AZORULT

TECHNICAL ANALYSIS REPORT

ZAYOTEM

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

Contents

Overview

Azorult is a Trojan horse that steals information and has been in use since 2016. It is distributed mostly through spam emails, although there are many versions. With the 2018 update, the Azorult malware can also act as an installer. It has been updated to delete itself after it has fulfilled its information exfiltration function.

Azorult malware searches browsers, email and FTP servers for saved passwords, cookies, cryptocurrency wallet files, message history of messaging applications such as Skype, desktop files, lists of running processes, usernames, computer information, and sends this information to the command-and-control server.

This malicious software, once it infects a computer, provides access to:

- Credentials saved in web browsers,
- Messaging clients,
- FTP/SSH clients,
- Cryptocurrency wallets,
- System information,
- Desktop file.

Azorult.exe Analysis

Name	Azorult.exe
MD5	64CE3428700D7A0797CC4D779AC37C39
SHA256	6fa0833240b9e814ed3640ef92ae275eb3741b19358f46779a768ec6f5151c42
File Type	Portable Executable 32

The original name of the malware "6fa0833240b9e814ed3640ef92ae275eb3741b19358f46779a768ec6f5151c42.exe" has been changed to **"Azorult.exe"** for convenience during the analysis.

Static Analysis

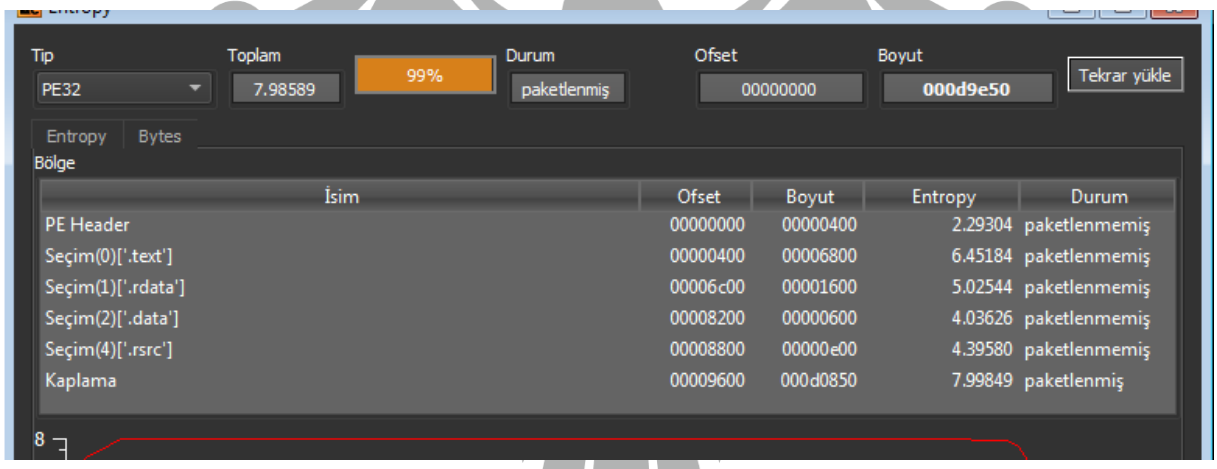

Figure 1 Examination of the Malware in the DIE Tool

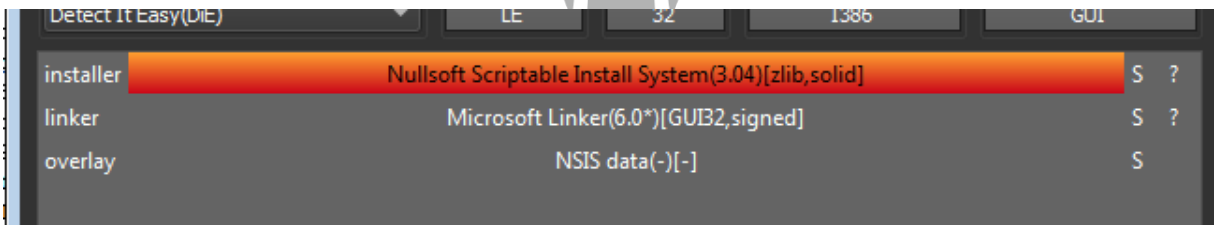

Figure 2 Nullsoft Scriptable Install System

When examining the Azorult.exe malware using the DIE Tool, it appears **packed.** In addition, the malware has a **NSIS (Nullsoft Scriptable Install System)** based structure.

Kernel32.dll	User32.dll	Gdi32.dll
Shell32.dll	Advapi.dll	Comctl32.dll
Ole32.dll		

Table 1 DLLs Used by The Malware

The DLLs used by the malware are shown in Table 1.

000000000909E	00000049C89E	0	Comments
0000000090B0	00000049C8B0	0	frankotvang deguelia vigter
0000000090EE	00000049C8EE	0	FileDescription
000000009110	00000049C910	0	reprosper affrdigelserne
00000000914A	00000049C94A	0	InternalName
000000009164	00000049C964	0	kondensvandet capetonian.exe
0000000091A6	00000049C9A6	0	LegalCopyright
0000000091C4	00000049C9C4	0	vurderingstidspunkternes papirstrrelsers kabinescooterne

Figure 3 Examining Strings of the Malware in the bintext Tool

After analysing the strings, it was discovered that the file description was **"reprosper affrdigelserne"** and the internal name was **"kondensvandet capetonian.exe"**.

Dynamic Analysis

Uxtheme.dll	Userenv.dll	Setupapi.dll
Apphelp.dll	Propsys.dll	Dwmapi.dll
Cryptbase.dll	Oleacc.dll	Clbcatq.dll
Ntmarta.dll		

Table 2 Dynamically Extracted DLLs

Some DLLs used by the malware are seen when examined dynamically. These DLLs are shown in Table 2.

```
.   50                  push eax
.   FF75 0C             push dword ptr ss:[ebp+C]          [ebp+C]:L"C:\\User
.   66:0155 FC          add word ptr ss:[ebp-4],dx
.   FF15 DC804000       call dword ptr ds:[<&GetTempFileNameW>]
.   85C0                test eax,eax
.   75 0D               jne azorult.405E2F
.   85FF                test edi,edi
```

Figure 4 Using the GetTempFileNameW API to Create the TMP File

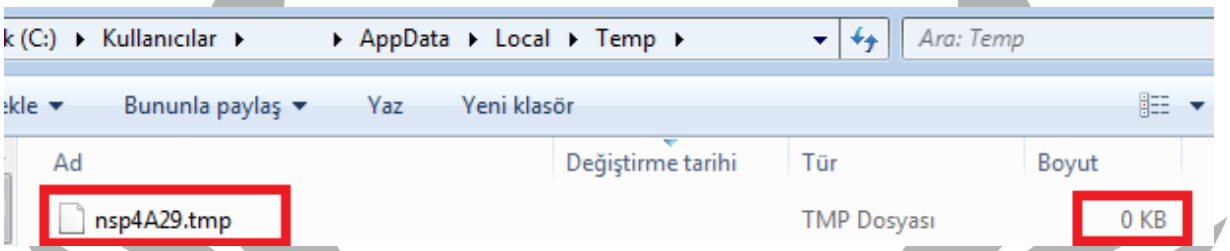

Figure 5 TMP File Created in the Temp Folder

The malware creates a **0KB TMP** file in the "**C:\Users\%username%\AppData\Local\Temp"** path. It generates a unique name for this file each time using the **GetTempFileNameW** API. The names of these files always start with "ns".

```
62D   0F84 CB000000     je azorult.4036FE
633 > 68 00104400       push azorult.441000                441000:L"C:\\User
638   FF15 40814000     call dword ptr ds:[<&DeleteFileW>]
63E   FF7424 1C         push dword ptr ss:[esp+1C]
642   E8 96F8FFFF       call <azorult.sub_402EDD>
```

Figure 6 Deleting the TMP File Using the DeleteFileW API

It then deletes this TMP file it created using the **DeleteFileW** API.

```
.   56                  push esi                                esi:L"Fugitated Setup"
.   E8 95250000         call <azorult.sub_4062DC>
.   56                  push esi                                esi:L"Fugitated Setup"
.   FF35 28D24200       push dword ptr ds:[42D228]
.   FF15 5C824000       call dword ptr ds:[<&SetWindowTextW>]
.   8BC6                mov eax,esi                             esi:L"Fugitated Setup"
```

Figure 7 Changing the Window Title Bar Using the SetWindowTextW API

It then dynamically resolves the **"Fugitated Setup"** string, which it sets as the title bar of the installer window that will open when it runs malicious with the **SetWindowTextW** API.

```
.v  74 1B           je azorult.403D34
.   8BF8            mov edi,eax                     eax:L"Nondistributively"
>   8B06            mov eax,dword ptr ds:[esi]      eax:L"Nondistributively"
.   85C0            test eax,eax                    eax:L"Nondistributively"
.v  74 0A           je azorult.403D2B
.   50              push eax                        eax:L"Nondistributively"
.   8D46 18         lea eax,dword ptr ds:[esi+18]   eax:L"Nondistributively"
.   50              push eax                        eax:L"Nondistributively"
.   E8 B1250000     call <azorult.sub_4062DC>
>   81C6 18080000   add esi,818
.   4F              dec edi
```

Figure 8 Resolving Strings

The malware generates strings, both meaningful and meaningless, using a loop. These strings are resolved in a consistent order each time the malware is executed. Table 3 shows the first ten strings in this order.

Stnderforsamlingen
Concourse206
Sanses
Hviderne
Nondistributively
Gioldaoram
Stokkepryglenes
Bevidsthedsniveau
Spillebordene
Prolongeredes

Table 3 Dynamically Resolved Strings

```
50                  push eax
FF35 E04E4300       push dword ptr ds:[434EE0]
FF15 44824000       call dword ptr ds:[<&DialogBoxParamW>]
6A 05               push 5
8BF0                mov esi,eax
E8 C9D7FFFF         call <azorult.sub_40140B>
6A 01               push 1
```

Figure 9 DialogBoxParamW API

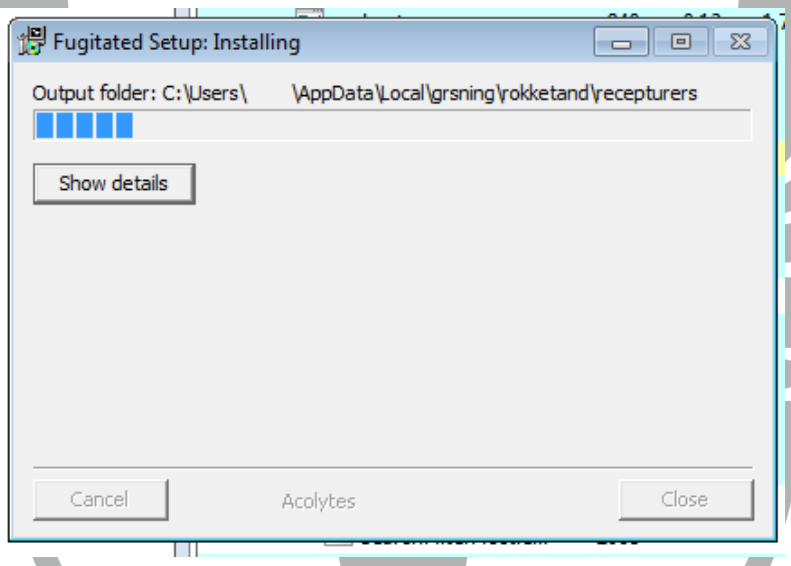

Figure 10 "Fugitated Setup" Installation Window

Then, the setup file dialogue box is created using the **DialogBoxParamW** API. Then a folder named **"grsning"** is created in the file path **"C:\Users\%username%\AppData\Local"**. Inside this folder there are various nested folders and files.

The following shows the names and hierarchical location of the created files.

- grsining
  - rokketand
    - recepturers
      - Daddelpalmers
        - Studiesituationerne
          - afslapningsvelsernes.dip
  - Enakter
    - Astigmatikeren
      - Gladiatorism
        - Eklektikerne.smu
        - Perceptiveness.Puc
        - Vremaaderne.txt
  - masseskrivelse
    - Sukkerskeerne
      - Ssygen126
        - equison.mul
        - lydhrt.non
  - rflen.pol
  - skipperlgnenes.toe
  - tallotterier.cut

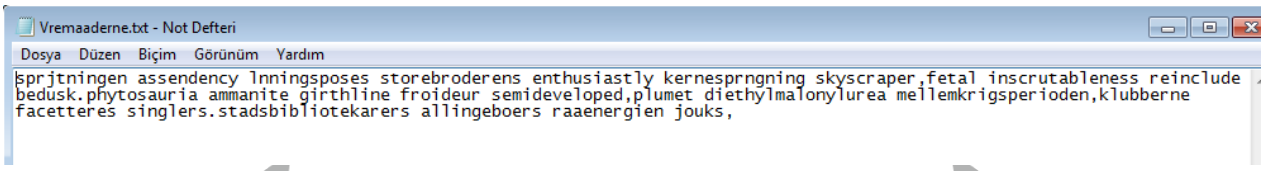

Figure 11 Contents of the file "Vremaaderne.txt"

When the text file with the file path **"grsining\rokketand\recepturers\Enakter\Astigmatikeren\Gladiatorism\Vremaaderne.txt"** is opened, it contains the text shown in Figure 11. This text contains some Danish words but does not make sense.

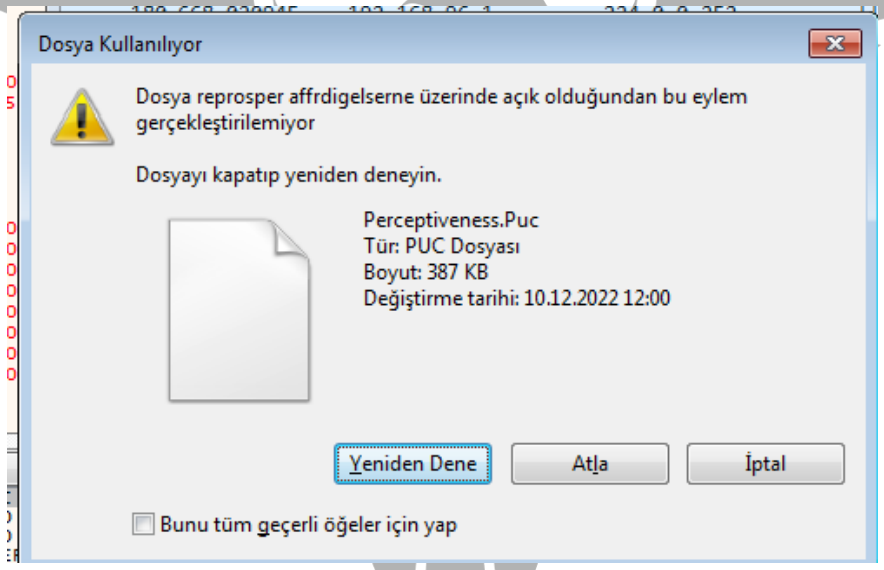

Figure 12 PUC File Continued Use

After closing or hiding the installation screen, only the binary data type file with PUC extension located at **"grsining\rokketand\recepturers\Enakter\Astigmatikeren\Gladiatorism\Perceptiveness.Puc"** is used.

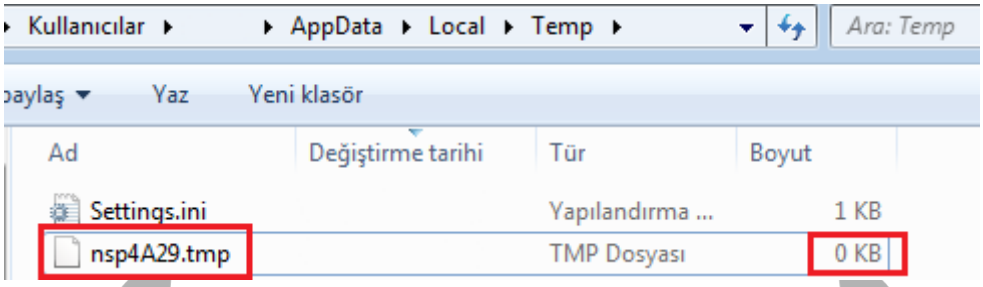

Figure 13 Generated TMP and Settings.ini File

However, in the file path **"C:\Users\%username%\AppData\Local\Temp",** a **TMP file** of size **0KB** with a different name each time and a configuration file named **Settings.ini** are created. This TMP file has the same name as the previously deleted TMP file.

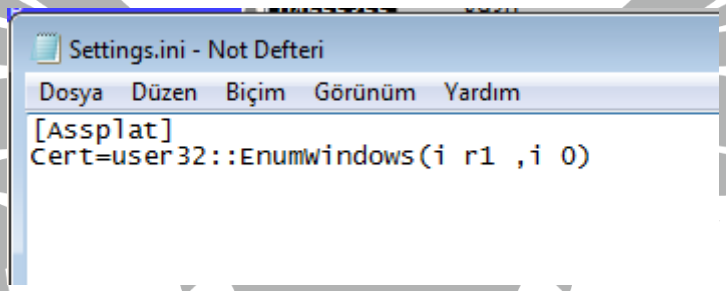

Figure 14 Contents of "Settings.ini" File and EnumWindows API

The contents of the configuration file are shown in Figure 14.

The DialogBoxParamW API is used to change the value of the **"HKCU\Software\SetupADA\Alloi now"** key in the registry during the operation of the setup window shown in Figure 10. The set values are shown in Table 4 respectively.

user32::ShowWindow(ir4,i0)
kernel32::CreateFileA(m r4 ,i 0x80000000,i 0, p 0, i 4, i 0x80, i 0)i.r5
kernel32::VirtualAlloc(i 0,i R2,i 12288,i 64)p.r1
kernel32::ReadFile(i r5, i r1,i R2,*i 0, i 0)i.r3
user32::EnumWindows(i r1 ,i 0)

Table 4 Values of the key "Alloi now" in the Registry

These values represent the functions used to install the system with **NSIS (Nullsoft Scriptable Install System)**. **ShowWindow** function hides or closes the window with the value "i0". The **CreateFileA** function creates a file with read permission (0x80000000; GENERIC_READ). The **VirtualAlloc** function allocates a 12,288-byte region of virtual memory. The **ReadFile** function reads this file and finally the **EnumWindows** function obtains the information of the top-level windows that are open.

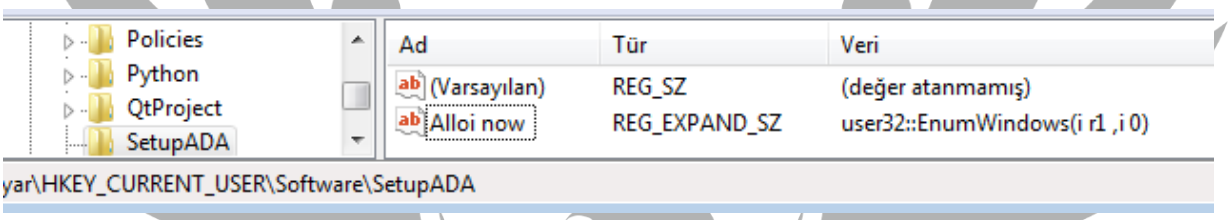

Figure 15 Final Status of the Value in the Registry

The EnumWindows value, which is the last value assigned in the registry, does not change.

Figure 16 "mellemmands" Key

It then attempts to access a key in the registry named **HKCU\Software\mellemmands**. There is no key named **mellemands** in the specified location in the registry. This word means **'intermediary'** in Danish.

64	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR
64	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR
64	RegQueryValue	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR\Disable
64	RegCloseKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\Windows Error Reporting\WMR

Figure 17 Disabled Key to Prevent Error Display

During these processes, the malware sets the **"HKLM\\\\SOFTWARE\\\Wow6432Node\\\Microsoft\\\Windows\\\Windows Error Reporting\\\WMR\\Disable"** key to 'Disable' to prevent any errors from appearing on the screen.

192.168.96.132	104.120.110.77	TCP	66	49224 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
104.120.110.77	192.168.96.132	TCP	60	443 → 49224 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
192.168.96.132	104.120.110.77	TCP	54	49224 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
192.168.96.132	104.120.110.77	TLSv1.2	253	Client Hello
104.120.110.77	192.168.96.132	TCP	60	443 → 49224 [ACK] Seq=1 Ack=200 Win=64240 Len=0
104.120.110.77	192.168.96.132	TLSv1.2	1514	Server Hello
104.120.110.77	192.168.96.132	TCP	1514	443 → 49224 [ACK] Seq=1461 Ack=200 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
104.120.110.77	192.168.96.132	TLSv1.2	1222	Certificate, Certificate Status, Server Key Exchange, Server Hello Done
192.168.96.132	104.120.110.77	TCP	54	49224 → 443 [ACK] Seq=200 Ack=4089 Win=64240 Len=0
192.168.96.132	104.120.110.77	TLSv1.2	236	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
104.120.110.77	192.168.96.132	TCP	60	443 → 49224 [ACK] Seq=4089 Ack=382 Win=64240 Len=0
104.120.110.77	192.168.96.132	TLSv1.2	161	Change Cipher Spec, Encrypted Handshake Message
192.168.96.132	104.120.110.77	TCP	54	49224 → 443 [ACK] Seq=382 Ack=4196 Win=64133 Len=0
192.168.96.132	104.120.110.77	TLSv1.2	267	Application Data
104.120.110.77	192.168.96.132	TCP	60	443 → 49224 [ACK] Seq=4196 Ack=595 Win=64240 Len=0
104.120.110.77	192.168.96.132	TCP	1514	443 → 49224 [ACK] Seq=4196 Ack=595 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
104.120.110.77	192.168.96.132	TCP	1514	443 → 49224 [ACK] Seq=5656 Ack=595 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
104.120.110.77	192.168.96.132	TCP	1514	443 → 49224 [ACK] Seq=7116 Ack=595 Win=64240 Len=1460 [TCP segment of a reassembled PDU]

Figure 18 Data Analysed on Wireshark

The malware establishes an encrypted connection with the command-and-control server by sending a request to socket **104[.]120[.]110[.]77[:]443**. It sends some packets to this IP address and creates the original PDU by combining TCP segments.

DNS	88	Standard query 0x20cf A javadl-esd-secure.oracle.com
DNS	195	Standard query response 0x20cf A javadl-esd-secure.oracle.com CNAME javadl-esd-secure.oracle.com.edgekey.net CNAME e13073.g.akamaiedge.net A 104.120.110.77
TCP	66	49226 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
TCP	60	443 → 49226 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

Figure 19 Complex DNS Resolution

The malware performs complex DNS resolutions. It uses the Akamai content distribution network to bypass security measures and hide malicious content.

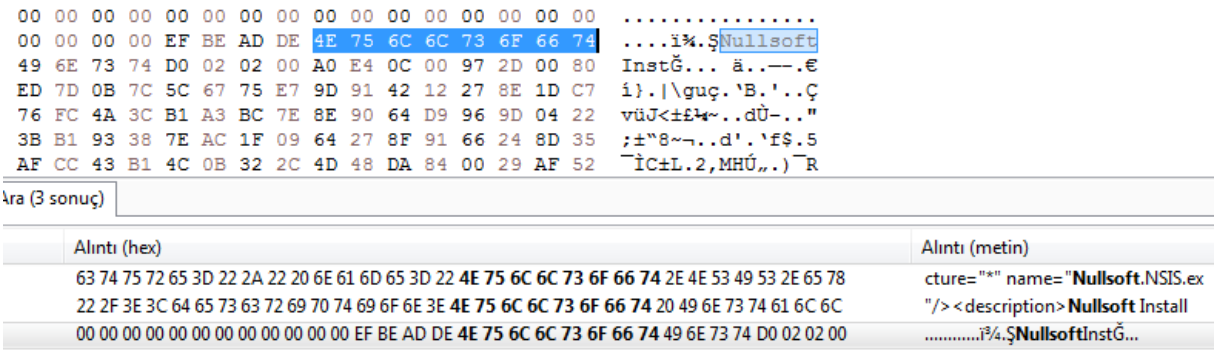

Figure 20 Using the NSIS-based Structure

The malware contains a structure based on **NSIS (Nullsoft Scriptable Install System)**. It performs the resolves and distribution of the malware using **NSIS Installer**. NSIS includes a series of scripts and commands to allow users to customise their installation files. Thus, it can perform various operations during installation.

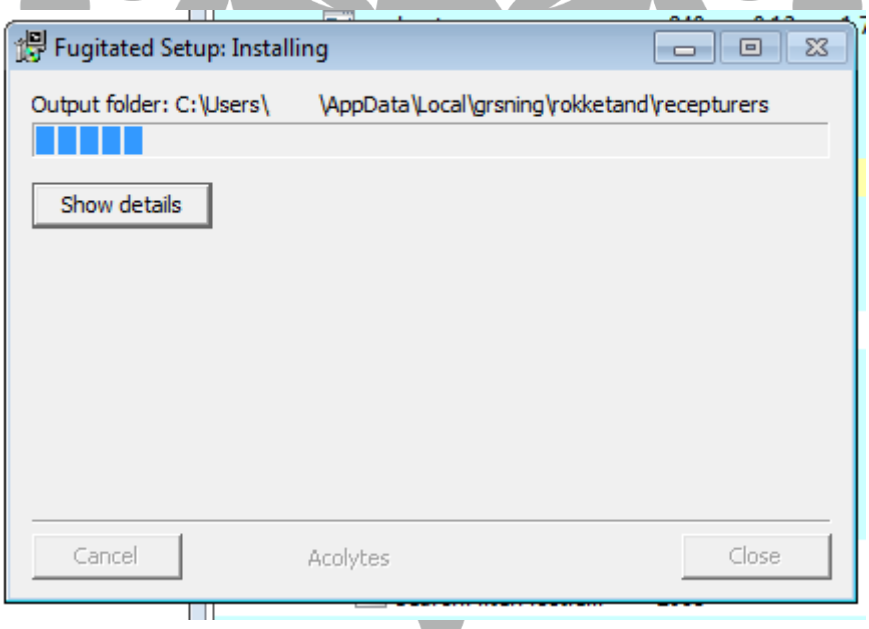

Figure 21 Installation File Named "Fugitated Setup"

As previously stated, the installation file containing the malicious code opens a window titled **"Fugitated Setup"** and proceeds to execute the code, resulting in the theft of targeted data.

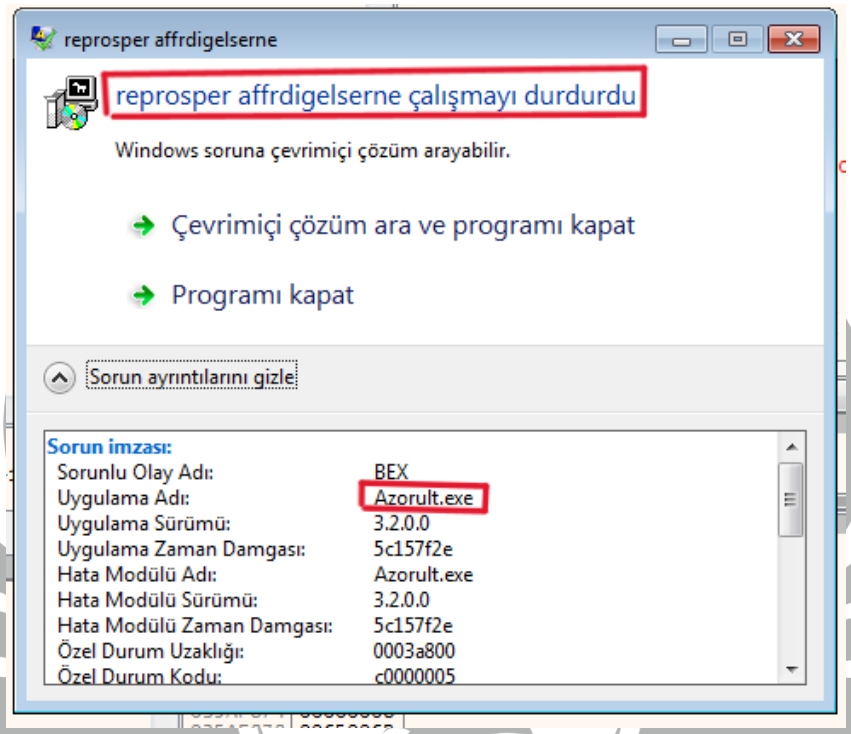

Figure 22 "reprosper affrdigelserne has stopped running" Error Screen

The malware terminates its own process by giving an error The malware terminates its own process by giving an error **"reprosper affrdigelserne has stopped running"**.

YARA Rules

```
import “hash”

rule Azorult {

meta:

        description = "Azorult"

        aouthor = “zayotem”

strings:

        $api1 = "DialogBoxParamW" ascii

        $api2 = "DeleteFileW" ascii

        $api3 = "GetTempFileNameW" ascii

        $api4 = "ExitProcess" ascii

        $api5 = "SetWindowTextW" ascii


        $str1 = "\\Microsoft\\Internet Explorer\\Quick Launch" wide

        $str2 = "Software\\Microsoft\\Windows\\CurrentVersion" wide

        $str3 = "Control Panel\\Desktop\\ResourceLocale" wide
```

```
        $str4 = "http://nsis.sf.net/NSIS_Error" wide

        $str5 = "reprosper affrdigelserne" wide

        $str6 = "kondensvandet capetonian" wide


        $hex_1 = {74 1B 8B F8 8B 06 85 C0 74 0A 50 8D 46 18 50 E8}

        $hex_2 = {4E 75 6C 6C 73 6F 66 74}


condition:

        hash.md5 (0, filesize) == "64CE3428700D7A0797CC4D779AC37C39" or (4 of
($api*)) or (5 of ($str*)) or (1 of ($hex*))

}
```

MITRE ATTACK TABLE

Reconnaissance	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	C&C	Exfliration
Gather Victim Host Information (T1592)	Command and Scripting Interpreter (T1059)	Account Manipulation (T1098)	Create Process with Token (T1134.002)	Access Token Manipulation	Credentials from Web Browsers (T1555.003)	File Transfer Protocols (T1071.002)	Exfliration Over C2 Channel (T1041)
Gather Victim Network Information (T1590)	Native API (T1106)	Valid Accounts (T1078)	Process Injection (T1055)	Deobfuscate/ Decode Files or Information (T1140)	Steal Web Session Cookie (T1539)	Encrypted Channel (T1573)	
				Process Injection (T1055)	Input Capture (T1056)		
				Input Capture (T1056)			

Recommendations

1. Use up-to-date antivirus protection,
2. Do not download files from unknown sources,
3. Untrusted e-mails should not be opened, attachments should not be downloaded,
4. Be technology literate,
5. The operating system must be kept up to date,
6. Do not click on links of unknown origin

PREPARED BY

Betül ŞAHİN https://www.linkedin.com/in/betulsahinn/