Raccoon back with new claws!

labs.k7computing.com/index.php/raccoon-back-with-new-claws/

By Rahul R July 18, 2022

Raccoon infostealer was first released in April 2019, the initial Version1(V1) was distributed in telegram groups and other forums as Malware-as-a-service (MaaS). The detailed blog on V1 can be seen here. Now the stealer has been updated with new features, and comes packed with Commercial packers. It has a stealthy way of gaining information from the system using Windows API's. This blog discusses in depth on the Version2(V2) of Racoon Stealer and its method to obtain the information.

The Stealer is usually downloaded when a user tries to download cracked software, thus the malware is added with around 400MB of junk in the overlay along with an invalid digital signature from AVG.

Analysis

The sample is around **417MB** disguises itself as **Windows File System Proxy,** has an invalid digital Signature and comes packed with VMProtect. The analysis is based on the unpacked binary.

Property	Value
CompanyName	Navimatics LLC
FileDescription	Windows File System Proxy
FileVersion	1.9.21096.9d76495
InternalName	launchctl.exe
LegalCopyright	2015-2021 Bill Zissimopoulos
OriginalFilename	launchctl.exe
ProductName	WinFsp

Figure 1: Sample Version Information

Dynamic API Resolving

The malware begins with resolving the required API's dynamically through LoadLibrary and GetProcAddress.

```
push    offset aGetenvironment ; "GetEnvironmentVariableW"
mov     GetCurrentProcess_0, eax
mov     eax, GetProcAddress_2
push    esi
call    eax ; GetProcAddress_2
push    offset aGetfilesize ; "GetFileSize"
mov     GetEnvironmentVariableW, eax
mov     eax, GetProcAddress_2
push    esi
call    eax ; GetProcAddress_2
push    offset aGetdrivetypew ; "GetDriveTypeW"
mov     GetFileSize, eax
mov     eax, GetProcAddress_2
push    esi
call    eax ; GetProcAddress_2
push    offset aGetlasterror ; "GetLastError"
mov     GetDriveTypeW, eax
mov     eax, GetProcAddress_2
push    esi
call    eax ; GetProcAddress_2
push    offset aGetlocaleinfow ; "GetLocaleInfoW"
mov     GetLastError_0, eax
mov     eax, GetProcAddress_2
push    esi
call    eax ; GetProcAddress_2
push    offset aGetlogicaldriv ; "GetLogicalDriveStringsW"
mov     GetLocaleInfoW, eax
mov     eax, GetProcAddress_2
push    esi
call    eax ; GetProcAddress_2
```

Figure 2: Dynamic API resolving procedure

It uses LoadLibrary to get the handles of kernel32.dll, shell32.dll, user32.dll, advapi32.dll, wininet.dll, ole32.dll, crypt32.dll and pass on the returned handle as an argument to LoadLibrary to the get the address of the required WinAPI and stores them at a memory offset.

String Decryption

The sample uses the RC4 algorithm for decrypting the base64 strings stored in binary. At first the string is base64 decoded using CryptStringToBinary API passing the dwFlags argument as **CRYPT_STRING_BASE64(0x1).**

```
push    edi             ; lpString
call    eax ; lstrlenA
mov     ecx, LocalAlloc_1
add     eax, 40h ; '@'
push    eax             ; uBytes
push    40h ; '@'       ; uFlags
mov     [ebp+len_deb64_str], eax
call    ecx ; LocalAlloc_1
mov     ecx, lstrlenA
mov     ebx, eax
mov     esi, CryptStringToBinaryA
lea     eax, [ebp+len_deb64_str]
push    0
push    0
push    eax
push    ebx
push    CRYPT_STRING_BASE64
push    edi             ; lpString
call    ecx ; lstrlenA
push    eax
push    edi
call    esi ; CryptStringToBinaryA
```

Figure 3: Base64 decode using CryptStringToBinaryA

The decoded base64 string is saved in a variable and it is passed as an argument to the function which RC4 decrypts the string using the hardcoded symmetric key "**edinayarossiya**"("United Russia" – a political party in Russia)

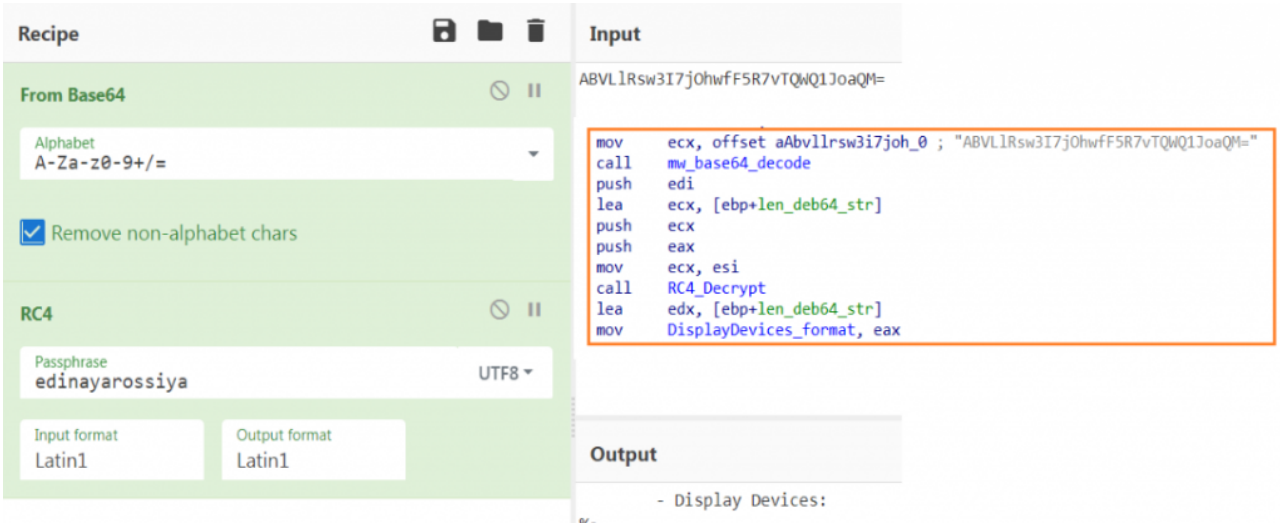

Figure 4: String decryption procedure

Complete list of strings decrypted is listed in Appendix A.

Retrieve C2 URL

The binary uses the same string decryption method discussed above to retrieve the C2 URL. For the decryption of the C2 it uses a different hardcoded RC4 symmetric key "**b616297870490e1028b141f53eb3afe8**" which is later used as config ID when initial information is sent.

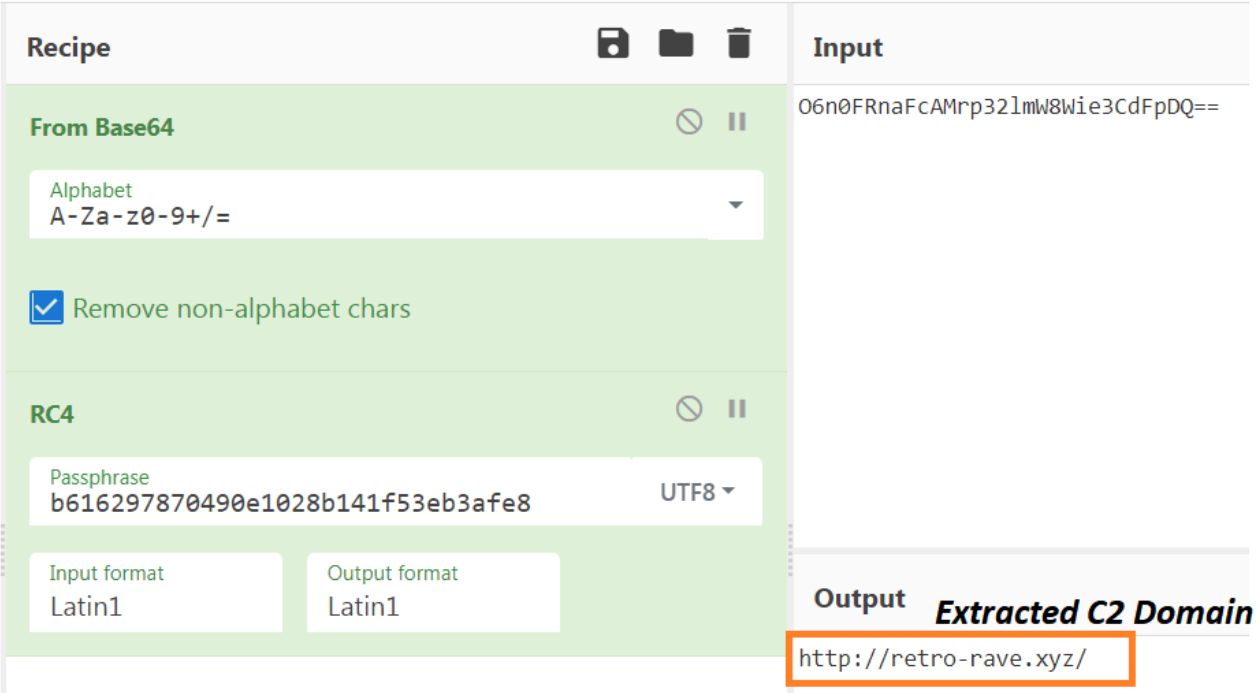

Figure 5: Decryption of Command and control server

Checks system locale

The malware then proceeds to check the locale of the system using

GetUserDefaultLocaleName API, and checks the returned string with a dword from virtual address 0x40E000. In this variant, this locale check does not affect the behaviour of the malware. Usually threat actors opt for an option for excluding victims from certain geolocale. Seems like the threat actors here have that option but are not using it.

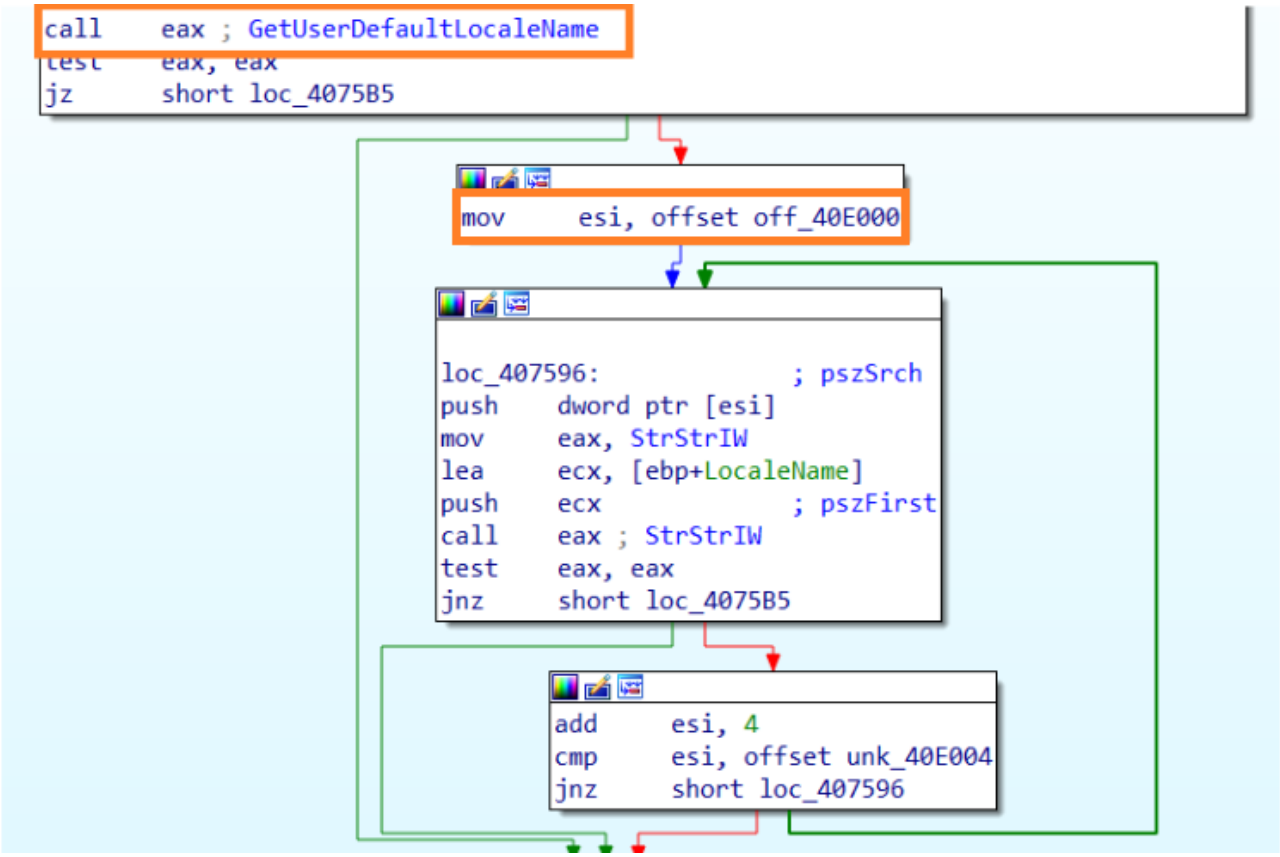

Figure 6: Get Locale of Execution system

Checks mutex

The malware checks for a mutex with name "**8724643052**", if not, then creates one. If the mutex exists then it kills itself to stop itself from running multiple times.

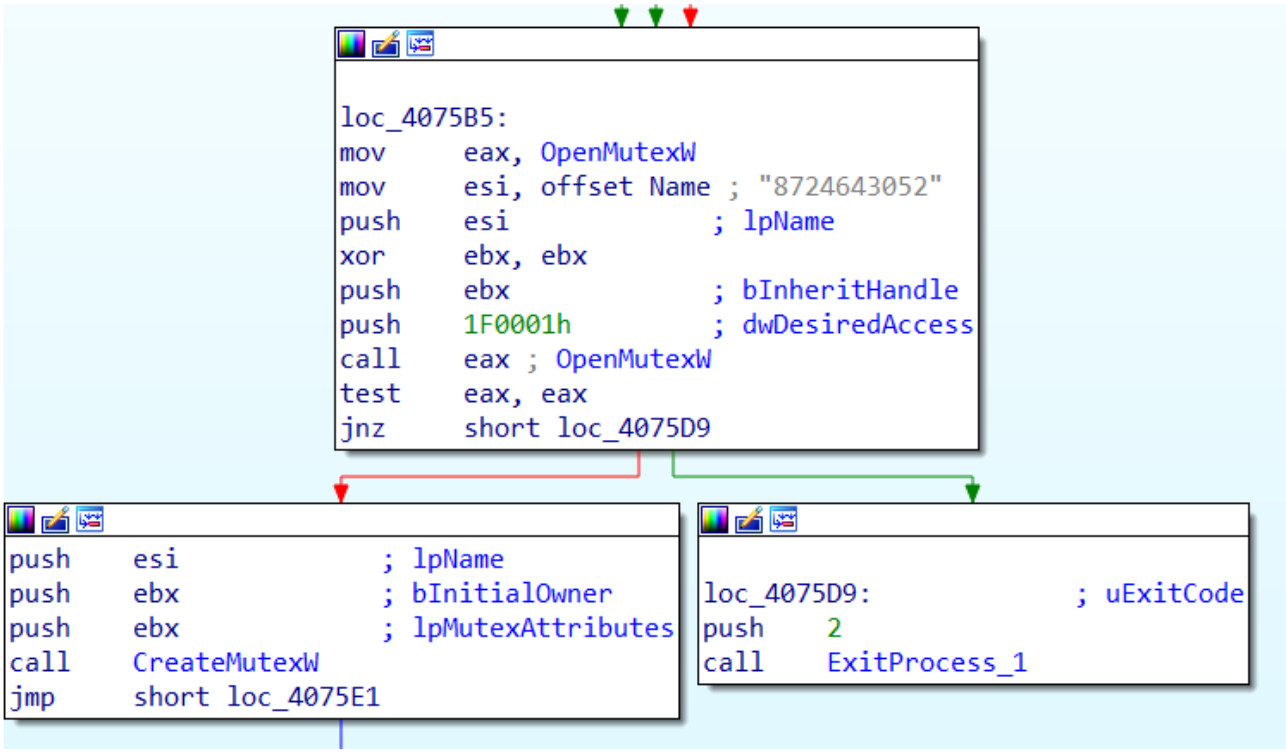

Figure 7: Malware checks if Mutex Exists

Checks for system privilege

The malware retrieves the Current Process access token and compares it to the SID of **NTAuthority\System("S-1-5-18")**. If it matches it executes the function to enumerate the active process list.

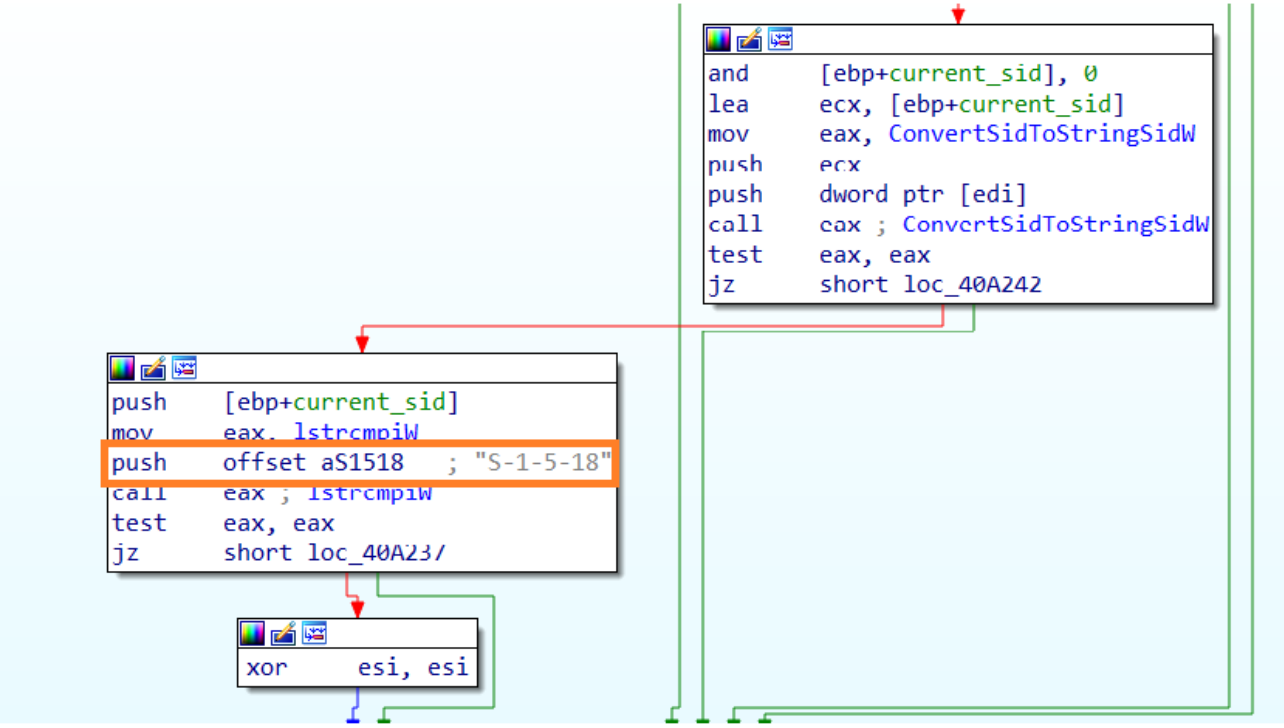

Figure 8: Check System Privilege

Similar to locale check, there is no change in behaviour

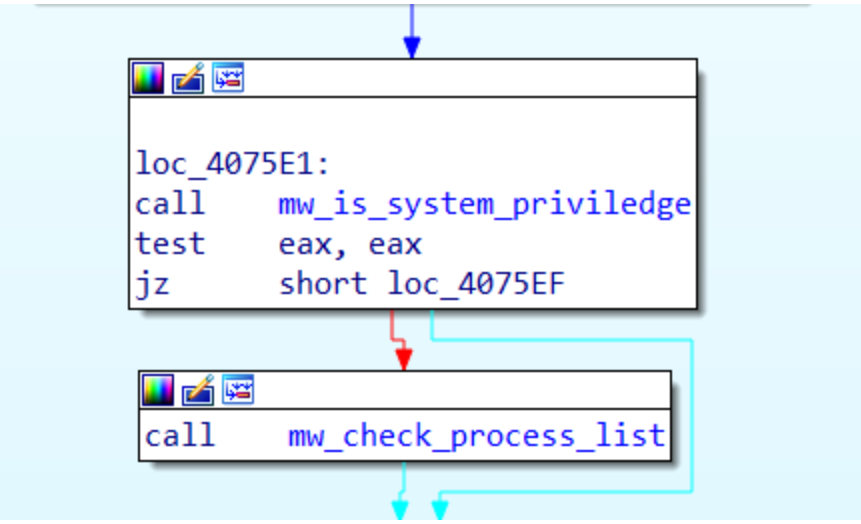

Figure 9: Enumerate process list if it has System privilege

Gather Initial Information

The malware initially collects machine GUID, username and sends it to C2 and awaits response from C2 for further information gathering.

Machine GUID is obtained from the registry key “HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Cryptography” under “MachineGUID”

The malware sends the initial information to C2 in the following syntax

machineId=<machineGUID>|<username>&configid=<RC4_key used to decrypt C2>

Sends initial collected data

After converting the collected initial data into Unicode string. It sends a POST request to the decrypted C2 using an unusual User-Agent String “**record**”. The data is sent in form data format.

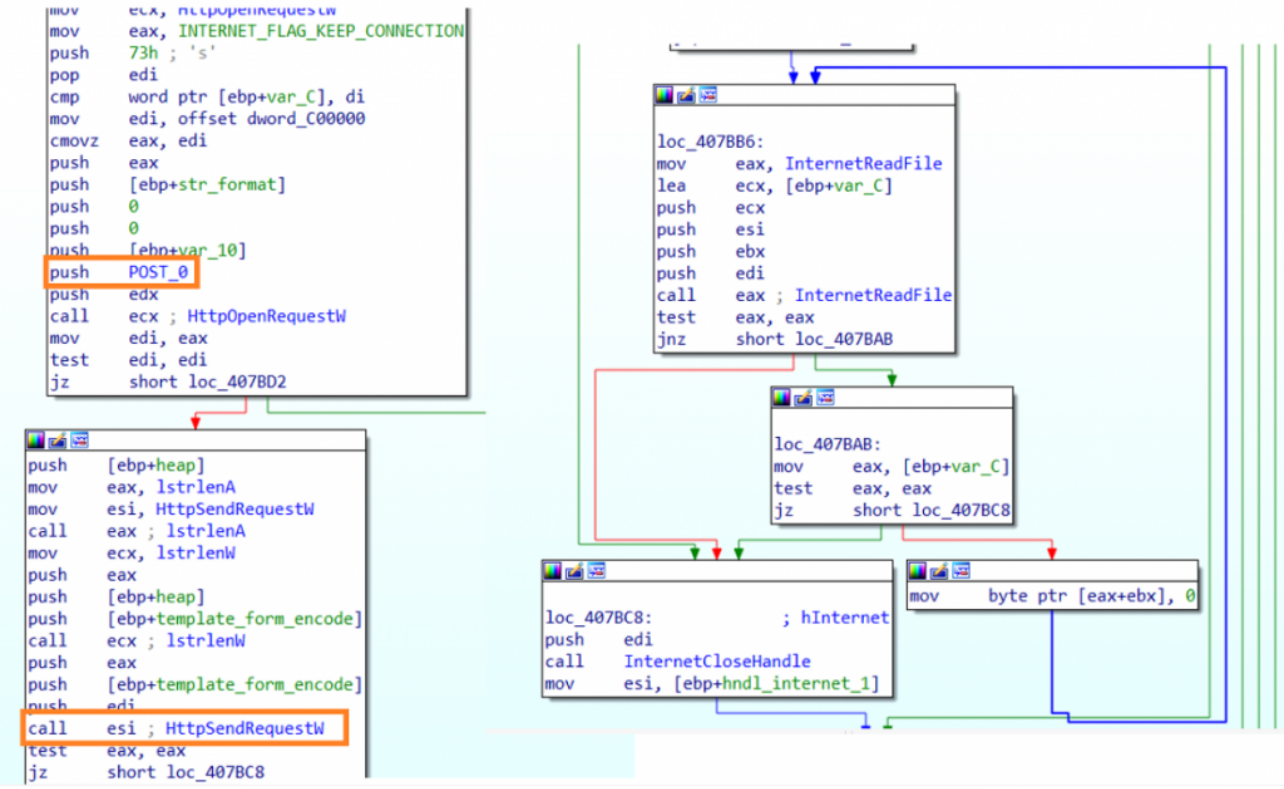

Figure 10: Procedure to send request to C2 and wait for response

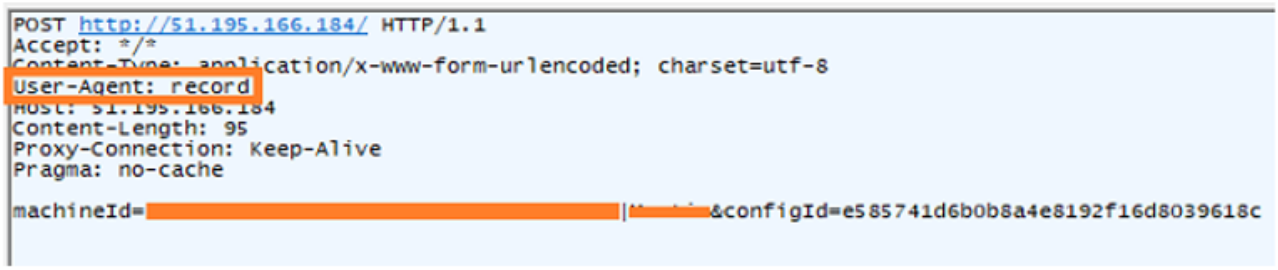

Figure 11: Request sent to C2

After making the request the connection handle is kept open until it receives a data response. It waits for the POST response until the size of response is greater than 64 bytes.

Process C2 Response

```
libs_nss3:http://94.158.247.24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/nss3.dll
libs_msvcp140:http://94.158.247.24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/msvcp140.dll
libs_vcruntime140:http://94.158.247.24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/
vcruntime140.dll
libs_mozglue:http://94.158.247.24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/mozglue.dll
libs_freebl3:http://94.158.247.24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/freebl3.dll
libs_softokn3:http://94.158.247.24/aN7jD0qO6kT5bK5bQ4eR8fE1xP7hL2vK/softokn3.dll
ews_meta_e:ejbalbakoplchlghecdalmeeeajnimhm;MetaMask;Local Extension Settings
ews_tronl:ibnejdfjmmkpcnlpebklmnkoeoihofec;TronLink;Local Extension Settings
libs_sqlite3:http://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;*;-
```

Figure 12: C2 response

The C2 response contains the urls of the dlls which are needed to collect detailed information

A GET request is made to download all the Dll and it is saved in the APPDATA_LOCAL folder.The path to APPDATA_LOCAL is retrieved using the API SHGetFolderPath with CSIDL passed as "**CSIDL_LOCAL_APPDATA**"**(0x1c).**.If the response doesn't have the String "**Token**" in it the malware kills itself.

Collect detailed information

After downloading the required dlls, it changes the current working directory and adds the path to the APPDATA_LOCAL directory to "PATH" Environment Variable using SetEnvironmentVariableW.

System Info.txt

The malware first collects the system information and sends it as a POST request to the C2. Let us see what and how the system information is collected using WinAPI.

Locale : The malware collects the current locale using the API **GetLocaleInfoW**

TimeZone : Timezone is retrieved using API GetTimeZoneInformation

```
mov     edi, eax
push    104h            ; cchData
push    ebx             ; lpLCData
push    1001h           ; LCType
call    ds:dword_40C00C
push    eax             ; Locale
call    esi ; GetLocaleInfoW
push    ebx
push    locale_format_0
push    edi
call    wsprintfW
mov     esi, [ebp+arg_0]
add     esp, 0Ch
mov     edx, edi
mov     ecx, [esi]
call    mw_strcat
```

```
push    edi
lea     eax, [ebp+TimeZoneInformation]
push    eax             ; lpTimeZoneInformation
call    GetTimeZoneInformation
push    400h
push    40h ; '@'
call    eax ; LocalAlloc_1
mov     edx, [ebp+TimeZoneInformation.Bias]
xor     ecx, ecx
neg     edx
mov     edi, eax
push    edx
push    2Bh ; '+'
pop     eax
test    edx, edx
cmovg   ecx, eax
push    ecx
push    TimeZone_format_0
push    edi
call    wsprintfW
```

Figure 13: collection of Locale and TimeZone using API

Product Name : Windows version is retrieved by querying the registry key **"HKEY_LOCAL_MACHINE\software\microsoft\windows nt\currentversion\"** and data **"ProductName"**

```
push    20119h
push    esi
push    registry_currentVersion_0
push    80000002h
call    ecx ; RegOpenKeyExW
test    eax, eax
jnz     short loc_408FFE
```

```
lea     eax, [ebp+cbData]
push    eax             ; lpcbData
push    edi             ; lpData
push    esi             ; lpType
push    esi             ; lpReserved
push    ProductName_0   ; lpValueName
push    [ebp+hKey]      ; HKEY_LOCAL_MACHINE\software\microsoft\windows nt\currentversion\
call    RegQueryValueExW
```

```
loc_408FFE:             ; hKey
push    [ebp+hKey]
call    RegCloseKey
mov     eax, lstrlenW
```

Figure 14: retrieve productname from registry

Architecture : The malware checks if SYSWOW64 directory exists on the system, if it is unavailable it considers the architecture as 32bit, else architecture is 64bit.

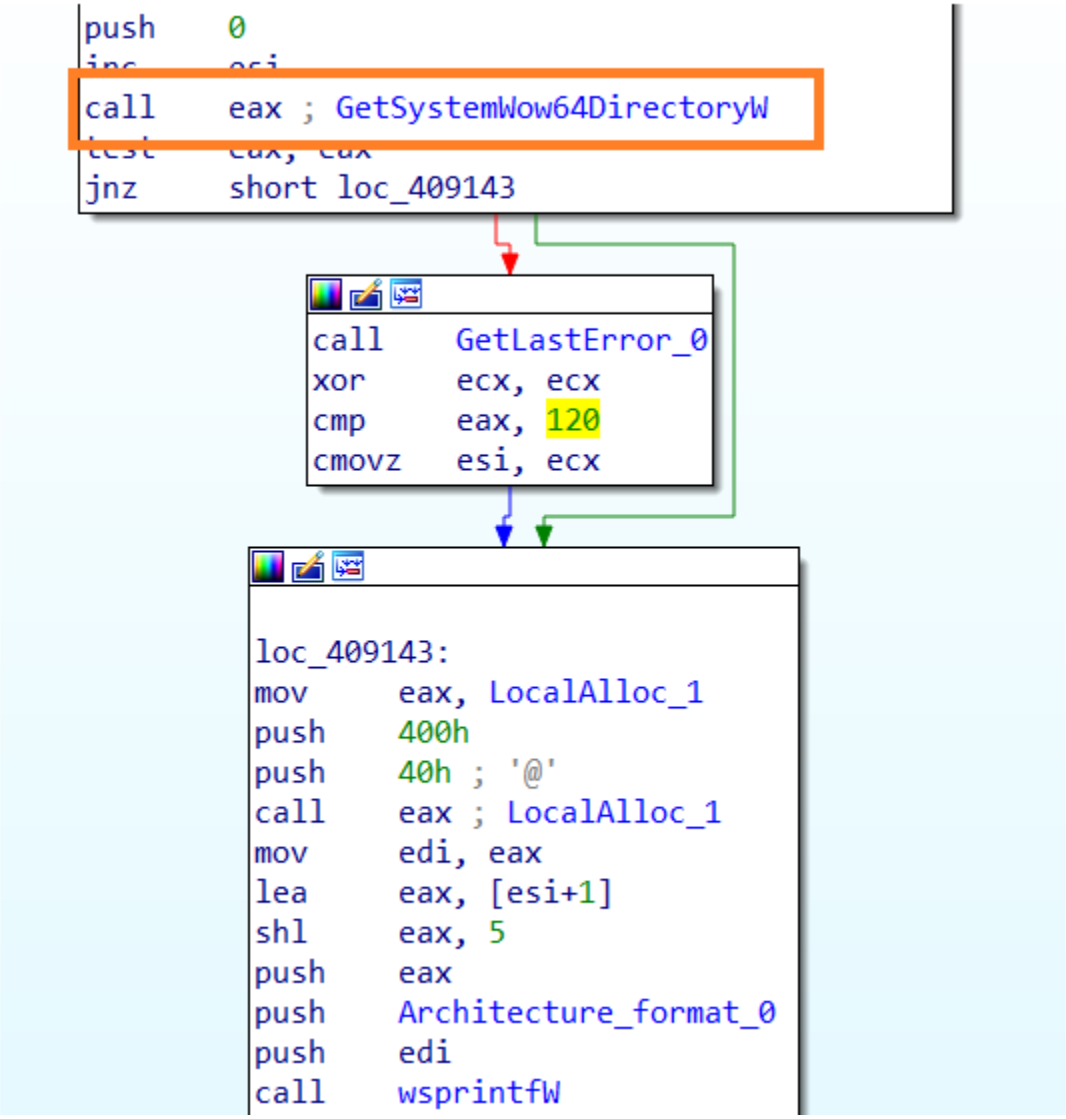

Figure 15: Find system architecture

Processor : The processor information is obtained with the usage of ASM instruction “CPUID”(CPU Identification).

RAM : The Exact amount of physical storage is retrieved using the API GlobalMemoryStatusEx,which returns the “LPMEMORYSTATUSEX” structure. From the returned structure the malware takes the field “ullTotalPhys” and right shift by 20 bits to convert it into MB.

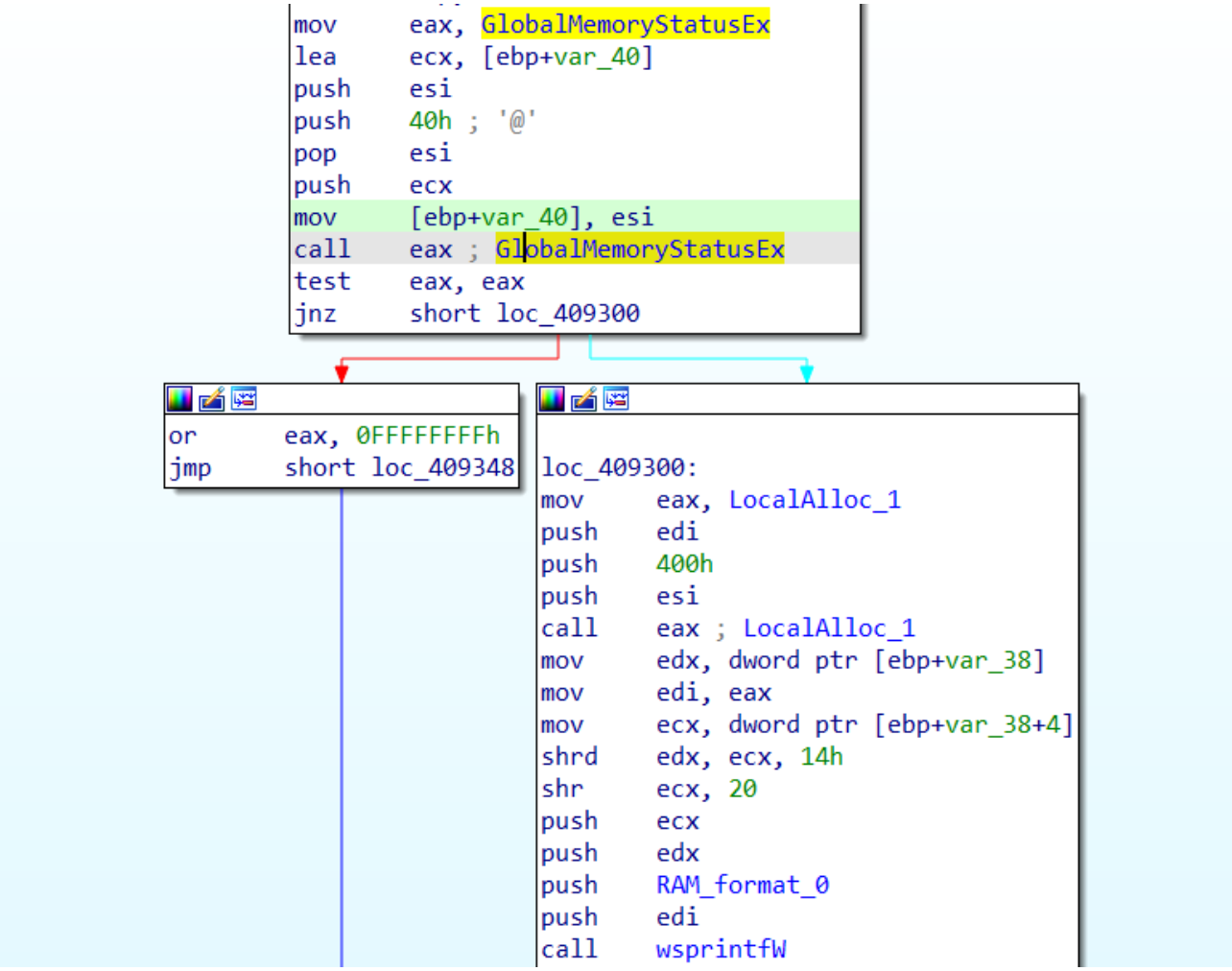

Figure 16: Get RAM information

Display height and width : Display height and width is obtained using the API "GetSystemMetrics" by passing the argument **0x0(SM_CXSCREEN)** to retrieve width and **0x1(SM_CYSCREEN)** to get height.

Display Devices : The display enumerated and saved using the API "**EnumDisplayDevicesW**"

Screen dimension and display devices could be checked at the server if the malware is executed in a VM or sandbox.

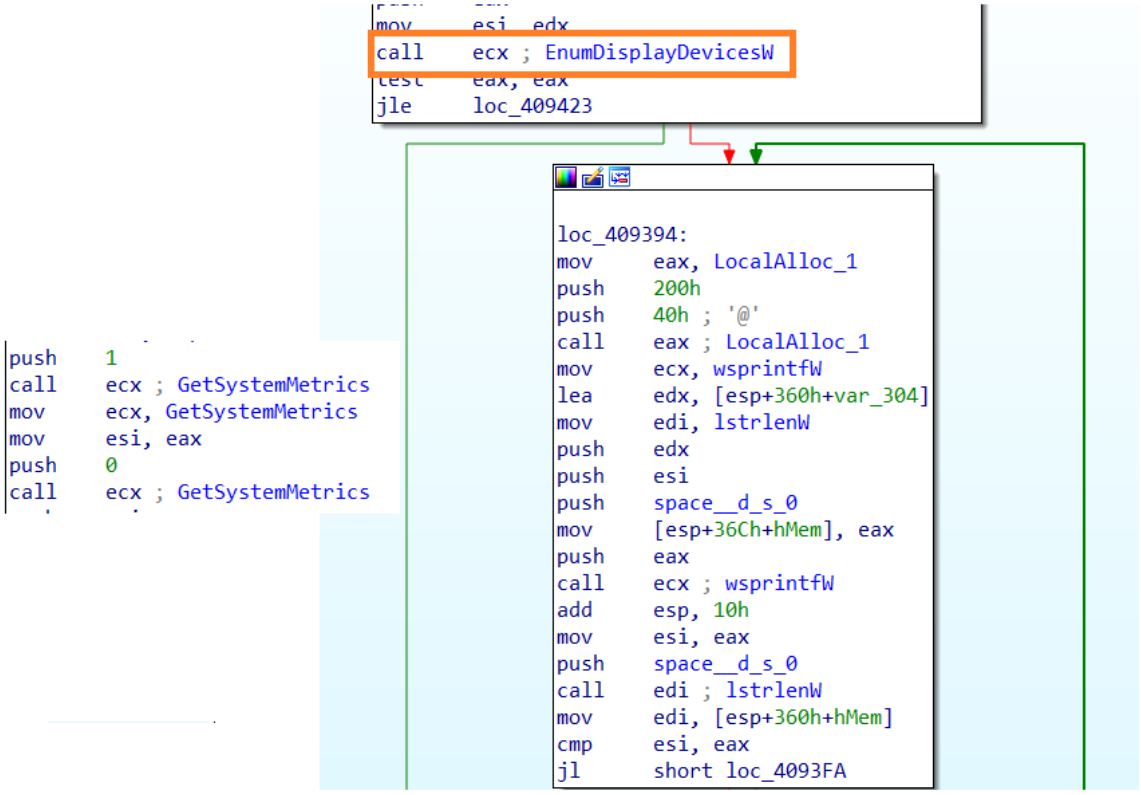

Figure 17: Get Display information

List of Installed Products : The complete list of products which are installed are obtained by looping through all the subkeys under "HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall"

All the collected information about the System is sent immediately to the C2 without saving it to a file.

Cookies.txt

After collecting all the information related to system, it proceeds to collect browser saved passwords, credit card details and cookies using the following dll

1. Sqlite3.dll – to collect login id and passwords from chrome(ium) based browsers
2. mozglue.dll/nss3.dll – to collects login id and passwords from firefox

The following queries are used to query the required information.

- 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
- SELECT host, path, isSecure, expiry, name, value FROM moz_cookies
- SELECT fieldname, value FROM moz_formhistory

- SELECT name_on_card, card_number_encrypted, expiration_month, expiration_year FROM credit_cards

The Stealer even has the capability to collect the crypto wallets if found on the system and sends all the collected information to C2 immediately.

Captures screenshot

A series of Windows API is used to capture the screenshot of the infected machine, and is sent to C2. The flow is similar to the example code given by microsoft here.

Cleanup

The malware deletes all the files which are downloaded from the internet, after the information is sent to C2.

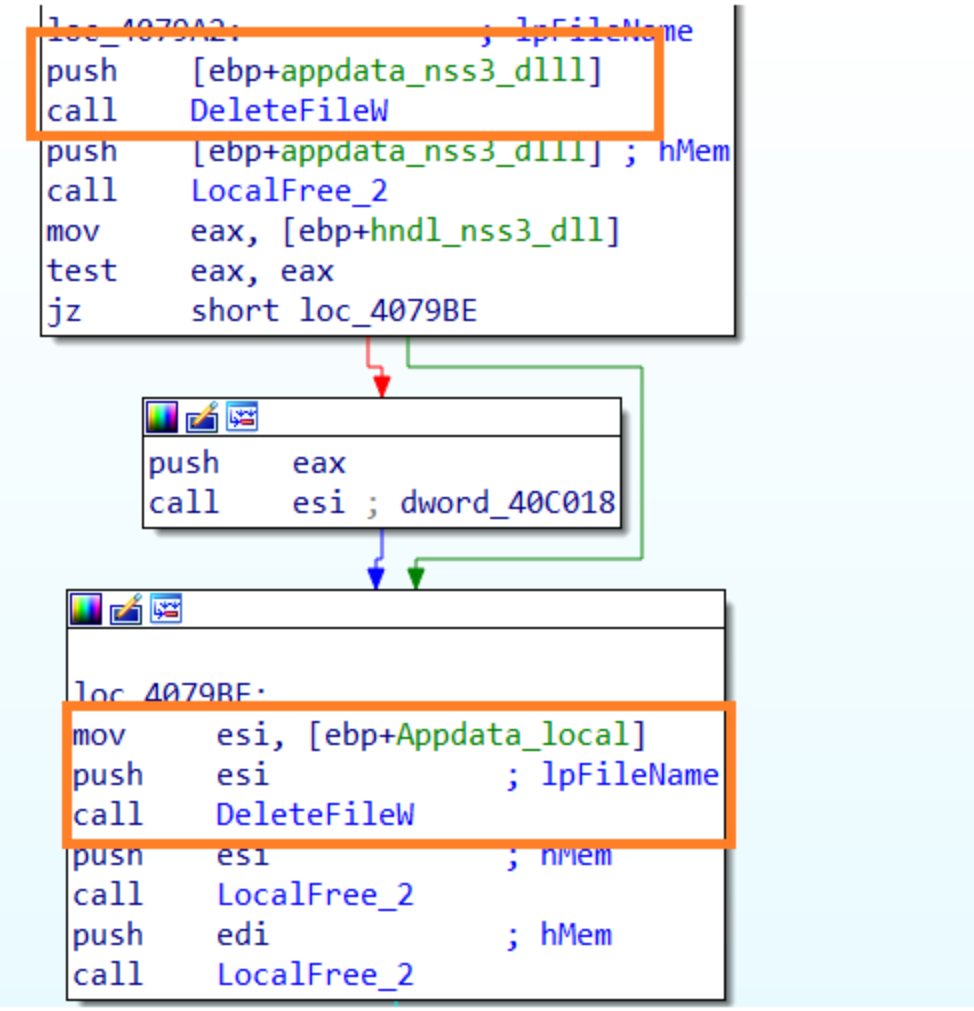

Figure 18: Cleanup

Activity

We strongly recommend not to download any cracked software to get infected with malware.

We at K7 Labs provide detection against latest threats and also for this newer variant of Racoon Stealer. Users are advised to use a reliable security product such as **“K7 Total Security”** and keep it up-to-date so as to safeguard their devices.

Indicators of Compromise(IOC)

File Name	Hash	K7 Detection Name
launchctl.exe	b0bc998182378e73e2847975cc6f7eb3	Trojan (005690671)

C2

hxxp://www[.]retro-rave[.]xyz

IP

51.195.166[.]184

User-Agent

record

Appendix : Strings Decrypted during Runtime (Using RC4 key: “edinayarossiya”)

tlgrm_

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_

scrnsht_

sstmnfo_

token:

nss3.dll

sqlite3.dll

SOFTWARE\Microsoft\Windows NT\CurrentVersion

PATH

ProductName

Web Data

sqlite3_prepare_v2

sqlite3_open16

sqlite3_close

sqlite3_step

sqlite3_finalize

sqlite3_column_text16

sqlite3_column_bytes16

sqlite3_column_blob

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

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