

# Vidar

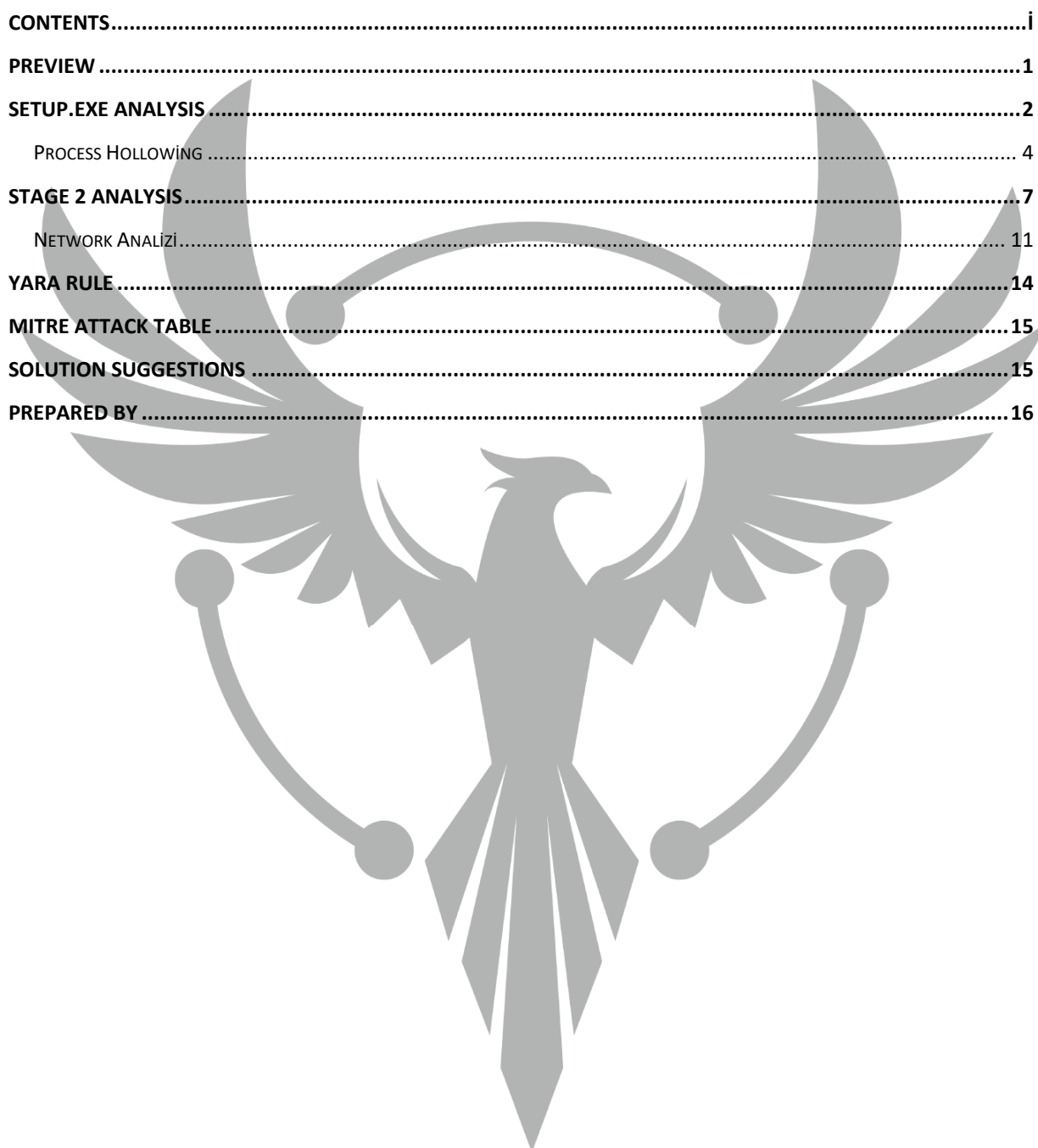
## TECHNICAL ANALYSIS REPORT

**ZAYOTEM**

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

# Contents

CONTENTS.....	i
PREVIEW .....	1
SETUP.EXE ANALYSIS.....	2
PROCESS FOLLOWING .....	4
STAGE 2 ANALYSIS.....	7
NETWORK ANALIZI.....	11
YARA RULE.....	14
MITRE ATTACK TABLE.....	15
SOLUTION SUGGESTIONS .....	15
PREPARED BY .....	16



## Preview

The Vidar malware was first discovered by security experts in 2018. This malware is designed to commit financial information theft and, like other similar malware, works to steal information by infecting users' computers.

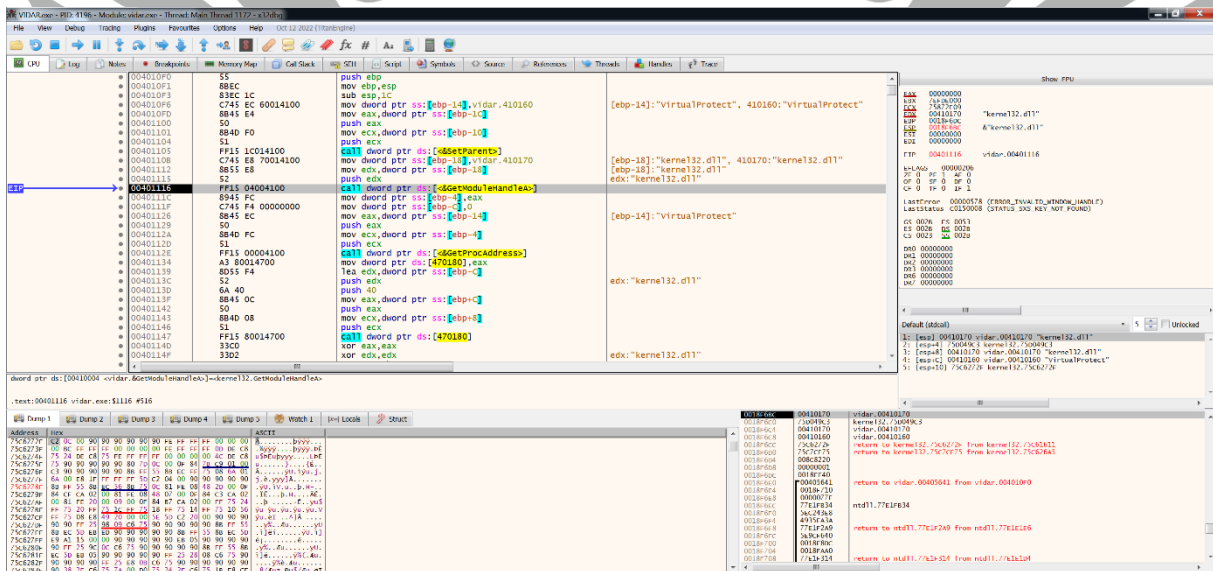
Vidar is a software that specifically targets users with financial goals and aims to steal payment information as well as important information such as bank account information, money transfers and other financial transactions. For this, crypto wallets and internet browser try to collect all the personal information on the targeted computer by recording its history.

It may use methods such as spam emails, fake software updates, malicious websites, and online advertisements as distribution methods. It is known that there are different versions of Vidar and each version may show different features.

Since its discovery, the Vidar malware, along with its different versions, has infected many computers and has caused harm to many users by stealing financial information.

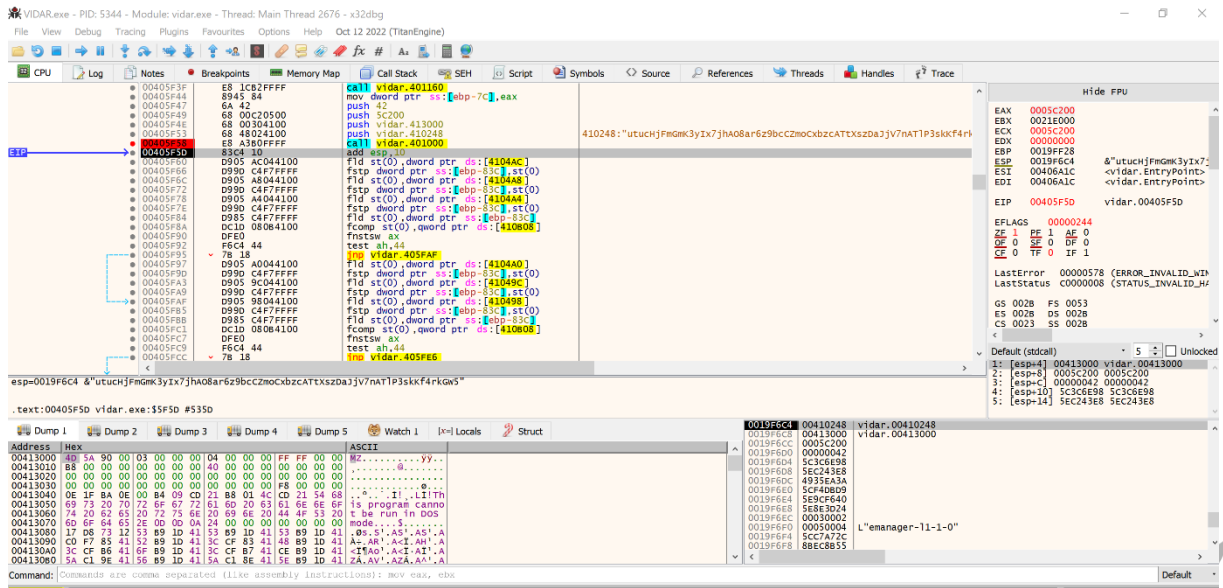
## Setup.exe Analysis

Name	Setup.exe
MD5	dcd26511183f2d7eb30678661a88b765
SHA256	8f0d2909498e32a88ea7a3873958edd5456e0d9d3e766ce7c8bcc303f67d8984
File Type	PE32 / EXE



### Şekil 1- API Resolution

As a result of API resolve with the `GetModuleHandle` and `GetProcAddress` APIs, the malicious has enabled executive, read-only, or read/write access to the specified virtual memory space by using the `VirtualProtect` API.



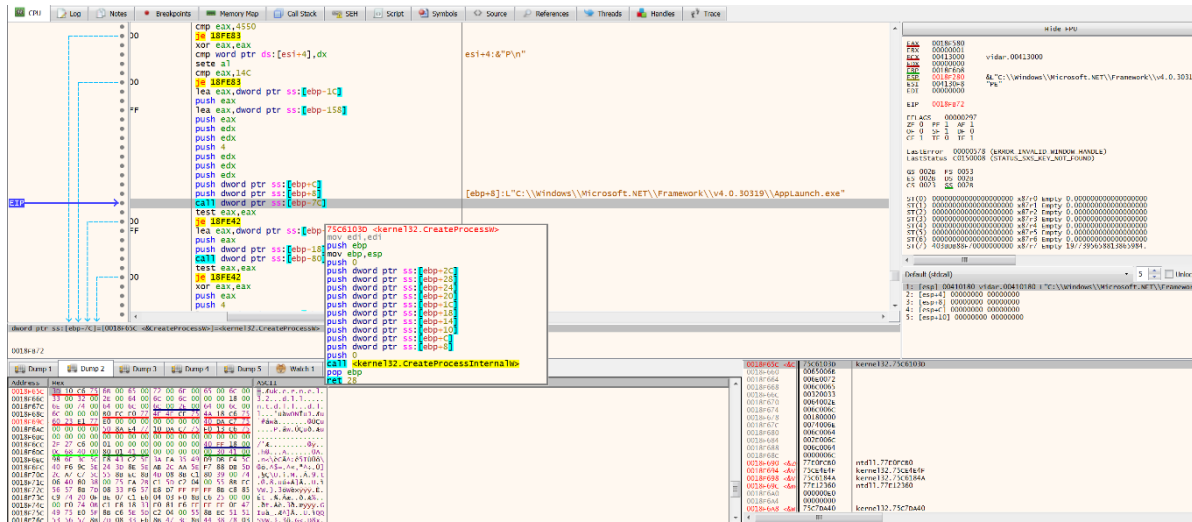
Şekil 2- Resolved file

The file titled “MZ”, which the malware analyzes at runtime, was found.

```
1 unsigned int __cdecl sub_401000(int a1, int a2, unsigned int a3)
2 {
3     unsigned int result; // eax
4     char v4; // [esp+7h] [ebp-5h]
5     unsigned int i; // [esp+8h] [ebp-4h]
6
7     for ( i = 0; i < a3; ++i )
8     {
9         SetActiveWindow(hWnd);
10        v4 = (36 * *(_BYTE *)(a1 + (int)i % 60)) & 0x70 ^ *(_BYTE *)(i + a2);
11        *(_BYTE *)(i + a2) = 2 * v4;
12        *(_BYTE *)(i + a2) -= v4;
13        result = i + 1;
14    }
15    return result;
16 }
```

Şekil 3- Analysis algorithm

## Process Hollowing



Şekil 4- Seen that a process is started with the CreateProcess API.

The malware creates a "suspended" process using the **CreateProcess** API. The full path for this action is

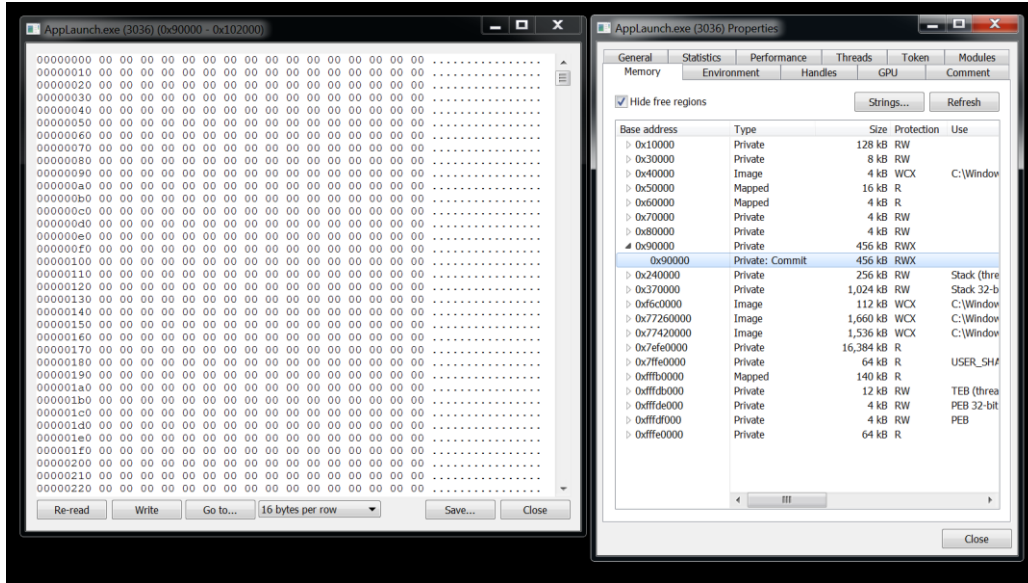
"C:\\Windows\\Microsoft.NET\\Framework\\v4.0.3019\\AppLaunch.exe"

explorer.exe	2756	0.01		87.65 MB	ice
vmtoolsd.exe	2848	0.09	1.2 kB/s	29.36 MB	ice
chrome.exe	2540	0.14	1.14 kB/s	123.64 MB	ice
Everything.exe	4024			15.85 MB	ice
x32dbg.exe	2984	0.35	36 B/s	54.69 MB	ice
VIDAR.exe	3296	0.01		1.04 MB	ice
Applaunch.exe	3000			408 kB	ice
ProcessHacker.exe	2224	0.61		15.07 MB	ice

Şekil 5- AppLaunch.exe

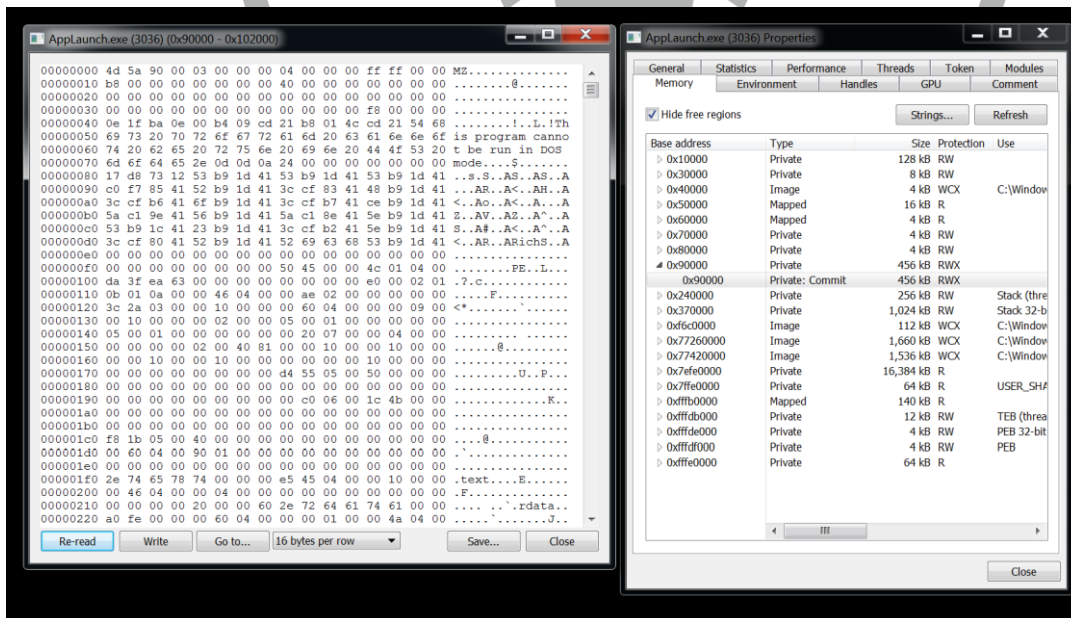
It is seen that the malware allocates memory space in the process it created in the "suspend" state using the "VirtualAllocEx" API.





Şekil 6- Ayrılan bellek alanı

It is seen that it writes the executable file it parses to this memory area using the "WriteProcessMemory" API.



Şekil 7- WriteProcessMemory API'si sonrası bellek alanı





## Stage 2 Analysis

Name	-
MD5	c404e69187afab5fd694570220660576
SHA256	279fff770c6678a1839799bd83aa9ace0c78380b9f93bd4b4a689c245382b4e6
File Type	PE32 / EXE

```

012E47A8 26C3 sub eax,ebx
012E47AD 8985 ECFDFFFF mov dword ptr ss:[ebp-214],eax
012E47B3 8D85 F4FDFFFF lea eax,dword ptr ss:[ebp-20C]
012E47B9 50 push eax
012E47BA E8 6CC10200 call vidar_00413000.131092B
012E47BF 8D8D F4FDFFFF lea ecx,dword ptr ss:[ebp-20C]
012E47C5 51 push ecx
012E47C6 E8 60C10200 call vidar_00413000.131092B
012E47CB 8B95 F0FDFFFF mov edi,dword ptr esi:[ebp-210]
012E47D1 52 push edi
012E47D2 8D3C1E lea edi,dword ptr ds:[esi+ebx]
012E47D5 E8 C6C00200 call vidar_00413000.strlenida
012E47DA 8BC8 mov ecx,eax
012E47DC 33D2 xor ecx,edx
012E47DE 8BC6 mov eax,esi
012E47E0 F7FL div ecx
012E47E2 8B85 F0FDFFFF mov eax,dword ptr ss:[ebp-210]
012E47E8 8A0C02 mov cl,byte ptr ds:[edx+eax]
012E47EB 8B95 ECFDFFFF mov edx,dword ptr ss:[ebp-214]
012E47F1 320C3A xor cl,byte ptr ds:[edx+edi]
012E47F4 8D85 F4FDFFFF lea eax,dword ptr ss:[ebp-20C]
012E47FA 50 push eax
012E47FB 880F mov byte ptr ds:[edi],cl
012E47FD E8 29C10200 call vidar_00413000.131092B
012E4802 8D8D F4FDFFFF lea ecx,dword ptr ss:[ebp-20C]
012E4808 51 push ecx
012E4809 E8 1DC10200 call vidar_00413000.131092B
012E480E 46 inc esi
012E480F 83C4 14 add esp,14
012E4812 3B85 E8FDFFFF cmp esi,dword ptr ss:[ebp-218]
012E4816 72 99 jle vidar_00413000.12E47B3

```

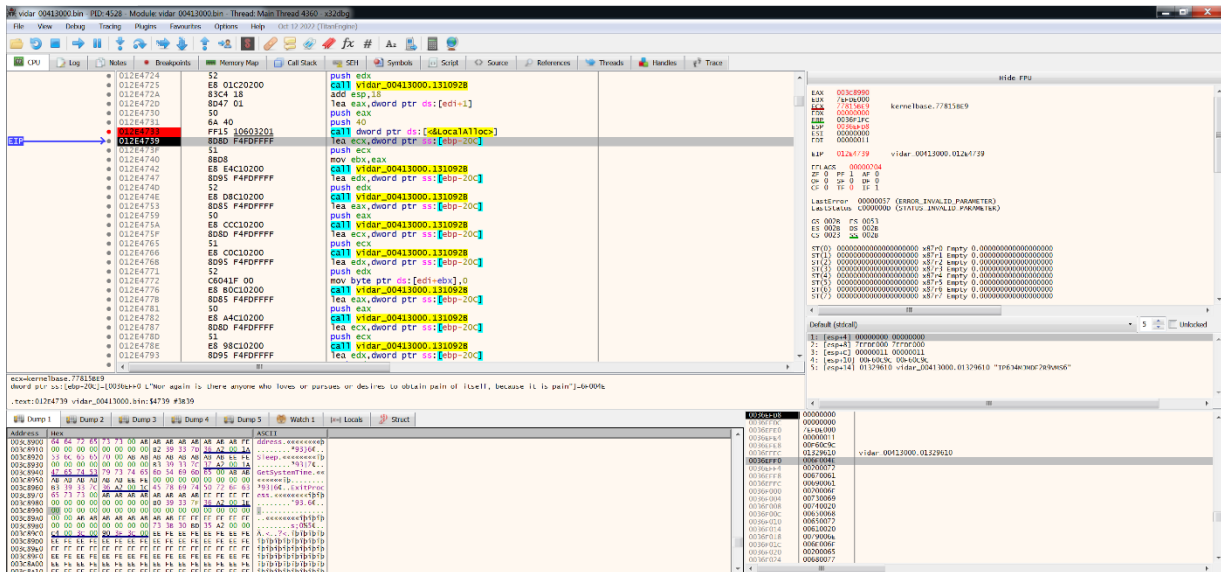
Decoded string: "Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain"

Şekil 9- Decoding of ciphertxts

It uses the string **"Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain"** and a unique key that it uses to decode each encrypted string.

Strings that the malware will use are decoded using this method.

Examples are given in Table-1 and Table-2.



Şekil 10- Using the LocalAlloc API

It can be seen that decoded string expressions are written to the allocated memory area using "LocalAlloc".

<pre> cmp dword ptr ds:[ebx+4],40 jb vidar_00413000.12E4F8 mov eax,dword ptr ds:[ebx] push 40 push eax call vidar_00413000.130EE0 mov ecx,5A4D </pre>	<pre> 40:'g' eax:"\\Microsoft\\Edge\\User Data\\" eax:"\\Microsoft\\Edge\\User Data\\" </pre>
<pre> cmp dword ptr ds:[ebx+4],40 jb vidar_00413000.12E4F8 mov eax,dword ptr ds:[ebx] push 40 push eax push esi call vidar_00413000.130EE0 mov ecx,5A4D </pre>	<pre> 40:'g' eax:"\\Opera Software\\Opera Stable\\" eax:"\\Opera Software\\Opera Stable\\" </pre>
<pre> cmp dword ptr ds:[ebx+4],40 jb vidar_00413000.12E4F8 mov eax,dword ptr ds:[ebx] push 40 push eax push esi call vidar_00413000.130EE0 mov ecx,5A4D </pre>	<pre> 40:'g' eax:"\\AppData\\Roaming\\FileZilla\\recentServers.xml" eax:"\\AppData\\Roaming\\FileZilla\\recentServers.xml" </pre>
<pre> cmp dword ptr ds:[ebx+4],40 jb vidar_00413000.12E4F8 mov eax,dword ptr ds:[ebx] push 40 push eax push esi call vidar_00413000.130EE0 mov ecx,5A4D </pre>	<pre> 40:'g' eax:"\\Soft\\Discord\\discord_tokens.txt" eax:"\\Soft\\Discord\\discord_tokens.txt" </pre>
<pre> cmp dword ptr ds:[ebx+4],40 jb vidar_00413000.12E4F8 mov eax,dword ptr ds:[ebx] push 40 push eax push esi call vidar_00413000.130EE0 mov ecx,5A4D </pre>	<pre> 40:'g' eax:"\\BraveSoftware\\Brave-Browser\\User Data\\" eax:"\\BraveSoftware\\Brave-Browser\\User Data\\" </pre>

Şekil 11- Some browser directories used to collect information

It scans the directory to obtain malicious sensitive data. The directories it scanned are given in Table-1.

MicrosoftEdge\\Cookies	\\AppData\\Roaming\\FileZilla\\recentservers.xml
\\Mozilla\\Firefox\\Profiles\\	\\Moonchild Productions\\Pale Moon\\Profiles\\
\\Google\\Chrome\\User Data\\	\\Chromium\\User Data\\
\\Amigo\\User Data\\	\\Torch\\User Data\\
\\Comodo\\Dragon\\User Data\\	\\Epic Privacy Browser\\User Data\\
\\Vivaldi\\User Data\\	\\CocCoc\\Browser\\User Data\\
\\CentBrowser\\User Data\\	\\TorBro\\Profile\\
\\Chedot\\User Data\\	\\7Star\\7Star\\User Data\\
\\Microsoft\\Edge\\User Data\\	\\360Browser\\Browser\\User Data\\
\\Tencent\\QQBrowser\\User Data\\	\\Opera Software\\Opera Stable\\
\\Opera Software\\Opera GX Stable\\	

Tablo 1-Browser directories

```

cmp dword ptr ds:[ebx+4],40
jb vidar_00413000.12E48F8
mov eax,dword ptr ds:[ebx]
push 40
push eax
push esi
call vidar_00413000.130EEE0
mov ecx,5A4D
40:'@'
eax:"BinanceChainWallet"
eax:"BinanceChainWallet"

cmp dword ptr ds:[ebx+4],40
jb vidar_00413000.12E48F8
mov eax,dword ptr ds:[ebx]
push 40
push eax
push esi
call vidar_00413000.130EEE0
mov ecx,5A4D
40:'@'
eax:"Coinbase"
eax:"Coinbase"

cmp dword ptr ds:[ebx+4],40
jb vidar_00413000.12E48F8
mov eax,dword ptr ds:[ebx]
push 40
push eax
push esi
call vidar_00413000.130EEE0
mov ecx,5A4D
40:'@'
eax:"MathWallet"
eax:"MathWallet"

```

Şekil 12- Some wallet names used to collect information

EQUALWallet	BitAppWallet	iWallet
Wombat	MewCx	GuildWallet
RoninWallet	NeoLine	CloverWallet
LiquidityWallet	Terra_Station	Keplr
AuroWallet	PolymeshWallet	ICONex
KardiaChain	EVER Wallet	Rabby
Harmony	Coin98	Ledger Live
Bitwarden	Leap Terra	Martian Wallet
Petra Wallet	Pontem Wallet	Gero Wallet
Eternl	Hashpack	OKX Web3 Wallet
Exodus Web3 Wallet	Trust Wallet	Tronium
Braavos	Enkrypt	Finnie

Tablo 2-Crypto Wallets

The malware has been observed to target "password manager" applications to obtain sensitive data. These are given in Table-3.

KeePass Tusk	Trezor Password Manager
KeePassXC-Browser	Microsoft AutoFill

Tablo 3-Password Managers

It has been observed that the malware collects system information.

Şekil 13- Getting MachineGuid information

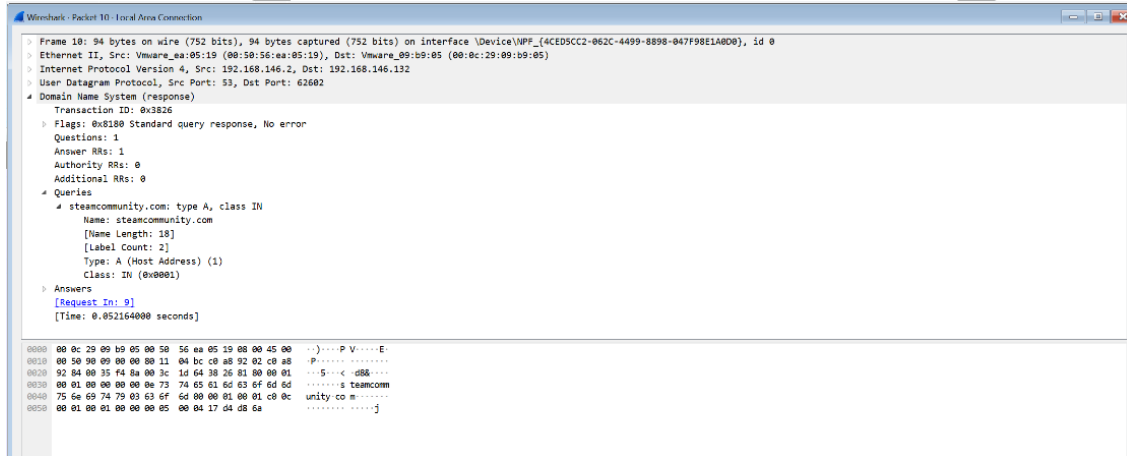
Şekil 14-Using the GetSystemInfo API

The malware obtains system information such as processor architecture, processor type, number of processors with the "GetSystemInfo" API.

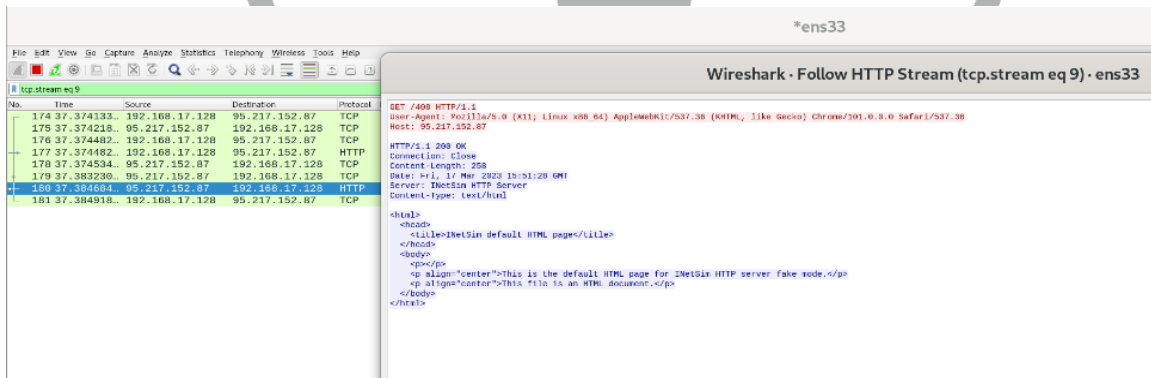
Şekil 15-Using the GetCurrentHwProfileA API

Using the "**GetCurrentHwProfileA**" API, information about the hardware profile of the local computer is collected.

## Network Analizi



Şekil 16-DNS request



Şekil 17- HTTP GET request

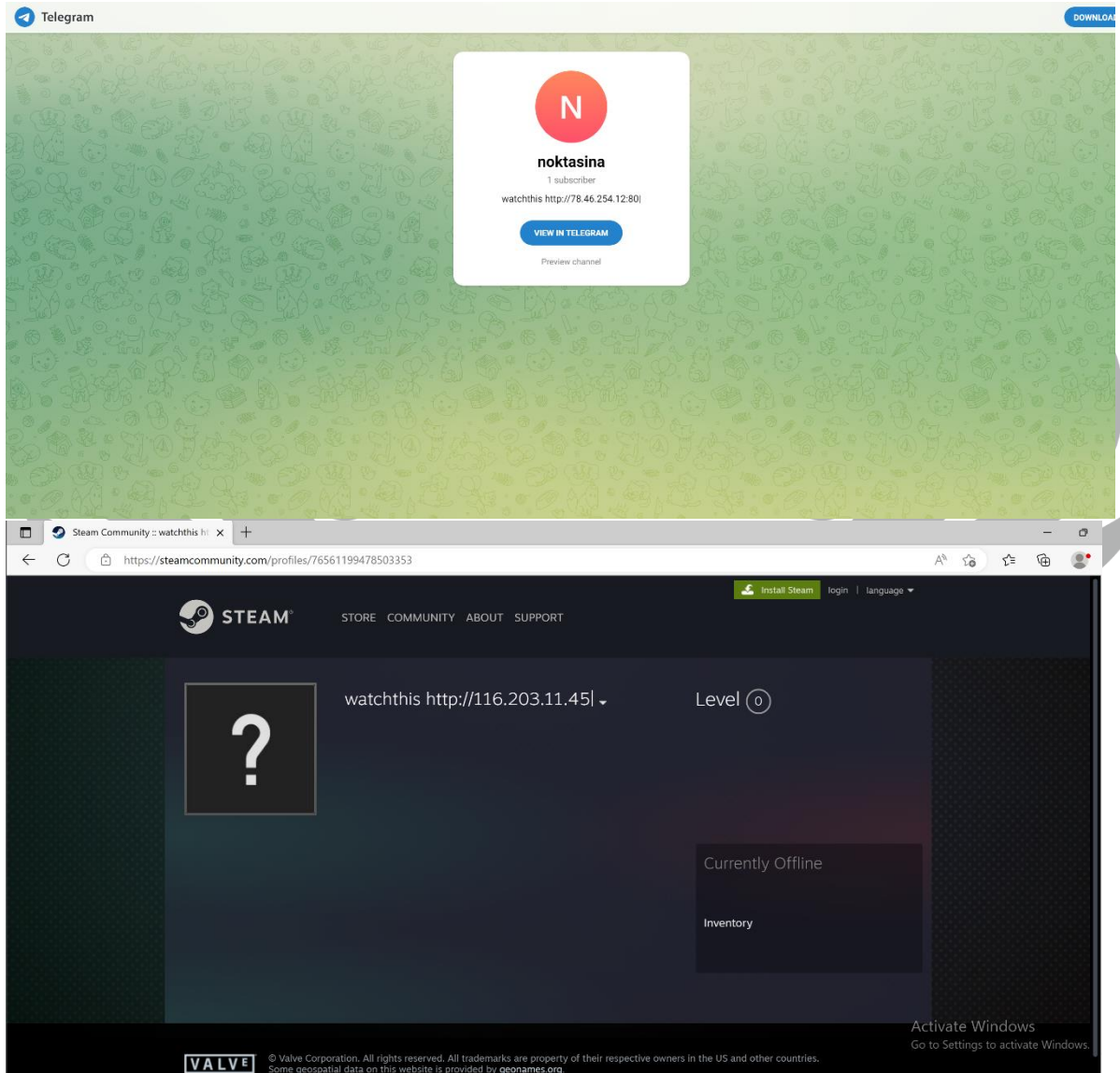
<pre> push 7CF lea ecx,dword ptr ss:[ebp-8E0] push ecx push esi call dword ptr ds:[&amp;InternetReadFile] test eax,eax je vidar_00413000.12F2655 nop mov eax,dword ptr ss:[ebp-9F0] test eax,eax je vidar_00413000.12F2655 lea edx,dword ptr ss:[ebp-9A4] mov byte ptr ss:[ebp+eax-8E0],0 push edx lea eax,dword ptr ss:[ebp-8FC] lea ebx,dword ptr ss:[ebp-8E0] call &lt;vidar_00413000.mayconcat&gt; add esp,8 mov edi,eax lea esi,dword ptr ss:[ebp-9A4] mov hvre ntr &lt;&lt;ehh-19 </pre>	<pre> eax:&amp;"&lt;html&gt;\n &lt;head&gt;\n &lt;title&gt;Inetsim default HTML page&lt;/title&gt;\n &lt;/head&gt;\n &lt;body&gt;\n &lt;p&gt;&lt;/p&gt;\n eax:&amp;"&lt;html&gt;\n &lt;head&gt;\n &lt;title&gt;Inetsim default HTML page&lt;/title&gt;\n &lt;/head&gt;\n &lt;body&gt;\n &lt;p&gt;&lt;/p&gt;\n [ebp-8FC]:"&lt;html&gt;\n &lt;head&gt;\n &lt;title&gt;Inetsim default HTML page&lt;/title&gt;\n &lt;/head&gt;\n &lt;body&gt;\n &lt;p&gt; eax:&amp;"&lt;html&gt;\n &lt;head&gt;\n &lt;title&gt;Inetsim default HTML page&lt;/title&gt;\n &lt;/head&gt;\n &lt;body&gt;\n &lt;p&gt;&lt;/p&gt;\n eax:&amp;"&lt;html&gt;\n &lt;head&gt;\n &lt;title&gt;Inetsim default HTML page&lt;/title&gt;\n &lt;/head&gt;\n &lt;body&gt;\n &lt;p&gt;&lt;/p&gt;\n </pre>
--	---

Şekil 18-Using the InternetReadFile API

The malicious reads the contents of the returned request using the **"InternetReadFile"** API.

The request sent fails because the C2 servers are down.





Şekil 19-C2 servers

http://116[.]203[.]11[.]45/408	https://steamcommunity.com/profiles/76561199478503353
http://95[.]217[.]152[.]87:80	https://t.me/noktasina
http://95[.]217[.]152[.]87:80/epson.zip	

Tablo 3-URLs

# YARA Rule

```
import "hash"

rule vidar_rule {

  meta:

    description = "This is a YARA rule"

    author = "Dilara Behar"

  strings:

    $watchthis = "watchthis"

    $epson_zip = "epson.zip"

    $caf_racer = "A caf\\? racer is a genre of sport motorcycles that
    originated among British motorcycle enthusiasts of the early 1960s in
    London"

    $user_agent = "Mozilla\\5\\.0 \\(X11\\; Linux x86\\_64\\)
    AppleWebKit\\537\\.36 \\(KHTML\\, like Gecko\\) Chrome\\101\\.0\\.0\\.0
    Safari\\537\\.36"

    $st="https:\\\\steamcommunity\\.com\\profiles\\76561199478503353"

    $update_zip="update.zip"

  condition:

    hash.md5(0, filesize) == "dcd26511183f2d7eb30678661a88b765" or
    any of them
}
```

## MITRE ATTACK TABLE

Reconnaissance	Execution	Discovery	Privilege Escalation	Defense Evasion	Credential Access	C&C	Collection
	T1106-Native API	T1083-File and Directory Discovery	T1055-Process Hollowing	T1055-Process Hollowing		T1573 - Encrypted Channel	T1005- Data from Local System
		T1087-Account Discovery				T1071-Application Layer Protocol	
		T1082-System Information Discovery					

### Solution Suggestions

1. Using antivirus software is one of the most effective methods for detecting and removing malware. Antivirus software can detect malware by scanning files and websites that you download or open on your computer.
2. By regularly updating your operating system and other software, you can ensure the security of your computer. Updates help close various security gaps.
3. When downloading files, be careful to download from trusted sources. Files downloaded from unknown or suspicious sources may contain malware.

## **PREPARED BY**

Dilara BEHAR

<https://www.linkedin.com/in/dilara-behar-0530b3195>

