

Summary

In April 2022, ThreatLabz discovered several newly registered domains, which were created by a threat actor to spoof the official Microsoft Windows 11 OS download portal. We discovered these domains by monitoring suspicious traffic in our Zscaler cloud. The spoofed sites were created to distribute malicious ISO files which lead to a Vidar infostealer infection on the endpoint. These variants of Vidar malware fetch the C2 configuration from attacker-controlled social media channels hosted on Telegram and Mastodon network.

ThreatLabz believes that the same threat actor is actively leveraging social engineering to impersonate popular legitimate software applications to distribute Vidar malware, as we have also identified an attacker-controlled GitHub repository which hosts several backdoored versions of Adobe Photoshop. These binaries hosted on GitHub, distribute Vidar malware using similar tactics of abusing social media channels for C2 communication.

In this blog, ThreatLabz analyzes the Vidar distribution vector, threat actor correlation, and technical analysis of the binaries involved in this campaign.

Key points

- ThreatLabz discovered several newly registered domains spoofing the official Microsoft Windows 11 OS download portal
- The spoofed domains were distributing malicious ISO files containing samples of the Vidar infostealer malware
- The actual C2s used by the malware samples are obtained from attacker-controlled social media channels hosted on Telegram and Mastodon network
- Using data obtained from this campaign, ThreatLabz was also able to identify another similar one using backdoored versions of Adobe Photoshop

Distribution Vector - Windows 11 Theme

The threat actor registered several domains beginning 20th April 2022 that host web pages that masquerade as the official Microsoft Windows 11 download page, which is the latest version of the operating system. ThreatLabz found several other domains registered by this threat actor similar to the one shown below in Figure 1. All of these domains were used to spread malicious ISO files spoofed as a Windows 11 download.

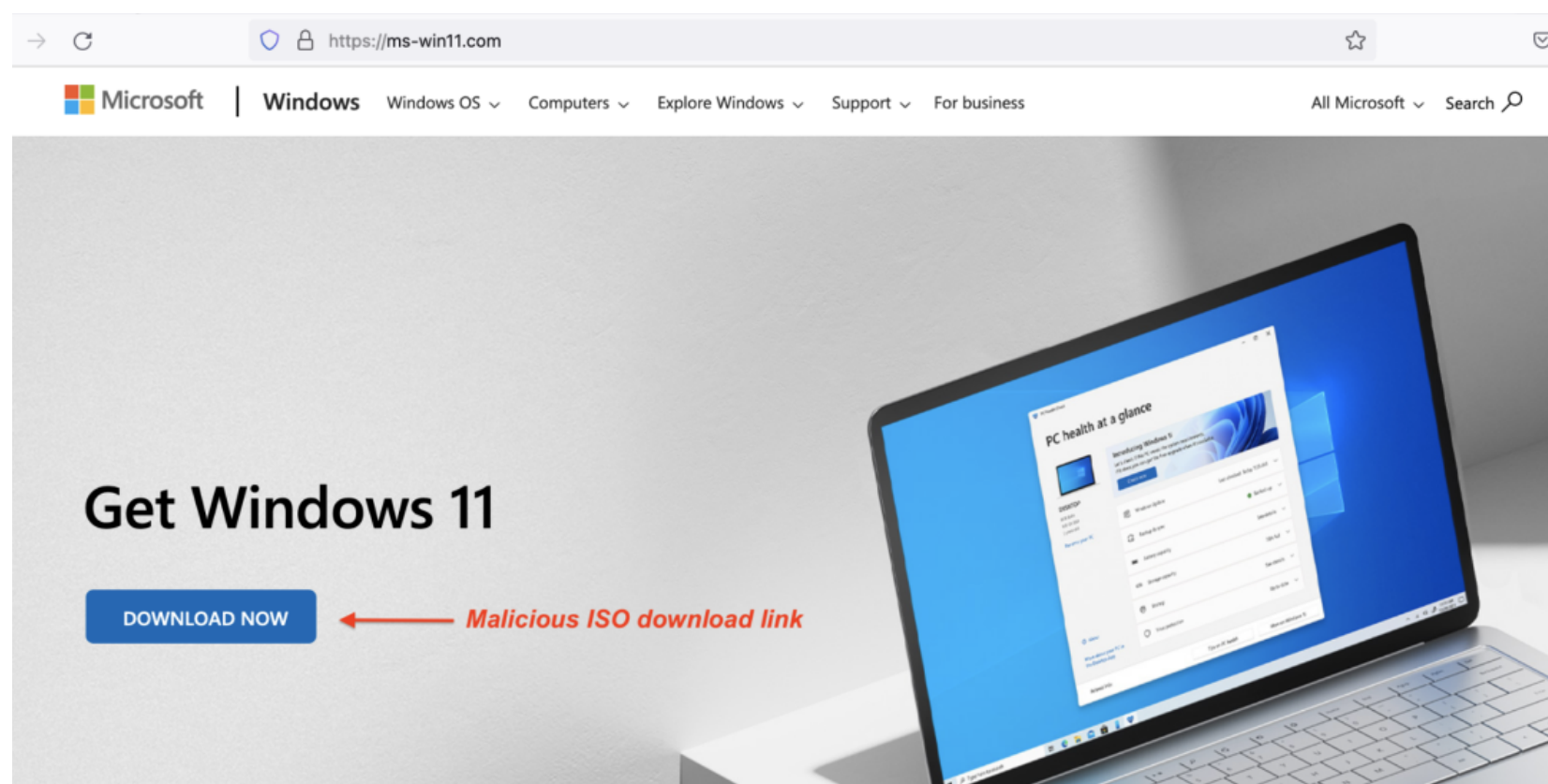


Figure 1: Vidar attacker-controlled domain serving malicious ISO file

The complete list of domains linked to this threat actor that were used in this campaign are mentioned in the Indicators of Compromise (IOC) section.

Technical Analysis

ISO file

The binary inside the ISO file is a PE32 binary. The size of the ISO file is very large (more than 300 MB), which helps the attackers evade network security products where there is a file size limitation in place. Example MD5 hashes for this campaign are shown below:

ISO file MD5 hash: 52c47fdda399b011b163812c46ea94a6 PE32 file MD5 hash: 6352540cf679dfec21aff6bd9dee3770

The binary inside the ISO file is digitally signed with a certificate by AVAST. However, this certificate is expired and hence invalid.

Figure 2 shows the details of the certificate and the corresponding serial number.

Figure 2

Figure 2: Details of the certificate used to sign the malicious Vidar binary

All of the binaries in this campaign were signed by a certificate with the same serial number. By pivoting on this serial number, we were able to discover several other malicious binaries from multiple different campaigns and actors, which likely indicates that this is a stolen certificate coming from the AVAST compromise back in 2019.

Vidar Samples

The Vidar samples in these campaigns are all packed with Themida (except for the MD5 hash 6ae17cb76cdf097d4dc4fcccfb5abd8a) and over 330MB in size. However, the sample contains a PE file that is only around 3.3MB. Figure 3 shows that the rest of the file content is just artificially filled up with 0x10 bytes to increase the file's size. The Vidar strings extracted from these samples is provided in the Appendix section at the end of the blog.

Figure 3

Figure 3: Padding of bytes to inflate the Vidar binary size from 3.3MB to 330MB

All of the binaries below are related to the same Windows 11 theme campaign:

MD5: 6352540cf679dfec21aff6bd9dee3770

The Vidar static configuration below contains the embedded parameters needed by the sample to communicate with its C2 and information including the malware version:

- Profile: 670
- Profile ID: 739
- Version: 51.9
- URL marker: hello
- URL1: https://t.me/btc20220425
- Real C2: 195.201.250.209 (Carved out from URL1)
- URL2: https://ieji.de/@ronxik213
- Real C2: 107.189.11.124 (Carved out from URL2)

The botnet can be identified by its profile ID. Both of the hardcoded URLs are from social media sites. However, they are used as a dead drop resolver as a first stage. The URL marker instructs Vidar to parse the second stage URL from the social media profiles located at the dead drop resolver URL.

The following is an example Vidar stealer configuration downloaded from the C2: 1,1,1,1,1,1,1,1,1,250,Default;%DESKTOP%

.txt:.dat:*wallet*.*:*2fa*.*:*backup*.*:*code*.*:*password*.*:*auth*.*:*google*.*:*utc*.*:*UTC*.*:*crypt*.*:*key*.*;50>true;movies:music:mp3;

This configuration is the default with every stealing function enabled (passwords, cryptocurrency wallets, two-factor authentication, etc)

The following libraries are downloaded from the C2:

- update.zip (66cf4ebdceedcd9214caab7ca87908d), which contains the following DLL libraries:
- freebl3.dll (ef2834ac4ee7d6724f255beaf527e635)
- mozglue.dll (8f73c08a9660691143661bf7332c3c27)
- msvcp140.dll (109f0f02fd37c84bfc7508d4227d7ed5)
- nss3.dll (bfac4e3c5908856ba17d41edcd455a51)
- softokn3.dll (a2ee53de9167bf0d6c019303b7ca84e5)
- sqlite3.dll (e477a96c8f2b18d6b5c27bde49c990bf)
- vcruntime140.dll (7587bf9cb4147022cd5681b015183046)

All of these libraries are legitimate that Vidar leverages in order to extract credentials and other data from different applications and browsers.

MD5: da82d43043c101f25633c258f527c9d5 MD5: e9a3562f3851dd2dba27f90b5b2d15c0

Vidar static configuration:

- Profile: 1281
- Profile ID: 755
- Version: 51.9
- URL marker: hello
- URL1: 5.252.178.50
- URL2: https://koyu.space/@ronxik123
- Real C2: 107.189.11.124 (Carved out from URL2)

For these samples, the URL1 field in the static configuration is a real C2, and a social media profile is used as a backup URL.

The Vidar stealer configuration downloaded from this C2 was the following: 1,1,0,1,1,1,1,0,0,1,250,none;

This configuration is customized to extract social media passwords with all of the other Vidar features disabled.

The libraries downloaded from the C2 are the same as the previous sample with the same update.zip (66cf4ebdceedecd9214caab7ca87908d).

Distribution Vector - Adobe Photoshop Theme

ThreatLabz also identified an attacker-controlled GitHub repository which hosts backdoored versions of the application Adobe Photoshop Creative Cloud, which we attribute to the same threat actor. Figure 4 shows the GitHub repository (<https://github.com/AdobeInstal>) used by the attacker to host a backdoored version of Adobe Photoshop.

Figure 4

Figure 4: Vidar attacker-controlled GitHub repository

Technical Analysis

The sample with the MD5 hash below belongs to this Adobe Photoshop theme campaign.

MD5 6ae17cb76cdf097d4dc4fccfb5abd8a

Vidar static configuration:

- Profile: 1199
- Profile ID: 0
- Version: 51.8
- URL marker: hello
- URL1: https://t.me/mm20220428
- Real C2: 195.201.250.209 (Carved out from URL1)
- URL2: https://koyu.space/@ronxik123
- Real C2: 107.189.11.124 (Carved out from URL2)

The Vidar stealer configuration downloaded from the C2 was the following: 1,1,1,1,1,1,1,1,1,250,Default;%DESKTOP%

.txt:.dat:*wallet*.*:*2fa*.*:*backup*.*:*code*.*:*password*.*:*auth*.*:*google*.*:*utc*.*:*UTC*.*:*crypt*.*:*key*.*;50>true;movies:music:mp3;

The libraries downloaded from the C2 are the same as the previous sample with the same update.zip (66cf4ebdceedecd9214caab7ca87908d).

Social media abuse for C2 communication

All the binaries involved in this campaign fetch the IP addresses of the C2 servers from attacker-registered social media accounts on the Telegram and Mastodon networks. In the past, the threat actors distributing Vidar have abused other social media networks such as Mastodon. However, the abuse of Telegram is a new tactic that they added to their arsenal.

Telegram abuse

In these campaigns, the threat actor created several Telegram channels with the C2 IP address in the channel description. The format used to store the C2 IP address on social media profiles is the following for this campaign:

<C2_Url_Marker> <C2_IP_address>|

The C2_Url_Marker field in these campaigns was hello. The naming convention for the Telegram channels includes a date that corresponds to the date when these channels were created. As an example, the channel with the handle btc20220425 corresponds to a channel created on April 25, 2022, using btc_stacking as the name as shown in Figure 5.

Figure 5

Figure 5: Vidar attacker-controlled Telegram channel with the C2 IP address included in the channel description

Mastodon network abuse

The Mastodon network is a decentralized social network which allows anyone to deploy their own instance of a self-hosted online community. There are several instances of such online communities on the Internet, which are built using Mastodon. Two such instances are ieji[.]de and koyu[.]space. The threat actor created a profile on both of these communities and stored the C2 IP address in the profile section using a format similar to the one used for Telegram channels. Figure 6 and Figure 7 show the profiles created by the threat actor on ieji[.]de and koyu[.]space, respectively.

Figure 6

Figure 6: Vidar attacker-controlled profile on the Mastodon community ieji[.]de with the C2 IP address included in the channel description

Figure 7

Figure 7: Vidar attacker-controlled profile on Mastodon community koyu[.]space with the C2 IP address included in the channel description

Conclusion

The threat actors distributing Vidar malware have demonstrated their ability to social engineer victims into installing Vidar stealer using themes related to the latest popular software applications. As always, users should be cautious when downloading software applications from the Internet and download software only from the official vendor websites. The Zscaler ThreatLabZ team will continue to monitor this campaign, as well as others, to help keep our customers safe.

Zscaler cloud sandbox detection

Figure 8

Figure 8: Zscaler cloud sandbox detection

In addition to sandbox detections, Zscaler’s multilayered cloud security platform detects indicators at various levels.

Win32.Downloader.Vidar Win64.Downloader.Vidar

Indicators of compromise

Hashes

52c47fdda399b011b163812c46ea94a6 da82d43043c101f25633c258f527c9d5 e9a3562f3851dd2dba27f90b5b2d15c0
6ae17cb76cdf097d4dc4fcccfb5abd8a

Domains

ms-win11[.]com ms-win11.midlandscancer[.]com win11-serv4[.]com win11-serv[.]com win11install[.]com ms-teams-app[.]net

URLs for fetching C2 addresses

https://t.me/btc20220425 https://ieji.de/@ronxik213 https://koyu.space/@ronxik123 https://t.me/mm20220428

URLs for fetching ISO files

files.getsnpyer[.]com/files/msteams/Setup.iso files.getsnpyer[.]com/files/windows11/Setup.iso files.getsnpyer[.]com/files/msteamsww/Setup.iso

Actual C2s

195.201.250.209 107.189.11.124 5.252.178.50 107.189.11.124

Appendix

Decoded Strings

Wallets Plugins *wallet*.dat \\Wallets\\ keystore Ethereum\\ \\Ethereum\\ Electrum \\Electrum\\wallets\\ ElectrumLTC \\Electrum-LTC\\wallets\\ Exodus \\\n\\Exodus\\ exodus.conf.json window-state.json \\Exodus\\exodus.wallet\\ passphrase.json seed.seco info.seco ElectronCash \\ElectronCash\\wallets\\ \\\ndefault_wallet MultiDoge \\MultiDoge\\ multidoge.wallet JAXX \\jaxe\\Local Storage\\ file__0.localstorage Atomic \\atomic\\Local Storage\\leveldb\\ \\\n000003.log CURRENT LOCK LOG MANIFEST-000001 0000* Binance \\Binance\\ app-store.json Coinomi \\Coinomi\\Coinomi\\wallets\\ *.wallet \\\n*.config wallet_path SOFTWARE\\monero-project\\monero-core \\Monero\\ SELECT fieldname, value FROM moz_formhistory \\files\\Soft \\files\\Soft\\ \\\n\\Authy \\Authy Desktop\\Local Storage\\ \\Authy Desktop\\Local Storage*.localstorage \\Opera Stable\\Local State INSERT_KEY_HERE JohnDoe \\\nHAL9TH api.faceit.com /core/v1/nicknames/ about Mozilla/5.0 (iPhone; CPU iPhone OS 6_0 like Mac OS X) AppleWebKit/536.26 (KHTML, like \\\nGecko) Version/6.0 Mobile/10A5376e Safari/8536.25 C:\\ProgramData\\ .exe :Zone.Identifier [ZoneTransfer] ZoneId=2 Windows ProgramData \\\nRECYCLE.BIN Config.Msi System Volume Information msdownld.tmp Recovery Local\\Temp Program Files Recycle.Bin All Users MicrosoftEdge\\ \\\n\\Cookies Users\\Public Local\\Packages Local\\NuGet Roaming\\WinRAR Local\\Microsoft Microsoft fee_estimates peers mempool banlist governance \\\nmncache mnpayments netfulfilled passwords.txt Login Data Cookies Web Data \\files\\Autofill \\files\\Cookies \\files\\CC \\files\\History \\files\\Downloads \\\n\\files\\ \\files\\Files hwid os platform profile user cccount fcount telegram ver vaultcli.dll VaultOpenVault VaultCloseVault VaultEnumerateItems \\\nVaultGetItem VaultFree SELECT url FROM moz_places %s\\Mozilla\\Firefox\\profiles.ini \\signons.sqlite SELECT encryptedUsername, \\\nencryptedPassword, formSubmitURL FROM moz_logins \\logins.json formSubmitURL usernameField encryptedUsername encryptedPassword guid \\\nSELECT host, name, value FROM moz_cookies SELECT origin_url, username_value, password_value FROM logins SELECT name, value FROM \\\nautofill SELECT name_on_card, expiration_month, expiration_year, card_number_encrypted FROM credit_cards SELECT target_path, tab_url from \\\ndownloads SELECT url, title from urls SELECT HOST_KEY, is_httponly, path, is_secure, (expires_utc/1000000)-11644480800, name, encrypted_value \\\nfrom cookies C:\\Users\\ \\AppData\\Roaming\\FileZilla\\recentservers.xml <Host> <Port> <User> <Pass encoding=\\ Soft: FileZilla\\n \\AppData\\Roaming\\ \\\n\\purple\\accounts.xml <protocol> <name> <password> Soft: Pidgin\\n \\Thunderbird\\Profiles\\ C:\\Program Files (x86)\\Mozilla Thunderbird APPDATA \\\nLOCALAPPDATA Thunderbird \\files\\Telegram \\Telegram Desktop\\tdata* D877F783D5D3EF8C* \\Telegram Desktop\\tdata\\ key_datas \\Telegram \\\nDesktop\\tdata\\D877F783D5D3EF8C* map* \\Telegram Desktop\\tdata\\D877F783D5D3EF8C\\ firefox.exe plugin-container.exe update_notifier.exe \\\nMozilla Firefox \\Mozilla\\Firefox\\Profiles\\ Pale Moon \\Moonchild Productions\\Pale Moon\\Profiles\\ Waterfox \\Waterfox\\Profiles\\ Cyberfox \\\n\\8pecxstudios\\Cyberfox\\Profiles\\ BlackHawk \\NETGATE Technologies\\BlackHawk\\Profiles\\ IceCat \\Mozilla\\Icecat\\Profiles\\ K-Meleon \\K-Meleon\\ \\\nGoogle Chrome \\Google\\Chrome\\User Data\\ Chromium \\Chromium\\User Data\\ Kometa \\Kometa\\User Data\\ Amigo \\Amigo\\User Data\\ Torch \\\n\\Torch\\User Data\\ Orbitum \\Orbitum\\User Data\\ Comodo Dragon \\Comodo\\Dragon\\User Data\\ Nichrome \\Nichrome\\User Data\\ Maxthon5 \\\n\\Maxthon5\\Users\\ Sputnik \\Sputnik\\User Data\\ Epic Privacy Browser \\Epic Privacy Browser\\User Data\\ Vivaldi \\Vivaldi\\User Data\\ CocCoc \\\n\\CocCoc\\Browser\\User Data\\ URAN \\uCozMedia\\Uran\\User Data\\ QIP Surf \\QIP Surf\\User Data\\ Cent Browser \\CentBrowser\\User Data\\ \\\nElements Browser \\Elements Browser\\User Data\\ TorBro Browser \\TorBro\\Profile\\ Suhba Browser \\Suhba\\User Data\\ Mustang Browser \\Rafotech\\ \\\n\\Mustang\\User Data\\ Chedot Browser \\Chedot\\User Data\\ Brave_Old \\brave\\ 7Star \\7Star\\7Star\\User Data\\ Microsoft Edge \\Microsoft\\Edge\\User \\\nData\\ 360 Browser \\360Browser\\Browser\\User Data\\ QQBrowser \\Tencent\\QQBrowser\\User Data\\ Opera \\Opera Software\\Opera Stable\\ OperaGX \\\n\\Opera Software\\Opera GX Stable\\ Local State Cookies %s_%s.txt TRUE FALSE \\Microsoft\\Windows\\Cookies\\Low\\ Cookies\\IE_Cookies.txt \\\n

\\Packages\\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\\AC\\#!001\\MicrosoftEdge\\Cookies\\ Cookies\\Edge_Cookies.txt \\files\\Wallets %USERPROFILE%
%DESKTOP% KERNEL32.DLL LoadLibraryA GetProcAddress VirtualAllocExNuma gdi32.dll ole32.dll user32.dll psapi.dll BCRYPT.DLL
BCryptCloseAlgorithmProvider BCryptDestroyKey BCryptOpenAlgorithmProvider BCryptSetProperty BCryptGenerateSymmetricKey BCryptDecrypt
CRYPT32.DLL CryptUnprotectData CryptStringToBinaryA C:\\ProgramData\\nss3.dll NSS_Init NSS_Shutdown PK11_GetInternalKeySlot PK11_FreeSlot
PK11_Authenticate PK11SDR_Decrypt advapi32.dll RegOpenKeyExA RegQueryValueExA RegCloseKey RegOpenKeyExW RegGetValueW
RegEnumKeyExA RegGetValueA GetUserNameA GetCurrentHwProfileA wininet.dll InternetCloseHandle InternetReadFile HttpSendRequestA
HttpOpenRequestA InternetConnectA InternetOpenA HttpAddRequestHeadersA HttpQueryInfoA InternetSetFilePointer InternetOpenUrlA
InternetSetOptionA DeleteUrlCacheEntry CreateCompatibleBitmap SelectObject BitBlt DeleteObject CreateDCA GetDeviceCaps CreateCompatibleDC
CoCreateInstance CoUninitialize GetDesktopWindow ReleaseDC GetKeyboardLayoutList CharToOemA GetDC wsprintfA EnumDisplayDevicesA
GetSystemMetrics GetModuleFileNameExA GetModuleBaseNameA EnumProcessModules

- [Security Research](#)
- [Insights and Research](#)
-

Authors

[Sudeep Singh](#)

[Santiago Vicente](#)

[Brett Stone-Gross](#)

Recommended for You

[Peeking into PrivateLoader](#)

[Targeted attack on Thailand Pass customers delivers AsyncRAT](#)

[A "Naver"-ending game of Lazarus APT](#)

[Zscaler ThreatLabz Discovers Multiple Product Bugs in Adobe Acrobat](#)