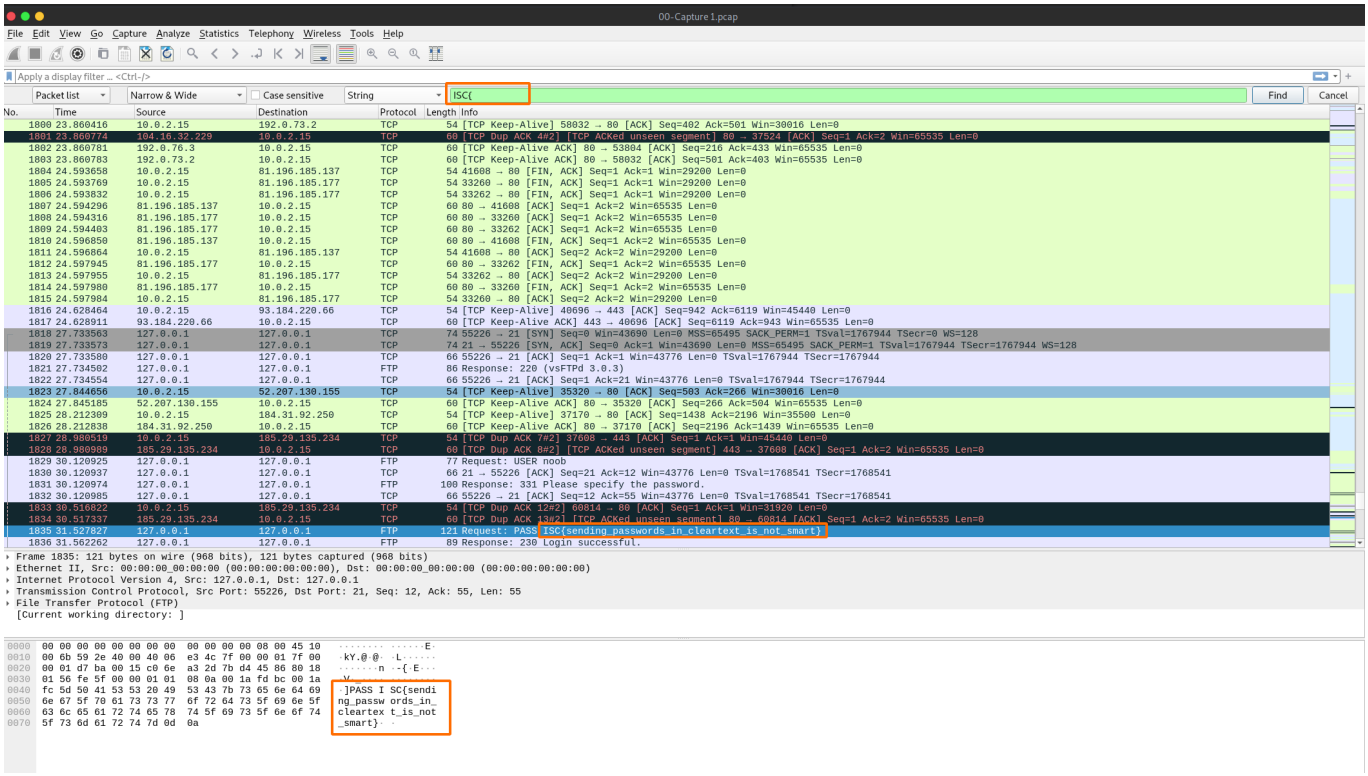
00 - Capture 1

Dupa ce am deschis captura in Wireshark si am cautat stringul **ISC{** in packete, am gasit un match



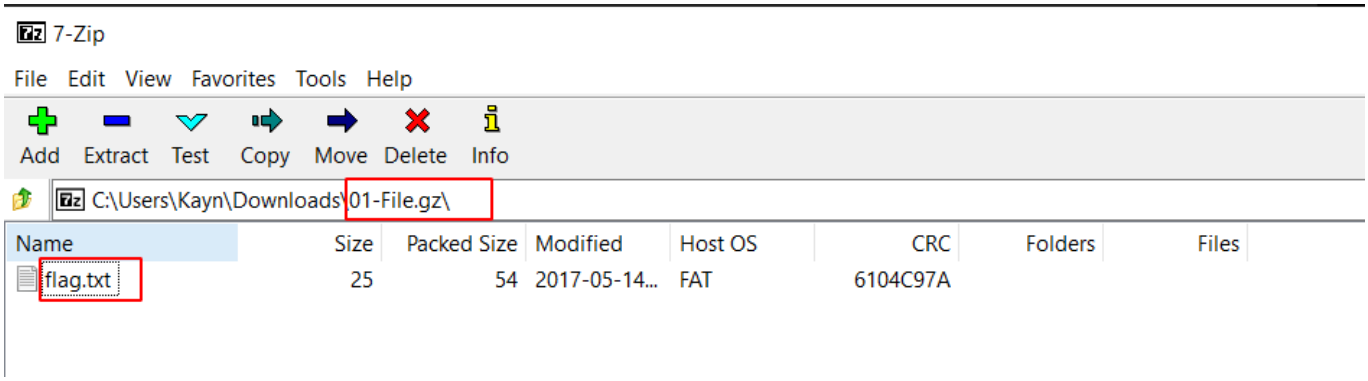
Flag

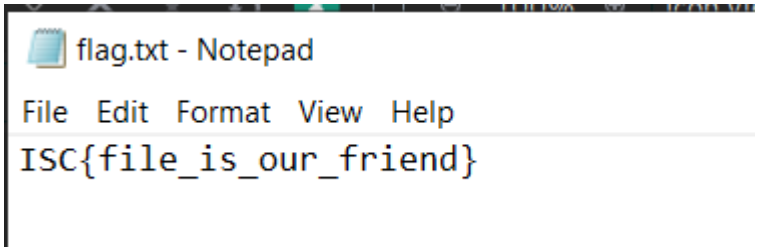
ISC{sending_passwords_in_cleartext_is_not_smart}

01 - File

```
└──╼ $file 01-File
01-File: gzip compressed data, was "flag.txt", last modified: Sun May 14
01:09:57 2017, max compression, from FAT filesystem (MS-DOS, OS/2, NT),
original size modulo 2^32 25
```

Astfel, am redenumit fisierul pentru a adauga extensia **gz** si l-am uploadat pe un host de Windows pentru a vedea flagul.





Flag

ISC{file_is_our_friend}

02 - Hidden 1

```
└── $strings 02-Hidden\ 1.png | grep ISC
%tEXtdate:ISC{we_all_love_grep}59:18+02:00
```

Flag

ISC{we_all_love_grep}

03 - Corrupted.jpg

Trebuie sa reconstruim headerul fisierului cu extensia JPEG pentru a fi unul valid.

file-recovery.com/jpg-signature-format.htm

OSEP Cryptomarket CodWer SS64 static-binaries Cipher Identifier DCTF-Writeups

Home Features Download Order Guide Support Contacts

JPEG/EXIF is the most common image format used by digital cameras and other image capture devices.

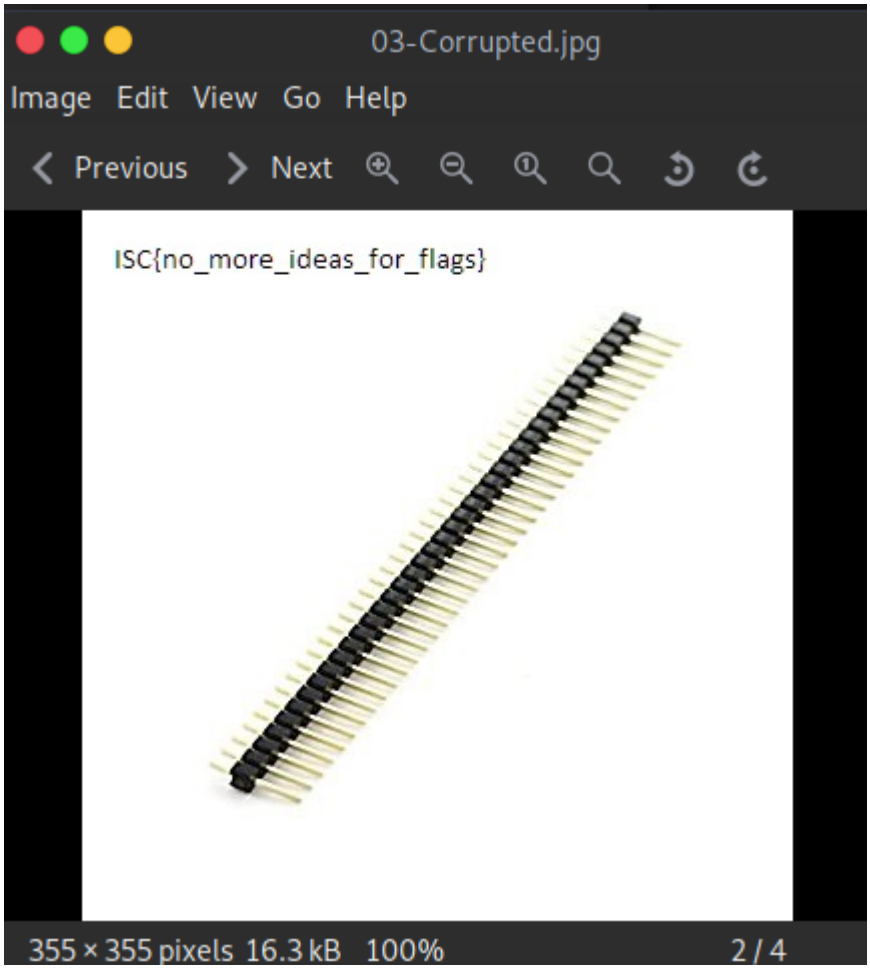
JPEG/JFIF, it is the most common format for storing and transmitting photographic images on the Internet.

JPEG files (compressed images) start with an image marker which always contains the marker code hex values **FF D8 FF**. It does not have a length of the file embedded, thus we need to find JPEG trailer, which is **FF D9**.

Let's examine the example

When inspecting example.jpg file's binary data using any Hex Viewer, like Active@ Disk Editor we can see it starts with a signature **FF D8 FF**:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	48	яШяа..JFIF.....H
00000010	00	48	00	00	FF	DB	00	43	00	01	01	01	01	01	01	01	.H..яЫ.C........
00000020	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....
00000030	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....
00000040	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....
00000050	01	01	01	01	01	01	01	01	01	FF	DB	00	43	01	01	01	.........яЫ.C...
00000060	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....
00000070	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....
00000080	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	....

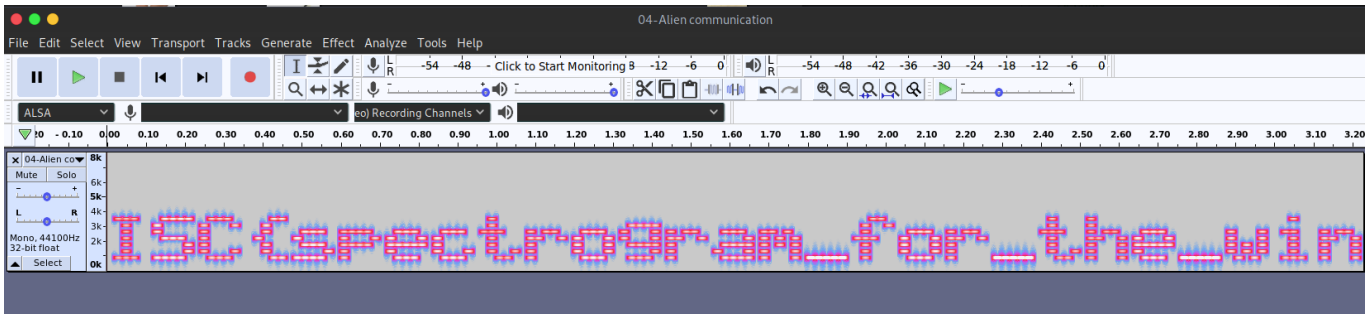


Flag

ISC{no_more_ideas_for_flags}

04 - Alien communication

Trebuie sa analizam spectograma aferenta fisierului **.wav** folosind Audacity.



Flag

ISC{spectogram_for_the_win}

05 - Idea

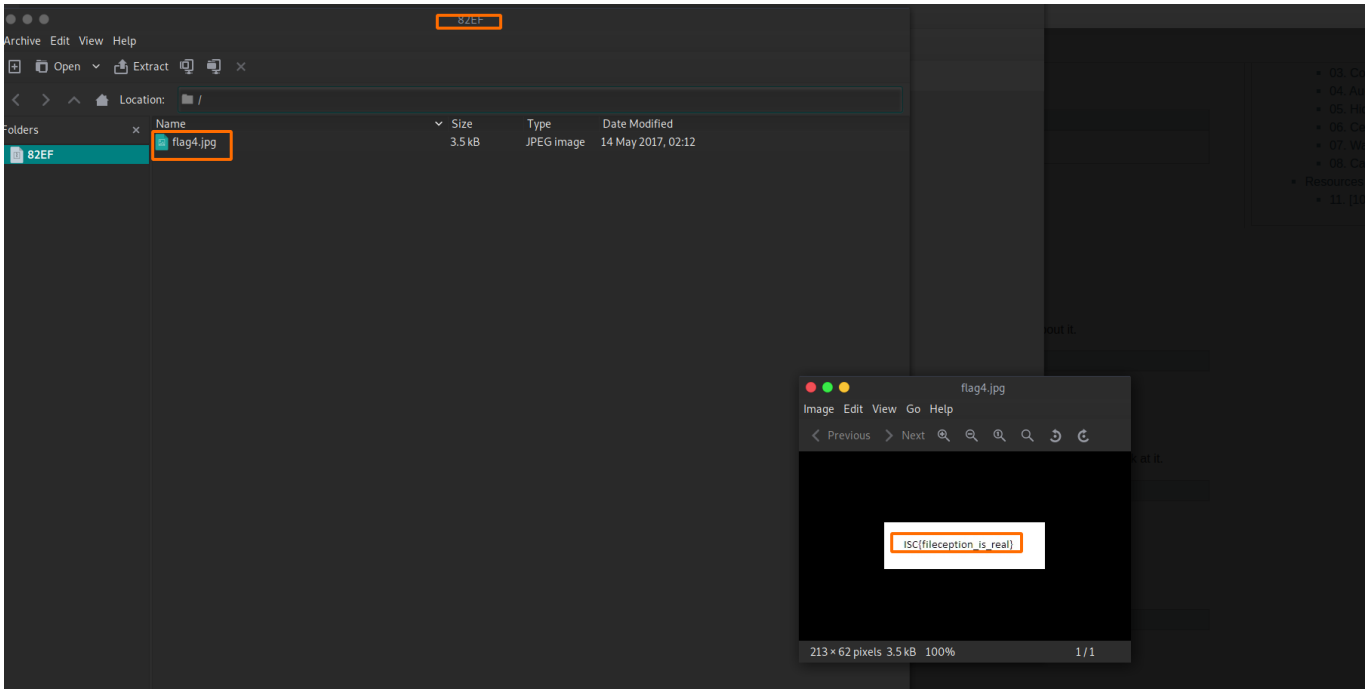
```
┌─[kayn@parrot]─[~/Documents/ISC/lab09]
└──╼ $binwalk -e 05-Idea.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
-
0             0x0             JPEG image data, JFIF standard 1.01
33519         0x82EF          7-zip archive data, version 0.4
```

```
└──╼ $binwalk --dd='.*' 05-Idea.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
-
0             0x0             JPEG image data, JFIF standard 1.01
33519         0x82EF          7-zip archive data, version 0.4

┌─[kayn@parrot]─[~/Documents/ISC/lab09]
└──╼ $ls
'00-Capture 1.pcap'   01-File  '02-Hidden 1.png'   03-Corrupted.jpg  '04-Alien
communication.wav'   05-Idea.jpg   _05-Idea.jpg.extracted   06-Letter.pdf   07-
Dumb.gif  '08-Capture 2.pcap'   output
┌─[kayn@parrot]─[~/Documents/ISC/lab09]
└──╼ $cat _05-Idea.jpg.extracted/
0     82EF
```

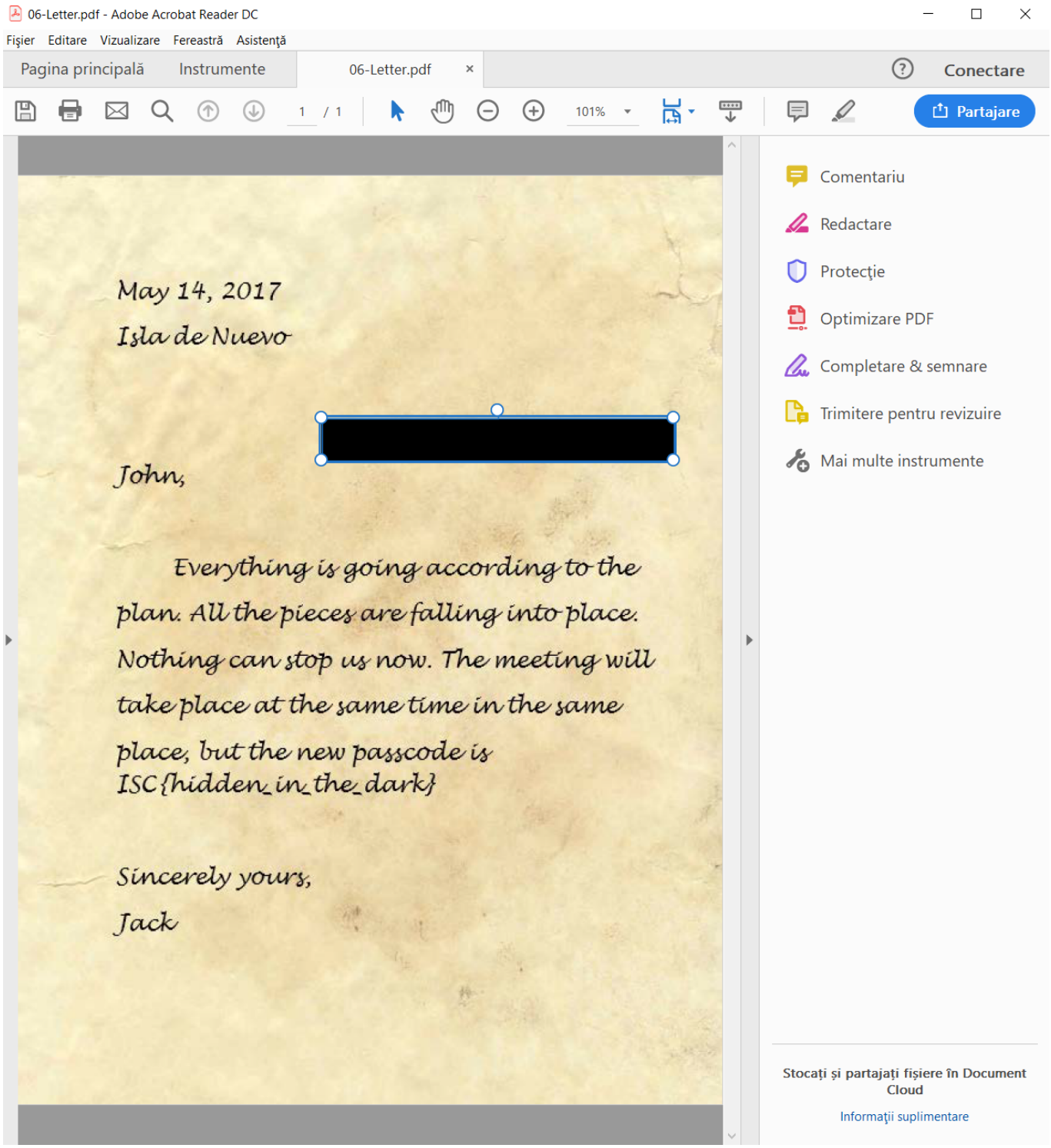


Flag

ISC{fileception_is_real}

06 - Censored

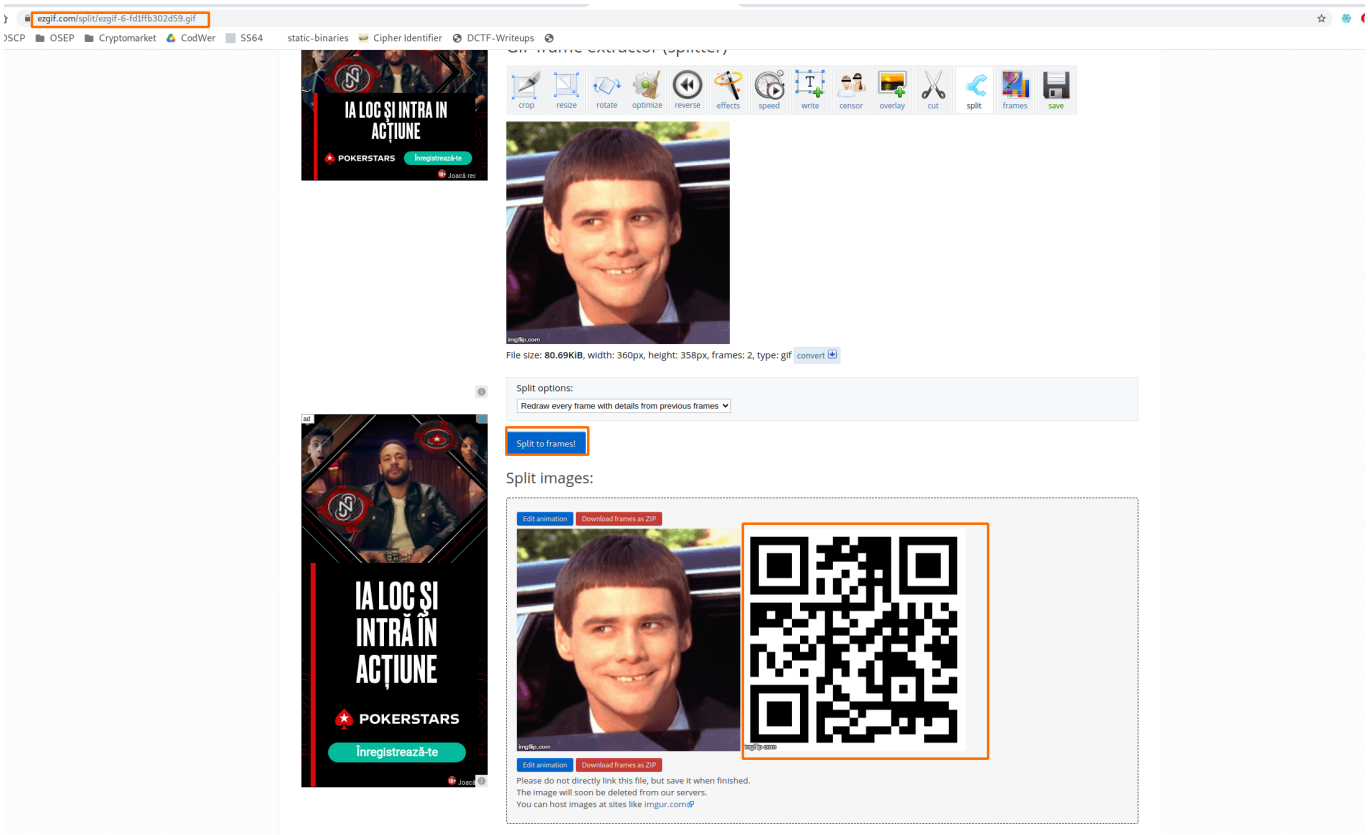
Folosind un program precum Adobe Acrobat Reader, putem muta chenarul negru.



Flag

ISC{hidden_in_the_dark}

07 - Dumb



Daca scanam codul QR, primim automat flagul.

Flag

ISC{what_were_you_waiting_for}

08 - Capture 2

github.com/TeamRocketIst/ctf-usb-keyboard-parser

SCP OSEP Cryptomarket CodWer SS64 static-binaries Cipher Identifier DCTF-Writeups

usbkeyboard.py — Fixing an issue that prevented some output to be shown when there is … — 25 days ago

README.md

ctf-usb-keyboard-parser

This is the updated script from https://teamrocketist.github.io/2017/08/29/Forensics-Hackit-2017-USB-ducker/

Usage

```
$ python usbkeyboard.py <file>
```

Extract file from pcap (might not work for every pcap)

```
$ tshark -r ./usb.pcap -Y 'usb.capdata && usb.data_len == 8' -T fields -e usb.capdata > usbPcapData
```

Some versions of tshark don't add ":" between each byte like this:

```
$ tshark -r ./usb.pcap -Y 'usb.capdata && usb.data_len == 8' -T fields -e usb.capdata
0000240000000000
0000000000000000
...
```

If this happens you can use sed to add them like this:

```
shark -r ./usb.pcap -Y 'usb.capdata && usb.data_len == 8' -T fields -e usb.capdata | sed 's/../:&/
30:24:00:00:00:00:00:00
30:00:00:00:00:00:00:00
```

Packages

No packages published

Contributors 2

ada-l0velace

godspeedcurry godspeedcurry

Languages

Python 100.0%

```
┌─[kayn@parrot]─[~/Documents/ISC/lab09/ctf-usb-keyboard-parser]
└──╼ $tshark -r ../08-Capture\ 2.pcap -Y 'usb.capdata && usb.data_len == 8' -T fields -e usb.capdata | sed 's/../:&/g2' > usbPcapData
┌─[kayn@parrot]─[~/Documents/ISC/lab09/ctf-usb-keyboard-parser]
└──╼ $cat usbPcapData
20:00:00:00:00:00:00:00
20:00:0c:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:16:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:06:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:2f:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:0e:00:00:00:00:00
00:00:00:00:00:00:00:00
```

```
00:00:08:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:1c:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:06:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:04:00:00:00:00:00
00:00:00:00:00:00:00:00
00:00:13:00:00:00:00:00
00:00:00:00:00:00:00:00
20:00:00:00:00:00:00:00
20:00:30:00:00:00:00:00
20:00:00:00:00:00:00:00
00:00:00:00:00:00:00:00
┌─[kayn@parrot]─[~/Documents/ISC/lab09/ctf-usb-keyboard-parser]
└──╼ $python2 usbkeyboard.py usb
usbkeyboard.py  usbPcapData
┌─[kayn@parrot]─[~/Documents/ISC/lab09/ctf-usb-keyboard-parser]
└──╼ $python2 usbkeyboard.py usb
usbkeyboard.py  usbPcapData
┌─[kayn@parrot]─[~/Documents/ISC/lab09/ctf-usb-keyboard-parser]
└──╼ $python2 usbkeyboard.py usbPcapData
ISC{keycap}┌─[kayn@parrot]─[~/Documents/ISC/lab09/ctf-usb-keyboard-parser]
```

Flag

ISC{keycap}