# Vidar

TECHNICAL ANALYSIS REPORT

ZAYOTEM
ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

# Contents

**CONTENTS** ..... **i**

**PREVIEW** ..... **1**

**SETUP.EXE ANALYSIS** ..... **2**

PROCESS HOLLOWİNG ..... 4

**STAGE 2 ANALYSIS** ..... **7**

NETWORK ANALİZİ ..... 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.

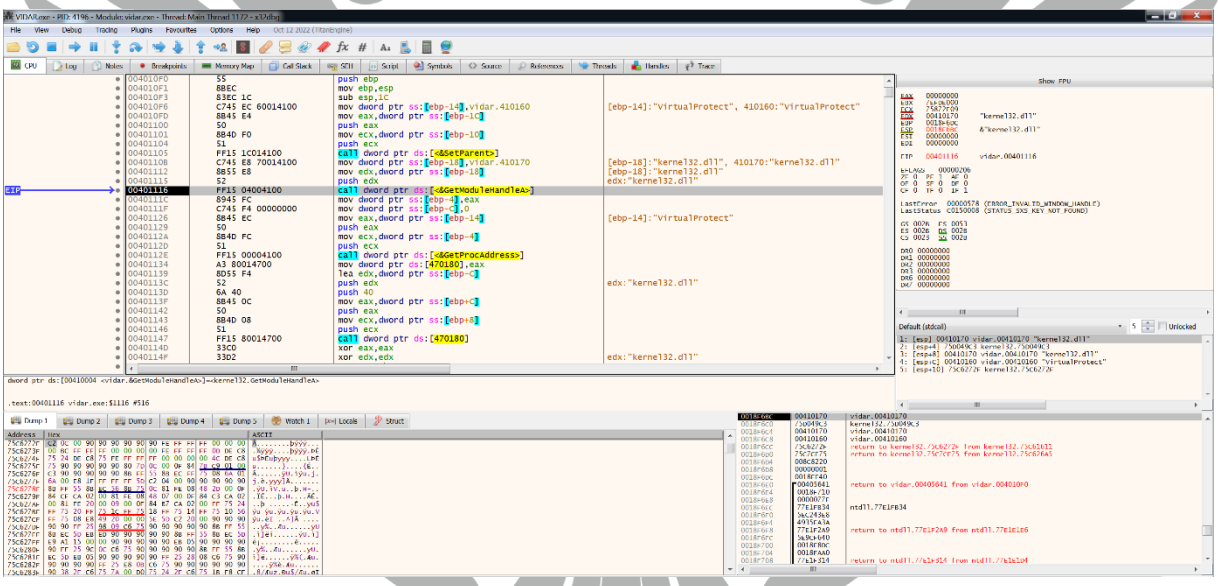
Vidar is a software that specifically targets users with financial goals and aims to steal payment information as well as important information such as bank account information, money transfers and other financial transactions. For this, crypto wallets and internet browser try to collect all the personal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

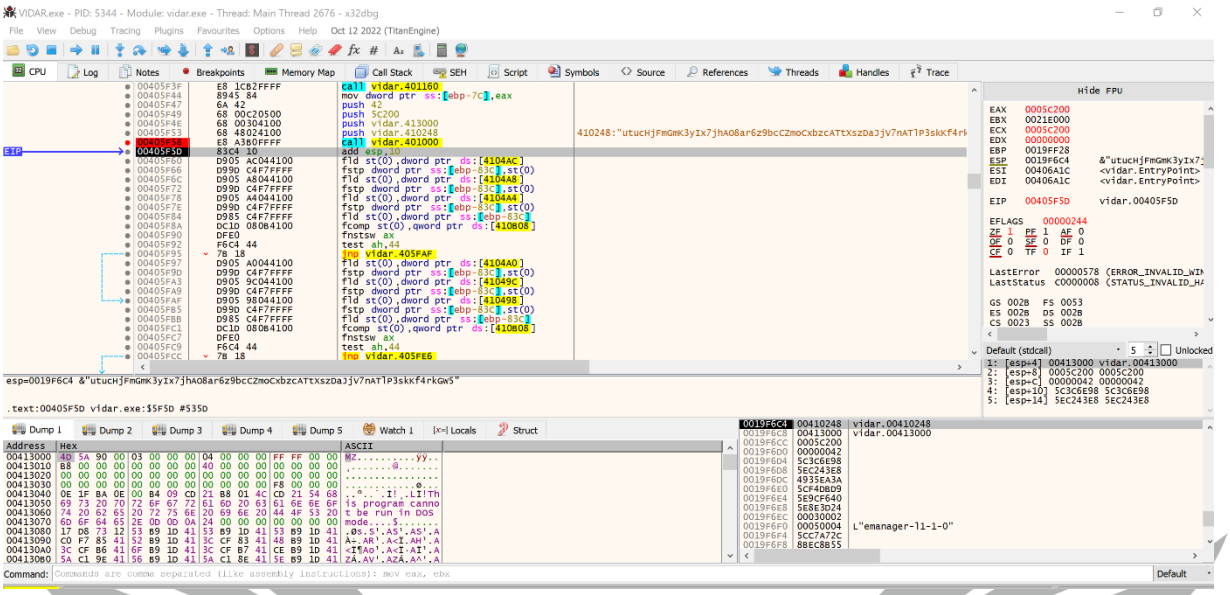
# Setup.exe Analysis

| 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*

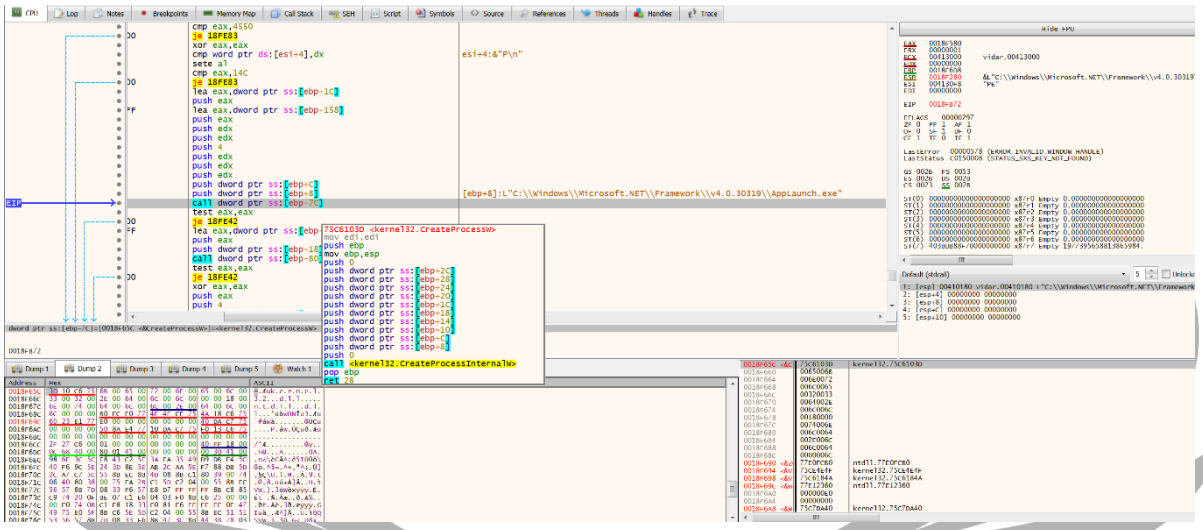
The file titled **“MZ”,** which the malware analyzes at runtime, was found.

```
unsigned int __cdecl sub_401000(int a1, int a2, unsigned int a3)
{
  unsigned int result; // eax
  char v4; // [esp+7h] [ebp-5h]
  unsigned int i; // [esp+8h] [ebp-4h]

  for ( i = 0; i < a3; ++i )
  {
    SetActiveWindow(hWnd);
    v4 = (36 * *(_BYTE *)(a1 + (int)i % 60)) & 0x70 ^ *(_BYTE *)(i + a2);
    *(_BYTE *)(i + a2) = 2 * v4;
    *(_BYTE *)(i + a2) -= v4;
    result = i + 1;
  }
  return result;
}
```

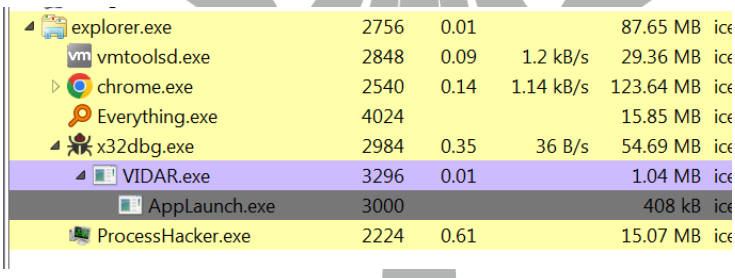
*Şekil 3- Analysis algorithm*

## Process Hollowing



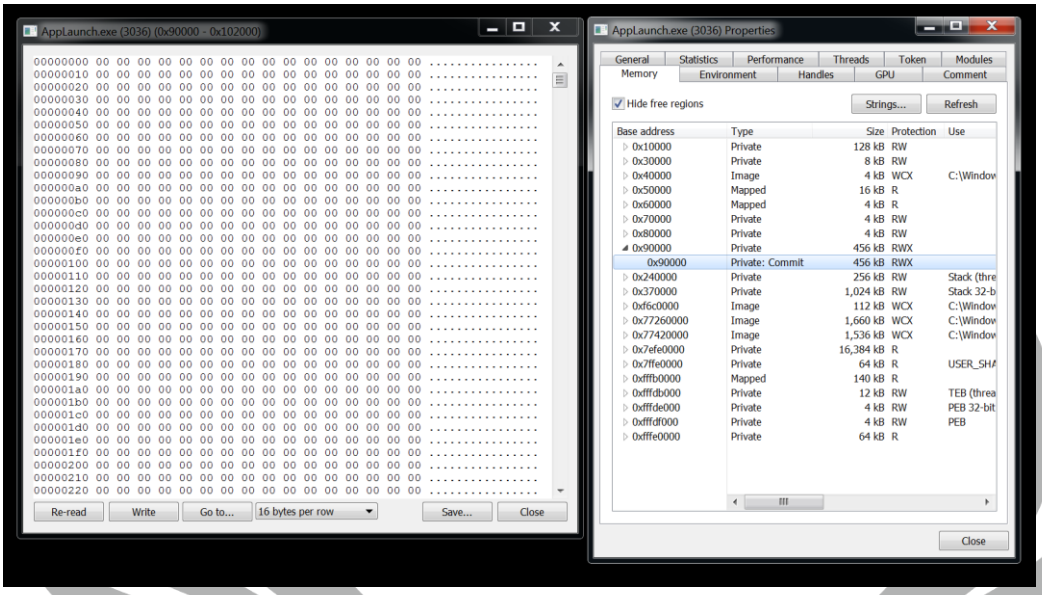
*Şekil 4- Seen that a process is started with the CreateProcess API.*

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"**



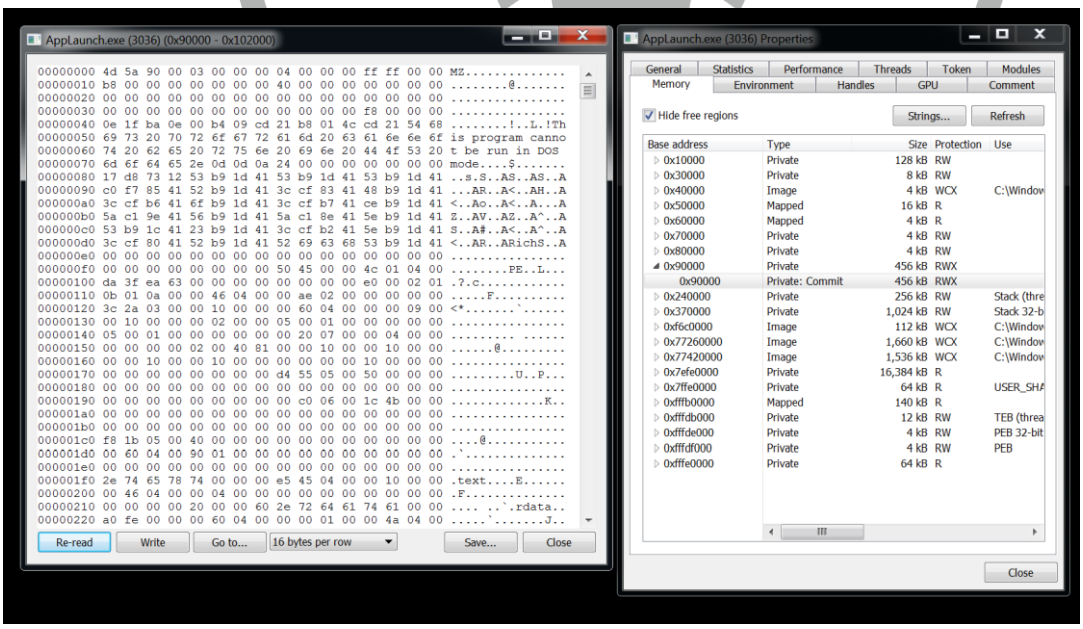

*Şekil 5- AppLaunch.exe*

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

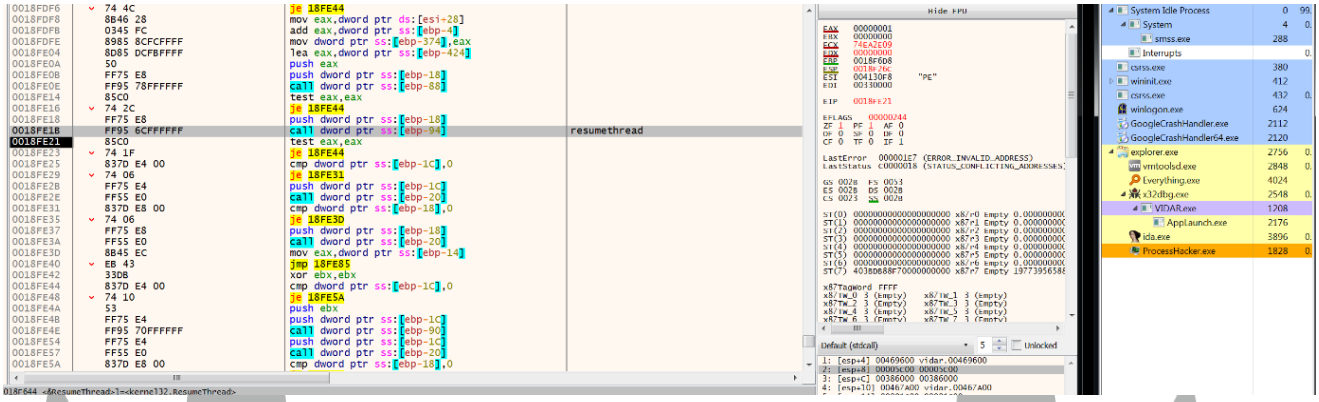



*Şekil 6- Ayrılan bellek alanı*

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*

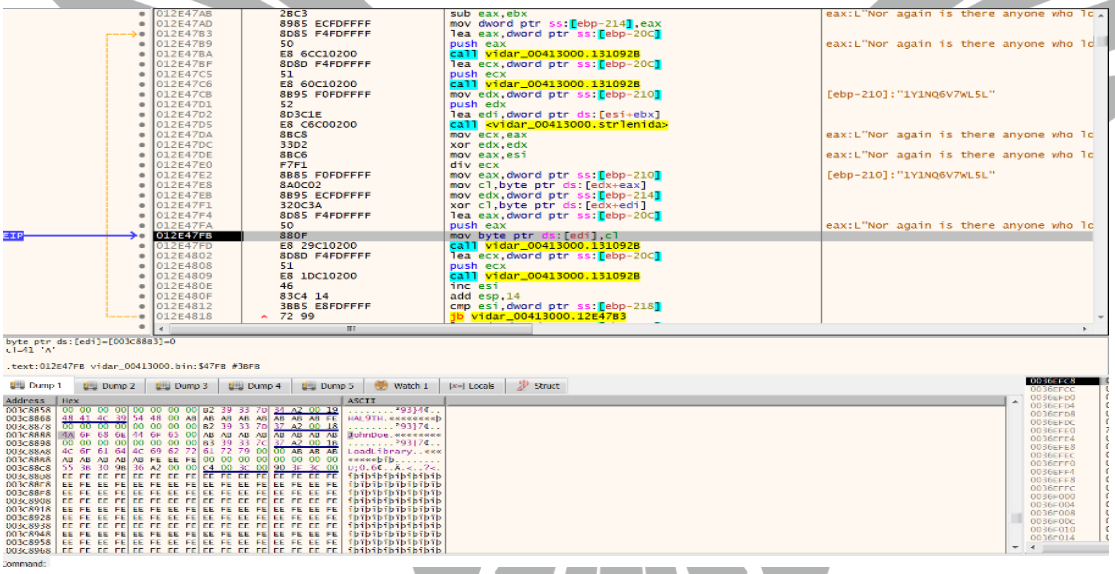



*Şekil 8- ResumeThread API'sinin kullanılması*

After the writing process is finished, the suspended process becomes active with the "ResumeThread" API and starts working.

# Stage 2 Analysis

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

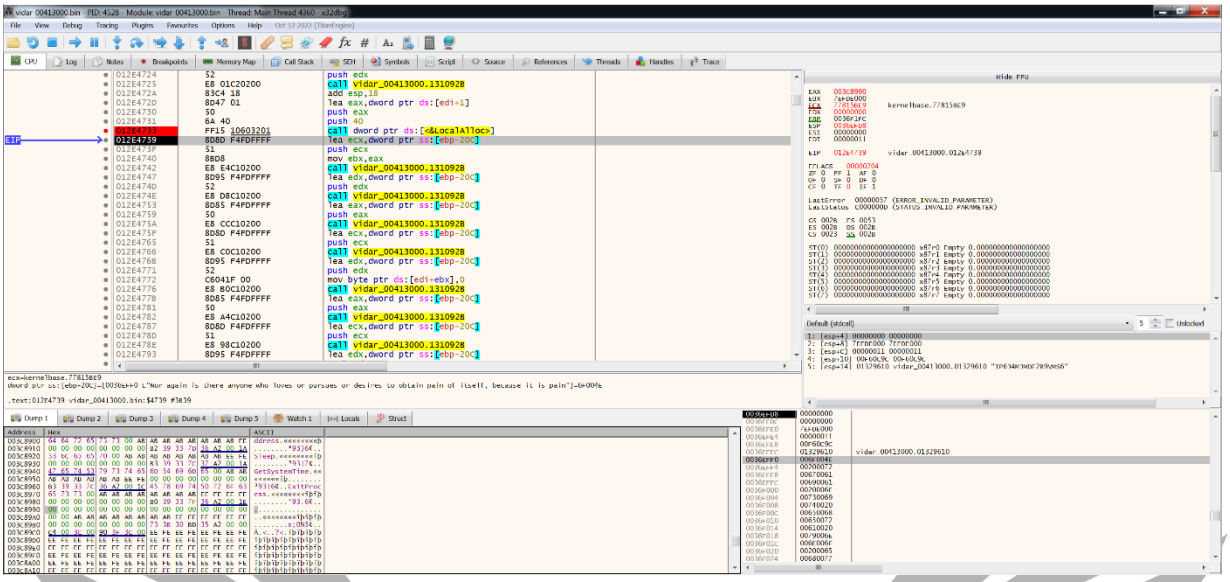


*Şekil 9- Decoding of ciphertexts*

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.

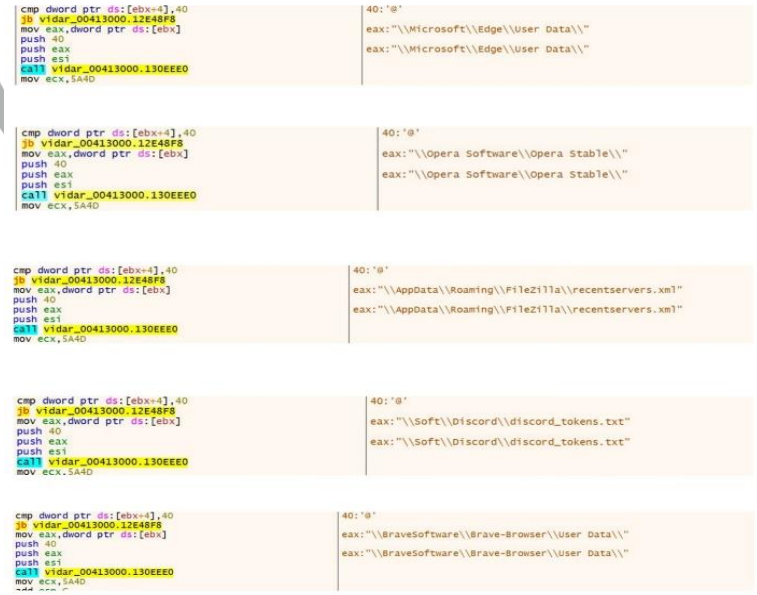
Strings that the malware will use are decoded using this method.

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".**

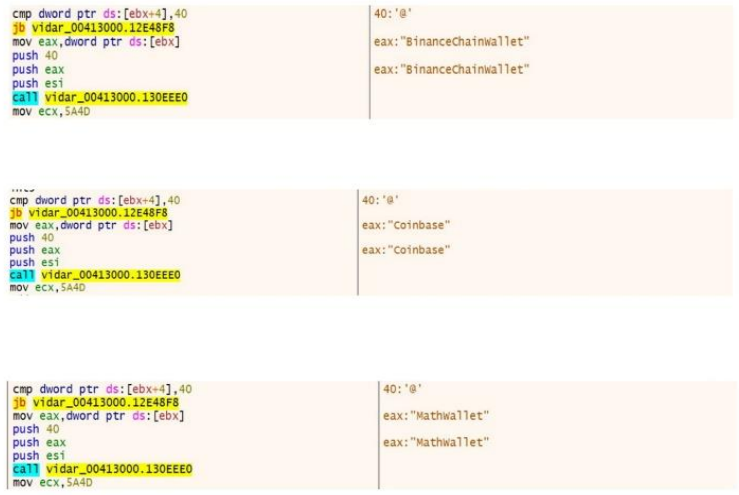



*Şekil 11- Some browser directories used to collect information*

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*




*Şekil 12- Some wallet names used to collect information*

| | | |
|---|---|---|
| EQUALWallet | BitAppWallet | iWallet |
| Wombat | MewCx | GuildWallet |
| RoninWallet | NeoLine | CloverWallet |
| LiqualityWallet | 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 |

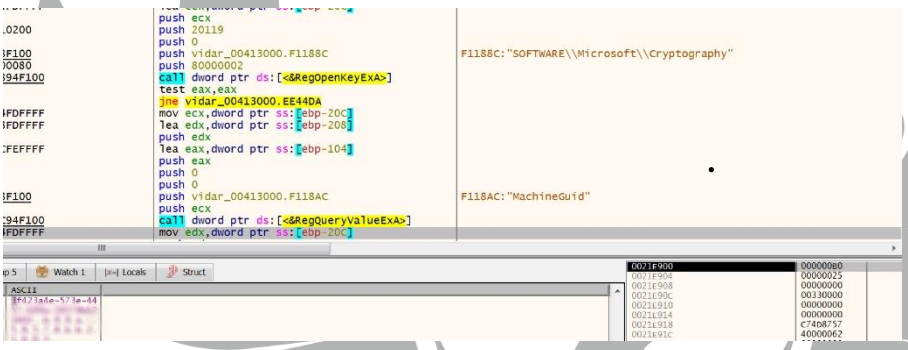
*Tablo 2-Crypto Wallets*

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

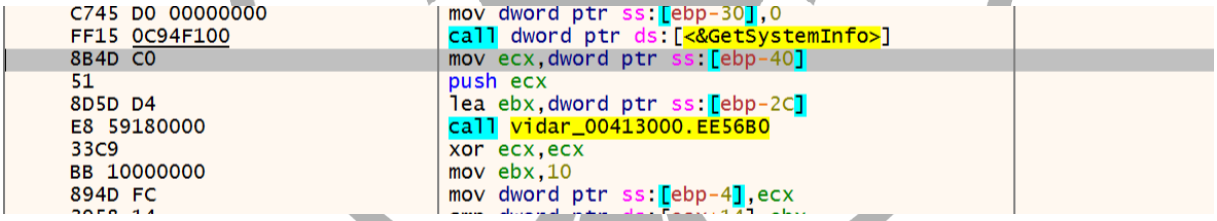
| KeePass Tusk | Trezor Password Manager |
|---|---|
| KeePassXC-Browser | Microsoft AutoFill |

*Tablo 3-Password Managers*

It has been observed that the malware collects system information.

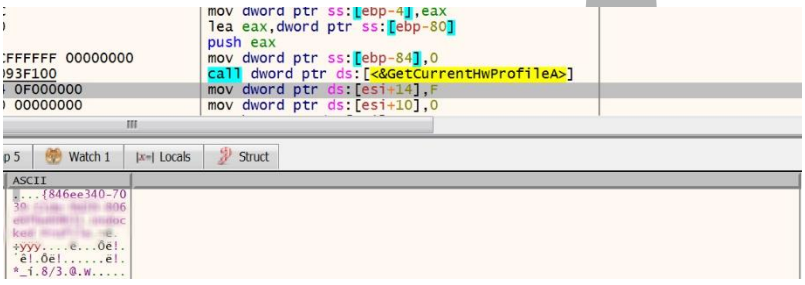



*Şekil 13- Getting MachineGuid information*




*Şekil 14-Using the GetSystemInfo API*

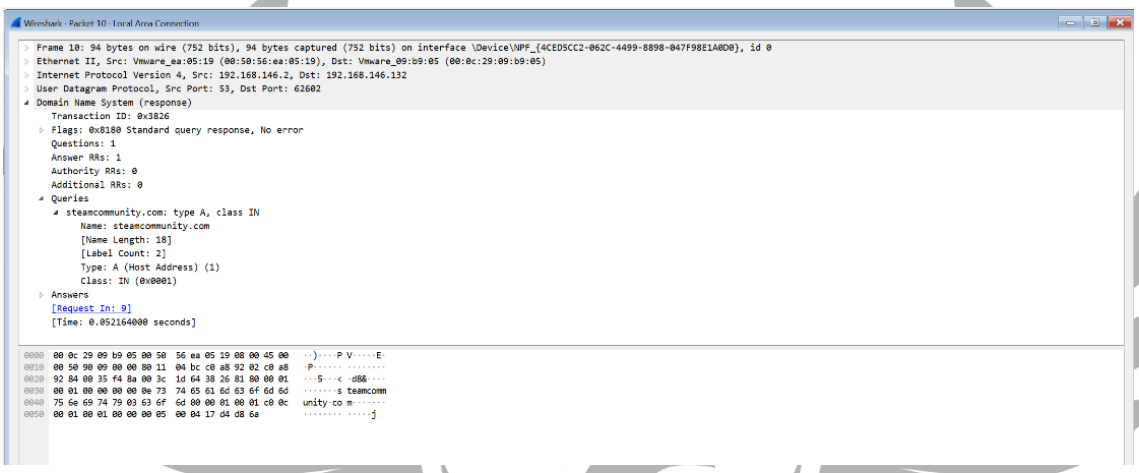
The malware obtains system information such as processor architecture, processor type, number of processors with the **"GetSystemInfo"** API.



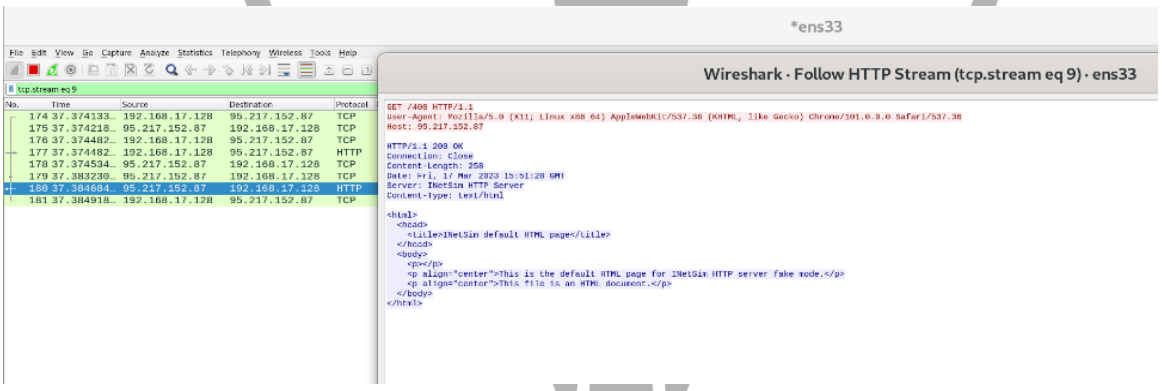

*Şekil 15-Using the GetCurrentHwProfileA API*

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

## Network Analizi




*Şekil 16-DNS request*

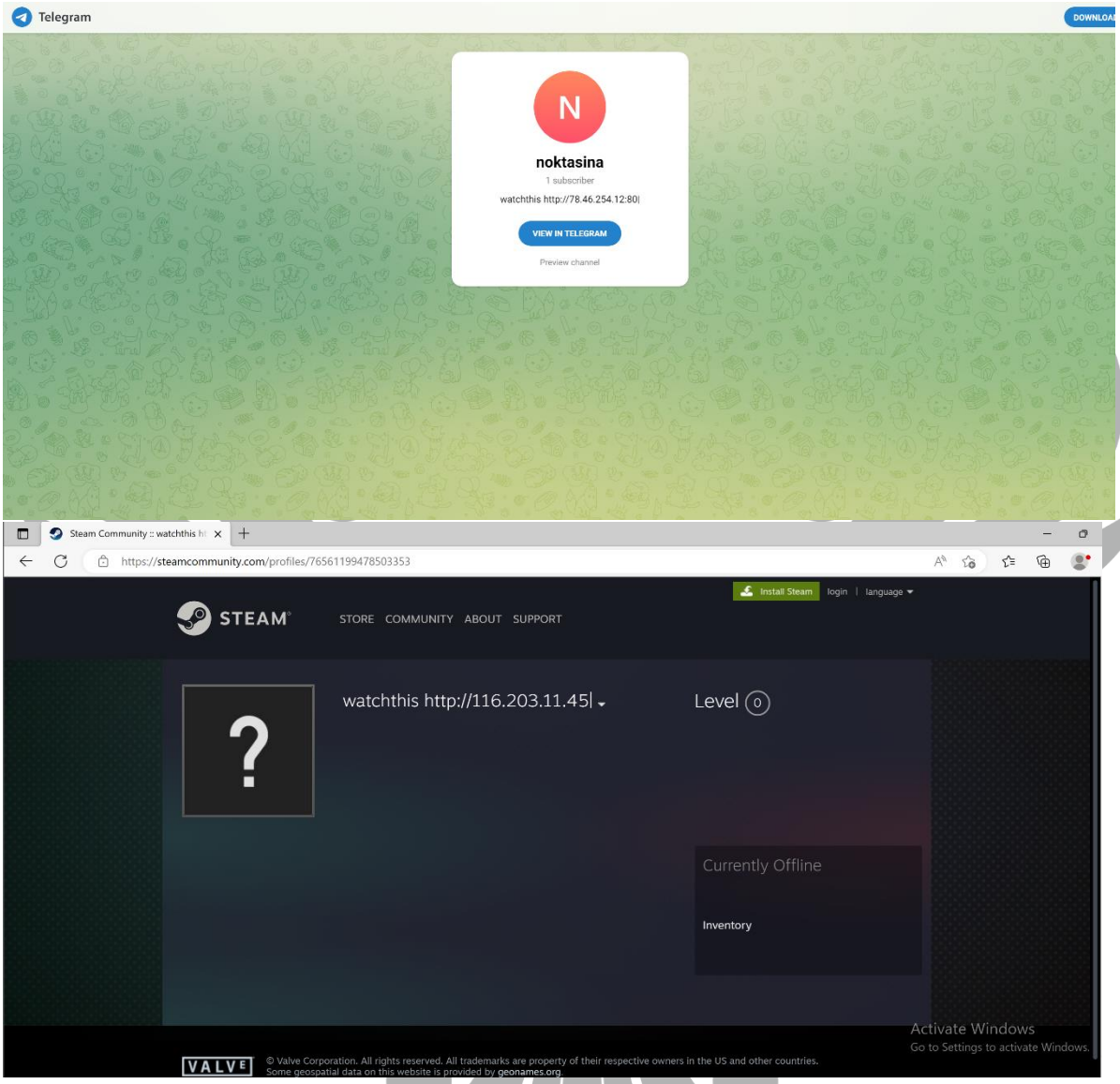



*Şekil 17- HTTP GET request*


*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