2016

ZU
YD

MICT1 – Exercise Week 8

Group 1

Zuyd University

4/5/2016

MICT1 – Exercise Week 8

Reverse Engineering Data – Exercise week 8

Group	Group 1	
Students	Delano Cörvers	(1306669)
	Davy Heutmekers	(1309730)
	Rik Kierkels	(1354442)
Module	MICT1 – Reverse Engineering Data	
Assignment	Exercise – Week 8	
School year	2015 – 2016	

Table of content

1 What we found

We found the Star Wars related message “These are your first steps…” in the following image.

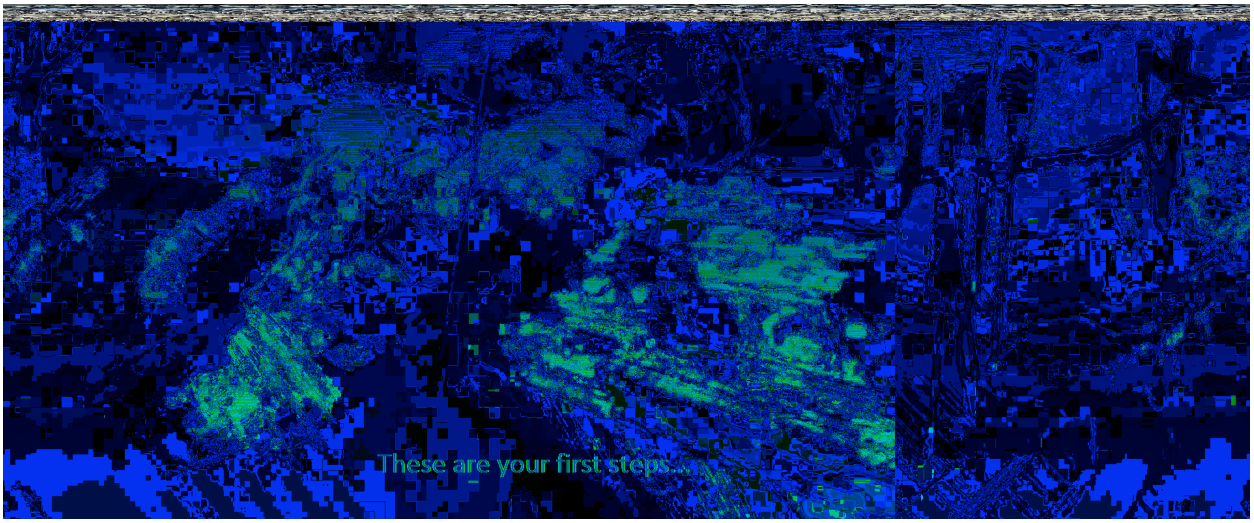


Figure 1: modified.zif with Star Wars related message

2 How we found it

2.1 File Structure

After analyzing the provided source.zif file, we noticed a structure within the hex code of the file.

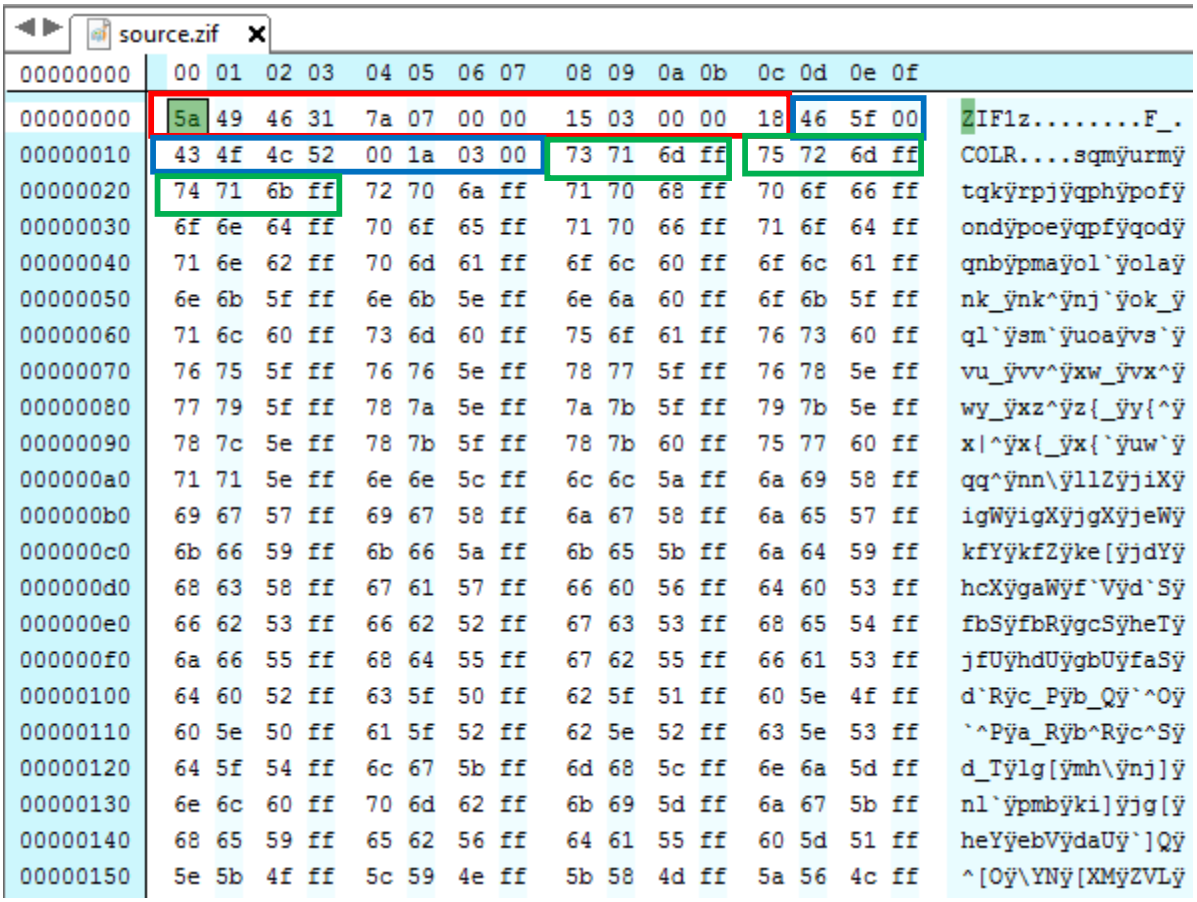


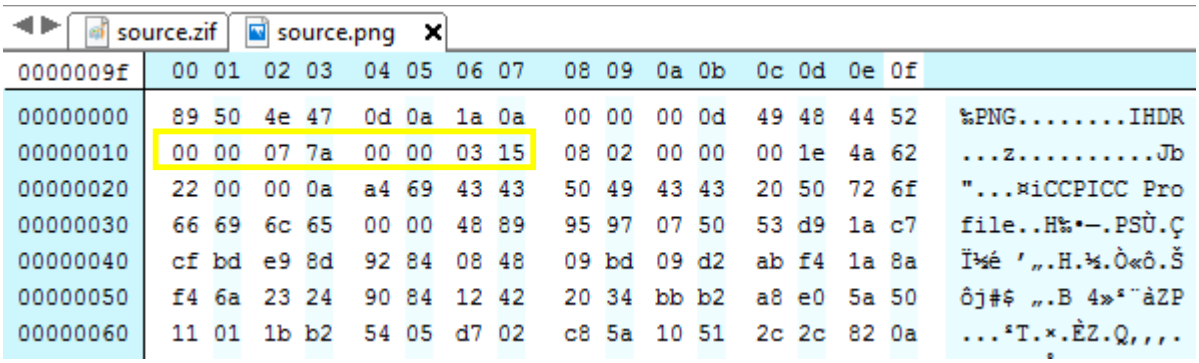
Figure 2: First bytes of the provided source.zif file

The red block in Figure 1 consists of the following elements.

Description	Length (bytes)	Value
File header	4	ZIF1
Width	4	7a 07 00 00
Length	4	15 03 00 00
?	1	18

Table 1: Data found in the ZIF1 header, red selected data block in Figure 1

When we examine the PNG header we find the following width and height values.


source.zif source.png

0000009f	00 01 02 03	04 05 06 07	08 09 0a 0b	0c 0d 0e 0f	
00000000	89 50 4e 47	0d 0a 1a 0a	00 00 00 0d	49 48 44 52	‰PNG........IHDR
00000010	00 00 07 7a	00 00 03 15	08 02 00 00	00 1e 4a 62	...z..........Jb
00000020	22 00 00 0a	a4 69 43 43	50 49 43 43	20 50 72 6f	"...¤iCCPICC Pro
00000030	66 69 6c 65	00 00 48 89	95 97 07 50	53 d9 1a c7	file..H‰•—.PSÙ.Ç
00000040	cf bd e9 8d	92 84 08 48	09 bd 09 d2	ab f4 1a 8a	Ï½é ’„.H.½.Ò«ô.Š
00000050	f4 6a 23 24	90 84 12 42	20 34 bb b2	a8 e0 5a 50	ôj#$ „.B 4»²¨àZP
00000060	11 01 1b b2	54 05 d7 02	c8 5a 10 51	2c 2c 82 0a	...²T.×.ÈZ.Q,,‚.

Figure 3: First bytes of the provided source.png file

Description	Length (bytes)	Value
Width	4	00 00 07 7a
Length	4	00 00 03 15

Table 2: Data found in the PNG header, yellow selected block in Figure 2

After converting the found PNG values to decimal we find a width of 1918 and a height of 789. These values correspond with the dimensions of the image after visual examination. When we compare the found width and length values from the PNG header to the values found in the ZIF1 header we can conclude that ZIF1 is **little-endian formatted**.

The blue block contains a chunk identifier with the following elements.

Description	Length (bytes)	Value	ASCII
File header	7	46 5f 00 43 4f 4c 52	F_.COLR
?	2	00 1a	
?	2	03 00	

Table 3: Data found in the blue selected data block in Figure 1

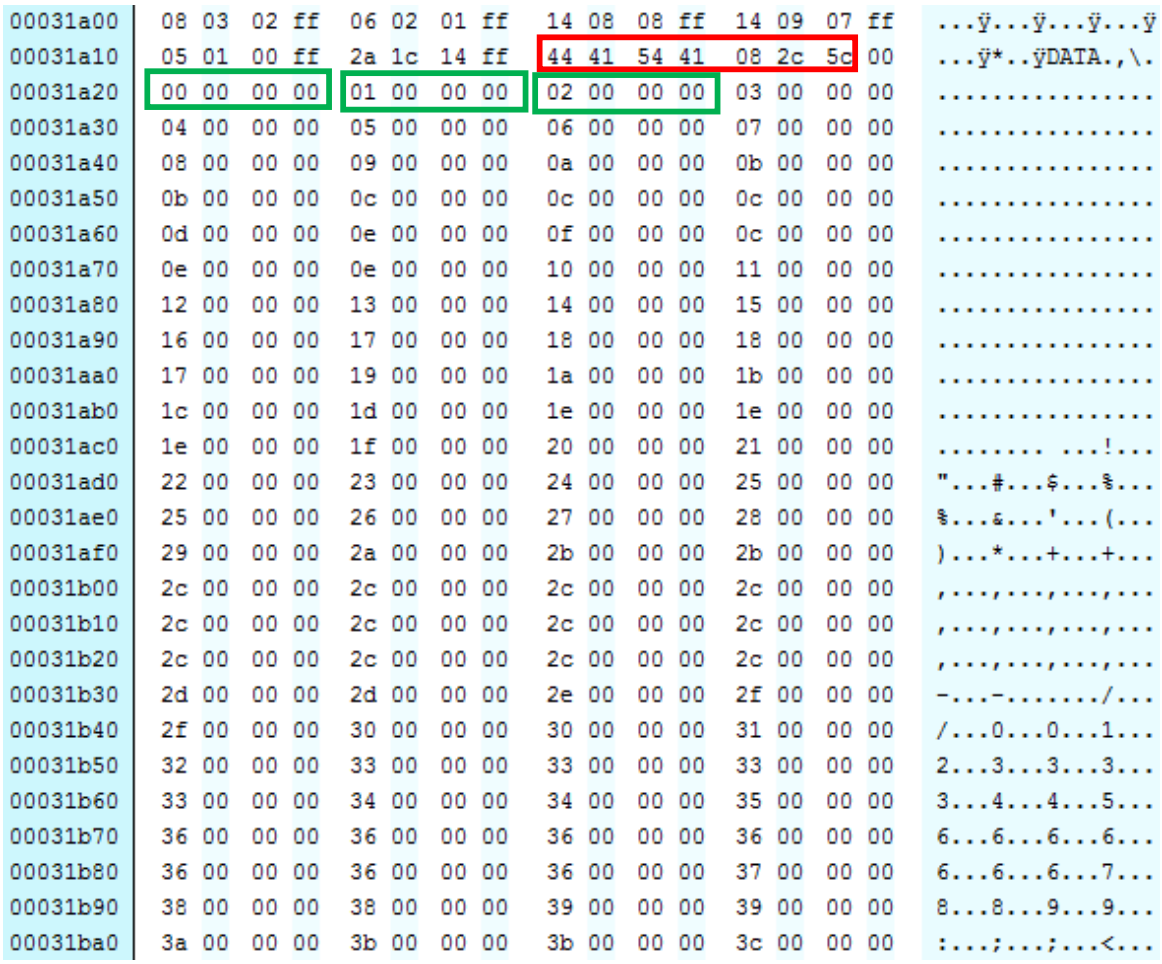
The green block contains a data structure with the following recurring elements.

Description	Length (bytes)	Value	Possible Expl.
Data block	4	73 71 6d ff	RR GG BB AA

Table 4: Data found in the green selected data block in Figure 1

The data blocks found following the chunk type “F_.COLR” always use the same 4-byte structure, we assume they are RGB values for the pixels. The following format seems logical: Byte 1: RR (Red Red), Byte 2: GG (Green Green), Byte 3: BB (Blue Blue), Byte 4: ff (this is either a terminator or a value used for the alpha channel).

The green blocked data structure found in Figure 1 continues for 203,275 bytes, the file then continues with a different structure.



00031a00	08	03	02	ff	06	02	01	ff	14	08	08	ff	14	09	07	ff	...ÿ...ÿ...ÿ...ÿ
00031a10	05	01	00	ff	2a	1c	14	ff	44	41	54	41	08	2c	5c	00	...ÿ*..ÿDATA.,\.
00031a20	00	00	00	00	01	00	00	00	02	00	00	00	03	00	00	00	....
00031a30	04	00	00	00	05	00	00	00	06	00	00	00	07	00	00	00	....
00031a40	08	00	00	00	09	00	00	00	0a	00	00	00	0b	00	00	00	....
00031a50	0b	00	00	00	0c	00	00	00	0c	00	00	00	0c	00	00	00	....
00031a60	0d	00	00	00	0e	00	00	00	0f	00	00	00	0c	00	00	00	....
00031a70	0e	00	00	00	0e	00	00	00	10	00	00	00	11	00	00	00	....
00031a80	12	00	00	00	13	00	00	00	14	00	00	00	15	00	00	00	....
00031a90	16	00	00	00	17	00	00	00	18	00	00	00	18	00	00	00	....
00031aa0	17	00	00	00	19	00	00	00	1a	00	00	00	1b	00	00	00	....
00031ab0	1c	00	00	00	1d	00	00	00	1e	00	00	00	1e	00	00	00	....
00031ac0	1e	00	00	00	1f	00	00	00	20	00	00	00	21	00	00	00	....!...
00031ad0	22	00	00	00	23	00	00	00	24	00	00	00	25	00	00	00	"...#...$...%...
00031ae0	25	00	00	00	26	00	00	00	27	00	00	00	28	00	00	00	%...&...'...(...
00031af0	29	00	00	00	2a	00	00	00	2b	00	00	00	2b	00	00	00	)...*...+...+...
00031b00	2c	00	00	00	2c	00	00	00	2c	00	00	00	2c	00	00	00	,...,...,...,...
00031b10	2c	00	00	00	2c	00	00	00	2c	00	00	00	2c	00	00	00	,...,...,...,...
00031b20	2c	00	00	00	2c	00	00	00	2c	00	00	00	2c	00	00	00	,...,...,...,...
00031b30	2d	00	00	00	2d	00	00	00	2e	00	00	00	2f	00	00	00	-...-...../...
00031b40	2f	00	00	00	30	00	00	00	30	00	00	00	31	00	00	00	/...0...0...1...
00031b50	32	00	00	00	33	00	00	00	33	00	00	00	33	00	00	00	2...3...3...3...
00031b60	33	00	00	00	34	00	00	00	34	00	00	00	35	00	00	00	3...4...4...5...
00031b70	36	00	00	00	36	00	00	00	36	00	00	00	36	00	00	00	6...6...6...6...
00031b80	36	00	00	00	36	00	00	00	36	00	00	00	37	00	00	00	6...6...6...7...
00031b90	38	00	00	00	38	00	00	00	39	00	00	00	39	00	00	00	8...8...9...9...
00031ba0	3a	00	00	00	3b	00	00	00	3b	00	00	00	3c	00	00	00	:...;...;...<...

Figure 4: Data structure in the provided file source.zif

The red selected data block in figure 3 contain the chunk type and the green selected data blocks in figure 3 contain values that we assume to be offsets for the pixels used in the COLR table in figure 2. The green selected data blocks seem to have a recurring format of 4 bytes.

2.2 Photoshop conversion

We made the assumption that the file would be edited using Adobe Photoshop, so we decided to try and convert the provided source.png file to a different file type with different encoding settings. Below is a table of the different formats and settings we used.

File Type	Settings
BMP	8-bit
BMP	16-bit
BMP	32-bit
BMP	32-bit + alpha channel
TIF	32-bit conversion
BMP	Labcolors
BMP	Indexcolors

Table 5: Trial and errors settings adobe PS

We saved all of the converted PNG files and copied their headers over the ZIF file header in the provided modified.zif file. We used the header of the PNG file that was converted to BMP, saved as 32-bit with flipped rows. The header of the modified.zif file was changed to the header in figure 4.

Blue Image BMP Header.bmp

000000cb	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	42	4d	40	2c	5c	00	00	00	00	00	36	00	00	00	28	00	BM@,\.....6...(.
00000010	00	00	7a	07	00	00	eb	fc	ff	ff	01	00	20	00	00	00	..z...ëüÿÿ.. ...
00000020	00	00	0a	2c	5c	00	12	0b	00	00	12	0b	00	00	00	00	...,\...........
00000030	00	00	00	00	00	00	73	71	6d	ff	75	72	6d	ff	74	71	......sqmÿurmÿtq
00000040	6b	ff	72	70	6a	ff	71	70	68	ff	70	6f	66	ff	6f	6e	kÿrpjÿqphÿpofÿon
00000050	64	ff	70	6f	65	ff	71	70	66	ff	71	6f	64	ff	71	6e	dÿpoeÿqpfÿqodÿqn
00000060	62	ff	70	6d	61	ff	6f	6c	60	ff	6f	6c	61	ff	6e	6b	bÿpmaÿol`ÿolaÿnk
00000070	5f	ff	6e	6b	5e	ff	6e	6a	60	ff	6f	6b	5f	ff	71	6c	_ÿnk^ÿnj`ÿok_ÿql
00000080	60	ff	73	6d	60	ff	75	6f	61	ff	76	73	60	ff	76	75	`ÿsm`ÿuoaÿvs`ÿvu
00000090	5f	ff	76	76	5e	ff	78	77	5f	ff	76	78	5e	ff	77	79	_ÿvv^ÿxw_ÿvx^ÿwy
000000a0	5f	ff	78	7a	5e	ff	7a	7b	5f	ff	79	7b	5e	ff	78	7c	_ÿxz^ÿz{_ÿy{^ÿx|
000000b0	5e	ff	78	7b	5f	ff	78	7b	60	ff	75	77	60	ff	71	71	^ÿx{_ÿx{`ÿuw`ÿqq

Figure 5: Modified.zif with BMP 32-bit flipped rows header

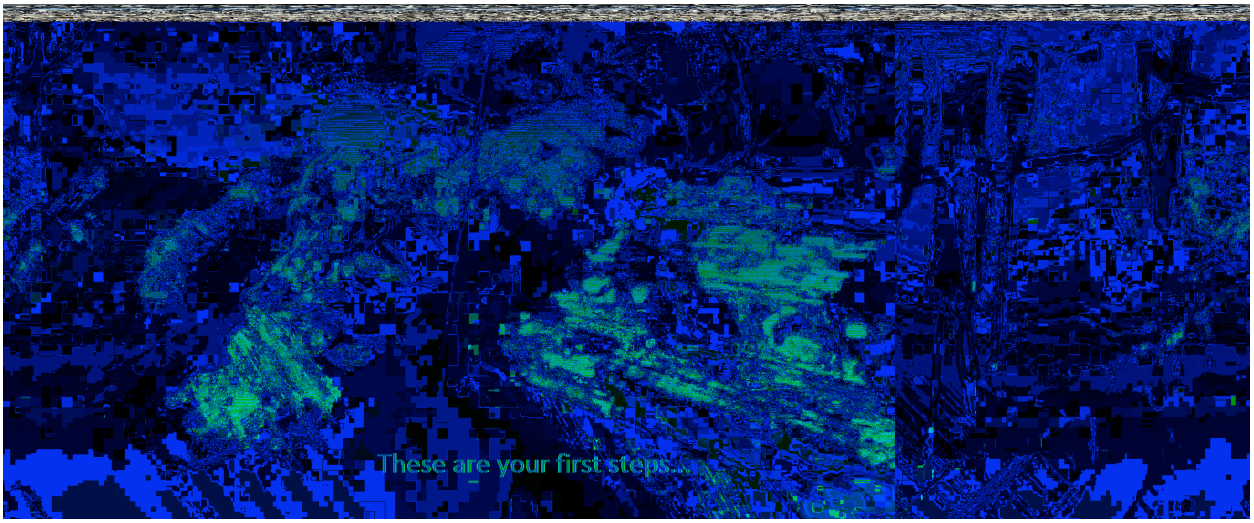


Figure 6: Retrieved image after converting modified.zif

When opened the file looks like figure 5. We can conclude that the RGB values are incorrect, we mainly see blue colors and green colors. However, we can retrieve the Star Wars related message from the Image.

Because the generated images were all Blue, Red or Green we thought the RGBA-values in the color palette might have been flipped to ABGR-values. We wrote a little script to flip every 4 bytes e.g. 01 02 03 04 would result in 04 03 02 01. To achieve this we wrote the script below.

```
inputFile = open('toBeReversed.zif', 'rb+')
outputFile = open('reversed.zif', 'wb+')
while 1:
    data = inputFile.read(4)
    if not data:
        break
    tempData = []
    index = 3
    while index >= 0:
        tempData.append(data[index])
        index -= 1

    outputFile.write(bytes(tempData))
```

After reversing the color palette we've overwritten the existing color palette in the modified.zif file. Unfortunately this resulted in a corrupt BMP-file.

3 Where we found it

The retrieved message was found in the modified.zif image after we changed the ZIF header to a BMP header.

4 Tools

Hex Editor Neo

Hex Editor Neo is a free hex editor tool which is optimized to work with large files.

http://www.hhdsoftware.com/free-hex-editor

Notepad++

Notepad++ is a free text editor which supports several plugins, which can be downloaded in the application.

https://notepad-plus-plus.org/

HEX-Editor

The HEX-Editor plugin converts the selected text into a HEX view.

Adobe Photoshop

Adobe Photoshop is a raster graphics editor developed and published by Adobe Systems for Windows and OS X. Adobe Photoshop can be used to edit and changes images.

https://www.adobe.com/nl/products/photoshop.html