

MAAS VIP_Keylogger Campaign

labs.k7computing.com/index.php/maas-vip_keylogger-campaign

Dhanush

March 6, 2026

While surfing through VirusTotal we found an interesting email content persuading the victim to open an attached purchase order which in fact is a RAR file which contained a exe (ÜRÜN ÇİZİMİ VE TEKNİK ÖZELLİKLERİ_xlsx.exe). The .exe file when executed ultimately extracts and executes VIP_Keylogger in memory without touching the disk.

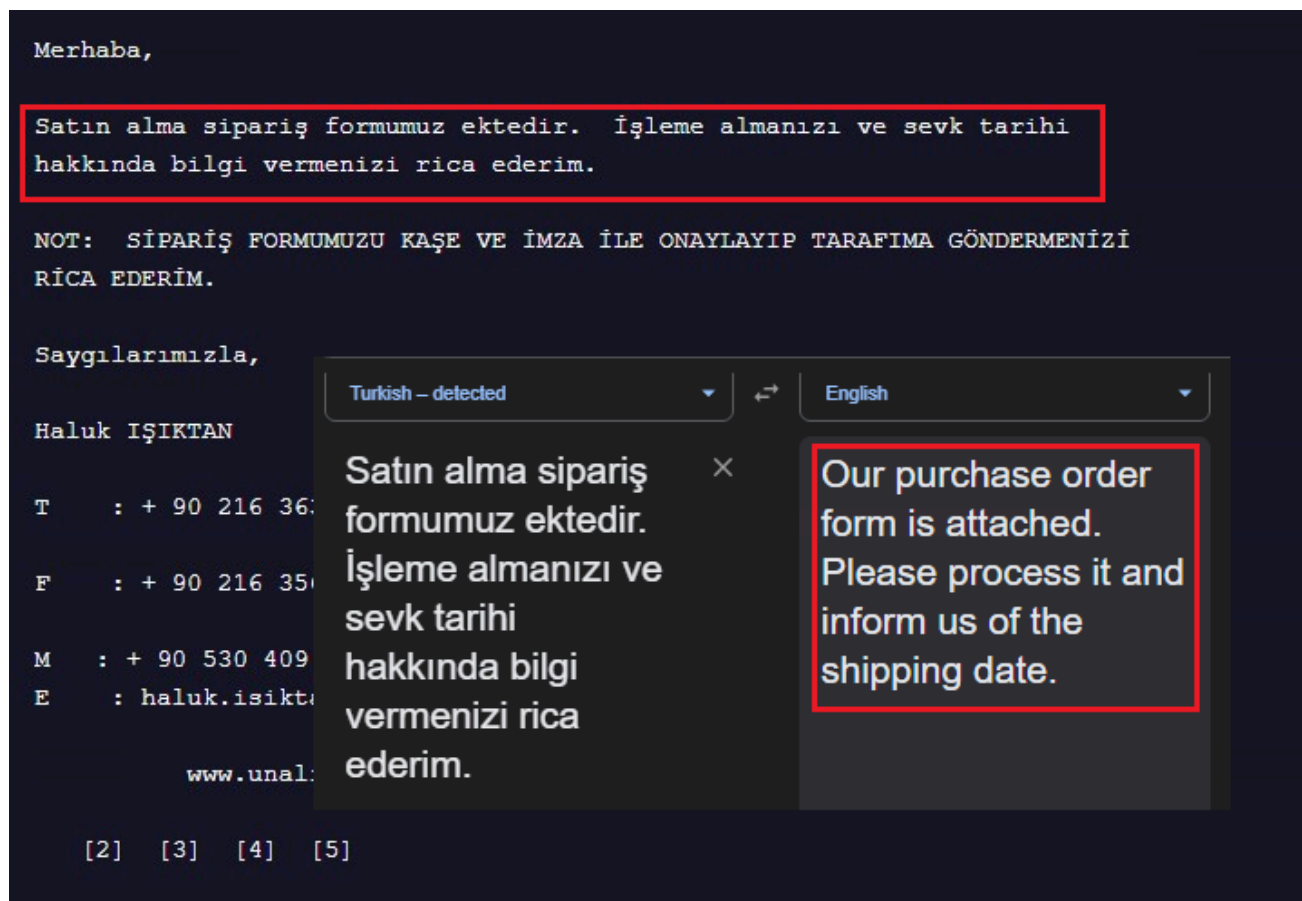


Fig1: Spear-Phishing email

On gathering more information from VirusTotal we noticed multiple instances of this campaign targeting multiple countries with only the packaging style being different and slight differences in its programming flow and execution method.

Let us now delve into the kill-chain of 2 different cases that we noticed.

Kill-Chain

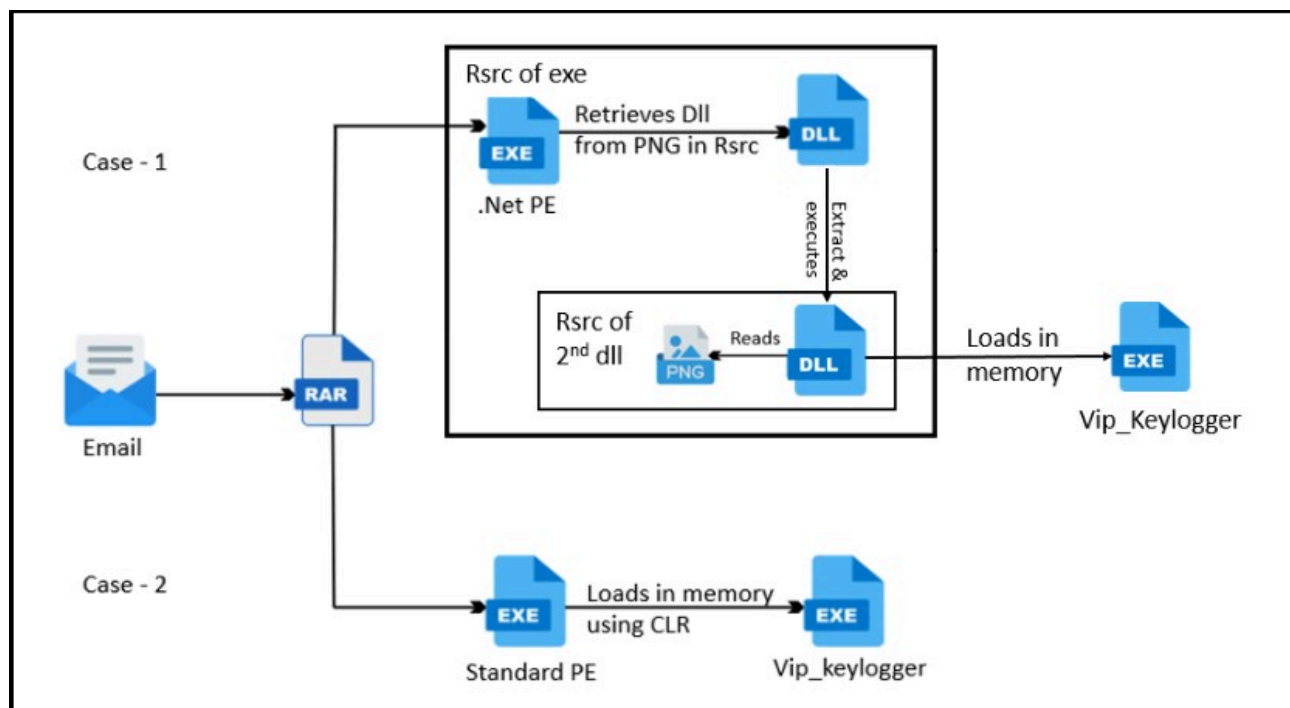


Fig2: Kill chain

Case 1

A .NET PE file containing 2 DLLs hidden in the resource section using steganography technique. One of the DLL (Turboboost.dll) retrieves the second DLL (Vertical bars.dll) from the resource section of the source exe. which in turn contains the final payload in the resource section as a .PNG file (also using steganography). It then retrieves the final payload Vip_Keylogger using the listed api and stores the function pointers in variables and makes a sequential call to do process hollowing on the host exe by creating it again in suspended mode.

The APIs retrieved from Kernel32.dll and Ntdll for process hollowing:

ResumeThread

Wow64SetThreadContext

SetThreadContext

Wow64GetThreadContext

GetThreadContext

VirtualAllocEx

WriteProcessMemory

ReadProcessMemory

ZwUnmapViewOfSection

CreateProcessA

Case 2

In this case, it was a standard PE file containing AES encrypted bytes as the payload present inside the .data section. After decrypting it in memory it patches the AMSI and ETW and gets the required APIs to load the VIP_Keylogger file using CLR.

MAAS Payload

The final payload might either be in the development stage or delivered as MAAS with disabled features as per their client's need because some of the important features such as AntiVM, ProcessKiller, DownloaderFile features are made NULL or cannot be executed. So, we at K7Labs will be getting into the nuances of the available features in this VIP_Keylogger Payload.

```

public static void AntiVm()
{
}

// Token: 0x06000094 RID: 148 RVA: 0x000021DE File Offset: 0x000003DE
public static void DownloaderFile()
{
}

// Token: 0x06000095 RID: 149 RVA: 0x000021DE File Offset: 0x000003DE
public static void AntiSandboxie()
{
}

// Token: 0x06000096 RID: 150 RVA: 0x000021DE File Offset: 0x000003DE
public static void DelayExcute()
{
}

// Token: 0x06000097 RID: 151 RVA: 0x000021DE File Offset: 0x000003DE
public static void FakeMessageBox()
{
}

// Token: 0x06000098 RID: 152 RVA: 0x000021DE File Offset: 0x000003DE
public static void SelfHidden()
{
}

// Token: 0x06000099 RID: 153 RVA: 0x000021DE File Offset: 0x000003DE
public static void OpenWebsites()
{
}

// Token: 0x0600009A RID: 154 RVA: 0x000021DE File Offset: 0x000003DE
public static void DisableWD()
{
}

```

Fig3: Null Features

Features

This Keylogger captures either browser cookies, logins, credit card details, autofills, visited URLs, downloads or top sites from the appropriate files in each of the application's designated folders as listed in the table below and converts it into a SQLite db. It then queries the required data from the DB. If the browser uses AES-256 GCM encryption format it identifies it with the identifier "V10", it decrypts the password by retrieving the master key and then processes it with DPAPIs to retrieve the password. If it were a Legacy browser, it simply decrypts the password with CryptUnprotectdata API.

Chrome	Iridium	Slimjet	Blisk	QIPSurf
Nichrome	Vivaldi	UC	Epic	Sleipnir
Kometa	Iron	Torch	Kinza	Chrome_Canary
CocCoc	Chromium	Brave	BlackHawk	CoolNovo
QQ	xVast	Comodo	Citrio	Sputnik
Orbitum	Ghost	360_China	Uran	Falkon
Amigo	Cent	360_English	Coowon	MicrosoftEdge
Xpom	Chedot	Superbird	7Star	Opera
LieBao				

It then retrieves the logins from the listed Firefox based browsers files and it uses the PK11SDR_Decrypt API from nss3.dll to decrypt their user names and passwords.

CyberFox	PaleMoon	Slim
FireFox	PostBox	Thunderbird
IceDragon	SeaMonkey	WaterFox

```
VIPF_Based.COVIDSTSlim();
VIPF_Based.COVIDSTFireFox();
VIPF_Based.COVIDSTSeaMonkey();
VIPF_Based.COVIDSTIceDragon();
VIPF_Based.COVIDSTCyberFox();
VIPF_Based.COVIDSTPaleMoon();
VIPF_Based.COVIDSTWaterFox();
```

Fig4: Firefox based browser

Emails

This keylogger steals the email accounts from Outlook, Foxmail, ThunderBird and Postbox. In Outlook, it retrieves the profiles from the registry (9375CFF0413111d3B88A00104B2A6676) and searches for the POP3, IMAP, HTTP, SMTP passwords. If found, it proceeds to decrypt the SMTP Server url, email account and password using Unprotect API.

In Foxmail, it checks the foxmail.url.mailto\open\command and then retrieves the directory and from the account.rec0 it retrieves the email and password.

In ThunderBird and PostBox, it retrieves the emails' username and password in the same way as Firefox based browsers by using the PK11SDR_Decrypt Api from nss3.dll.

```
VIPSeassion.COVIDSTOutlook();  
VIPSeassion.COVIDSTFoxmail();  
VIPF_Based.COVIDSTThunderbird();  
VIPF_Based.COVIDSTPostBox();
```

Fig5: Mail theft

Discord & FileZilla & Pidgin

It then collects the Discord token and recently connected Filezilla and Pidgin accounts as these are not encrypted; it would simply search through the appropriate data from each of the targeted applications.

In Discord, it reads the log file from the leveldb folder by searching through “token”. In Filezilla, it reads the recentServers.xml file and searches and retrieves the Host, User, Pass, Port. In Pidgin, it reads the accounts.xml file and retrieves the protocol, username and password.

```
VIPSeassion.COVIDSTDiscord();  
VIPSeassion.COVIDSTFileZilla();  
VIPSeassion.COVIDSTPidgin();
```

Fig6: discord,Filezilla,Pidgin

Exfiltration

It then transfers the stolen information through 5 different ways; FTP, SMTP, Telegram, through the web using POST and via Discord. But the one we analysed has been transferring the information via email from “logs@gtpv[.]online” to “log@gtpv[.]online” via hosting2[.]ro[.]hostsailor[.]com server using the SMTP port 587. The other ways didn't have any of the proper urls or inputs during our analysis.

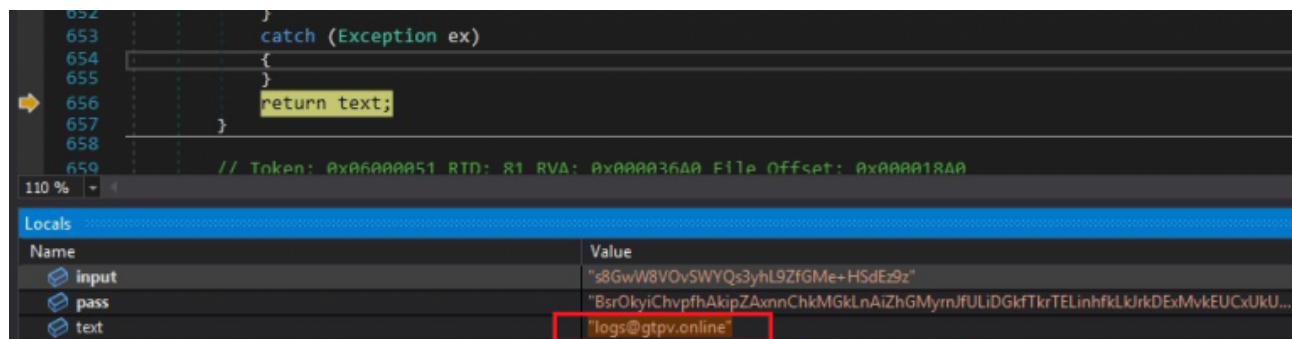


Fig7: Decrypted Email ID

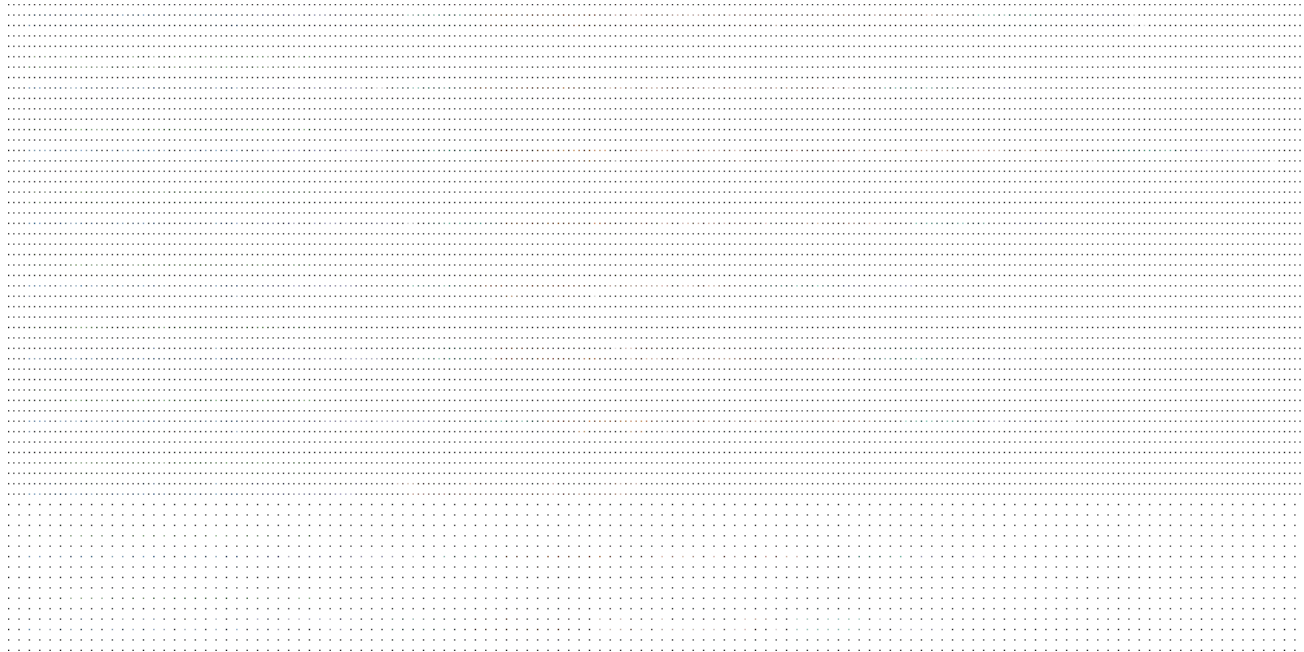


Fig8: List of available urls/IDs

Disabled Features

Apart from the mentioned ones, It can also collect the keylogs along with foreground window, clipboard data, available WiFi-list with passwords, and screenshots but these features have not been executed. There were also 3 domains where no connection had been made by this .exe.

As we see, this kind of campaign is spreading using social engineering techniques while concealing sophisticated malware like “VIP Keylogger”. It is important to use a trusted security product like K7 to safeguard the enterprise resources.

IOCs

Hash	Detection Name
D1DF5D64C430B79F7E0E382521E96A14	Trojan (700000211)
E7C42F2D0FF38F1B9F51DC5D745418F5	Trojan (006d73c21)
EA72845A790DA66A7870DA4DA8924EB3	Trojan (005d5f371)
694C313B660123F393332C2F0F7072B5	Spyware (004bf6371)

C2

varders[.]kozow[.]com

aborters[.]duckdns[.]org

anotherarmy[.]dns[.]army

51[.]38[.]247[.]67