+found" jest standardowym katalogiem w systemie plików ext2/ext3/ext4. Jest to miejsce, gdzie system przechowuje pliki, które zostały uszkodzone lub nie mają właściwych atrybutów.

```
┌──(root㉿user)-[/mnt/hgfs]
└─# touch newitem

┌──(root㉿user)-[/mnt/hgfs]
└─# ls
lost+found  newitem
```

```
┌──(root㉿user)-[/mnt/hgfs]
└─# dd if=/dev/zero of=/dev/loop0 count=1 bs=1024 seek=1
1+0 records in
1+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 0.00636488 s, 161 kB/s
```

Po tym jest pusto.

```
┌──(root㉿user)-[/mnt/hgfs]
└─# ls -a
```

Próba odmontowania kończy się niepowodzeniem, trzeba „zabić” proces.

```
┌──(root㉿user)-[/mnt/hgfs]
└─# umount /dev/loop0
umount: /mnt/hgfs: target is busy.
```

Zabicie procesu.

Próba zamontowania kończy się niepowodzeniem, „zły superblock”.

```
┌──(root㉿user)-[/mnt/tmp]
└─# mount /dev/loop0 /mnt/hgfs
mount: /mnt/hgfs: wrong fs type, bad option, bad superblock on /dev/loop0, missing codepage or helper program, or ot
her error.
       dmesg(1) may have more information after failed mount system call.
```

Rezultat odzyskanych danych.

```
┌──(root㉿user)-[/mnt/tmp]
└─# fsck -f -y -b 8193 /dev/loop0 /mnt/hgfs
fsck from util-linux 2.39.2
e2fsck 1.47.0 (5-Feb-2023)
e2fsck 1.47.0 (5-Feb-2023)
fsck.ext2: Is a directory while trying to open /mnt/hgfs

The superblock could not be read or does not describe a valid ext2/ext3/ext4
filesystem.  If the device is valid and it really contains an ext2/ext3/ext4
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>
 or
    e2fsck -b 32768 <device>

Superblock needs_recovery flag is clear, but journal has data.
Recovery flag not set in backup superblock, so running journal anyway.
/dev/loop0: recovering journal
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Block bitmap differences:  +(8193--8450) +(24577--24834) +(40961--41218) +(57345--57602) +(73729--73986)
Fix? yes

Free inodes count wrong for group #0 (1957, counted=1956).
Fix? yes

Free inodes count wrong (25573, counted=25572).
Fix? yes

Padding at end of inode bitmap is not set. Fix? yes


/dev/loop0: ***** FILE SYSTEM WAS MODIFIED *****
/dev/loop0: 12/25584 files (0.0% non-contiguous), 12081/102400 blocks
```

Tutaj odzyskany wolumen.