

# Reality - ROOTCON 2020 Easter Egg Hunt

(by @katipuzer0)

**Disclaimer:** I am new to malware analysis so my approach for this problem is mostly by trial-and-error plus some 'googling'. My target audience for this write-up are beginners so the text might be verbose. I started working on the problem on April 15 but I gave up because I cannot find the flag. I decided to document it today April 17 and luckily I was able to find the flag after repeating the steps! I used ABIWord for this write-up to save time.

In this problem, we are given a MS Excel file(**topsecret.xls**) that is possibly malicious. To the uninformed (like me), the initial thing to do is to open it in MS Excel. Better do this in a VM!

First my working folder for this problem. I used Windows 10 and Kali.

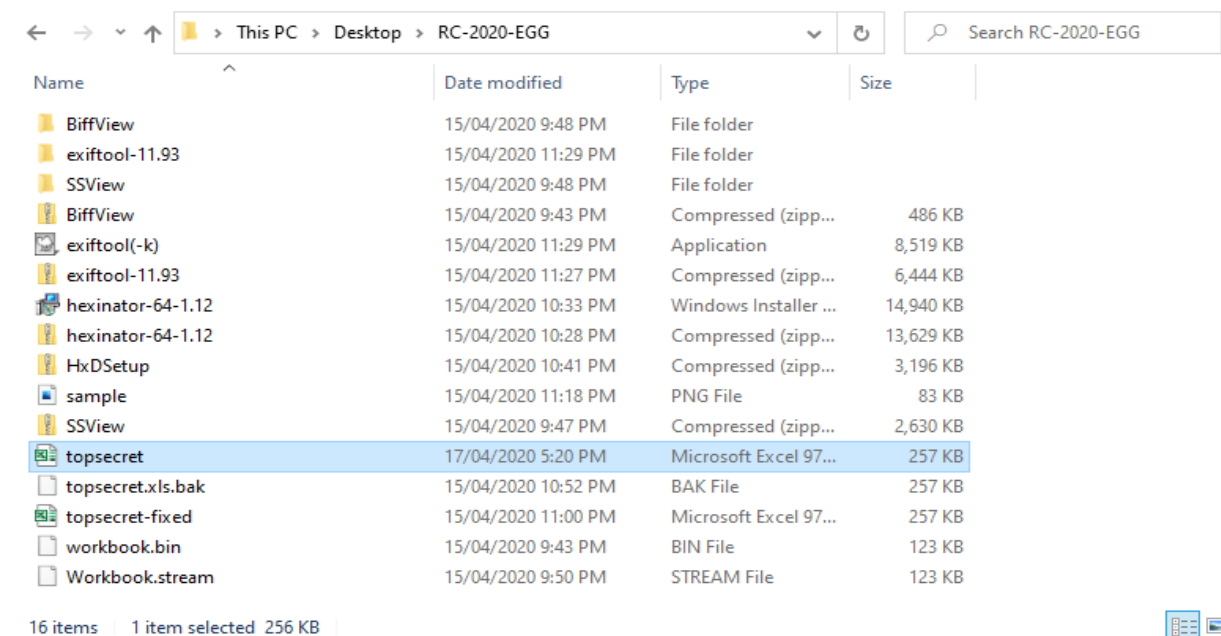


Figure 1.

This is how the spreadsheet looks like before I “enabled content”.

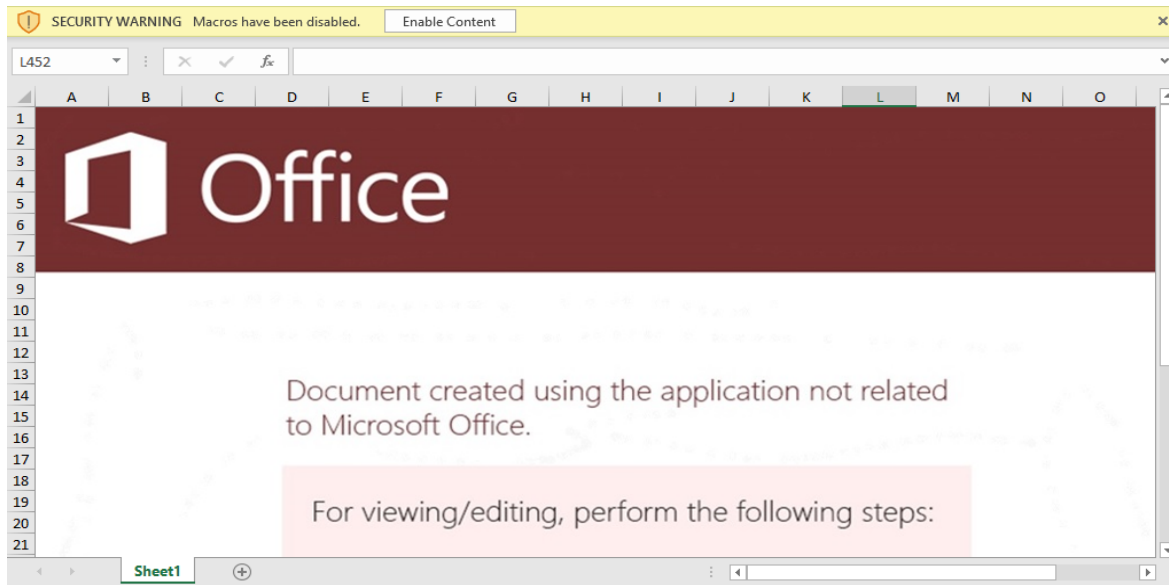


Figure 2.

After enabling the content, the calculator application appeared and a dialog box pops-up stating that the document cannot be opened. From this, we can infer that there is something fishy going on.

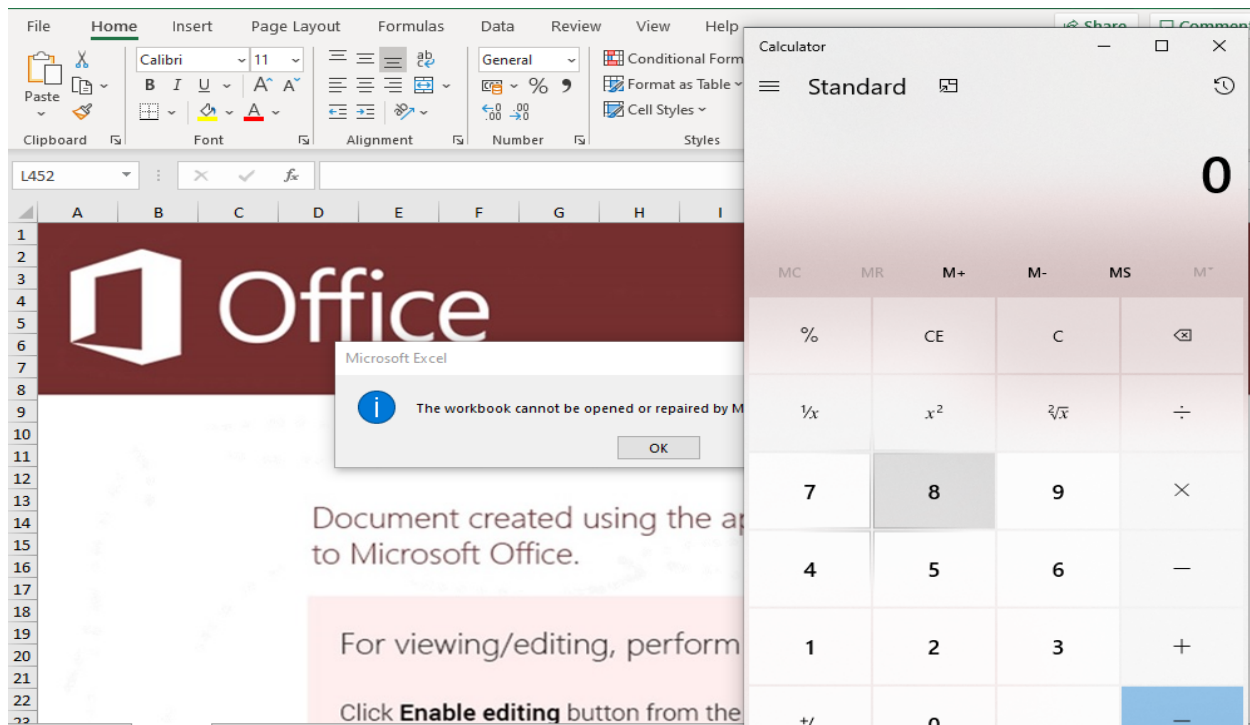


Figure 3.

Time to try some linux OLE tools [1][2]. I admit I have limited knowledge of OLE.

```
2020-04-17 05:34:11 [23 16] hagar in ~/CTFs/rc14-egg/reality
o → ole
olebrowse  olefile  olemap  oleobj  olevba
oledir     oleid    olemeta oletimes olevba3

2020-04-17 05:34:11 [23 16] hagar in ~/CTFs/rc14-egg/reality
o → ole
```

Figure 4.

```
2020-04-17 05:36:53 [23 16] hagar in ~/CTFs/rc14-egg/reality
o → oleid topsecret.xls
oleid 0.54 - http://decalage.info/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

Filename: topsecret.xls
Indicator                                Value
OLE format                              True
Has SummaryInformation stream            True
Application name                         Microsoft Excel
Encrypted                                False
Word Document                            False
VBA Macros                               False
Excel Workbook                           True
PowerPoint Presentation                  False
Visio Drawing                            False
ObjectPool                               False
Flash objects                             0
```

Figure 5. **oleid**

```
2020-04-17 05:38:08 hagar in ~/CTFs/rc14-egg/reality
o → oledir topsecret.xls
oledir 0.54 - http://decalage.info/python/oletools
OLE directory entries in file topsecret.xls:
```

id	Status	Type	Name	Left	Right	Child	1st Sect	Size
0	<Used>	Root	Root Entry	-	-	5	FFFFFFFE	0
1	unused	Empty		0	0	0	0	0
2	unused	Empty		0	0	0	0	0
3	<Used>	Stream	\x05DocumentSummaryInf ormation	-	-	-	FD	4096
4	<Used>	Stream	Workbook	-	-	-	10A	125341
5	<Used>	Stream	\x05SummaryInformation	4	3	-	F5	4096
6	unused	Empty		0	0	0	0	0
7	unused	Empty		0	0	0	0	0

id	Name	Size	CLSID
0	Root Entry	-	00020820-0000-0000-C000-0000000000046 Microsoft Microsoft Excel 97-2003 Worksheet (Excel.Sheet.8)
3	\x05DocumentSummaryInformati	4096	

Figure 6. oledir

```

2020-04-17 05:41:59 [23] hagar in ~/CTFs/rc14-egg/reality
o → olemeta topsecret.xls
olemeta 0.54 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues
=====
FILE: topsecret.xls

Properties from the SummaryInformation stream:
+-----+-----+
|Property|Value|
+-----+-----+
|codepage|1252|
|author|Windows User|
|last_saved_by|admin|
|create_time|2020-04-10 04:47:00|
|last_saved_time|2020-04-11 08:30:26|
|creating_application|Microsoft Excel|
|security|0|

```

Figure 7. olemeta

```

2020-04-17 05:43:23 [23] hagar in ~/CTFs/rc14-egg/reality
o → oleobj topsecret.xls
oleobj 0.55 - http://decalage.info/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues
-----
File: 'topsecret.xls'

```

Figure 8. oleobj

```
2020-04-17 05:44:50 hagar in ~/CTFs/rc14-egg/reality
o → oletimes topsecret.xls
oletimes 0.54 - http://decalage.info/python/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues
=====
FILE: topsecret.xls
```

Stream/Storage name	Modification Time	Creation Time
Root	1601-01-01 00:05:49	None
'\x05DocumentSummaryInformation'	None	None
'\x05SummaryInformation'	None	None
'Workbook'	None	None

Figure 9. oletimes



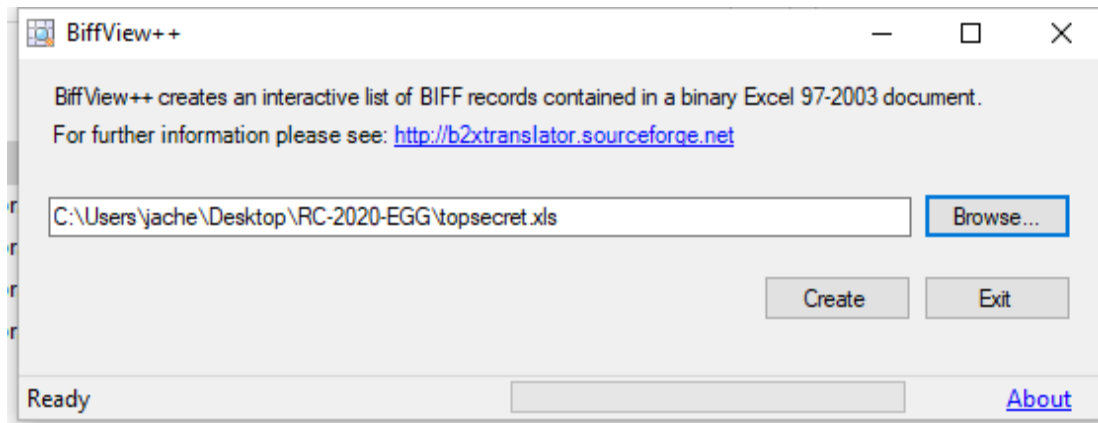


Figure 12.

BIFF	<a href="#">USESELS</a>	(160h)	2	00	00														
BIFF	<a href="#">BOUNDSHEET</a>	(85h)	14	29	5B	01	00	00	00	06	00	53	68	65	65	74	31		
BIFF	<a href="#">BOUNDSHEET</a>	(85h)	14	A3	5D	01	00	02	01	06	00	53	68	65	65	74	32		
BIFF	<a href="#">MTRSETTINGS</a>	(89Ah)	24	9A	08	00	00	00	00	00	00	00	00	00	00	01	00	00	00

Figure 13.

The one in green is the hidden sheet because its *hsState* is *0x2*, its offset is at *0x00015DA3* based on the discussions in [3]. We open the document in HxD and searched for the offset values.

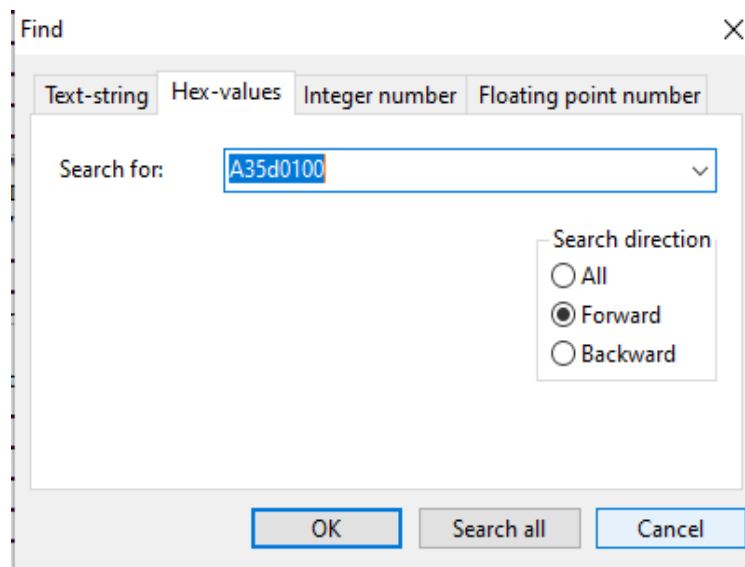


Figure 14.



topsecret.xls									
Offset(h)	00	01	02	03	04	05	06	07	Decoded text
000243B0	69	00	67	00	68	00	74	00	i.g.h.t.
000243B8	31	00	36	00	60	01	02	00	1.6.`...
000243C0	00	00	85	00	0E	00	29	5B	.....)
000243C8	01	00	00	00	06	00	53	68	.....Sheet1
000243D0	65	65	74	31	85	00	0E	00	et1.....
000243D8	A3	5D	01	00	02	01	06	00	E].....
000243E0	53	68	65	65	74	32	9A	08	Sheet2š.
000243E8	18	00	9A	08	00	00	00	00	..š.....

Figure 15.

We change the value from *0x2* (very hidden) to *0x0* (visible) then save the document.

topsecret.xls									
Offset(h)	00	01	02	03	04	05	06	07	Decoded text
000243B0	69	00	67	00	68	00	74	00	i.g.h.t.
000243B8	31	00	36	00	60	01	02	00	1.6.`...
000243C0	00	00	85	00	0E	00	29	5B	.....)
000243C8	01	00	00	00	06	00	53	68	.....Sheet1
000243D0	65	65	74	31	85	00	0E	00	et1.....
000243D8	A3	5D	01	00	00	01	06	00	E].....
000243E0	53	68	65	65	74	32	9A	08	Sheet2š.
000243E8	18	00	9A	08	00	00	00	00	..š.....

Figure 16.

Yay! We can now see Sheet2.

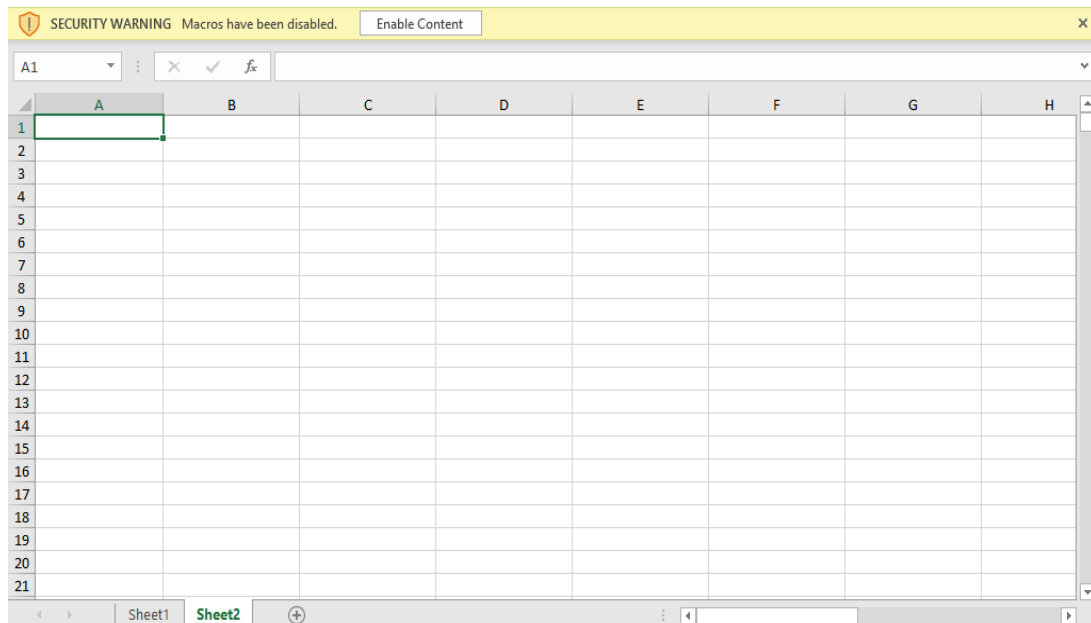


Figure 17.

Going back to the output of olevba (Figure 10), we go to cell **R50C18** and here's what we get.

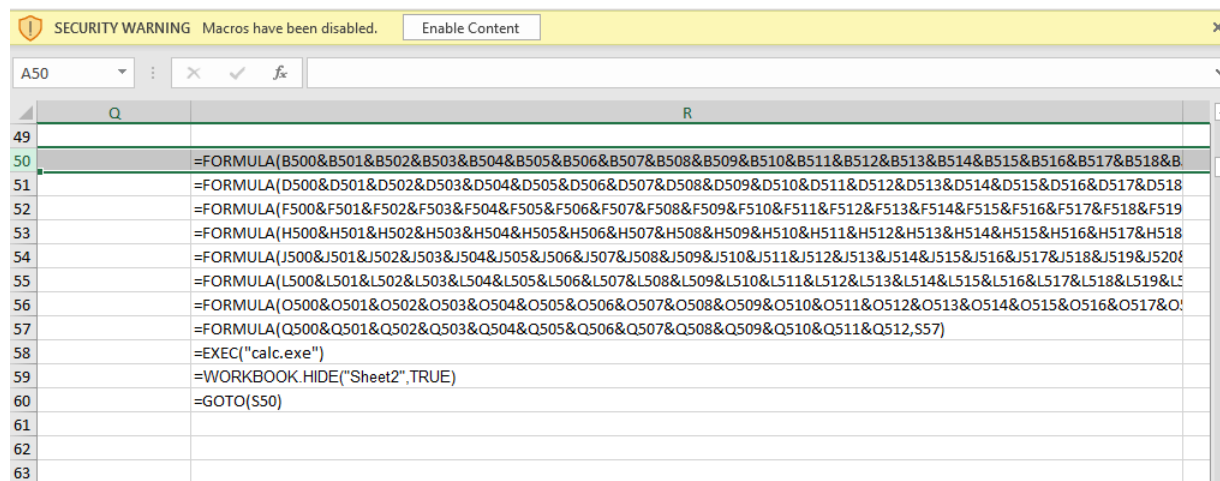


Figure 18.

If we go to B500 we get the following:

SECURITY WARNING Macros have been disabled. Enable Content							
A	B	C	D	E	F	G	H
499							
500 61	=CHAR(A500)	61	=CHAR(C500)	61	=CHAR(E500)	61	=CHAR(G500)
501 65	=CHAR(A501)	73	=CHAR(C501)	73	=CHAR(E501)	73	=CHAR(G501)
502 76	=CHAR(A502)	70	=CHAR(C502)	70	=CHAR(E502)	70	=CHAR(G502)
503 69	=CHAR(A503)	40	=CHAR(C503)	40	=CHAR(E503)	40	=CHAR(G503)
504 82	=CHAR(A504)	71	=CHAR(C504)	71	=CHAR(E504)	71	=CHAR(G504)
505 84	=CHAR(A505)	69	=CHAR(C505)	69	=CHAR(E505)	69	=CHAR(G505)
506 40	=CHAR(A506)	84	=CHAR(C506)	84	=CHAR(E506)	84	=CHAR(G506)
507 34	=CHAR(A507)	46	=CHAR(C507)	46	=CHAR(E507)	46	=CHAR(G507)
508 84	=CHAR(A508)	87	=CHAR(C508)	87	=CHAR(E508)	87	=CHAR(G508)
509 104	=CHAR(A509)	79	=CHAR(C509)	79	=CHAR(E509)	79	=CHAR(G509)
510 101	=CHAR(A510)	82	=CHAR(C510)	82	=CHAR(E510)	82	=CHAR(G510)
511 32	=CHAR(A511)	75	=CHAR(C511)	75	=CHAR(E511)	75	=CHAR(G511)
512 119	=CHAR(A512)	83	=CHAR(C512)	83	=CHAR(E512)	83	=CHAR(G512)
513 111	=CHAR(A513)	80	=CHAR(C513)	80	=CHAR(E513)	80	=CHAR(G513)
514 114	=CHAR(A514)	65	=CHAR(C514)	65	=CHAR(E514)	65	=CHAR(G514)
515 107	=CHAR(A515)	67	=CHAR(C515)	67	=CHAR(E515)	67	=CHAR(G515)
516 98	=CHAR(A516)	69	=CHAR(C516)	69	=CHAR(E516)	69	=CHAR(G516)
517 111	=CHAR(A517)	40	=CHAR(C517)	40	=CHAR(E517)	40	=CHAR(G517)
518 111	=CHAR(A518)	49	=CHAR(C518)	49	=CHAR(E518)	49	=CHAR(G518)
519 107	=CHAR(A519)	51	=CHAR(C519)	52	=CHAR(E519)	57	=CHAR(G519)

Figure 19.

In order to execute these code snippets, I created a new document and copy-pasted the cell values.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
199																						
200	61.00	=	61.00	=	61.00	=	61.00	=	61.00	=	61.00	=	61.00	61.00	=							
201	65.00	A	73.00	I	73.00	I	73.00	I	73.00	I	73.00	I	67.00	67.00	C							
202	76.00	L	70.00	F	70.00	F	70.00	F	70.00	F	70.00	F	65.00	65.00	A							
203	69.00	E	40.00	(	40.00	(	40.00	(	40.00	(	40.00	(	76.00	76.00	L							
204	82.00	R	71.00	G	71.00	G	71.00	G	71.00	G	73.00	I	76.00	76.00	L							
205	84.00	T	69.00	E	69.00	E	69.00	E	69.00	E	83.00	S	40.00	40.00	(							
206	40.00	(	84.00	T	84.00	T	84.00	T	84.00	T	78.00	N	34.00	34.00	"							
207	34.00	"	46.00	.	46.00	.	46.00	.	46.00	.	85.00	U	117.00	117.00	u							
208	84.00	T	87.00	W	87.00	W	87.00	W	87.00	W	77.00	M	114.00	114.00	r							
209	104.00	h	79.00	O	79.00	O	79.00	O	79.00	O	66.00	B	108.00	108.00	I							
210	101.00	e	82.00	R	82.00	R	82.00	R	82.00	R	69.00	E	109.00	109.00	m							
211	32.00		75.00	K	75.00	K	75.00	K	75.00	K	82.00	R	111.00	111.00	o							
212	119.00	w	83.00	S	83.00	S	83.00	S	83.00	S	40.00	(	110.00	110.00	n							
213	111.00	o	80.00	P	80.00	P	80.00	P	80.00	P	83.00	S	34.00	34.00	"							
214	114.00	r	65.00	A	65.00	A	65.00	A	65.00	A	69.00	E	44.00	44.00	,							
215	107.00	k	67.00	C	67.00	C	67.00	C	67.00	C	65.00	A	34.00	34.00	"							
216	98.00	b	69.00	E	69.00	E	69.00	E	69.00	E	82.00	R	85.00	85.00	U							
217	111.00	o	40.00	(	40.00	(	40.00	(	40.00	(	67.00	C	82.00	82.00	R							
218	111.00	o	49.00	1	49.00	1	49.00	1	52.00	4	72.00	H	76.00	76.00	L							
219	107.00	k	51.00	3	52.00	4	57.00	9	50.00	2	40.00	(	68.00	68.00	D							
220	32.00		41.00	)	41.00	)	41.00	)	41.00	)	34.00	"	111.00	111.00	o							
221	98.00	b	69.00	E	69.00	E	69.00	E	69.00	E	82.00	R	85.00	85.00	U							

Figure 20.

Examining column O, **it is a code fragment(vertically organized) that downloads something from rootcon.net** so I extracted the URL parameter and pasted it in the browser. Originally I saved it as is (text file) but it was actually a binary file.

(Note: If this is a real malware doc, my machine could have been compromised already since I opened it in my main machine without an AV product! Anyways..)

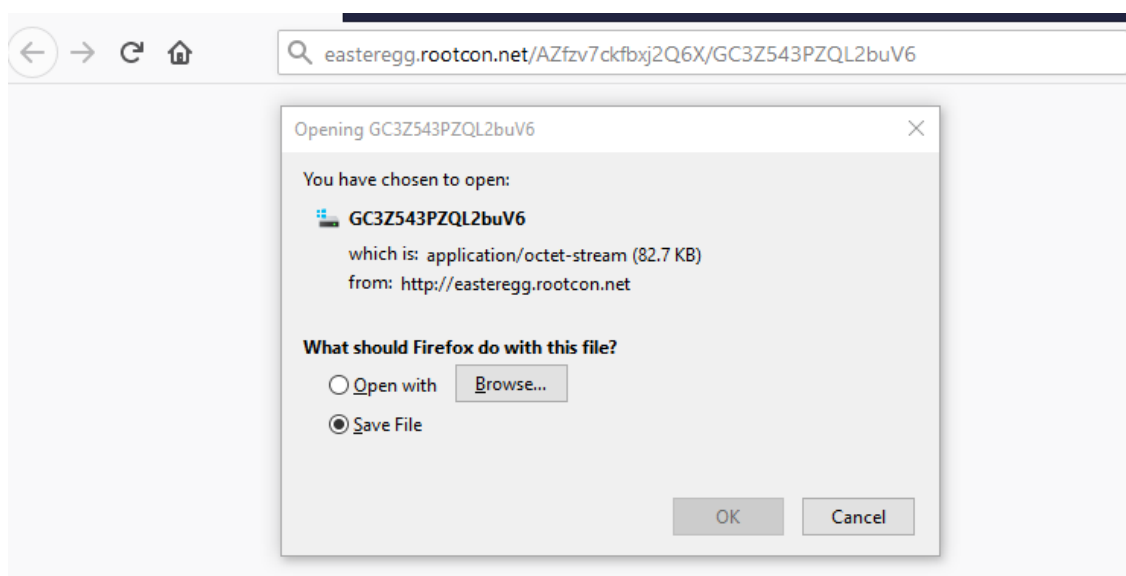


Figure 21.

I tried saving it as .png and here is what I got.

File name:	GC3Z543PZQL2buV6.png	▼
Save as type:	All Files	▼

Figure 22.



Figure 23.

I noticed that the column M is used as input to the code in column O. What about column N? It seems unused. So I copied column N to column M and a different url is accessed.

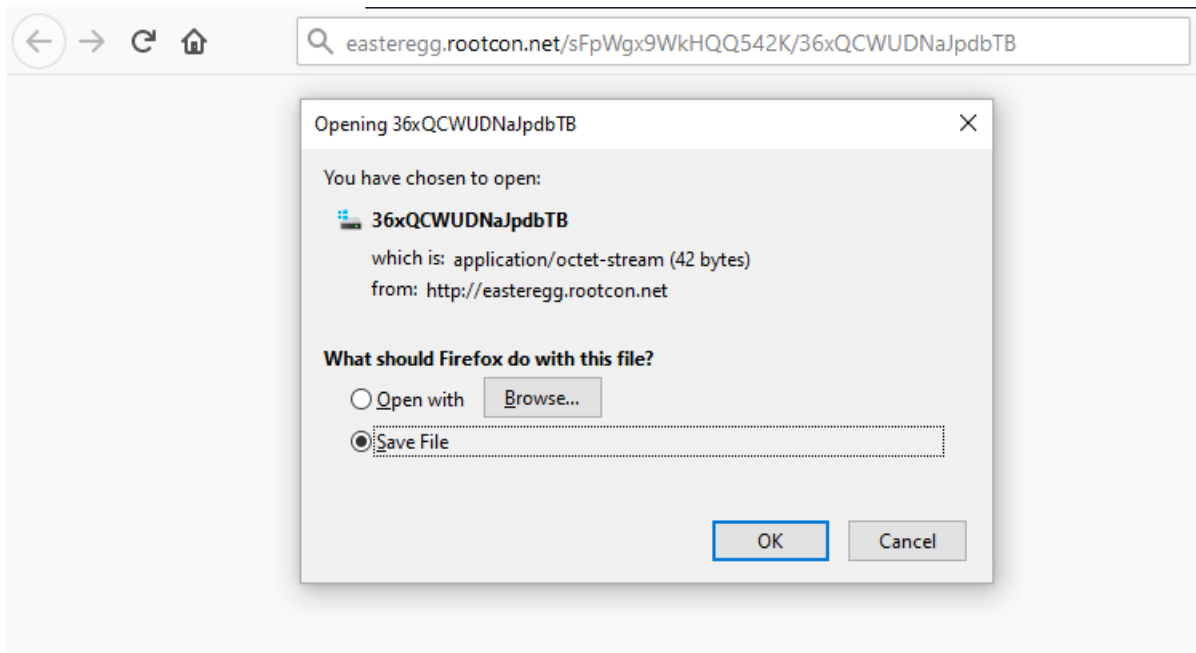


Figure 24.

And there goes the flag!

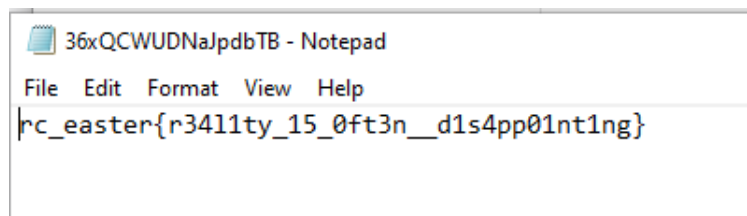


Figure 25.

References:

- [1] <https://github.com/decalage2/oletools>
- [2] <https://blog.didierstevens.com/programs/oledump-py/>
- [3] <https://inquest.net/blog/2019/01/29/Carving-Sneaky-XLM-Files>