## NZCSC 2020 Solutions

July 2020

Theme: Cryptography – Caesar cipher

Tools used:

- https://www.dcode.fr/caesar-cipher
- https://www.rapidtables.com/convert/number/decimal-to-hex.html



### Solution:

1. Find the most suitable text which is (+14 shift), that gives us the first part of the flag osnkxk

DCODE	CAESAR CIPHER	Lire en Français
	Cryptography > Substitution Cipher > Caesar Cipher	Summary 🗉
Search for a tool	CAESAR CIPHER DECODER	🖌 Caesar Cipher Decoder
* SEARCH A TOOL ON DCODE BY KEYWORDS: e.g. type scrabble GO	Kszącas hc hvwg ącadshwhwcb! Vsfs wg hvs hslh hc hvs twfgh vozt ct hvs qvozzsbus: cgbyly	<ul> <li>Caesar Encoder</li> <li>How to encrypt using Caesar cipher?</li> </ul>
Results 🖱 🛎 🛎 🛎		• How to decrypt Caesar
Brute-Force mode: all shifts are tested, text is limited to the a few hundr of characters. To find the full text back with punctuation and space, plea indicate the correct shift found (+xx) in the form.	KNOWING THE SHIFT:     TEST ALL POSSIBLE SHIFTS (BRUTE-FORCE ATTACK)	cipher? How to recognize Caesar ciphertext? How to decipher Caesar
11 11	DECRYPT CAESAR CODE	without knowing the shift?
Welcome to this competition! Here is +14 the text to the first half of the challenge: osnkxk	See also: ROT Cipher — Shift Cipher WITH A CUSTOM ALPHABET	<ul> <li>What are the variants of the Caesar cipher?</li> <li>How to encrypt digits and</li> </ul>
Qyfwigy ni nbcm wigjyncncih! Byly cm +20 nby nyrn ni nby zclmn bufz iz nby wbuffyhay: imhere	ALPHABET ABCDEFGHIJKLINNOPQRSTUVWXYZ      USE THE ASCII TABLE AS ALPHABET      DECRYPT	numbers using Caesar cipher? Why the name Caesar Cipher?
Ltardbt id iwxh rdbetixixdc! Wtgt xh		😠 What is August Cipher?

2. Next, convert the negative decimal number to a hexa decimal, signed 2's complement representation, giving us the second part of the flag  $\tt FC6A85$ 

## Decimal to Hexadecimal converter

From		То	
Decimal	•	Hexadecimal	-
Enter decimal nu	umber:		
-234875			10
Convert	🗙 Reset 🛛 1	Swap	
Hex number:			
-3957B			16
Hex signed 2's o	complement:		
FFFC6A85			16

- 3. Get the last 6 digits to form the second part of the 12-character flag
- 4. Concatenate this second part of the flag with the first part

Flag:

flag:osnkxkFC6A85

Theme: Steganography

## Tool used:

- ExifTool: https://exiftool.org/
- unzip: https://linux.die.net/man/1/unzip

This challenge has a link to the file Finale.gif.



#### Solution:

1. Open the gif in ExifTool.

r00t:~\$ exiftool finale.gif		
ExifTool Version Number	10.80	
File Name	finale.gif	
Directorv		
File Size	173 kB	
File Modification Date/Time	2020:07:22	14:34:59+12:00
File Access Date/Time	2020:07:22	14:35:19+12:00
File Inode Change Date/Time	2020:07:22	14:35:12+12:00
File Permissions	<b>rw-rw-r</b>	
File Type	GIF	
File Type Extension	gif	
MIME Type	image/gif	
GIF Version	89a	
Image Width	220	
Image Height	193	
Has Color Map	Yes	
Color Resolution Depth	8	
Bits Per Pixel	8	
Background Color	240	
Animation Iterations	Infinite	
Comment	imgjpeg	
Frame Count	11	
Duration	1.63 s	
Image Size	220x193	
Megapixe <u>l</u> s	0.042	
r00t:~\$		

- 2. In the comment section, there is a hint about the existence of an imgjpeg
- 3. Unzip the GIF to get an img.jpeg
- 4. Examine the img.jpeg using Exiftool

r00t:~\$ unzip finale.gif Archive: finale.gif		
warning [finale.gif]: 165232 ex	t	ra bytes at beginning or within zipfile
(attempting to process anyway)		
inflating: img.ipeg		
r00t:~\$ exiftool ima.ipea		
ExifTool Version Number		10.80
File Name		ima.ipea
Directory		
File Size		11 kB
File Modification Date/Time		2019:12:19 14:50:47+13:00
File Access Date/Time		2019:12:19 14:56:40+13:00
File Inode Change Date/Time		2020:07:22 14:37:16+12:00
File Permissions		rw-rr
File Type		JPEG
File Type Extension		jpg
MIME Type		image/jpeg
JFIF Version		1.01
Exif Byte Order		Big-endian (Motorola, MM)
X Resolution		1
Y Resolution		1
Resolution Unit		None
Y Cb Cr Positioning		Centered
Exif Version		0232
Components Configuration		Y, Cb, Cr, -
Image History		f945cc88e1ee:galf
Flashpix Version		0100
Color Space		Uncalibrated
Image Width		300
Image Height		168
Encoding Process		Baseline DCT, Huffman coding
Bits Per Sample		8
Color Components		3
Y Cb Cr Sub Sampling		YCbCr4:4:4 (1 1)
Image Size		300x168
Megapixels		0.050
r00t:~\$		

5. The flag can be found in the Image History section, by reversing the order of the characters.

Flag:

flag:ee1e88cc549f

### Theme: Reverse Engineering

#### Tool: JSF\*ck

Participants are given an encoded file, which is encoded as a series of 6 characters: (`, `)', `!', `+', `[`, `]'



## Solution:

JSF\*ck is an online educational programming style based on the atomic parts of JavaScript. It uses only six different characters to write and execute code. Alternatively, one could write a script to decode the file.

JSF*ck						
JSF*ck is an esoteric and educational progra atomic parts of JavaScript. It uses only si write and execute code.	amming style based on the x different characters to					
It does not depend on a browser, so you can	even run it on Node.js.					
Use the form below to convert your own scri get back a plain string.	ot. Uncheck "eval source" to					
alert(1) Encode C Eval Se	burce					
<pre>[]]+(!![]+[])[+!+[]]+[])[!+[]+!+[]]+(!![]+[]](![]+[])[+[]]+ ([![]]+(])[])[+!+[]]+(!]]+(![]+[])[!+[]]+(!![]+[]]+(!![]]+ (])[!+[]+!+[]]+(!]]+(![]+[])[!+!]]+(!![]+[]])(!!]]+(!]]+(!]]+(!]]+(!]]+(!]]+(!]]+(</pre>						
[]]])[!+[]+!+[]+[+[]]]](!+[]+!+[]+[+[]])+[+ 896 chars "flag:d382557fdab1"						
Links	-					
<ul> <li>Share on <u>Twitter, Google+</u></li> <li>View source on <u>GitHub</u></li> <li>Follow <u>@aemkei</u> (Martin Kleppe)</li> <li>Original discussion at <u>Sla.ckers.org</u></li> </ul>	ОК					

After running the code through the online tool, the flag can pops up as an alert.

## Flag:

flag:d382557fdab1

Theme: Network Traffic Analysis

Tools:

• https://www.wireshark.org/

We intercepted this communication. We believe it has something you are looking for.

Click here for the File.



File	Edit View Go G	Capture Analyze Stat	istics Telephony <u>W</u> ire	ess <u>T</u> ools	Help				
	. 20 🖿	। 🗋 🖹 🎑 🤇	🔶 🔿 🖀 🚡 :	<u>ا ا</u>	● •	् 🎹			
Ap	ply a display filter	. <ctrl-></ctrl->					🖬 •	Expression	ŀ
No	Time	Source	Dectination	Protocol L	anath Info				
140.	1 0 000000	102 169 0 121	222 12 02 22	HOD	47 20222 -	- 20222 Lor	n=6		
r.	2 0 022405	102 168 0 122	222 12 02 22	LIDD	47 20232 -	- 20232 Lor			
	2 9.033493	102 100 0 123	232.12.32.32	UDD	60 20222	- 29232 Lei	1-0 -0		
	A 22 720222	102 100 0 121	232.12.32.32	UDD	106 20202 -	20232 Lei	1-20 n=454		
	5 20 972205	102 100 0 122	232.12.52.32	UDD	161 20232 -	20232 Lei	1-104 0-110		
	6 34 071230	102 100 0 123	232.12.02.32	UDD	157 20232 -	20232 Ler	1-240 n=445		
	7 41 411225	102 168 0 122	222 12 02 22	LIDD	70 20232	20232 Ler	1744M n=97		
	9 50 695916	102 168 0 121	222 12 02 22	LIDD	54 20232	20232 Lor	1997 n=10		
	9 57 785395	102 168 0 123	232 12 02 32	LIDD	99 20232	20232 Ler	1746 n-40		
	10 66 455828	192 168 0 131	232 12 02 32	UDP	53 29232	29232 Ler			
	11 74 869875	192 168 0 123	232 12 92 32	UDP	57 29232 -	29232 Ler	1744 n=15		
1	12 83.643109	192,168,0,131	232.12.92.32	UDP	76 29232 -	- 29232 Ler	1184		
	13 93,522519	192,168,0,123	232.12.92.32	UDP	84 29232 -	- 29232 Ler	n=42		
	14 99, 192410	192.168.0.131	232.12.92.32	UDP	54 29232 -	- 29232 Ler	-12		
	15 105,353783	192,168,0,131	232.12.92.32	UDP	626 29232 -	- 29232 Ler	n=584		
	16 114,618969	192.168.0.131	232.12.92.32	UDP	74 29232 -	- 29232 Ler	n=32		
	17 123,759984	192.168.0.123	232.12.92.32	UDP	74 29232 -	- 29232 Ler	n=32	_	
	18 133.625523	192.168.0.131	232.12.92.32	UDP	138 29232 -	- 29232 Ler	n=96		
	19 142.666756	192.168.0.123	232.12.92.32	UDP	202 29232 -	- 29232 Ler	n=160		1
	20 152.489327	192.168.0.131	232.12.92.32	UDP	106 29232 -	- 29232 Ler	n=64		
	21 160.039901	192.168.0.131	232.12.92.32	UDP	106 29232 -	<ul> <li>29232 Ler</li> </ul>	n=64		
L	22 168.024333	192.168.0.131	232.12.92.32	UDP	106 29232 -	- 29232 Ler	n=64		
	23 176.791193	192.168.0.123	232.12.92.32	UDP	74 29232 -	<ul> <li>29232 Ler</li> </ul>	n=32		
	24 157346.214637	192.168.0.1	255.255.255.255	UDP	215 36966 -	<ul> <li>7437 Len:</li> </ul>	=173		
	25 157346.541296	192.168.0.131	192.168.0.156	TCP	176 34536 -	- 8009 [PSF	H, ACK] Seq=1 Ack=1 Win=321 Len=110 TSval=2524009097 TSecr=2565328 [TCP segment of a reassembled PDU]		
	26 157346.542860	192.168.0.156	192.168.0.131	TCP	176 8009 -	34536 [PSF	H, ACK] Seq=1 Ack=111 Win=243 Len=110 TSval=2565829 TSecr=2524009097 [TCP segment of a reassembled PDU]		
	27 157346.542893	192.168.0.131	192.168.0.156	TCP	66 34536 -	8889 [ACI	K] Seq=111 Ack=111 Win=321 Len=0 TSval=2524009099 TSecr=2565829		
	28 157347.381968	162.159.136.234	192.168.0.131	TLSv1.2	155 Applica	ition Data			
	29 157347.382013	192.168.0.131	162.159.136.234	TCP	54 33744 -	443 [ACK]	] Seq=1 Ack=102 Win=2227 Len=0		
	30 157347.386456	162.159.136.234	192.168.0.131	TLSv1.2	115 Applica	tion Data			Ŧ
Fra	ame 12: 76 bytes	on wire (608 bits),	76 bytes captured (60	8 bits)					
Eth	hernet II, Src: G	iga-Byt_9f:f5:63 (1c	:1b:0d:9f:f5:63), Dst	: IPv4mcast	0c:5c:20 (8	1:00:5e:0c	::5c:20)		
Int	ternet Protocol V	ersion 4, Src: 192.10	68.0.131, Dst: 232.12	.92.32					
Use	er Datagram Proto	col, Src Port: 29232,	, Dst Port: 29232						
Dat	ta (34 bytes)								
	01 00 5e 0c 5c 2	0 1c 1b 0d 9f f5 63	08 00 45 00 ···^·	· · · · c · · E ·					
0010	69 3e 7h 6c 49 9	0 01 11 f8 ca c0 a8	88 83 e8 8c ->/18						
0020	5c 28 72 38 72 3	8 88 28 11 34 41 66	79 77 61 79 \ r8r8-	44000000					
6636	2c 28 6d 79 28 6	b 65 79 28 69 73 28	66 6c 61 67 . my ke	v is flag			— Whoons the flag is here this was supposed to be		
0040	3a 37 32 34 35 6	5 31 36 35 34 65 37	:7245e1	6 54e7	-		moops, the hag is herein this has supposed to be		
0040	04 01 02 04 00 0	0 01 00 00 04 00 01	1124043	0 0401			an appropriate key, not the flag. Guass it's a freehie i)		
							an encryption key, not the hage Guess it's a neeple .)		
0.7	Pata (data) 34 k	autor					Backate: 24340 - Dieplanad: 24340 (100.04)	Profile: Defau	.14
	Data (data), 34 t	lytes					Packets: 24349 - Displayed: 24349 (100.0%)	Prome: Defau	л¥.

## Flag:

flag:7245e1654e7

Theme: Forensics

Tools:

- A hex-editor of your choice
- https://www.gimp.org/

This challenge has a link to the file img.xcf. This is a corrupted image file whose header has been modified.

We found this <u>file</u> but it doesn't seem to work.

#### Solution:

Open the img.xcf file using a hex-editor, and search for the correct file header for xcf file  $(67 \ 69 \ 6d \ 70 \ 20 \ 78 \ 63 \ 66 \ 20)$ . Replace the file header with the correct value.

Memory Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	θE	θF	De	eco	od€	ed	Tex	(t							
00000000	67	69	6D	70	20	78	63	66	20	76	30	31	31	00	00	00	g	i	m	р	Х	С	f	v	0	1	1		
00000010	19	C0	00	00	13	88	00	00	00	00	00	00	00	96	00	00		À											
00000020	00	11	00	00	00	01	01	00	00	00	13	00	00	00	08	43													С
00000030	96	00	00	43	96	00	00	00	00	00	14	00	00	00	04	00				С									
00000040	00	00	02	00	00	00	16	00	00	00	04	00	00	00	01	00													
00000050	00	00	15	00	00	01	47	00	00	00	0D	67	69	6D	70	2D						G				g	i١	n p	) -
00000060	63	6F	6D	6D	65	6E	74	00	00	00	00	01	00	00	00	12	с	0	m	m	e n	t							

Open the img.xcf file using Gimp to obtain the flag.

f/a	9:20.1
	-wo292+2.2
	~93/
	×

Flag:

flag:zwb2qdtzq314

Theme: Steganography

Tools:

- Sonic Visualiser: https://sonicvisualiser.org/
- steghide: https://github.com/StefanoDeVuono/steghide

This challenge has a file called FINAL.wav, which is an audio file containing a clue to find the flag.



#### Solution:

Download and analyse the audio file using Sonic Visualiser. When analysing the audio, take note of the phrase "toor".



导数	112	١ŀ,				ł,
		14				
				-		
		łE	F		Ŧ	
			1			

Next, open the command prompt and extract a secret text file. When prompted for a password, enter the phrase toor.

The secret text file contains the flag.

r00t:~\$ steghide extract -sf FINAL.wav
Enter passphrase:
wrote extracted data to "secret.txt".
r00t:~\$ cat secret.txt
flag:a37666fc86de
r00t:~\$

Flag:

flag:a37666fc86de

#### Theme: SQL Injection

#### Tools Used:

#### None

Recently, a security team discovered an organisation involved in cybercriminal activities. Someone has managed to find a UI to the database which the criminals store information in. In one of the database's tables, they noticed a known individual, identified by the first name, Andrew. However, their main goal is to find the flag which will be a vital clue for them to identify the suspect. Using the information above, help the security team find the flag!

#### Solution:

Perform SQL injection to the search box to get all the data from the table: ' or '1' = '1

Recently, a security team discovered an organisation involved in cybercriminal activities. Someone has managed to find a UI to the database which the criminals store information in. In one of the database's tables, they noticed a known individual, identified by the first name, Andrew. However, their main goal is to find the flag which will be a vital clue for them to identify the suspect.

Using the information above, help the security team find the flag!

Search: ' or	'1' = '1		Check
Name	Phone	Country	UID
Timothy	16370111 4112	Thailand	nhtljvcgrjnvf
Linus	16940409 5292	Peru	qeqkxnqtbtycq
Rogan	16950604 6508	Palau	jpdbopvaamlre

Recently, a security team discovered an organisation involved in cybercriminal activities. Someone has managed to find a UI to the database which the criminals store information in. In one of the database's tables, they noticed a known individual, identified by the first name, Andrew. However, their main goal is to find the flag which will be a vital clue for them to identify the suspect.

Using the information above, help the security team find the flag!

Search: FROM tsebehtsiworc where '1' = '1 Check

Name	Phone	Country	UID
Allegra X. Cameron	A	IVA	quis massa. Mauris vestibulum, neque sed dictum eleifend, nunc risus varius orci, in consequat enim diam vel
Kiona T. Parker	A	Piura	odio, auctor <b>larges5/24aces</b> , imperdiet nec, leo. Morbi neque tellus, imperdiet non, vestibulum nec, euismod in,

The UID columns represent all the different tables in the database

After trying all the tables, the correct table is **tsebehtsiworc** 

The SQL injection which retrieves the flag is ' UNION SELECT \* FROM tsebehtsiworc where '1' = '1

Flag:

flag:d89c5f24ace0

Theme: Affine Cipher

Tools Used:

• https://www.dcode.fr/affine-cipher

This challenge involves steganography and encryption.

# Y = 25X + 5

Upon downloading the image, we use the strings command to test whether the image contains a hidden element.

r00t:~\$	strings	linear.png	tail
т]]]^			
Jgye			
8Tom			
vA'!			
#U;`			
ΟΡΧΥ			
^#i'1			
J@2+			
IEND			
aufz:mer	nrtgzjrih	חר	
r00t:~\$			

It can be seen that there is a string of the same format as a flag - aufz:menrtgzjrihm.

A brute force approach with the Caesar cipher will fail to retrieve the key. However, the name of the file hints at using a linear cipher.

An Affine cipher encrypts data using a linear equation of the form  $E(x) = ax + b \mod m$ . This matches the form of the equation in the image: E(x) = 25x + 5.

Since the format of the flag only contains lower case alphabetic characters, the value of m should be 26. This results in the equation  $E(x) = 25x + 5 \mod 26$ , which can be easily solved using a tool such as https://www.dcode.fr/affine-cipher.



## Flag:

flag:tbsomzgwoxyt