Phemedrone Stealer Analysis | SpyCloud Labs

spycloud.com/blog/phemedrone-stealer/

James

September 6, 2024

At SpyCloud, we recapture logs from more than sixty infostealer malware families, but very few of them are open source stealers. Intrigued, our team at SpyCloud Labs took on the task of dissecting Phemedrone, an open source stealer available to anyone on Telegram.

When we dug in, we found Phemedrone to have some other unique characteristics as well, namely:

- Its ability to encrypt logs with RSA + AES

- A high-level of configurability

- Apparent targeting of Russian users/services

Here’s what we found.

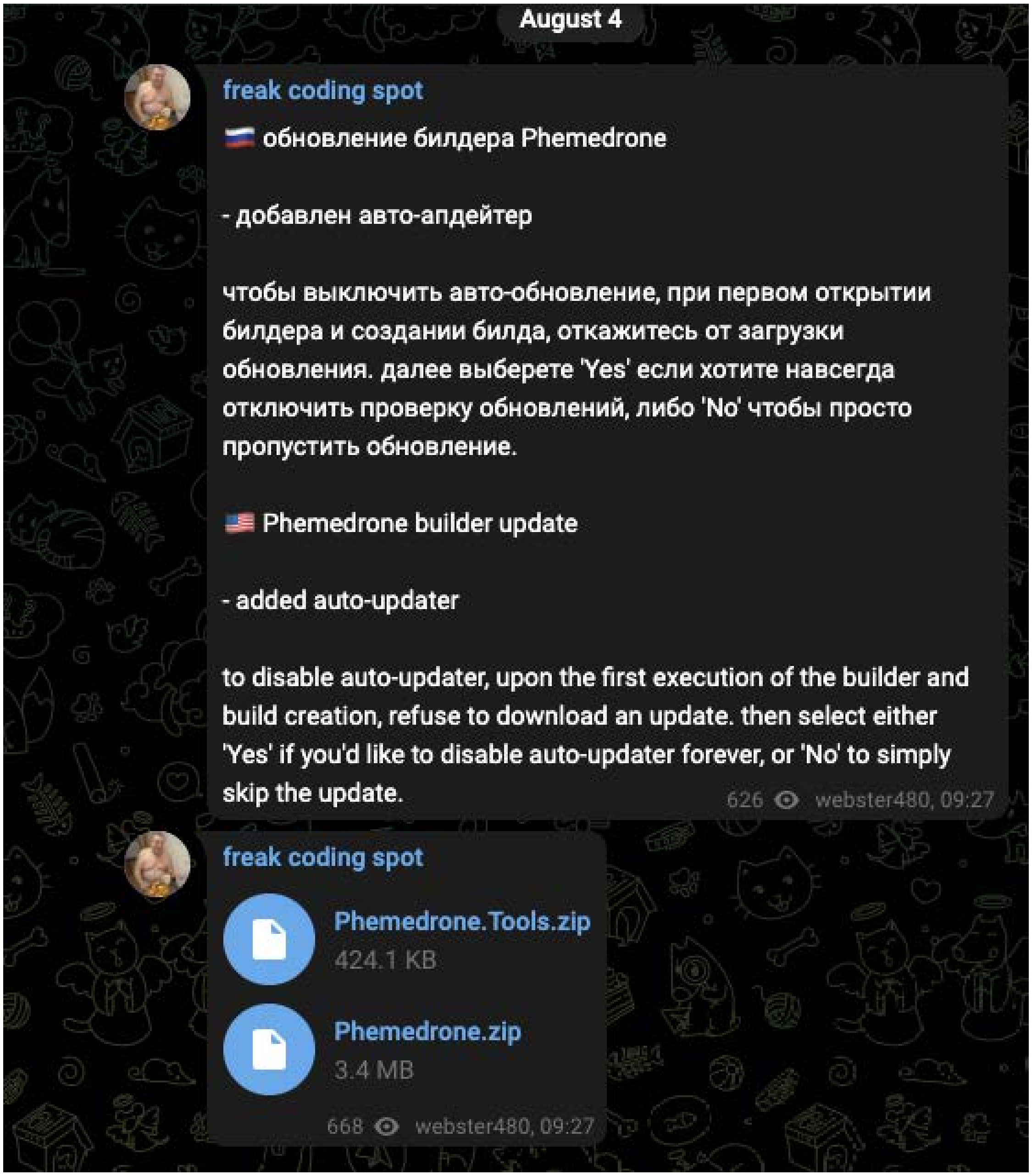
About Phemedrone Stealer

Phemedrone, which as we mentioned is an entirely open source stealer, is written in C# and therefore provides abundant opportunities for actors to customize the malware to suit their needs. It also gives bad actors an easy snapshot of what they have stolen within its logs, leveraging password/cookie “tagging” for various categories. However, when looking at the definitions for these tags, it becomes clear that many of these tags focus on Russian targets, which is pretty unique for a stealer.

Phemedrone’s offering

With code distributed mainly over Telegram (and previously on GitHub before being taken down), bad actors can acquire and deploy Phemedrone for free. Phemedrone offers log encryption when sending to Telegram, browser/application theft, cookie tagging, and more, as well as the ability to easily tweak the stealer in C#.

Phemedrone's devs release regular updates for both their panel as well as their builder, which keeps Phemedrone active and well-used. They also offer a chat for people to discuss Phemedrone.



August 4

freak coding spot

🇷🇺 обновление билдера Phemedrone

- добавлен авто-апдейтер

чтобы выключить авто-обновление, при первом открытии билдера и создании билда, откажитесь от загрузки обновления. далее выберете 'Yes' если хотите навсегда отключить проверку обновлений, либо 'No' чтобы просто пропустить обновление.

🇺🇸 Phemedrone builder update

- added auto-updater

to disable auto-updater, upon the first execution of the builder and build creation, refuse to download an update. then select either 'Yes' if you'd like to disable auto-updater forever, or 'No' to simply skip the update.

626 👁 webster480, 09:27

freak coding spot

Phemedrone.Tools.zip
424.1 KB

Phemedrone.zip
3.4 MB

668 👁 webster480, 09:27

Image 1: Phemedrone's chat offering, in both English and Russian.

Binary analysis

Phemedrone’s operation is fairly simple, opting to do password/cookie parsing on the victim’s machine instead of just stealing entire raw password database files to be parsed on a panel later. This allows Phemedrone to then tag stolen passwords and cookies with a variety of categories to make it easy to identify which logs are useful. By default, many of these tag values are associated with primarily Russian targets, such as tinkoff and sberbank for “BANK”.

Stealer capabilities

Browser theft

Phemedrone accesses a variety of Chromium and Firefox/Gecko based browsers in order to steal data from them. Phemedrone steals data from the internal Chromium/Firefox storage databases that store passwords, credit cards, cookies, and more. Additionally, when stealing from Chromium based browsers, Phemedrone also targets the following extensions:

Extension name	Extension GUID
Authenticator	bhghoamapcdpbohphigoooaddinpkbai
EOS Authenticator	oeljdldpnmdbchonielidgobddffflal
BrowserPass	naepdomgkenhinolocfifgehidddafch
MYKI	bmikpgodpkclnkgmnpphehdgcimmided
Splikity	jhfjfclepacoldmjmkmdlmganfaalklb
CommonKey	chgfefjpcobfbnpmiokfjjaglahmnded
Zoho Vault	igkpcodhieompeloncfnbekccinhapdb
Norton Password Manager	admmjipmmciaobhojoghlmleefbicajg
Avira Password Manager	caljgklbbfbcjjanaijlacgncafpegll
Trezor Password Manager	imloifkgjagghnncjkhggdhalmcnfklk
MetaMask	nkbihfbeogaeaoehlefnkodbefgpgknn
TronLink	ibnejdfjmmkpcnlpebklmnkoeoihofec
BinanceChain	fhbohimaelbohpjbbldcngcnapndodjp
Coin98	aeachknmefphepccionboohckonoeemg
iWallet	kncchdigobghenbbaddojjnnaogfppfj
Wombat	amkmjjmmflddogmhpjloimipbofnfjih

NeoLine	cphhlgmgameodnhkjdmkpanlelnlohao
Terra Station	aiifbnbfobpmeekipheeijimdpnlpgpp
Keplr	dmkamcknogkgcdfhhbddcghachkejeap
Sollet	fhmfendgdocmcbmfikdcogofphimnkno
ICONex	flpiciilemghbmfalicajoolhkkenfel
KHC	hcflpincpppdclinealmandijcmnkbgn
TezBox	mnfifefkajgofkcjkemidiaecocnkjeh
Byone	nlgbhdfgdhgbiamfdfmbikcdghidoadd
OneKey	ilbbpajmiplgpehdikmejfemfklpkmke
Trust Wallets	pknlccmneadmjbkollckpblgaaabameg
MetaWallet	pfknkoocfefiocadajpngdknmkjgakdg
Guarda Wallet	fcglfhcjfpkgdppjbglknafgfffkelnm
Exodus	idkppnahnmmggbmfkjhiakkbkdpnmnon
JaxxxLiberty	mhonjhhcgphdphdjcdoeodfdliikapmj
Atomic Wallet	bhmlbgebokamljgnceonbncdofmmkedg
Electrum	hieplnfojfccegoloniefimmbfjdgcgp
Mycelium	pidhddgciaponoajdngciiemcflpnnbg
Coinomi	blbpgcogcoohhngdjafgpoagcilicpjh
GreenAddress	gflpckpfdgcagnbdfafmibcmkadnlhpj
Edge	doljkehcfhidippihgakcihcmnknlphh
BRD	nbokbjkelpmlgflobbohapifnnenbjlh
Samourai Wallet	apjdnokplgcjkejimjdfjnhmjlbpgkdi
Copay	ieedgmmkpkbiblijbbldefkomatsuahh
Bread	jifanbgejlbcmhbbdbnfbfnlmbomjedj
KeepKey	dojmlmceifkfgkgeejemfciibjehhdcl
Trezor	jpxupxjxheguvfyhfhahqvxvyqthiryh

Ledger Live	pfkcfdjnlfjcmkjnhcbfhfkkoflnhjln
Ledger Wallet	hbpfjlflhnmkddbjdchbbifhllgmmhnm
Bitbox	ocmfilhakdbncmojmlbagpkjfbmeinbd
Digital Bitbox	dbhklojmlkgmpihhdooibnmidfpeaing
YubiKey	mammpjaaoinfelloncbbpomjcihbkmmc
Google Authenticator	khcodhlfkpmhibicdjjblnkgimdepgnd
Microsoft Authenticator	bfbdnbpibgndpjfhonkflpkijfapmomn
Authy	gjffdbjndmcafeoehgdldobgjmlepcal
Duo Mobile	eidlicjlkaiefdbgmdepmmicpbggmhoj
OTP Auth	bobfejfdlhnabgglompioclndjejolch
FreeOTP	elokfmmmjbadpgdjmgglocapdckdcpkn
Aegis Authenticator	ppdjlkfkedmidmclhakfncpfdmdgmjpm
LastPass Authenticator	cfoajccjibkjhbdjnpkbananbejpkkjb
Dashlane	flikjlpgnpcjdienoojmgliechmmheek
Keeper	gofhklgdnbnpcdigdgkgfobhhghjmmkj
RoboForm	hppmchachflomkejbhofobganapojjol
KeePass	lbfeahdfdkibininjgejjgpdafeopflb
KeePassXC	kgeohlebpjgcfiidfhhdlnnkhefajmca
Bitwarden	inljaljiffkdgmlndjkdiepghpolcpki
NordPass	njgnlkhcjgmjfnfahdmfkalpjcneebpl
LastPass	gabedfkgnbglfbnplfpjddgfnbibkmbb
Nifty Wallet	jbdaocneiiinmjbjlgalhcelgbejmnid
Math Wallet	afbcbjpbpfadlkmhmclhkeeodmamcflc
Coinbase Wallet	hnfanknocfeofbddgcijnmhnfnkdnaad
Equal Wallet	blnieiiffboillknjnepogjhkgnoac
EVER Wallet	cgeeodpfagjceefieflmdfphplkenlfk

Jaxx Liberty	ocefimbphcgjaahbclemolcmkeanoagc
BitApp Wallet	fihkakfobkmkjojpchpfgcmhfjnmnfpi
Mew CX	nlbmnnijcnlegkjjpcfjclmcfggfefdm
GU Wallet	nfinomegcaccbhchhgflladpfbajihdf
Guild Wallet	nanjmdkhkinifnkgdeggcnhdaammmj
Saturn Wallet	nkddgncdjgifcddamgcmfnlhccnimig
Harmony Wallet	fnnegphlobjdpkhecapkijjdkgcjhkib
TON Wallet	nphplpgoakhhjchkkhmiggakijnkhfnd
OpenMask Wallet	penjlddjkjgpnkllboccdgccekpkcbin
MyTonWallet	fldfpgipfncgndfolcbkdeeknbbbnhcc
DeWallet	pnccjgokhbngghddhahcnaopgeipafg
TrustWallet	egjidjbpglichdcondbcbdnbeeppgdph
NC Wallet	imlcamfeniaidioeflifonfjeeppblda
Moso Wallet	ajkifnllfhikkjbjopkhmjoieikeihjb
Enkrypt Wallet	kkpllkodjeloidieedojogacfhpaihoh
CirusWeb3 Wallet	kgdijkcfiglijhaglibaidbipiejjfdp
Martian and Sui Wallet	efbglgofoippbgcjepnhiblaibcnclgk
SubWallet	onhogfjeacnfoofkfgppdlbmlmnplgbn
Pontem Wallet	phkbamefinggmakgklpkljjmgibohnba
Talisman Wallet	fijngjgcjhjmmpcmkeiomlglpeiijkld
Kardiachain Wallet	pdadjkfkgcafgbceimcpbkalnfnepbnk
Phantom Wallet	bfnaelmomeimhlpmgjnjophhpkkoljpa
Oxygen Wallet	fhilaheimglignddjgofkcbgekhenbh
PaliWallet	mgfffbidihjpoaomajlbgchddlicgpn
BoltX Wallet	aodkkagnadcbobfpggnjeongemjbjca
Liquality Wallet	kpopkelmapcoipemfendmdghnegimn

xDefi Wallet	hmeobnffcmdkdcmlb1gagmfpfboieaf
Nami Wallet	Ipfcbjknijpeeillifnkikgncikgfhdo
MaiarDeFi Wallet	dngmlblcodfobpdpecaadgfbeggfjfnm
MetaMask Edge Wallet	ejbalbakoplchlghecdalmeeeajnimhm
Goblin Wallet	mlbafbjadjidk1bhgopoamemfibcpdfi
Braavos Smart Wallet	jnlgamecbpmbajjfhmmmlhejkemejdma
UniSat Wallet	ppbibelpcjmhbdihakflkdcoccbgbkpo
OKX Wallet	mcohilncbfahbmgdjkbpemcciiolgcge
Manta Wallet	enabgbdfcbaehmbigakijjabdpdnimlg
Suku Wallet	fopmedgnkfpebgllppeddmmochcookhc
Suiet Wallet	khpkpbbcccdmmclmpigdgddabeilkdpd
Koala Wallet	lnnnmfcpbkafcpgdilckhmhbkkbpkmid
ExodusWeb3 Wallet	aholpfdialjgjfhomihkjbmgjidlcdno
Aurox Wallet	kilnpioakcdndlodeeceffgjdpojajlo
Fewcha Move Wallet	ebfidpplhabeedpnhjnobghokpiioolj
Carax Demon Wallet	mdjmfdffdcmnoblignmgpommbefadffd
Leap Terra Wallet	aijcbedoijmgnlmjeegjaglmepbmpkpi

Cryptowallet theft

Phemedrone also targets cryptowallets on the victim’s machine, looking for “wallet.dat” files to steal from. Additionally, Phemedrone steals from the following hardcoded cryptowallets:

Armory

- Atomic

- Bytecoin

- Coninomi

- ›

Jaxx

- Electrum

- Exodus

- ›

Guarda

- ZCash

This functionality allows Phemedrone to steal victims’ cryptocurrency with ease.

Discord token theft

Phemedrone will target Discord tokens by accessing the Discord leveldb database, stored on a victim’s computer. It will then regex for “dQw4w9WgXcQdQw4w9WgXcQ:[^\”]*”, which it will use to extract the victim’s Discord token for authentication purposes. This string is appended to each encrypted Discord token stored in the victim’s Discord leveldb database. The exact string is actually a rickroll easter egg.

FileGrabber

Phemedrone also includes a basic filegrabber, which will iterate through My Documents and Desktop and steal all files based on config supplied max file size and directory depth.

FTP theft

Phemedrone will target a popular FTP application, FileZilla, for theft. From FileZilla, Phemedrone will steal a victim's "recentservers.xml" as well as their "sitemanager.xml"

Screenshot

Phemedrone will automatically obtain a screenshot of the victim's screen post installation for exfiltration.

Steam theft

Phemedrone will target the game application Steam for theft, stealing *ssfn* and \\config.vdf files, which attackers can use to take over a victim's Steam account.

Telegram theft

Phemedrone targets Telegram for theft, too. Phemedrone grabs the DefaultIcon from a victim's registry, in addition to stealing a victim's tdata information, which can be used to take over their Telegram account.

VPN theft

Phemedrone targets several common VPN providers for theft in order to steal a victim's VPN connection info. Phemedrone targets the following applications:

- **OpenVPN**: Steals Profiles and ovpn files
- **ProtonVPN**: Steals ProtonVPN user.config
- **SurfShark**: Steals SurfShark *.dat

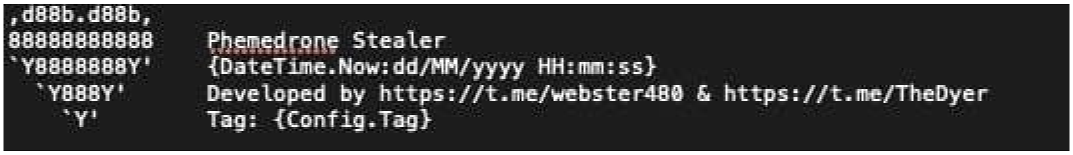
Cookie and password tagging

Phemedrone has the ability to look through stolen cookies/passwords and provide a "snapshot" of what was stolen using a list of tags contained in the binary. These tags look for domains and are as follows:

Tag Category	Tag Domain
Cheats	celka.
Cheats	nursultan.
Cheats	xone

Cheats	akrien
Cheats	interium
Cheats	nixware
Cheats	skeet
Games	roblox.com.
Games	genshin
Games	minecraft.net
Games	epicgames.com
Games	steampowered.com
Bank	tinkoff
Bank	sberbank
Money	yoomoney
Money	amazon
Money	funpay
Money	americanexpress
Crypto	binance
Crypto	bybit

These tags are added to the generated Information.txt, along with information about the victim's system, total passwords stolen, total cookies stolen, and an ASCII heart with the Phemedrone author signature. These tags are easily customizable, and in fact, in variants such as "Mephedrone", we can see tags added to the list, such as "FACEBOOK".



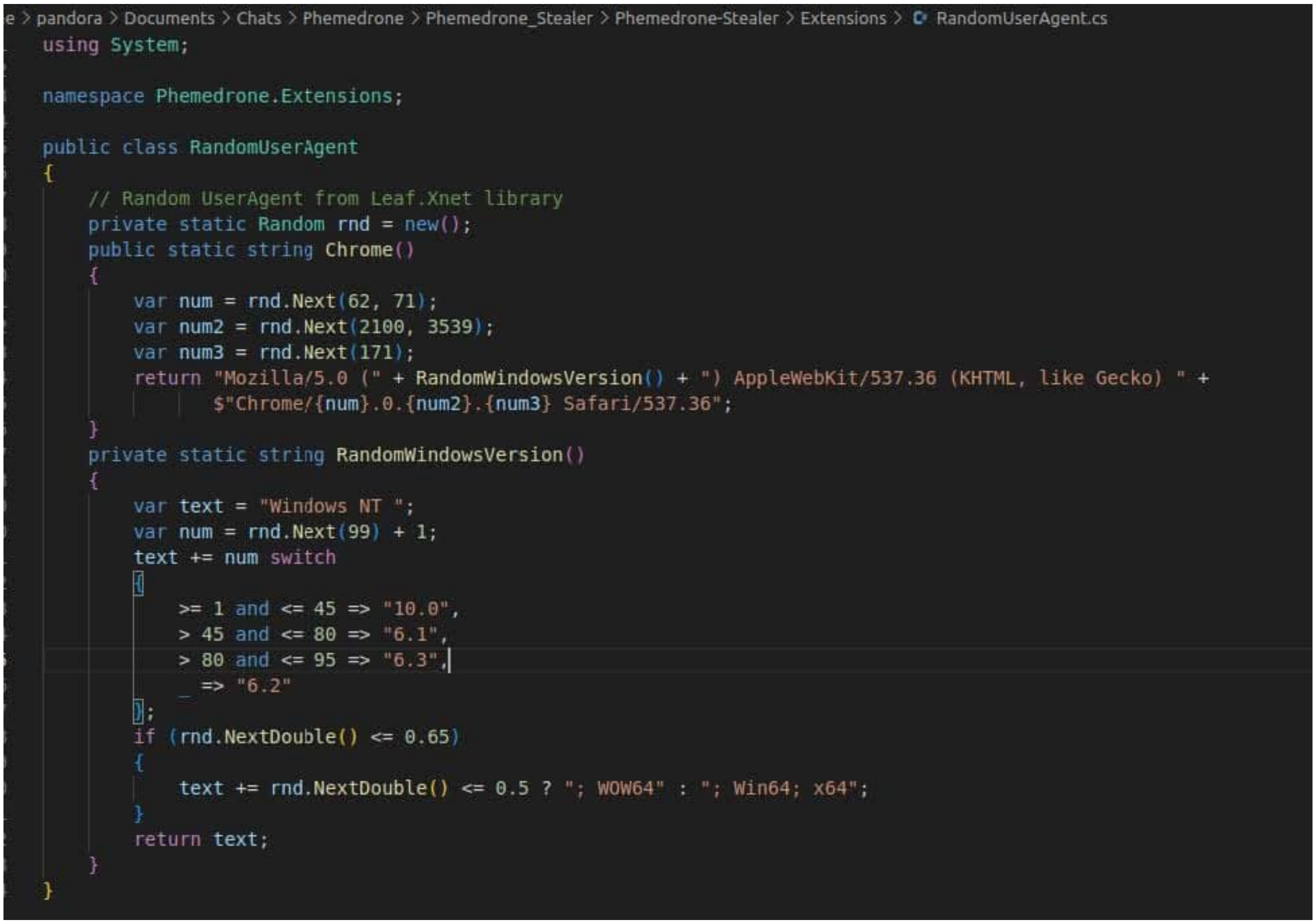
```
,d88b.d88b,
88888888888    Phemedrone Stealer
`Y8888888Y'    {DateTime.Now:dd/MM/yyyy HH:mm:ss}
  `Y888Y'      Developed by https://t.me/webster480 & https://t.me/TheDyer
    `Y'        Tag: {Config.Tag}
```

Image 2: The Phemedrone Stealer author tag added to the top of logs.

As observed in the above table, in the BANK section, both of the domains are for banks commonly used in Russia. Additionally, in the MONEY section, half (yoomoney, funpay) are services commonly used in Russia. As will be discussed in later sections, while this malware does have a CIS check in the binary, this check is an optional toggle switch during the creation of a bot and can easily be toggled off, allowing Phemedrone to target areas where the MONEY/BANK sections could be used to their fullest.

Useragent generation

As observed in the screenshot below, Phemedrone has the ability to generate random useragents, which it uses during communication with its C2. This possibly helps it sneak by detections that might rely on hardcoded useragent values.


e > pandora > Documents > Chats > Phemedrone > Phemedrone_Stealer > Phemedrone-Stealer > Extensions > RandomUserAgent.cs

```
using System;

namespace Phemedrone.Extensions;

public class RandomUserAgent
{
    // Random UserAgent from Leaf.Xnet library
    private static Random rnd = new();
    public static string Chrome()
    {
        var num = rnd.Next(62, 71);
        var num2 = rnd.Next(2100, 3539);
        var num3 = rnd.Next(171);
        return "Mozilla/5.0 (" + RandomWindowsVersion() + ") AppleWebKit/537.36 (KHTML, like Gecko) " +
               $"Chrome/{num}.0.{num2}.{num3} Safari/537.36";
    }
    private static string RandomWindowsVersion()
    {
        var text = "Windows NT ";
        var num = rnd.Next(99) + 1;
        text += num switch
        {
            >= 1 and <= 45 => "10.0",
            > 45 and <= 80 => "6.1",
            > 80 and <= 95 => "6.3",
            _ => "6.2"
        };
        if (rnd.NextDouble() <= 0.65)
        {
            text += rnd.NextDouble() <= 0.5 ? "; WOW64" : "; Win64; x64";
        }
        return text;
    }
}
```

Image 3: Code from Phemedrone which shows how it can easily change its useragent on the fly.

Anti-analysis checks

Phemedrone contains several anti-analysis checks which can be enabled during the build phase of the malware. If any of the checks described below are successful, Phemedrone exits.

Anti-debugger

Phemedrone’s anti-debugger check checks the victim’s environment for the following processes, which may indicate that Phemedrone is being debugged:

“wireshark”

- “httpdebuggerui”

“mtmproxy”

- “sniffer”

Anti-VM

Phemedrone’s anti-VM check checks the victim’s computer for the following virtual machine (VM) strings, which indicate that Phemedrone is being run in a VM:

- VirtualBox

- VBox

- VMware Virtual

- VMware

- Hyper-V Video

- Microsoft

CIS check

Phemedrone has a check that checks if a victim is a speaker of the following languages spoken in Commonwealth of Independent States (CIS) countries, by using a keyboard language check, as observed in Image 4:

- Russian

- Kazakh

- Moldovan

Uzbek

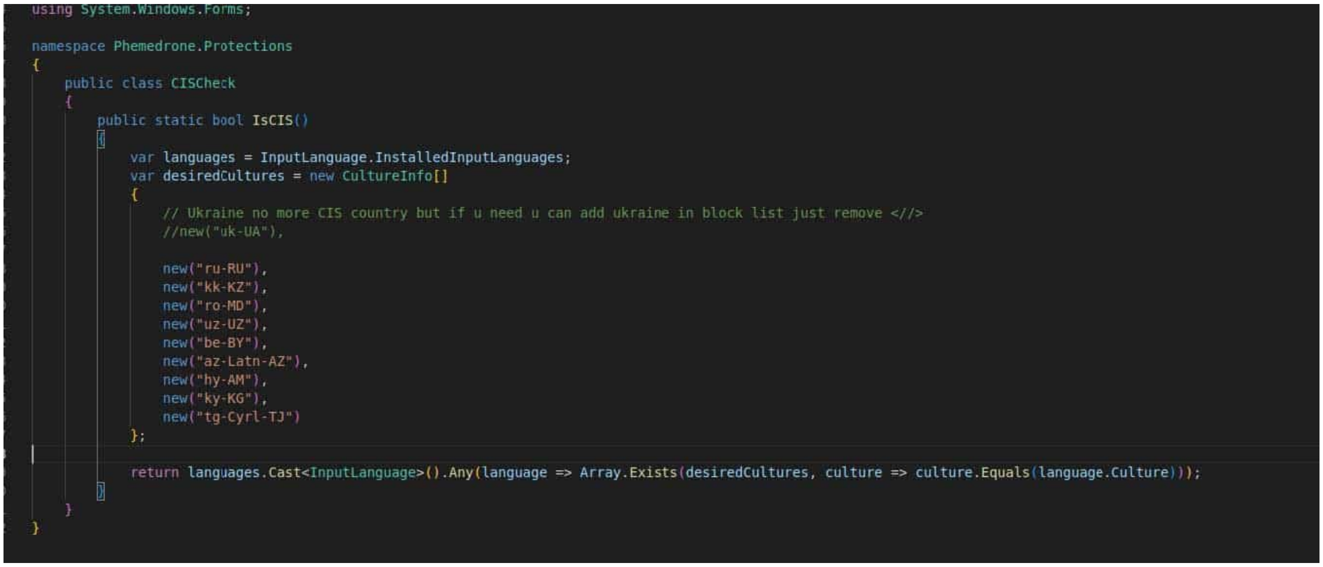
- Belarusian

- Azerbaijani

- Armenian

- Kyrgyz

- Tajik



```
using System.Windows.Forms;

namespace Phemedrone.Protections
{
    public class CISCheck
    {
        public static bool IsCIS()
        {
            var languages = InputLanguage.InstalledInputLanguages;
            var desiredCultures = new CultureInfo[]
            {
                // Ukraine no more CIS country but if u need u can add ukraine in block list just remove <//>
                //new("uk-UA"),

                new("ru-RU"),
                new("kk-KZ"),
                new("ro-MD"),
                new("uz-UZ"),
                new("be-BY"),
                new("az-Latn-AZ"),
                new("hy-AM"),
                new("ky-KG"),
                new("tg-Cyrl-TJ")
            };

            return languages.Cast<InputLanguage>().Any(language => Array.Exists(desiredCultures, culture => culture.Equals(language.Culture)));
        }
    }
}
```

Image 4: This is an optional check in the build process for a bot and is disabled by default.

Mutex check

Using the hardcoded config, Phemedrone checks to see if it is already running by checking to see if its mutex already exists.

Senders

Phemedrone’s bot builder has three different “sender” customization options, with some of the options behaving differently than the others. The three options are as follows:

- Gate - Sends logs to a php gate

- Panel - Sends logs to an IP/PORTcombo

- Telegram - Sends logs to a telegram channel

Gate sender

Phemedrone's gate sender allows actors using Phemedrone to specify a C2 that hosts the Phemedrone gate.php script. Bots that connect to this php gate will send their logs there, and then:

- Logs that are larger than 52MB will be stored on the server, and

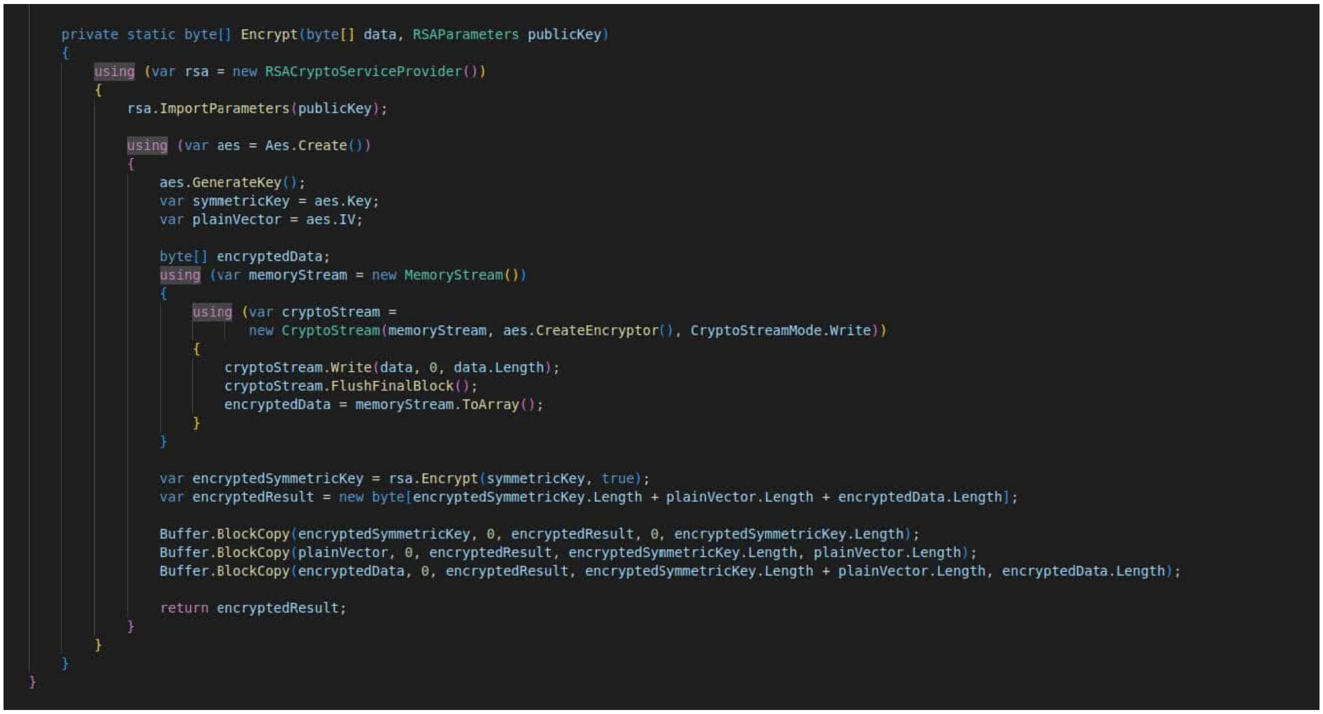
- Logs that are smaller than 52MB will be uploaded directly to Telegram.

Panel sender

Phemedrone’s panel sender allows actors to stand up a panel on a domain they control and then specify the IP/PORT combination when building their bot. This sender stores logs on the server, and then also notifies a Telegram chat when logs arrive. Connected victims as well as logs can be viewed in Phemedrone’s console-based panel application.

Telegram sender

Phemedrone’s Telegram sender allows actors to specify a Telegram channel/telegram bot as the preferred destination for exfiltrated logs. The Telegram sender also has an option to encrypt all logs sent with this method, so that the logs are not sitting in Telegram unencrypted. Phemedrone leverages a basic AES + RSA encryption algorithm for all logs, as observed in Image 5. Telegram exfil is an increasingly popular choice for malware, as well as phishing, and this encryption adds an extra layer of security for people choosing to use that option.



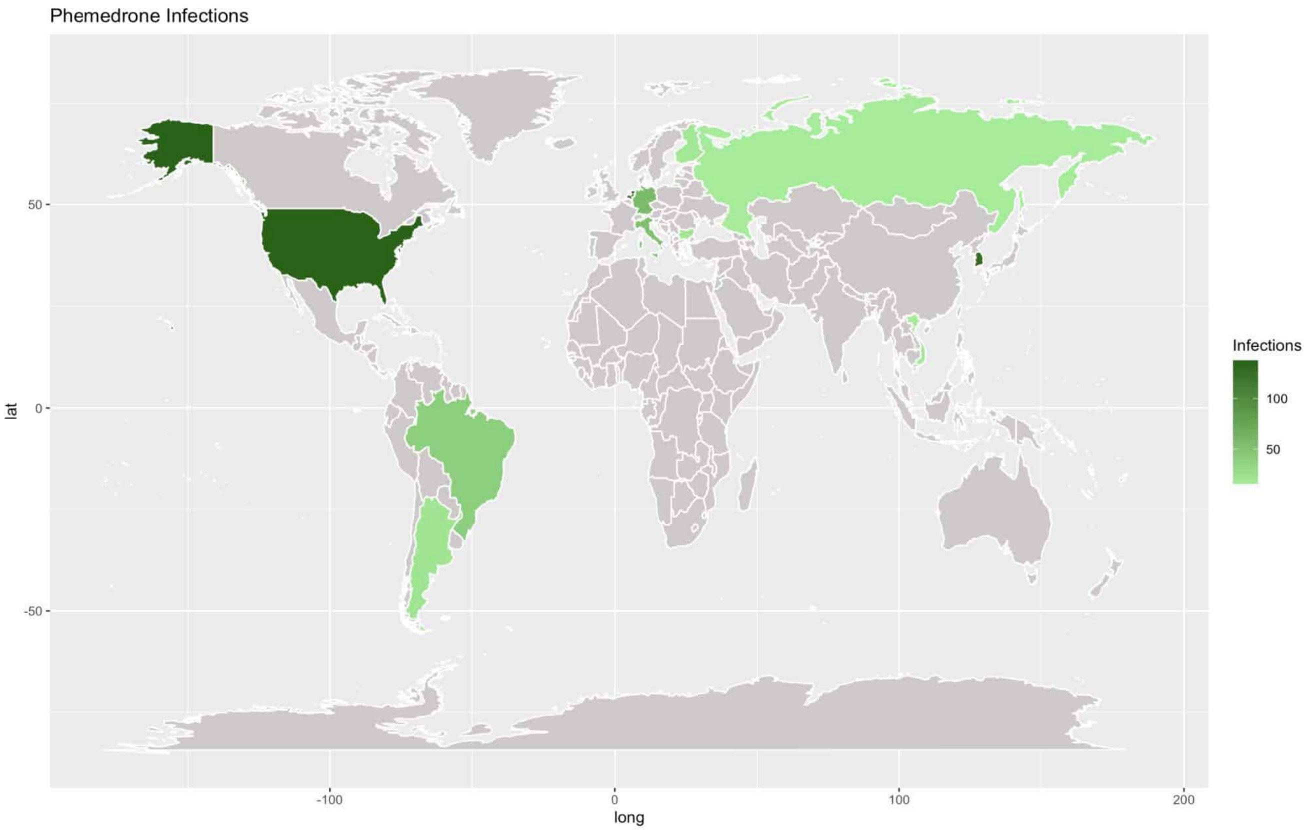
```
    private static byte[] Encrypt(byte[] data, RSAParameters publicKey)
    {
        using (var rsa = new RSACryptoServiceProvider())
        {
            rsa.ImportParameters(publicKey);

            using (var aes = Aes.Create())
            {
                aes.GenerateKey();
                var symmetricKey = aes.Key;
                var plainVector = aes.IV;

                byte[] encryptedData;
                using (var memoryStream = new MemoryStream())
                {
                    using (var cryptoStream =
                           new CryptoStream(memoryStream, aes.CreateEncryptor(), CryptoStreamMode.Write))
                    {
                        cryptoStream.Write(data, 0, data.Length);
                        cryptoStream.FlushFinalBlock();
                        encryptedData = memoryStream.ToArray();
                    }
                }

                var encryptedSymmetricKey = rsa.Encrypt(symmetricKey, true);
                var encryptedResult = new byte[encryptedSymmetricKey.Length + plainVector.Length + encryptedData.Length];

                Buffer.BlockCopy(encryptedSymmetricKey, 0, encryptedResult, 0, encryptedSymmetricKey.Length);
                Buffer.BlockCopy(plainVector, 0, encryptedResult, encryptedSymmetricKey.Length, plainVector.Length);
                Buffer.BlockCopy(encryptedData, 0, encryptedResult, encryptedSymmetricKey.Length + plainVector.Length, encryptedData.Length);

                return encryptedResult;
            }
        }
    }
}
```

Image 5: Code from Phemedrone shows that it can successfully encrypt information using AES+RSA.

Log analysis

Based on an overlap between behavior and log format, we’ve noticed that there are variants of Phemedrone with logs sold on forums. One of those variants is a family called “Mephedrone”.

Checking our logs, we’ve noticed that we most often see Phemedrone affecting the United States, with 20% of logs attributed to that country. A full breakdown of countries can be found in the image and corresponding table below:



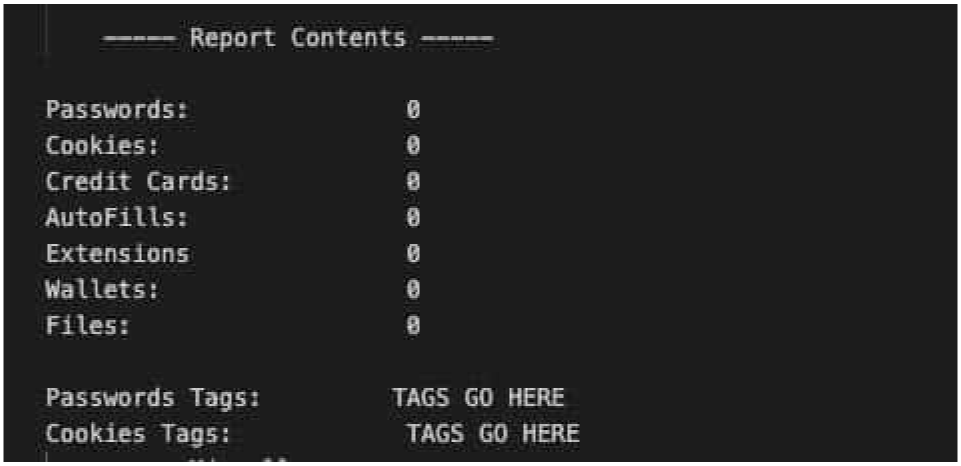
Country	Percentage
United States	20.00%
Netherlands	19.00%
Republic of Korea	18.58%
Germany	8.41%
Italy	7.67%
Brazil	5.9%
Israel	3.24%
Argentina	3.24%
Bulgaria	3.1%

Finland	2.95%
Singapore	2.8%
Vietnam	2.51%
Russia	2.36%

Interestingly, Russia consisted of 2% of the total infections, despite the CIS check in the malware.

Information file

A final interesting feature of Phemedrone is that – as it parses the passwords out of its respective password stores on the victim computer (instead of on a panel) – it's able to create snapshots in a generated Information.txt file, which allows actors to rapidly see which logs they've obtained. As observed in Image 6, the generated Information.txt file has a snapshot where log count can be observed:



```
    ----- Report Contents -----

Passwords:            0
Cookies:              0
Credit Cards:         0
AutoFills:            0
Extensions            0
Wallets:              0
Files:                0

Passwords Tags:         TAGS GO HERE
Cookies Tags:            TAGS GO HERE
```

Image 6: Phemedrone's Information.txt log snapshot, which shows what log counts can be observed.

Recap of findings

Phemedrone is an interesting case study in the evolution of infostealer families. As this article describes, there are several characteristics that make Phemedrone particularly attractive to cybercriminals:

- Phemedrone is an open source stealer that receives regular updates, making it accessible and reliable for criminals

- The stealer is highly configurable, making it flexible to bad actors’ needs

Unlike most malware sold in Russian-langauge cybercrime groups, Phemedrone is unique in its targeting of Russian banks and other service providers

While Phemedrone appears to be used to target Russian users and services, particularly in instances where banking or financial information can be harvested, the US is still the most affected country according to our research.

Defending against Phemedrone Stealer

User exposures from Phemedrone infections (even on personal devices) can threaten businesses if actors gain access to credentials and other identity data that opens doors to your environment. We recommend security teams integrate Post-Infection Remediation steps into existing malware remediation playbooks for confirmed exposures to minimize risk and prevent follow-on attacks like account takeover and fraud.

We’ll continue to monitor developments of Phemedrone’s capabilities and review recaptured logs to better understand exfiltration trends. Keep an eye out for more reverse-engineering analyses from our team at SpyCloud Labs.