

Agent Tesla

Malware

CONTENTS

Overview	3
Agent Tesla and What you need to know	4
What is Agent Tesla ?	4
Infection Chain	5
Initial Access	6
İş_Bankası_Döviz_Transferi_(148038560)_PDF.exe Analysis	7
Static Analysis	7
Dynamic Analysis	9
55d1ae96-5104-40e5-8b90-722f4c79c9a6.exe Analysis	10
Static Analysis	10
Dynamic Analysis	11
IOCs	17
IPs :	17
Domains :.....	17
Hashs:	17
YARA RULES	18
fbGE.exe Yara Rule	18
55d1ae96-5104-40e5-8b90-722f4c79c9a6.exe Yara Rule	19
MITRE ATT&CK TABLE	20
MITIGATIONS	21

Overview

On September 29, 2023, an email sent by İş Bankası underwent a detailed examination. The email contained a Swift message with the intention to deceive the user. This Swift message was presented as a zipped file. However, upon opening this zipped file, a file named "**İş Bankası Döviz Transferi (148038560)_PDF.exe**" was revealed. While it appeared to be a PDF file to avoid suspicion, it was, in fact, a malicious software with a .exe extension. After investigations, it was determined that this email was a phishing attack email.

At this point, the aim was to execute this malicious software without the user noticing. Upon initial access, the .exe file was examined in detail. It was found that this file was actually a malicious version of an original file named "**fbEG.exe**."

It was determined that the malicious file was embedded within another file using a technique known as packing, which is a method often used to bypass security products. The unpacking process revealed that the contained file was a Portable Executable (PE) file, indicating that it was an executable file.

In conclusion, it was determined that this malicious software belonged to the **Agent Tesla Malware Family**. To provide more information about protection from such attacks and security measures, all analysis steps and details are included in the report.

Agent Tesla and What you need to know

What is Agent Tesla ?

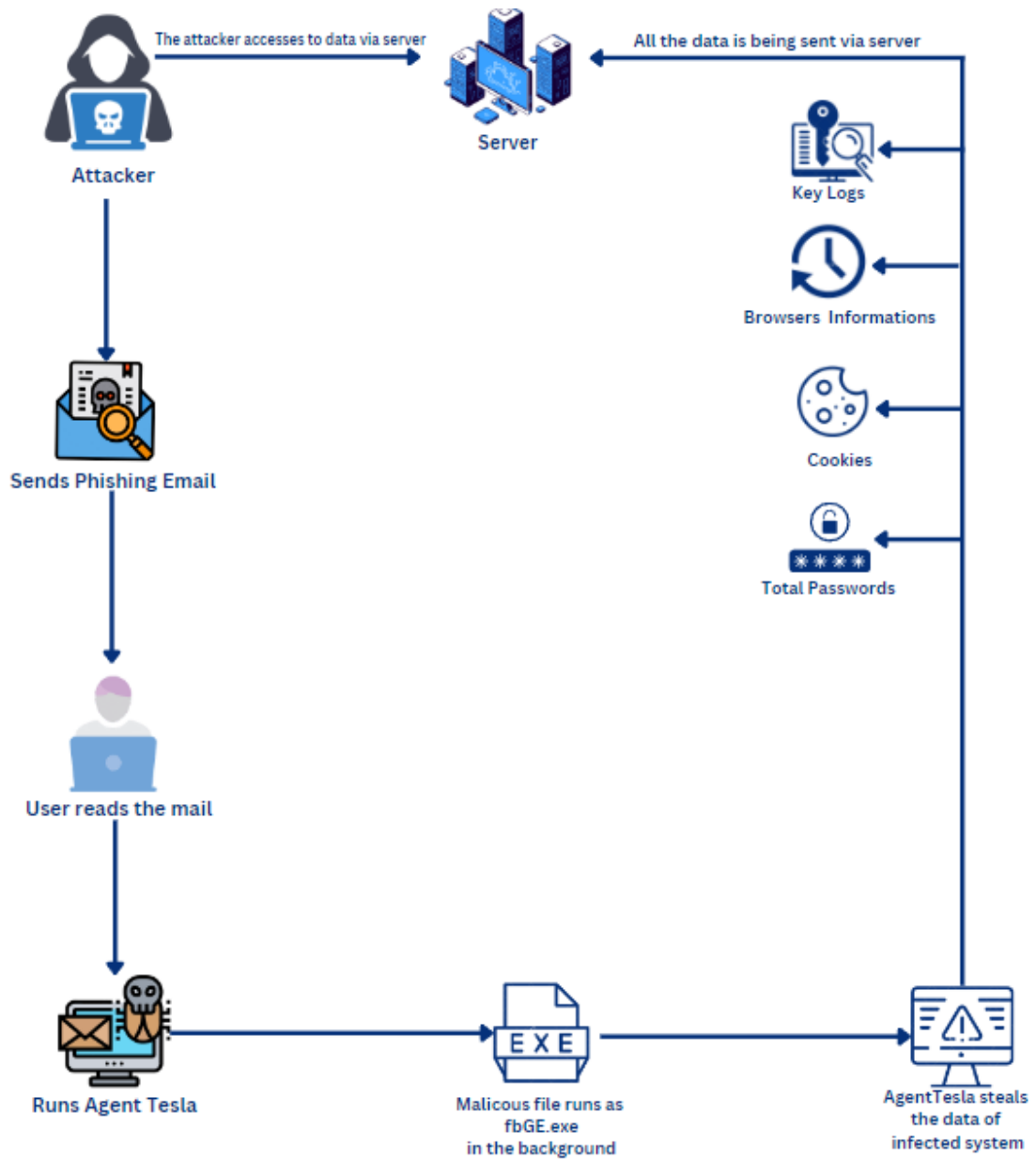
Agent Tesla is known as a .Net-based Remote Access Trojan (RAT) and is commonly used by cybercrime groups that offer Malware as a Service (MaaS). In this type of criminal business model, threat actors, known as Initial Access Brokers (IABs), provide their specialized skills to criminal groups to bypass organizations' security measures and steal information from within. Agent Tesla is used in the initial stages of such cyberattacks. Once the initial access is gained, it is employed to deliver more complex malicious software, such as ransomware, to victim systems.

Agent Tesla is a malicious software that was first detected in 2014 and has been extensively used, especially in the 2020s, in phishing campaigns with COVID-19 pandemic themes. Nowadays, this malware continues its attacks by masquerading as prominent companies. To achieve its objectives, **Agent Tesla delivers malicious attachments through email messages, often with file extensions such as .zip, .gz, .cab, .msi, and .img.** Additionally, it commonly employs malicious Visual Basic Application (VBA) macros while targeting Microsoft Office documents.

One of the distinguishing features of Agent Tesla is its ability to mimic the logos and fonts of legitimate companies in phishing campaigns. This is an effective tactic used to deceive victims and facilitate the further spread of the malware. Agent Tesla is a dangerous threat used to infiltrate computer systems and steal sensitive data.

While not as complex as other malware families in terms of locally operating second-stage capabilities, Agent Tesla can effectively steal various types of sensitive data. Furthermore, it offers an easy-to-use interface for criminals, making it an appealing choice for Initial Access Brokers to direct attacks and gather stolen information.

Infection Chain



Initial Access

From: Türkiye İş Bankası A.Ş. [mailto:bilgilendirme@ileti.isbank.com.tr]
Sent: Friday, September 29, 2023 11:52 AM
Subject: İş Bankası - Diğer Bankalara Döviz Transferi- 11:36 - 29.09.2023

Sayın Müşterimiz,

Talep ettiğiniz SWIFT mesajı ekte yer almaktadır.

Güzel günler dileriz
Türkiye İş Bankası

Finansal bildirim alma tercihinizi İşCep'ten veya İnternet Şubemizden güncelleyebilir, hassas verileri veya bankacılık sırrı niteliğindeki bilgilerinizi e-posta ile almak yerine bu bilgilere yalnızca dijital kanallarımızdan daha güvenli bir şekilde ulaşabilirsiniz. Ayrıntılı bilgi için [tıklayınız](#).

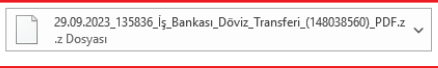


Figure 1- Phishing mail

An e-mail that appears to have been originally sent from İş Bankası is shown in figure 1.

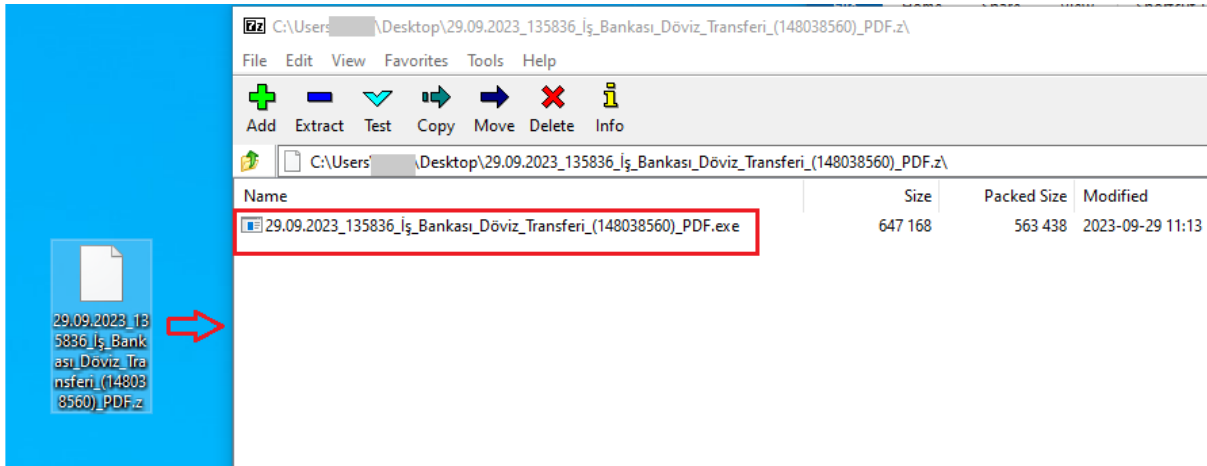


Figure 2- Email attachment content

When the email attachment is downloaded and examined, **İş_Bankası_Döviz_Transferi_(148038560)_PDF.exe** file appears.

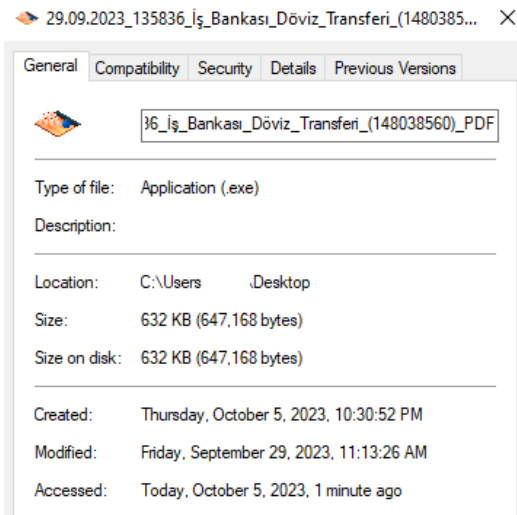


Figure 3- Detailed information of the file

İş_Bankası_Döviz_Transferi_(148038560)_PDF.exe Analysis

Static Analysis

File Name	fbGE.exe
MD5	37404d6df0a039dc790897f52ffc7538
SHA256	a5888a484f812f66cd39e16dbbcbb0891fd7f88e28a02660acfb95635055696
File Type	PE/32

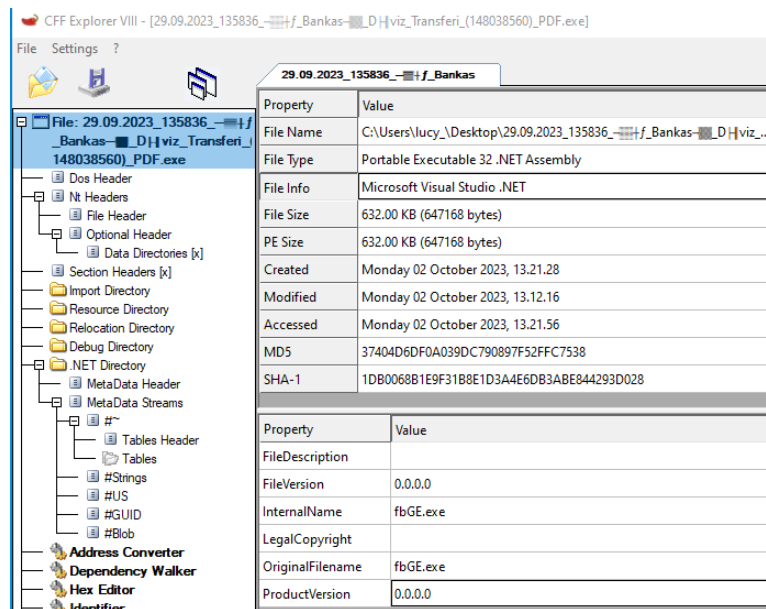


Figure 1- Information about the malicious file

It can be seen that the original file name of the **29.09.2023_135836_İş_Bankası_Döviz_Transferi_(148038560)_PDF** file is **fbGE.exe**. This file is written in **.NET**.

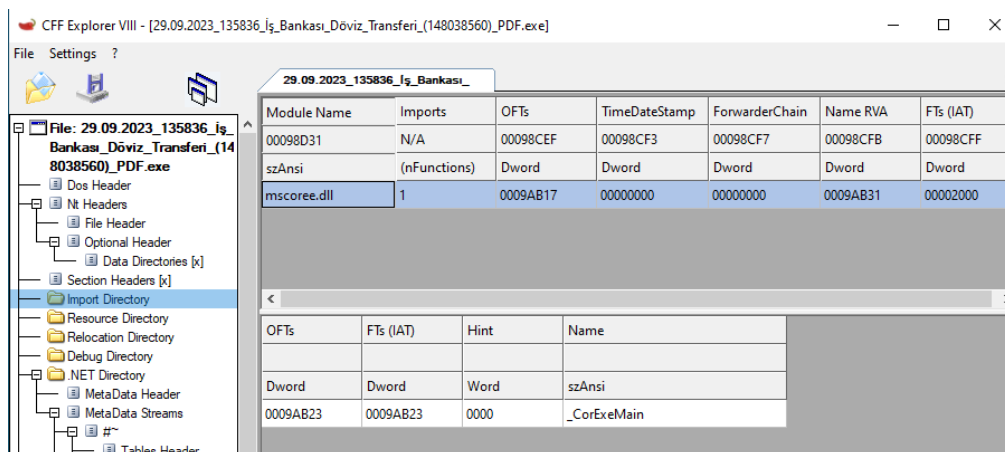


Figure 2- Imported functions

The file uses certain dlls to provide the expected functionality. Only one dll was displayed.

Agent Tesla Malware

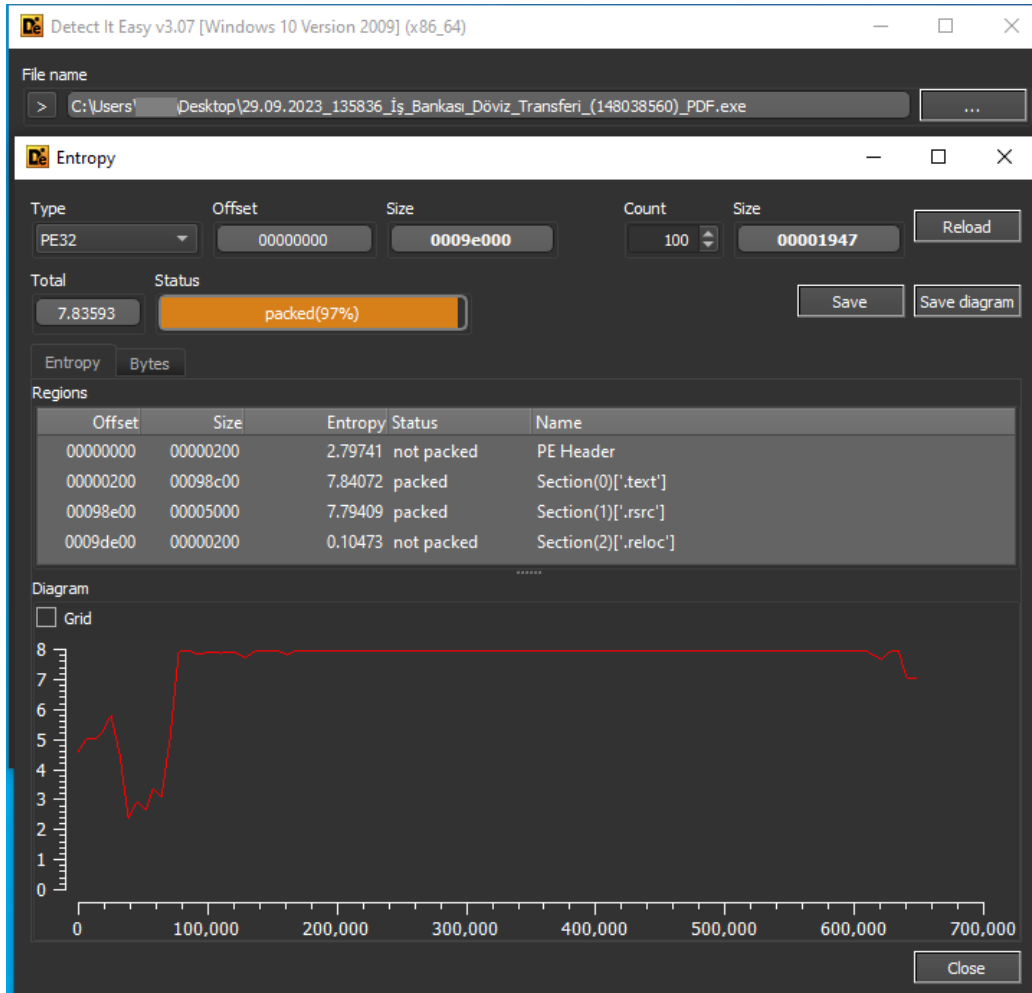


Figure 3- Pack information of the file

It seems that the pack process has been applied to the file with its original name **fbGE.exe**.

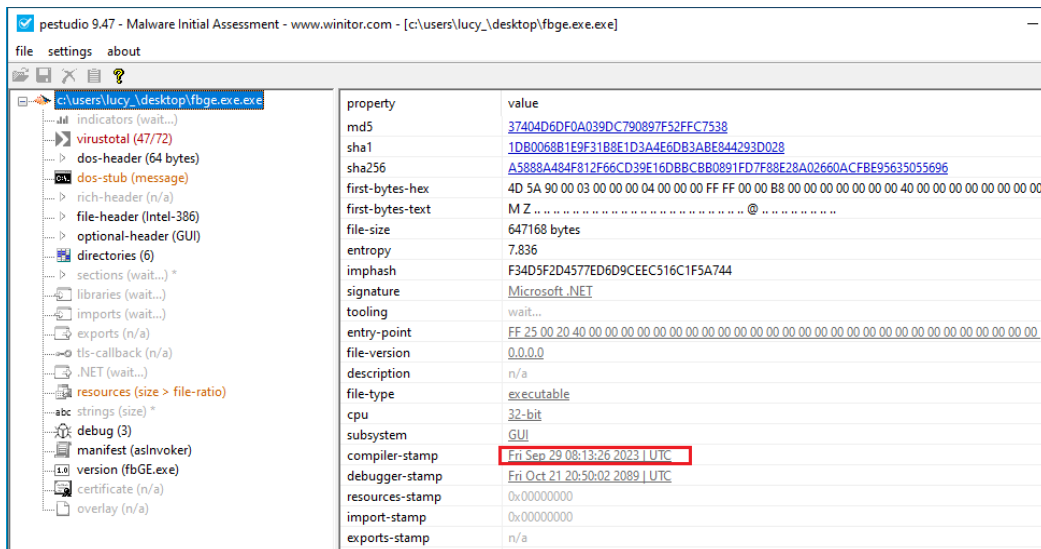


Figure 4- Detailed information about the file

It is seen that the file was compiled on September 29 at 08.13.

Dynamic Analysis

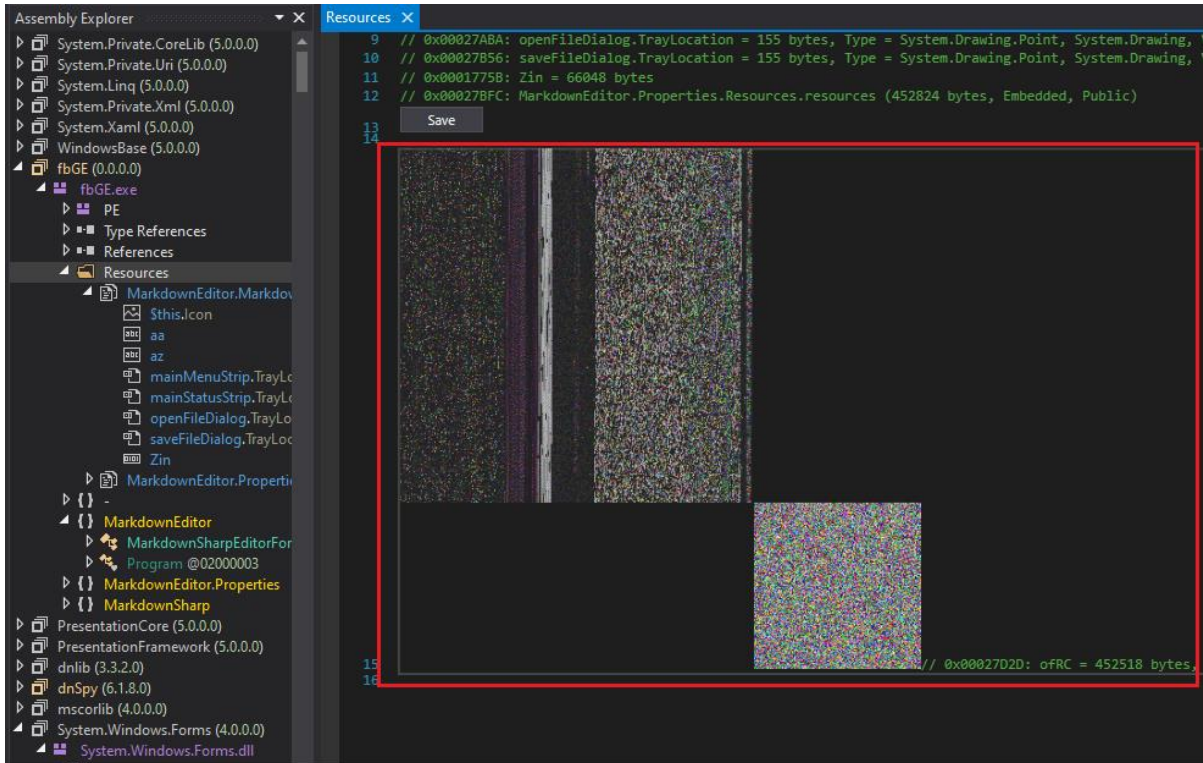


Figure 5- Embedded PE file

This malicious software has embedded a malicious component in the PE (Portable Executable) file format to evade detection and prevention by security products used to identify malicious activities. This PE file contains the main component of the malware or a portion of the malicious code, which it aims to conceal within itself.

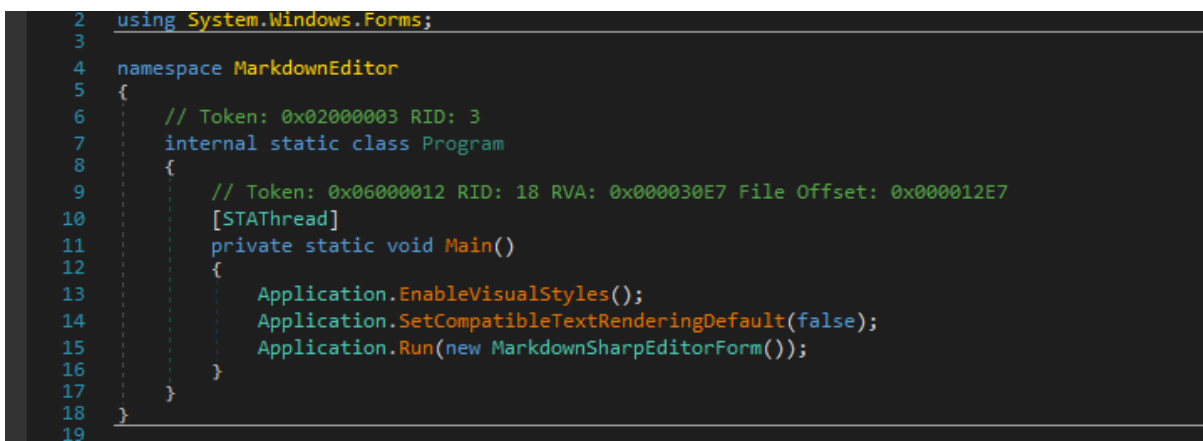


Figure 6- Main function

The packaged PE file was extracted from this file and the analysis process continued. The file that performs malicious is named **55d1ae96-5104-40e5-8b90-722f4c79c9a6.exe**.

55d1ae96-5104-40e5-8b90-722f4c79c9a6.exe Analysis

Static Analysis

File Name	55d1ae96-5104-40e5-8b90-722f4c79c9a6.exe
MD5	ec5e9334f65168cce67cd57bc6391d0a
SHA256	1105c0024a2f2173d5bbda6f209168a34ed95d5cdb05f72be075ef301ee0f63c
File Type	PE/32

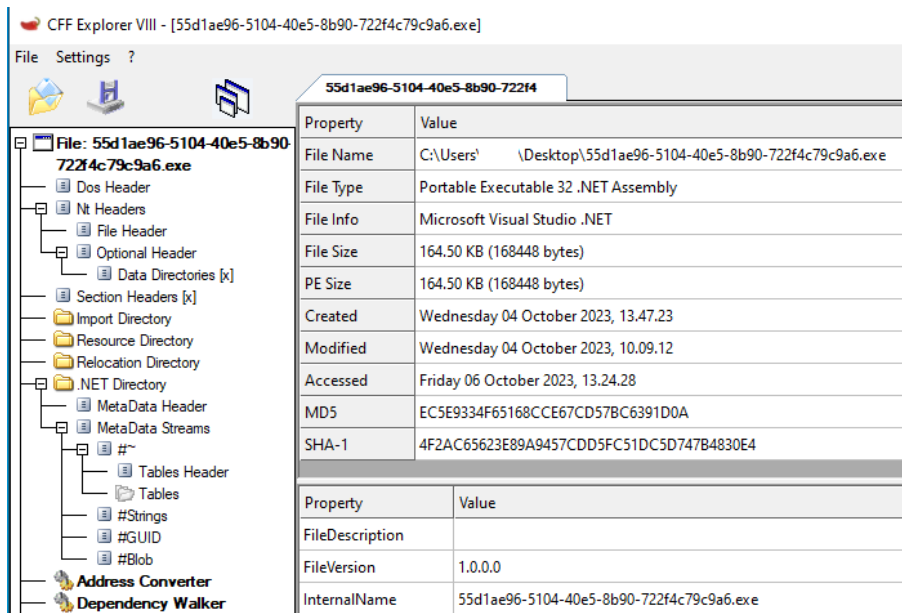


Figure 7- Information of the Extracted File

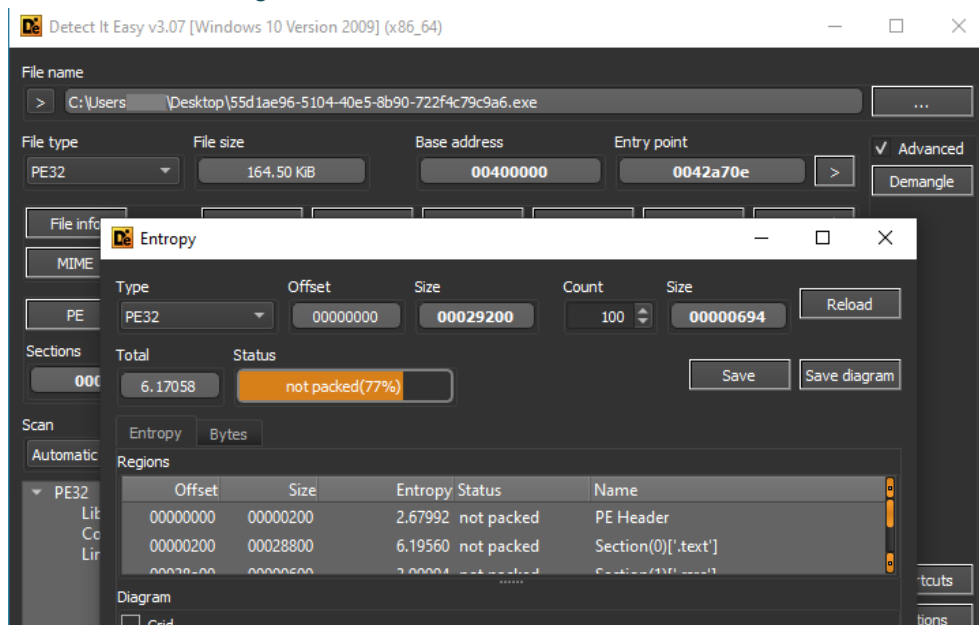


Figure 8- Information of the Extracted File

It appears to be written in **.NET** and no packaging process has been applied.

Dynamic Analysis

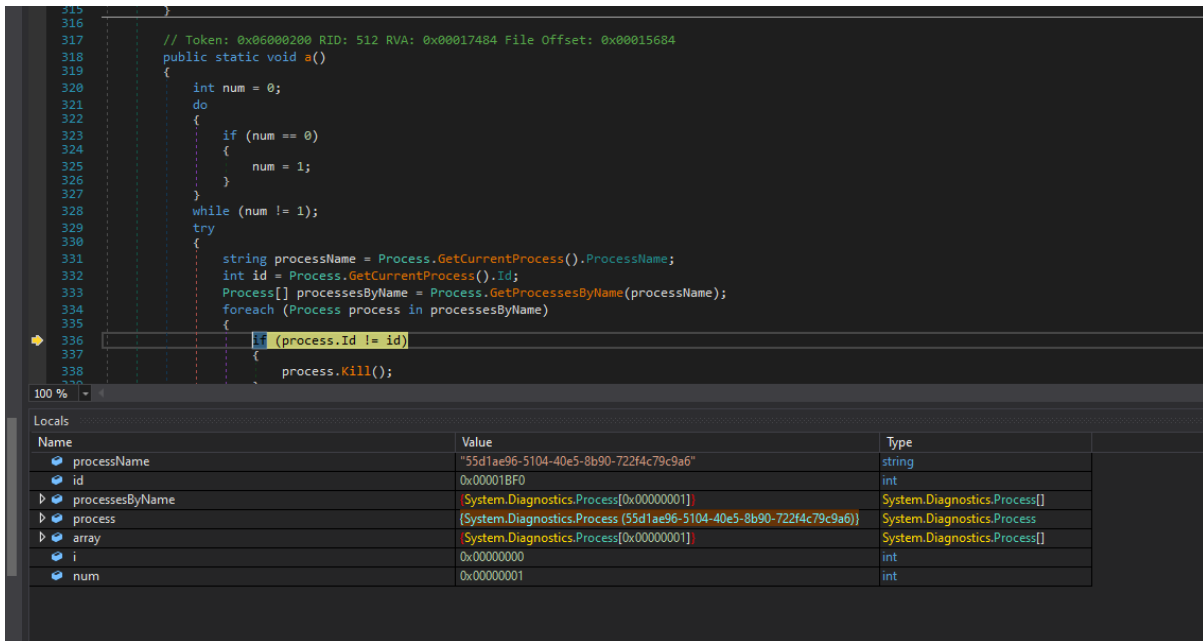


Figure 9- Process Id and Name check

First, it takes the name and id of the process and compares them. If there is a process with that process name, it terminates it.

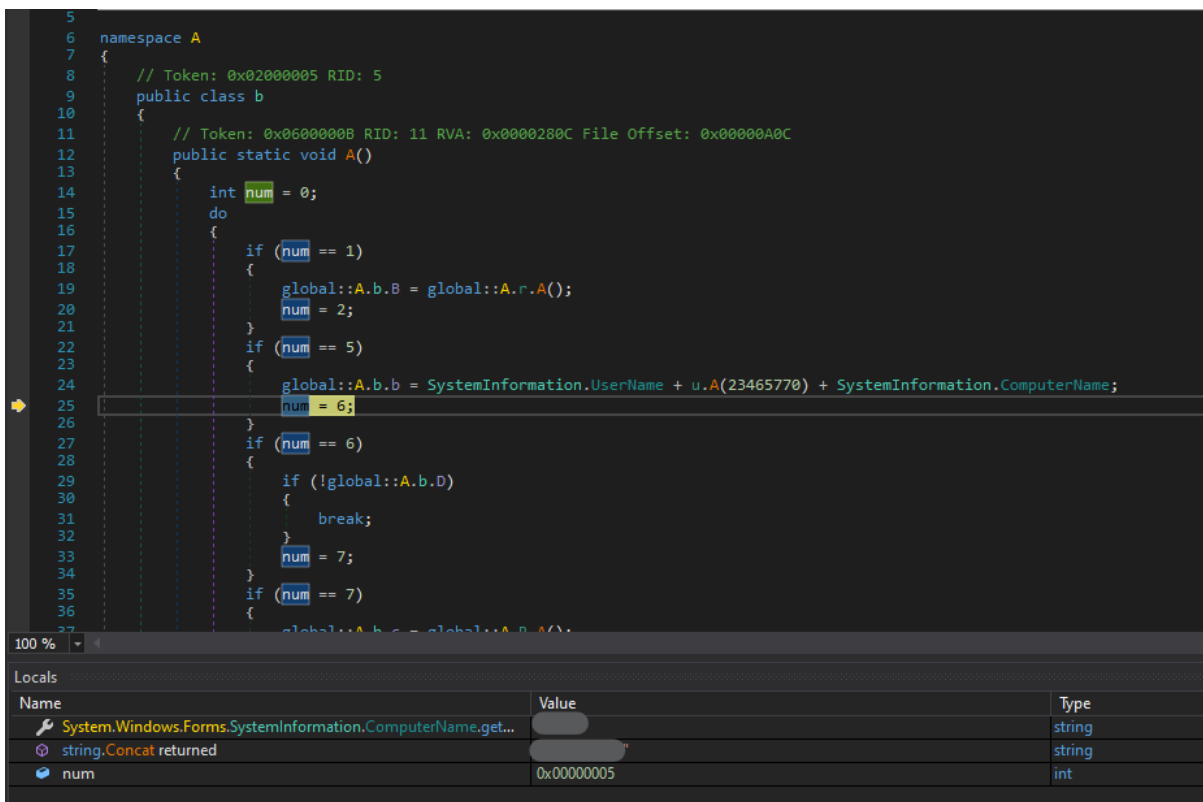


Figure 10- Username and Computer name

It takes the username and computer name.

Agent Tesla Malware

```
128     try
129     {
130         while (enumerator.MoveNext())
131         {
132             string text = enumerator.Current;
133             P.A a = P.a(text);
134             string[] directories = Directory.GetDirectories(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + u.A(23462187));
135             foreach (string str in directories)
136             {
137                 string path = str + u.A(23462166) + a.B().ToString();
138                 if (file.Exists(path))
139                 {
140                     try
141                     {
142                         byte[] array2 = P.A(u.A(-23462166), Directory.GetParent(path).FullName);
143                         Dictionary<string, string> dictionary = P.a(File.ReadAllBytes(path), array2);
144                         byte[] array3 = P.a(File.ReadAllBytes(text), dictionary, u.A(23462159), false);
145                         I.I2 = P.A(array3, A.1);
146                         if (I2 != null)
147                         {
148                             list.Add(I2);
149                         }
150                     }
151                     catch
152                     {
153                     }
154                 }
155             }
156         }
157     }
158 }
159 }
160 }
```

Name	Value
A.u.A returned	@ "\
string.Concat returned	@ "C:\Users\lucy\AppData\Roaming\Microsoft\Protect\S-1-5-21-600822206-3944175320-3390199345-1001\50c3a514-e325-40b7-b9cf-97ad24a074a0"
this	(Ax)
value	DH
list2	Count = 0x00000002
list	Count = 0x00000000
text	@ "C:\Users\lucy\AppData\Local\Microsoft\Credentials\E754AA87B36C627AB2D32C6F319486A4"

Figure 11- Receiving Data

C:\Users\Admin\AppData\Roaming\Microsoft\Protect\S-1-5-21-600822206-3944175320-3390199345-1001\50c3a514-e325-40b7-b9cf-97ad24a074a0 This directory contains user data (browser history, cookies, contains login information, etc.). The malicious file receives this data.

```
681 // Token: 0x060017D2 RID: 6098 RVA: 0x0004C674 File Offset: 0x0004A874
682 [SecuritySafeCritical]
683 public static byte[] ReadAllBytes(string path)
684 {
685     return File.InternalReadAllBytes(path, true);
686 }
687
688 // Token: 0x060017D3 RID: 6099 RVA: 0x0004C67D File Offset: 0x0004A87D
689 [SecurityCritical]
690 internal static byte[] UnsafeReadAllBytes(string path)
691 {
692     return File.InternalReadAllBytes(path, false);
693 }
694
695 // Token: 0x060017D4 RID: 6100 RVA: 0x0004C688 File Offset: 0x0004A888
696 [SecurityCritical]
697 private static byte[] InternalReadAllBytes(string path, bool checkHost)
698 {
699 }
```

Name	Value
path	@ "C:\Users\lucy\AppData\Local\Microsoft\Credentials\DFBE70A7E5CC19A398EBF1B96859CE5D"

Figure 12- Receiving Sensitive Data

Authentication information and security-related data are stored by the Windows operating system in the **C:\Users\Admin\AppData\Local\Microsoft\Credentials** directory. **(Credentials, Certificates, System Security Settings, etc.)** The malicious file aims to steal users' sensitive information by accessing this data.

Agent Tesla Malware

```

1 // Token: 0x0600007F RID: 127 RVA: 0x000052A8 File Offset: 0x000034A8
2 public List<I> A()
3 {
4     int num = 0;
5     do
6     {
7         if (num == 0)
8         {
9             num = 1;
10        }
11    } while (num != 1);
12    List<I> result;
13    try
14    {
15        string path = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), u.A(23463116));
16        string text = u.A(-23463116);
17        foreach (string text2 in Directory.GetFiles(path, u.A(23463742), SearchOption.AllDirectories))
18        {
19            if (text2.Contains(u.A(23463735)) & !text2.EndsWith(u.A(23463705)))
20            {
21                text = text2;
22                break;
23            }
24        }
25        result = g.A(text, this.a(), u.A(23463688));
26    }
27 }

```

	Value	Type
this	A.w	A.w
path	@'C:\Users\lucy\AppData\Local\UCBrowser'	string
text		string
text2	null	string
result	Count = 0x00000000	System.Collections.Generic.List<A...
files	null	string[]
	0x00000000	int
num	0x00000001	int

Figure 13- Index control

It sequentially checks the **C:\Users\Admin\AppData\Local** directory using the **GetFolderPath** API. If available, it retrieves the data in these directories. The directories it checks are as in the Table.

Table 1

C:\Users\Admin\AppData\Local\UCBrowser\
C:\Users\Admin\AppData\Local\Tencent\QQBrowser\User Data\Default\EncryptedStorage
C:\Users\Admin\AppData\Local\VirtualStore\Program Files\Foxmail\mail
C:\Users\Admin\AppData\Local\falkon\profiles
C:\Users\Admin\AppData\Local\Microsoft\Credentials\E754AA87B36C627AB2D32C6F319486A4
C:\Users\Admin\AppData\Local\NordVPN
C:\Users\Admin\AppData\Local\VirtualStore\Program Files (x86)\FTP Commander Deluxe\Ftplist.txt

Agent Tesla Malware

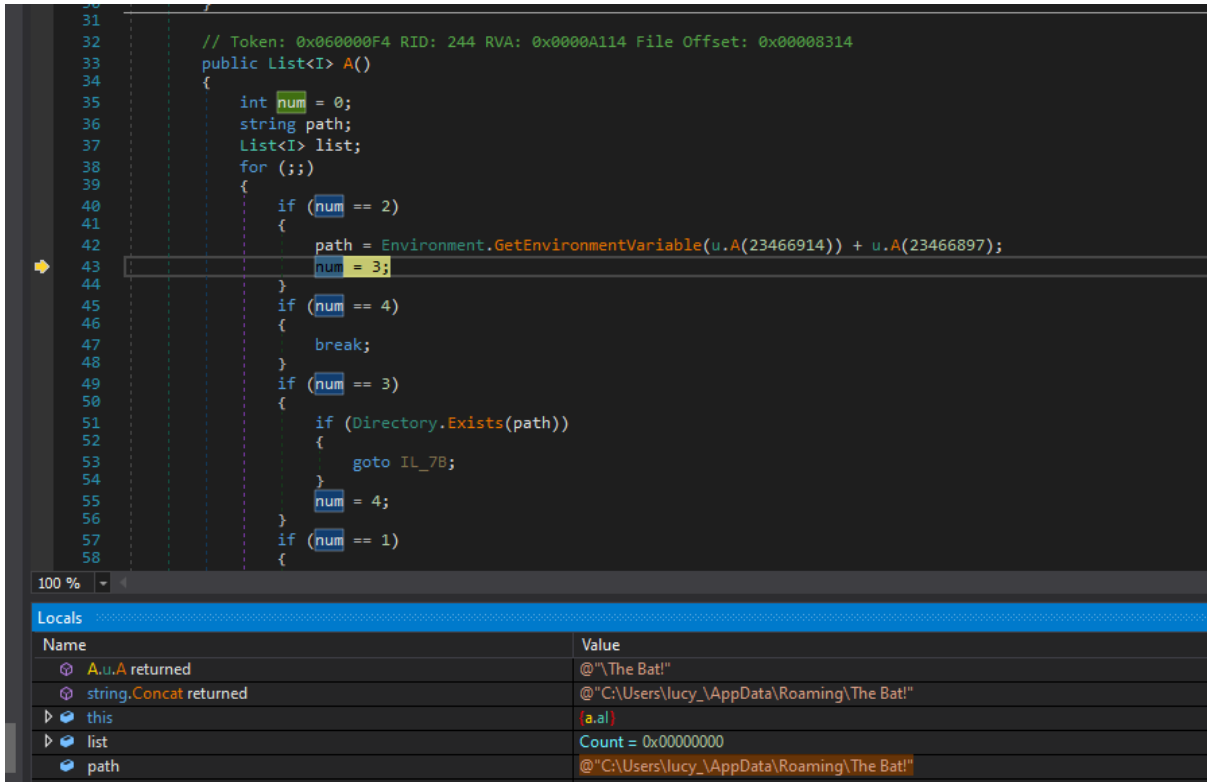


Figure 14- Directory control

Using the **GetEnvironmentVariable** function, it retrieves the variables that store the data of the applications. The variables it takes are shown in Table 2.

Table 2

C:\Users\Admin\AppData\Roaming\The Bat!
C:\Users\Admin\AppData\Roaming\Pocomail\accounts.ini
C:\Users\Admin\AppData\Roaming\eM Client\accounts.dat
C:\Users\Admin\AppData\Roaming\FileZilla\recentservers.xml
C:\Users\Admin\AppData\Roaming\Opera Mail\Opera Mail\wand.dat
C:\Users\Admin\AppData\Roaming\MySQL\Workbench\workbench_user_data.dat
C:\Users\Admin\AppData\Roaming\Claws-mail
C:\Users\Admin\AppData\Roaming\discordcanary"
C:\Users\Admin\AppData\Roaming\Discord
C:\Users\Admin\AppData\Roaming\Apple Computer\Preferences\keychain.plist
C:\Users\Admin\AppData\Roaming\Trillian\users\global\accounts.dat
C:\Users\Admin\AppData\Roaming\FTPGetter\servers.xml

Agent Tesla Malware

```

63     }
64     }
65     try
66     {
67         RegistryKey registryKey = Registry.CurrentUser.OpenSubKey(u.A(23473756));
68         if (registryKey == null)
69         {
70             return list;
71         }
72         foreach (string text in registryKey.GetSubKeyNames())
73         {
74             using (RegistryKey registryKey2 = registryKey.OpenSubKey(text))
75             {
76                 if (registryKey2 != null)
77                 {
78                     string text2 = (string)registryKey2.GetValue(u.A(23473834));
79                     string text3 = (string)registryKey2.GetValue(u.A(23473831));
80                     Version version = Environment.OSVersion.Version;
81                     int major = version.Major;
82                     int minor = version.Minor;
83                     Type typeFromHandle;
84                     if (major >= 6 && minor >= 2)
85                     {
86                         typeFromHandle = typeof(C.C);
87                     }
88                 }
89             }
90         }
91     }
92     }
93     }
94     }
95     }
96     }
97     }
98     }
99     }
100    }

```

Name	Value
this	a.aj
list	Count = 0x00000000
registryKey	{HKEY_CURRENT_USER\Software\Microsoft\ActiveSync\Partners}
text	null
registryKey2	null
text2	null
text3	null

Figure 15- Registry Operations

The malicious file targets Windows registry subkeys to carry out this process, aiming to steal or damage sensitive data from users or systems. The subkeys opened by the malicious file are displayed in Table 3.

Table 3

HKEY_CURRENT_USER\Software\Microsoft\ActiveSync\Partners
HKEY_CURRENT_USER\Software\Aerofox\Foxmail\V3.1
HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676
HKEY_CURRENT_USER\Software\Microsoft\Windows Messaging
HKEY_CURRENT_USER\Software\Microsoft\ActiveSync\Partners
HKEY_CURRENT_USER\Software\IncrediMail\Identities\
HKEY_CURRENT_USER\Software\ORL\WinVNC3
HKEY_CURRENT_USER\Software\Qualcomm\Eudora\CommandLine
HKEY_CURRENT_USER\SOFTWARE\Wow6432Node\RealVNC\WinVNC4
HKEY_CURRENT_USER\SOFTWARE\FTPWare\COREFTP\Sites
HKEY_CURRENT_USER\Software\OpenVPN-GUI\configs
HKEY_CURRENT_USER\Software\TightVNC\Server

Agent Tesla Malware

```
194      goto IL_1EA;
195      IL_1B6:
196      result = string.Intern(Encoding.UTF8.GetString(array));
197      num = 26;
198      goto IL_1CC;
199      IL_138:
200      ptr3 = (byte*)&u.bk;
201      num = 6;
202      goto IL_142;
203  }
204  return result;
205  }
206
207  // Token: 0x04000186 RID: 390 RVA: 0x0001BCE7 File Offset: 0x00019EE7
208  private static u.A bk;
209
210  // Token: 0x02000087 RID: 135
```

Name	Value
A.u/*0x02000086*/.A/*0x06000260*/ returned	"cp5ua.hyperhost.ua"

Figure 16- Server address

```
188      goto IL_97;
189      IL_1D5:
190      byte[] array5 = array;
191      int num7 = num5;
192      array5[num7] ^= array[num3];
193      num = 16;
194      goto IL_1EA;
195      IL_1B6:
196      result = string.Intern(Encoding.UTF8.GetString(array));
197      num = 26;
198      goto IL_1CC;
199      IL_138:
200      ptr3 = (byte*)&u.bk;
201      num = 6;
```

Name	Value
A.u.A returned	"royallog@saonline.xyz"

Figure 17- Username info

The malicious file communicates with the server **cp5ua[.]hyperhost[.]ua** to send the stolen data. It specifies the username for the account that will be used to send emails, which is **"royallog@saonline.xyz"** and it provides the password used for this account, which is **"7213575aceACE@#"**.

644	18.565058	204.79.197.200	10.127.0.113	TCP	54 443 → 54983 [ACK] Seq=7262 Ack=360 Win=4195072 Len=0
649	18.565872	204.79.197.200	10.127.0.113	TCP	54 443 → 54983 [ACK] Seq=7262 Ack=447 Win=4195072 Len=0
652	18.566515	204.79.197.200	10.127.0.113	TLSv1.2	396 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
654	18.566561	204.79.197.200	10.127.0.113	TLSv1.2	123 Application Data
655	18.566654	10.127.0.113	204.79.197.200	TCP	60 54983 → 443 [ACK] Seq=447 Ack=7604 Win=261632 Len=0
658	18.566823	10.127.0.113	204.79.197.200	TCP	60 54983 → 443 [ACK] Seq=447 Ack=7673 Win=261632 Len=0
665	18.567933	10.127.0.113	204.79.197.200	TLSv1.2	92 Application Data
721	18.583475	204.79.197.200	10.127.0.113	TLSv1.2	92 Application Data
722	18.583617	10.127.0.113	204.79.197.200	TCP	60 54983 → 443 [ACK] Seq=485 Ack=7711 Win=261632 Len=0
723	18.584691	204.79.197.200	10.127.0.113	TCP	54 443 → 54983 [ACK] Seq=7711 Ack=485 Win=4195072 Len=0
4235	19.381231	10.127.0.113	204.79.197.200	TCP	60 54983 → 443 [RST, ACK] Seq=485 Ack=7711 Win=0 Len=0

Figure 18- IP informations

The IP address reached by **cp5ua[.]hyperhost[.]ua** in domain name resolution is **204[.]79.197.200**.

IOCs

IPs :

IOC Type	IOC
IPv4	192[.]229.211.108
IPv4	204[.]79.197.200
IPv4	26[.]35.223.20
IPv4	91[.]235.128.141

Domains :

IOC Type	IOC
Domain	https[:]//tse1.mm.bing.net
Domain	cp5ua[.]hyperhost[.]ua

Hashs:

IOC Type	IOC
SHA256	5ab11a933c95891b62f1ba94f38cdf01ded8f19061946601670c430d084dd007
SHA256	1105c0024a2f2173d5bbda6f209168a34ed95d5cdb05f72be075ef301ee0f63c
MD5	ec5e9334f65168cce67cd57bc6391d0a
SHA256	bc6e1487bee00a8fd2b639ee4e60867d7e409bd3cb6be1451f5ddbce26340766

YARA RULES

fbGE.exe Yara Rule

```
import "hash"
rule AgentTesla
{
meta:
    author = "Kerime Gencay"
    description = "AgentTesla Rule"
    file_name = "fbGE.exe"
    hash = "37404d6df0a039dc790897f52ffc7538"
strings:

    $str1 = "CalculateNextDataValue"
    $str2 = "HyperlinkEvaluator"
    $str3 = "fbGE.exe"
    $str4 = "ImageReferenceEvaluator"
    $str5 = "MarkdownEditor"
    $str6 = "AnchorInlineEvaluator"
    $str7 = "MatchEvaluator"
    $str8 = "CodeSpanEvaluator"
    $str9 = "AtxHeaderEvaluator"
    $str10 = "ItalicsEvaluator"

condition:

    uint16(0) == 0x5A4D and (all of ($str*))
}
```

55d1ae96-5104-40e5-8b90-722f4c79c9a6.exe Yara Rule

```
import "hash"
rule AgentTesla
{
meta:
    author = "Kerime Gencay"
    description = "AgentTesla Rule"
    file_name = "55d1ae96-5104-40e5-8b90-722f4c79c9a6.exe"
    hash = "ec5e9334f65168cce67cd57bc6391d0a"
strings:

    $str1 = "HMACSHA512"
    $str2 = "SecuritySafeCriticalAttribute"
    $str3 = "set_UseShellExecute"
    $str4 = "Marshal"
    $str5 = "get_InvariantCulture"
    $str6 = "55d1ae96-5104-40e5-8b90-722f4c79c9a6.exe"

    $opc1 = {07 6F AA 00 00 0A 13 28 16 13 29 38 A4 03 00 00 11}
    $opc2 = {0A 0C 08 13 04 16 13 05 2B 1B 11 04 11 05 9A 0D 09}
    $opc3 = {16 0C 2B 28 00 08 17 FE 01 2C 0E 02 28 6B 01 00 06 6F 5F 01
00 0A 0B 18 0C 00 08 16 FE 01 2C 03}
    $opc4 = {00 11 0B 19 FE 01 2C 0B 1F 1A 28 84 00 00 0A 0C 1A 13 0B 00
11 0B 1C FE 01 2C 08 38 72}

condition:

    uint16(0) == 0x5A4D and (any of ($str*, $opc*))
}
```

MITRE ATT&CK TABLE

Discovery	Command and Control	Collections	Defence Evasion	Credential Access	Reconnaissance
T1012 Query Registry	T1102 Web Service	T1005 Data From Local System	T1406.002 Software Packing	T1552 Unsecured Credentials	T1566 Phishing
T1082 Information Discovery		T1564.001 Hidden Files and Directories		T1552.001 Credentials In Files	

MITIGATIONS

- Configure firewalls on your network to block incoming and outgoing connections from suspicious IP addresses. This can prevent RATs from establishing communication with command and control servers.
- Keep your operating system, applications, and security software up-to-date. Updates often include patches that fix vulnerabilities exploited by RATs.
- Install antivirus and anti-malware software. Perform regular scans to detect and remove any RAT infections.
- If not needed, disable remote desktop services. If needed, ensure strong passwords and proper authentication methods are in place.
- Unplug or disable devices such as webcams, microphones, or USB drives when not in use. RATs can abuse these devices for surveillance.
- Whenever possible, enable 2FA for all accounts, including email and cloud services. This can thwart unauthorized access.
- Monitor your system's running processes for any unusual or unfamiliar ones. Use task managers or specialized tools to detect suspicious activity.
- Ensure strong and unique passwords for all accounts. Avoid using easily guessable information.
- Be cautious of unsolicited emails, attachments, or links. RATs can often be delivered through phishing emails.
- Allow only approved applications to run on your system. This can prevent RATs from executing even if they manage to infiltrate.
- Regularly review and update your firewall rules to ensure they're effective against RATs and other malicious traffic.
- Keep an eye on system performance and behavior. Unexpected slowdowns, crashes, or unusual network activity could indicate a RAT

Agent Tesla

Malware
